
JUAN JESÚS ROMERO
CARLOS DAFONTE
ÁNGEL GÓMEZ
FERNANDO JORGE PENOUSAL
(EDS.)

INTELIGENCIA ARTIFICIAL Y
COMPUTACIÓN AVANZADA

Fundación Alfredo Brañas
Santiago de Compostela

CONSEJO EDITORIAL
de la
FUNDACIÓN ALFREDO BRAÑAS

Marcelino	AGIS VILLAVERDE
Carlos	BALIÑAS FERNÁNDEZ
José Gabriel	BARREIRO PÉREZ
Domingo	BELLO JANEIRO
M ^a Xosé	CIMADEVILA CEA
Xosé	CUÍÑA CRESPO
Adolfo	DE ABEL VILELA
Antonio	ERIAS REY
Xerardo	FERNÁNDEZ ALBOR
Manuel	FRAGA IRIBARNE
Xesús	GARRIDO SUÁREZ
Santiago	GONZÁLEZ-BRAÑAS BARREIRO
Raúl	GRIEN DOCAMPO
Xosé L.	IRAVEDRA LESTAL
Manuela	LÓPEZ BESTEIRO
Carlos X.	OTERO DÍAZ
Milagros	OTERO PARGA
Xesús	PALMOU LORENZO
Carlos	PARDO MENÉNDEZ
Xaime	PITA VARELA
Xesús	PRECEDO LAFUENTE (Secretario)
Andrés	PRECEDO LEDO
Pedro	PUY FRAGA
Francisco	PUY MUÑOZ (Presidente)
Xosé A.	QUIROGA Y PIÑEYRO
Mariano	RAJOY BREY
M ^a Teresa	REY BARREIRO-MEIRO
Xosé	RIVAS FONTÁN
Dositeo	RODRÍGUEZ RODRÍGUEZ
Xosé M.	ROMAY BECCARÍA
M ^a Xesús	SÁINZ GARCÍA
Máximo X.	SANZ VALDÉS
Manuel	VARELA REY

JUAN JESÚS ROMERO
CARLOS DAFONTE
ÁNGEL GÓMEZ
FERNANDO JORGE PENOUSAL
(EDS.)

INTELIGENCIA ARTIFICIAL Y
COMPUTACIÓN AVANZADA

Fundación Alfredo Brañas
COLECCIÓN INFORMÁTICA
Número 13/2007

FICHA EDITORIAL

ROMERO CARDALDA, Juan Jesús
DAFONTE VÁZQUEZ, Carlos
GÓMEZ GARCÍA, Ángel
PENOUSAL MARTINS MACHADO, Fernando Jorge (Eds.)
Inteligencia Artificial y Computación Avanzada
Juan Jesús Romero Cardalda, Carlos Dafonte Vázquez, Ángel Gómez García, Fernando Jorge Penousal Martins Machado
Santiago de Compostela: Fundación Alfredo Brañas, 2007. - 400 pág.
- 22 cm. - Índice p. 5. - D.L.: C-2349-07. - ISBN.: 84-934497-0-9
I. Fundación Alfredo Brañas.
II. Autores: Alejandro Pazos, Nieves Pedreira, Juan Ramón Rabuñal, Javier Pereira, Jesús A. Hernández, Julián Dorado, Marcos Gestal, Ana Belén Porto, Julián Flores, Antonio Otero, José Ángel Taboada, Florentino Fernández, Arno Formella, Fernando Jorge Penousal, M^a Luisa Santos, Amílcar Cardoso, Juan Jesús Romero, Víctor Maojo, Fernando Martín, José Manuel Vázquez, José Manuel Cotos, Alfonso Iglesias, Pedro Saco, Bernardino Arcay, Javier Andrade, Santiago Rodríguez, Sonia Suárez, Alejandra Rodríguez, Carlos Dafonte, Iciar Carricajo, Minia Manteiga, José M^a Barreiro, Juan Pazos, Alfonso Rodríguez-Patón, Andrés Silva.
III. Ilustración: Fernando Jorge Penousal
IV. Maquetado: Juan J. Romero, Carlos Dafonte, Ángel Gómez y Fernando J. Penousal
Colección Informática nº 13/2007.

PUBLICACIONES DE LA FUNDACIÓN ALFREDO BRAÑAS
COLECCIÓN INFORMÁTICA
Número 13/2007

Reservados todos los derechos. De conformidad con lo dispuesto en el artículo 534 bis del Código Penal vigente, podrán ser castigados con penas de multa y privación de libertad los que, sin la preceptiva autorización, reprodujesen o plagiasen, en todo o en parte, una obra literaria, artística o científica dispuesta en cualquier tipo de soporte.

Edita Fundación Alfredo Brañas

C/ Ourense, 11- bajo
15701. Santiago de Compostela
fab@infonegocio.com / www.fundacionbranas.org
Copyright: Fundación Alfredo Brañas
Registro: c:\informat.04
Imprime: Tórculo, Artes Gráficas. Santiago
Maquetado: Juan J. Romero, Carlos Dafonte, Ángel Gómez y Fernando J. Penousal
Cubiertas: Ángel Gómez
ISBN.: 84-934497-0-9
Dep.Leg.: C-2349-07

ÍNDICE

Nota de los Editores	7
Alejandro Pazos, Nieves Pedreira, Juan Ramón Rabuñal y Javier Pereira Inteligencia Artificial y Computación Avanzada	9
Jesús A. Hernández, Julián Dorado, Marcos Gestal y Ana Belén Porto Avances en Algoritmos Evolutivos	35
Julián Flores, Antonio Otero y Jose Ángel Taboada Una Aproximación a Sistemas Inmersivos Colectivos Basada en Arquitectura de Bajo Coste	55
Florentino Fernández Predicción No Supervisada de Entornos Cambiantes: Aplicación Práctica al Problema de las Mareas Rojas	85
Arno Formella Búsqueda Aproximada de Patrones de Puntos en 2D y 3D	95
Fernando Jorge Penousal y M^a Luisa Santos Computación Evolutiva y Creatividad	121
Amílcar Cardoso y Juan Jesús Romero Creatividad Computacional	149
Víctor Maojo, Fernando Martín y José Manuel Vázquez Nuevas Perspectivas en Informática Médica y Bioinformática	177
José Manuel Cotos, Alfonso Iglesias, Pedro Saco y Bernardino Arcay Predicción Pesquera Utilizando Algoritmos Basados en I.A.....	203
Javier Andrade, Santiago Rodríguez y Sonia Suárez Un Modo Inteligente de Gestionar el Conocimiento.....	271

Alejandra Rodríguez, Carlos Dafonte, Bernardino Arcay, Iciar Carricajo y Minia Manteiga Técnicas de Inteligencia Artificial para el Tratamiento de Datos Astronómicos.....	287
José M^a Barreiro, Juan Pazos, Alfonso Rodríguez-Patón y Andrés Silva BIOINFORMÁTICA: Genética e Informática	317

NOTA DE LOS EDITORES

La Inteligencia Artificial (I.A.) se puede definir como aquella “inteligencia” exhibida por “cientefactos” o artefactos científicos contruidos por humanos, o sea que dice que un sistema artificial posee inteligencia cuando es capaz de llevar a cabo tareas que, si fuesen realizadas por un humano, se diría de este que es inteligente. Dentro de las ciencias de la computación, la rama de la I.A. se basa en intentar dotar al funcionamiento de las aplicaciones informáticas de un comportamiento inteligente similar al humano para la toma de decisiones.

La I.A. dentro de la computación no es algo reciente ya que sus albores se pueden referir al comienzo de la Cibernética, hace más de 60 años, pero en los últimos años su avance se ha hecho más notable y se ha extendido a numerosas áreas. Así, ha sido llevada a intentar simular comportamientos humanos, no sólo razonamiento, sino también aspectos más difíciles de medir como es el caso de la creatividad o arte artificial. Una prueba de ello es la imagen de fondo que ilustra la portada de este libro, generada por un algoritmo genético.

El objetivo de este libro será presentar una idea general que sirva para acercar al lector a diversos campos de desarrollo dentro la I.A. y la Computación Avanzada, desde una introducción a términos y técnicas, hasta mostrar algunas novedosas aplicaciones específicas que emplean diferentes técnicas de I.A. para solucionar problemas y mostrar comportamientos creativos dentro de las áreas más diversas: arte, medicina, visualización gráfica, pesca, etc.

Estas breves notas no estarían completas sin antes agradecer a los autores de los diferentes capítulos su participación y el esfuerzo dedicado a intentar dar a conocer sus trabajos de investigación. Estamos seguros de que estos temas, de total actualidad y de diferentes temáticas dentro de las Ciencias de la Computación e Inteligencia Artificial, conseguirán atraer la atención del lector.



Inteligencia Artificial y Computación Avanzada

Alejandro Pazos, Nieves Pedreira,

Juan Ramón Rabuñal y Javier Pereira

Laboratorio RNASA-IMEDIR

Facultad de Informática. Universidade da Coruña

Email: {apazos, nieves, juanra, javierp}@udc.es

Introducción

Muchos han sido los avances realizados en la historia reciente para conseguir reproducir en máquinas las capacidades cognitivas del ser humano. La Inteligencia Artificial (en adelante IA) se encarga de canalizar los esfuerzos investigadores en este sentido.

En este capítulo, se presentará una idea general de cómo nació y cómo se desarrolla el campo de la IA, así como de algunas aproximaciones dentro del mismo: técnicas más o menos conocidas, algunas de las cuales serán tratadas más profundamente en otros capítulos, y un acercamiento al maravilloso mundo de la investigación científica a través de los genios que han influido en el estado actual de la IA.

Inteligencia artificial

La IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente. Además, la IA es muy interdisciplinar, y en ella intervienen disciplinas tan variadas como la Neurociencia, la Psicología, las Tecnologías de la Información, la Ciencia Cognitiva, la Física, las Matemáticas, etc.

Se considera que el origen de la IA se remonta a los intentos del hombre desde la antigüedad por incrementar sus potencialidades físicas e intelectuales, creando artefactos con automatismos y simulando la forma y las habilidades de los seres humanos. La primera referencia escrita a un sistema artificial inteligente ocurre en la mitología griega. En ella se atribuye a Hefestos, dios del fuego y las forjas, la fabricación de “dos sirvientes de oro macizo y con inteligencia en su mente y capacidad de hablar”. En la Edad Media, San Alberto Magno construye un “mayordomo” que abría la puerta y saludaba a los visitantes. Ya en la edad moderna (s. XVII), los Droz, famosos relojeros de centroeuropa, construyeron tres androides: un niño que escribía, otro que dibujaba y una joven que tocaba el órgano y simulaba respirar. Esta realización, basada en mecanismos de

relojería, les costó ser detenidos y encerrados por la Inquisición. En cuanto a los avances del Siglo XIX y del Siglo XX, hay que destacar los trabajos de Pascal, Leibnitz, Babbage y Boole. También es importante la aportación de Ada Lovelace, colaboradora de Charles Babbage y mujer de Lord Byron, que en el conocido “régimen de Lovelace” establece lo siguiente: “las máquinas sólo pueden hacer todo aquello que sepamos cómo ordenarles que hagan. Su misión es ayudar a facilitar lo ya conocido”. Esto que está todavía vigente en la “informática convencional” fue superado por los avances en IA. También cabe destacar y es revelador el hecho de que varios de los padres indiscutibles de la ciencia de la computación, como son Turing [3], Wiener [1] y Von Neumann [2], dedicaran un gran esfuerzo al estudio de la inteligencia humana.

La contribución del matemático inglés Alan Turing al mundo de la computación en general, y al de la IA en particular, fue muy considerable. Turing, participó en el diseño de uno de los primeros computadores que existieron, desarrollado para el ejército inglés, entre los años 40 y 50. Además de aportar los conceptos básicos de la arquitectura secuencial de los computadores, publicó en 1950 un provocador artículo que tiene por título “Computer Machinery and Intelligence”, que comienza con la no menos provocadora pregunta: ¿Pueden las máquinas pensar? En dicho artículo el autor intenta abordar formalmente acciones consideradas hasta ese momento propias de los seres humanos, de forma que después pudieran implementarse en las máquinas, dotándolas así de capacidades como: aprender, crear, etc.

En el citado artículo Turing también desarrolla el test que sería central en la definición de un elemento artificial que posea “inteligencia”, el famoso “Test de Turing”, también llamado “Test de Imitación”. En este test se sitúan dos terminales en una sala, uno de ellos conectado a un computador y el otro a un ser humano. Se pretende que un interrogador realice preguntas a través de un puesto conectado a los dos terminales. El test es superado si el interrogador no puede discernir cuál de los dos terminales está conectado a un computador y cuál está conectado a un ser humano. Turing propone que “el juego de la imitación debe aceptarse como un criterio básico. Porque es así como nosotros reaccionamos entre nosotros”.

Si bien el origen inmediato del concepto y de los criterios de desarrollo de la IA se debe a Alan Turing, el apelativo “Inteligencia Artificial” se debe a John McCarthy. McCarthy era uno de los integrantes del “Grupo de Dartmouth” que se reunió en 1956 con fondos de la Fundación Rockefeller para discutir la posibilidad de construir máquinas que no se limitaran a hacer cálculos prefijados, sino operaciones “inteligentes”. Entre los integrantes de este grupo estaban: Samuel, que había escrito un programa de juego de damas capaz de aprender de su propia experiencia; el propio McCarthy, que estudiaba sistemas capaces de efectuar razonamientos de sentido común; Minsky, que trabajaba sobre razonamientos analógicos de geometría; Selfridge, que estudiaba el reconocimiento visual por computador; Newell, Shaw y Simon, que habían construido un programa que pretendía la demostración automática de teoremas; y algunos otros. A partir de este grupo inicial, se formaron dos

grandes “escuelas” de IA. Por un lado, Newell y Simon lideraron el grupo de la Universidad de Carnegie-Mellon, proponiéndose desarrollar modelos de comportamiento humano con aparatos cuya estructura se pareciese lo más posible a la del cerebro, lo que posteriormente derivó en la llamada postura “conexionista” y en los trabajos sobre “Redes de Neuronas Artificiales” también llamadas “Sistemas Conexionistas”. Por otro lado, McCarthy y Minsky formaron otro grupo en el mítico Instituto Tecnológico de Massachussets (MIT), centrándose en que los resultados del procesamiento tuviesen el carácter de inteligente, sin preocuparse si el funcionamiento o la estructura de los componentes eran parecidos o no a los del ser humano. Ambos enfoques sin embargo corresponden a los mismos objetivos prioritarios de la IA: entender la inteligencia natural humana, y usar máquinas inteligentes para adquirir conocimientos y resolver problemas considerados como intelectualmente difíciles.

Sin embargo, en la IA “tradicional” (escuela del MIT), que tuvo mayor auge en los primeros veinte años, los investigadores se encontraron con que sus sistemas sucumbían ante la creciente longitud y complejidad de su programación. Stewart Wilson, investigador del Instituto Roland (Massachussets), se dio cuenta de que algo andaba mal en el campo de la IA “tradicional” y, preguntándose cuáles eran las raíces de la inteligencia, se convenció de que la mejor manera de comprender cómo funciona algo en un ser humano, es comprenderlo primero en un animal más simple. Teniendo en cuenta que, en última instancia, la IA intentaba conseguir una réplica de la inteligencia humana, Wilson decidió que lo primero que había que hacer era replicar la inteligencia animal. Se trataba de una idea que nunca había

tenido mucha popularidad entre los investigadores de IA, pero él y otros investigadores pronto la transformaron en un primer principio informal de un nuevo enfoque de ésta, basado en la naturaleza.

Enfoques de la IA

Se pueden definir dos puntos de vista o enfoques de la IA, el punto de vista tecnológico o ingenieril y el punto de vista científico.

Por un lado, la rama ingenieril busca la creación de sistemas informáticos que realicen tareas para las que se precisa inteligencia. Se persigue desde este planteamiento la resolución de problemas concretos, sin limitar las técnicas a utilizar a aquellas que utilizan los seres inteligentes.

Por otro lado, la rama científica de la IA se puede definir como “el estudio del comportamiento inteligente, siendo su fin conseguir una teoría de la inteligencia que explique la conducta que se produce en seres de natural inteligentes, y que guíe la creación de entes artificiales capaces de alcanzar dicho proceder inteligente” [4].

Las técnicas clásicas, desde el punto de vista tecnológico han tenido un relativo éxito, y sus productos (Sistemas Expertos, Sistemas Basados en el Conocimiento, etc.) se usan ampliamente. El principal problema de estas técnicas radica en que no son capaces de adaptarse a los cambios del entorno y que es preciso tener un conocimiento explícito del problema para poder abordarlo satisfactoriamente. Estos sistemas han de ser programados y no pueden autoprogramarse y adaptarse así a nuevos requisitos del

entorno. Para resolver este problema, se han desarrollado diversas aproximaciones computacionales conocidas de forma global como Técnicas Adaptativas. A continuación se describirán las más utilizadas.

Técnicas adaptativas

Las Técnicas Adaptativas son aquellas que, aplicadas a un problema, son capaces de seguir funcionando adecuadamente a pesar de las circunstancias cambiantes del entorno. Una posibilidad de funcionamiento es ajustar la relativa importancia de los parámetros de entrada de manera autónoma para conseguir la resolución de ese problema. Lo que se persigue con el uso de estas técnicas es conseguir una autonomía en el aprendizaje de las máquinas, lo cual llevaría a uno de los sueños de la IA: la programación automática. Los sistemas en los que se aplican estas técnicas reciben el nombre de Sistemas Adaptativos. A continuación se describen las Técnicas Adaptativas más utilizadas.

Redes de neuronas artificiales

Las “Redes de Neuronas Artificiales” (en adelante RNA) también llamadas “Sistemas Conexionistas” son un sistema de procesamiento de información inspirado en el funcionamiento del sistema nervioso biológico (paradigma de sistema inteligente). Representan un intento de **emular** el comportamiento del cerebro [5] y aumentar nuestros conocimientos acerca de él. Funcionan como un sistema masivamente paralelo que se ha mostrado eficiente para la resolución inteligente de distintos tipos de problemas.

Las RNA constan de elementos básicos de proceso, normalmente organizados en capas (basados en las neuronas biológicas) y con gran cantidad de conexiones entre ellos (basadas en las sinapsis de las neuronas).

Historia

A los precursores de las RNA en las bases biológicas hay que situarlos temporalmente entre los años de 1890 y 1940.

Los fundamentos biológicos del enfoque conexionista se encuentran en el importante trabajo del premio Nóbel español D. Santiago Ramón y Cajal, quién consiguió hace más de un siglo descifrar gran parte de la estructura y del modo de funcionar de las neuronas. Cajal advirtió que el cerebro se reorganiza constantemente, reforzando y debilitando una multiplicidad de sinapsis y su teoría neuronal ha sido fuertemente establecida como la verdadera base de los Sistemas Conexionistas. Ya en 1.895 sienta las bases de lo que llamaba “Ingeniería Neuronal”. Se considera necesario hacer constar dos reflexiones realizadas por Cajal y que conservan hoy en día toda su vigencia: 1) al no encontrar diferencias cualitativas entre las células del sistema nervioso (SN) de los humanos y de los animales, decidió postular que *“la superioridad funcional del SN de los humanos estaría relacionada con la abundancia prodigiosa y con la cuantía considerable de las llamadas neuronas de axón corto”*, y 2) *“el artificio soberano de la sustancia gris es tan intrincado que desafía y desafiará por muchos siglos la obstinada curiosidad de los investigadores”*.

De momento ya han pasado más de cien años y sigue siendo vigente su vaticinio en ambos casos.

Los descubrimientos de Cajal, junto con los del Nóbel italiano Camilo Golgi, el desarrollo posterior de la neurofisiología y trabajos como los de Williams, que en 1890 describe el primer sistema conexionista, Lashley, que es el primero, en 1900, en estudiar cómo almacenar y procesar información utilizando para ello una representación distribuida de la misma, John Eccles o Gerald Edelman, confirman y actualizan las principales hipótesis en las que descansa la rama Conexionista de la IA. Las conexiones entre sus elementos y los cambios de éstas constituyen el factor central de estos sistemas, tratando de emular la estructura del cerebro y su forma de operar. Una “máquina neuronal” se compone de elementos equivalentes a las neuronas y que imitan sus conexiones y comportamiento en red. En cuanto a la forma de operar, imita el proceso de aprendizaje relacionado con el cambio de estado de las conexiones entre las neuronas. De este modo, una “máquina neuronal” no se programa en la forma tradicional, traduciendo algoritmos en secuencias de órdenes y operaciones, sino que se ajusta progresivamente en función del uso, a modo de proceso de aprendizaje socrático, esto es, a partir de ejemplos.

Como precursores computacionales de las RNA cabe destacar a los investigadores McCulloch, médico de la Universidad de Illinois, y Pitts, matemático del MIT, quienes proponen en 1.943 el primer modelo de neurona artificial.

Las RNA nacen como consecuencia de la aparición de tres trabajos teóricos relativos a lo que ahora se conoce como “cibernética” y que, curiosamente, se publican todos en el mismo año de 1.943:

“Behaviour, Purpose and Teleology” de Rosenblueth, Wiener y Bigelow, en el MIT, donde sugieren distintas formas de conferir fines y propósitos a las máquinas, es decir, hacerlas teleológicas.

“The Nature of Explanation” de Craik, en la Universidad de Cambridge, quien propone que las máquinas empleen modelos y analogías en la resolución de problemas, i.e., establecer la capacidad de abstracción de las máquinas.

“A Logical Calculus of the Ideas Inmanent in Nervous Activity” de McCulloch y Pitts, los cuales pusieron de manifiesto de qué modo las máquinas podían emplear los conceptos de la lógica y de la abstracción, y demostraron cómo cualquier ley de entrada-salida podía modelizarse con una red de neuronas formales por ellos definidas.

En 1949 Donald Hebb introduce dos ideas fundamentales para las RNA: 1) una percepción o un concepto se representa en el cerebro por un conjunto de neuronas activas simultáneamente y 2) la memoria se localiza en las conexiones entre las neuronas (sinapsis) [7]. Además, Hebb propone un sistema de aprendizaje para la modificación de la sinapsis denominado “Regla de Aprendizaje Sináptico” o “Regla de Hebb”, que consiste en que una conexión entre neuronas se refuerza cada vez que es utilizada.

A finales de los 50, Widrow diseña el Adaline (Neurona Adaptativa Lineal) y Rosenblatt el Perceptrón [8], una máquina con un comportamiento adaptativo capaz de reconocer patrones, dotada de la regla del aprendizaje denominado de “autoasociación”, donde el estímulo y la unidad de respuesta están asociados por la acción de las entradas.

En 1969 Minsky y Papert publican “*Perceptrons: An Introduction to Computation Geometry*”, donde realizan una seria crítica del Perceptrón, debido principalmente a su naturaleza lineal [9].

A partir de aquí existe un declive en las RNA, debido a que surgen algunas demostraciones de que no era factible crear una teoría automatizada de la inteligencia. Pero en 1980 surge un renacimiento de las RNA, con investigadores como Hopfield, que construyen nuevos modelos de células, arquitecturas y algoritmos de aprendizaje que servirán de base a los modelos más investigados actualmente, los denominados “Modelos Biológicos”. Hopfield presenta, con un fuerte fundamento matemático y de forma coherente, el modo de trabajar de los modelos de RNA y establece que dichas redes deben ser primero modelos cerebrales y luego dispositivos útiles de procesamiento. Fue clave su claridad y el análisis matemático en los modelos analizados.

Actualmente, las RNA son un campo de gran interés debido, por un lado, a la habilidad de estos sistemas para aprender automáticamente y para poder funcionar de forma aceptable tanto en presencia de información inexacta como cuando se producen deterioros o fallos en sus componentes, y por otro, al interés existente por la búsqueda de arquitecturas de

computadoras que permitan el procesamiento en paralelo, de forma similar a los modelos neurofisiológicos del cerebro, pudiéndose de este modo intercambiar descubrimientos entre los modelos de RNA y los modelos en Neurociencia.

Algoritmos genéticos

Los Algoritmos Genéticos (en adelante AG) son algoritmos de búsqueda inspirados en procesos de selección natural, basados en la teoría de la evolución de Darwin. Establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural. Se aplican principalmente en problemas de optimización y se comportan de un modo muy eficaz en problemas de superficie compleja, con múltiples mínimos locales y grandes espacios de búsqueda.

Está justificado su uso en aquellos problemas cuya complejidad no permita una solución directa. Por ejemplo los no resolubles polinomialmente (NP-duros).

Los fundamentos de los AG se encuentran detrás de la idea de que la naturaleza utiliza potentes medios para impulsar la evolución satisfactoria de los organismos. Los organismos que son poco aptos para un determinado ambiente mueren, en tanto que los que están bien adaptados para vivir, se reproducen. Ocasionalmente se producen mutaciones al azar y, aunque la mayor parte de éstas implican la pronta muerte del individuo mutado, algunas dan como resultado nuevas y satisfactorias especies. Así, los AG se encuentran dentro de las llamadas técnicas de Computación Evolutiva.

Los AG trabajan sobre poblaciones de individuos que representan soluciones al problema. Para ello utilizan las operaciones de **selección**, **cruce** y **mutación**, propias de la genética. De esta manera, mediante individuos se codifican diferentes soluciones al problema, inicialmente aleatorias. Luego estos individuos se agrupan en una población y se van combinando entre sí de forma que a medida que pasan las generaciones se van ajustando más a la solución final. Los individuos de una generación genética compiten entre sí para ofrecer la mejor solución al problema que se está tratando de resolver [11] [12].

Historia

Los primeros intentos para relacionar informática y evolución, a finales de los 50 y principios de los 60, tuvieron poco éxito porque, basándose en los textos de biología del momento, se creía en la mutación en lugar del cruce para producir nuevas combinaciones de genes. Fue a principios de los 60, cuando Hans J. Bremermann de la Universidad de California-Berkeley añadió una nueva forma de cruce. En ella, las características de la descendencia se determinaban sumando los genes correspondientes de los dos progenitores. Lamentablemente, este procedimiento de cruce estaba limitado, porque sólo se podía aplicar a características que se pudieran sumar de alguna manera significativa.

En esta época, John Holland, considerado el padre de los AG, había estado investigando en el análisis matemático de la adaptación y se había convencido de que la recombinación de grupos de genes mediante el cruce,

era la parte más importante de la evolución. Holland se preguntaba cómo lograba la naturaleza crear seres cada vez más perfectos. A principios de los 60 se dio cuenta de que la evolución era una forma de adaptación más potente que el simple aprendizaje y tomó la decisión de aplicar esta idea para desarrollar programas bien adaptados para un fin determinado. A mediados de los 60 había desarrollado una nueva técnica de programación, el **algoritmo genético**, que se adaptaba a la evolución tanto por cruce como por mutación.

David Goldberg, discípulo de Holland y uno de los líderes actuales en AG, fue uno de los primeros que trató de aplicar los esta técnica a problemas industriales.

Programación genética

John Koza, en 1992, adaptó los AG para la creación de programas de computador, permitiendo que las técnicas de Computación Evolutiva se utilizaran para construir grandes poblaciones de programas que evolucionan y finalmente obtienen, de forma automática, un programa que constituye la solución a un problema dado. Él mismo bautizó a esta nueva técnica como “Genetic Programming” o “Programación Genética”.

Así, la Programación Genética (en adelante PG) surge como una evolución de los AG tradicionales, manteniendo el mismo principio de selección natural. Es una técnica de búsqueda en la que, a partir de un conjunto inicial de programas (población), éstos se van combinando sucesivamente, formando distintas generaciones para dar lugar a programas

mejores. Al igual que los AG, la PG también forma parte de las denominadas técnicas de Computación Evolutiva.

La mayor diferencia entre los AG y la PG es la forma de codificación de la solución al problema. En PG la codificación se realiza en forma de árbol, de forma similar a cómo los compiladores leen los programas según una gramática prefijada.

La codificación de cada solución particular se realiza en términos del propio problema. En los AG tradicionales era necesario realizar una tarea de codificación/decodificación, pero en la PG esto ya no es necesario, puesto que la solución buscada va a ser encontrada, como ya se dijo previamente, como un algoritmo en forma de árbol.

La PG proporciona una gran versatilidad y variedad de problemas resolubles y, además, permite la obtención de soluciones de forma jerárquica.

En definitiva, la PG es muy potente pero también compleja, ya que, entre otras exigencias, requiere especificar el juego de instrucciones, con el fin de delimitar el problema. A pesar de su complejidad es legible, robusta y versátil y se utiliza mucho en robótica y optimización.

Otras técnicas adaptativas

Hardware evolutivo

Tras esta técnica subyace la idea del desarrollo de máquinas y robots que sean capaces de replicarse a partir de componentes simples y adapten su comportamiento a fallos.

Así, se pretende que el Hardware Evolutivo sea capaz de autoadaptarse en tiempo real, reconfigurando su arquitectura de manera dinámica y autónoma, mediante una interacción con el ambiente. Se utilizan técnicas evolutivas para la síntesis de circuitos robustos al ruido y a los cambios, como mecanismo de adaptación al ambiente.

Learning classifier systems

Los científicos han soñado durante años con conseguir máquinas que pudiesen **reaccionar con el entorno**, de manera semejante a como los humanos lo hacen. Los sistemas clasificadores “con aprendizaje” o Learning Classifier Systems (en adelante LCS), pueden considerarse como una aproximación con AG al aprendizaje a partir de la interacción con el entorno. Aunque esta definición tal vez esté un poco obsoleta, ya que los LCS han evolucionado empleando técnicas muy diferentes.

En general, un LCS toma un conjunto de entradas y produce un conjunto de salidas que indican una cierta clasificación de las entradas. Un LCS “aprende” cómo clasificar sus entradas. Esto implica a menudo

“mostrar” al sistema muchos ejemplos de los patrones de la entrada, y sus salidas correctas correspondientes.

Esta técnica se puede situar en un punto intermedio entre los Algoritmos Genéticos y los Sistemas Conexionistas.

Ant colony simulation

Es una meta-heurística (técnica que sirve como apoyo a otra) inspirada en el comportamiento de colonias reales de hormigas.

Para su implementación, existe una población de agentes simples que producen en su conjunto un comportamiento global más complejo. Simulan el comportamiento de las hormigas, las cuales se unen para conseguir objetivos que no podrían lograr de forma individual. Las hormigas dejan un rastro de “feromonas” por donde pasan. Esto les permite por ejemplo encontrar el camino más corto hasta una fuente de comida. Cuando una hormiga encuentra una fuente de comida, coge la que puede, la lleva al hormiguero y vuelve a por más. La hormiga que encuentra el camino más corto puede hacer más “viajes”, lo cual incrementa el nivel de feromona de ese camino en particular. Esto atraerá al resto de las hormigas, que seguirán el camino con un rastro mayor de feromonas, lo cual reforzará todavía más la ruta, por lo que en poco tiempo todas las hormigas seguirán el camino más corto hasta la fuente de comida y se habrá conseguido una optimización del proceso.

Esta técnica ofrece resultados muy interesantes en problemas de optimización combinatoria.

ADN molecular

En esta técnica se utiliza el ADN de seres vivos para resolver problemas computacionales.

Es una técnica masivamente paralela y muy prometedora, pero en la actualidad el procesado de la información (probetas con piezas de ADN) es realizado por el investigador y es muy laborioso y lento.

Coevolución

Consiste en el uso de varias poblaciones dependientes mediante técnicas evolutivas.

La relación típica es la de depredador-presa. Esto es, en paralelo, se entrena una población de depredadores y otra de presas. Así, a medida que los depredadores mejoran, las presas se ven obligadas a perfeccionarse y viceversa.

Artificial immune systems

Estos sistemas están inspirados en el Sistema Inmune Natural. Son sistemas masivamente paralelos. Permiten detectar situaciones no habituales, proporcionar una memoria direccionable por contenido o

distinguir entre entradas típicas y anormales. Pueden utilizarse en problemas de optimización y búsqueda.

Vida artificial

La Vida Artificial trata acerca de la creación de modelos computacionales con comportamientos biológicos. Se emplean múltiples agentes simples para conseguir un comportamiento complejo.

Ronald ofrece una de las definiciones más completa: “*la Vida Artificial es una disciplina experimental que consiste fundamentalmente en la observación de los comportamientos de ejecución, aquellas interacciones complejas generadas cuando las poblaciones de criaturas artificiales hechas por el hombre están inmersas en entornos reales o simulados*” [13].

En muchos casos se busca determinar los mecanismos de interacción entre los individuos de una colectividad que hacen emerger comportamientos adaptativos o inteligentes al nivel de toda la colectividad de organismos.

IA y tecnologías convergentes

Como se explicó al comienzo de este capítulo, la IA es una rama de la ciencia en la que intervienen múltiples disciplinas y áreas de conocimiento y como tal debe ser entendida para comprender la importancia de la integración de los conocimientos de dichas disciplinas en aras de conseguir nuevos avances.

La ciencia en general y la IA en particular han alcanzado un punto en el cual varias de sus disciplinas deben unificarse para seguir avanzando rápidamente. La convergencia de tales disciplinas puede iniciar un nuevo renacimiento, incluyendo una visión integral de la tecnología basada en herramientas de transformación, en las matemáticas de sistemas complejos, y en un entendimiento unificado “causa-efecto” del mundo físico que abarca desde la nanoescala a la escala planetaria.

Así, los adelantos conseguidos en campos aparentemente tan diferentes como la Biología, la Física, las Ciencias de los Materiales, la Microelectrónica o los Microsistemas, convergen en un punto: el estudio de entidades de dimensiones físicas similares, comparables a los tamaños moleculares. Este punto común facilita e impulsa la sinergia entre los conocimientos científicos y tecnológicos que lo sustentan, dando lugar a la aparición de las denominadas “Tecnologías Convergentes” (NBIC, por Nano y Microtecnologías, Biotecnología, Tecnologías de la Información y Ciencias del Conocimiento).

La integración y sinergia de las cuatro tecnologías (nano-bio-info-cogno) tienen su origen en la nanoescala, dónde se establecen los bloques de construcción de la materia.

Las NBIC aparecen hoy en día como una tendencia tecnológica con un gran impacto desde los puntos de vista científico, tecnológico, económico y social. La convergencia tecnológica hará posible en un futuro el uso indistinto y combinado de los diferentes “elementos básicos” (átomos, células, genes, “nanodispositivos” y “nanopartículas”, agentes,

etc.) como unidades constituyentes de la arquitectura de verdaderos sistemas híbridos.

El enorme valor añadido de estas Tecnologías Convergentes proviene no tanto del desarrollo de cada una de ellas sino del conocimiento y la experiencia necesarios para su uso combinado. Si bien es cierto que estos campos se han estudiado tradicionalmente desde ángulos diferentes (Biología, Física, etc.), el verdadero impacto de los mismos provendrá del efecto multiplicador de su desarrollo y utilización conjunta.

Se prevé el nacimiento y el rápido desarrollo de una combinación sinérgica entre Nanotecnología, Biotecnología, Informática y Ciencias del Conocimiento que requerirá un nuevo planteamiento de los diferentes campos del conocimiento humano y que llevará a desarrollos tecnológicos absolutamente innovadores. En un futuro inmediato estas tecnologías conjuntas aportarán mejoras en la calidad de vida así como soluciones a una realidad social, económica y geográfica en sectores tan relevantes como la salud, agricultura y ganadería, medio ambiente, y otros muchos.

Las Tecnologías Convergentes (NBIC) tendrán claras implicaciones en áreas clave de la actividad humana, incluyendo: el trabajo, el aprendizaje, las interrelaciones personales, y la evolución humana. Avanzando simultáneamente en varias de estas actividades se podría alcanzar una era de innovación y prosperidad que constituiría un punto de inflexión en la evolución de la sociedad humana.

Aplicaciones de las NBIC

Las NBIC tendrán infinidad de aplicaciones en el futuro, pero en este capítulo nos centraremos en los avances que las Tecnologías Convergentes proporcionarán en los campos de Ciencias de la Salud y Ciencias de la Computación.

Avances en ciencias de la salud: Integración de nano-bio-info-cogno-chips

El principal avance de las NBIC en este campo provendrá de la posibilidad de desarrollar dispositivos electrónicos de tamaño tan reducido que podrán integrarse con los organismos vivos, incluso a nivel celular. De este modo será posible utilizar “nanochips”, por ejemplo, en el procesamiento de la sinapsis en el sistema nervioso central. Los nanochips podrán entonces reemplazar a los circuitos cerebrales. Estos chips contendrán o se conectarán con Redes de Neuronas Neurogliales (Artificial NeuroGlial Networks, ANGN) que podrán controlar la actividad sináptica o regenerar células, en función del potencial de acción (Action Potential, AP) grabado, sustancias entregadas, etc. Así, podrán inhibir acciones excitadoras de neuronas presinápticas en una sinapsis, generar AP excitador, etc. Podrán también liberar o inhibir neurotransmisores u otras sustancias en áreas dañadas. Estas sustancias podrían almacenarse dentro de nano-buffers hechos de bio-tejidos. Además, permitirán reparar neuronas dañadas, introduciéndoles material genético. De esta forma las neuronas dañadas podrán sintetizar sustancias de nuevo (neurotransmisores, proteínas de

canales dañados, etc.). Así, los nanochips serán útiles en el tratamiento de enfermedades como Alzheimer, Parkinson, etc.

Los nanochips permitirán desarrollar nuevas técnicas, como la Nano-electroneurografía o la Nano-electromiografía, en las que un nanochip, con una ANGN asociada, se conectaría a un nervio o a un músculo. El chip tendría además un nanoelectrodo grabador y un nanoelectrodo de estimulación, y se conectaría a una batería. La ANGN compararía las señales grabadas por el nanoelectrodo en el nervio o en el músculo con las señales óptimas. El nanochip podría, por ejemplo, enviar datos a un PC o PDA (a través de una conexión inalámbrica) y generarse entonces un informe para el paciente o para el doctor. Esto sería muy útil en diagnóstico. También podría utilizarse un nanochip para generar AP de cara a estimular la neurona postsináptica o el músculo. Esto sería muy útil en el tratamiento. Otra posibilidad sería utilizar nanochips para introducir material genético con el objetivo de sintetizar alguna proteína (Oligodendrocyte & Schwann). Esto sería muy útil en el tratamiento de enfermedades como la esclerosis múltiple. Por último, un nanochip conectado con biosensores podría sustituir neuronas sensitivas.

Avances en ciencias de la computación

Las NBIC permitirán mejorar el comportamiento de los computadores con el uso de sustancias biológicas. Se podrán, por ejemplo, utilizar proteínas para crear “biocomputadores” que superen las actuales limitaciones de la tecnología del silicio, o desarrollar interfaces biológicas

que remitan directamente sus percepciones al computador. También se puede pensar en “nanorobots” construidos con carbono, silicio y oxígeno, para capturar imágenes en el interior del cuerpo humano. Tales robots podrían analizar las imágenes y compararlas con las óptimas mediante ANGN (reconocimiento de patrones, etc.).

Además, el desarrollo de nuevos modelos computacionales basados en los circuitos cerebrales permitirá crear nuevos modelos de sistemas adaptativos, como las ciudades ANGN. Los descubrimientos hechos con la ayuda de estos modelos permitirán comprender mejor la estructura y funcionamiento del cerebro. Esto permitirá desarrollar nuevas y mejores técnicas de IA. Será entonces posible de fabricar dispositivos protésicos neurales. Por ejemplo, vista y oído artificiales, creados a partir de ANGN con información de nanoelectrodos grabando en el cortex motor, visual, auditivo, etc., entrenadas para visualizar imágenes, para escuchar sonidos, o bien para permitir realizar movimientos de forma precisa con el cursor en un monitor, para mover un brazo-robot, etc.

Así, las Tecnologías Convergentes permitirán nuevos e importantes avances en la IA y, más allá de eso, permitirán la integración de los mismos con los seres vivos, consiguiendo así verdaderos sistemas híbridos.

Bibliografía

- [1] Wiener, N., 1985. "*Cibernética*". Tusquets editores, ISBN: 84-7223-452-5.
- [2] Neumann J.V., 1996. "*The Theory of Self-Reproducing Automata*". En Burks, A.W (Ed). University of Illinois Press. Illinois.
- [3] Turing, A., 1950. "*Computing Machinery and Intelligence*". En MIND, nº 236, pp. 433-460.
- [4] Gómez, A., Juristo, N., Montes, C., Pazos, J., 1997. "*Ingeniería del Conocimiento*". Centro de Estudios Ramón Areces S.A. Madrid.
- [5] Pazos, A., Col., 1991. "*Estructura, dinámica y aplicaciones de las Redes de Neuronas Artificiales*". Centro de Estudios Ramón Areces S.A. Madrid.
- [6] McCulloch, W., Pitts, W., 1943. "*A Logical Calculus Of The Ideas Inmanent In Nervous Activity*". Bull. Mathem. Biophys, nº 7, pp. 115-133.
- [7] Hebb, D.O., 1949. "*The organization of Behavior*". John Wiley & Sons.

- [8] Rosenblatt, F., 1962. *"Principles of Neurodynamics"*. New York: Spartan Books.
- [9] Minsky, M., Papert, S., 1969. *"Perceptrons: An Introduction to Computation Geometry"*. MIT Press. Cambridge MA.
- [10] Holland, J., 1962. *"Outline for a Logical Theory of Adaptive Systems"*. Journal of ACM, nº 9, pp. 297-314.
- [11] Holland, J., 1975. *"Adaption in Natural and Artificial Systems"*. The University of Michigan Press. Ann Arbor. Michigan.
- [12] Holland, J., 1984. *"Sistemas Adaptativos Complejos"*. En Pazos, A., "Redes de neuronas artificiales y algoritmos genéticos" (pp. 259-297). Universidade da Coruña. A Coruña.
- [13] Ronald, E., Sipper, M., Capcarrère, M.S., 1999. *"Design, Observation, Surprise! A Test of Emergence"*. Artificial Life Journal, vol. 5 (3), pp. 225-239. MIT Press. Cambridge MA.
- [14] Kurzweil, R., 1999. *"La era de las máquinas espirituales. Cuando los ordenadores superen la mente humana"*. Editorial Planeta. Barcelona, ISBN: 84-080-3218-6.
- [15] Brockman, J. (ed.), 2002. *"The Next Fifty Years. Science in the First Half of the Twenty-first Century"*. Vintage Books. New York, ISBN: 0-375-71342-5.

Avances en Algoritmos Evolutivos

Jesús A. Hernández¹, Julián Dorado²,

Marcos Gestal² y Ana Belén Porto²

¹ Instituto de Automática y Procesos Industriales, Fac. de Minas,

Universidad Nacional de Colombia, Sede Medellín

Email: jahernan@unalmed.edu.co

² Laboratorio RNASA-IMEDIR

Universidade da Coruña

Email: {julian, mgestal, anuska}@udc.es

Introducción

Los Algoritmos Evolutivos (AE) han probado ser herramientas poderosas para solucionar problemas difíciles de búsqueda y optimización en casi todos los dominios del conocimiento. Los AE son un campo de investigación relacionado con muchas áreas, entre ellas están la biología, la inteligencia artificial, la optimización y la toma de decisiones. La biología destaca por ser su fuente de inspiración. Hoy en día se trabaja en su hibridación con métodos tradicionales y con otros modelos bio-inspirados, en particular con la computación neuronal y la computación borrosa. La más reciente tendencia en los AE es superar los problemas de carácter paramétrico para abordar aspectos creativos. Se presenta en este capítulo un panorama de los AE partiendo desde sus orígenes, pasando por su definición formal y sus formas canónicas, para terminar en las nuevas perspectivas.

Inspiración biológica de los AE

Los sistemas vivos se distinguen por su organización y por su robustez. De la observación de los seres vivos se nota que éstos se reproducen, se adaptan, y evolucionan en relación con el entorno donde se desarrollan. Se ha advertido también, que algunas de las características que se adquieren en el transcurso de vida pueden ser heredables por la siguiente generación de la especie. La teoría sintética de la evolución o teoría neo-darwiniana de la evolución natural de las especies ha logrado explicar con gran detalle estos procesos [1]. Los Algoritmos Evolutivos (AE) emulan la teoría neo-darwiniana de la evolución de las especies.

La teoría sintética de la evolución se basa en los genes como unidades de transferencia de la herencia. El material genético de un individuo se encuentra en su genotipo. El genotipo consiste en un conjunto de cromosomas, los cuales conforman una organización de estructuras jerárquicas de genes. Los genes son la unidad bioquímica de codificación de un producto génico y constituyen la unidad funcional de la información básica para el desarrollo de un organismo. Los genes pueden ocasionalmente ser modificados por mutaciones. La compleja información contenida en el genotipo se expresa en la apariencia y funcionalidad de los individuos, es decir, en el fenotipo o características visibles de un organismo. En el proceso evolutivo se reconoce la ocurrencia de pequeñas variaciones en los fenotipos, aparentemente aleatorias y sin un propósito claro. Esas variaciones, llamadas mutaciones, si prueban su eficacia a la luz del entorno actual, prevalecerán a través de la selección del individuo, de

otra manera desaparecerán. La selección de los individuos se motiva en el fenómeno natural de la necesidad de producir descendencia. Por la severa competencia para alcanzar la reproducción, que sólo logran los individuos más aptos, se asume que los descendientes mejorarán a sus progenitores al heredar sus características mezcladas. En un medio ambiente con recursos abundantes, la densidad poblacional de la especie puede crecer exponencialmente. Cuando los recursos en el entorno se hacen insuficientes para toda la población, sólo tendrán mayor posibilidad de reproducirse, aquellos individuos que tengan una ventaja comparativa en el manejo eficiente de tales recursos. A estos individuos se les denomina como los más aptos. La presión selectiva sobre los individuos de una especie hace que ésta mejore continuamente respecto a su medio ambiente.

Por su parte, en los AE se tiene un conjunto de estructuras de información que interpreta a los individuos que conforman el problema a resolver, en analogía a los genotipos. Estos genotipos artificiales se componen de tantos elementos como variables haya en el problema a resolver, en semejanza a los cromosomas, y pueden estar codificados como valores reales o en algún otro lenguaje. Sobre estas estructuras de información se aplican las dependencias y restricciones que existan en el problema real, y están sujetas a las condiciones que impone el espacio de trabajo de éste. Un fenotipo corresponde a la valoración actual de un genotipo. El AE comienza con una valoración aleatoria del conjunto de estructuras de información. Sobre los individuos se aplican operadores genéticos, y aquellos que satisfagan la condición de aptitud establecida por el usuario se seleccionan para la siguiente generación. En cada generación,

la población de estructuras de información, como resultado del previo proceso evolutivo, comienza con una nueva valoración. Por su mejor adaptación, reflejada en un valor de aptitud, algunos individuos pueden prolongar su existencia durante varias generaciones. El proceso se repite un cierto número de pasos generacionales o hasta cumplir una condición dada. La solución al problema se establece como el individuo de mejor aptitud en la última generación, o durante todo el proceso evolutivo. En contraste con la evolución natural, que parece tener como único fin la preservación de la vida misma sin diferenciaciones, lo que se ha denominado como evolución de ciclo abierto, la evolución artificial en los AE se dirige a un propósito específico. Sin embargo, aunque se trata de seres virtuales actuando en un entorno conocido, debe mantenerse presente que la evolución de la población y la respuesta final en los AE, también son una característica emergente.

La evolución biológica natural, es decir, sin intervención humana, es un sistema dinámico y complejo, donde simultáneamente pueden estar modificándose el medio ambiente, los planes de adaptación y las medidas de ajuste. Esto significa que las estructuras bajo adaptación deben estar en capacidad de manejar enormes cantidades de información, presentada en diversas formas, junto con elevados grados de incertidumbre y ambigüedad. Como no se tiene claridad sobre los objetivos a corto plazo de la evolución natural, ésta se puede asociar de forma ligera a la lógica del azar con la producción indiscriminada de elementos. Algo muy lejano de lo que se observa en la naturaleza donde la fórmula para el proceso evolutivo parece ser la de caos y orden trabajando en armonía. No hay ninguna

contradicción, porque del azar puede emerger cualquier cosa, incluso la posibilidad de suprimirse a sí mismo en el interior de algún dominio, al provocar la organización. Con el proceso de evolución aparecerá con el tiempo un nuevo mundo de “objetos organizados”, en los cuales ya no existirá el azar, sino que en su interior estarán sometidos a un cierto orden, a una cierta lógica. Puede decirse que el nacimiento de un ser se da con el brote de comportamientos reconocibles, sostenidos, durante varias generaciones. Comportamientos que facultan a los objetos a funcionar de forma relativamente autónoma de los acontecimientos que suceden en su medio exterior [2].

Computación Evolutiva	Evolución Natural
<i>Propiedades Relevantes</i>	
Autonomía Operativa	Autonomía Operativa
Procesos Controlados	Procesos Controlados y No Controlados
Múltiples Métodos de Reproducción	Reproducción Sexual y Asexual
(potencialmente) Evolutivo	Evolutivo
Orientado a Propósitos	Selección Natural
“Operadores Genéticos”	Operadores Genéticos
Agentes Cooperativos	Agentes Cooperativos y No Cooperativos
Benigno	Benigno y Maligno
Autores Conocidos	Autores “Conocidos” y Anónimos
Propagación de Información	Propagación y Destrucción de Información
De Frágil a Robusto	Muy Robusto
Normalmente no Envejece	Envejece y Muere

Tabla 1: Características de los Procesos Evolutivos

No obstante, en los AE el proceso evolutivo que en ellos se encuentra, se acerca más a la evolución forzada que ha ejercido el ser humano sobre otras especies naturales. Casos en los cuales la exigencia de adaptación se conoce, las estructuras bajo adaptación están claramente delimitadas, y tanto la medida de aptitud para evaluar los cambios estructurales, así como las operaciones que se aplican para realizar las modificaciones están preestablecidas. En estos casos, se puede manejar interna o externamente el reconocimiento del estado del sistema, la memoria, el método de asignación del resultado final y los parámetros que controlan el proceso. En la Tabla 1 se presenta una comparación entre los aspectos comunes y las diferencias entre la computación evolutiva y la evolución biológica.

Enfoques en los AE

Los AE son un paradigma de programación de ordenadores bio-inspirado, que emula los procesos de selección natural del organismo más apto de una población, para satisfacer una exigencia del medio ambiente. Sus costos de desarrollo y aplicación son bajos pues no exigen presunciones sobre el espacio del problema, es fácil hibridarlos con otros métodos, funcionan con un paralelismo intrínseco, son robustos, tolerantes a fallos e interactivos con el usuario, su convergencia y confiabilidad se han demostrado [3].

Los AE son muy valorados en los problemas de mayor exigencia computacional o en aquellos difíciles de solucionar por métodos

tradicionales. En problemas en los que para encontrar una respuesta aceptable se requiere hacer búsquedas en cantidades enormes de posibilidades de solución, los AE ofrecen un paralelismo fácil de implantar y estrategias inteligentes para decidir el camino a seguir. Los AE son más eficientes en estos casos pues la búsqueda de la solución no se hace de forma secuencial, sino en paralelo en el espacio completo de posibles soluciones al problema, dirigiendo la exploración y la selección a las porciones del espacio donde residen las soluciones más cercanas a lo esperado [4, 5, 6].

Se denomina Algoritmo Evolutivo a cualquier método de solución de problemas, que utilice procesos operativos provenientes del paradigma Neo-Darwiniano. Procesos tales como: reproducción, mutación, competencia y selección.

Un AE emula el proceso natural no-determinista de evolución adaptativa, en el cual un individuo de una población es afectado por otros individuos de esa misma población y por el entorno de esa población. El AE actúa sobre un espacio de búsqueda específico, compuesto por todas las soluciones potenciales del problema a resolver. Los elementos de ese espacio de búsqueda son cadenas finitas de símbolos, cada una representando un individuo que codifica una posible solución al problema. Usualmente, la estructura de información de los individuos es de longitud fija y el tamaño de la población es invariable.

Los AE, en general, comienzan con un conjunto de posibles soluciones (no necesariamente correctas ni conocidas) para el problema en

estudio. Luego, a partir de ese conjunto, se crean, aleatoriamente, nuevas soluciones a probar, modificando las existentes por medio de operadores genéticos. Para conocer si una solución ensayada es aceptable se usa un indicador de ajuste, llamado medida o función de aptitud, y , a partir del “error” encontrado, se seleccionan las soluciones que se conservarán como padres para el siguiente paso generacional. El proceso se repite hasta encontrar una solución adecuada entre los descendientes, o hasta llegar a un criterio de terminación predeterminado. En la Figura 1, puede apreciarse el procedimiento seguido por un AE típico.

```

t := 0; //Iniciar con un tiempo inicial
IniciarPoblacion P(t); //Generar aleatoriamente población inicial
Evaluar P(t); //Evaluar la aptitud de los individuos de
                la población
While not criterio do //Criterio de terminación
t := t + 1; //incrementar el tiempo generacional
    P'(t):= seleccionarPadres P(t); //Seleccionar padres
    reproducir P'(t); // Generar hijos
    P := sobrevivientes P, P'(t); //Construir la nueva población
                                a partir de los hijos y algunos
                                miembros de la población actual
    Evaluar P(t); //Evaluar aptitud de la nueva población
End do

```

Figura 1: Algoritmo general usado por los métodos basados en el modelo de evolución

En el proceso de evolución mencionado, los individuos descendientes heredan la información genética de las estructuras padre. Con base en esta información seguirán interactuando para la creación de nueva información genética y su correspondiente evaluación, selección y

diseminación. Con el transcurso del tiempo irán desapareciendo aquellos individuos que por diversos motivos no tienen la suficiente aptitud para lograr el ajuste adecuado y se van generando comportamientos y estructuras capaces de enfrentar con éxito diferentes situaciones.

Tipos de AE

Siguiendo una notación similar a la establecida en [3] una definición conceptual del algoritmo evolutivo, es una 8-tupla:

$$AE = (I, F, O, S, T, C, p, h), \quad (1)$$

Donde, $I = A_x \times A_s$ es el espacio de los individuos, con A_x y A_s conjuntos arbitrarios.

$F = I \longrightarrow \mathbf{R}$ es la función de aptitud que asigna valores reales a los individuos.

$O = \{w\theta_1, \dots, w\theta_z \mid w\theta_i : I^p \longrightarrow I^{p+h}\} \cup \{w\theta_o : I^p \longrightarrow I^h\}$ el conjunto de operadores genéticos $w\theta_i$, cada uno controlado por parámetros específicos contenidos en los conjuntos $\theta_i \in \mathbf{R}$. Donde p = número de individuos padre, h = número de individuos descendientes, y $(p, h) \in \mathbf{N}$.

$S\theta_i : (I^p \cup I^{p+h}) \longrightarrow I^p$ denota el operador selección, que puede cambiar el tamaño de la población, con $p = h$ permitido.

$$T = s \circ w\theta_1 \circ \dots \circ w\theta_{ij} \circ w\theta_o \text{ con}$$

$T(P) = s\theta (Q \cup w\theta_1 (\dots(w\theta_{ij} (w\theta_o(P))) \dots)$ con $Q \in \{0, P\}$
composición de funciones que transforma una población en la siguiente generación aplicando los operadores genéticos y la selección.

$C = I^p \longrightarrow \{\text{verdad, falso}\}$ es un criterio de terminación.

La investigación en AE se basa, principalmente, en tres enfoques desarrollados de manera independiente por grupos de investigación diferentes e inicialmente no relacionados, son: Estrategias de Evolución (Evolutionstrategie), Programación Evolutiva, y Algoritmos Genéticos. Cada técnica determina los factores por los cuales un individuo puede diferir de otro y por tanto la variación genética que se permite. En términos generales, todos ellos siguen, lo expuesto en los apartados anteriores. En las diferentes técnicas de algoritmos evolutivos son más las semejanzas que las diferencias. Estas se reducen al tipo de estructuras de datos para representar los individuos, los modos de cambio en las soluciones para crear los descendientes y en el procedimiento para seleccionar los nuevos padres [2]. De estos tres tipos básicos han surgido otros que en esencia mantienen las características de la clase original. Sus características diferenciales se presentan a continuación.

Algoritmos Genéticos. Propuestos por John Holland como una abstracción de la evolución biológica, más que como una técnica de optimización [11]. Las estructuras en adaptación usualmente son cadenas de caracteres binarios, con longitud fija. También son posibles otros tipos de estructuras. La medida de aptitud se determina ejecutando rutinas y algoritmos específicos para cumplir una tarea, y se interpreta la cadena de

caracteres como un conjunto de parámetros. El operador genético principal es la recombinación sexual, llamada cruce. El operador mutación se utiliza pero se considera de importancia secundaria. Los algoritmos genéticos, desde entonces, han seguido desarrollándose. Inicialmente hubo cambios en la representación de los individuos a cadenas no binarias, luego a cadenas de longitud variable, y hoy día con variantes como los sistemas clasificadores y la programación genética.

Programación Evolutiva. Fue introducida por Fogel [7] como una técnica para crear inteligencia artificial, basada en máquinas de estado finito con capacidad de predecir eventos observando los estados previos. Se usan diferentes estructuras de representación de los individuos en adaptación, desde sencillas, como cadenas de caracteres, hasta complejas, como máquinas de estado finito. Un rasgo común en la representación es la utilización de variables objetivo con valores del mundo real. La medida de aptitud se determina ejecutando rutinas y algoritmos para cumplir una tarea, cuyos argumentos son variables objetivo del problema de interés. Aunque la mutación es el único operador genético que se emplea, la estrategia de solución es computacionalmente adecuada.

Estrategias de Evolución. De forma independiente a las anteriores, en 1973, Rechenberg desarrolló un método que él llamó Estrategias de Evolución, para solucionar problemas discretos y continuos de optimización de parámetros en experimentos reales en dispositivos aéreos [8]. Las estructuras en adaptación son, por lo general, conjuntos de variables con valores del mundo real. Para conformar un individuo se asocian a estas variables objetivo las llamadas variables de estrategia, también en valores

reales. La medida de aptitud se determina ejecutando rutinas y algoritmos para cumplir una tarea, utilizando las variables objetivo como parámetros. La mutación es el operador genético principal, y el modo como cambia cada una de las variables objetivo para producir nuevos individuos, es controlado por las variables de estrategia. También se aprovecha la recombinación tanto en las variables objetivo como en las variables de estrategia.

Todas estas técnicas han seguido extendiéndose con los trabajos de diferentes investigadores. Al usar cualquiera de ellas algunos detalles a tener en cuenta [9, 10, 11] son: la representación de los individuos, el método de selección y la forma de operar el cruce y la mutación.

Fronteras en los AE

A partir del trabajo publicado en 1989 por Goldberg [12] los AE se han venido utilizando de forma extensiva en exploración, búsqueda y optimización en espacios multidimensionales, pero no son las únicas formas de aprovechar su potencial. Puede decirse que los AE se han aplicado [2] en todos los campos del conocimiento, desde la localización de óptimos, la solución de complejos sistemas de ecuaciones, la programación de la producción, problema del transporte, evaluación de créditos, predicciones financieras, estructuras mecánicas, control automático, redes de datos, optimización combinatoria, robots móviles, visualización en tres dimensiones hasta la producción de modelos de software para casos específicos. En particular, problemas en donde los métodos tradicionales se

han mostrado inoperantes o que para encontrar una solución requieren de una excesiva carga computacional.

En la actualidad, se están empleando los AE para impulsar desarrollos avanzados, entre ellos destacan: los tejidos electrónicos de la embriónica con arquitecturas incrementales basadas en el desarrollo de los organismos vivos a partir de un conjunto de instrucciones genéticas; Sistemas Inmunes Artificiales, paquetes informáticos robustos, adaptativos, ligeros y distribuidos para diagnóstico de fallos, detección de intrusos internos/externos, control de tráfico, y descubrimiento de patrones; y Sistemas Evolutivos Creativos, una representación basada en componentes con relajación de restricciones, para aplicaciones no convencionales donde la búsqueda del óptimo es irrelevante tales composición musical, arte y diseño conceptual.

La mayoría de las aplicaciones que se han hecho con AE se han resuelto siguiendo la orientación tradicional en donde se actúa sobre individuos que tienen características similares, es decir son de una misma clase, y que conforman una población que se desarrolla en un entorno estable. Esta es una situación ideal que se aleja mucho de lo observado en la naturaleza, en donde el medio ambiente cambia permanentemente, la lucha por la supervivencia se hace entre grupos de individuos pertenecientes a diferentes especies biológicas, y el propósito de la herencia genética está orientado a la preservación del ente colectivo, o especie, más que a la permanencia entre generaciones de un ente unitario (o individuo). En la actualidad, como un acercamiento a las especies biológicas, se investiga en métodos de AE que operen con subpoblaciones evolucionables. Aunque

para estas subpoblaciones se preestablecen los aspectos de su transformación y características físicas como tamaño, forma, y vecindad, los resultados en optimización multimodal han sido exitosos [13].

Con la orientación tradicional de evolución de individuos ya se solucionaban problemas multimodales, pero se obtenía una sola respuesta: la del óptimo general o la del óptimo más cercano al general. Con la evolución de subpoblaciones se logran tantas soluciones como especies se hayan definido en el planteamiento del problema. En la Figura 2 se aprecian dos casos donde la utilización de subpoblaciones es ventajosa, uno con varios óptimos locales y el otro con dos óptimos globales de igual valor. En el primer caso el óptimo general es evidente, la respuesta apropiada depende de la precisión establecida, pero en el segundo caso, cualquiera de las dos soluciones corresponde al óptimo global. Si sólo se espera una respuesta, el AE tradicional encuentra una de las dos, pero no daría suficiente información para una toma de decisiones correcta.

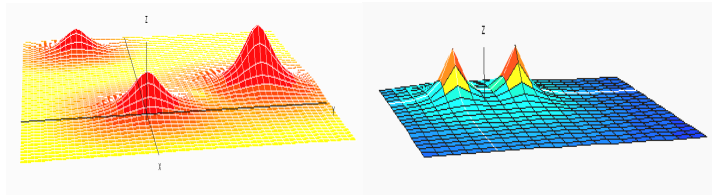


Figura 2: Dos casos donde los AE con subpoblaciones son ventajosos

Una problemática vigente en los AE es la de garantizar que la respuesta encontrada es el óptimo global, y si existiesen varios óptimos globales del mismo valor, garantizar su localización. La toma de decisiones espaciales y el tratamiento de sistemas dinámicos también demandan otras soluciones [14, 15], como es la situación que se muestra en la Figura 3, donde la respuesta más adecuada no es el máximo global sino que es preferible estar cerca de un óptimo local más alejado del punto crítico (mínimo global), que tiene un comportamiento diferente al resto del sistema. Para enfrentar estos desafíos se han extendido los AE al manejo de diferentes clases de individuos, en contraste con la aproximación de subpoblaciones de una misma especie.

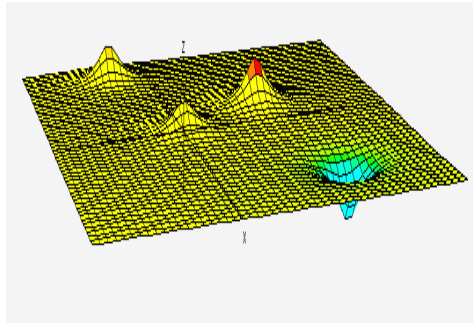


Figura 3: Un caso que requiere el tratamiento de diferentes especies en el AE

La existencia de varias poblaciones con diferentes características que evolucionan de modo concurrente en un mismo espacio lleva a considerar la fusión de los AE con la ecología. La sinecología es la parte de la ecología

que se concentra tanto en las relaciones entre diferentes especies, como de las relaciones de ellas con el entorno. Esta combinación de AE con relaciones sinecológicas se ajusta de manera natural a un espacio multimodal en evolución. Ese espacio multimodal se puede considerar una comunidad en permanente cambio, en donde cada óptimo es un nicho. En cada generación los individuos se distribuyen entre los nichos identificados. Los individuos que se encuentran en cada nicho se organizan como una especie que evoluciona autónomamente hacia la plena colonización del nicho. Simultáneamente, cada especie tiene relaciones con las demás especies presentes en la comunidad multimodal, esas interacciones entre dos o tres individuos son de competencia, depredación o mutualismo. Como fruto de esas interacciones las especies se ven sometidas a mayores presiones evolutivas y se puede dar la migración entre especies. El proceso continúa hasta que se llega al clímax de la distribución poblacional.

En esta línea de trabajo, recientemente se han propuesto los Algoritmos Eco-Evolutivos (AEE) que aprovechan la solidez que tienen los AE para la transformación interna que se logra en las poblaciones con el fin de lograr un objetivo establecido, en unión con el conocimiento existente de las relaciones generales entre grupos de individuos que se maneja en la sinecología [16]. En los AEE no se preestablecen el número de especies, ni su vecindad, el tamaño, la forma, ni la distribución de las mismas. Las especies se auto-organizan por lo que las soluciones halladas corresponden a los diferentes óptimos existentes en el espacio de búsqueda, aún siendo de diferente clase.

Una relación de los AE con la estructura jerárquica de la materia de acuerdo con la dimensión y la función, se presenta en la Tabla 2. En esta tabla se ubica la orientación tradicional de los AE con tratamiento evolutivo de individuos, el campo de acción actual de manejo de subpoblaciones, el nuevo rumbo como los AEE, donde ya se tienen comunidades compuestas de especies de diferente clase que compiten o cooperan entre sí y debe mantenerse cierto equilibrio general en todo el espacio de búsqueda, y el futuro que se visualiza como la ontogénesis de los individuos, es decir, obviar la prescripción de la representación de los individuos; la auto-organización, en otras palabras, la relajación y remoción de restricciones para la evolución; la escalabilidad en los AE, o la posibilidad de ejercer los mismos procedimientos tanto en entes unitarios como en entes colectivos, en un sólo proceso evolutivo sin detrimento del mismo; y un auto-reconocimiento permanente del estado de la evolución para encauzarla hacia el objetivo, que requiere una memoria de conservación y a corto plazo.

Universo				
Galaxias				
Sistemas solares				
Planetas vecinos				
Planeta Tierra				
Biosfera				Ámbito de la Ecología
Bioma				
Paisaje				Sinecología
Ecosistemas				
Comunidades			Nuevo	
Poblaciones/especies		Ámbito	Ámbito	
Organismos	Ámbito tradicional AE	Actual		Autoecología
Sistema de órganos			Futuro	
Órganos			Ámbito	
Tejidos			de los	
Células			AE	
Protoplasma				
Moléculas				
Átomos				
Partícula subatómica				

Tabla 2: AE y estructura jerárquica de la materia

Bibliografía

- [1] Freeman, S., Herron, J., 2002. *“Análisis Evolutivo”*. Prentice Hall. Madrid, ISBN: 84-205-3390-4.
- [2] Hernández, J.A., 1999. *“Evolución de la Computación Evolutiva”*. II Taller en Renovación y Adaptación de Sistemas. Universidad Nacional de Colombia. Medellín.

- [3] Bäck, T., 1996. "*Evolutionary Algorithms in Theory and Practice*". Oxford University Press. New York, ISBN: 0-19-509971-0.
- [4] Angeline, P.J., Kinnear, K.E., 1996. "*Advances in Genetic Programming*". vol. 2. MIT Press. Cambridge, MA.
- [5] Bäck, T., Hammel, U., Schwefel, H., April, 1997. "*Evolutionary Computation: Comments on the History and Current State*". IEEE Transactions on Evolutionary Computation. Piscataway, NJ.
- [6] Holland, J.H., 1992. "*Adaptation in Natural and Artificial Systems*". MIT Press. Cambridge, MA.
- [7] Fogel, L.J., Owens, A.J., Walsh, M.J., 1966. "*Artificial Intelligence through Simulated Evolution*". John Wiley and Sons. New York.
- [8] Rechenberg, I., 1973. "*Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*". Frommann-Holzboog, Stuttgart.
- [9] Michalewicz, Zbigniew, 1996. "*Genetic Algorithms + Data Structures = Evolution Programs*". 3^a Ed. Springer-Verlag. Berlin, ISBN: 3-540-60676-9.

- [10] Mitchell M., 1996. *"An Introduction to Genetic Algorithms"*. MIT Press. Cambridge, Mass.
- [11] Torres, J., Hernández, J.A., Septiembre 2000. *"Modelación de Sistemas No Lineales por Algoritmos Genéticos y Orientación a Objetos"*. XXVI CLEI. México.
- [12] Goldberg, D.E., 1989. *"Genetic Algorithms in Search Optimization and Machine Learning"*. Addison-Wesley, ISBN: 0-201-15767-5.
- [13] Ursem, R., 2000. *"Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments"*. Proceedings of the Genetic and Evolutionary Computation Conf. Morgan Kaufmann. San Francisco, CA.
- [14] Fleming, P., Purshouse, R., 2002. *"Evolutionary Algorithms in Control Systems Engineering: a survey"*. Control Engineering Practice. vol. 10 (11), pp. 1223-1241. Pergamon Press.
- [15] Hernández, J.A., Peña, A., 2005. *"Interpolation and Exploration of Response Surfaces using Evolutionary NURBS, LaGrange Constraint and Cylindrical Operators"*. Aceptado para publicación en GECCO-2005. Washington, USA.
- [16] Hernández, J.A., 2004. *"Mantenimiento de la Diversidad en Algoritmos Evolutivos"*. Tesis Doctoral. Universidade da Coruña. A Coruña.

Una Aproximación a Sistemas Inmersivos Colectivos Basada en Arquitectura de Bajo Coste

Julián Flores¹, Antonio Otero y José Ángel Taboada
Instituto de Investigaciones Tecnológicas
Departamento de electrónica y Computación
Universidad de Santiago de Compostela
Email: eljulian@usc.es¹

Antecedentes

La realidad virtual (RV), como área de la computación gráfica dedicada a la construcción de espacios tridimensionales en los cuales el usuario tiene la sensación de estar inmerso era, hasta hace poco tiempo, utilizada únicamente en el entorno militar o centros de investigación debido, principalmente, al elevado precio de los equipos utilizados. Las dos características más destacables de la R.V. son la realización del proceso de visualización o *render*, en tiempo real y el uso de dispositivos de inmersión.

En general, el proceso de visualización de una aplicación de R.V. se realiza mediante estaciones gráficas de altas prestaciones, siendo S.G.I. la compañía que lidera casi la totalidad de las ventas de este tipo de equipos. Cuando nos referimos a dispositivos de inmersión aparecen dos claras tendencias: equipos unipersonales, donde el *Head Mounted Display* (HMD)

ha sido, durante años, la principal herramienta de investigación y sistemas de inmersión colectiva donde, sin duda, el CAVE [6] es la herramienta más ampliamente utilizada [16]. CAVE, acrónimo de Cave Automatic Virtual Environment, se diseñó al principio de 1991 y su presentación operativa se realizó en el Siggraph'92. Básicamente es una habitación de 3x3x3 metros cuyas paredes, suelo y techo son pantallas. Todas las pantallas son retro-proyectadas en estéreo activo mediante el uso de LCD estéreo *shutter glasses*. El usuario dispone de dos tracker uno situado en la cabeza y otro en su mano que permite determinar su punto de vista e interactuar con el entorno. La imagen es generada por una o varias estaciones gráficas. Y aunque la calidad del sistema es muy alta, su precio, también elevado, lo hace en muchos casos inaccesible.

En la actualidad existen un conjunto amplio de investigaciones que aplican la tecnología del CAVE a diferentes áreas de investigación, que van desde la propia implementación de la arquitectura propuesta, al software utilizado, su trabajo en red, y su aplicación en diferentes áreas [16][20][1]. Es importante saber que el termino “CAVE” no puede ser usado para hacer referencia genéricamente a sistemas de este tipo ya que dicha palabra ha sido registrada y comercializada por Fakespace Systems en exclusiva a nivel mundial. Por lo tanto, las variaciones en este tipo de sistemas deberemos denominarlas o de otra forma o genéricamente CLS (CAVE like system) [25]. No obstante, el CAVE y los CLS podemos englobarlos dentro de los sistemas de visualización denominados más genéricamente “tiled displays” [10][22][14][32] que podríamos traducirlo como baldosa o puzzle

de imágenes, es decir sistemas donde la imagen final es la suma de muchas otras como si de un mosaico se tratara.

Recientemente, la aparición de las nuevas aceleradoras gráficas para ordenadores personales con prestaciones similares a estaciones de trabajo con precios muy superiores, y la aparición de nuevos dispositivos de inmersión, casi de uso doméstico hacen pensar en una rápida evolución y una amplia difusión de la realidad virtual. Esta evolución está fuertemente impulsada por la industria del vídeo juego que ha llevado a que cada 6 meses aparezcan tarjetas aceleradoras con capacidades que duplican a sus predecesoras. Pese a ello, actualmente, las aplicaciones de R.V. que emplean el P.C. como plataforma hardware son escasas y están principalmente dirigidas a experiencias unipersonales [19].

Otro elemento a tener en cuenta en el desarrollo de aplicaciones R.V., es la evolución y adecuación de las API's de desarrollo. Así, casi la totalidad de las tarjetas gráficas son compatibles con los estándares gráficos, principalmente, OpenGL [OpenGL, web] y/o DirectX [DirectX, web], en entornos PC, permitiendo, de esta forma, que la misma aplicación sea compatible con diferentes plataformas hardware, y sistemas operativos: Irix (SGI), Solaris (SUN), Linux (PC, SGI), WindowsXX (PC).

Por tanto, un interesante campo de investigación es el estudio de la capacidad de plataformas de bajo coste, principalmente PC's, en el desarrollo de aplicaciones de R.V. Este estudio debe presentar dos líneas principales: por una parte, el estudio de la capacidad de los PC's para generar aplicaciones gráficas complejas con una calidad elevada y por otra,

el uso y/o desarrollo de dispositivos de inmersión similares a los utilizados hasta ahora sobre computadoras de más altas prestaciones.

A continuación tratamos cada uno de los elementos que componen un sistema inmersivo o semi-inmersivo empleando para ello elementos de bajo coste.

Sistema de proyección

La elección del sistema de proyección es uno de los elementos fundamentales del sistema, ya que a través de él se presenta al usuario el entorno y es utilizado como primer elemento de interacción. En muchos sistemas, un monitor con una calidad alta suele ser suficiente pero no para un sistema multiusuario con características de C.L.S. donde varios usuarios tienen que compartir la misma escena, donde se precisa de un área de visualización amplia y libre de interferencias.

Un buen método para conseguir un área de proyección grande es el uso de proyectores coordinados. La mayor parte de los sistemas que utilizan esta estrategia realizan la proyección por la parte trasera de la pantalla (retro-proyección) a fin de evitar interferencias de los usuarios

La retro-proyección permite al usuario verse delante del entorno virtual pero no tiene la sensación de estar inmerso en él ya que la imagen no tiene propiedades tridimensionales. Para obtener estas cualidades es necesario que la proyección sea estereoscópica mediante la cual el usuario posee una mayor sensación de inmersión.

Distribución de las pantallas

Uno de los gran avance propuesto por el CAVE ha sido que el sistema de proyección cubría un campo de visión de 180° (figura 1).

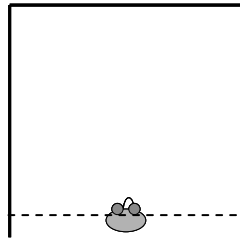


Figura 1: Esquema básico del CAVE, campo de visión 180°

Esta aproximación cubre la totalidad de la visión de un usuario, no obstante presenta un problema importante. Debido a que la distancia a todos los puntos de las pantallas no es constante, la resolución (número de *pixels* por ángulo de visión) depende del punto de la pantalla considerado. El caso ideal sería en el que la pantalla fuera una semiesfera (figura 2), donde la distribución de *pixels* por ángulo es constante.

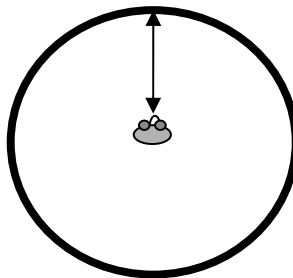


Figura 2: Display ideal

No obstante el uso de este tipo de pantallas presenta problemas importantes atribuibles a la generación de la imagen y su proyección; calibración proyección sobre superficies curvas, corrección de la imagen por hardware o software lo que incremente en muchos casos el precio de los equipos utilizados y en todos la complejidad del sistema hardware y software.

Muchos casos optan por una solución de compromiso que se acerque más que a una solución cuadrada a la de una esfera. La arquitectura es la de un sistema semi-hexagonal (figura 3).

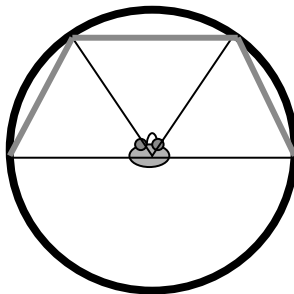


Figura 3: Distribución de las pantallas

Como podemos ver con esta distribución se consigue que la distorsión por píxel sea menor y en cambio el campo de visión sigue manteniéndose. Para una resolución en cada una de las pantallas 1024x768 y puesto que el usuario forma con las pantalla un triángulo equilátero obtenemos una resolución de $1024/60^\circ = 17.0 \text{ pixel por grado}$.

Generación de visión estereoscópica

La generación de la visión estéreo o estereoscópica permite al usuario verse inmerso en la escena al proporcionar a cada uno de los dos ojos información diferente, generando una imagen 3D del entorno tal como se realiza en la vida real. Existen dos métodos para la generación de visión estereoscópica; estéreo activo y pasivo.

Estéreo activo

Los sistemas activos emplean un único elemento (suma de tarjeta gráfica y sistema de proyección) que modifica en tiempo real alguno de los elementos del sistema de proyección. Todos ellos deben trabajar sincronizadamente para la generación de la imagen de los dos ojos y asegurando que la imagen izquierda llega al ojo izquierdo y que la derecha al ojo derecho. Esto se consigue mediante el empleo de dispositivos especiales que permiten filtrar la imagen correspondiente a cada ojo, en general gafas especiales. Los dos dispositivos utilizados en este tipo de estéreo son; *Shutter Glasses* y *pantallas "Z"*.

Shutter Glasses

En este caso el dispositivo son unas gafas que disponen de lentes de cristal líquido que puede hacerse opacos o transparentes independientemente mediante control hardware [Stereographics, web]. Las gafas son controladas por los sistemas gráficos a los que se encuentran conectadas mediante una red inalámbrica de infrarrojos. Su nombre

proviene justamente de los cambios inducidos en los cristales de las gafas (activas).

Este método implica que el *frame rate* del sistema deba ser superior a 60f/s, 30 por ojo, ya que de ser menos se notaría que uno de nuestros ojos se encuentra oscurecido. Este es uno de los métodos más comúnmente empleados y sus problemas vienen asociados al elevado *frame rate* necesario y al alto precio de las gafas.

Estéreo activo mediante una pantalla “Z”

Una pantalla “Z” es un filtro óptico activo que produce cambios en la polarización de la luz que lo atraviesa en nuestro caso controlado por el subsistema gráfico. La imagen generada es polarizada en función de si la imagen corresponde al ojo derecho o izquierdo encontrándose la polarización de ambos ojos desfasadas entre sí 90°. Para filtrar las imágenes se utilizan las denominadas gafas polarizadoras o gafas pasivas. Las gafas pasivas, al contrario que las activas, poseen cristales polarizados desfasados entre si 90 grados filtrando, por tanto, la imagen que no se encuentre en sincronía angular con el ángulo del filtro de cada ojo. El único problema que se plantea es que se debe mantener la alineación entre la pantalla y las gafas a fin de no perder la sincronía angular del par imagen gafas. Este sistema es activo no por las gafas, como ocurría en el caso anterior, sino por el funcionamiento dinámico de la pantalla polarizadora.

Estéreo Pasivo

El estereo pasivo se basa en la generación de imágenes independientes para cada ojo, polarizadas y desfasadas 90°. El observador viste gafas con lentes polarizadas que filtrarán las imágenes, de forma que a cada ojo llegue sólo la imagen adecuada. Los polarizadores utilizados son, en general, lineales, no obstante, existen experiencias con sistemas que utilizan polarizadores circulares, aunque estos son mucho más difíciles de conseguir.

Existen dos técnicas principales para la generación de estereo pasivo; mediante un único o dos proyectores. En el primer caso se emplean dos dispositivos (tarjeta gráfica + proyector), uno por ojo, para la generación del par estereoscópico. La imagen de cada proyector es polarizada y presentada superpuesta junto con la otra sobre la pantalla. En el segundo caso sólo se empleará un dispositivo (tarjeta + proyector) en este caso el proyector recibirá las dos imágenes y será él mismo el que se encargue de polarizar la imagen adecuadamente para su posterior filtrado.

Generación del par estereoscópico

La esencia del par estereoscópico es la generación, procesado y visualización de dos imágenes, una para el ojo derecho y otra para el ojo izquierdo paralelamente. Este proceso debe ser lo más transparente posible al programador y/o al operario de la aplicación y pasar a depender de otros factores como son el hardware o las API's utilizadas.

La aproximación desarrollada por la mayor parte de los sistemas gráficos con estéreo activo es emplear cuatro “*frame buffers*” (*quad-buffer*), las imágenes para el ojo izquierdo y el ojo derecho son dibujadas en el *frame* oculto mientras al usuario se le muestra la imagen que hay en el *buffer* visible para cada uno de los dos ojos. El problema estriba en que no todos los sistemas gráficos permiten este tipo de buffer y la mayoría de los sistemas operativos no son capaces de gestionarlos. En este sentido, este tipo de estéreo se ve totalmente condicionado por el hardware utilizado. Por el contrario, cuando recurrimos a la generación de estéreo pasivo el abanico de posibilidades se amplía.

Dos computadores con dos sistemas gráficos

Dos computadores con un adaptador gráfico y un monitor o un proyector, conectados mediante una red. Cada uno de ellos es identificado como derecho, o izquierdo generando las imágenes independientes pero coordinadamente superponiéndolas sobre la pantalla (figura 4).

Las grandes ventajas de esta configuración se fundamentan en que las imágenes para el ojo derecho e izquierdo son generadas en paralelo lo cual incrementa la velocidad de visualización respecto a sistemas con un único equipo al paralelizarse la visualización. No obstante, en la práctica uno de los dos computadores debe coordinar el sistema. Uno de ellos visualizará su propia escena en su adaptador y enviará la información necesaria al otro equipo sobre qué *frame* debe visualizar en cada momento. Este proceso puede implicar problemas de sincronización sobre la red debido a diferencias en el tiempo de *render* y latencias de transmisión. La

posible solución radica en la sincronización mediante un tercer ordenador que coordina los dos encargados de la visualización. Debemos tener en cuenta que en este caso el tráfico de red se duplica.

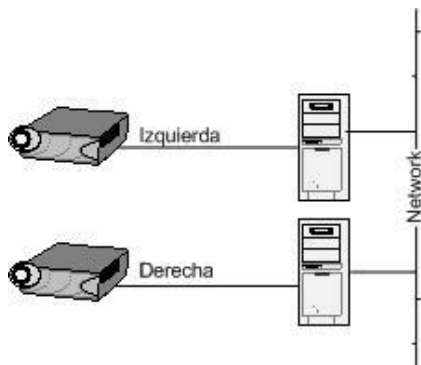


Figura 4: Generación de estéreo mediante el uso de dos computadores

Un computador con dos sistemas gráficos

En principio el problema de sincronización descrito anteriormente puede ser resuelto mediante el uso de dos adaptadores sobre el mismo computador (figura 5). En este caso la aplicación genera las dos imágenes en el mismo ordenador. Esta configuración es difícilmente implementable sobre arquitecturas comunes, ya que la mayoría de los adaptadores gráficos usan un puerto AGP y los PC solo disponen de uno. Sí es verdad que existen tarjetas gráficas PCI y que los PC's disponen de varios de estos puertos, pero debemos de tener en cuenta que la *performance* de estos adaptadores es mucho menor que los AGP principalmente debido a un menor ancho de banda del bus PCI.



Figura 5: Un computador con dos adaptadores

Un computador con un sistema gráfico con múltiple salidas

En este caso el sistema gráfico dispone de varias salidas, a las cuales se conectan varios monitores o proyectores (figura 6). La sincronización es en este caso es automática, el problema es el desarrollo de una aplicación que permita separar la imagen correspondiente a cada ojo sobre el mismo adaptador, ya que en este proceso puede perderse la aceleración del sistema gráfico.



Figure 6: Un computador con un adaptador dual

Un Computador con un adaptador grafico y doble buffer

Esta técnica consiste en la generación de las dos imágenes sobre el mismo *frame buffer*. Para la separación de las imágenes, el *buffer* es dividido en dos. La división puede realizarse utilizando dos técnicas (figura 7 a,b):

- El *buffer* es dividido en dos mitades independientes donde la imagen correspondiente a cada ojo es generada en el espacio correspondiente a cada una de ellos.
- Interlineada donde las imágenes son generadas en líneas alternas así la imagen del ojo derecho se generaría por ejemplo sobre las líneas pares y la del ojo izquierdo sobre las impares.

El principal problema de esta técnica es que se reduce la resolución a la mitad y evidentemente necesitaremos un conjunto de proyectores que permitan el procesamiento adecuado de la imagen generada.

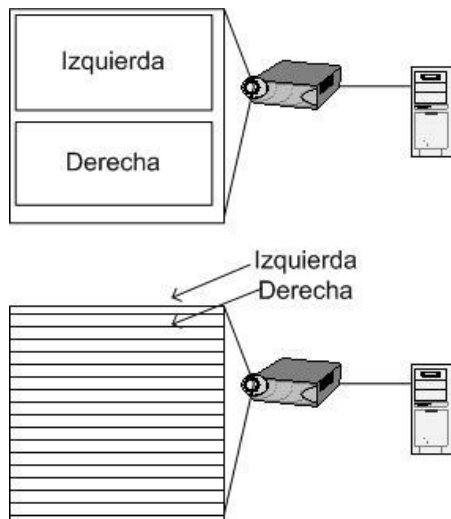


Figura 7 a, b: Adaptador gráfico con doble buffer, superior-inferior, interlineado

Sincronización de las imágenes

Como vimos en el apartado anterior en muchos casos la generación del par estereográfico supone la coordinación de las imágenes generadas independientemente sobre la misma o incluso en diferentes máquinas. Existen dos técnicas que permiten obtener sincronía entre las diferentes imágenes a fin de que correspondan al mismo instante temporal.

Gen-locking

Este sistema de coordinación está dirigido principalmente a la sincronización de vídeos mediante técnicas de estéreo activo. Por lo tanto, el refresco de vídeo de todos los proyectores debe ser realizado simultáneamente y todas las señales de sincronismo horizontal y vertical deben estar absolutamente coordinadas a fin de poder usar gafas activas.

Algunas Workstation resuelven el problema mediante la utilización de sistemas gráficos con salidas complementarias de coordinación que se conectan a todos los sistemas gráficos, necesitando en muchos casos una fuente externa de tiempos para su sincronización. No obstante, las tarjetas gráficas comunes para PC's no disponen de esta posibilidad, siendo el ámbito de utilización de esta técnica limitado.

Frame buffer o frame locking

Frame-locking es un proceso relativo a la sincronización de los *frame buffer* en cada uno de los elementos del cluster. El objetivo de esta

sincronización es generar la apariencia de un único *frame* buffer que está distribuido por todos los nodos.

Normalmente las tarjetas gráficas usan la técnica de *doble buffer* para mantener en sincronía las imágenes generadas con el refresco de vídeo. En esta situación una imagen es presentada mientras la nueva está siendo generada. Cuando el nuevo *frame* se finaliza la tarjeta gráfica intercambia el nuevo *frame* re-dibujándolo sobre el viejo. Esta técnica busca que el proceso de refresco imagen nueva-vieja se realice de una forma coordinada en todos ellos.

La técnica de *frame-locking* es la que hemos desarrollado en nuestro trabajo. Todos los ordenadores visualiza la parte de la escena que le corresponde pero la imagen generada no es visualizada hasta que se recibe una señal de control por parte de un computador que coordina el sistema, el único problema radica en que la señal de control llegue a todos ellos a la vez.

Proyectores

Los proyectores son el elemento del que depende la representación de la imagen sobre la pantalla. Existen un conjunto importante de propiedades que deben ser examinadas a la hora de elegir un sistema de proyección: características de la luz que emiten, luminosidad, frecuencia de muestreo, resolución, zoom, foco, brillo, calidad de la imagen, ajuste de la misma, alineamiento, correcciones de color, apertura de campo, interface con otros equipos, etc. La elección de los proyectores debe ser realizada

individualmente y es imposible proponer un sistema de proyección de carácter general. No obstante, comentaremos brevemente su posible clasificación así como sus parámetros determinantes. Tipos de proyectores [11]:

- Light Cristal Display (LCD) o Digital Light Projector (DLP), empleados para presentaciones, portables, y de precio económico.
- Multichip LCD o DLP, diseñados para instalaciones permanentes
- Proyectores de altas prestaciones que emplean tecnología CRT, LCD, DLP...

A continuación detallaremos los parámetros más significativos que se deben de tener en cuenta así como sus valores típicos. En cuanto a la resolución, valores entre 1024x768 y 1280x1024 pixels suelen ser adecuados; este parámetro se ve influenciado por el tamaño de la pantalla de proyección y la distancia a la que éstos estén situados. Un brillo de entre 500 a 1000 lumens es suficiente para una sala iluminada tenuemente.

Debe de permitir color real uniforme a lo largo de toda la proyección al menos un 80% y la posibilidad de su corrección automática en función del color de los restantes proyectores del sistema. Los proyectores económicos no suelen permitir una fácil modificación del color, su bloqueo y su conexión con otros de forma que automáticamente puedan obtener la misma calidad cromática.

Los proyectores deben tener una amplia variedad de frecuencias de muestreo, típicamente esta tasa se centra entre 80 y 100 f/s la cual no es suficiente para el estéreo activo. Por tanto, deberemos asegurarnos que esta tasa es suficiente para la aplicación final que se pretende desarrollar.

En cuanto a los sistemas ópticos, zoom, focos, etc., deberemos asegurarnos que no introducen distorsiones en las imágenes y es posible estabilizarlo con respecto a los otros elementos.

Materiales de las pantallas

En los últimos años el uso de pantallas de proyección se ha popularizado incluso a un nivel de consumo doméstico. Esta popularización ha incrementado el mercado de materiales empleados para su construcción, no obstante debido a las especiales características de nuestro sistema no todos ellos son aplicables. Las pantallas deben tener unas características especiales en cuanto a su reflectancia, contraste, posibilidad de ser encontrada en distintos formatos, solidez, peso y por supuesto propiedades ópticas.

En primer lugar el material debe tener unas características físicas suficientes. Por ejemplo, en proyecciones de gran tamaño el material debe poseer un equilibrio entre peso y flexibilidad de forma que no se doble por su propio peso, en cuyo caso no es necesario una estructura de soporte. El material debe mantener las propiedades ópticas de la luz que sobre ella incide, en especial la capacidad de mantener la polarización de la luz incidente. Si, por ejemplo, queremos realizar una proyección estéreo pasivo

donde la luz debe estar polarizada, el material debe mantenerla. Esta serie de propiedades debe ser indicada por el fabricante.

Elección de plataforma y sistemas operativos

Actualmente dos tipos de sistemas de computadores se emplean para el desarrollo de sistemas C.L.S.. Estaciones gráficas con múltiples tarjetas gráficas (*pipes*) y cluster de PC's. Cada uno, de las arquitecturas resuelve los mismos retos: manejo de datos, velocidad de cálculo, ancho de banda de los sistemas gráficos, sincronización de *displays*, y desarrollo de aplicaciones desde distintos puntos de vista.

Estaciones de trabajo

Las estaciones de trabajo principalmente de la compañía S.G.I. (Silicon Graphics Interface) en su gama Onyx2 presentan una arquitectura escalable ya que es posible que nuevas *pipelines* gráficas puedan ser instaladas sobre el mismo computador y proporcionan así un dispositivo multimonitor o multiproyección. Esta estrategia simplifica el desarrollo de aplicaciones "title display" al facilitar la coordinación del sistema pero presenta dos problemas:

Primero, el número de tarjetas gráficas por máquina viene condicionada por su diseño, siendo no flexible y no escalable; y segundo, el elevado precio de los equipos que no siempre los hace accesibles a cualquier grupo de trabajo.

Granjas de PC's

Una interesante alternativa al sistema basado en Workstation es el empleo de *cluster* o granjas de PC's. Esta debe abordar los problemas antes planteados desde un nuevo punto de vista:

Primeramente, los cluster de PC's no pueden compartir memoria. Lo que hace necesario dividir y distribuir los procesos de generación de la imagen de forma que cada uno de los PC's procese una parte de la misma y su unión mediante un "mosaico" genere la imagen final.

Segundo, aunque la capacidad gráfica de las aceleradoras para este tipo de equipos crece día a día, todavía dista del de las Workstation. Por tanto, deberemos desarrollar soluciones imaginativas que permitan la visualización de escenas complejas en tiempo real.

Tercero, la granja de PC's necesita tener una red de trabajo con un buen ancho de banda que debe permitir compartir datos y sincronizar los procesos. Una mala sincronización o latencias bajas en la red pueden producir problemas de actualización y coordinación en los procesos.

Como podemos intuir a pesar de los problemas que a priori plantea una aproximación de este estilo ofrecería una relación precio/rendimiento mayor que las Workstation. De este modo, la visualización en tiempo real de escenas complejas mediante cluster de PC's se ha convertido en uno de los mas interesantes campos de la computación gráfica actual [Argonne, AT&T Infolabs].

Arquitectura en una granja de PC's

Hay varias posibles arquitecturas que podemos definir en un *cluster*, las cuales determinarán, su rendimiento, coste, portabilidad, escalabilidad y flexibilidad. Es posible clasificarlas en función de dos elementos: la configuración de los nodos que componen el cluster y la distribución de dispositivos sobre dichos nodos.

Todos los nodos pueden ser iguales en cuyo caso hablamos de nodos homogéneos o diferentes tratándose entonces de una granja de nodos heterogéneos. Si los dispositivos son distribuidos de una forma homogénea, es decir, todos los elementos del nodo tienen acceso a los mismos dispositivos, diremos que la distribución es simétrica y en caso contrario asimétrica. En el caso de una distribución asimétrica existirá un nodo dominante el cual soportará mayor parte del trabajo y estará por tanto dotado de una serie de dispositivos especiales.

Las diferentes arquitecturas pueden variar desde heterogéneas asimétricas a homogéneas y simétricas. A continuación presentamos un ejemplo de cluster heterogéneo y asimétrico (figura 8). Por ser asimétrico los nodos pueden presentar características hardware que los hacen diferentes, por ser asimétrico los dispositivos se encontrarán centralizados en un único elemento del cluster. Así, el elemento dominante tendrá varias tarjetas gráficas o múltiples salidas de video a cada una de las cuales se conectaría su proyector de video o monitor. Este gestionará la información procesada por los restantes elementos del cluster. Esta arquitectura no es

demasiado flexible ya que está implementada sobre mínimas fuentes. Sí es verdad que el precio de su implementación puede ser considerablemente bajo, ya que teóricamente cualquier equipo puede ser incorporado al cluster.

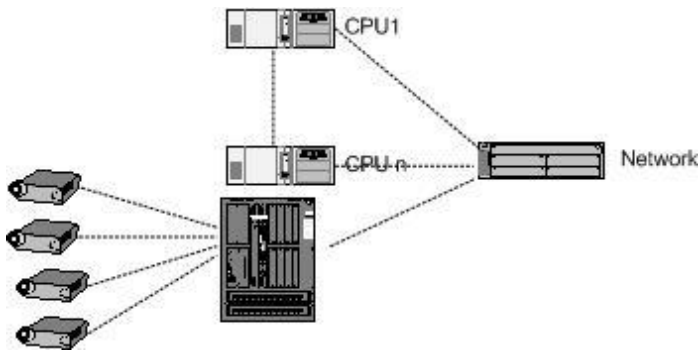


Figura 8: Organización de un cluster asimétrico y heterogéneo

Los cluster simétricos y homogéneos, (figura 9) están basados en computadores de similares características conectados a una red, todos ellos dotados de tarjetas gráficas y sistemas de proyección con preformase equivalentes. Esta es la mejor arquitectura en lo que se refiere a flexibilidad, performance, escalabilidad y facilidad para el desarrollo de aplicaciones. Este tipo de arquitectura cuenta con un ordenador de consola o *server* mediante el cual es posible gestionar el resto de equipos repartiendo las tareas a realizar por cada uno de ellos.

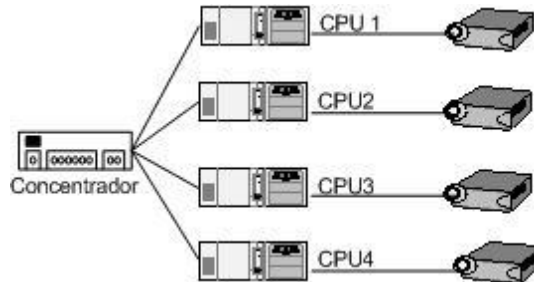


Figura 9: Organización de un cluster simétrico y homogéneo

Las aplicaciones

El desarrollo de aplicaciones sobre cluster de PC's es complejo. Esta complejidad aumenta si se debe trabajar en tiempo real como es el caso de aplicaciones V.R.. En la bibliografía es posible encontrar diversas experiencias sobre el desarrollo de este tipo de aplicaciones sobre arquitecturas similares a la nuestra y que pueden ser extrapoladas en mayor o menor medida a nuestro proyecto; éstas son:

- Diseño a medida
- Aplicación distribuida
- Distribución de las 2D/3D primitivas.

Diseño a medida

En este caso la aplicación está compuesta por dos elementos: un controlador de la aplicación y la aplicación propiamente dicha. El

controlador se ejecuta sobre el *server* y la aplicación en cada uno de los clientes de cluster. El controlador envía comandos sobre la red que la aplicación de los clientes lee y que cada uno de ellos interpretará en función de un conjunto de parámetros que lo identifican dentro del cluster. Las aplicaciones leerán el escenario o elementos necesarios de un disco compartido para todas las aplicaciones y visualizarán la parte de la escena que le ha sido encomendada.

Este tipo de implementación se emplea en aplicaciones que no necesitan trabajar en tiempo real ya que el proceso de carga y descarga de la escena por parte de las aplicaciones es lento.

Distribución de las 2D/3D primitivas

Este método a diferencia de los anteriores que trabajan básicamente a nivel de usuario, es operativo a nivel de *driver* no siendo necesario realizar modificaciones en la aplicación para su funcionamiento en el cluster.

Cuando se trabaja bajo el sistema operativo (S.O.) Windows se tiene cierta ventaja respecto a otros S.O. por su estructura. Como es sabido, el método de programación en Windows es mediante *API's* cuyas librerías se encuentran almacenadas en DLL (librerías de acceso dinámico). Es posible, por tanto, sustituir estas librerías o interceptar las instrucciones o llamadas que se realizan a la tarjeta gráfica modificando el comportamiento que se realizaría y adaptándola a nuestras necesidades.

Esta estrategia posee una gran ventaja respecto a las anteriores dada su total transparencia respecto al usuario y programador. En este sentido todas las aplicaciones desarrolladas para un único elemento son automáticamente extrapolables al cluster.

Un problema inherente a este tipo de diseño es la necesidad de actualizar el nuevo *driver* desarrollado en función del nuevo hardware del mercado.

Aplicación distribuida

Cuando hablamos de una aplicación distribuida en este tipo de sistemas tratamos con aplicaciones distribuidas a nivel de comando. Es decir, cada PC del cluster tiene una copia de la aplicación que ejecuta de un modo u otro en función de comandos que recibe. Los comandos proceden de una nueva copia de la aplicación que se ejecuta en un *server*. La copia del *server* tiene características especiales ya que coordina el cluster, realiza tareas comunes a todos los nodos y actúa de interfaz con el usuario.

Interface Aplicación-Usuarios

Uno de los principios de la VR es la inmersión e interacción en espacios virtuales [3]. Por lo tanto, la aplicación necesita recibir y proporcionar información al usuario; posición de su punto de vista, interacción con el entorno virtual y la generación del sentido de la presencia. Estas tareas se realizan mediante distintos tipo de elementos, los más comúnmente empleados son:

- Sistemas de posicionamiento *Tracking*.
- Uso de *joystick*.
- Conversores analógicos digitales, PLC's

Los sistema de posicionamiento, *trackers*, nos permiten conocer esta información y procesarla de un modo adecuado. El joystick como elemento fundamental de navegación en espacios virtuales es actualmente de amplia utilización en especial en la industria de los videojuegos. Por último conversores digitales y PLC's que no permitirán acceder más fácilmente a otros elementos del entorno tales como motores, actuadores, ventiladores,...

Todos estos elementos han sido combinados en el sistema. El punto de vista del sistema se conoce en base a un sistema de *tracking* que el usuario utiliza. Del mismo modo y a fin de que pueda desplazarse por el sistema de una manera dinámica, el usuario el usuario utiliza un *joystick* bien sea en su modalidad de palanca o volante. Una tarjeta de conversión analógico digital nos permite interactuar con distintos elementos del entorno.

Bibliografía

- [1] Alborzi, H., Druin, A., Montemayor, J., Sherman, L., Taxn, G., Best, J., Hammer, J., Kruskal, A., Lal, A., Plaisant, S.T., Sumida, L., Wagner, R., Hendler, J., 2000. "*Designing StoryRooms: Interactive Storytelling Spaces for Children.*" In Proceedings of Designing Interactive Systems ACM Press, pp. 95-104.

- [2] Anderson, D.B., Casey, M.A., “*The sound dimension*”, IEEE Spectrum, March, 1997, pp. 46-50.
- [3] Bowman, D., Johnson, D., Hodges, L., 2001. “Testbed Evaluation of Virtual Environment Interaction Techniques. *Presence: Teleoperators and Virtual Environments*”, vol. 10 (1), pp. 75-95.
- [4] Bowman, D., Kruijff, E., LaViola, J., Poupyrev, I., 2001. “An Introduction to 3D User Interface Design. *Presence: Teleoperators and Virtual Environments*”, vol. 10 (1), pp. 96-108.
- [5] Cheney S., Ichnowski, J., Forsyth, D., March/April 1999, “*Dynamics modeling and culling*”, IEEE Computer Graphics and Applications, vol. 19(2), pp. 79-87.
- [6] Cruz-Neira C., Sandin, D., DeFanti, T., 1993. “*Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE*”, In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, New York, pp. 135-142.
- [7] Flores, J., Arias, J., Saavedra, S., Varela, E., Ferro, J.M., Saco, P., 2000. “*Una aproximación a sistemas inmersivos colectivos basada en arquitectura PC*”, CEIG’2000, pp. 93-96.
- [8] Friesen, J.A., Tarman, T.D., 2000. “*Remote High-Performance Visualization and Collaboration*”, IEEE Computer graphics and Application, vol. 20, pp. 45-49.

- [9] Schaufler, G., 1996, “*Exploiting Frame-to-Frame Coherence in a Virtual Reality System*”, VRAIS 96 Proceedings, pp. 95-102.
- [10] Hereld, M., et al., 2000, “*Developing Tiled Projection Displays Wall System*”, IEEE Computer graphics and Application, vol. 20 (4), pp. 29-37.
- [11] Hereld, M., Steven, R.L., 2000, “*Introduction to building Projection-based Tiled Displays System*”, IEEE Computer graphics and Application, vol. 20 (4), pp. 22-28.
- [12] Hodges L., Pair, J., Jensen, C., Flores, J., Wilson, J., Gotz, D., 2000. “*The NAVE: Design and Implementation of Non-Expensive Immersive Virtual Environment*”, Siggraph2000 Sketches and Applications, pp. 249-250.
- [13] Hubbard, P., 1995. “*Collision Detection for Interactive Graphics Application.*” IEEE Trans. On Visualization and Computer Graphics, vol. 1 (3), pp. 218-230.
- [14] Humphreys, G., Hanrahan, P., 1999. “*A Distributed Graphics System for Large Tled Displays*”, Proc. IEEE Visualization, ACM Press, New York.
- [15] I3DGS, April 1999. Demo presented in Interactive 3D Graphics Symposium’99 Atlanta, Georgia.

- [16] Leight, J., et al., 1999. "*A review of Tele-Immersive Applications in the CAVE Research Network*", Proceedings Virtual Reality'99 Conference (VRAIS'99), Huston, Texas, pp. 180-187.
- [17] Rafferty, M.M., Aliaga, D.G., Lastra, A.A., 1998. "*3D Image Warping in Architectural Walkthroughs*", Proceedings Virtual Reality'98 Conference VRAIS '98, March 14-18, pp. 228-233.
- [18] Preparata, F.P., Shamos, S., 1985. *Computacional Geometry: An Introduction*, Springer-Verlag, New York.
- [19] Ribarsky, W., 1998. "*The times they are a-changing: PC graphics moves in*", IEEE Computer Graphics and Applications, May/June 1998, vol. 18 (3), pp. 20-25.
- [20] Roy, T.M., Cruz-Neira, C., DeFanti, T.A., 1995. "*Cosmic Worm in the CAVE: Steering a High Performance Computing Application from a Virtual Environment.*" PRESENCE: Teleoperators and Virtual Environments, MIT Press, vol. 4 (2), pp. 121-129.
- [21] Sanmartin, J.R., 1984. "*O Botafumeiro: parametric pumping in the Middle ages.*" Am. Journal of Physics, vol. 52 (10), pp. 937-945.
- [22] Schikore, D.R., et al., 2000. "*High-Resolution Multiproyector Display Walls*", IEEE Computer graphics and Application, vol. 20 (4), pp. 38-44.

- [23] Schubert, T.W., Friedmann, F., Regenbrecht, H.T., April 1999. “*Decomposing the Sense of Presence: Factor Analytic Insights.*” 2nd International Workshop on Presence, University of Essex, UK, pp. 6-7.
- [24] Schubert, T.W., Friedman, F., Regenbrecht, H.T., 1999. “*Embodied Presence in Virtual Environments.*” Visual Representations and Interpretations, Springer-Verlag, London, pp. 269-278.
- [25] Serón, F.J., Gutierrez, D., Magallón, J.A., 2002. “*Case study: full-size virtual model of trains.*” Proceeding SIAG 2002, pp. 49-58.
- [26] Sheridan, T.B., 1999. “*Descartes, Heidegger, Gibson, God: Towards and Eclectic Ontology of Presence.*” Presence, vol. 8 (5), pp. 551-559.
- [27] Singhal, S., Zyda, M., July 1999. “*Networked virtual environment: Design and implementation*”, ACM Press Siggraph Series, New York, ISBN: 0-201-32557-8.
- [28] SIPAC, October 1999. Demo presented in SIPAC’99 International Symposium of Heritage and Culture. Santiago de Compostela, Spain.

- [29] Slater, M., Usoh, M., 1993. “*Representation Systems, Perceptual Positions, and Presence in Immersive Virtual Environments.*” *Presence*, vol. 2 (3), pp. 221-33.
- [30] Tanenbaum, A., 2003.”*Computer Networks, Fourth Edition.*” Prentice Hall, ISBN: 0-13-066102-3.
- [31] Walker, J., Walker, J., 1977. “*The Flying Circus of Physics.*” John Wiley & Sons, pp. 38, ISBN: 0-471-02984-X.
- [32] Wei, B., et al, 2000. “*Visualization Research with Large Displays*”, *IEEE Computer graphics and Application*, vol. 20, pp. 50-54.
- [33] Whitaker, R., Crampo, C., Breem, D., Tuceryan, M., Rose, E., 1995. “*Object Calibration for Augmente Reality.*” *Computer and Graphcis Forum*, vol. 14 (3), pp. 15-27.
- [34] Witmer, B.G., Singer, M.J., 1998. “*Measuring Presence in Virtual Environments: A Presence Questionnaire.*” *Presence*, vol. 7 (3), pp. 225-40.

Predicción No Supervisada de Entornos Cambiantes: Aplicación Práctica al Problema de las Mareas Rojas

Florentino Fdez-Riverola
Departamento de Informática
Escuela Superior de Ingeniería Informática
Universidad de Vigo
Email: riverola@uvigo.es

Introducción

Las mareas rojas (*red tides*), son un tipo de fenómeno biológico marino caracterizado por la proliferación masiva y esporádica (*bloom*) de especies fitoplanctónicas. La intensidad de estos fenómenos puede ser tal, que la concentración de estos microorganismos sea tan elevada como para cambiar la coloración del agua del mar, de ahí su nombre. En determinadas ocasiones estos episodios resultan ser tóxicos (HAB, *Harmful Algae Blooms*), como el producido por algunas especies de diatomea del género *pseudo-nitzschia spp.*, asociada a episodios de toxicidad amnésica (ASP, *Amnesic Shellfish Poisoning*). Aunque estos episodios pueden ser frecuentes, en general son impredecibles y su duración es breve e irregular.

Con el fin de resolver el problema que presenta la predicción de este tipo de fenómenos en la costa gallega, en este trabajo se desarrolla un

sistema híbrido neuro-simbólico, capaz de predecir parámetros en entornos complejos y dinámicos de forma no supervisada. En situaciones en las que no se conocen las reglas que determinan la evolución de un sistema, la predicción de parámetros que caracterizan su comportamiento puede ser una tarea complicada. En estas situaciones, se ha constatado que un modelo híbrido centrado en un sistema de razonamiento basado en instancias (*Instance-Based Reasoning*) puede ofrecer un método más efectivo de realizar predicciones que el proporcionado por otras técnicas conexionistas o simbólicas.

Modelo de predicción desarrollado

El modelo implementado utiliza un sistema de razonamiento basado en instancias [1] que incorpora una estructura adaptativa de nodos (EAN) [2], una red de funciones de base radial (RBF) [3] y un conjunto de sistemas difusos de tipo Sugeno [4], con el objetivo de proporcionar una predicción lo más precisa posible. Cada una de estas técnicas, se utiliza en una etapa diferente del ciclo de vida del sistema IBR para recuperar datos históricos, adaptarlos al problema actual y revisar la solución propuesta.

Cada vez que es necesario obtener una nueva predicción, el sistema híbrido evoluciona a través de 3 etapas que permiten: (i) identificar las situaciones pasadas más semejantes a la actual, (ii) adaptar el conocimiento existente para generar una predicción inicial y (iii) validar o ajustar el valor generado mediante un mecanismo de revisión. El ciclo se completa, cuando se conoce el valor real para el cual se generó la predicción. La fase de

aprendizaje garantiza que el sistema evoluciona de forma paralela a como lo hace el sistema real, basando las acciones correctoras en el cálculo del error cometido con anterioridad.

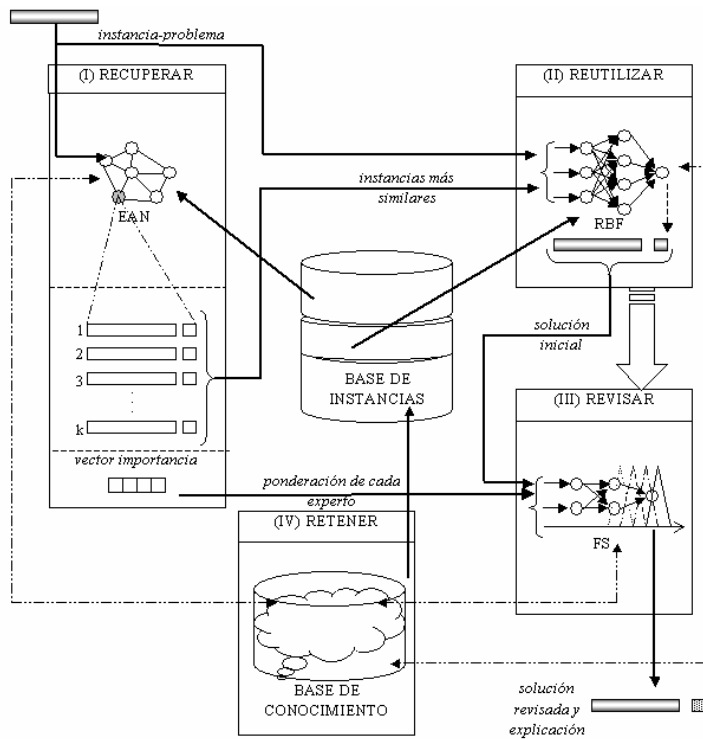


Figura 1: Arquitectura del sistema híbrido desarrollado

La metodología seleccionada como marco para la construcción del sistema híbrido, es la proporcionada por los sistemas de razonamiento basado en instancias, automatizando completamente las cuatro fases de su

ciclo de vida clásico. La figura 1 muestra las relaciones de alto nivel existentes entre los distintos componentes del sistema híbrido. El ciclo de operación está inspirado en el trabajo de Aamodt y Plaza [5].

Tecnologías empleadas

La figura 2 muestra un resumen de las tecnologías empleadas en cada etapa del sistema IBR. Para cada una de ellas se muestran los datos de entrada, el proceso llevado a cabo y los datos de salida que serán utilizados en etapas posteriores del proceso de predicción.

FASE-IBR	Tecnología	Entrada	Proceso	Salida
Recuperación	Red EAN.	Instancia-problema.	Recuperación de todas las instancias pertenecientes al prototipo asociado a la instancia-problema.	k instancias más similares. Vector importancia asociado.
Reutilización	Red RBF.	Instancia-problema. k instancias más similares.	Adaptación de la red con las k instancias recuperadas y generación de una predicción inicial.	<i>Solución inicial:</i> concentración de pseudo-nitzschia spp.
Revisión	4 Sistemas difusos.	Instancia-problema. Vector importancia. Solución inicial.	Cada uno de los 4 sistemas difusos genera una solución que se combina para contrastarla con la predicción de la red RBF.	<i>Solución confirmada:</i> concentración de pseudo-nitzschia spp.
Aprendizaje	Red EAN. Red RBF. 4 Sistemas difusos.	Instancia-problema. Error en la predicción.	Actualización de las configuraciones de las redes EAN y RBF y los 4 sistemas difusos en función de la precisión de la predicción generada.	Parámetros de configuración de las redes EAN y RBF, así como de los 4 sistemas difusos.

Figura 2: Tecnologías que componen el sistema híbrido

Funcionamiento del sistema híbrido de IA

En el entorno operacional, diariamente se introducen nuevos datos al sistema procedentes de cálculos realizados a partir de boyas oceanográficas, información meteorológica perteneciente a diferentes sensores y datos generados a partir del análisis de imágenes enviadas por satélite. Todos estos datos son almacenados en una base de datos temporal, hasta que se obtiene toda la información necesaria para construir un nuevo descriptor de problema (o instancia-problema). Este descriptor está formado por un vector numérico, que contiene el conjunto de valores que definen e identifican una situación, para la cual el sistema deberá llevar a cabo un ciclo completo de razonamiento (un nuevo problema para el que habrá que realizar una predicción).

Con una periodicidad semanal, se recoge información de distintas estaciones oceanográficas que forman parte de la red de monitorización costera del CCCMM [6], completando el ciclo de adquisición de información y generando el vector de valores que define al nuevo problema, a partir de los datos almacenados en la base de datos temporal. A este descriptor de problema generado con periodicidad semanal, se le denomina instancia-problema, y describe numéricamente en el sistema un nuevo caso real para el cual es necesario generar una predicción.

Semanalmente se presenta una nueva instancia-problema (parte superior-izquierda de la figura 1) al sistema. La red EAN identifica la clase o prototipo de problemas más similar al problema actual (etapa 1 en la

figura 1), que se utiliza para recuperar de la memoria del sistema IBR aquellas instancias (situaciones anteriores), cuyo descriptor sea más semejante al que define el problema presente (o instancia-problema). Debido a la estructura de la red EAN utilizada como mecanismo de indexación y cálculo de similitud, el número de instancias inicialmente seleccionadas es variable, dependiendo de la posición que ocupe el vector instancia-problema en el espacio de entrada y de la densidad de información alrededor de ese punto.

Las k instancias recuperadas en la fase anterior, se utilizan para reentrenar la red RBF modificando el valor de los pesos y la posición de los centros, pero manteniendo constante el número de éstos (etapa *II* en la figura 1). Tras esta fase de adaptación de la red RBF, el nuevo descriptor de problema se presenta a la red y se genera una solución inicial (predicción propuesta).

Durante el proceso de revisión (etapa *III* en la figura 1), la solución inicial generada en la fase anterior se ajusta con el objetivo de obtener una solución final contrastada. Para ello, a partir de la configuración de la red RBF existente en la base de conocimiento (parte inferior de la figura 1), se construyen varios sistemas de inferencia difusos que constituyen el subsistema de revisión del modelo. Para cada prototipo (agrupación de instancias similares) identificado por la red EAN, se mantiene un vector local de pesos que pondera el grado de exactitud de cada sistema difuso a la hora de realizar la revisión de la predicción inicial. La solución inicial propuesta por la red RBF se contrasta con la salida generada por cada uno de los sistemas difusos, aceptando o refutando dicho valor y generando una

explicación (en forma de reglas difusas) de la solución adoptada (predicción final contrastada).

El proceso de aprendizaje (etapa *IV* en la figura 1), se lleva a cabo cuando se conoce el valor real de la solución al problema planteado. Se crea un nuevo caso añadiendo a la instancia-problema el valor real de la concentración de la diatomea *pseudo-nitzschia spp.*, e incorporándolo a la memoria del sistema IBR para su uso posterior. El error cometido en la predicción de la instancia-problema, se utiliza para actualizar localmente las estructuras de conocimiento de cada uno de los subsistemas que componen el híbrido.

Aplicabilidad del modelo desarrollado

La producción de mejillón en España es de unas 75 toneladas anuales por batea. Del total de la producción española, la comunidad gallega genera el 98% con un total de 3.242 bateas. A nivel mundial, España ocupa el segundo puesto después de China, con una producción total de 255.928 toneladas anuales, lo que supone un movimiento económico aproximado de 108 millones de euros (18.000 millones de pesetas). A parte de los ingresos económicos, esta actividad productiva genera aproximadamente 6.000 puestos de trabajo directos y 15.000 indirectos. El desarrollo de especies fitoplanctónicas de forma masiva en las costas gallegas, afecta drásticamente a esta industria de tanto auge e importancia. Los daños ocasionados por estos episodios causan secuelas a dos niveles: (i) afectando a la salud de los consumidores y (ii) produciendo daños a los

organismos cultivados. En este último caso se producen grandes pérdidas económicas, tanto por el deterioro o la muerte de los cultivos, como por la imposibilidad de su comercialización.

El modelo propuesto se ha adaptado al problema de la predicción de mareas rojas, en las aguas costeras del noroeste de la Península Ibérica. En este sentido, la posibilidad de poder predecir la aparición de especies tóxicas, conlleva una mejora en la capacidad para poder detectar la toxicidad de los moluscos a tiempo, disminuyendo la probabilidad de que moluscos contaminados lleguen al mercado. En caso de que existan especies perjudiciales difícilmente detectables por técnicas de bioensayo, también permite advertir al sector del peligro existente. En ambos supuestos, se disminuye el problema económico y se elimina el problema sanitario.

Bibliografía

- [1] Fdez-Riverola, F., Corchado, J.M., 2003. “*Forecasting Red Tides Using an Hybrid Neuro-Symbolic System*”. *AI Communications*, vol. 16 (4), pp. 221-233.
- [2] Azuaje, F., Dubitzky, W., Black, N., Adamson, K., 2000. “*Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach*”. *IEEE Transactions on Systems, Man and Cybernetics, B*, vol. 30 (3), pp. 448-460.

- [3] Haykin, S., 1999. “*Neural Networks: a Comprehensive Foundation*”. 2nd Edition. Prentice Hall, ISBN: 0-132-73350-1.
- [4] Takagi, T., Sugeno, M., 1985. “*Fuzzy identification of systems and its applications to modeling and control*”. IEEE Transactions on Systems, Man and Cybernetics, vol. 15, pp. 116-132.
- [5] Aamodt, A., Plaza, E., 1994. “*Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches*”. AI Communications, vol. 7 (1), pp. 39-59.
- [6] CCCMM, 2002. “*Centro de Control da Calidade do Medio Mariño*”. [<http://www.cccmm.cesga.es>].



Búsqueda Aproximada de Patrones de Puntos en 2D y 3D

Arno Formella

Departamento de Informática

Universidad de Vigo

Email: formella@ei.uvigo.es

Introducción

La búsqueda de información específica en un gran conjunto de datos es una tarea común en muchas aplicaciones. Cualquier usuario ya ha realizado alguna vez una búsqueda de todas las ocurrencias de cierta palabra en un texto, incluso en el caso de que no se busque solamente la palabra exacta, sino también ciertas variaciones. La tarea se complica bastante si se trata de búsquedas aproximadas en espacios geométricos de dos o tres dimensiones (2D o 3D) [1]. Por ejemplo, la búsqueda en una base de datos de huellas dactilares de todas aquellas huellas que tengan cierta similitud con una huella encontrado parcialmente en el lugar del crimen es una tarea bien conocida, pero su solución no es tan obvia [15].

Otro ejemplo fácil de entender es la búsqueda de una constelación estelar en un atlas astronómico de la cual se ha tomado una fotografía [16]. A lo mejor el observador quiere averiguar si hay nueva información en la

toma porque se ha mejorado la técnica del telescopio o porque ha nacido una nueva estrella en el espacio.

Un ejemplo 3D sería el caso de buscar una zona activa, es decir, una zona donde actúa por ejemplo un medicamento, en un conjunto de proteínas. Así se puede comparar y alinear proteínas en su estructura tridimensional que podría ser una ayuda para diseñar nuevos medicamentos basándose en la información estructural de las proteínas [6, 7, 10, 13, 17].

Presentamos en esta ponencia un algoritmo que se puede aplicar en muchos de los campos descritos. Hablando de forma más abstracto se trata de, dado un conjunto de puntos como patrón (normalmente pocos puntos con pequeña extensión espacial) y otro conjunto de puntos como espacio de búsqueda (normalmente muchos puntos con cierta extensión espacial), calcular la parte más grande del patrón y el sitio dónde dicha parte se alinea mejor en el espacio de búsqueda. El algoritmo aprovecha de conceptos de optimización tales como minimización de funciones con o sin derivadas, retro-propagación, Algoritmos Genéticos y Búsqueda Tabú Reactiva. Para llegar rápidamente a soluciones aceptables, el algoritmo necesita estructuras de datos eficientes y emplea heurísticas adaptadas al problema concreto.

Formalización del problema

Informalmente podemos describir el problema propuesto como: dado un espacio de búsqueda y un patrón, encuentra el sitio en el espacio que mejor representa el patrón. Una extensión natural es el problema de encontrar la parte más grande del patrón que esté representado de forma

aproximada en el espacio de búsqueda. Con otras palabras, estamos interesados en encontrar las partes comunes de dos conjuntos de puntos. En casos prácticos, a lo mejor, también se quiere permitir cierta deformación del patrón.

Emparejamientos

Sean el espacio de búsqueda $S = \{s_0, s_1, \dots, s_{n-1}\} \subset \mathfrak{R}^d$ y el patrón de búsqueda $P = \{p_0, p_1, \dots, p_{k-1}\} \subset \mathfrak{R}^d$ dos conjuntos de puntos discretos en el espacio euclidiano de la dimensión $d = 2$ o $d = 3$. Asumimos que el patrón no tenga más puntos que el espacio de búsqueda, es decir $|S| = n \geq k = |P|$.

Un emparejamiento es una función que asigna a cada punto del patrón de búsqueda un punto diferente del espacio de búsqueda. Entonces, un emparejamiento es una función inyectiva $\mu: P \rightarrow S$, es decir, si p_i es diferente a p_j también $\mu(p_i)$ es diferente a $\mu(p_j)$. Para abreviar la notación, escribimos $S' = \mu(P)$ y $s_i' = \mu(p_i)$.

Una pregunta importante en relación con un emparejamiento es: ¿Cómo podemos transformar el patrón para que esté posicionado de cierta manera próxima a su homólogo en el espacio de búsqueda?

Transformaciones

Una transformación es una función que asigna a cada punto del espacio euclidiano un nuevo punto en el mismo espacio, es decir, una transformación es una función $T : \mathfrak{R}^d \rightarrow \mathfrak{R}^d$. Para abreviar la notación escribimos para un punto transformado $p_i' = T(p_i)$ y para todo el conjunto transformado $P' = T(P)$.

Las transformaciones que consisten solamente en una combinación de traslaciones, rotaciones y reflexiones mantienen las distancias, es decir, para dos puntos $p_i, p_j \in P$ tenemos $d(p_i, p_j) = d(p_i', p_j')$ siendo d una función que calcule la distancia entre dos puntos. Si permitimos también un escalado solamente los ángulos se mantienen iguales después de haber aplicado una transformación, es decir, para tres puntos $p_i, p_j, p_k \in P$ tenemos $\angle(p_i, p_j, p_k) = \angle(p_i', p_j', p_k')$. Las transformaciones que deforman el conjunto son, por ejemplo, el sesgado o las proyecciones. En el contexto de la búsqueda de patrones se puede pensar también en transformaciones que realizan deformaciones locales, por ejemplo, en el caso de buscar pequeñas moléculas dentro de proteínas se puede permitir cierta flexibilidad de los patrones.

Las transformaciones que consisten solamente en una combinación de traslaciones y rotaciones se llaman transformaciones rígidas, o también transformaciones euclidianas o transformaciones congruentes. Si se permite

también un escalado se habla de transformaciones similares. Podemos observar que dichas transformaciones son reversibles.

Alineamiento

Llamamos un emparejamiento μ en combinación con una transformación T un alineamiento (μ, T) . Si la transformación es una transformación rígida (o congruente), se trata de un alineamiento congruente. Si la transformación es una transformación similar, se trata de un alineamiento similar. Si se permite una reflexión en la transformación, llamamos el alineamiento un L-alineamiento; congruente o similar según el caso.

Dado un alineamiento (μ, T) de P en S , podemos calificar las distancias entre los puntos del patrón transformados y sus correspondientes parejas, es decir, podemos calcular las distancias $d_i = d(p_i', s_i')$ entre todas las parejas. Obviamente, si $d_i = 0$ para todos los puntos $p_i \in P$, el alineamiento es perfecto.

Para alineamientos no perfectos, introducimos una métrica para comparar dos alineamientos que se distinguen en su transformación pero que coinciden en su emparejamiento. Existen muchas posibilidades de definir tal métrica. Ejemplos son la raíz de la distancia cuadrática media

$$d = \sqrt{\frac{1}{n} \sum_i (p_i' - s_i')^2}$$

la distancia media entre las parejas de puntos

$$d = \frac{1}{n} \sum_i |p_i' - s_i'|$$

y la distancia máxima entre las parejas de puntos

$$d = \max_i |p_i' - s_i'|$$

Depende de la aplicación cual de las métricas es la más conveniente.

Las distancias d son valores en el intervalo $[0, \infty[$. Para comparar más tarde también alineamientos parciales, definimos la calidad Q de un alineamiento como $Q = 1/(1 + d)$ siendo d su distancia. Entonces, un alineamiento es perfecto si $Q = 1$. Como se observa fácilmente la calidad Q es un valor en el intervalo $]0, 1]$.

El problema formal

Entonces, podemos describir el problema de la búsqueda de un patrón en un espacio de búsqueda formalmente como: dado un espacio de

búsqueda S y un patrón de búsqueda P , encuentra un alineamiento (μ, T) de P en S con máxima calidad Q (o mínima distancia d).

Algoritmos

Antes de detallar algunos algoritmos para alineamientos aproximados veremos los resultados que se ha obtenido hasta hoy para alineamientos perfectos.

Alineamientos perfectos

Para alineamientos congruentes perfectos en tres dimensiones Boxer et al. [3, 4, 5] presentaron un método con el siguiente tiempo de ejecución

$$O\left(n^{2.5} \sqrt[4]{\log^* n} + \underbrace{kn^{1.8} (\log^* n)^{O(1)}}_{\text{salida}} \log n\right)$$

que es solamente por un factor $\log n$ más lento que el posible número de alineamientos que se puede encontrar en el espacio de búsqueda si el tamaño del patrón k es suficientemente grande. Si k es pequeño, el primer término en $n^{2.5}$ domina el tiempo de cálculo. Los factores en términos de $\log^* n$ podemos considerar constantes para cualquier caso práctico (si $\log^* n = 5$, entonces n ya es un número muy grande,

$n \approx 2^{65536}$). En el mismo trabajo, el tiempo de ejecución para alineamientos similares perfectos está dado como

$$O\left(n^3 + \underbrace{kn^{2.2}}_{\text{salida}} \log n\right)$$

Las principales ideas detrás del método podemos resumir como sigue. Escogemos cualquier triángulo de puntos, por ejemplo (p_0, p_1, p_2) , del patrón P . Buscamos todos los triángulos congruentes y las transformaciones adecuadas para alinear (p_0, p_1, p_2) en el espacio de búsqueda S . Verificamos los demás puntos del patrón una vez haber transformado dichos puntos. El tiempo no es directamente proporcional a n^3 en el caso de congruencia, porque se puede enumerar los triángulos en S de forma sofisticada y se puede comprobar que no hay tantos triángulos relevantes. Para la búsqueda de alineamientos similares la enumeración de los triángulos sí cuesta tiempo $O(n^3)$. Obviamente, si buscamos alineamientos aproximados y/o parciales el problema se complica más.

Alineamientos aproximados

Para encontrar alineamientos aproximados seguimos un camino muy parecido. Dividimos el problema en dos fases: buscamos primero un emparejamiento μ adecuado (según alguna tolerancia) y calculamos

después la transformación T óptima (según alguna métrica). Finalmente elegimos un mejor alineamiento.

Antes de explicar el proceso como encontrar un emparejamiento adecuado, discutimos el problema de calcular la transformación óptima para un emparejamiento dado.

Transformaciones óptimas

La raíz de las distancias cuadráticas medias, como mencionada arriba, es una métrica que se usa en muchas aplicaciones. Para la transformación rígida y óptima T , tenemos que calcular el mínimo de la función

$$d = \sqrt{\frac{1}{n} \sum_i d(p_i', s_i')^2} = \sqrt{\frac{1}{n} \sum_i (U \cdot p_i + t - s_i')^2}$$

donde U es una matriz 3×3 de rotación, es decir, U tiene que ser ortonormal, y t es un vector de traslación. Se observa rápidamente que t y U son independientes. Sabiendo que el numerador de la derivada parcial de d según t

$$\left. \frac{\partial d}{\partial t} \right|_{\text{num}} = 2 \cdot \sum_i (U \cdot p_i + t - s_i') = 2U \cdot \sum_i p_i + 2nt - 2 \cdot \sum_i s_i'$$

tiene que ser igual a cero en el valor extremo para t que estamos buscando, calculamos el vector de traslación t como

$$t = -U \cdot \frac{1}{n} \sum_i p_i + \frac{1}{n} \sum_i s_i' = -U \cdot p_c + s_c'$$

siendo p_c y s_c' los centroides de ambos conjuntos de puntos respectivamente. Si sustituimos t en la función de d por la expresión obtenemos

$$d = \sqrt{\frac{1}{n} \sum_i (U \cdot p_i + t - s_i')^2} = \sqrt{\frac{1}{n} \sum_i (U \cdot (p_i - p_c) - (s_i' - s_c'))^2}$$

Para resolver el problema restante de buscar los puntos extremos de la función bajo restricciones (recordamos que U tiene que ser ortonormal), convertimos el problema con restricciones mediante multiplicadores de Lagrange en un problema sin restricciones que tenga los mismos extremos.

Sin entrar en los detalles, una vez haber introducido los multiplicadores calculamos primeras y segundas derivadas respecto a las nueve variables libres u_{ij} que forman la matriz U y buscamos los extremos con los métodos clásicos. Este algoritmo fue presentado por Kabsch [11,12] y el proceso del cálculo necesario podemos resumir como sigue.

Sea \bar{S} la matriz de filas con los puntos s_i' y sea \bar{P} la matriz de filas con los puntos p_i donde asumimos que ya hemos eliminado la traslación tal y como está explicado arriba. Calculamos una matriz $\bar{R} = \bar{S} \cdot \bar{P}^T$ y después calculamos los vectores propios de $\bar{R}^T \bar{R}$. Sea \bar{A} la matriz de los vectores propios normalizados $\bar{A} = [\bar{a}_0 \ \bar{a}_1 \ \bar{a}_2]$. Construimos una nueva matriz $\bar{B} = \left[\|\bar{R}\bar{a}_0\| \ \|\bar{R}\bar{a}_1\| \ \|\bar{R}\bar{a}_2\| \right]$ donde las columnas se forman por los vectores que obtenemos si multiplicamos los vectores propios por \bar{R} y los normalizamos posteriormente. Con las dos matrices \bar{A} y \bar{B} obtenemos la matriz de rotación U finalmente como $U = \bar{B} \cdot \bar{A}^T$. Hay que considerar los casos especiales en los cuales los puntos de P están colocados en un plano o en una recta.

Es fácil introducir también un factor de escalado σ , es decir, la función de distancia d se extiende como

$$d = \sqrt{\frac{1}{n} \sum_i (\sigma \cdot U \cdot (p_i - p_c) - (s_i' - s_c'))^2}$$

Si usamos $p_i'' = U \cdot (p_i - p_c)$ siendo el punto p_i centrado y rotado y $s_i'' = s_i' - s_c'$ siendo el punto s_i' centrado la solución para σ óptimo es

$$\sigma = \frac{\sum_i \langle s_i'', p_i'' \rangle}{\sum_i \langle p_i'', p_i'' \rangle}$$

Una implementación ejemplar de todo el algoritmo se encuentra en [8] en forma de código abierto.

Este algoritmo de Kabsch resuelve el problema de encontrar la transformación óptima si la función de distancia está basada en la distancia cuadrática media. Puede ser que para otras funciones de distancia no exista una solución directa, sobre todo cuando las funciones no son derivables, sin embargo, dichas métricas pueden ser de interés en ciertas aplicaciones. Especialmente en muchos casos la media simple o el máximo ofrece un alineamiento más intuitivo.

Si la función para la distancia d no es derivable, necesitamos un método de optimización sin gradiente que trabaja solamente con evaluaciones de la propia función. A continuación se presenta un método iterativo recientemente desarrollado que converge seguro hacia un mínimo local [9]. La idea de Rodríguez/García-Palomares se puede resumir brevemente. Sea $f(x)$ la función por minimizar donde x simboliza el vector de las variables libres. El algoritmo itera contrayendo y dilatando adecuadamente parámetros $h^k > 0$ y $\tau > 0$ de tal manera que

$$f(x_{i+1}) = f(x_i) \pm h^k d_k \leq f(x_i) - \tau^2$$

donde d_k es una dirección escogida de un conjunto finito de direcciones. Dicho conjunto forma una base para el espacio que forma el vector de las variables libres y depende del punto x_i actual. Lo importante de dicha iteración es que en cada paso se ejecuta solamente un número finito de operaciones para adaptar los parámetros. Con $\tau \rightarrow 0$ el vector x_i converge a un óptimo local. El algoritmo funciona mientras no haya restricciones.

Para aplicar el método de Rodríguez/García-Palomares eliminamos las restricciones describiendo la rotación en el espacio de los cuaterniones. Una rotación $U \cdot P$ del punto P con la matriz U se puede expresar con la fórmula de Hamilton $q * \bar{p} * q^{-1}$ en el espacio de los cuaterniones \mathbb{H} . El espacio de los cuaterniones es un cuerpo algebraico similar a los números complejos, pero cada elemento se representa con cuatro componentes. La multiplicación en este cuerpo ya no es conmutativa. El punto \bar{p} en la fórmula de Hamilton es el cuaternión canónico del punto P , es decir, se construye $\bar{p} = (0, p) \in \mathbb{H}$. El cuaternión de rotación q en la fórmula de Hamilton se construye como $q = (\sin(\varphi/2), \cos(\varphi/2) \cdot u)$ siendo $u \in \mathbb{R}^3$ el eje de rotación normalizado y $\varphi \in \mathbb{R}$ el ángulo de rotación. Aunque la matemática parezca más compleja, la rotación con cuaterniones refleja más el proceso como nosotros describimos normalmente una rotación, es decir, con eje y ángulo, y no con una matriz ortonormal.

En vez de la matriz U con 9 variables con restricciones, hay ahora solamente 4 variables sin restricciones, es decir, el vector u y el ángulo φ . Entonces, para distancias no derivables podemos aplicar el algoritmo de Rodríguez/García-Palomares que encuentra todas las variables de la transformación, incluyendo la traslación. Para comenzar la iteración partimos de una solución aproximada basada en el algoritmo de Kabsch.

Algoritmos para encontrar emparejamientos

En la literatura se describe tres tipos de algoritmos para encontrar emparejamientos [13], cuales son: la detección de cliques, es decir, subgrafos completos, el concepto de geometric hashing, es decir, el uso de tablas de dispersión para características geométricas, y la geometría de distancias. Resumimos brevemente los dos primeros métodos y seguimos con una descripción más detallada de un programa basada en la geometría de distancias.

Detección de cliques

Para este método generamos un grafo $G(V, A)$ llamado grafo de distancias compatibles donde los vértices $v_{ij} \in V$ son todas las parejas (p_i, s_j) , y se añade una arista $a = (v_{ij}, v_{kl})$ en A , si las distancias entre los puntos correspondientes son similares, es decir,

$d(p_i, p_k) \approx d(s_j, s_l)$. En dicho grafo G buscamos cliques de tamaño k que obviamente representan emparejamientos.

Entre los inconvenientes de la detección de cliques destaca que los algoritmos rápidos necesitan matrices adyacentes para una representación del grafo [2]. Entonces resulta que para problemas todavía pequeños, por ejemplo, $n = |S| = 5000$ y $k = |P| = 100$, se llega a un consumo de memoria muy elevado, en el ejemplo unos 30 GByte de memoria contando solamente un bit por arista.

Geometric hashing

El uso de las tablas de dispersión para características geométricas [18] se basa en un preprocesado del espacio de búsqueda. Describimos el caso bidimensional. Alineamos cada pareja (s_i, s_j) del espacio de búsqueda de tal manera que s_i está posicionado en el origen y s_j en dirección x . Insertamos la información sobre cada punto de S en una tabla de dispersión definida por una rejilla de cierto tamaño sobre S . Así conseguimos finalmente que la tabla de dispersión contenga la información necesaria para todas las orientaciones posibles del conjunto de puntos S .

La búsqueda del patrón se realiza simulando una inserción de los puntos de P en la tabla de dispersión, pero se cuenta solamente las

entradas no vacías, es decir, si encontramos una pareja (p_i, p_j) de puntos del patrón de tal manera que cada otro punto del patrón encuentra una casilla llena en la tabla de dispersión, hemos encontrado un candidato para un emparejamiento.

Aunque el tiempo de búsqueda con $O(k^{d+1})$ es bastante rápido con una buena implementación de la tabla de dispersión, el tiempo de preprocesado con $O(n^{d+1})$ es bastante alto. Otro inconveniente consiste en la necesidad de la predeterminación de la rejilla que influye en el consumo y manejo de la memoria en la tabla de dispersión. Además es difícil considerar transformaciones que permiten deformaciones.

Geometría de distancias

A continuación se presenta el funcionamiento del programa psm desarrollado y implementado por el autor que está basado en la geometría de distancias. Eso significa que ambos conjuntos S y P están representados directamente como grafos de distancias. Los vértices de los grafos son los puntos de los conjuntos. Las aristas de los grafos codifican las distancias entre los puntos correspondientes.

Sean G_P y G_S los grafos de distancias adecuados (se explica más adelante los detalles). Primero buscamos subgrafos $G_{S'}$ de G_S que son

congruentes al grafo G_P permitiendo ciertas tolerancias. Después alineamos el grafo G_P con los subgrafos de $G_{S'}$ de forma óptima y elegimos el mejor alineamiento. Más adelante veremos cómo se puede ampliar la búsqueda para que funcione también con subgrafos de G_P .

Los cuatro pasos principales de psm son: i) la construcción de los grafos donde se aprovecha de propiedades de localidad, ii) la búsqueda de subgrafos con un algoritmo de retro-propagación sofisticada (o backtracking), iii) el alineamiento como explicado antes, y, finalmente, iv) la búsqueda de patrones parciales basada en una Búsqueda Tabú Reactiva [2].

Para la construcción de los grafos asumimos al principio que el patrón P sea pequeño y construimos G_P como grafo completo. Ambas restricciones se evitan más adelante. Para la construcción del grafo sobre S generamos primero un diccionario D , es decir, una estructura de datos ordenada que contiene todos los intervalos de distancias entre parejas de puntos en P . Entonces, consideramos una arista entre dos puntos para el conjunto de aristas de G_S , si su distancia se encuentra en el diccionario D .

Más preciso, sea $d_{ij} = d(p_i, p_j)$ la distancia entre dos puntos en P . El diccionario contendrá el intervalo

$$\left[(1 - \varepsilon) \cdot d_{ij}, (1 + \varepsilon / (1 - \varepsilon)) \cdot d_{ij} \right] \in D$$

siendo $0 \leq \varepsilon < 1$ una tolerancia predeterminada. El factor para el límite superior se puede simplificar a $1 + \varepsilon$, pero se pierde la simetría. Obviamente podemos unir intervalos en el diccionario D , si se intersecan los intervalos.

Si se mantiene el grafo G_P sobre el patrón como un grafo completo el conjunto de aristas del grafo G_S sobre el espacio de búsqueda puede resultar grande. Pero podemos disminuir este número de aristas significativamente, si mantenemos el grafo G_P solamente como grafo conectado y, para evitar emparejamientos con deformaciones, rígido. Preferiblemente se mantiene solamente las aristas cortas. Conseguimos una construcción del grafo G_S más rápido, si ordenamos los puntos de S previamente en una rejilla del tamaño del límite superior más grande de los intervalos en el diccionario. Así, hay que explorar solamente todas las parejas de vértices de G_S que caen dentro de una casilla o en dos casillas vecinas de la rejilla.

Para encontrar los emparejamientos, psm usa un algoritmo de retro-propagación para la búsqueda. Asumimos que G_P es un grafo completo para simplificar la explicación. Es fácil trabajar también con grafos conectados. Ordenamos los puntos de G_P en cualquier orden, por ejemplo, (p_0, \dots, p_{k-1}) y aplicamos el algoritmo de retro-propagación que intenta encontrar para cada punto p_i una pareja s_i siguiendo el orden establecido. La siguiente descripción inductiva precisa el funcionamiento del algoritmo.

Asumiendo que ya hemos hallado un subgrafo en G_{s_0, \dots, s_i} similar al grafo G_{p_0, \dots, p_i} . Buscamos candidatos s_{i+1} para el siguiente punto p_{i+1} que tienen que ser vecinos del punto s_i dentro de G_S que todavía fueron elegidos, y que tengan distancias similares a los s_j ($j \leq i$) como el p_{i+1} a los p_j ($j \leq i$). Mientras haya candidato avanzamos con i . Si no hay más candidatos para un s_{i+1} , s_i tampoco es pareja para p_i , y retro-propagamos. El algoritmo informa cada vez cuando se ha encontrado un candidato para el último punto p_{k-1} . El algoritmo termina cuando no quedan más candidatos para el primer punto p_0 o, si se quiere, cuando hemos encontrado la primera solución.

Podemos optimizar el algoritmo básico con las siguientes heurísticas. Una reducción del número de las aristas en el grafo G_P implica una reducción de las aristas en el grafo G_S . Una ordenación sutil de los puntos P_i implica una reducción de candidatos para los vecinos. Una consideración del tipo del vértice (por ejemplo, el elemento del átomo en el caso de proteínas) implica una reducción de candidatos en la búsqueda. Todas las heurísticas implican que la retro-propagación avanza más rápido.

Búsqueda de alineamientos parciales

La búsqueda de alineamientos parciales consiste en encontrar el subconjunto más grande de los puntos del patrón que mejor se puede representar en el espacio de búsqueda. Hay $|\wp(P)| = 2^k$ posibilidades para elegir un subconjunto y el problema cae en la clase de los problemas NP-completo.

Ambos métodos brevemente presentados arriba se pueden extender fácilmente para que busquen también alineamientos parciales. En el caso de la detección de cliques basta con encontrar los cliques más grandes. El número de votos menores que k indica en el caso de las tablas de dispersión candidatos para alineamientos parciales.

Se ha ampliado el programa psm que usa la geometría de distancias como base con dos extensiones para que encuentre también alineamientos

parciales. Una de ellas que no se detallará en este artículo usa un Algoritmo Genético [14], la otra el método de la Búsqueda Tabú Reactiva [2].

La idea detrás de la Búsqueda Tabú Reactiva se entiende fácilmente. Primero nos situamos en una solución factible, después buscamos posibilidades para ampliar o mejorar la solución actual. Si eso es posible elegimos una de ellas aleatoriamente y seguimos con los intentos de ampliación. Si no es posible, buscamos posibilidades para reducir o empeorar la solución actual. Para evitar movimientos repetitivos la búsqueda aprovecha de una memoria que almacena soluciones intermedias, es decir, se marca ciertos movimientos como tabú durante cierto número de iteraciones. El término reactiva significa que se adapta el periodo del tabú dinámicamente.

Para evaluar la calidad de un alineamiento parcial que es necesario para comparar soluciones intermedias, psm usa la suma entre el número de puntos alineados y la calidad de la transformación exigiendo que la calidad Q esté encima de cierto umbral. Entonces, la calidad máxima de un alineamiento perfecto es $|P| + 1$. La búsqueda de candidatos para ampliar la solución considera los vecinos (rígidos) dentro del grafo parcial ya emparejado. La búsqueda de candidatos para reducir la solución considera cualquier punto de la solución actual que mantiene el grafo conectado (y rígido).

La terminación de la Búsqueda Tabú no es tan sencilla. Normalmente se termina cuando la solución encontrada es lo

suficientemente buena o cuando se ha ejecutado cierto número máximo de iteraciones.

Con la heurística de reducir el número de aristas en los grafos de distancias podemos usar psm para buscar el patrón más grande que dos conjuntos de puntos tengan en común. Obviamente, encontramos solamente patrones compactos, es decir, aquellos que se pueden reflejar como subgrafos conectados de los grafos iniciales.

Resumen

En este artículo se ha presentado el problema de la búsqueda aproximada de patrones en espacios 2D o 3D. Se ha descrito las dificultades principales y se ha introducido a los algoritmos y estructuras de datos disponibles para resolver el problema. Se ha demostrado el programa psm que implementa un algoritmo concreto.

Entre las posibles ampliaciones del trabajo se encuentra una enumeración más rigurosa de todas las localizaciones del patrón. Hasta ahora se ha concentrado solamente en el mejor alineamiento. Las próximas versiones de psm van a implementar un método para aflojar la condición de la rigidez del grafo y tratar así la flexibilidad natural de las moléculas como la torsión de partes de ellas. Además se va a permitir tolerancias locales en el patrón y permitir cierto escalado del mismo. Esta planeado de instalar un servidor en la Red para ofrecer el servicio de buscar patrones a los usuarios de la Internet.

Agradecimientos

El autor quiere agradecer a Thorsten Pöschel y Kristian Rother su colaboración en el trabajo y a Ubaldo García-Palomares y José Rodríguez la disponibilidad de su algoritmo de optimización.

Bibliografía

- [1] Alt, H., Guibas, L., 1999. “*Discrete geometric shapes: Matching, interpolation, and approximation*”. Handbook of Computational Geometry, pp. 121-153. Elsevier Science, Publishers B.V. North-Holland.
- [2] Battiti, R., Protasi, M., 2001. “*Reactive local search for the maximum clique problem*”. Algorithmica, vol. 29 (4), pp. 610-637.
- [3] Boxer, L., 1996. “*Point set pattern matching in 3-d*”. Pattern Recognition Letters, vol. 17, pp. 1293-1297.
- [4] Boxer, L., 1998. “*Faster point set pattern matching in 3-d*”. Pattern Recognition Letters, vol. 19, pp. 1235-1240.
- [5] Boxer, L., Haralick, R., 1999. “*Even faster point set pattern matching in 3-d*”. Proceedings of the International Society for Optical Engineering Conference (SPIE), vol. 3811, pp. 168-178.

- [6] Chakraborty, S., Biswas, S., 1999. “*Approximation algorithms for 3-d common substructure identification in drug and protein molecules*”. Workshop on Algorithms and Data Structures, pp. 253-264.
- [7] Finn, P.W., Kaviraki, L.E., Latombe, J.C., Rajeev Motwani, R., Shelton, C.R., Venkatasubramanian, S., Yao, A., 1997. “*RAPID: Randomized pharmacophore identification for drug design*”. Proceedings of 13th Annual ACM Symposium on Computational Geometry, pp. 324-333.
- [8] Formella, A., 2003. “*Código Abierto*”. Página Web: [<http://www.ei.uvigo.es/~formella/inv/aaa/index.html>]
- [9] García-Palomares, U.M., Rodríguez, J.F., 2002. “*New sequential and parallel derivative-free algorithms for unconstrained optimization*”. SIAM Journal on Optimization, vol. 13 (1), pp. 79-96.
- [10] Gilbert, D., Westhead, D., Viksna, J., Thornton, J., 2002. “*A computer system to perform structure comparison using TOPS representations of protein structure*”. Computers and Chemistry, vol. 26 (1), pp. 23-30.
- [11] Kabsch, W., 1976. “*A solution for the best rotation to relate two sets of vectors*”. Acta Crystallographica A, vol. 32, pp. 922-923.

- [12] Kabsch, W., 1978. "*A discussion of the solution for the best rotation to relate two sets of vectors*". Acta Crystallographica A, vol. 34, pp. 827-828.
- [13] Lemmen, C., Lengauer, T., 2000. "*Computational methods for the structural alignment of molecules*". Journal of Computer-Aided Molecular Design, vol. 14, pp. 215-232.
- [14] Mitchell, M., 1996. "*An Introduction to Genetic Algorithms*". MIT Press.
- [15] Qun, R., Jie, T., Yuliangi, H., Jiangang, C., 2002. "*Automatic fingerprint identification using cluster algorithm*". Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), vol. 2, pp. 398-401, IEEE Computer Society Press.
- [16] Renken, H., Rath, H.J., 1997. "*Method for three-axis attitude determination by image-processed star constellation matching*". Proceedings of the International Society for Optical Engineering Conference (SPIE), vol. 3086, pp. 254-263.
- [17] Stark, A., Sunyaev, S., Russell, R.B., 2003. "*A model for statistical significance of local similarities in structure*". Journal on Molecular Biology, vol. 326, pp. 1307-1316.

- [18] Wolfson, H.J., Rigoutsos, I., 1997. “*Geometric hashing: An overview*”. IEEE Computational Science and Engineering, pp. 10-21.

Computación Evolutiva y Creatividad

Fernando Jorge Penousal¹ y M^a Luisa Santos²

¹Instituto Superior de Ingeniería de Coimbra

3030 Coimbra, Portugal

Email: machado@dei.uc.pt

²Facultad de Informática. Universidade da Coruña

Email: mhyso@udc.es

Introducción

Entre las principales preguntas sobre el estudio de la Creatividad Computacional se incluyen las siguientes: “¿Pueden ser creativos los ordenadores?”, “¿Puede contribuir el estudio de la creatividad computacional a la comprensión de la creatividad humana?”, “¿Cómo pueden utilizarse los ordenadores para fomentar la creatividad humana?”, “¿Resulta la creatividad una condición necesaria para la inteligencia?”, “¿Qué es exactamente la creatividad, ya sea la computacional u otra cualquiera?”.

El desarrollo de sistemas creativos artificiales se inspira a menudo en modelos de creatividad humana. No obstante, la que nos interesa es una aproximación distinta.

A lo largo del tiempo, la evolución natural consiguió crear un sinnúmero de soluciones para un único problema: la supervivencia. La

amplia variedad de formas y estrategias que encontramos en la naturaleza nos permiten clasificar la evolución natural como un proceso altamente creativo. En realidad puede verse como la fuente de toda la creatividad. Por ello surge de modo natural la idea de utilizar la Computación Evolutiva (CE) para desarrollar sistemas computacionales creativos.

Por este motivo, en lugar de centrarnos en las preguntas ya mencionadas, el punto de arranque de este artículo es que la evolución es creativa y la pregunta a formular es: “¿Cómo puede utilizarse la Computación Evolutiva para desarrollar Sistemas Creativos?”

Este artículo puede considerarse una introducción al campo de la Creatividad Evolutiva. Como tal y tras una breve presentación de la CE (Sección *Computación evolutiva*), proponemos una taxonomía para la clasificación de los Sistemas Creativos y ofrecemos una vista general del estado actual del arte en el campo de la Creatividad Evolutiva (Sección *Computación evolutiva–Estado del arte*), centrándonos en las aplicaciones en las áreas de la música y de las artes visuales.

A continuación, y con el objetivo de facilitar la comprensión de los asuntos relativos al desarrollo de un sistema Creativo Evolutivo, realizamos una descripción detallada de NEvAr (Sección *NEvAr*), un proyecto de investigación destinado a la construcción de un artista artificial. Por último, la sección *Conclusiones* presenta algunas conclusiones globales.

Computación evolutiva

En los últimos años ha surgido un nuevo paradigma para la resolución de problemas denominado Computación Evolutiva (CE). La CE puede contemplarse como una serie de procedimientos de búsqueda estocástica inspirados en los principios biológicos de la selección natural y de la genética. Este conjunto de técnicas permite una exploración robusta del espacio de búsqueda que permite el descubrimiento de soluciones nuevas y adecuadas, habiéndose utilizado para resolver una amplia gama de problemas.

Históricamente la CE se ha dividido en cuatro familias¹, que serían Estrategias de Evolución (EE); Programación Evolutiva (PE); Algoritmos Genéticos (AG) y Programación Genética (PG). A pesar de sus diferencias, todos ellos pueden considerarse ejemplos particulares del Algoritmo Evolutivo Genérico (AEG) presentado en la figura 1.

```
t ← 0
G(0) ← generate random(t)
P(0) ← growth(G(0))
F(0) ← eval(P(0))
while stop criterion not met do
    G'(t) ← sel(G(t); P(t); F(t))
    G''(t) ← op(G'(t))
    G(t + 1) ← gen(G(t); G''(t))
    P(t + 1) ← growth(G(t + 1))
    F(t + 1) ← eval(P(t + 1))
```

¹Ver para obtener una idea general de la CE [1].

```
t ← t + 1
end while
return result
```

Figura 1: Algoritmo evolutivo genérico

El primer paso lo constituye la creación de un conjunto de genotipos al azar, $G(0)$. Estos genotipos se convierten en fenotipos (individuos plenamente desarrollados) por medio de la aplicación de una función de *crecimiento*. En la mayoría de los casos no existe una distinción clara entre genotipo y fenotipo, por ello suele omitirse este paso. El siguiente paso consiste en la evaluación de los individuos que se realiza a nivel del fenotipo utilizando una función fitness, *eval*.

A continuación sigue el principal ciclo evolutivo. Comenzamos seleccionando un conjunto de progenitores, utilizando *sel*. Para el éxito de la CE resulta crucial que aquellos individuos con valores fitness más altos tengan mayores probabilidades de ser seleccionados para reproducirse. Los individuos con valores fitness bajos deberían tener también alguna probabilidad de ser seleccionados. El siguiente paso consiste en la aplicación de operadores genéticos, *op*, tales como el cruce y la mutación. Dicha operación se realiza a nivel del genotipo y genera un nuevo conjunto de genotipos, $G''(t)$. La siguiente generación, $G(t+1)$, se crea por medio de la selección de elementos de la anterior, $G(t)$, y de $G''(t)$. Esta operación se realiza por medio de *gen*; que en su forma más sencilla consiste en el reemplazo de todos los individuos por sus descendientes, i.e.

$G(t+1) \leftarrow G''(t)$. El siguiente paso consiste en la generación de los fenotipos y en su evaluación, las cuales ya se han descrito.

El ciclo evolutivo continúa hasta que se cumple algún criterio de finalización. El resultado del algoritmo varía dependiendo del problema específico. Al menos suele interesarnos el mejor individuo obtenido durante la ejecución.

Creatividad evolutiva – Estado del arte

En esta sección se presenta una visión general del estado actual del arte en el campo de la creatividad evolutiva, enfatizando especialmente los campos de la música y de las artes visuales.

Comenzamos proponiendo una taxonomía para la clasificación de los sistemas creativos:

Herramientas de Apoyo a la Creatividad – Los sistemas ofrecen apoyo a la creatividad del usuario. El usuario sigue siendo el autor de la obra de arte y el responsable de la generación de la *idea*. Estos sistemas son esencialmente similares a un piano o a un lápiz. La diferencia estriba en que son más fáciles de usar y en que están diseñados para ayudar al usuario a concentrarse en los aspectos creativos de la tarea.

Creatividad Asistida por Ordenador – El usuario ya no es el responsable de generar la *idea*. En lugar de ello, la idea surge de la interacción entre usuario y ordenador y, como tal, existe una autoría

compartida. En el recordatorio de este capítulo se ofrecen diversos ejemplos de este tipo de sistema.

Autores – Puede considerarse al ordenador como el creador de la obra de arte, siendo el responsable de generar la *idea*. Sin embargo, el usuario puede ajustar o afectar al rendimiento del sistema. No obstante, el sistema debería mostrar un grado de autonomía comparable al de un creador humano.

Existe otro tipo de sistema potencialmente relevante al campo de los sistemas creativos:

Software para el Análisis de Tareas Creativas – Son sistemas que realizan algún tipo de análisis de una tarea creativa (e.g. análisis de música o imágenes). En cierto sentido, estos sistemas pueden verse como herramientas de apoyo creativo. Pueden utilizarse para fomentar la creatividad del usuario o como parte de otro sistema creativo.

En los últimos años se han utilizado dos técnicas de CE, los AG [10] y la PG [15], para implementar la creatividad computacional, dando como resultado el desarrollo de diversas aplicaciones en campos tales como la generación de música y de imágenes, la arquitectura y el diseño. A continuación ofrecemos una breve descripción de los trabajos más relevantes en este campo, centrándonos en la Creatividad Asistida por Ordenador y en los Sistemas Autores.

Creatividad asistida por ordenador

La Evolución Interactiva (EI) es la aproximación evolutiva más común a la Creatividad Asistida por Ordenador. En estos sistemas, la función de fitness la proporciona el usuario/ usuarios humanos que guían el proceso evolutivo. Esta aproximación se inspira en el trabajo original de Richard Dawkins (1986) [8] que utiliza la EI para desarrollar la morfología de unos organismos virtuales denominados *biomorphs*. Aparentemente su obra fue la fuente de inspiración para los sistemas desarrollados por Karl Sims (1991) [26] y William Latham (1992) [30], los cuales pueden considerarse las primeras aplicaciones de EI en el campo de las artes visuales, siendo asimismo consideradas como las obras más influyentes de este campo.

Estos son algunos trabajos adicionales en los que se utiliza la EI para desarrollar imágenes: Steven Rooke [24], Unemi [31], Lewis [16], Machado y Cardoso [18], y Lewis [17]. Asimismo se ha utilizado la EI para desarrollar películas [26], dibujos de líneas [2], figuras humanas [17], rostros de dibujos animados [16] y figuras [32], edificios [27], botellas de vodka [14], etc.

Estos son algunos ejemplos relevantes de EI en el terreno de la música: Horowitz [11], Ralley [23], Jacob [12][13], Biles [4], Pazos et al. [21][22], y Moroni et al. [20].

La principal desventaja de esta clase de sistema es el tiempo necesario para que el usuario evalúe las obras de arte, lo cual limita

considerablemente el número de generaciones. Esta desventaja resulta especialmente grave en el campo de la música. Además también genera dificultades la falta de consistencia de la evaluación humana.

Autores

Los defectos de la evaluación humana han llevado a los investigadores a desarrollar sistemas que tratan de aprender las preferencias del usuario a partir de un conjunto de ejemplos. La aproximación más común se basa en el uso de las Redes de Neuronas Artificiales (RNA). Si tiene éxito la inclusión de un evaluador autónomo, éste permite la creación de un sistema *Autor*. A continuación realizaremos una breve revisión de algunos intentos de automatizar el paso de asignación de fitness.

Burton y Vladimirova [7], y Burton [6] utiliza una RNA ARTMAP para clasificar los patrones rítmicos. La RNA es entrenada por medio de ejemplos de ritmos procedentes de cajas de ritmos. Esta RNA clasifica el conjunto de entrenamiento en grupos o “estilos”. El usuario puede añadir o eliminar grupos de manera dinámica. El output del sistema refleja la distancia del vector del prototipo con respecto al grupo más similar. Este trabajo ha dado lugar a resultados muy interesantes. Se utilizó este sistema conjuntamente con un algoritmo genético que crea nuevas piezas. Las piezas generadas son similares a aquellas utilizadas para entrenar a la RNA, lo cual demuestra el éxito de la tarea de aprendizaje. Por desgracia esta es también una de las desventajas del sistema: las piezas generadas son

demasiado similares a las del conjunto de aprendizaje, y por tanto ofrecen pocas novedades.

Spector y Alpern [28] presentan otro sistema interesante que combina una RNA con un sistema de PG. Este sistema utilizó una RNA feed-forward con una capa oculta para determinar “las continuaciones adecuadas para una serie de fragmentos de melodías de jazz” [28]. Se entrenó a la RNA con combinaciones de fragmentos de Charlie Parker y otros elegidos al azar, y ésta asignaba respuestas positivas a las buenas combinaciones y respuestas negativas a las combinaciones inversas o hechas al azar. Los resultados de la RNA son muy prometedores. Por ejemplo, le asignó una respuesta positiva a las piezas de Jimmy Hendrix. Sin embargo, cuando se utiliza conjuntamente con el motor de PG, el sistema obtiene resultados “no satisfactorios” Los autores lograron mejorar los resultados combinando con la RNA reglas simbólicas explícitas. No obstante, como afirman los propios autores, la solución a largo plazo consiste en mejorar la RNA.

Al Biles (1996) [5] también desarrolló un evaluador de RNA integrado en un compositor musical de computación evolutiva. La RNA posee una arquitectura de correlación en cascada (Casacor). Los conjuntos de entrenamiento y prueba se construyen usando un sistema de EI. Biles introdujo diversas representaciones del tema musical como inputs de la RNA, entre las que se incluyen: parámetros estadísticos (el número de eventos únicos de notas nuevas, el tamaño del intervalo máximo, etc.); un histograma de intervalos, así como las estructuras de arranque de las notas e intervalos. Los resultados de estos experimentos no resultaron totalmente

satisfactorios. Como afirma Biles: “cuando la red fue capaz de aprender el conjunto de entrenamiento, falló en el de prueba”.

Wiggins et al. (1999) [33] incorporan un evaluador musical basado en reglas a un compositor de computación evolutiva que genera “solos” de jazz. Las reglas del evaluador se basan en la bibliografía sobre improvisación, en el análisis estadístico informal de solos de jazz y en la heurística intuitiva derivada del autor. No se analizan en detalle los resultados del evaluador. No obstante, los autores señalan que los resultados del sistema evolutivo guiado por este evaluador son interesantes, aunque siguen estando lejos del ideal.

Todd y Werner (1998) [29] presentan una aproximación co-evolutiva. En su trabajo hay una población de creadores y una de evaluadores que evolucionan conjuntamente dando lugar a nuevas formas musicales. Este trabajo resulta muy interesante desde el punto de vista del estudio de la dinámica en una sociedad. Sin embargo, la estética engendrada es independiente de la estética humana. Como tal, resulta difícil evaluar los resultados.

Baluja et al. (1994) [3] propusieron un sistema de PG en el campo de las artes visuales en el que la evaluación la realiza la RNA. Igual que anteriormente, el conjunto de entrenamiento y el de prueba se construyen utilizando una versión EI del sistema. Baluja et al. probaron cinco arquitecturas diferentes de RNA. Los resultados experimentales son “un tanto decepcionantes” y “muy difíciles de cuantificar” [3]. Aunque las RNAs logran demostrar un cierto grado de aprendizaje, fallan cuando se las

utiliza para guiar la evolución. De acuerdo con los autores, el proceso evolutivo se volvió “muy limitado y en su mayor parte sin interés” al ser guiado por las RNAs.

Saunders y Gero hicieron uso de una red Kohonen auto-organizada para evaluar la novedad de una imagen dada [25]. Igual que antes, se utilizó un sistema de PG para generar imágenes y cada imagen se clasificó dentro de una categoría representada por una de las neuronas de la red. Al ir avanzando la evolución, la RNA aprende un mapa de las obras de arte típicas que está generando el sistema de PG. El valor novedoso de una imagen nueva es evaluado por medio del cálculo del error de categorización de la imagen (es decir, si la imagen no encaja bien en ninguna categoría, entonces posee un alto valor novedoso). Esta aproximación resulta muy interesante, ya que tiene en cuenta el aspecto novedoso de las imágenes, evitando así que se estanque el proceso evolutivo. La desventaja reside en que éste es el único aspecto de la imagen que se contempla.

Greenfield [9] propone una aproximación co-evolutiva para desarrollar imágenes. La idea consiste en desarrollar una población de portadores (imágenes) y una población de parásitos (filtros). Las imágenes son generadas por un sistema de PG. Los filtros son filtros estándar de circunvolución digital. Estos se aplican a pequeñas porciones de las imágenes y se compara el resultado a la imagen original. El fitness de la imagen es proporcional a la falta de similitud entre las imágenes originales y las circunvolucionadas. El fitness de los filtros depende de cómo se las arreglen para “pasar desapercibidos” Ello provoca una carrera de armas entre los portadores y los parásitos. De modo similar a la anterior

aproximación co-evolutiva, estos resultados experimentales son también difíciles de evaluar. Parece ser que el sistema tiende a producir imágenes de complejidad moderada que pueden resultar potencialmente interesantes. Sin embargo, tales imágenes tienden a la falta de estructura y a la abundancia de amplias zonas de ruido.

El análisis general de los sistemas descritos nos lleva a la conclusión de que, por el momento, aún no se ha logrado desarrollar un sistema *Autor* satisfactorio. Además, los trabajos del campo musical están considerablemente más desarrollados que los del campo visual. Resulta interesante percibir que la situación referente a los sistemas de Creatividad Asistida por Ordenador es la contraria: la aplicación a las artes visuales ha tenido un alto grado de éxito, existiendo diversos sistemas adecuados.

La siguiente sección contiene una descripción detallada de un sistema creativo evolutivo: NEvAr. El objetivo es facilitar una comprensión mejor de los asuntos relativos al desarrollo de esta clase de sistemas.

NEvAr

NEvAr (Arte Neuro-Evolutivo) es un proyecto de investigación actual. Su objetivo último es el de construir un Artista Artificial (una aplicación informática capaz de generar autónomamente obras de arte). En su estado actual de desarrollo, NEvAr es un sistema de Creatividad Asistida por Ordenador.

1. En su forma básica, NEvAr opera del modo siguiente:

2. El programa genera una población de imágenes al azar;
3. El usuario evalúa las imágenes asignándoles un “valor de fitness”;
4. El programa “reproduce” una nueva población de imágenes a través de la recombinación y mutación del código genético de las imágenes de la población actual. Aquellas imágenes con unos valores de fitness más altos tienen mayores probabilidades de ser seleccionadas para reproducirse.
5. Volver al punto 2.

Material genético

En NEvAr, las características de los individuos (imágenes) son determinadas por su código genético. Existe un fenotipo (el individuo) y un genotipo (el código genético) que una vez expresados, dan como resultado el individuo.

Los genotipos son árboles construidos a partir de un léxico de funciones y terminales. El conjunto de funciones se compone principalmente de funciones simples, tales como las operaciones aritméticas, trigonométricas y lógicas. El conjunto de terminales se compone de un conjunto de variables “ x ” e “ y ” y de unas constantes al azar. El fenotipo se genera por medio de la evaluación del genotipo para cada par (x,y) perteneciente a la imagen. Así, las imágenes generadas pueden contemplarse como retratos gráficos de expresiones matemáticas.

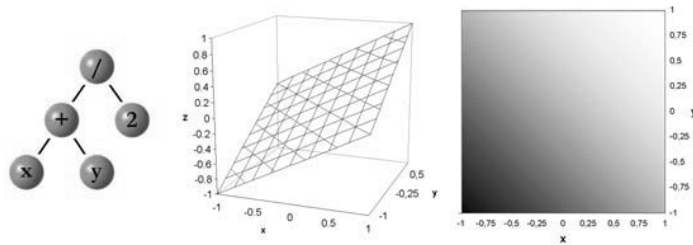


Figura 2: La expresión $f(x)=(x+y)/2$ representada en formato de árbol, un gráfico en 3d de la expresión y una imagen que ha sido generada asignando un valor en escala de grises a cada valor $f(x)$

Operadores genéticos

Las operaciones genéticas (recombinación y mutación) se realizan a nivel del genotipo. Para producir imágenes en color, utilizamos un tipo especial de terminal que genera un valor distinto al azar dependiendo del canal de color que se está procesando (rojo, verde o azul).

El operador de recombinación permite el intercambio de código entre individuos. La figura 3 presenta un ejemplo de dicha operación y muestra los cambios introducidos a nivel tanto del fenotipo como del genotipo.

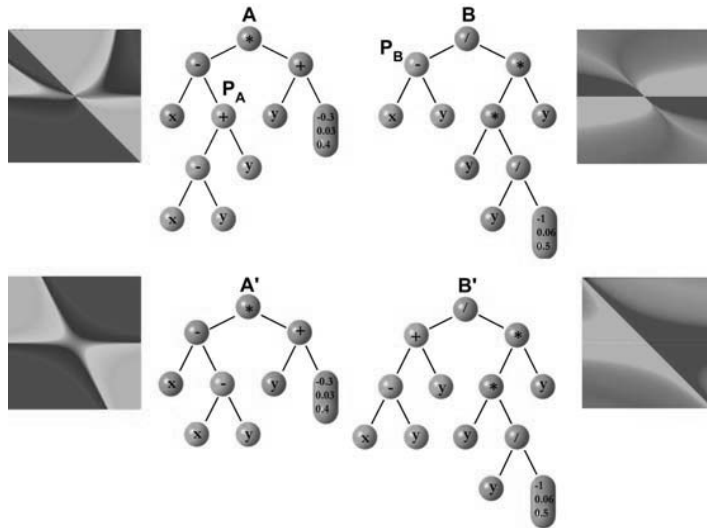


Figura 3: Ejemplo de la operación de recombinación. El código de los individuos A y B se recombina por medio del intercambio de los sub-árboles definidos explícitamente por dos puntos escogidos al azar PA y PB, dando lugar a los individuos A' y B'

Al contrario que en la recombinación, los operadores de mutación inducen cambios al azar en el código genético de un solo individuo. Utilizamos cinco operadores de mutación distintos: Reemplazar sub-árbol; Cambiar sub-árbol; Eliminar nodo; Insertar nodo y Cambiar nodo.

NEvAr como herramienta

El trabajo con NEvAr es un proceso iterativo. Al ir aumentando el número de la población, también tiende a aumentar la calidad promedio de

las imágenes, lo cual da lugar a imágenes nuevas, interesantes, potencialmente creativas y hermosas (al menos a ojos del usuario que conduce el programa). En la Figura 4 presentamos una serie de ejemplos generados mediante NEvAr.

Desde un punto de vista artístico, consideramos NEvAr una herramienta de gran potencialidad. Las ideas se generan a partir de un proceso evolutivo y de la interacción entre el artista y la herramienta. Así, el uso de NEvAr implica un cambio en el proceso artístico y creativo. No obstante, a pesar de estos cambios, las obras de arte obedecen a los principios artísticos y estéticos del artista.

Uno de los conceptos erróneos acerca de las herramientas de arte evolutivo reside en que las capacidades generativas de un sistema están profundamente conectadas con las primitivas utilizadas. Nuestra experiencia con NEvAr demuestra que esto es erróneo. Lo que se necesita es un conjunto de primitivas “básicas” que puedan combinarse de modo poderoso.

Resulta de gran importancia la inclusión de un mecanismo de memoria a largo plazo, ya que permite reutilizar las ideas previamente generadas. También constituye la base de la inclusión en NEvAr de mecanismos de Razonamiento Basado en Casos. Los experimentos preliminares indican que dichos mecanismos pueden generar resultados interesantes y seguir expandiendo las capacidades de nuestro sistema.

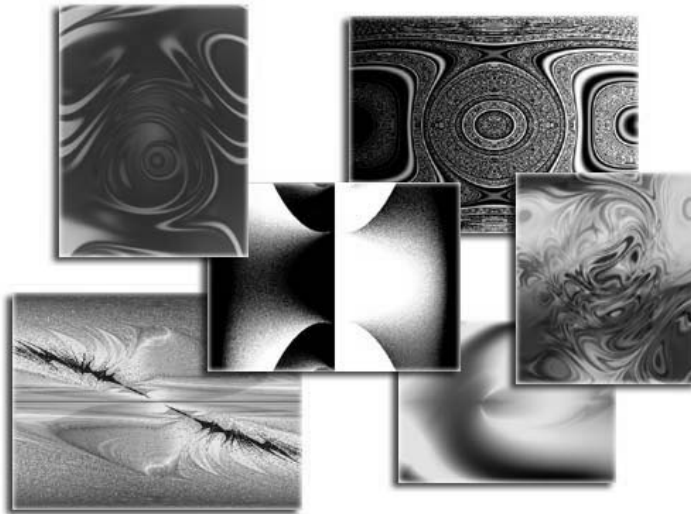


Figura 4: Algunos ejemplos de imágenes creadas con NEvAr

NEvAr como artista

Como ya se ha indicado, el objetivo último del presente proyecto es la construcción de un artista artificial.

Actualmente estamos estudiando la manera de otorgar autonomía al programa por medio de la automatización de la asignación del fitness. Nuestra idea inicial consistía en la de entrenar una red neuronal y utilizarla para automatizar esta tarea. Ahora creemos que la automatización plena no resulta alcanzable a corto plazo y nuestra idea actual reside en la utilización

de redes neuronales (junto con otras técnicas) como filtro para eliminar a los individuos indeseables.

En su forma actual, el procedimiento de asignación automática de fitness sólo toma en consideración la información de luz de las imágenes, descartando la información sobre tono y saturación. Por ello, en este modo de ejecución estamos limitados a la escala de grises.

La Figura 5 muestra imágenes creadas por NEvAr sin ninguna clase de intervención humana.



Figura 5: Ejemplos de imágenes desarrolladas por NEvAr sin intervención humana.

Coloreando las imágenes

La limitación de NEvAr a la escala de grises en su versión autónoma resulta, cuando menos, frustrante.

El análisis del papel del color y de la manera en que éste se asigna, sobre todo en el arte abstracto, lleva a la conclusión de que los artistas (ciertamente no todos pero sí una proporción significativa) suelen trabajar con una paleta de colores limitada y que la relación espacial entre colores

suele seguir ciertas pautas. Esto resulta coherente con la opinión de que cada artista construye su propio lenguaje artístico, el cual sigue una gramática implícita.

Surgió espontáneamente la idea de desarrollar un programa para colorear una serie de imágenes en escala de grises que fueron creadas por NEvAr. Pero desgraciadamente esto generó algunas dificultades [19]. Tomándolas en consideración, decidimos abandonar la idea y crear un sistema que aprenda a colorear imágenes a partir de un conjunto de ejemplos de entrenamiento.

Esta aproximación presenta varias ventajas potenciales frente a un procedimiento de coloreado incorporado, entre las que se encuentran las siguientes:

- No necesitamos codificar a mano un conjunto de reglas de coloreado.
- Los resultados del sistema son menos predecibles.
- Podemos utilizar como conjunto de entrenamiento las pinturas realizadas por artistas conocidos, aprendiendo así a colorear imágenes de acuerdo con su estilo.

Además, resulta un método indirecto de probar si los procedimientos de coloreado seguidos por algunos artistas pueden expresarse formalmente

En nuestra aproximación se utiliza la PG para desarrollar poblaciones de programas que dan color a las escalas de grises.

Comenzamos seleccionando una imagen de entrenamiento (o un conjunto de imágenes) de color auténtico, la cual se divide en sus canales de Luz, Tono y Saturación.

Los programas desarrollados toman la imagen de escala de grises que corresponde al canal de Luz como input, siendo el output una imagen en escala de grises. Se compara el output con el canal de Tono de la imagen de entrenamiento. Cuanto más cerca esté el output al deseado, más alto será el fitness. Puede aplicarse el mismo procedimiento utilizando el canal de Saturación para desarrollar programas que generen una información de saturación.

Al ir avanzando la evolución, la calidad de los individuos aumenta y, llegado cierto punto, hallamos programas que generan colores muy similares a los originales.

No obstante, es importante reconocer que esto no es suficiente: la idea consiste en utilizar estos programas para colorear diversas imágenes. El hecho de que un programa genere un output que encaje exactamente con el de entrenamiento no garantiza que vaya a producir un coloreado interesante de imágenes diferentes. Dicho de otro modo, los programas evolucionados tienen que poder generalizar adecuadamente. Con objeto de promover su capacidad de generalización, hemos seleccionado con cautela el conjunto de funciones y también pusimos cuidado al construir la función de fitness.

Los resultados logrados con este sistema son muy prometedores, a pesar de su carácter preliminar. Al aplicar los programas evolucionados a

las imágenes de entrenamiento es posible acercarse mucho al color original. Además, cuando se aplican a imágenes diferentes, son capaces de generar colores coherentes, atractivos y creíbles.

Conclusiones

El presente artículo ofrece una vista general del estado del arte actual en el campo de la Creatividad Evolutiva, centrándose en sus aplicaciones en las áreas de la música y de las artes visuales.

El análisis general del campo demuestra que la Computación Evolutiva resulta una aproximación viable al desarrollo de sistemas creativos. El desarrollo de sistemas de Creatividad Asistida por Ordenador por medio de la Computación Evolutiva ya es una realidad. A pesar de que es más limitado el éxito en la construcción de sistemas Autor, ya existen algunos trabajos prometedores.

Hemos presentado también el proyecto de investigación NevAr cuyo objetivo es la construcción de un artista artificial. El análisis detallado de NEvAr permite una mejor comprensión de los temas relativos a la construcción de un sistema creativo evolutivo.

A pesar de que nos hemos centrado en los campos artísticos, éstos no son los únicos campos de aplicación de la Creatividad Computacional. De hecho se está aplicando ya la Creatividad Computacional a áreas tales como: el Descubrimiento Científico, el Data Mining, la Recuperación de Información, etc.

Bibliografía

- [1] Bäck, T., Fogel, D.B., Michalewicz, Z., editors, 2000. “*Evolutionary Computation 1: Basic Algorithms and Operators*”. Institute of Physics Publishing and Oxford University Press.
- [2] Baker, E., 1993. “*Evolving line drawings*”. In Forrest, S., editor, Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA. Morgan Kaufman.
- [3] Baluja, S., Pomerleau, D., Jochem, T., 1994. “*Towards automated artificial evolution for computer-generated images*”. Connection Science, vol. 6 (2 and 3), pp. 325–354.
- [4] Biles, J.A., 2000. “*Genjam in perspective: A tentative taxonomy for genetic algorithm music and art systems*”. In Johnson, C.G. and Cardalda, J.J.R., editors, Genetic Algorithms in Visual Art and Music, pp. 133–135, Las Vegas, Nevada, USA.
- [5] Biles, J.A., Anderson, P.G., Loggi, L.W., 1996. “*Neural network fitness functions for a musical IGA*”. In Proceedings of the International ICSC Symposia on Intelligent Industrial Automation and Soft Computing, pp. B39–B44.
- [6] Burton, A.R., 1998. “*A hybrid neuro-genetic pattern evolution system applied to musical composition*”. PhD thesis, University of Surrey, School of Electronic Engineering.

- [7] Burton, A.R., Vladimirova, T., 1997. “*Genetic algorithm utilising neural network fitness evaluation for musical composition*”. In G.D. Smith, N.S. and Albrecht, R., editors, International Conference on Artificial Neural Networks and Genetic Algorithms, Springer, Norwich, UK.
- [8] Dawkins, R., 1986. “*The Blind Watchmaker*”. W.W. Norton, New York, ISBN: 0-393-31570-3.
- [9] Greenfield, G.R., 2002. “*On the co-evolution of evolving expressions*”. International Journal of Computational Intelligence and Applications, vol. 2 (1), pp. 17-31.
- [10] Holland, J.H., 1975. “*Adaptation in Natural and Artificial Systems*”. University of Michigan Press, Ann Arbor, MI.
- [11] Horowitz, D., 1994. “*Generating rhythms with genetic algorithms*”. In Proceedings of the 1994 International Computer Music Conference, pp. 142-143, Aarhus, Denmark.
- [12] Jacob, B., 1995. “*Composing with genetic algorithms*”. In Proceedings of the 1995 International Computer Music Conference, pp. 452-455, Banff (Alberta), Canada.
- [13] Jacob, B., 1996. “*Algorithmic composition as a model of creativity*”. Organized Sound, vol. 1, pp. 157-165.

- [14] Kelly, K., 1995. “*Out of Control: The New Biology of Machines, Social Systems and the Economic World*”. Perseus Publishing, London, ISBN: 0-201-48340-8.
- [15] Koza, J.R., 1990. “*Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*”. Technical Report STAN-CS-90-1314, Stanford University.
- [16] Lewis, M., 2000. “*Aesthetic evolutionary design with data flow networks*”. In Soddu, C., editor, *Generative Art*.
- [17] Lewis, M., 2000. “*Evolving human figure geometry*”. Technical Report OSU-ACCAD-5/00-TR1, Ohio State University.
- [18] Machado, P., Cardoso, A., 2000. “*NEvAr – the assessment of an evolutionary art tool*”. In Wiggins, G., editor, *Proceedings of the AISB00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, Birmingham, UK.
- [19] Machado, P., Dias, A., Cardoso, A., 2002. “*Learning to colour greyscale images*”. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, vol. 1 (2), pp. 209–219.

- [20] Moroni, A., Zuben, V.F., Manzoli, J., 2002. “*Arbitration: Human machine interaction in musical domains*”. Leonardo, vol. 35, pp. 185–189.
- [21] Pazos, A., del Riego, A.S., Dorado, J., Cardalda, J.J.R., 1999. “*Adaptive aspects of rhythmic composition: Genetic music*”. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1794, Morgan Kaufmann, Orlando, Florida, USA.
- [22] Pazos, A., del Riego, A.S., Dorado, J., Cardalda, J.J.R., 1999. Genetic music compositor. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A., editors, *Proceedings of the Congress on Evolutionary Computation*, vol. 2, pp. 885–890, IEEE Press, Mayflower Hotel, Washington D.C., USA.
- [23] Ralley, D., 1995. “*Genetic algorithms as a tool for melodic development*”. In Proceedings of the 1995 International Computer Music Conference, pp. 501–502, Banff (Alberta), Canada.
- [24] Rooke, S., 2002. “*Eons of genetically evolved algorithmic images*”. In David W. Corne, P. J. B., editor, Creative Evolutionary Systems. Morgan Kaufmann.
- [25] Saunders, R., Gero, J., 2001. “*Artificial creativity: A synthetic approach to the study of creative behaviour*”. In Gero, J., editor,

Proceedings of the Fifth Conference on Computational and Cognitive Models of Creative Design, Key Centre of Design Computing and Cognition.

- [26] Sims, K., 1991. “*Artificial evolution for computer graphics*”. ACM Computer Graphics, vol. 25 (4), pp. 319–328. SIGGRAPH’91 Proceedings, Las Vegas, Nevada, USA.
- [27] Soddu, C., 1999. “*Recognizability of the idea: the evolutionary process of argenia*”. In Patrizio, A., Wiggins, G. A., and Pain, H., editors, Proceedings of the AISB’99 Symposium on Creative Evolutionary Systems, pp. 18–27.
- [28] Spector, L., Alpern, A., 1994. “*Criticism, culture, and the automatic generation of artworks*”. In Proceedings of Twelfth National Conference on Artificial Intelligence, pp. 3–8, AAAI Press/MIT Press, Seattle, Washington, USA.
- [29] Todd, P.M., Werner, G.M., 1998. “*Frankensteinian methods for evolutionary music composition*”. In Musical Networks: Parallel Distributed Perception and Performance. MIT Press, Cambridge, MA.
- [30] Todd, S., Latham, W., 1992. “*Evolutionary Art and Computers*”. Academic Press, London, ISBN: 0–124–37185–X.

- [31] Unemi, T., 2000. “*SBART 2.4: An IEC tool for creating 2D images, movies, and collage*”. In Johnson, C. G. and Cardalda, J. J. R., editors, *Genetic Algorithms in Visual Art and Music*, pp. 153–157, Las Vegas, Nevada, USA.
- [32] Ventrella, J., 1995. “*Disney meets Darwin: evolution of funny animated figures*”. In *Computer Animation '95 Proceedings*, IEEE.
- [33] Wiggins, G.A., Papadopoulos, G., Phon-Amnuaisuk, S., Tuson, A., 1999. “*Evolutionary methods for musical composition*”. *International Journal of Computing Anticipatory Systems*, vol. 1(1).



Creatividad Computacional

Amílcar Cardoso¹ y Juan Jesús Romero²

¹ Creative Systems Group, AILab

Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)

Pólo II, Pinhal de Marrocos, 3030 Coimbra, Portugal

Email: amilcar@dei.uc.pt

² Facultad de Informática. Universidade da Coruña

Email: jj@udc.es

Resumen

¿Tiene sentido creer que un enfoque computacional de la creatividad nos ayudará a entenderla? ¿Merece la pena investigar herramientas y entornos computacionales que puedan contribuir a la creatividad humana? ¿Es factible construir programas que puedan considerarse creativos?

Para aquellos que no acepten la idea romántica de la creatividad como inspiración divina, que no ubican su origen en una intuición innata, es decir, para aquellos que no se adhieren a lo que Margaret Boden (1990) [1] llamó *las vistas románticas e inspirativas de la creatividad*¹, el concepto de explorar enfoques computacionales a la creatividad es todo menos una pérdida de tiempo y recursos: la creatividad es conjuntamente aceptada como una de las más notables características de la mente humana. Debe por

¹ *inspirational and romantic views of creativity*

lo tanto ser investigada seriamente por los expertos en Inteligencia en general e Inteligencia Artificial en particular.

Es difícil medir, observar e interpretar la creatividad. Desde hace muchos siglos, pero sobre todo desde finales del siglo XIX, es un desafío para muchos científicos e investigadores que trabajan en áreas como la Ciencia Cognitiva y la Psicología. Cada vez más, los investigadores en IA trabajan en teorías explicativas y modelos computacionales adecuados, porque creen que los sistemas computacionales creativos son potencialmente eficaces en un amplio abanico de campos artísticos, técnicos y científicos, donde la innovación es un elemento clave. El descubrimiento científico, la prueba de teoremas y el diseño técnico son sólo algunos terrenos de aplicación, y la naturaleza multidisciplinar de la investigación resulta ser un factor muy atractivo. Finalmente, se trata de una investigación que puede contribuir a entender los mecanismos que rigen la creatividad.

Dimensiones de Creatividad

La creatividad es un campo de investigación muy complejo. Intervienen gran cantidad de factores que complican su definición, valoración, modelización y mejora, por lo cual su estudio debe considerar varias dimensiones.

Podemos por ejemplo adoptar cuatro perspectivas de análisis diferentes: podemos considerar el proceso creativo, el ser que crea, el entorno de la creación, y el producto final. En la modelización de la creatividad, el proceso creativo constituye el enfoque más relevante, pero

también resulta primordial la perspectiva del producto creativo, ya que solemos reconocer la creatividad por las características del producto. Este punto será tratado más adelante.

En su interpretación del fenómeno de la creatividad, Boden (1990) [1] propone la distinción entre la perspectiva del creador, llamada *Creatividad Psicológica*², cuando el hecho es “nuevo” para el individuo, y la perspectiva de la sociedad o *Creatividad Histórica*³, cuando el hecho es “nuevo” para la sociedad, (abreviados: Creatividad P y H). Esta distinción significa que alguien puede considerar algo como creativo, p.ej. el propio creador, sin que la sociedad en general lo perciba así.

Boden también proponía una distinción ortogonal entre la creatividad exploratoria y la creatividad transformacional (abreviados: Creatividad E y T). La primera se define esencialmente como un proceso de exploración del espacio conceptual, con el propósito de localizar e identificar nuevos objetos conceptuales, mientras que la segunda se caracteriza por la transformación del propio espacio conceptual. Ejemplos de la creatividad E serían productos de arte o música en estilos conocidos y establecidos; cambios de paradigmas, como trabajos que lleven a estilos nuevos, corresponderían a la creatividad T. Esta distinción será analizada más adelante, ahora nos centraremos en la distinción entre el proceso creativo y

² *Psychological Creativity*

³ *Historical Creativity*

el producto creativo, con el fin de introducir de manera básica los modelos de creatividad y su aplicabilidad.

El Proceso Creativo

Varios modelos explicativos que describen el proceso creativo humano tienen su origen en los campos de la Psicología y la Ciencia Cognitiva, como los modelos sugeridos por Dewey (1919) [7], Poincaré (1913) [18], Wallas (1926) [22], Rossman (1931) [21], y Guilford (1968) [12] (ver el informe de Brown (1989) [2]). Los modelos que adoptan un enfoque de procesamiento de la información (ej., Wallas (1926) [22]) son particularmente útiles para la creatividad computacional. En términos generales, estos modelos describen el proceso como un procedimiento gradual que puede ser representado de la siguiente manera: (i) adquisición del problema y asimilación del conocimiento, (ii) búsqueda consciente o inconsciente de una solución, (iii) propuesta de una solución, y (iv) verificación de la solución propuesta. Estos modelos pueden constituir una importante fuente de inspiración para los modelos computacionales de creatividad.

En su teoría “Estructura del Intelecto⁴”, Guilford (1967) [11] propone el concepto de producción divergente como la habilidad de “generar variedad y cantidad de información; muy involucrado en el potencial creativo” (Brown 1989 [2]). Según Michael (1977) [15], esto

⁴ Structure of Intellect

puede conseguirse recurriendo al proceso de “reutilización de la transformación”, en el cual la misma información se aplica a distintos contextos (como si se intentara introducir una vieja pieza de puzzle dentro de un puzzle nuevo).

También resultan relevantes los estudios de De Bono (1986) [6] sobre el concepto de los pensamientos lateral y vertical, y su relación con la creatividad. Y podemos buscar otras fuentes de inspiración que no sean la creatividad humana. En la naturaleza, observamos que todas las especies vivas luchan permanentemente por la supervivencia. La teoría Neo-Darwinista, que revisa las primeras ideas de Darwin con la genética moderna, nos da un marco científico que explica cómo las especies sobreviven gracias a su capacidad de adaptación a los cambios medioambientales. El centro de este proceso es un mecanismo que selecciona los individuos más fuertes y recombina su material genético, combinando las “buenas” partes de entidades diferentes para crear una entidad todavía mejor. Esto es un claro ejemplo de soluciones innovadoras (Goldberg, 1998 [10]).

El Producto Creativo

Es primordial analizar la creatividad también bajo el punto de vista de su resultado, el producto creativo: una sinfonía, un invento, la prueba de un teorema, son generalmente consideradas como productos creativos, no como procesos específicos que llevaron a una producción. Aunque no queramos entender la creatividad únicamente a través del producto, es

difícil imaginar un concepto de creatividad que no considere el producto como uno de sus componentes principales.

Dos propiedades principales caracterizan un producto creativo. Una es la *novedad*: para que un producto sea reconocido como creativo, debe tener un alto nivel de originalidad. Sin embargo, la novedad sola no es suficiente: la noción de *valor*, o adecuación, es a menudo asociada a la creatividad. Una secuencia aleatoria de símbolos puede ser nueva, pero no será considerada como creativa sin un cierto significado en un lenguaje compartido y aceptado.

¿Qué caracteriza la creatividad?

Recientemente, G. Wiggins (2001, 2003) [23], [24] ha propuesto una caracterización formal que justifica la distinción de Boden entre la *creatividad E*, la más común, definida como una búsqueda/exploración en un espacio conceptual, y la *creatividad T*, que tiene lugar cuando el espacio conceptual es transformado. De este modo, el proceso creativo puede ser modelado como la alternancia entre la exploración y la transformación, durante la cual varios espacios conceptuales son buscados en varias fases exploratorias del proceso, siendo cada espacio conceptual el resultado de una fase transformacional.

Para justificar la existencia de múltiples espacios conceptuales, a “Universo de Posibilidades⁵”, \mathcal{U} , es definido como un superconjunto del espacio conceptual en cualquier punto del proceso creativo: “ \mathcal{U} es un espacio multidimensional, cuyas dimensiones son capaces de representar cualquier cosa, y todos los posibles conceptos distintos corresponden con distintos puntos en \mathcal{U} ” (Wiggins 2001 [23]).

Los espacios conceptuales se definen de modo generativo: cada espacio tiene un conjunto de reglas \mathcal{R} que determina su calidad de miembro del espacio, es decir reglas que limitan el espacio conceptual. Estas reglas son supuestamente aceptadas por la mayoría de los investigadores que trabajan en el campo en cuestión. Un conjunto distinto de reglas \mathcal{T} permite la travesía del espacio, cada conjunto definiendo una estrategia de búsqueda particular. En principio, cada creador particular desarrollará su propio conjunto de reglas, es decir su propia "técnica". Los conceptos logrados son evaluados mediante otro conjunto de reglas \mathcal{E} , que dista de los otros dos conjuntos en que intenta reflejar la manera en que la audiencia evalúa los productos creativos, y que cambia en el curso del proceso.

Podemos asumir, sin ser demasiado específicos, que todas las reglas están escritas en el mismo alfabeto \mathcal{L} . Por lo tanto, cada regla es un subconjunto del idioma \mathcal{L}^* .

⁵ Universe of Possibilities

A continuación describiremos de manera más formal ambos tipos de creatividad, lo cual nos permitirá distinguir mejor entre ellos.

Creatividad E

La creatividad exploratoria puede ser representada de manera abstracta por un septillo:

$$\langle \mathcal{U}, \mathcal{L}, [[\cdot]], \langle\langle\cdot\rangle\rangle, \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle$$

El significado de los símbolos es el siguiente (ver Figura 1):

- \mathcal{U} un universo de posibles conceptos
- \mathcal{L} un alfabeto desde el cual se construyen las reglas
- $[[\cdot]]$ un generador de funciones que traza un subconjunto de \mathcal{L}^* a una función que selecciona elementos de \mathcal{U}
- $\langle\langle\cdot\rangle\rangle$ un generador de funciones que traza un subconjunto de \mathcal{L}^* a una función que selecciona y *enumera* elementos de \mathcal{U}
- \mathcal{R} un subconjunto de \mathcal{L}^* , el conjunto de reglas de pertenencia al espacio conceptual
- \mathcal{T} un subconjunto de \mathcal{L}^* , el conjunto de reglas de la estrategia de búsqueda
- \mathcal{E} un subconjunto de \mathcal{L}^* , el conjunto de reglas de evaluación

La creatividad \mathcal{E} puede describirse como una búsqueda en un espacio conceptual \mathcal{C} limitado por las reglas \mathcal{R} usando una estrategia de búsqueda codificada por las reglas \mathcal{T} .

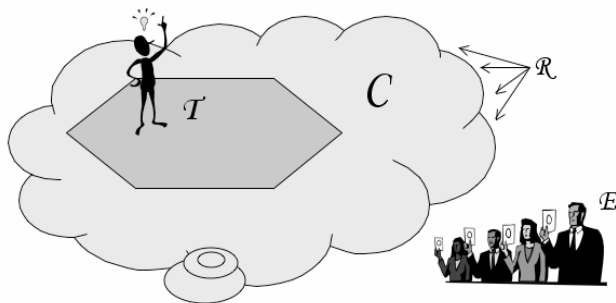


Figura 1: Creatividad Exploratoria

Con este conjunto de símbolos podemos definir formalmente el espacio conceptual como un espacio compuesto por los conceptos de \mathcal{U} seleccionados por la función $[[\mathcal{R}]]$:

$$C = [[\mathcal{R}]](\mathcal{U})$$

La exploración de \mathcal{C} puede definirse como un proceso iterativo que en cada paso produce nuevos elementos de \mathcal{C} a partir de otros elementos visitados previamente. La exploración se hace en un orden particular, controlado por \mathcal{T} y limitado por \mathcal{R} , y puede describirse a través de una

búsqueda basada en una agenda donde una función $\langle\langle \mathcal{R} \cup \mathcal{T} \rangle\rangle$ traza repetidamente conceptos sucesivos de \mathcal{U} :

$$c_{\text{out}} = \langle\langle \mathcal{R} \cup \mathcal{T} \rangle\rangle(c_{\text{in}})$$

En cualquier momento podemos partir C en C_1 , el espacio de conceptos que ya han sido descubiertos, y C_2 , el espacio de los conceptos remanentes. Típicamente, C tiene muchas circunvoluciones y algunos de sus puntos son difíciles de alcanzar por las estrategias de búsqueda regulares; algunos elementos en C pueden incluso ser inaccesibles bajo \mathcal{T} . Por lo tanto, una \mathcal{T} particular tendrá más éxito en atravesar C y encontrar miembros de C_1 , es decir, en producir soluciones nuevas, si puede alcanzar tales puntos (ver Figura 2).

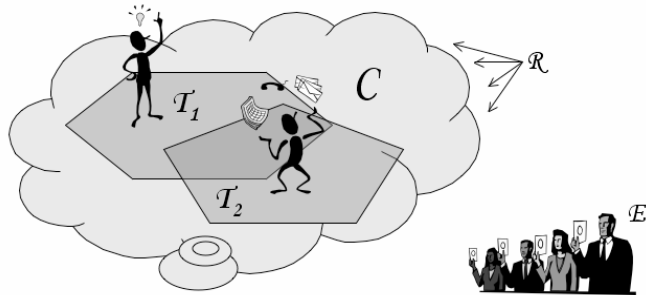


Figura 2: Dos creadores, mismo tipo, estilos diferentes

Como explicaba Wiggins, esto nos da la posibilidad de simular dos compositores que trabajan de manera distinta (distintas estrategias de búsqueda) dentro del mismo estilo (espacio limitado por las mismas reglas).

Creatividad T

Mientras que la creatividad E se describe como una exploración del espacio conceptual, la creatividad T se refiere, según Boden, a la transformación del propio espacio conceptual: los grandes momentos creativos, tales como los cambios paradigmáticos, entran en esta categoría. Bajo la formalización de Wiggins, transformar C implica cambiar las reglas \mathcal{R} de modo que los puntos fuera de C sean alcanzables (ver Figura 3). Esto se puede conseguir cambiando el conjunto de reglas \mathcal{R} en \mathcal{R}' , lo cual resulta en un nuevo espacio conceptual C' .

$$C' = [[\mathcal{R}']](\mathcal{U})$$

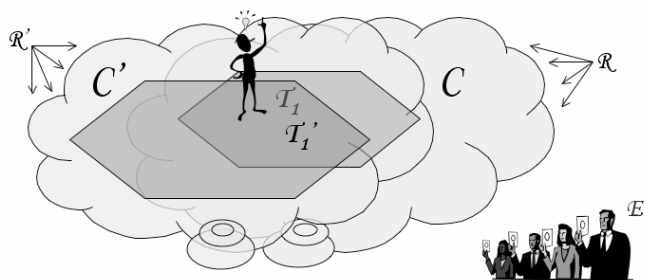


Figura 3: Transformando \mathcal{R}

Si \mathcal{R}' es suficientemente valorado por el actual conjunto de evaluación \mathcal{E} , puede ser aceptado por un número significativo de personas que trabajan en el campo y acabar como cambio paradigmático.

Wiggins argumenta que un segundo tipo de creatividad T tiene lugar cuando hay un cambio en el conjunto de reglas \mathcal{T} hacia un nuevo \mathcal{T}' (ver Figura 4). Esto corresponde a un cambio en la estrategia de búsqueda de un creador particular, resulta en una manera distinta de atravesar C , y permite al creador conseguir los mismos puntos de manera más eficaz y alcanzar algunos puntos previamente inalcanzables. Sin embargo, \mathcal{R} sigue igual, la naturaleza del espacio de búsqueda sigue igual, no existe transformación de C , y por lo tanto, no tiene lugar un cambio paradigmático. La transformación ocurre en la superficie de búsqueda del creador individual.

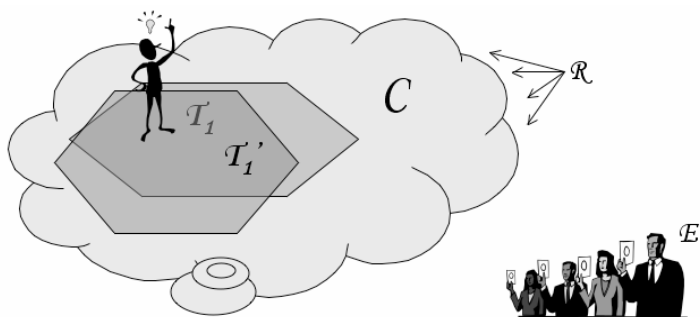


Figura 4: Transformando \mathcal{T} : el creador adopta una estrategia nueva

Esto no significa que el hecho de cambiar \mathcal{R} sea más significativo para la creatividad H que el de cambiar \mathcal{T} . La historia del arte, de la música y de la ciencia nos ha demostrado ampliamente que grandes descubrimientos pueden ser alcanzados sin cambio paradigmático, solo a través del refinamiento de una técnica hasta el punto de excelencia.

Terminamos este capítulo con la interesante conclusión que se puede sacar de la caracterización mencionada: la creatividad T puede describirse como un *proceso de exploración* en el meta-espacio de los conjuntos de reglas \mathcal{R} y \mathcal{T} , creando interesantes preguntas sobre el significado de la distinción entre la creatividad T y E - una discusión que queda fuera del alcance de este texto.

Modelos de creatividad por ordenador

Los psicólogos cognitivos han observado que el potencial creativo humano está intensamente relacionado con la habilidad de adoptar distintos “puntos de vista” de la realidad. Por ejemplo, según Marin y de la Torre (91) [14], nuestras capacidades de abstracción, de análisis simbólico, y de encontrar relaciones que no son obvias, entre otras cosas, son asociadas a la producción creativa. Una de estas teorías cognitivas (Guilford, 1967 [11]) profundiza la idea de “producción divergente”.

En términos computacionales, explorar espacios con circunvoluciones para llegar a puntos difíciles requiere unos mecanismos de búsqueda flexibles y que ofrezcan la posibilidad de buscar “en paralelo” en

áreas dispares del espacio de búsqueda, encontrando máxima locales sin dejar de divergir.

A continuación ilustraremos la caracterización abstracta con una breve introducción de una arquitectura computacional similar a la que proponen Cardoso et al. (2000) [3], y que puede explicar la flexibilidad y la fluidez requeridas por los Sistemas Creativos.

Una simple arquitectura de exploración

La Figura 5 representa una simple arquitectura que modela el proceso creativo como una secuencia iterativa de pasos; en esto se parece mucho a los ya mencionados modelos de procesamiento de información que surgieron de la Psicología y la Ciencia Cognitiva. El objetivo principal es la propuesta de nuevas *ideas*, que son transformaciones de estructuras de datos contenidas en una Memoria de Conceptos (MC)⁶. La cualidad de la idea nueva depende de su novedad y de su capacidad de resolver un problema específico. Dos módulos clave desempeñan un papel central en el proceso: el Generador y el Manipulador. El primero produce ideas (estructuras de datos) a través de la exploración de un espacio conceptual, mientras que el segundo manipula las ideas seleccionadas y le propone ideas nuevas al usuario (humano). Para poder nutrir de manera conveniente al Manipulador, el primer módulo ha de ser fluido (es decir, debe ser capaz de producir una amplia variedad de ideas), incluso si esto implica tomar riesgos cognitivos.

⁶ Concept Memory (CM)

El segundo módulo, que debe contribuir al control de la cualidad global de las ideas propuestas, puede adoptar dos estrategias: puede intentar incrementar la novedad general de las ideas propuestas (p.ej., recombiniéndolas), o puede incrementar su valor general (p.ej, adaptándolas al problema).

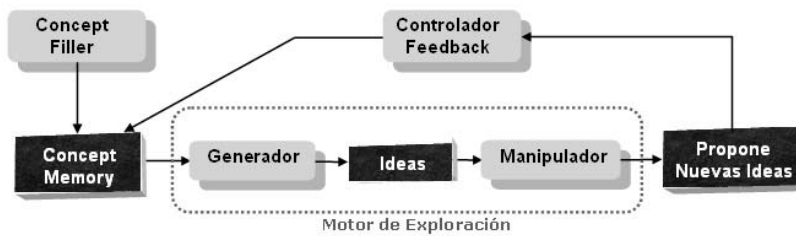


Figura 5: Una arquitectura de exploración simple

Inicialmente, el MC pasa por un Rellenador de Conceptos⁷, que puede ser controlado por un usuario humano o actuar autónomamente. El Rellenador de Conceptos puede llenar el MC usando conocimiento del terreno para crear las estructuras necesarias, o puede crearlas aleatoriamente.

El MC puede evolucionar durante el proceso gracias al Controlador de Feedback, que nutre al MC con nuevas ideas propuestas por el

⁷ Concept Filler

Manipulador. Igual que el Rellenador de Conceptos, el Controlador de Feedback puede actuar autónómicamente.

Instanciación usando RBC y CE

Podemos establecer un paralelismo entre esta arquitectura y la manera en que la Computación Evolutiva (CE) (Fogel 1999 [8]) y el Razonamiento Basado en Casos (RBC) (Leake 1996 [13]) resuelven problemas (ver Tabla 1). Esto nos puede dar una idea más concreta sobre cómo esta arquitectura puede ser instanciada usando cualquiera de estas técnicas.

Arquitectura propuesta	Computación Evolutiva	Raz. Basado en Casos
Memoria de Conceptos	Población	Caso Base
Generador	Evaluación/Selección (c./función fitness)	Selección (c./ métrica)
Manipulador	Operadores Genéticos (cruce, mutación, ...)	Operadores de Adaptación

Tabla 1: Similitud entre la arquitectura híbrida, CE y CBR

En CE, una población (MC) es evaluada y unos individuos (ideas) son seleccionados según una función de fitness, lo cual puede ser considerado un proceso de *generación* en la mencionada arquitectura. A continuación, los operadores genéticos desempeñan el papel de Manipulador y las ideas propuestas son devueltas al MC. En RBC, el proceso de *generación* puede llevarse a cabo seleccionando casos de una

base de casos (MC) en función de una métrica. Los operadores de adaptación *manipulan* los casos seleccionados, y los casos propuestos pueden ser devueltos al MC.

La arquitectura es muy flexible y puede ser explorada de muchas maneras. Si adoptamos una representación que es apta para conceptos, esos conceptos pueden ser reproducidos, utilizados y manipulados por los mecanismos RBC y CE. Podemos usar CBR en el Generador y aplicar operadores genéticos en el Manipulador para mejorar la diversidad. Podemos también usar la función de fitness para seleccionar ideas y operadores de adaptación para así ganar aptitud. Podemos incluso cambiar los mecanismos por el Generador y el Manipulador en cada ciclo; la elección de cada combinación puede depender de la evaluación de los resultados intermedios y/o de los propósitos específicos del usuario.

Representación

Según Guilford (1967) [11], la creatividad está directamente relacionada con nuestra capacidad de conectar y relacionar ideas aparentemente divergentes, es decir, de ver similitudes donde, por defecto, parece no haber ninguna (ver el famoso ejemplo de la serpiente que se muerde y de la molécula de benceno en el sueño de Kekulé). En términos computacionales, este efecto sólo puede obtenerse divagando a través de conceptos no relacionados, es decir potenciando las asociaciones entre conceptos de varias áreas, lo cual exige un alto nivel de flexibilidad en la representación. El programa que es capaz de hacer esto debe aceptar una

representación abierta (en el sentido de que debe aceptar varios campos) y debe disponer de mecanismos para operar en los propios conceptos sin perder valor semántico.

Una representación aceptable puede permitirnos coger lo mejor de la flexibilidad de la arquitectura presentada. Si queremos por ejemplo explorar la posibilidad de usar, de modo flexible, técnicas de CE y RBC en los módulos de Generador y Manipulador, las estructuras del concepto deben ser diseñadas de tal modo que ambos mecanismos pueden procesarlas libremente, p.ej. ensamblando de nuevo los mismos fragmentos en combinaciones nuevas.

Cardoso et al. (2000) [3] proponían un lenguaje de representación común: un conjunto de objetos estructurados jerárquicamente (OEHs) que son fácilmente manipulados por mecanismos CE y RBC. Este lenguaje puede definirse más exactamente por una gramática que enseñamos parcialmente usando el tradicional formalismo BNF:

$$\langle \text{structure} \rangle ::= \langle \text{node} \rangle \mid \langle \text{node} \rangle (\{ \langle \text{structure} \rangle \}^+)$$
$$\langle \text{node} \rangle ::= \langle \text{name} \rangle \mid \langle \text{name} \rangle (\{ \langle \text{attribute} \rangle \}^+)$$
$$\langle \text{attribute} \rangle ::= \langle \text{type-attr} \rangle (\langle \text{name} \rangle)$$
$$\langle \text{type-attr} \rangle ::= \text{link-in}(\langle \text{value} \rangle) \mid \text{link-out}(\langle \text{value} \rangle)$$

El mismo lenguaje común puede usarse para representar objetos de dominios distintos. Los objetos representados en las Figuras 6, 8, 9 y 10 son instancias de los OEHS, y sólo los no terminales <name> y <value> dependen del dominio. La principal ventaja de adoptar tal representación es que permite integrar plenamente los mecanismos CE y RBC en la misma arquitectura, así como incluir objetos de distintos dominios en la misma base de conocimiento. Gracias a esta última característica, la implementación de *mecanismos de transferencia metafórica* es posible.

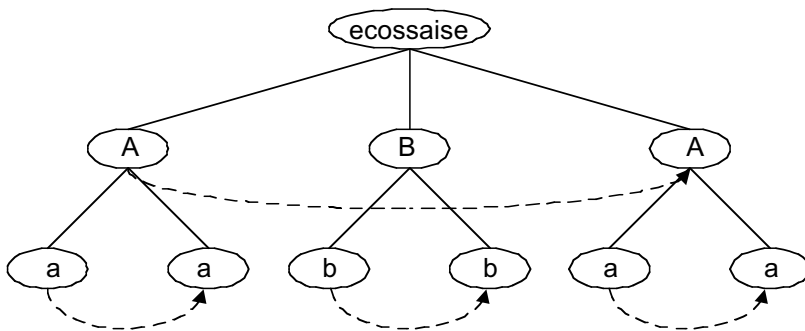


Figura 6: Representación del nivel abstracto de una pieza musical de Mauro Giuliani

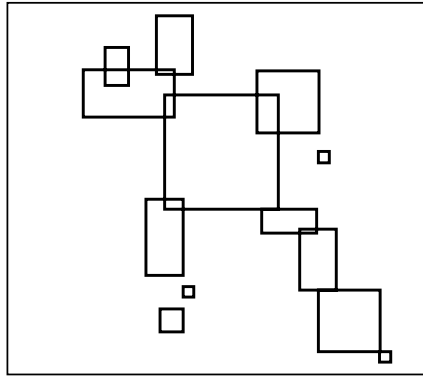


Figura 7: Un sketch basado en "Thirteen Rectangles" de Kandinsky

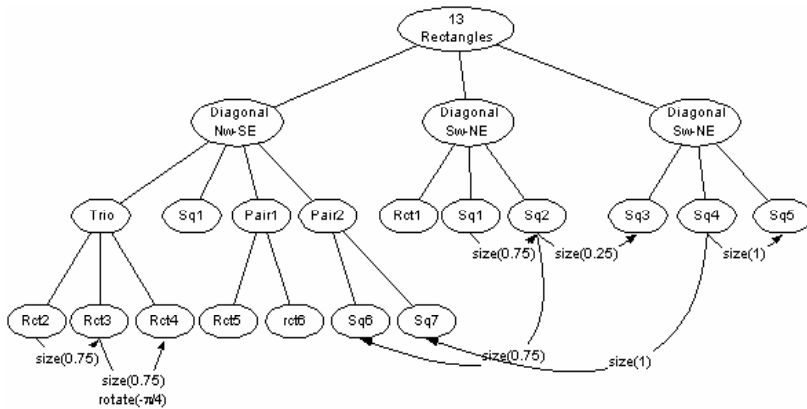


Figura 8 – Una representación del dibujo de la Figura 3

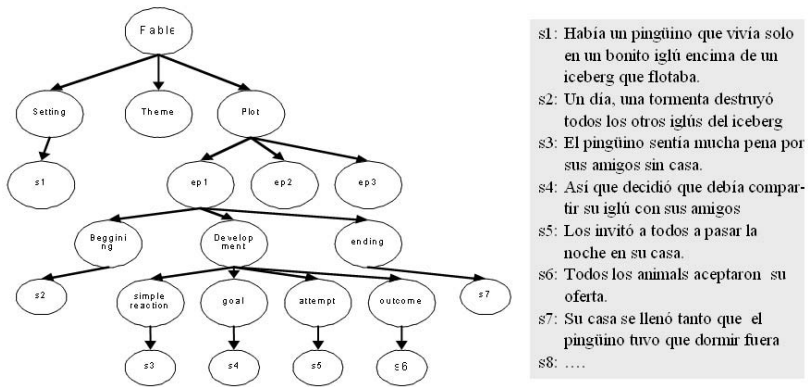


Figura 9: Una representación de la estructura de un cuento

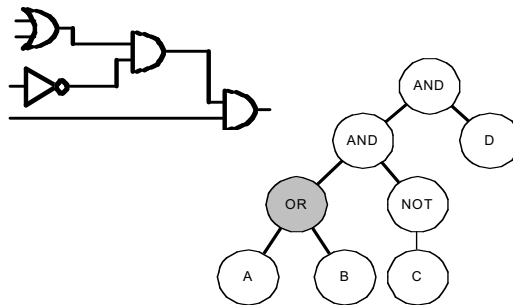


Figura 10: La estructura de un circuito digital

Caracterizando la arquitectura

Aunque la arquitectura de la Figura 5 es muy sencilla, ilustra una posible implementación de creatividad exploratoria y podemos interpretarlo bajo la formalización de Wiggins pintando algunas analogías.

La arquitectura puede verse como la exploración de un espacio conceptual C , con un subconjunto representado en cada ciclo por la Memoria de Conceptos. Otras partes de C son dinámicamente incluidas en la MC por el Controlador Feedback, y podemos admitir que el Rellenador de Conceptos puede también añadir otras partes en cualquier momento del proceso creativo. Otras partes de C son representadas solo implícitamente: el sistema visita puntos que pueden ser sintetizados en tiempo real por el Generador y el Manipulador.

La estrategia de búsqueda \mathcal{T} es típicamente codificada en el Generador y el Manipulador, ya que estos modelos son los responsables de decidir qué puntos hay que visitar y en qué orden. El Rellenador de Conceptos puede también tener un papel en esto, particularmente si le permitimos intervenir en el proceso iterativo añadiendo conceptos nuevos. Esta intervención puede ser particularmente eficaz en versiones más elaboradas de la arquitectura, donde se le da a este modulo la capacidad de analizar los contenidos de la MC.

El conjunto de evaluación \mathcal{E} es codificado en el Controlador Feedback. Debemos recordar que este módulo podría funcionar

autonómicamente, o bajo el mando de algún agente externo, posiblemente humano, que trabaja como un crítico y por lo tanto controla indirectamente el proceso. Podemos asimismo admitir que el Manipulador contenga algunas formas restringidas de asistencia al conocimiento, filtrando ideas que considera sin suficiente valor o novedad.

El conjunto \mathcal{R} que define la pertenencia al Espacio Concepto, sólo está implícitamente en esta arquitectura. La representación que adoptamos para los conceptos contiene una cantidad substancial de limitaciones definidas implícitamente por C . Por ejemplo, la representación puede no permitir colores en dibujos, o restringir los elementos de los dibujos a rectángulos. Este carácter no explícito de \mathcal{R} , que es probablemente una característica de cualquier mecanismo convencional para la solución de problemas, inhibe cualquier intento substancial de usar esta arquitectura para ilustrar la creatividad transformacional.

Podemos, sin embargo, considerar algunos posibles refinamientos: consideramos que parte de \mathcal{R} puede ser codificado explícitamente en los módulos, particularmente en el Manipulador, que puede filtrar ideas “no válidas”. Podemos argumentar que el cambio de estos mecanismos de filtración puede resultar en dar acceso a conceptos fuera del C original. Podemos admitir que el Generador y el Manipulador evolucionen durante el proceso creativo, dando lugar a cambios en T , que es una forma de creatividad T en los términos de Wiggins. Y podemos admitir cambios profundos, inducidos en la MC por el Rellenador de Conceptos, que correspondan a un cambio en \mathcal{R} . Estos son sólo algunos pensamientos sobre

un tema muy complejo: la *creatividad T* subsume la habilidad de razonar al nivel meta, para así posibilitar la creación de conceptos previamente imposibles. Esta investigación puede parecer la búsqueda del Santo Grial, pero nos lleva hacia temas que pueden avanzarnos en procesos de transferencia entre dominios como Analogía, Metáfora y Mezcla Conceptual (Pereira y Cardoso 2003) [17].

Una nota sobre la Evaluación

Los límites de este documento no nos permiten incluir una sección sobre la Evaluación de Creatividad. Dada la importancia del tema, invitamos al lector interesado a consultar algunas obras publicadas recientemente que tratan el tema: (Ritchie 2001 [19], Colton et al. 2001 [5], Pease et al. 2001 [16], Gervás 2002 [9], Romero et al. 2003 [20]).

Bibliografía

- [1] Boden, M., 1990. “*The Creative Mind: Myths and Mechanisms*”. Abacus, London, ISBN: 0-415-31452-6.
- [2] Brown, R.T., 1989. “*Creativity: what are we to measure?*”. In Glover, J.; Ronning, R. and Reynolds, C., *Handbook of Creativity*, Plenum Press, New York, ISBN: 0-306-43160-2.
- [3] Cardoso, A., Costa, E., Machado, P., Pereira, F.C., Gomes, P., 2000. “*An Architecture for Hybrid Creative Reasoning*”. In Pal, S.

K., Dillon, T. and Yeung, D. (Eds.), *Soft Computing in Case Based Reasoning*, Springer-Verlag.

- [4] Cardoso, A., Bento, C., Machado, P., Pereira, F.C., Junio 2002. “*Darwin and creationism reconciled: When computers should surprise us*”. *i3 magazine*, Mimo Caenepeel (Ed.), Issue 12, pp. 20–25, The European Network for Intelligent Information Interfaces.
- [5] Colton, S., Pease, A., Ritchie, G., 2001. “*The Effect of Input Knowledge on Creativity*”. Proceedings of the ICCBR'01 Workshop on Creative Systems. Vancouver, BC, Canada. Bento and Cardoso (Eds.). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.
- [6] De Bono, E., 1986. “*El pensamiento lateral. Manual de la creatividad*”. Paidós, Barcelona, Spain.
- [7] Dewey, J., 1919. “*How we think*”. D.C. Heath , Boston.
- [8] Fogel, D.B., 1999. “*Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*”. Wiley-IEEE Press, 2nd. Edition, 1999.
- [9] Gervás, P., Julio 2002. “*Exploring Quantitative Evaluations of the Creativity of Automatic Poets*”. Procs. of the 2nd. Workshop on Creative Systems, Approaches to Creativity in Artificial

Intelligence and Cognitive Science, Bento, Cardoso and Wiggins (Eds.). 15th European Conference on Artificial Intelligence (ECAI 2002) Lyon, France.

- [10] Goldberg, D., 1998. *"The design of innovation: lessons from genetic algorithms, lessons for the real world"*. IlliGAL Report N° 98004, Department of General Engineering, University of Illinois at Urbana-Champaign.
- [11] Guilford, J.P., 1967. *"The Nature of Human Intelligence"*. McGraw-Hill, New York.
- [12] Guilford, J.P., 1968. *"Intelligence, creativity and their educational implications"*. Robert Knapp, San Diego, CA.
- [13] Leake, D.B., 1996. *"Case-Based Reasoning: Experiences, Lessons, and Future Directions"*. AAAI Press.
- [14] Marin, R., de la Torre, S., 1991. *"Manual de la Creatividad"*. Vicens Vives, Barcelona.
- [15] Michael, W.B., 1977. *"Cognitive and affective components of creativity in mathematics and the physical sciences"*. In J.C. Stanley, W.C. George & C.H. Solano (Eds.), *The gifted and the creative: A fifty-year perspective*. Johns Hopkins University Press, Baltimore.

- [16] Pease, A., Winterstein, D., Colton, S., 2001. “*Evaluating Machine Creativity*”. Proceedings of the ICCBR'01 Workshop on Creative Systems. Vancouver, BC, Canada. Bento and Cardoso (Eds.). Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC.
- [17] Pereira, F.C., Cardoso, A., 2003. “*Optimality Principles for Conceptual Blending: A First Computational Approach*”. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour, AISB Journal, Alonso, E., Wiggins, G. (Eds.), 1(4), SSAISB.
- [18] Poincaré, H., 1913. “*The foundations of science*”. Science Press, Lancaster, PA.
- [19] Ritchie, G., 2001. “*Assessing Creativity*”. Procs. of the AISB'01 Symposium on AI and Creativity in Arts and Science, York, SSAISB, UK.
- [20] Romero, J., Machado, P., Santos, A., Cardoso, A., 2003. “*On the Development of Critics in Evolutionary Computation Artists*”. Proceedings of the EvoMusart workshop, 6th European Conference on Genetic Programming, Essex, UK.
- [21] Rossman, J., 1931. “*The psychology of the inventor: A study of the patentee*”. Inventors Publishing Co., Washington, DC.

- [22] Wallas, G., 1926. *“The art of thought”*. Harcourt Brace, New York.
- [23] Wiggins, G., 2001. *“Towards a more precise characterisation of creativity in AI”*. In Bento, C. and Cardoso, A., ed., Proceedings of the ICCBR’01 Workshop on Creative Systems. Naval Research Laboratory, Washington, DC.
- [24] Wiggins, G., 2003. *“Characterising Creative Systems”*. In Bento, C., Cardoso, A. and Gero, J., Proceedings of the IJCAI’03 Workshop on Creative Systems, IJCAI.

Nuevas Perspectivas en Informática Médica y Bioinformática

Victor Maojo¹, Fernando Martín² y José Manuel Vázquez³

¹Laboratorio de Inteligencia Artificial

Facultad de Informática. Universidad Politécnica de Madrid

Email: vmaojo@fi.upm.es

²Laboratorio de Inteligencia Artificial

Área de Bioinformática y Salud Pública

Instituto de Salud Carlos III. Madrid

³Laboratorio RNASA-IMEDIR

Universidade da Coruña

Email: jose@udc.es

Introducción

Los estudios sobre el carácter de arte/ciencia/ingeniería de muchas disciplinas han sido frecuentes durante los últimos cincuenta años. Un campo de la importancia de la informática ha sufrido este mismo debate, comenzando por el mismo nombre. Si en Estados Unidos recibe el nombre de “computer science” o ciencia de las computadoras, en Francia y otros países europeos se denomina como “informática” —en francés, *informatique*—, con un significado más amplio, incluyendo el estudio de los diversos aspectos relacionados con el manejo de información. Esta última acepción ha sido ampliamente utilizada en IM y BI, indicando no sólo el

uso de computadoras en medicina y biología, sino diversas teorías y métodos relacionados con la adquisición, almacenamiento, procesamiento, análisis, transferencia, búsqueda y visualización de información.

Un debate similar ha sido desarrollado en una disciplina, la IA, que tan relacionada ha estado con la IM. De hecho, la investigación de mayor impacto académico en IM, durante más de una década, fue realizada en temas de IA [1][2][3]. En la IA ha habido una cierta polémica acerca de su carácter científico y sus contenidos curriculares. El autor de este trabajo, asistiendo a un curso de IA en una universidad de los EE.UU., “sufrió” la impaciencia de un estudiante de doctorado en informática, que consideraba que los temas del curso eran meras repeticiones de otros que ya había recibido en cursos diferentes; pero, de hecho, la IA era el área original donde muchos de estos contenidos se habían investigado y generado aunque, una vez obtenidos, fueron rápidamente adoptados en otras áreas. Por ello, la IA ha tenido que estar siempre avanzando hacia delante para mantener su naturaleza científica. Un caso similar podría plantearse en la IM.

Naturaleza científica de la IM y BI

En el caso de la IM una de las justificaciones habituales [4] [5] para justificar la naturaleza científica de la disciplina es que está en relación con la medicina, que sería diferente al resto de las ciencias —aunque similares dudas podrían plantearse sobre el carácter científico de la medicina, como se verá más tarde—. En relación con la medicina, hay un acuerdo general

(al menos dentro de la IM) acerca de estas diferencias, que han podido afectar claramente el curso de la IM. Diferentes argumentos se han propuesto para soportar esta visión:

La complejidad de la información en medicina es diferente a otras disciplinas [6]. Se puede considerar que hay varios niveles de información, desde los de “bajo nivel”, relacionados con ciencias como la bioquímica o física, hasta los de “alto nivel”, utilizados en diagnóstico o manejo de información en la asistencia al paciente.

Metodologías. En medicina el método experimental es el más común para el desarrollo del campo científico. Las hipótesis pueden ser creadas de formas diferentes —por ejemplo, por abducción, generalización desde observaciones, analogías, simetrías—. Los experimentos se llevan a cabo siguiendo metodologías específicas para probar la veracidad de las hipótesis. La epidemiología y bioestadística han contribuido en las últimas décadas a diseño experimentales más cuidadosos en medicina [7], lo que ha llevado a la realización de ensayos clínicos más fiables y al auge actual de la medicina basada en la evidencia.

Creación vs descubrimiento. La medicina, como la IM, tratan con el cuerpo humano y los modelos que se pueden construir partiendo de él. Por ello, la investigación básica se realiza a niveles físicos, bioquímicos o biológicos para comprender fenómenos y elaborar teorías que engloben los hechos descubiertos. El cuerpo humano no se puede cambiar —al menos, por el momento— y los investigadores buscan descubrir hechos científicos. En otras disciplinas —por ejemplo, las ingenierías, economías, derecho— el

hombre puede crear nuevos componentes, reglas y modelos, sustituyendo aquellas piezas que considera oportuno. Por eso, en la medicina hay más una labor de descubrimiento de lo que todavía no es conocido en el cuerpo humano y la naturaleza, mientras en otras áreas —por ejemplo, la farmacogenómica—, hay también una labor de creación y diseño.

Aspectos cognitivos. Simon, Anderson y otros investigadores de la Ciencia Cognitiva [8], han demostrado, desde los años 1950, que los expertos en un área particular tienen estrategias de razonamiento que son utilizadas para esa área. Estas estrategias son específicas para cada dominio y sólo pueden ser transferidas a dominios y problemas similares. Por ejemplo, la pericia y habilidades en el razonamiento del juego del ajedrez no son útiles en áreas como, por ejemplo, música o derecho. Una conclusión similar puede ser obtenida analizando la informática y la medicina. Los métodos de análisis del razonamiento en medicina incluyen el método hipotético-deductivo, razonamiento abductivo, heurísticas (por ejemplo, representatividad, disponibilidad, anclaje y ajuste), abstracciones, analogías de casos, reconocimiento de patrones en datos de pacientes, inferencias causales, estrategias globales, comparación de conceptos, etc.

Estos aspectos cognitivos pueden cambiar durante la carrera profesional de un médico, desde una fase interpretada, en la que analiza cada hecho de forma individualizada y secuencial a otra en la que el médico elabora su proceso de razonamiento de forma automatizada, combinando de forma paralela varias estrategias y conocimiento declarativo y procedimental. Por ello, asistir a una rutinaria visita de pacientes en una sala de un hospital puede ser una pesadilla para un ingeniero o informático

no acostumbrado. El número y tipo de conceptos y vocabulario médico que el clínico maneja, la introducción de estimaciones difusas de la incertidumbre, la aproximación no algorítmica al diagnóstico médico, la comparación con múltiples casos anteriores y las explicaciones diagnósticas son difíciles de comprender para informáticos e ingenieros. Esta complejidad de los aspectos cognitivos ha provocado la dificultad existente para la creación de sistemas automatizados de ayuda a la toma de decisiones, que imiten los procesos cognitivos de razonamiento de los médicos.

Aspectos éticos. La investigación y práctica de la medicina han tenido regulaciones estrictas. Por ejemplo, la aprobación de un nuevo medicamento puede necesitar un lapso de 10-20 años, debido a la posibilidad de efectos secundarios. En Estados Unidos, la administración, a través de la agencia FDA, impone regulaciones muy estrictas y evaluaciones de software médico y dispositivos. De esta forma, todas las investigaciones realizadas deben tener en mente la necesidad de evitar daños a las personas.

Aspectos del entorno y ambiente. Los cambios en las circunstancias, naturales o artificiales, que rodean un grupo poblacional específico pueden provocar cambios en la conducta humana, naturaleza y en la investigación científica. Las especies animales, y en particular el hombre, pueden verse afectados por su entorno (por ejemplo, contaminación, ruido, radiaciones físicas) y los resultados de una investigación en un medio o entorno determinado no pueden ser extrapolables inmediatamente. Por ejemplo, las conclusiones de un estudio realizado en un hospital del tercer

mundo pueden no tener ninguna validez en un hospital de un país avanzado. En medicina —y también en IM— cualquier investigación debe tener en cuenta los factores ambientales que afectan al lugar donde se realiza dicha investigación.

Aspectos económicos. Según muchos investigadores, la mejora de las condiciones económicas en los países avanzados ha llevado a un incremento de medidas de higiene y prevención en salud pública que podrían ser tan o más importantes como el arsenal terapéutico que se ha creado en los últimos cincuenta años para la mejora de la salud de los ciudadanos [9].

Aspectos emocionales. Es conocido que la actitud emocional de una persona puede estar directamente relacionada con el resultado de cualquier estrategia terapéutica en una enfermedad específica. Los aspectos emocionales han sido dejados fuera, tradicionalmente, de la investigación médica. También de la investigación en Ciencia Cognitiva. Pero parece ahora que no sólo los aspectos cognitivos y médicos pueden ser reelaborados desde un punto de vista emocional. También la computación y la investigación de la interacción hombre-máquina puede ser reelaborada [10].

Si bien todas estas cuestiones pueden ser propuestas para ver las particularidades de la medicina, es interesante ver cómo cuestiones muy similares pueden ser planteadas en el uso de la biología. Así, en (1), la complejidad de la información en la biología no es muy diferente y se puede observar, por ejemplo, en los llamados “pathways” o vías metabólicas, de

enorme complejidad, lo que hace difícil su representación. En (2), las metodologías en biología son también similares. En relación a (3) hay algún matiz distinto, porque el conocimiento de la genética ha permitido, desde Mendel —considerando un punto de vista científico, porque ya antes se aplicaba en la agricultura o ganadería, de modo puramente empírico o intuitivo—, introducir variaciones en los códigos genéticos de las diversas especies. La ingeniería genética permitirá avanzar más aún en esta línea. En (4), los aspectos cognitivos de la biología no son tan complejos como en la medicina, como se verá más adelante, pero no son en absoluto obvios. Si en la medicina existe un componente de “arte”, también se produce en la investigación en biología. Con respecto a (5), existen también en la biología aspectos éticos y se han desarrollado aún más en los últimos años fuertes restricciones y condicionamientos de este tipo en múltiples investigaciones en biología, como son los estudios de células madre, experimentos de clonación, el mantenimiento de la privacidad y confidencialidad de los datos genéticos o las modificaciones genéticas de diversas especies en el laboratorio. En (6), aunque las condiciones de investigación en biología son, en general, más estables, al ser analizadas en el laboratorio, el comportamiento mismo de los genes depende del entorno. Como se volverá a comentar más adelante, la representación ADN \rightarrow ARN \rightarrow proteínas, parte clave del llamado dogma central de la genética, debido al premio Nobel Crick [11], ya no es válida para representar la complejidad de las múltiples interacciones que los genes tienen con otros genes y su entorno; por esta razón, surge la llamada Biología de Sistemas [12], como una continuación actualizada de conceptos que habían sido ya propuestos en la cibernética. En (7) y (8), estos aspectos, más en relación con la situación de

las personas, no parece haber una relación tan directa como en las anteriores, marcando ciertas diferencias entre la medicina y la biología.

Análisis basado en el “Informe Denning”

Para analizar las características científicas de ambas disciplinas se va a considerar a continuación un informe ya clásico, que es usado como referencia para considerar las perspectivas diferentes de la IM y BI como disciplinas científicas.

El informe conocido como “Denning”, elaborado por la ACM y el IEEE para el estudio de la informática como disciplina [13], planteaba la consideración de tres perspectivas para dicho análisis: (1) Teorías. (2) Abstracción. (3) Diseño. Un análisis similar podría ser de interés también para la IM y BI. De esta forma se pretende mostrar como la metodología usada por Denning puede estar indicada en áreas interdisciplinarias de la informática, como son la IM y BI.

Teoría

Desde esta perspectiva, habría cuatro fases en el desarrollo de una teoría en MI y BI: (1) Caracterizar objetos de estudio. (2) Generar hipótesis sobre las relaciones entre ellos, (3) Determinar si las relaciones son verdaderas. (4) Interpretar los resultados.

Como se verá más tarde, uno de los papeles o roles principales de los profesionales de la IM y la BI es la construcción de modelos informáticos

que puedan predecir la conducta de un paciente o situaciones sanitarias bajo circunstancias específicas (por ejemplo, mediante protocolos clínicos que tengan en cuenta el estado de un paciente ante determinados factores externos y cómo este estado puede ir evolucionando en el tiempo). Basándose en teorías de otras disciplinas científicas tales como, por ejemplo, las matemáticas, lógica, IA o física, se pueden desarrollar nuevas teorías para caracterizar los modelos que prueben estas teorías.

Si se considera otra disciplina informática como, por ejemplo, la “informática legal”, sería una cuestión muy discutible y llevaría seguramente a una amplia controversia ver si existen teorías científicas por detrás de algunos de los sistemas o aplicaciones construidos en esta disciplina. El autor de este trabajo y otros colegas han propuesto con anterioridad [14], que no son necesarias las teorías científicas subyacentes para probar las propiedades del modelo existente de leyes que se analiza, ya que dichas propiedades son, de hecho, ya explícitas. Después de analizar más extensamente este tema y considerar que existen muchos juristas y publicaciones que defienden el carácter científico de las bases del derecho, con diferentes ideas y perspectivas, el autor de este trabajo se inclina a pensar de forma diferente —al menos en una primera aproximación—, considerando ahora que la informática legal podría tener una perspectiva teórica más consistente y profunda que lo que podría aparecer a simple vista. Cambios en las teorías del derecho podrían llevar a modificar los modelos existentes y la informática legal tendría que tenerlas en consideración —aunque la duda es si el interés va más allá de aspectos

puramente filosóficos de la disciplina, como si se puede argumentar en el caso de la IM y la BI.

En medicina y biología se deben considerar varios niveles en la complejidad de la información y se pueden construir diversos modelos para justificar y predecir el determinado comportamiento de un objeto, en comparación con otros objetos ya existentes y en relación con su entorno. Ya que estos modelos se basan en teorías de los diferentes aspectos de los componentes del cuerpo humano o en hechos obtenidos por experimentación, se pueden probar las relaciones entre las características de un modelo. Nuevas teorías podrían ser también desarrolladas dentro de la IM y la BI, para, por ejemplo, describir procesos intrínsecos —tales como los metabólicos o fisiopatológicos— en términos de transferencia y procesamiento de información. Como ejemplo se puede citar un ejemplo de investigación anterior en IM [15], realizado para validar, desde un punto de vista lógico, criterios de uso apropiado médico. El método para crear estos criterios ha sido desarrollado por investigadores de la corporación RAND en Santa Mónica, California y está basado en estudios de la literatura científica y las opiniones de los panelistas. La metodología tiene dos problemas principales: Uno, las inconsistencias en los conocimientos expresados por los panelistas. Otro, las dificultades para el uso de los resultados en la práctica clínica, ya que el método ha sido diseñado para estudios en servicios de salud y evaluaciones de tecnologías sanitarias.

Sin entrar a considerar los componentes del modelo ni el diseño del sistema construido para automatizar el proceso de búsqueda de inconsistencias, se han creado avances teóricos para verificar la consistencia

lógica del modelo construido. Con un método basado en lógica multivalente y álgebra computacional, se puede probar la consistencia del conocimiento expresado en forma de conjuntos de criterios, expresados por los participantes en un panel. Una vez que la consistencia de todos los criterios ha sido confirmada, es posible inferir nuevo conocimiento. De esta forma el modelo soporta nuevas interpretaciones de los criterios, que pueden ser usados como guías para la práctica clínica.

Viendo este ejemplo y otros muchos realizados en múltiples áreas —por ejemplo, tratamiento de la incertidumbre en sistemas expertos médicos o modelos de razonamiento cualitativo en la toma de decisiones”— es posible considerar que ha habido una amplia investigación teórica dentro de la IM y la BI. Sin embargo, es también posible preguntarse si la investigación original pertenece a otras áreas —por ejemplo, probabilidad, IA o procesamiento digital de la señal— y, en realidad, su importancia en la IM y BI es relativa, al tratarse más bien de aplicaciones de teorías, métodos y modelos de otras disciplinas en ellas. Basándose en estas consideraciones sería posible hacerse la siguiente pregunta: ¿Existen teorías y modelos propios dentro de la BI y MI, que pudiesen ser considerados como pertenecientes a ambas disciplinas, al ser analizados desde otras disciplinas externas?

Abstracción

Según esta perspectiva se podría analizar la IM y la BI, considerando cuatro pasos: (1) Formación de hipótesis. (2) Construcción de modelos y

realización de predicciones, (3) Diseño de experimentos y recogida de datos, y (4) Análisis de resultados.

Según esta idea, en computación —según el informe Denning, aunque en este trabajo se generaliza a las informáticas interdisciplinarias o aplicadas—, una de las perspectivas de trabajo es la realización de un proceso de abstracción de situaciones particulares, partiendo de una hipótesis, que permita construir un modelo con capacidad predictiva, cuya validez pueda ser demostrada mediante experimentación. Esta definición, en el informe Denning, es un tanto ambigua, desde el punto de vista de metodología de la ciencia, ya que no está claro cómo se realizan las hipótesis “a priori”. Se entiende, en principio, que debe existir una teoría para construir modelos en informática, lo que es coherente desde un punto de vista científico.

Un ejemplo sería, en BI, una investigación de estructuras en 3-D de proteínas. Una vez construidos estos modelos partiendo de una hipótesis, se realiza una predicción de cuál debería ser la estructura y función de una determinada proteína. Una vez hecha esta aproximación, se realiza un experimento para validar su resultado. Esta investigación propuesta, que coincide con la perspectiva adoptada en este subapartado, no coincide sin embargo, con muchas de las investigaciones planteadas hasta la fecha en IM y BI.

Diseño

Desde esta perspectiva, hay cuatro pasos básicos en el desarrollo de aplicaciones informáticas en medicina: (1) Exponer requisitos. (2) Detallar especificaciones. (3) Diseño e implementación de sistemas. (4) Pruebas y evaluación del sistema.

El desarrollo de aplicaciones informáticas en medicina tiene algunas particularidades, comparado con los enfoques comunes en ingeniería del software. Los informáticos médicos pueden trabajar como desarrolladores, pero la complejidad creciente de los sistemas médicos hace necesario que se usen equipos con especialistas en temas como, por ejemplo, interfaces humanos, bases de datos, programación “web” o seguridad.

Los profesionales de la IM e BI, de forma similar a los profesionales de cualquier otro campo informático localizado en la intersección de dos campos diferentes (por ejemplo, informática jurídica) pueden ser considerados, desde otro punto de vista, como “mediadores” —en inglés, “brokers” —. Los profesionales de estas áreas informáticas pueden actuar como mediadores, intermediarios de información o consejeros, estableciendo puentes entre profesionales de la medicina, informática u otras áreas más específicas como la investigación básica, salud pública, epidemiología o economía de la salud, ya que cada una de ellas tiene sus particularidades. En informática, se pueden considerar ramas como la ingeniería del software, arquitecturas, IA y otras.

Desde el punto de vista de la ingeniería, la misión de estos profesionales de áreas interdisciplinarias informáticas sería similar: diseñar e implementar buenos sistemas. Los profesionales de la BI o IM actuarían como desarrolladores o mediadores en el diseño de sistemas informáticos. No habría demasiadas diferencias entre los bioinformáticos e informáticos médicos en este sentido. Desde esta perspectiva, los profesionales de ambas disciplinas actuarían con una perspectiva ingenieril.

Los profesionales de ambas disciplinas, por su conocimiento de las áreas implicadas en cada una de ellas —medicina e informática para la IM o biología e informática para la BI—, podrían tener un papel importante asegurando que los sistemas, por ejemplo, responden a problemas reales y al estado de la cuestión del área de aplicación.

En las labores puramente de ingeniería del software e implementación de sistemas de gran complejidad, profesionales de la informática podrían liderar estos aspectos. De esta forma, quedarían, para los profesionales de la IM o BI, tareas que podrían parecer más sencillas pero que requieren un conocimiento profundo de la informática y de sus dominios de aplicación, así como un conocimiento extenso de las lecciones aprendidas en cada disciplina, en décadas de experiencia y práctica, en temas como:

- Coordinación del proyecto.
- Conocimiento del estado de la cuestión en aplicaciones de BI o IM.

- Experiencia en casos o aplicaciones similares.
- Asegurar la calidad de la información y conocimiento.
- Explicar los diferentes conceptos, vocabularios y relaciones a ambas partes del equipo (médicos e ingenieros).
- Participación en procesos de diseño y modelado.
- Determinación de las mejores prácticas.
- Control en la recogida y selección de datos.
- Guías de actuación para la implementación de interfaces de usuario.
- Consejos sobre sistemas de codificación y clasificación.
- Detección de problemas de comunicación.
- Anticipación de las necesidades de los usuarios.
- Filtrado de información subjetiva.
- Estrategias de gestión del conocimiento.
- Importación y exportación de métodos y herramientas de otras áreas.
- Monitorización de procesos.
- Revisión de planes de explotación.
- Facilitar la transferencia de tecnología y entrenamiento de usuarios.
- Mantenimiento y actualización de sistemas.
- Restricciones legales.
- Consideraciones éticas.

Según esta tercera perspectiva, la labor de los profesionales de la BI y la IM sería muy similar.

Contrastes en los fundamentos científicos de la IM y la BI. Modelado.

Se ha dicho repetidamente que la medicina tiene un componente de arte y otro de ciencia [4][9][16][17]. Aunque la medicina ha experimentado avances en sus bases científicas durante el último siglo (y en particular en las últimas décadas), todavía no hay una teoría científica unificada, en sus fundamentos básicos, que pueda proveer un marco para interpretar la variedad de evidencias que la práctica clínica y epidemiológica requieren.

El gran número de paradigmas propuestos para la medicina, como, por ejemplo, medicina basada en la evidencia [7], medicina molecular [18], telemedicina [19], medicina basada en la economía de la salud [20], asistencia sanitaria de tipo tecnológico [21] y muchos otros, algunos complementarios, ilustra una inestabilidad en los fundamentos tecnológicos y científicos subyacentes y que están sujetos a cambios continuos y a veces impredecibles. La aparición imprevista de la WWW, y sus efectos en la medicina, son un caso aparte. Esta falta de principios claros y centrados o de teorías unificadas en la medicina pueden haber contribuido también a la diversidad extrema de direcciones dentro de la IM, y su tendencia predominante hacia la ingeniería, educación y práctica clínica.

Una mirada rápida a algunos temas emergentes en biología, relacionados con la complejidad evolutiva y sus implicaciones sanitarias, muestra avances de interés. Así se ve en la recientemente secuenciada pareja huésped-parásito de *A. Gambiae* y la bacteria de la malaria [22], que

podría ser la primera de las múltiples fuentes de modelado, a diversos niveles, para ayudar a comprender las complejidades de las aceleraciones evolutivas causadas por los pesticidas y resistencias a drogas. Estos temas han llegado a ser un tema popular en biología [23][24].

Los científicos biomédicos tienen un conocimiento muy parcial de los mecanismos de regulación y control que se producen en los organismos, sin ni siquiera considerar las circunstancias y contextos relacionados con el medio ambiente y evolución. Describir completamente las funciones biológicas y construir modelos lo más completos posibles del cuerpo humano, en el momento presente, es una tarea prácticamente imposible, como se puede ver en los esfuerzos realizados para crear nomenclaturas descriptivas en anatomía y varias especialidades, que deben ser integradas en representaciones del conocimiento más flexibles, escalables y actualizables, denominadas ontologías [25]. El enlace con modelos virtuales humanos (o de otros organismos) produce cuestiones fundamentales, todavía abiertas, en relación con el modelado matemático y computacional y sus fundamentos científicos, así como dificultades de implementación y problemas éticos acerca de cómo aplicar tales modelos en la práctica.

Es posible destacar en el momento presente que las contribuciones de la BI a tales modelizaciones tienden a ser de “abajo-arriba”, de la biología molecular a la fisiología. En contraste, la IM ha tendido a hacer representaciones de arriba-abajo, desde las manifestaciones clínicas a los procesos fisiopatológicos. Este contraste puede ser ilustrado por las diferencias entre el diagrama de un modelo de red causal-asociacional,

diseñado para la consulta en el diagnóstico y manejo del glaucoma, hace ya tres décadas [26] y un diagrama típico de apoptosis o muerte celular, como se muestra en la figura 1. El primer modelo (izquierda) tuvo un éxito académico significativo, dando lugar al sistema experto médico CASNET. El segundo (derecha) es una conceptualización puramente visual de una vía metabólica que implica la muerte celular [27], lo que requiere un conocimiento humano considerable.

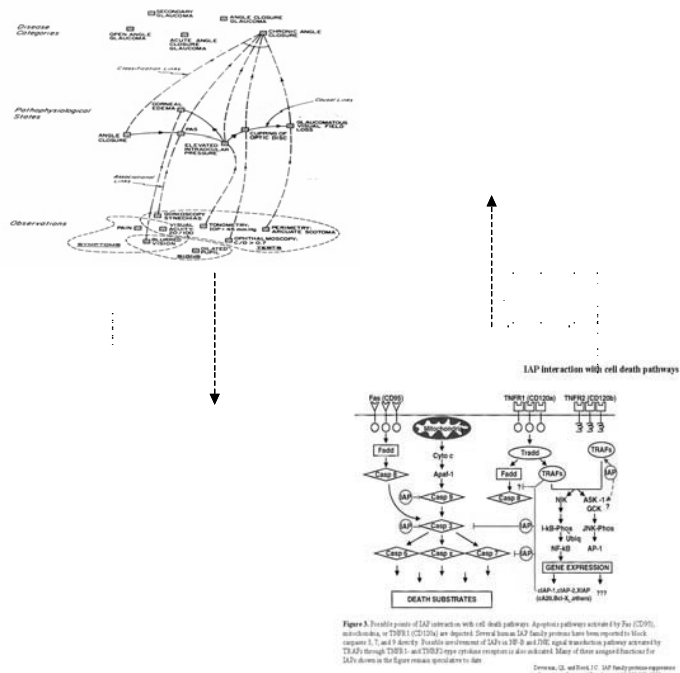


Figura 1: Una representación de los diferentes enfoques —arriba-abajo vs abajo-arriba— dentro de la IM y BI, respectivamente, para el modelado de procesos biomédicos

La BI ha estado centrada en tareas específicas que planteaban grandes retos, como ha sido el ejemplo del PGH, mientras que la IM se ha centrado más en información clínica y sus usos —aunque hay ejemplos tales como los anteriormente citados, y más recientemente el trabajo realizado en análisis de imágenes de datos y el Modelo Fundacional de la Anatomía, en el que se intenta crear puentes entre diversas disciplinas médicas [25]. Los tipos de modelado, en el caso de la BI, han sido determinados por las teorías científicas que dominan la biología, habiendo estado centradas, hasta hace poco tiempo, predominantemente en el análisis basado en secuencias.

De acuerdo a algunas perspectivas en la filosofía de la ciencia, los modelos deberían estar fuertemente relacionados con teorías [28] para tener éxito. La falta de teorías adecuadas que justifiquen las explicaciones fisiopatológicas dadas en la medicina, pueden explicar por qué el modelado causal ha probado ser una tarea tan difícil en la medicina. Hace años Jacob observó que la biología había fallado en intentar proporcionar ofrecer teorías matemáticas [29], como hacen las ciencias exactas, y los biólogos trabajaban usando modelos —“metáforas”— que no estaban directamente ligados a teorías.

Jacob observó que había —años 70— un número considerable de generalizaciones en biología, pero pocas teorías genuinas. La cuestión surge si la BI y la IM pueden ayudar a avanzar en la representación del conocimiento y su modelado, desde el nivel molecular al celular, tisular, orgánico, fisiológico y sistémico en organismos, podrían colocar la medicina en un lugar más científico. Cómo y cuando los científicos de

ambas disciplinas cooperarán y cómo y cuando competirán o si se juntarán en la disciplina llamada Informática Biomédica, es una cuestión de interés para el futuro.

Direcciones para el intercambio de métodos y técnicas entre la BI y la IM

Con anterioridad se ha visto el potencial de ambas disciplinas para una cierta interacción. No sólo comparten intereses, métodos y herramientas, sino que, además, cada una presenta necesidades que podrían ser complementadas por las características de la otra. En la figura 2 se presenta un modelo de interacción entre la IM y la BI.

La figura 2 ilustra una metáfora gráfica de las interacciones potenciales en temas que van desde los mecanismos biológicos básicos hasta sus aplicaciones en la asistencia sanitaria. Representa cómo ambas disciplinas tienen diferentes fuerzas y debilidades cuando tratan de aspectos que van desde los fundamentos biológicos al conocimiento experto en la construcción de aplicaciones médicas prácticas y genéricas, como pueden ser las basadas en la Web o de telemedicina. Refleja también cómo la IM (en el lado derecho), ha tenido, históricamente, un énfasis en el diseño e implementación de sistemas sanitarios, como se muestra en el estrato superior del diagrama, y ha contribuido al desarrollo de numerosas aplicaciones de software médico en los últimos 30-40 años. La flecha de mayor tamaño que señala desde el lado de la IM al lado de la BI, muestra cómo la experiencia de los profesionales de la IM podría ser de utilidad a

los profesionales de la BI en el desarrollo de aplicaciones médicas basadas en datos de expresión génomica u otros datos genómicos, como “microarrays”. En contraste, la BI que está más cerca de los fundamentos biológicos (especialmente en los niveles moleculares y celulares) que la IM, está representada con un área mayor en el nivel más bajo del estrato. Aquí la flecha grande que va de izquierda a derecha muestra cómo la IM se podría beneficiar de un flujo neto de experiencia y métodos de la BI en los niveles de fundamentos básicos y análisis biológico.

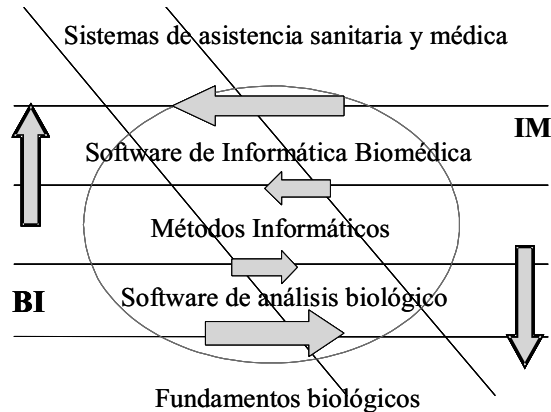


Figura 2: Metáfora gráfica de las sinergias potenciales y competición entre la BI y la IM, indicando las direcciones del énfasis actual hacia el intercambio de métodos y herramientas dentro de la Informática Biomédica. La zona diagonal representa el área de mayores complementariedades y sinergias, mientras que la zona oval central representa una zona de posible retos competitivos entre la IM y la BI

En el estrato central de los métodos informáticos el intercambio de métodos, experiencia y enfoques está más equilibrado, con flechas de talla menor señalando a cada dirección hacia la zona de sinergias potenciales que bajan hacia la diagonal central del diagrama. La BI tiende a usar más modelos matemáticos e informáticos en la actualidad, mientras la IM tiende a estar más centrada en la informática clínica, con una predominancia de sistemas, software y enfoques basados en la ingeniería del conocimiento. La zona oval central en la figura representa el potencial para la comparación entre ambas disciplinas, centrada en el estrato central de los métodos informáticos, que tienden a ser diferentes en metas y técnicas entre la IM y la BI en el momento presente. Estas diferencias tienden a reflejar las asimetrías en los estratos vecinos.

El mayor peso teórico en los principios biofísicos y bioquímicos y los logros en el modelado le dan a la BI una mayor fuerza científica que la IM, dado el énfasis en el descubrimiento científico que han tenido los investigadores del área, con el fin de avanzar los fundamentos biológicos de la disciplina. El énfasis de la IM en la práctica sanitaria no requiere esos modelos salvo en áreas tales como imágenes, análisis de señales especializadas e integración de instrumentación, que tienden a estar más cerca de la ingeniería biomédica que de la IM.

En resumen, se plantea en esta figura las posibilidades para una interacción y sinergia en la BI y la IM. Esto puede llevar a la fusión de ambas en la Informática Biomédica si la sinergia puede llevar a descubrimientos significativos en la ciencia o al mantenimiento de la separación entre ambas si los contrastes y las presiones competitivas

predominan. En este nuevo marco teórico y de investigación, una serie de técnicas y métodos pueden ser creados para servir de apoyo a la medicina genómica.

Bibliografía

- [1] Szolovits, P., (Ed.), 1982. “*Artificial Intelligence in Medicine*”. Westview Press. Boulder, CO.
- [2] Clancey, W., Shortliffe, E.H., (Eds), 1984. “*Readings in Medical Artificial Intelligence: The First Decade*”. Reading: Addison Wesley, ISBN: 0-201-10854-2.
- [3] Buchanan, B., Shortliffe, E.H., 1984. “*Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*”. Addison-Wesley, New York, USA.
- [4] Greenes, R.A., Shortliffe, E.H., 1990. “*Medical Informatics: An Emerging Academic Discipline and Institutional Priority*”. JAMA 1990, vol. 263, pp. 1114-1120.
- [5] Shortliffe, E.H., Perreault, L., (Eds), 2001. “*Medical Informatics: Computer Applications in Health Care*”. 2nd Edition. Springer Verlag, New York.
- [6] Blois, M., 1984. “*Information and Medicine: The Nature of Medical Descriptions*”. Univ. of California Press, Berkeley, CA.
- [7] Sackett, D., 1997. “*Evidence-Based Medicine: How to Practice & Teach EBM*”. Harcourt Brace.

- [8] Anderson J., 1999. “*Cognitive Psychology and Its Implications*” (5th Ed). Worth Publishing.
- [9] Weatherall, D., 1985. “*Science and the Quiet Art: Medical Research & Patient Care*”. Oxford University Press, Oxford.
- [10] Picard R., 1997. “*Affective Computing*”. The MIT Press. Boston, MA.
- [11] Crick F.H.C., 1958. “*The Biological Replication of Macromolecules*”. Symp. Soc. Exp. Biol., vol. XII, pp. 138–163.
- [12] Kitano, H., 2002. “*Systems biology: a brief overview*”. Science, vol. 295 (5560), pp. 1662-1664
- [13] Denning, P., Comer, D., Gries, D., Mulder, M.C., Tucker, A., Young, P., 1989. “*Computing as a discipline*”. Communications of the ACM, vol. 32 (1), pp. 9–23.
- [14] Maojo, V., Martín-Sánchez, F., Crespo, J., Billhardt, H., 2002. “*Theory, Abstraction and Design in Medical Informatics*”. Methods of Information in Medicine, vol. 41, pp. 44–50.
- [15] Maojo, V., Laita, L., Roanes-Lozano, E., Crespo, J., Rodríguez-Pedrosa, J., 2000. “*A New Computerized Method to Verify and Disseminate Medical Appropriateness Criteria*”. Lecture Notes in Computer Science 1933, pp. 212–217.
- [16] Kulikowski, C.A., 2002. “*The micro-macro spectrum of medical informatics challenges: from molecular medicine to transforming health care in a globalizing society*”. Methods Inf. Med., vol. 41 (1), pp. 20–4.

- [17] Kulikowski, C., 1996. “*The Scope of Medical Informatics as Art and Science*”. *Methods of Information in Medicine*, vol. 35, pp. 185–188.
- [18] Ross, D., 2002. “*Introduction to Molecular Medicine*”. Springer Verlag.
- [19] Bauer, M., Ringel, A., 1999. “*Telemedicine and the Reinvention of Healthcare*” (Healthcare Informatics Executive Management Series). McGraw-Hill Profesional.
- [20] Edwards, R.T., 2001. “*Paradigms and Research Programmes: Is it Time to Move from Health Care Economics to Health Economics?*”. *Health Econ.*, vol. 10, pp. 635–649.
- [21] Greenes, R., 2002. “*Future of medical knowledge management and decision support*”. *Stud. Health Technol. Inform.*, vol. 80, pp. 29–44.
- [22] Coluzzi, M., Sabatini, A., della Torre, A., Di Deco, M.A., Petrarca, V., 2002. “*A polytene chromosome analysis of the Anopheles gambiae species complex*”. *Science*, vol. 298 (5597), pp. 1415–1418.
- [23] Dawkins, R., 1999. “*The Extended Phenotype: The Long Reach of the Gene*”. Oxford University Press, Oxford, UK.
- [24] Palumbi, S., 2002. “*The Evolution Explosion: How Humans Cause Rapid Evolutionary Change*”. Norton & Company.

- [25] Rosse, C., Shapiro, L.G., Brinkley, J.F., 1998. “*The digital anatomist foundational model: principles for defining and structuring its concept domain*”. Proc. AMIA Symp., pp. 820–824.
- [26] Kulikowski, C., Weiss, S., 1971. “*Computer-based Models of Glaucoma*”. Report n° 3, Department of Computer Science, Computers in Biomedicine, Rutgers University.
- [27] Deveraux, Q.L., Reed, J.C., 1999. “*IAP family proteins-suppressors of apoptosis*”. Genes and Development, vol. 13, pp. 239–252.
- [28] Suppe, F., 1979. “*The Structure of Scientific Theories*”. Univ. of Illinois Press.
- [29] Jacob, F., 1993. “*The Logic of Life*”. Princeton Univ. Pr. Princeton, USA.

Predicción Pesquera Utilizando Algoritmos Basados en I.A.

José Manuel Cotos¹, Alfonso Iglesias²,

Pedro Saco¹ y Bernardino Arcay²

¹Depto de Electrónica y Computación.

Universidad de Santiago de Compostela

Email: {manel.cotos, elpedro}@usc.es

²Depto de Tecnologías de la Información y las Comunicaciones.

Universidade da Coruña

Email: {elfonso,cibarca}@udc.es

Introducción

Planteamiento del problema

La pesca, entendida como la explotación y aprovechamiento de los recursos marinos, es uno de los principales sectores económicos de Galicia, y también la base de otras actividades industriales y de servicios. Según datos de la Consellería de Pesca (ver figura 1a), el total de la población gallega ocupada en este sector alcanza los 41.600 habitantes, lo que supone el 4,5% de la población activa de la Comunidad Autónoma.

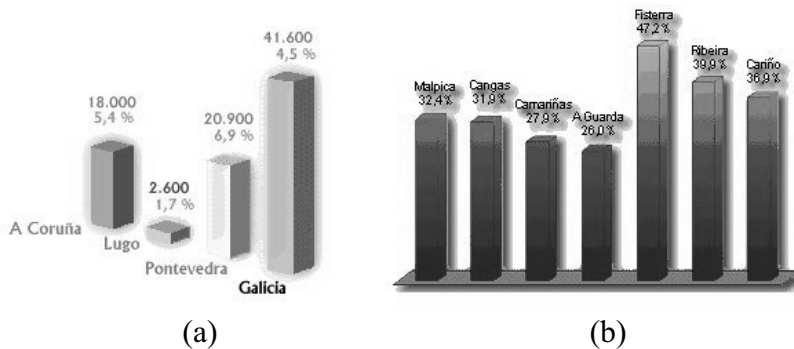


Figura 1: Datos sobre población activa gallega ocupada en el sector pesquero. (a). Población ocupada en la pesca por provincias, y porcentaje respecto a la población activa. (Datos de 2002. Xunta de Galicia [1]). (b). Importancia relativa de la pesca en algunas poblaciones costeras. (Datos de 2002. Xunta de Galicia [1])

Sin embargo, la pesca, por su propia naturaleza, no es una actividad que afecte igual a toda la comunidad gallega. Por eso, su importancia en algunas localidades costeras es muy superior a la media gallega (ver figura 1b), dándose casos como el de Fisterra, donde el 47,2% de la población activa se dedica a la pesca.

No deben olvidarse los puestos de trabajo vinculados con la pesca (ver figura 2), que aumentan a casi 120.000 los puestos de trabajo generados, lo que permite afirmar que el 12,2% de los empleos en Galicia

tienen como base la pesca, ya sea directa o indirectamente, convirtiéndose en la región de Europa con más dependencia de la misma.

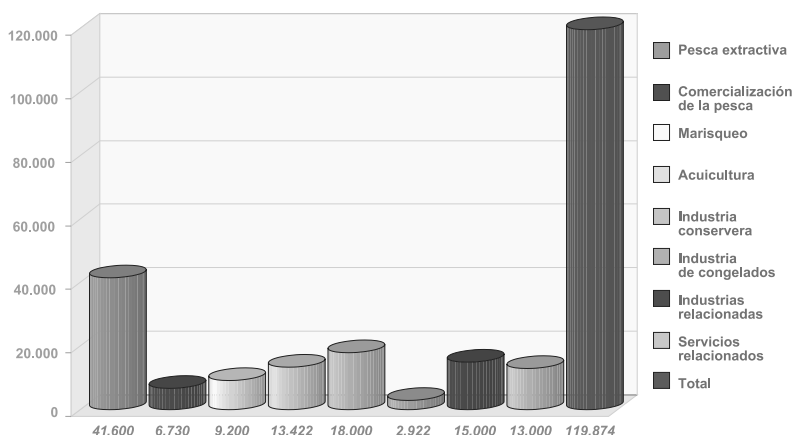


Figura 2: Empleos directos e indirectos producidos por el sector de la pesca. Las cifras indican el número total de empleos generados en cada sector. (Datos de 2002, Xunta de Galicia [1])

Queda patente, por tanto, la importancia del sector pesquero dentro del ámbito económico gallego. Lo que antes era un negocio artesanal localizado en las zonas costeras se ha convertido en una potente industria, motor de la economía gallega. Este sector pesquero, sin embargo, se ha visto amenazado recientemente por diversos acontecimientos tales como desastres ecológicos, políticas de cuotas, paros biológicos y ampliación de las zonas económicas exclusivas de los países ribereños. Además, acuerdos comerciales con otros países introducen en el mercado pescado de otros

países a precios muy bajos. Por otro lado, es bien conocida la tendencia de diversos organismos de ámbito nacional e internacional para la protección y conservación de los recursos. Con ello se pretende conseguir un desarrollo sostenible que mantenga los océanos como una importante fuente de alimentos de la población mundial.

Para mantener la competitividad de la flota gallega, pensamos que es necesario disponer de la tecnología más avanzada y de la mayor cantidad de información posible. En este ámbito de aplicación, los nuevos sensores remotos son unas herramientas privilegiadas si los utilizamos como fuentes de información de las condiciones biológicas del entorno pesquero. En el trabajo que aquí presentamos, aplicamos sistemas conexionistas para la predicción y optimización de los recursos disponibles, empleando para ello la información obtenida de manera remota.

De esta forma, podremos reducir los tiempos de búsqueda de pesquerías, con el consiguiente ahorro de tiempo y combustible que ello conlleva, y aumentando dentro de las posibilidades legales el volumen de capturas.

Las razones económicas no son las únicas que nos motivan a realizar este trabajo. El lanzamiento de nuevos satélites (Orbview-2) con su consiguiente tratamiento digital de la información así como los nuevos paradigmas neuronales y neuro-difusos aquí empleados, son nuevos retos que pretendemos abordar desde un punto de vista práctico.

Objetivos

Este trabajo explica los fundamentos de un nuevo sistema de apoyo a pesquerías. Para el desarrollo del mismo, hemos empleado técnicas conexionistas, tratamiento digital de imágenes, así como lógica difusa. Se han integrado todas estas técnicas con la intención de conseguir un sistema que facilite el aumento del rendimiento de pesquerías pelágicas. De esta forma, seremos capaces de gestionar y aprovechar los recursos pesqueros sin ponerlos en peligro. Concretamente, a partir de los datos obtenidos de varios sensores remotos y del cuaderno de pesca de un barco colaborador, nuestro sistema proporciona la probabilidad de conseguir capturas de una especie de tiburón pelágico, el *Prionace Glauca*, más conocido como tiburón azul.

El sistema desarrollado es extensible a cualquier especie marina de la cual se posea suficiente información. Por ello, puede convertirse en una herramienta útil para la optimización y gestión de los recursos pesqueros, sobre todo en ámbitos geográficos como el gallego, con una economía claramente dependiente del sector pesquero. Para demostrar la afirmación anterior, en esta introducción proporcionamos datos económicos que evidencian la importancia de un sector maltratado por desgracias ecológicas y, en algunos casos, por la sobreexplotación irracional de recursos.

Con este sistema que aquí presentamos, esperamos conseguir un aumento de los beneficios de los usuarios del mismo, mediante un desarrollo sostenible.

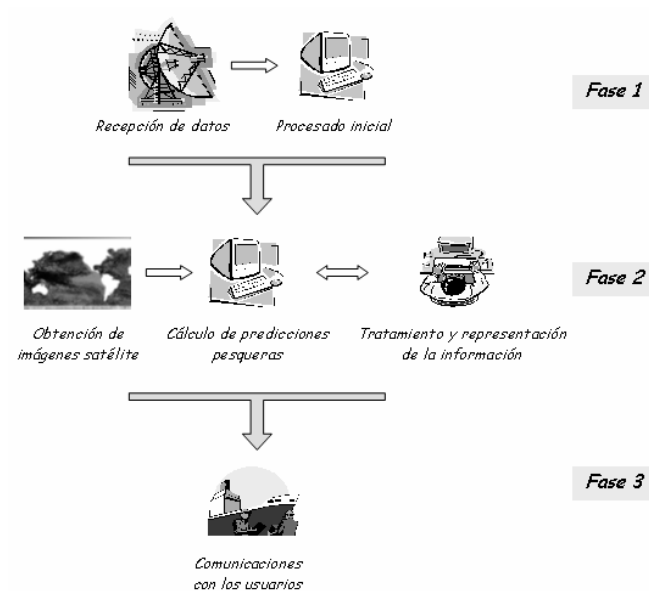


Figura 3: Organigrama del sistema de apoyo a pesquerías del Laboratorio de Sistemas de la Universidad de Santiago

El sistema de apoyo a pesquerías de la Universidad de Santiago se puede descomponer en tres fases diferenciadas, como se puede ver en la figura 3, cuyo objetivo final es transmitir productos susceptibles de ser enviados a las unidades embarcadas.

La *fase 1* tiene como misión principal la recepción de información procedente de varios satélites. El sistema se compone de un parque de antenas que permiten la recepción de distintos satélites como NOAA (en alta y baja resolución), OrbView-2 (imágenes del sensor SeaWifs) y Meteosat. Los datos recogidos por las antenas se almacenan en un sistema

de *backup* y se distribuyen a todos los equipos que están formando parte de una red local. Cualquiera de estos equipos tiene acceso a los datos para poder visualizarlos y procesarlos. Puede encontrarse una descripción exhaustiva de esta fase en [2], [3].

El objetivo de este trabajo es, empleando los datos recibidos como resultado de la *fase 1*, elaborar la *fase 2*, que consistirá en (ver figura 4):

- Aplicar técnicas de procesamiento digital sobre los productos iniciales para conseguir nuevos datos con significado biológico, por ejemplo los siguientes filtros pasa-alta para detectar la existencia de frentes térmicos:
 - DoG (ver [4]).
 - Cluster-Shade (ver [5]).
- Visualizar esta información independientemente de la plataforma usada.
- Estudio de la sensibilidad y correlación de los datos iniciales empleando:
 - Análisis de componentes principales.
 - Mapas Autoorganizativos de Kohonen.
- Calcular la probabilidad de obtener capturas pesqueras en función de parámetros medioambientales obtenidos mediante teledetección. Para ello probaremos las siguientes técnicas:
 - Redes entrenadas con el algoritmo de retropropagación (*backpropagation*).
 - Redes RBF (Radial Basis Function).

- Redes funcionales.
- ANFIS (Sistema de inferencia neuro-difuso)
- Gestionar toda la información de manera eficiente y con un control centralizado. Para ello, diseñaremos una base de datos apropiada siguiendo el modelo E-R.

En la *fase 3*, nuestro sistema se comunica con los usuarios. La información puesta a disposición de la unidad embarcada puede ser enviada haciendo uso de diversos medios de comunicación, por ejemplo la red global por satélite Inmarsat (figura 5). Esta red de satélites proporciona una gran diversidad de servicios: telefonía, datos, fax, web,... Es usada también para comunicaciones de desastre o emergencia marítima, por lo que no resulta infrecuente hallarse con barcos que cuentan con este servicio de comunicaciones.

Además, la disponibilidad de conexión a la red Internet permite realizar el envío de información haciendo uso del correo electrónico. El software embarcado TUNAFIS 4.2 desarrollado por otros miembros del Laboratorio de Sistemas permite gestionar cuentas de usuario de correo electrónico y extraer la información enviada para su gestión en el interfaz gráfico.

Una última posibilidad contemplada es el envío de información a través de los servicios de datos implementados en la telefonía móvil GSM. En las inmediaciones de costa es frecuente contar con cobertura para estas redes de comunicaciones, por lo que se está en condiciones de recibir la

información con un simple terminal móvil GSM conectado al ordenador personal de a bordo.

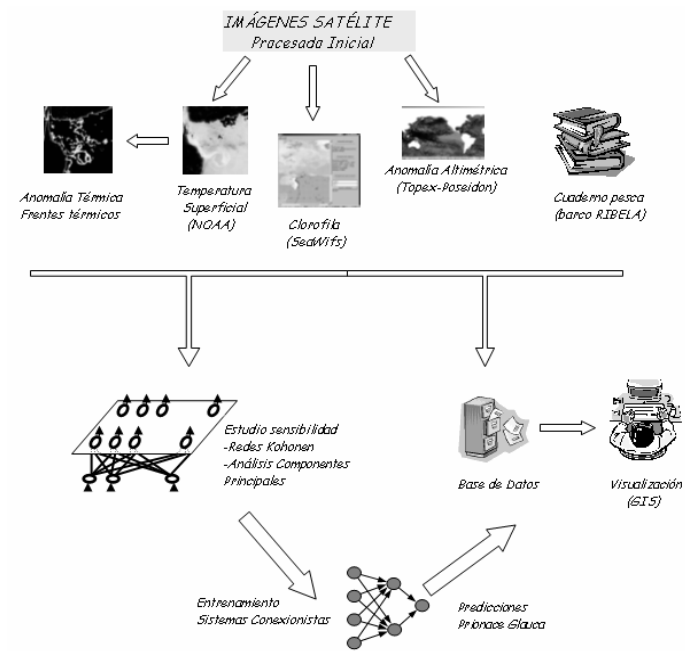


Figura 4: Organigrama de la fase 2

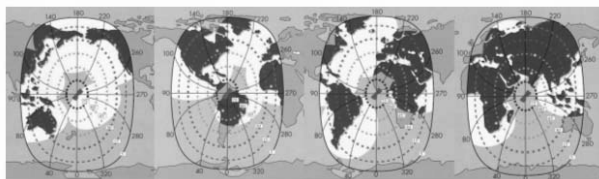


Figura 5: Cobertura de la red de satélites geoestacionarios Inmarsat

Fuentes de datos

Los datos que empleamos provienen de sensores remotos instalados a bordo de satélites. Se entiende por *teledetección* todo conjunto de técnicas que permiten obtener información de un objeto, área o fenómeno, a través del análisis de los datos obtenidos por un dispositivo que no está en contacto físico con el objeto, área o fenómeno que pretendemos estudiar [6]. Si particularizamos esta definición para el caso que nos ocupa, en el que el objeto a estudiar es la superficie marina, y el dispositivo un sensor instalado a bordo de un determinado satélite, podríamos definir la teledetección en oceanografía como la ciencia que estudia la superficie oceánica a través de las imágenes conseguidas por medio de satélites.

Podemos adelantar que la teledetección es una técnica que presenta una serie de ventajas respecto a las técnicas tradicionales de muestreo (barco tomando datos, por ejemplo), a saber:

- Cobertura global y periódica de la superficie.
- Visión panorámica de una gran extensión de manera simultánea.
- Información en zonas no visibles del espectro (infrarrojo térmico, infrarrojo medio y microondas).
- Formato digital de la información.

Hemos empleado para este trabajo imágenes de la serie NOAA, OrbView-2 y Topex-Poseidon. En capítulos posteriores explicaremos con

más detalle las ventajas enumeradas, así como los tipos de satélites e información obtenidos de cada uno de ellos.

Muy a menudo, la teledetección se complementa con otras técnicas convencionales para el tratamiento de información, como en el caso que nos ocupa. Es ya bastante frecuente que los datos derivados de la teledetección se combinen con datos de campo en *Sistemas de Información Geográficos* (SIG) [7].

Además de los datos conseguidos del procesado de imágenes de satélite, tenemos datos de campo procedentes de un barco pesquero colaborador. El método de pesca es el palangre. La información anotada consiste en las coordenadas geográficas del inicio y final del palangre, así como las capturas de cada especie recogidas diariamente a lo largo de los años 1998 y 1999.

Metodología

Redes Funcionales

Desde mediados de los años 80, han aparecido extensiones de los modelos de redes de neuronas. Ejemplos de extensiones son las redes de alto orden, redes neuronales probabilísticas [8] y las redes neuronales basadas en “*wavelets*” [9]. No obstante, estos nuevos modelos todavía actúan como cajas negras y no tienen en cuenta la estructura funcional y las propiedades del objeto a modelizar. Una característica importante de las redes funcionales es la posibilidad de tratar con restricciones funcionales

determinadas por propiedades que pueden conocerse del modelo. Estas restricciones conducirán a una topología determinada de red y, por tanto, a un sistema de ecuaciones funcionales.

Las ecuaciones funcionales son introducidas por *Castillo et al* [10] como una extensión de las redes neuronales. De una forma simple, pero rigurosa, podemos definir una *Red Funcional* como una Red Neuronal en la que los pesos de las neuronas han sido sustituidos por un conjunto de funciones. Algunas de sus ventajas son [11] :

1. Al contrario que las redes neuronales, las redes funcionales pueden reproducir algunas propiedades físicas, que conducen de una manera natural, a la correspondiente red. Ello es así si empleamos alguna expresión con significado físico dentro de la base de funciones. En nuestro caso, al no disponer de esa información, no podremos disfrutar de esta ventaja.
2. La estimación de los parámetros de la red se puede obtener resolviendo un sistema lineal de ecuaciones, lo que significa una solución rápida y única, que es el mínimo global de la función de error.

Las redes funcionales han sido aplicadas con éxito en problemas referentes al diagnóstico médico y a las distribuciones bayesianas conjugadas. La formulación de estos problemas, su resolución empleando redes funcionales, así como todo el formalismo propio de las ecuaciones y redes funcionales puede consultarse en el libro de *Castillo et al* [11].

Una red funcional consta de los siguientes elementos:

1. *Una capa de entrada de unidades de almacenamiento.* Esta capa contiene los datos de entrada.
2. *Una capa de salida de unidades de almacenamiento.* Esta capa contiene los datos de salida.
3. *Una o varias capas de unidades procesadoras.* Estas unidades evalúan un conjunto de valores de entrada, que vienen de la capa previa (una unidad intermedia o la capa de entrada) y calculan valores que serán tenidos en cuenta para la siguiente capa. Cada neurona va a tener asociada una función neuronal propia que puede poseer varios argumentos o entradas. De esta forma, podemos introducir en cada unidad procesadora parte del modelo matemático que ayude a explicar nuestro problema.
4. *Ninguna, una o varias capas de unidades de almacenamiento intermedias.* Estas capas contienen unidades que almacenan información intermedia producida por las unidades neuronales. Por tanto, nos permiten forzar la coincidencia de las salidas de las unidades procesadoras.
5. *Un conjunto de enlaces dirigidos.* Conectan unidades de entrada o capas intermedias a unidades neuronales, y unidades neuronales a unidades intermedias o de salida.

A continuación introducimos un ejemplo de redes funcionales, el modelo separable.

Modelo Separable

Supongamos la siguiente red funcional (ver figura 6), introducida por *Castillo et al* [11] en su capítulo referente a distintos modelos de redes funcionales. En él, los autores demuestran que la red de la figura 6 es el resultado obtenido al intentar simplificar topologías más complicadas.

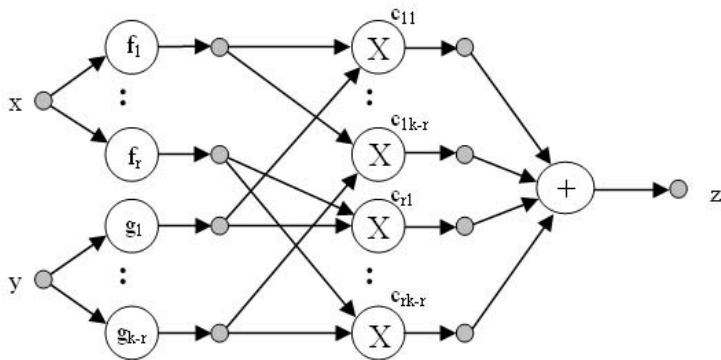


Figura 6: Red funcional, modelo de separabilidad

Unicidad de la solución

Si suponemos que $\{f_1(x), \dots, f_r(x)\}$ y $\{g_1(y), \dots, g_{k-r}(y)\}$ son funciones linealmente independientes, entonces

$$z = F(x, y) = \sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x) g_j(y) \quad (2.1)$$

Nuestro objetivo sería entonces calcular los coeficientes c_{ij} tales que se minimice el error. Puede demostrarse que la solución es única, para ello supongamos dos conjuntos de coeficientes distintos c_{ij} y c_{ij}^* tales que:

$$z = F(x, y) = \sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x) g_j(y) = \sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij}^* f_i(x) g_j(y) \quad (2.2)$$

Entonces, se puede escribir

$$\sum_{i=1}^r \sum_{j=1}^{k-r} (c_{ij} - c_{ij}^*) f_i(x) g_j(y) = 0 \quad (2.3)$$

Como los dos conjuntos de funciones $\{f_1(x), \dots, f_r(x)\}$ y $\{g_1(y), \dots, g_{k-r}(y)\}$ son linealmente independientes, también lo es el conjunto $\{f_i(x)g_j(y) \mid i=1, \dots, r; j=1, \dots, k-r\}$, por tanto

$$c_{ij} = c_{ij}^* \quad i = 1, \dots, r; j = 1, \dots, k-r \quad (2.4)$$

Aprendizaje del modelo

Supongamos que tenemos un conjunto de aprendizaje de dos entradas x e y , y una salida z . Nuestro objetivo será calcular los coeficientes c_{ij} óptimos de forma que se minimice el error entre la salida teórica y la

calculada por la red. Para ello emplearemos un simple método de mínimos cuadrados.

Supongamos que disponemos de conjuntos de funciones conocidas $\{f_1(x), \dots, f_r(x)\}$ y $\{g_1(y), \dots, g_{s-r}(y)\}$ que son linealmente independientes. Vamos a denotar por $\{(x_{0i}, x_{1i}, x_{2i}) \ i=1, \dots, n\}$ nuestro conjunto de entrenamiento, formado por n patrones distintos. x_0 , x_1 e x_2 se refieren a z , x e y respectivamente. Entonces, el error se puede medir según (2.5):

$$e_i = x_{0i} - \sum_{i=1}^r \sum_{j=1}^{s-r} c_{ij} f_i(x_{1i}) g_j(x_{2i}) \quad i=1, \dots, n \quad (2.5)$$

Por tanto, para encontrar los coeficientes óptimos, es necesario minimizar la suma $Q = \sum_{k=1}^n e_k^2$. Al no tener condiciones adicionales, el mínimo puede ser obtenido resolviendo el sistema de ecuaciones lineales, tomando como incógnitas los coeficientes c_{ij} .

$$\frac{\partial Q}{\partial c_{pq}} = 2 \sum_{k=1}^n e_k f_p(x_{1k}) g_q(x_{2k}) = 0 \quad p=1, \dots, r \ ; \ q=1, \dots, s-r \quad (2.6)$$

El modelo separable se ha empleado con éxito para asociar datos a modelos funcionales, como demuestran *Castillo et al* [11]. Concretamente,

se creó una tabla de tripletas de valores (x_0, x_1, x_2) empleando el modelo siguiente:

$$x_0 = f(x_1, x_2) = 1 + x_1 - x_2 - x_1^2 x_2 \quad (2.7)$$

Considerando una familia polinomial de funciones $\{1, x, x^2\}$, se llega, después del entrenamiento, al modelo con el cual se consiguieron los datos.

ANFIS

El empleo de redes neuronales permite el aprendizaje y minimización de una función error, con la ventaja de aprovechar la computación paralela. Sin embargo, estas técnicas presentan varios puntos débiles, como la representación del conocimiento y la extracción de reglas lingüísticas. En un perceptrón multicapa, por ejemplo, el conocimiento está distribuido en un conjunto de pesos difícil de interpretar y de convertir en reglas. La lógica difusa, en cambio, traduce los términos lingüísticos de las reglas en variables que el sistema acepta sin problema.

El propósito de los sistemas neuro-difusos (ANFIS) es intentar combinar los beneficios de ambas técnicas. Para ello, las reglas del sistema se expresan como expresiones difusas, y para que el sistema se entrene usamos los esquemas de aprendizaje de las redes neuronales. Los sistemas

neuro-difusos permiten la incorporación de datos lingüísticos y numéricos, y facilitan la extracción de reglas a partir de datos numéricos.

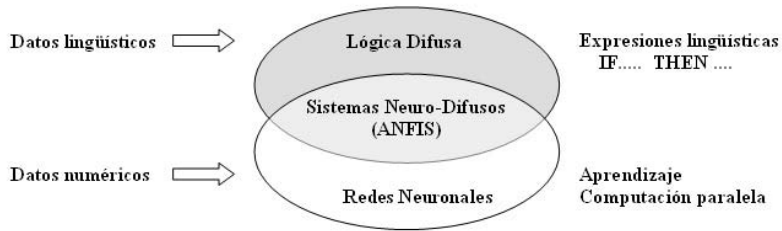


Figura 7: Los sistemas neuro-difusos integran la lógica difusa y las redes neuronales

Los sistemas neuro-difusos pueden ser divididos en 2 grupos principales:

1. Sistemas de inferencia neuro-difusos.
2. Redes neuronales difusas.

El origen de los sistemas de inferencia neuro-difusos es incorporar conceptos como aprendizaje y paralelismo a los sistemas de inferencia difusos. Los sistemas de inferencia neuro-difusos realizan inferencia difusa [12], [13] ajustando parámetros internos de las reglas con algoritmos de aprendizaje propios de redes neuronales. Una posible realización consistiría en que cada regla se ejecutaría mediante una red neuronal [14]. Otra posibilidad es la empleada en nuestro trabajo, la arquitectura ANFIS.

Las redes neuronales difusas están formadas por dos componentes en un mismo subsistema, una red neuronal y un sistema difuso. Una posibilidad de combinar ambos componentes sería sustituir la suma ponderada de una neurona por una operación difusa, o incluso un bloque externo que tradujera la información lingüística a la entrada correspondiente de la red.

Descripción de la arquitectura ANFIS

Por simplicidad, vamos a asumir que nuestro sistema de inferencia difuso de partida tiene dos entradas x e y , además de una salida f . Supongamos que la base de reglas contiene dos reglas difusas del tipo 3, Takagi-Sugeno [15].

En la figura 8 se ilustran el razonamiento difuso (figura 8a) empleando las reglas antes descritas y en la figura 8b la arquitectura ANFIS equivalente.

Regla 1: IF x es A₁ e AND es B₁, THEN f₁=p₁x+q₁y+r₁
 Regla 2: IF x es A₂ e AND es B₂, THEN f₂=p₂x+q₂y+r₂

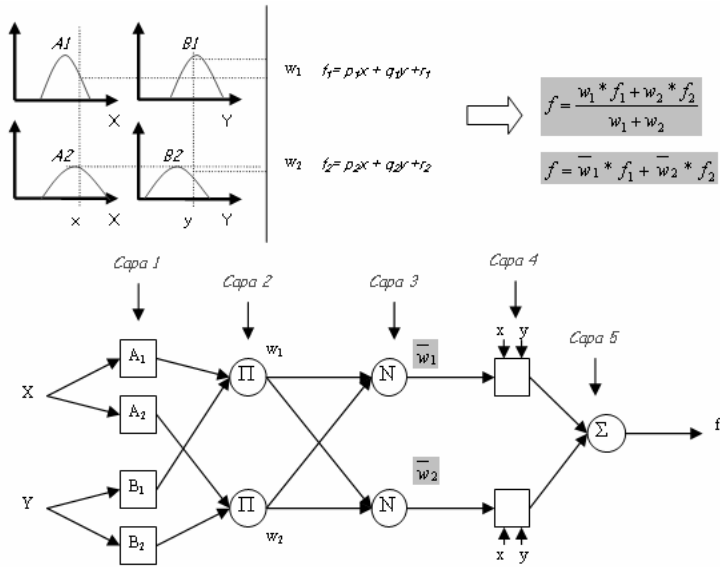


Figura 8: (a) Razonamiento difuso del tipo 3 (Takagi-Sugeno). (b) ANFIS equivalente

Las funciones de cada capa son las siguientes:

Capa 1

Cada nodo i en esta capa ejecuta la siguiente función:

$$O_i^1 = \mu_{A_i}(x) \tag{2.8}$$

Donde x es la entrada del nodo i , A_i es la etiqueta lingüística (pequeño, grande, etc.) asociada con este nodo. Dicho de otra forma, la salida de este nodo nos indica el grado en el cual la variable x satisface la etiqueta A_i , es decir su función grado de pertenencia. Esta función μ_{A_i} puede estar definida mediante un conjunto de parámetros. Si los valores de estos parámetros cambiasen, la forma de μ_{A_i} variaría y por tanto también la traducción numérica de la etiqueta lingüística correspondiente sería diferente. Este conjunto de parámetros variables los llamaremos *parámetros premisa*.

Capa 2

Cada nodo en esta capa está etiquetado con la letra Π . Mediante estos nodos multiplicamos las entradas y enviamos el resultado a la salida, es decir:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i=1,2 \quad (2.9)$$

Esta es uno de los operadores T-norma. Podría decirse que por tanto que en esta capa se efectúa la operación AND generalizada tomando como entradas las salidas de la capa 1. El resultado nos daría una idea del peso de la regla difusa correspondiente.

Capa 3

Cada nodo en esta capa está etiquetado con la letra N. Mediante estos nodos se realiza el siguiente cálculo:

$$\bar{w}_i = \frac{w_1}{w_1 + w_2}, i = 1, 2 \quad (2.10)$$

El resultado nos proporciona el peso normalizado de la regla i-ésima.

Capa 4

Los nodos en esta capa representan la siguiente función:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.11)$$

Donde se ha empleado el resultado de la capa 3 y se ha introducido un conjunto nuevo de parámetros que llamaremos *parámetros consecuencia*.

Capa 5

El nodo de salida, etiquetado con la letra Σ , computa la suma total de todas las señales de entrada, es decir:

$$O^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.12)$$

De esta forma, hemos construido una red adaptativa que es funcionalmente equivalente a un sistema de inferencia difuso de tipo 3. Para sistemas de inferencia del tipo 1, el ANFIS equivalente sería el mostrado en la figura 9.

Los ANFIS tipo 2 son bastante más complicados que los ANFIS tipo 1 y tipo 3, expuestos hasta ahora, y no presentan mejoría en los resultados, según Jang [12], el creador de esta arquitectura. Por ello, preferimos centrarnos en los ya explicados.

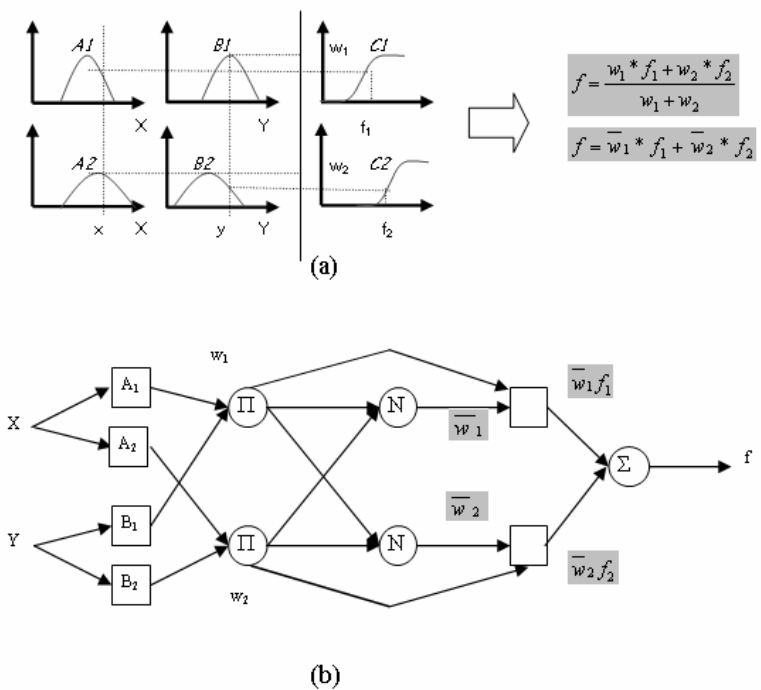


Figura 9: (a) Razonamiento difuso del tipo 1 . (b) ANFIS equivalente

Como hemos visto al comentar las capas 1 y 4, existen dos conjuntos de parámetros distintos, los parámetros premisa y los parámetros consecuencia. Los parámetros premisa de las reglas particionan el espacio de las entradas del sistema en unos subespacios difusos, mientras que los parámetros consecuencia especifican la salida de cada subespacio. Como ejemplo, un ANFIS de 2 entradas x e y, con 3 etiquetas distintas para cada entrada ($A_1, A_2, A_3, B_1, B_2, B_3$), tendría 9 reglas posibles, y por consiguiente 9 subespacios distintos (ver figura 10).

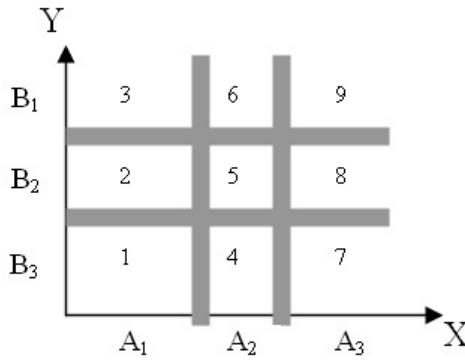


Figura 10: Subespacios difusos creados en un ANFIS de 2 entradas con 3 etiquetas lingüísticas por entrada

Algoritmo de entrenamiento híbrido

Supongamos un ANFIS tipo 3, es decir, que emplee reglas al estilo Takagi-Sugeno. A la vista de la figura 8, se observa que conocidos los parámetros premisa, la salida total puede ser expresada como combinaciones lineales de los parámetros consecuencia, es decir:

$$\begin{aligned}
 f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \\
 &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2
 \end{aligned}
 \tag{2.13}$$

El algoritmo de entrenamiento propuesto por Jang [12] consta de 2 fases. En la primera fase, fijamos los parámetros premisa y calculamos los parámetros consecuencia mediante mínimos cuadrados (LSE). En la segunda fase, se mantienen fijos los parámetros consecuencia calculados en la fase 1, y calculamos los parámetros premisa actualizándolos mediante el método de descenso de gradiente (ver tabla 1).

Entrenamiento HÍBRIDO	Fase 1	Fase 2
Parámetros <i>premise</i>	Fijos	Descenso gradiente
Parámetros <i>consecuencia</i>	Mínimos cuadrados	Fijos
Datos empleados	Salidas de los nodos	Tasa de error

Tabla 1: Pasos del entrenamiento híbrido de una arquitectura ANFIS

Es destacable que el coste computacional de la estimación por mínimos cuadrados es mayor que la actualización de parámetros mediante el descenso de gradiente. De hecho, existen varias combinaciones de estos métodos para actualizar los parámetros, que enunciamos a continuación de acuerdo a su coste computacional:

1. *Descenso del gradiente.* Todos los parámetros son actualizados mediante el descenso de gradiente.
2. *Descenso del gradiente y un paso de LSE.* El LSE se aplica sólo una vez al principio para conseguir los valores iniciales de los

- parámetros consecuencia y a partir de ahí se usaría el método del descenso de gradiente.
3. *Descenso del gradiente y LSE.* Este es el algoritmo de entrenamiento híbrido.
 4. *Sólo LSE.* Para más detalle de esta posibilidad, ver las referencias propuestas en la literatura de redes neuronales [16] ,[17], [18].

Para un ANFIS tipo 1 realizamos la aproximación lineal de la curva del nodo de salida, de forma que tenemos dos parámetros *consecuencia* por recta (ver figura 11).

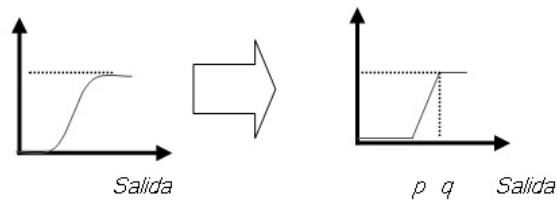


Figura 11: Aproximación lineal de la función de la parte consecuente de un ANFIS tipo 1

Trabajos previos del autor de esta arquitectura [19], [20] y de otros autores como Tommi Ojala [21] demuestran que ANFIS es una arquitectura útil para sistemas de control [19]. Además, Jang [12] presenta el sistema neuro-difuso ANFIS (Adaptive Network based Fuzzy Inference System) como un buen aproximador universal. Para demostrar esta afirmación, Jang simuló funciones no lineales de dos y tres entradas con ANFIS, y fue capaz

de predecir con éxito el comportamiento de la serie caótica resultante de resolver la ecuación diferencial de Mackey-Glass. Cabe destacar, además, la continua aparición de trabajos en donde se aplica ANFIS para fines muy diversos. Recientemente Arellano ha presentado un diseño y realización de un circuito integrado, en tecnología CMOS, que realice las funciones correspondientes a las capas 1 y 2 de la arquitectura ANFIS, para el proceso y clasificación de señales.

En lo relativo al presente trabajo, emplearemos la condición de buen aproximador universal de este sistema con la intención de conseguir una probabilidad de capturas de *Prionace Glauca* en función de parámetros medioambientales.

Resultados y discusión

Estudio previo mediante SOM (Mapas Autoorganizativos)

Vamos a realizar un estudio de las variables de entrada de los datos de capturas usando *Mapas Autoorganizativos (Redes de Kohonen)*. Estos mapas ofrecen como resultado una proyección bidimensional en la segunda capa de neuronas que nos permite, al analizar los pesos de cada neurona, averiguar si existe algún tipo de relación entre las variables iniciales de entrada. Comprobaremos que los resultados obtenidos mediante este método avalan los conseguidos anteriormente mediante el estudio de las componentes principales del mismo conjunto.

Hemos implementado una Red Kohonen con 5 unidades de entrada. Cada una de ellas se corresponde a una variable resultado del trabajo de procesado. En la segunda capa tenemos un conjunto bidimensional de 8*8 neuronas, en donde se proyectarán las clases distintas existentes en los conjuntos de entrada, como se puede apreciar en la figura 12.

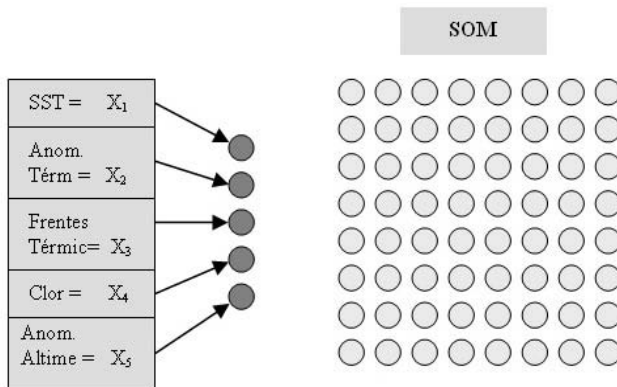


Figura 12: Red de Kohonen propuesta

Para los resultados que presentamos a continuación, hemos usado los siguientes parámetros, durante 2500 ciclos de entrenamiento:

$h(0)=0.999$
 $r(0)=3$
 $mult_H=1$
 $mult_R= 1$

En la figura 13 se muestra la capa bidimensional de 8*8 unidades, las cuales están agrupadas según los pesos ganadores, según la siguiente leyenda:

x
y

x = número correspondiente de la neurona o unidad de procesamiento

y = neurona/s de la capa de entrada ganadora/s, siendo:

1. SST (Sea Surface Temperature).
2. Tendencia Calentamiento-Enfriamiento.
3. Presencia de frente térmico, teniendo en cuenta su fuerza.
4. Concentración de clorofila en mg/mm^3 .
5. Altimetría.

6	7	8	9	10	11	12	13
3	3	3	3	3	1-3	1	1
14	15	16	17	18	19	20	21
3	3	3	3	3	1-3	1	1
22	23	24	25	26	27	28	29
2-3	2-3	2-3	2-3	3	1-3	1-3	1
30	31	32	33	34	35	36	37
2-3	2-3	2-3	2-3	2-3	2-3	1-2	1-4-5
38	39	40	41	42	43	44	45
2-3	2-3	2-3	2-3	2-3	2-3	1-2-3	1-5
46	47	48	49	50	51	52	53
2-3	2-3	2-3	2-3	2-3	2-3	1-2-3	1-2-5
54	55	56	57	58	59	60	61
3	3	2-3	3	3	2-3	1-2-3	1-2-5
62	63	64	65	66	67	68	69
4	4	3-4	3	3	3	2-3	2-5

Figura 13: Resultados del entrenamiento

Puede apreciarse cómo existe una clara relación entre las variables de entrada 2 (Tendencia Calentamiento-Enfriamiento de las aguas) y 3 (Presencia en las cercanías de un frente térmico), debido a que las neuronas amarillas han aprendido a responder conjuntamente a los dos estímulos, como si se tratase de uno solo.

En el Mapa Autoorganizativo existen relaciones menos claras entre las variables 1-3 y 1-2-3, que no aparecían antes en el estudio de componentes principales, debido a la pérdida forzosa de información del conjunto de datos al verse representado en 3 variables en vez de 5. Estas relaciones pueden considerarse lógicas por tratarse de productos derivados del mismo satélite, el NOAA. Además, puede interpretarse que la existencia de un frente durante varios días en la zona implica que o bien se enfría el agua (el frente se hace más intenso), o bien se calienta (el frente va desapareciendo). Por tanto, se llega a la conclusión de que los parámetros Frentes Térmicos y Anomalía Térmica son redundantes.

También existe un agrupamiento de unidades de procesamiento con pesos ganadores de las variables 1-5, es decir, SST- Altimetría, no tan fuerte como las anteriores, y que aparece reflejado en la componente principal número 3.

Resultados del entrenamiento de redes Backpropagation y RBF (Radial Basis Function)

El problema que vamos a analizar consiste en determinar las condiciones óptimas para la realización de un número máximo de capturas, en este caso de *Prionace Glauca*. Como conclusión después del estudio de componentes principales y de las redes de Kohonen (SOM), llegamos a que el conjunto de datos de entrada, formado por 5 parámetros, se podía reducir a 4 (Temperatura, calentamiento-enfriamiento de aguas, clorofila y altimetría). La 5ª variable no aportaba información nueva, por lo que

podemos prescindir de ella. Junto con los datos del libro de pesca tenemos, por tanto, un conjunto de patrones que pueden servir para el entrenamiento de sistemas conexionistas. Recordemos que las redes neuronales se aplican con éxito como herramientas de predicción una vez entrenadas satisfactoriamente [22], [23], [24], [25], [26], [27].

El perceptrón multicapa (MLP) entrenado con el algoritmo de retropropagación del error (backpropagation) y las redes RBF son dos tipos de redes neuronales bien conocidas. Analizamos a continuación su comportamiento ante este particular conjunto de entrenamiento.

Redes Backpropagation

Partiendo de 4 variables de entrada (SST, Calentamiento-Enfriamiento, concentración de clorofila, altimetría) y una salida (capturas de Quenlla), creamos un perceptrón multicapa del cual vamos a estudiar su comportamiento según:

- El número de unidades de su capa oculta.
- El número de patrones del conjunto de entrenamiento y conjunto de test.
- La variación de los parámetros del algoritmo de aprendizaje.

El objetivo es encontrar una red lo más sencilla posible, que nos permita extraer conclusiones sobre las condiciones en las cuales se puede maximizar el número de capturas de la unidad embarcada.

Una vez entrenada la red, puede comprobarse que el número de neuronas de la capa escondida apenas influye en el error que presenta la red. Existen pequeñas diferencias iniciales, que pueden ser debidas a la inicialización aleatorio de los pesos de la red. En todos los casos, el error MSE del conjunto de entrenamiento permanece cercano a 0.01, mientras que el error del conjunto de validación es ligeramente inferior a 0.02.

Ante estos resultados, nos decantaremos para continuar nuestro trabajo con la red más sencilla, es decir, aquella que posee 2 neuronas en su capa escondida. De esta forma podremos interpretar con mayor facilidad la información que nos presente el conjunto de pesos en futuros trabajos.

Comportamiento de la red al variar el número de patrones del conjunto de entrenamiento y el conjunto de validación.

Una vez contruidos 4 conjuntos de entrenamiento y validación distintos, vamos a usarlos para entrenar una red con 2 neuronas en la capa escondida. La cantidad de patrones de cada conjunto es el siguiente:

Conjunto	Nº patrones entrenamiento	Nº patrones validación
1	57	38
2	72	23
3	48	47
4	80	15

Tabla 2: Conjuntos de entrenamiento probados

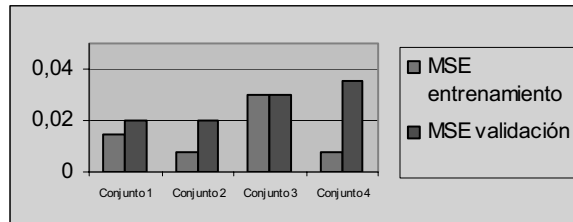


Figura 14: Representación de los errores conseguidos para los conjuntos de entrenamiento de la Tabla 2

Presentaremos los resultados de los entrenamientos en función del conjunto de entrenamiento/validación aplicado, y con los parámetros indicados:

Se puede apreciar que los peores resultados se obtienen con el conjunto 3 de datos. Ello es así por el bajo número de patrones que componen los datos de entrenamiento y como consecuencia de ello el MSE de estos patrones asciende hasta 0.03.

Con el conjunto 4 existe un aumento del error de validación, debido a que la red pierde capacidad de aprendizaje. No hay que olvidar que intentamos diseñar una red que además de tener un error mínimo en el aprendizaje, sea capaz de predecir la salida con la máxima precisión posible, cosa que no ocurre en el conjunto que nos ocupa.

En los casos 1 y 2 se puede observar que el error del conjunto de validación es prácticamente el mismo, por lo que este dato no nos va a servir para decantarnos por uno u otro. Sí que se observa una pequeña

diferencia entre el error de la red al presentar los patrones de entrenamiento, concretamente se puede apreciar cómo disminuye en el conjunto 2.

A la vista de los resultados presentados en las gráficas de error, al variar η el entrenamiento es más o menos rápido. Si nos fijamos en el conjunto 2, con $\eta=1.0$ el error alcanza un valor constante con 4000 ciclos de entrenamiento. En cambio, con $\eta=2.0$ el error constante llega en el ciclo 1000 aproximadamente. Por tanto, los resultados concuerdan con la exposición teórica del método de retropropagación.

Elección de la red

Para acabar, nos queda seleccionar de entre todos los resultados aquél que sea más convincente. Resumiendo lo expuesto en anteriores apartados, elegimos una red de las siguientes características:

1. Dos neuronas en la capa escondida. El número de neuronas en la capa escondida no influye en el resultado. Por tanto escogemos la más sencilla con vistas a la interpretación posterior del conjunto de pesos.
2. Como conjunto de entrenamiento elegimos el 2, es el que tiene un error, tanto de validación como de entrenamiento más aceptable. Además presenta una distribución bastante equilibrada de los diferentes patrones. A partir de este momento emplearemos este conjunto para entrenar los siguientes sistemas conexionistas.

3. Los parámetros del algoritmo de retropropagación propiciarán un entrenamiento más o menos rápido, pero no influirán en un resultado diferente.

Redes RBF (Radial Basis Function)

Vamos a comparar los resultados obtenidos empleando los siguientes métodos de inicialización:

1. Selección de los centroides a partir de los patrones de entrenamiento.
2. Selección de los centroides empleando Redes de Kohonen.

La red empleada para realizar estas pruebas empleaba funciones gaussianas como funciones de activación de las neuronas de la capa oculta. Esta capa estaba formada por una estructura bidimensional de 8x8 neuronas, conectadas todas ellas con las cuatro entradas y con la única salida:

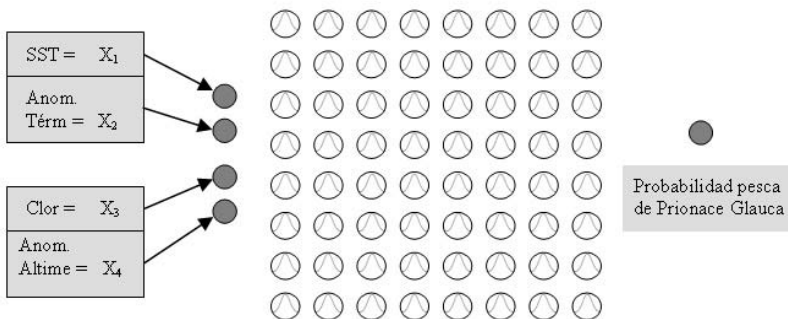


Figura 15: Red RBF diseñada

Por coherencia, hemos escogido las mismas dimensiones de la capa oculta que en el estudio previo realizado mediante mapas autoorganizativos (SOM). Recordemos que uno de los dos métodos de inicialización que hemos escogido emplea redes de Kohonen para colocar los valores de centroides de las funciones gaussianas.

Veamos los resultados obtenidos para los dos tipos de inicialización comentados:

Procedimiento 1 (Selección de los centroides a partir de los patrones de entrenamiento)

Recordemos que este procedimiento selecciona los centros t_j de los patrones de entrenamiento en curso y los asigna a los enlaces entre las capas de entrada y oculta.

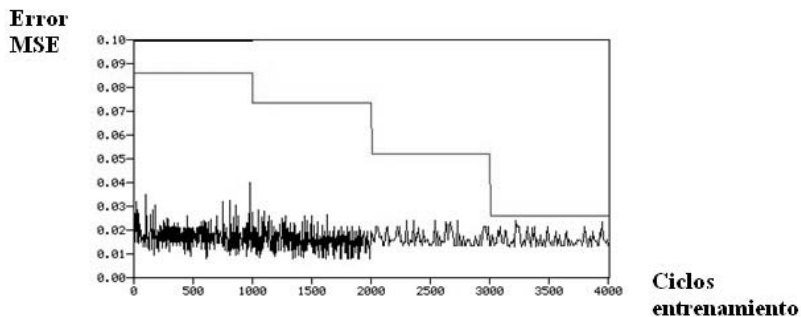


Figura 16: Representación del error MSE en función de los ciclos de entrenamiento para el procedimiento 1 de inicialización. En rojo se observa el error del conjunto de validación

Procedimiento 2 (Redes de Kohonen)

Usando el método autoorganizativo de los mapas característicos de Kohonen, los centros apropiados se generan en base a los patrones de aprendizaje. Como ya comentamos en apartados anteriores, α es la velocidad de aprendizaje. Un valor de 0 deja el vector de centros invariable mientras que un valor de 1 reemplaza el vector seleccionado por el patrón actual de entrenamiento. El parámetro *Shuffle* determina la selección de los vectores de centros iniciales al principio del procedimiento. Al tener el valor 1 tiene lugar una selección aleatoria de los vectores de centros del conjunto de patrones de entrenamiento.

<i>Ciclos de aprendizaje</i>	α	<i>Shuffle</i>
20000	0.5	1

Tabla 3: Parámetros empleados para inicializar con el procedimiento 2

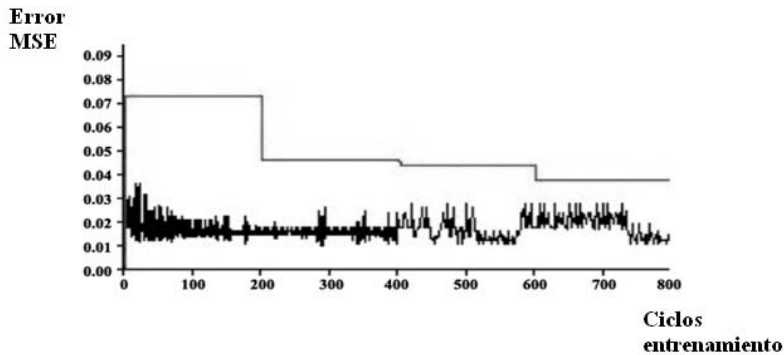


Figura 17: Representación del error MSE en función de los ciclos de entrenamiento para el procedimiento 2 de inicialización. En rojo se observa el error del conjunto de validación

Se observa en las figuras 16 y 17 que los resultados del entrenamiento son similares para los dos procedimientos de inicialización intentados apreciándose cómo el error inicial es más pequeño con el procedimiento 2. Sin embargo, la predicción de los datos de validación es ligeramente mejor para el procedimiento 1. Por ello será esta red la escogida para nuestro sistema.

Resultados obtenidos con las Redes Funcionales

Hemos adaptado el modelo de *separabilidad* explicado en el apartado Redes Funcionales (subapartado relativo al modelo separable) a nuestro problema. La topología de red que proponemos para predecir las capturas de *Prionace Glauca* es la mostrada en la figura 18.

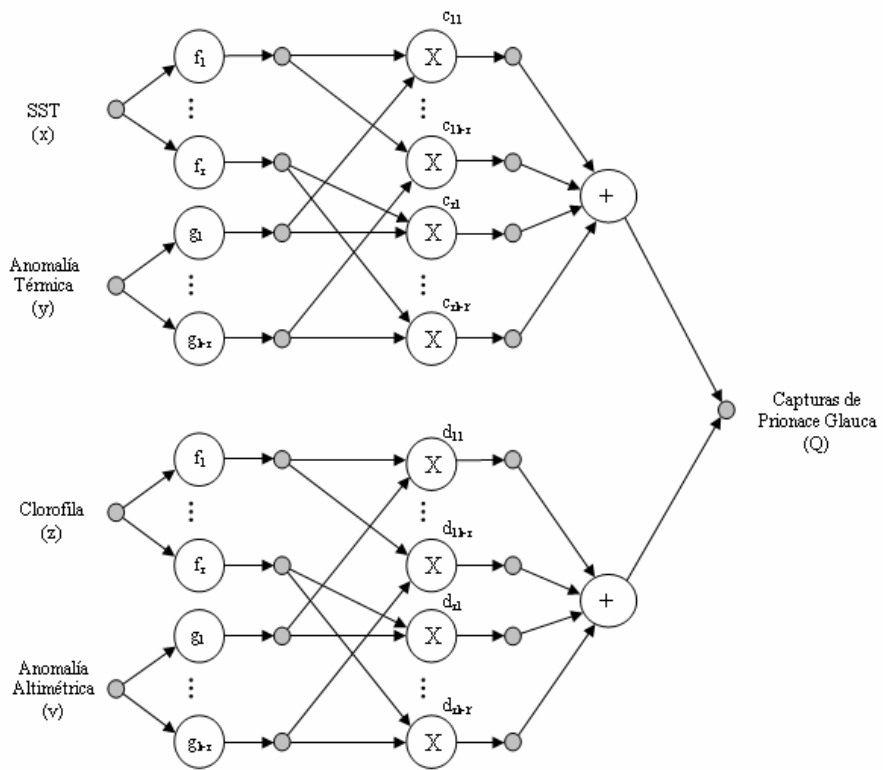


Figura 18: Red funcional basada en el modelo de separabilidad propuesta para predecir capturas

En este modelo se suponen conocidas las dos familias de funciones $\{f_i \mid i=1, \dots, r\}$ y $\{g_j \mid j=1, \dots, k-r\}$. A partir de estas funciones, la salida Q se puede calcular como:

$$Q = \sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x) g_j(y) + \sum_{i=1}^r \sum_{j=1}^{k-r} d_{ij} f_i(z) g_j(v) \quad (3.1)$$

Por tanto, es necesario calcular los coeficientes c_{ij} y d_{ij} adecuados. Para ello disponemos del conjunto de entrenamiento ya usado para las redes neuronales. El proceso de aprendizaje de nuestra red se describe como sigue:

Definimos el error e_k de cada patrón como:

$$e_k = x_{0k} - Q = x_{0k} - \sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x_k) g_j(y_k) + \sum_{i=1}^r \sum_{j=1}^{k-r} d_{ij} f_i(z_k) g_j(v_k) \quad (3.2)$$

El objetivo final de nuestro entrenamiento será minimizar la suma de los errores de todos los patrones, es decir, minimizar E:

$$E = \sum_{k=1}^n e_k^2 ; n = \text{número de patrones del conjunto de entrenamiento} \quad (3.3)$$

Por el método de mínimos cuadrados, el conjunto de parámetros que minimice E debe ser solución del siguiente sistema de ecuaciones:

$$\begin{cases} \frac{\partial E}{\partial c_{pq}} = 2 \sum_{k=1}^n e_k f_p(x_k) g_q(y_k) = 0 \\ \frac{\partial E}{\partial d_{pq}} = 2 \sum_{k=1}^n e_k f_p(z_k) g_q(v_k) = 0 \end{cases} \quad \text{con } p=1, \dots, r; \quad q=1, \dots, r-s$$

(3.4)

Sustituyendo e_k por la expresión (3.2), nos queda el siguiente sistema de ecuaciones:

$$\begin{cases} \frac{\partial E}{\partial c_{pq}} = 2 \sum_{k=1}^n \left(\sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x_k) g_j(y_k) + \sum_{i=1}^r \sum_{j=1}^{k-r} d_{ij} f_i(z_k) g_j(v_k) \right) e_k f_p(x_k) g_q(y_k) = 0 \\ \frac{\partial E}{\partial d_{pq}} = 2 \sum_{k=1}^n \left(\sum_{i=1}^r \sum_{j=1}^{k-r} c_{ij} f_i(x_k) g_j(y_k) + \sum_{i=1}^r \sum_{j=1}^{k-r} d_{ij} f_i(z_k) g_j(v_k) \right) e_k f_p(z_k) g_q(v_k) = 0 \end{cases}$$

con $p=1, \dots, r; \quad q=1, \dots, r-s$

(3.5)

Los patrones de entrenamiento son los mismos que los usados para entrenar las redes neuronales del apartado anterior, para poder así obtener resultados comparables. Recordemos que las redes funcionales están especialmente indicadas para resolver problemas con modelos matemáticos. En nuestro caso no contamos con ningún modelo, por lo cual escogeremos por sencillez la familia polinómica elemental como base para la resolución del sistema.

Se han realizado pruebas con las siguientes familias de funciones:

- Caso 1: $\{f_i\}=\{1,x,x^2,x^3\}$ y $\{g_i\}=\{x,x^2,x^3,x^4\}$.
- Caso 2: $\{f_i\}=\{1,x,x^2,x^3,x^4\}$ y $\{g_i\}=\{x,x^2,x^3,x^4,x^5\}$.
- Caso 3: $\{f_i\}=\{1,x,x^2,x^3,x^4,x^5\}$ y $\{g_i\}=\{x,x^2,x^3,x^4,x^5,x^6\}$.

Es necesario tener una especial precaución a la hora de escoger las familias de funciones, pues podría ocurrir que el determinante de la matriz de coeficientes tuviera dos columnas idénticas. Por ello la familia $\{g_i\}$ no contiene a la función elemental “1”.

Vamos a evaluar a continuación el número de aciertos en los conjuntos de entrenamiento y validación. En la figura 19 se representan los porcentajes. Se comprueba que a priori el mejor aprendizaje corresponde a las funciones del *caso 2*.

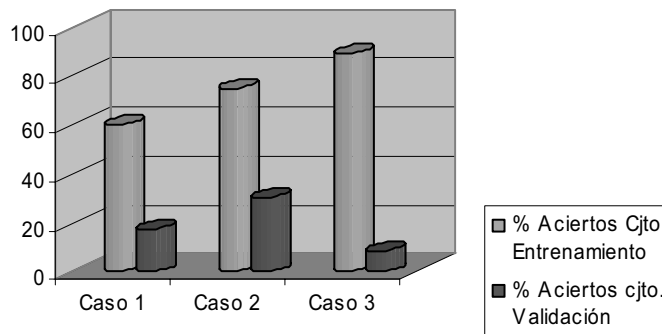


Figura 19: Comparación del % de acierto en los conjuntos de entrenamiento y validación

El estudio del número de aciertos es para nosotros meramente orientativo. Es posible que aún teniendo una elevada tasa de aciertos, el error global sea demasiado alto. Por ello, la medida del grado de aprendizaje de los sistemas estudiados en este trabajo es el MSE (Error cuadrático medio).

En la figura 20 representamos el MSE para los conjuntos de entrenamiento y validación, en cada uno de los tres casos. Podemos comprobar cómo el error MSE de entrenamiento se iguala e incluso mejora respecto al obtenido con las redes neuronales. Estos resultados son esperables, cuanto mayor sea el número de funciones empleadas, mayor será el grado de aproximación del modelo resultante de la red funcional con los datos de entrenamiento. Sin embargo, nosotros buscamos que la red haya extraído el conocimiento necesario para predecir las salidas a partir de las entradas. Para comprobar este hecho, hemos empleado los resultados conseguidos mediante el conjunto de validación.

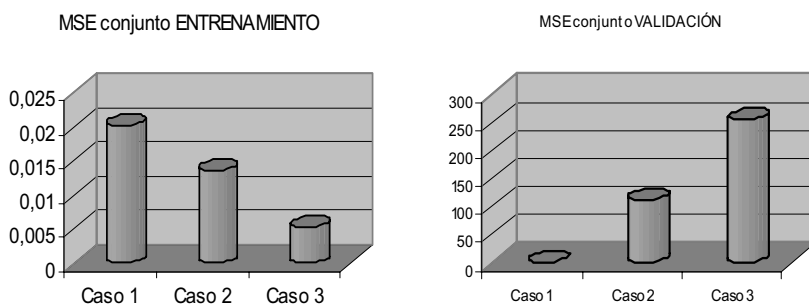


Figura 20: MSE de los conjuntos de entrenamiento y validación

Puede apreciarse que el error MSE menor si analizamos los conjuntos de validación corresponde al *caso 1* (0.404). Para los casos 2 y 3, el MSE se dispara hasta valores poco aceptables. Esto no es incoherente con la figura 19, pues se comprobó que existían grandes diferencias numéricas entre los valores reales y predichos en aquellos casos que no eran considerados como aciertos. Sin embargo, estas diferencias no eran tan importantes si entrenábamos la red funcional con la base de funciones correspondiente al *caso 1*. Podemos comprobar que el error en el conjunto de validación empeora los resultados conseguidos mediante las redes neuronales ya comentadas. El hecho de que no exista un modelo matemático explicativo del problema que podamos emplear explica este hecho. Recordemos que las redes funcionales están especialmente indicadas en aquellos modelos que sí posean este modelo subyacente. Es interesante sin embargo, comprobar la diferencia de comportamiento entre las modernas y las clásicas arquitecturas de aprendizaje.

Resultados obtenidos con ANFIS

Hemos implementado sistemas ANFIS tipo 3 (Takagi-Sugeno) usando como conjuntos de entrenamiento y validación los patrones ya empelados para redes neuronales y funcionales. Hemos buscado cuál es la mejor topología para nuestro sistema. Para las diferentes pruebas se han diseñado 10 ANFIS con las siguientes características:

- *Caso 1*: 2 funciones de grado de pertenencia para cada variable. La salida es de orden 0 (función constante).

- *Caso 2:* 4 funciones de grado de pertenencia para cada variable. La salida es de orden 0 (función constante).
- *Caso 3:* 3 funciones de grado de pertenencia para las variables SST y anomalía altimétrica, y 2 para la clorofila y anomalía térmica. La salida es de orden 0 (función constante).
- *Caso 4:* 2 funciones grado de pertenencia para las variables SST y anomalía altimétrica, y 3 para la clorofila y anomalía térmica. La salida es de orden 0 (función constante).
- *Caso 5:* 3 funciones grado de pertenencia para cada variable. La salida es de orden 0 (función constante).
- *Caso 6:* 2 funciones de membresía (o funciones características) para cada variable. La salida es de orden 1 (función lineal).
- *Caso 7:* 4 funciones de grado de pertenencia para cada variable. La salida es de orden 1 (función lineal).
- *Caso 8:* 3 funciones grado de pertenencia para las variables SST y anomalía altimétrica, y 2 para la clorofila y anomalía térmica. La salida es de orden 1 (función lineal).
- *Caso 9:* 2 funciones grado de pertenencia para las variables SST y anomalía altimétrica, y 3 para la clorofila y anomalía térmica. La salida es de orden 1 (función lineal).
- *Caso 10:* 3 funciones de grado de pertenencia para cada variable. La salida es de orden 1 (función lineal).

En todos los casos, los sistemas desarrollados para estos apartados tienen las siguientes características comunes:

- Regla AND empleada = producto.

- Regla OR empleada = OR probabilístico = $a + b - ab$.
- Método *desfuzzificación* => suma media ponderada.
- Método entrenamiento híbrido ya explicado en el apartado ANFIS (subapartado relativo al algoritmo de entrenamiento híbrido).
- Funciones grado de pertenencia del tipo gaussiano gaussianas.

El entrenamiento se ha llevado a cabo hasta que el error de los patrones de entrenamiento se ha mantenido constante. De forma paralela, se ha llevado el control del error en el conjunto de validación, así como de la tasa de aciertos y fallos conseguida. Recordemos que consideramos que la predicción de nuestro sistema es un *acierto* si difiere menos de un 10% del valor máximo de capturas. Como hemos normalizado todos los datos de capturas, los valores están en el intervalo [0,1]. Por tanto, consideraremos que la predicción de nuestro sistema es un *acierto* si difiere menos de 0.1 respecto al valor real de capturas del barco.

Siguiendo este criterio, hemos obtenido las siguientes tasas de aciertos y fallos en el conjunto de entrenamiento (ver figura 21).

En la figura 21 puede observarse que la tasa de aciertos después del entrenamiento es prometedora en varios casos (2, 7, 8, 9 y 10). Sólo en el caso 1 el número de fallos es equiparable al de aciertos. Ésta es sólo una medida orientativa, para conocer la capacidad de aprendizaje de nuestro sistema es necesario estudiar los datos del conjunto de validación y sobre todo, los respectivos MSE (errores cuadráticos medios).

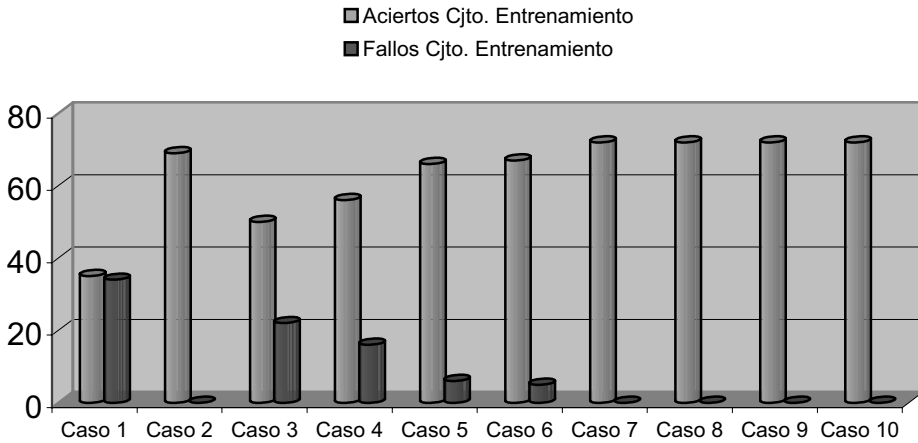


Figura 21: Comparación de la tasa de fallos y la tasa de aciertos en el conjunto de entrenamiento

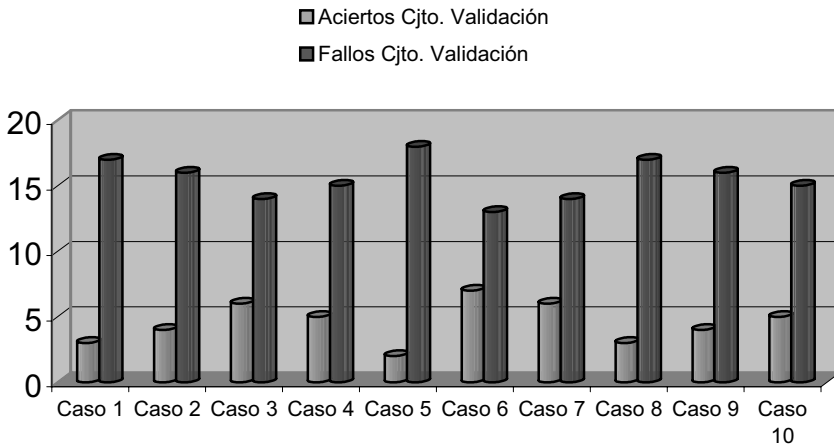


Figura 22: Comparación de la tasa de fallos y la tasa de aciertos en el conjunto de validación

Los resultados empeoran cuando estudiamos el conjunto formado por los patrones de validación. Podemos apreciar en la figura 22 cómo en ningún caso el número de aciertos es superior al número de fallos en la predicción.

Para ver la desviación global del conjunto de predicciones respecto a su valor real, veamos ahora el error MSE. En la figura 23 podemos comprobar cómo el error es comparable o menor respecto a los sistemas probados con anterioridad. El mayor error que se encuentra corresponde al *caso 1*. Su magnitud es del mismo orden que el error de las redes neuronales. Podemos ver también cómo a medida que el sistema aumenta de complejidad, posee más capacidad para ajustar los parámetros de entrenamiento de tal forma que el número de fallos llega a ser mínimo.

MSE Entrenamiento

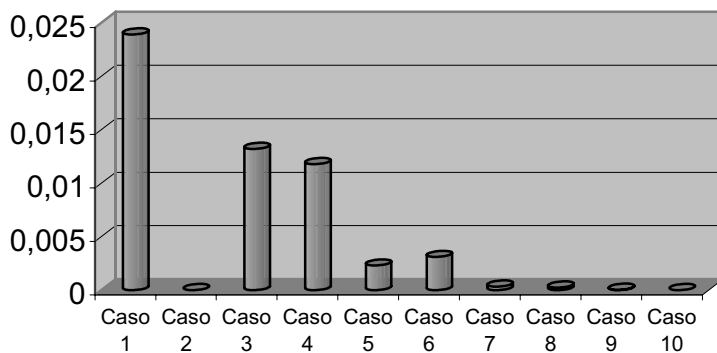


Figura 23: Comparación del error cuadrático medio (MSE) en el conjunto de entrenamiento. En todos los casos se igualan o mejoran los resultados de las redes neuronales

Para testear la capacidad de aprendizaje de estos sistemas y compararla con las redes antes probadas, es necesario comprobar el comportamiento del conjunto de validación. Un sistema con una buena capacidad de generalización poseerá un buen error MSE en el conjunto de validación y un MSE aceptable para los patrones de entrenamiento.

MSE Validación

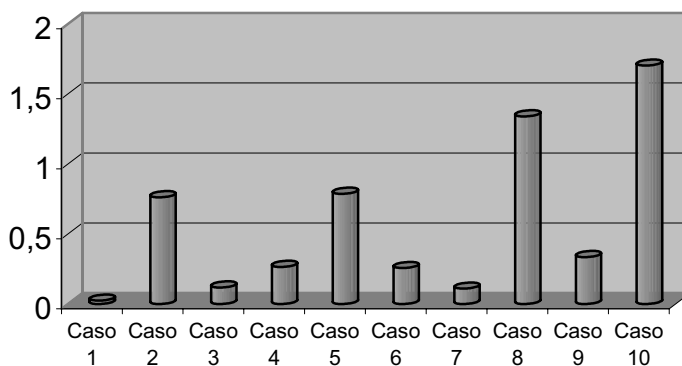


Figura 24: Comparación del error cuadrático medio (MSE) en el conjunto de validación. Sólo en el caso 1 el MSE es del mismo orden que el MSE conseguido con redes neuronales

A la vista de la figura 24, podemos comprobar que el único de los sistemas ANFIS con un error comparable al conseguido con las redes neuronales es el *caso 1*, en el resto de los casos el error se dispara. Ello demuestra que no por conseguir una tasa de aciertos alta o un error MSE con el conjunto de entrenamiento, se consigue un buen aprendizaje. Como comentaremos en el apartado siguiente, es un claro caso de sobreentrenamiento.

Como resumen de todos estos resultados, podemos concluir que el sistema que presenta una mejor capacidad de generalización es el correspondiente al *caso 1*. Vamos ahora a concretar cómo es su topología.

En la figura 25 podemos observar las características y distintas capas de nuestro sistema ANFIS tipo 3 de orden 0, con dos funciones características por cada variable de entrada. Comparado con los otros sistemas estudiados presenta sencillez y una capacidad mayor de aprendizaje.

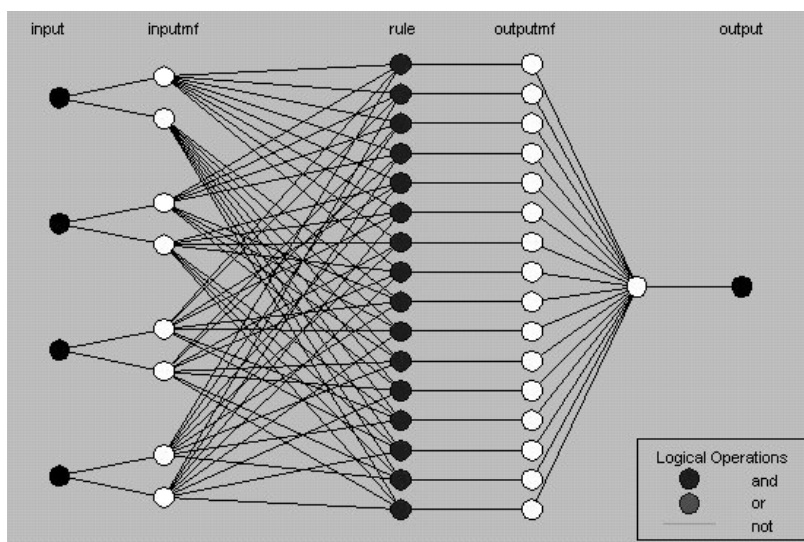


Figura 25: Topología correspondiente al sistema ANFIS del caso 1.

Podemos comprobar cómo por cada variable de entrada existen sólo dos funciones características (funciones membresía, MF)

Veamos cómo se ejecutan las reglas para llegar a calcular el dato de salida:

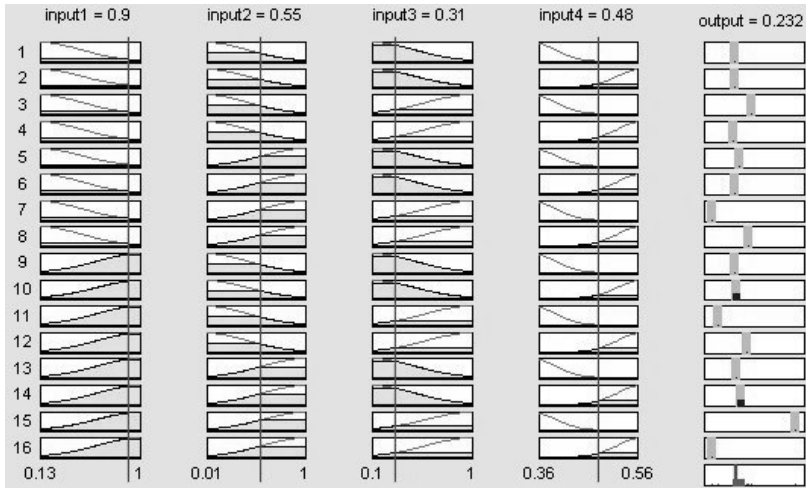


Figura 26: Comportamiento del sistema al presentarle el patrón formado por las entradas SST=0.9, Anomalia térmica=0.55, Clorofila=0.31, Anomalia altimétrica=0.48. La predicción del sistema es 0.232. La salida real es 0.22, por tanto puede considerarse un acierto del mismo

En la figura 26 podemos ver cómo cada regla da como resultado una salida. El ANFIS propuesto, al disponer de 4 variables de entrada con dos funciones grado de pertenencia cada una, puede formar un total de 16 reglas distintas. El resultado después de *fuzzificar* las entradas y realizar las operaciones correspondientes, se *desfuzzifica* mediante una suma ponderada media, conforme a lo explicado en el apartado ANFIS.

Ahora bien, como resultado del aprendizaje, los parámetros propios de las funciones características (funciones grado de pertenencia) gaussianas han variado. Veamos una representación gráfica de cómo quedan finalmente:

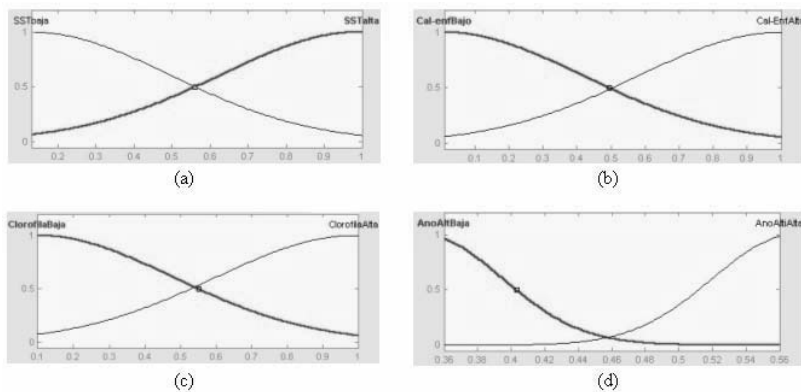


Figura 27: Funciones grado de pertenencia después del entrenamiento. (a) SST, (b) Anomalia Térmica, (c) Concentración de clorofila, (d) Anomalia altimétrica

Puede comprobarse en la figura 27d cómo al haber cambiado los parámetros de las gaussianas después del entrenamiento varía la forma de las dos funciones. Los cambios de las restantes funciones gaussianas son menos importantes.

Discusión de los resultados de los algoritmos

El problema del sobreentrenamiento

Para que el algoritmo estudiado sea un buen predictor, hemos de conseguir un error bajo para el conjunto de entrenamiento y el de validación. Podría darse el caso de que el error de validación sea mucho mayor que el error de entrenamiento, entonces se produce el problema de sobreajuste o sobreentrenamiento. En estadística es bien conocido que cuando se utiliza un modelo con muchos parámetros para ajustar un conjunto de datos procedente de un proceso con pocos grados de libertad, el modelo obtenido puede no descubrir las tendencias reales del proceso original, aunque pueda presentar un error pequeño de ajuste de datos. En este caso, el entrenamiento se reduce a la interpolación de los datos, incluyendo el ruido, utilizando una función sigmoïdal complicada.

Para ilustrar este problema, considérese el siguiente ejemplo. La curva mostrada en la figura 28a pasa exactamente por los puntos de entrenamiento (representados por puntos negros). Se trata de un polinomio interpolador de grado siete de los ocho puntos dados, por tanto es un modelo de error cero. No obstante, si se considera un conjunto alternativo de datos de comprobación del modelo, entonces la curva anterior no puede predecir con exactitud los nuevos datos, como es el caso de la figura 28b. El problema es el excesivo número de parámetros (los ocho coeficientes del polinomio) utilizados. Un modelo más realista pudiera ser obtenido considerando un pequeño número de parámetros (por ejemplo, el polinomio

de tercer grado de la figura 28c). En este caso, el error de aproximación es similar para los conjuntos de datos de entrenamiento y de validación, indicando que se han capturado las tendencias reales del modelo. La diferencia entre ambos modelos puede verse en la figura 28d.

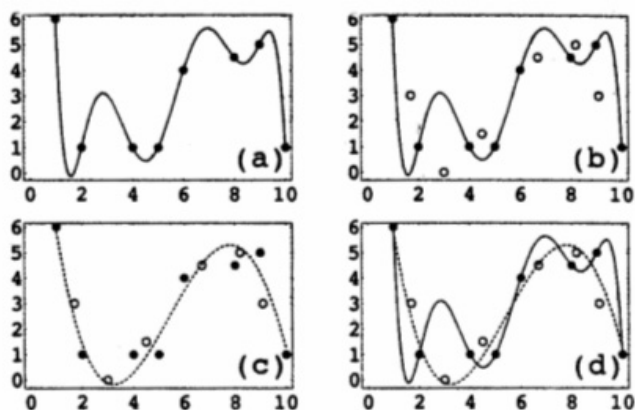


Figura 28: Un ejemplo del problema del sobreentrenamiento. (a) Modelo con demasiados parámetros para ajustar los datos. (b) Se ha obtenido un gran error en los datos de comprobación. (c) Un modelo con menos parámetros aproxima de igual forma los datos de entrenamiento y los de validación. (d) Se comparan ambos modelos. (Adaptado de [11])

Al analizar las figuras 20, 23 y 24 podemos apreciar cómo este problema aparece de manera recurrente en los resultados expuestos hasta ahora. En la figura 20 representábamos el MSE para los conjuntos de entrenamiento y validación para la red funcional con 3 bases de funciones distintas. Cuando el MSE de entrenamiento disminuye, vemos que el MSE

de validación crece de manera desmedida. Sólo para el caso 1 ambos errores alcanzan cotas interesantes simultáneamente, coincidiendo precisamente con la base más sencilla, es decir, aquella compuesta por un menor número de funciones y con menos parámetros que ajustar. En las figuras 23 y 24 hemos realizado un estudio similar varias estructuras ANFIS. Igual que en las redes funcionales, al disminuir el MSE de los patrones de entrenamiento (figura 23), aumenta el MSE de los patrones de validación en gran medida (figura 24). Sólo en el caso 1 se consiguen errores comparables con las redes neuronales, estimando los patrones del conjunto de entrenamiento y validación. El caso 1 vuelve a ser el que posee una estructura más sencilla y un menor número de parámetros.

Resultados obtenidos

Hemos escogido para la comparación los mejores resultados después de entrenar los diferentes casos de redes neuronales, funcionales y ANFIS. Para elegirlos, se han evitado todos aquellos casos en los cuales existía el problema de sobreajuste.

Para decidirnos por alguno de los cuatro algoritmos probados, una primera estimación se puede conseguir estudiando el número de aciertos y fallos. En la figura 29 podemos ver la tasa de aciertos y fallos en los conjuntos de entrenamiento y validación.

A la vista de la figura 29, los mejores resultados se corresponden con el Perceptrón Multicapa, entrenado con el algoritmo de retropropagación del error (backpropagation). Sin embargo, para estar seguros, es necesario

comprobar el error MSE de los conjuntos de entrenamiento y validación (ver figura 30).

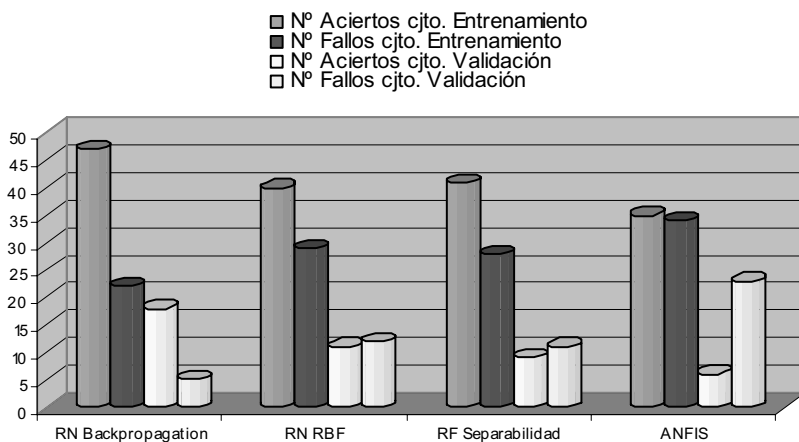


Figura 29: Comparativa de fallos y aciertos en los conjuntos de entrenamiento y validación para los 4 algoritmos

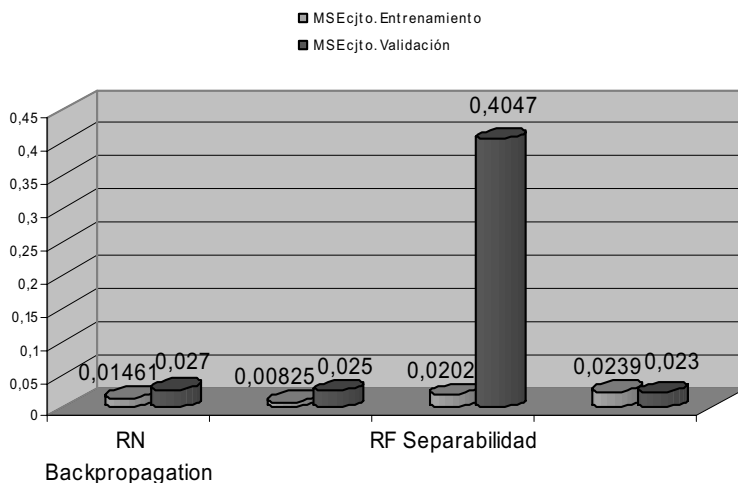


Figura 30: Comparación del error cuadrático medio (MSE) de los distintos algoritmos para los conjuntos de entrenamiento y validación

Los errores representados en la figura 30 son todos del mismo orden exceptuando el error de validación de la red funcional. Ello quiere decir que la capacidad de generalización de este algoritmo para nuestro problema es menor, probablemente debido a la no existencia de un modelo matemático capaz de explicar el problema.

Finalmente, hemos escogido para la implementación el perceptrón multicapa entrenado con algoritmo Backpropagation (retropropagación de error) por su sencillez frente al ANFIS y la red RBF.

Generación de mapas de probabilidad de pesca

En el sistema propuesto hasta ahora, los datos de partida son imágenes de los satélites OrbView-2, la serie NOAA y Topex-Poseidón. Como ya hemos comentado en apartados anteriores, los productos derivados del procesado son mapas de:

- Temperatura superficial (SST).
- Anomalía térmica semanal.
- Frentes térmicos.
- Concentración de clorofila
- Anomalía altimétrica

Justificamos en apartados anteriores (*Resultados y Discusión* - subapartado relativo al *Estudio previo mediante SOM*) que la información aportada por la anomalía térmica semanal y los frentes térmicos era redundante. Por tanto suprimimos los frentes térmicos como entrada de

nuestro sistema. Con lo cual disponemos de un mayor número de patrones de entrenamiento con valores válidos.

En la figura 31 podemos ver cuatro imágenes derivadas del procesado correspondientes al 7/agosto/1998.

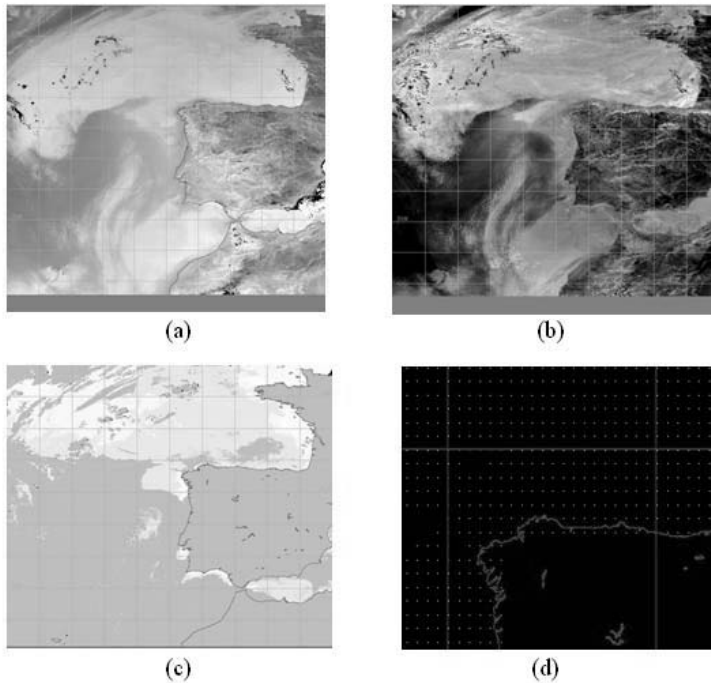


Figura 31: Imágenes obtenidas después del procesado inicial correspondientes al 7-8-1998. (a) Temperatura superficial (SST, serie NOAA). (b) Anomalía térmica semanal (serie NOAA). (c) Concentración de clorofila (Orbview-2). (d) Mapa de anomalía altimétrica (Topex-Poseidon)

Las figuras 31a y 31b fueron obtenidas a partir de la serie NOAA. Las imágenes de las figuras 31c y 31d se procesaron a partir de datos de Orbview-2 y Topex-Poseidon, respectivamente. Como ya indicamos antes, el formato de datos empleado por la serie NOAA es el *tdf*, el de Orbview-2 el *hdf*, mientras que las imágenes de Topex-Poseidon están en binario. Los lenguajes de programación empleados para tratar las imágenes fueron C en el caso de imágenes NOAA, e IDL 5.2 para imágenes Orbiew-2. Creemos conveniente unificar el formato de las imágenes como paso previo a la aplicación de la red neuronal, por ello hemos pasado las imágenes *hdf* y *binarias* a formato *tdf*, para luego tratarlas con código generado en lenguaje C. Hemos escogido el formato *tdf* por disponer de un conjunto versátil de funciones especialmente indicadas para tratamiento de imágenes en el citado formato. Es importante ajustar la corrección geográfica de las imágenes para que exista una correspondencia píxel a píxel entre ellas, y facilitar la aplicación del algoritmo correspondiente. En la figura 32, mostramos un ejemplo de una imagen mal corregida, puede apreciarse cómo la línea de costa no se ajusta a la situación real.

Para evitar este problema, hemos corregido las imágenes NOAA y OrbView-2 mediante las correcciones rectangular y cilíndrica equidistante, respectivamente. Ambas correcciones geográficas son equivalentes píxel a píxel si poseen la misma resolución espacial. Para la imagen de Topex-Poseidon, hemos calculado computacionalmente las coordenadas geográficas de cada punto para colocarlas después sobre una imagen inicialmente sin datos. Podemos verificar en la figura 31 que la corrección geográfica de las cuatro imágenes es correcta. Cabe destacar la diferente

resolución espacial de la imagen figura 31d respecto de sus compañeras, de ahí los puntos aislados y la ausencia de datos (el valor 0 se representa en negro). En efecto, la resolución espacial de esta imagen es de $0.25^\circ \times 0.25^\circ$, mientras que en el resto posee una resolución de $0.025^\circ \times 0.025^\circ$.

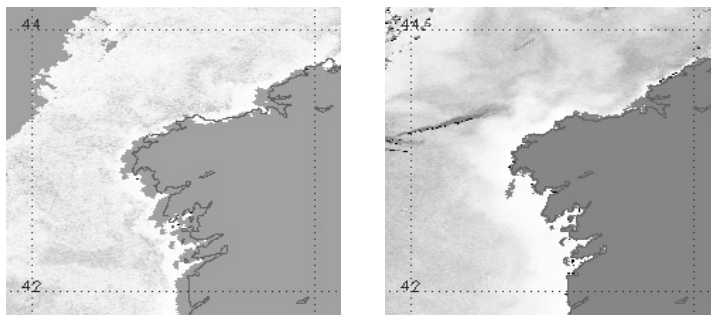


Figura 32: A la izquierda, imagen de clorofila procedente del OrbView-2 con errores de georreferenciación tras haberla proyectado a una proyección cilíndrica; a la derecha, imagen de OrBiew-2 georreferenciada correctamente

Estamos en condiciones de aplicar la red Backpropagation para calcular el mapa de probabilidad de obtener capturas del *Prionace Glauca*. Las entradas de la red serán los valores de los píxeles de cada imagen. Para la entrada correspondiente a la anomalía altimétrica tomaremos como dato el valor no nulo más cercano al píxel, si es que no existe ya. Con la salida de la red generamos un mapa de probabilidad de pesca (ver figura 33). Con niveles de gris altos (colores claros) tenemos una probabilidad alta de pesca.

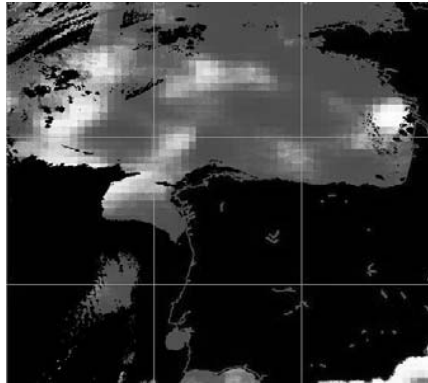


Figura 33: Mapa de probabilidad de capturas de Prionace Glauca correspondiente al 7-8-1998 conseguido mediante una red neuronal entrenada con el algoritmo Backpropagation. Con niveles de gris altos (colores claros) tenemos una probabilidad alta de pesca

Debido a la baja resolución espacial de los mapas MSLA procedentes de T/P, en el mapa de probabilidad se observa nítidamente que las probabilidades están agrupadas en cuadrículas de $0.25^\circ \times 0.25^\circ$ (ver figura 34).

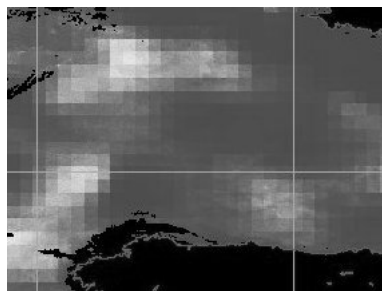


Figura 34: Detalle del mapa de probabilidad de la figura 33

El resultado mostrado en las figuras 33 y 34 está en formato *tdf*, con la misma corrección geográfica que las imágenes de partida. Si la transformamos al conocido formato JPEG, estamos en disposición de importarla a un SIG (Sistema de Información Geográfico). A modo de ejemplo, observemos el resultado conseguido empleando el ArcView 3.2 (figura 35):

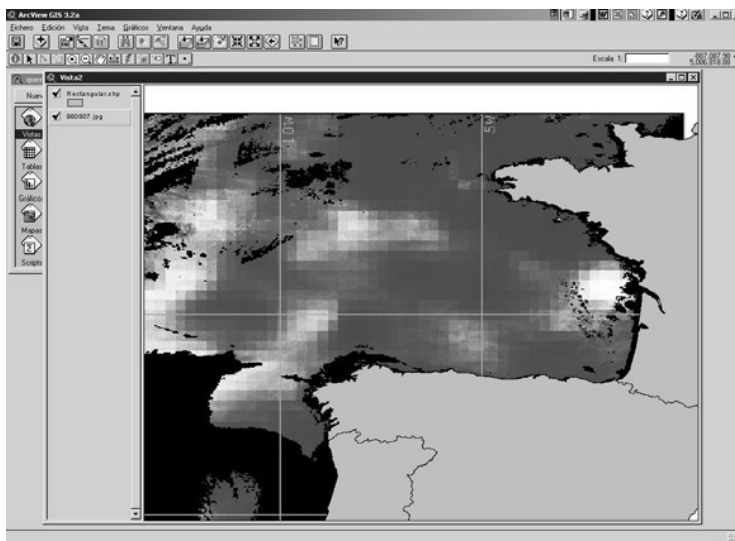


Figura 35: Resultado de exportar la imagen de la figura 32 a ArcView 3.2

Bibliografía

- [1] Consellería de Pesca, Marisqueo e Acuicultura, 1998. Xunta de Galicia. [http://www.xunta.es/conselle/pe/sector_emplo.htm].
- [2] Cotos, J.M., 1994. “*Dinámica y clasificación de estructuras oceánicas para aplicación operacional de pesquerías utilizando teledetección e ingeniería de conocimiento*”. Tesis doctoral. Departamento de Física Aplicada, Facultad de Física, Universidad de Santiago de Compostela.
- [3] Triñanes, J.A., 1998. “*Sistema de información basado en teledetección para ayuda a la explotación operacional de pesquerías de túnidos y otras especies pelágicas*”. Tesis doctoral. Departamento Electrónica e Computación, Universidad de Santiago.
- [4] Wells, W.M., 1986. “*Efficient Synthesis of Gaussian Filters by Cascaded Uniform Filters*”. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9 (2).
- [5] Harlow, C.A., Trivedi, M.M., Connors, R.W., 1986. “*Use of texture operators in segmentation*”. Optical Engineering, vol. 25 (11), pp. 1200-1206.
- [6] Lillesand, T.M., Kiefer, R.N., 1987. “*Remote sensing and image interpretation*”. Ed. John Wiley&Sons.

- [7] Aronoff, S., 1991. "*Geographic Information Systems: A Management Perspective*". 2ª Edición. Ottawa (Ontario), Canadá, WDL Publications. ISBN: 0-921804-91-1.
- [8] Specht, D.F., 1990. "*Probabilistic Neural Networks*". Neural Networks, vol. 3, pp. 109-118.
- [9] Zhang, Q., Benveniste, A., 1992. "*Wavelet Neural Networks*". IEEE Transactions on Neural Networks, vol. 3, pp. 889-898.
- [10] Castillo, E., Gutiérrez, J.M., 1998. "*Nonlinear Time Series Modeling and Prediction Using Functional Networks. Extracting Information Masked by Chaos*". Physics Letters A, vol. 244, pp. 71-84.
- [11] Castillo, E., Cobo, A., Gutiérrez, J.M., Pruneda, E., 1999. "*Introduction to Functional Networks with Applications. A Neural Based Paradigm*". Kluwer International Publishers.
- [12] Jang, J.-S.R., 1993. "*ANFIS: adaptive-network-based fuzzy inference system*". IEEE Trans. Syst., Man., Cybern, vol. 23 (5), pp. 665-685.
- [13] Berenji, H.R., 1992. "*Fuzzy logic controllers*". In An Introduction to Fuzzy Logic Applications in Intelligent Systems, R.R. Yager and L.A. Zadeh, Eds., Kluwer academic publishers, pp. 69-96.

- [14] Takagi, H., Suzuki, N., Koda, T., Kojima, Y., 1992. “*Neural networks designed on approximate reasoning architecture and their applications*”. IEEE Trans. Neural Networks, vol. 3 (5), pp. 752-760.
- [15] Takagi, T., Sugeno, M., 1983. “*Derivation of fuzzy control rules from human operator`s control actions*”. Proc. of the IFAC Symp. On Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55-60.
- [16] Shah, S., Palmieri, F., 1990. “*MEKA-A fast, local algorithm for training feedforward neural networks*”. In Proc. of International Joint Conference on Neural Networks, vol. 3, pp. 41-46.
- [17] Singhal, S., Wu, L., 1989. “*Training multilayer perceptrons with the extended Kalman algorithm*”. In David S. Touretzky, editor, Advances in neural information processing systems I, Morgan Kauffman Publishers, pp. 133-140.
- [18] Jang, J.-S.R., 1991. “*Fuzzy modelling using generalized neural networks and Kalman filter algorithm*”. In Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 762-767.
- [19] Jang, J.-S.R., 1992. “*Self-learning fuzzy controller based on temporal backpropagation*”. IEEE Trans. On Neural Networks, vol. 3 (5), pp. 714-723.

- [20] Jang, J.-S.R., 1992. “*Neuro-Fuzzy Modeling: Architecture, Analyses and Applications*”. Doctoral thesis, University of California, Department of Electrical Engineering and Computer Science.
- [21] Ojala, T., 1994. “*Neuro-fuzzy systems in control*”. Master of science thesis. Tampere University of Technology, Department of Electrical Engineering.
- [22] Komatsu, T., Aoki, I., Mitani, I., Ishii, T., 1994. “*Prediction of the Catch of Japanese Sardine Larvae in Sagami Bay Using a Neural Network*”. Fisheries Science, vol. 60 (4), pp. 385-391.
- [23] Aurelle, D., Lek, S., Giraudel, J., Berrebi, P., 1999. “*Microsatellites and artificial neural networks: tools for the discrimination between natural and hatchery brown trout (Salmo trutta, L.) in Atlantic populations*”. Ecological Modelling, vol. 120, pp. 313-324.
- [24] Dreyfus-Leon, M.J., 1999. “*Individual-based modelling of fishermen search behaviour with neural networks and reinforcement learning*”. Ecological Modelling, vol. 120, pp. 287-297.
- [25] Aussem, A., Hill, D., 2000. “*Neural-network metamodelling for the prediction of Caulerpa taxifolia development in the Mediterranean sea*”. Neurocomputing, vol. 30, pp. 71-78.

- [26] Brosse, S., Guegan, J., Tourenq, J., Lek, S., 1999. “*The use of artificial neural network to assess fish abundance and spacial occupancy in the litoral zone of a mesotrophic lake*”. *Ecological Modelling*, vol. 120, pp. 299-311.
- [27] Maas, O., Boulanger, J., Thiria, S., 2000. “*Use of neural networks for predictions using time series: Illustration with the El Niño Southern oscillation phenomenon*”. *Neurocomputing*, vol. 30, pp. 53-58.

Un Modo Inteligente de Gestionar el Conocimiento

Javier Andrade, Santiago Rodríguez y Sonia Suárez

Depto de Tecnologías de la Información y las Comunicaciones

Universidade da Coruña

Email: {jag, santi, ssuarez}@udc.es

Introducción

Con el fin de hacer frente a las nuevas necesidades impuestas en la actual Era del Conocimiento, las organizaciones deberán de adaptar sus estructuras organizativas. Actualmente, éstas poseen una estructura de “orden y mando” en la que el conocimiento reside en la parte superior de la pirámide organizativa y los órdenes se transmiten a través de un número excesivo de niveles intermedios.

A este respecto, P. F. Drucker considera que las organizaciones del futuro, para poder adaptarse rápidamente a los cambios del entorno, requerirán de una estructura más plana, con menos burocracia y compuesta por especialistas [1]. Actualmente, los mandos superiores de la jerarquía indican a sus subordinados qué tienen que hacer y cómo han de hacerlo. Sin embargo, las organizaciones del futuro estarán formadas por “trabajadores del conocimiento”, es decir, especialistas en su área de trabajo que no precisarán que nadie les diga cómo tienen que hacer su labor. De hecho,

este autor compara el funcionamiento de las organizaciones del futuro con el funcionamiento de una orquesta. Una orquesta está formada por un gran número de especialistas, dirigidos por un único director, en dónde cada músico se guía por una partitura para realizar su trabajo. En este caso, el especialista no necesita que el director le diga cómo ha de tocar su instrumento. Así pues, las organizaciones del futuro serán organizaciones más planas formadas por grupos de especialistas que se guíen por un conjunto de objetivos claros y sencillos. En este tipo de organizaciones el conocimiento residirá en la parte baja de la pirámide jerárquica, en la mente de los especialistas que realizarán diversos trabajos y se dirigirán a sí mismos.

Por otro lado, parece claro que las ventajas competitivas de toda organización dependerán del conocimiento que ésta posea. Para ser más exactos, dependen de qué sabe, cómo lo usa y cómo de rápido aprende algo nuevo. Sin embargo, el conocimiento siempre ha sido un elemento clave para las organizaciones [2] [3], aunque su gestión no se hiciese de forma explícita.

Hoy en día las cosas han cambiado, tal y como queda patente con los cambios que conlleva la entrada en la nueva Era del Conocimiento, y aquellas empresas que no gestionen explícitamente su conocimiento están poniéndose en peligro [4]. Precisamente, la Gestión del Conocimiento (en adelante GC) nace con la idea de gestionar, de manera explícita, el conocimiento que hay alrededor de toda organización.

Principales ventajas de la gestión del conocimiento

La implantación de un programa de GC proporciona un uso más racional del conocimiento que existe en una organización, lo que conlleva innumerables ventajas, entre las que destacan:

- Creación de un entorno que facilita la innovación y el desarrollo de productos de una forma más rápida. Zara, del grupo Inditex, es un ejemplo claro de cómo la GC permite innovar y abrir nuevas posibilidades de mercado [5]. Gracias a una correcta gestión del conocimiento, Zara descubrió cuales eran las necesidades de sus clientes potenciales y supo sacarles provecho. Cada cambio de moda supone una nueva oportunidad para esta empresa, pues es capaz de adaptarse a esos cambios antes que nadie. De hecho, Zara es capaz de poner sus diseños desde el tablero de dibujo al expositor en un plazo que oscila entre 10 y 15 días, lo que le confiere una posición de privilegio respecto al resto de compañías textiles.
- Ahorro de tiempo y recursos al evitar “reinventar la rueda” una y otra vez. Una empresa que no es consciente de lo que sabe, está condenada a resolver, una y otra vez, los mismos problemas. Explicitando el conocimiento organizativo, se evita dedicar recursos a problemas ya resueltos. Un ejemplo de esta situación es la empresa Hewlett-Packard, más conocida como HP, la cual desarrolló un sistema de GC denominado HP Network News para,

entre otras cosas, atender las demandas de comerciantes acerca de sus productos [6]. Gracias a este sistema, HP logró disminuir el número de llamadas telefónicas al servicio técnico, requiriendo menos personal para atender, una y otra vez, a las mismas llamadas.

- Localización de quién, qué y dónde está la experiencia y el conocimiento. Esto permite conocer qué sabe la organización y a quién se debe recurrir para solucionar un determinado problema. La compañía British Petroleum (BP) adoptó, a finales de 1994, un sistema de GC que le permitía poner en contacto grupos de trabajo con expertos. El proyecto se denominó Virtual TeamWork [7] y permitió reducir costes significativos a la organización gracias a que se conseguían poner en contacto, en el momento en el que se necesitase, un equipo perforador con un especialista en la materia. BP había conseguido realizar unas páginas amarillas de quién sabía qué en la organización.
- Como consecuencia del punto anterior, prevención de islas de conocimiento para determinar qué carencias de conocimiento existen en la organización. Un ejemplo de esta situación son los laboratorios farmacéuticos Hoffmann-La Roche [8]. Estos laboratorios habían estimado que, por cada día que se adelantase la salida al mercado de un medicamento, la compañía tendría un beneficio de un millón de dólares. Por ello, centraron sus esfuerzos en intentar acortar dicho periodo que, en la industria farmacéutica, ronda los 8 años y tiene un coste medio de 250 millones de dólares. Concretamente, centraron los esfuerzos en acortar el periodo de

tiempo que requieren los organismos competentes para evaluar si un medicamento es apto, o no, para su comercialización. Tras el correspondiente análisis, se dieron cuenta de que ese tiempo variaba, de forma significativa, en función de la información que se le proporcionase a los organismos evaluadores. Detectada la isla de información, se modificó el tipo de informe que se presentaba a los organismos evaluadores de forma que, al facilitar a estos su labor, también se acortaba el intervalo necesario para la salida al mercado del producto.

- Aumento del nivel de cooperación y transmisión de información, mejorando la comunicación interna. Al explicitar quién sabe qué, se establecen vías de comunicación entre generadores y consumidores de conocimiento. El ejemplo anterior de BP es claro: gracias a que se han identificado los expertos, se sabe a quien acudir en caso de necesidad.
- Aumento del grado de satisfacción de los clientes al conseguir productos y servicios que se adecuen, en mayor grado, a sus necesidades. Un ejemplo de esta situación es el caso de los laboratorios Hoffmann-La Roche presentado anteriormente. En este caso los organismos evaluadores eran clientes a los que había que proporcionarles un informe del producto farmacéutico a ser evaluado. Tras comprender mejor las necesidades del cliente, se consiguió un producto que se adecuase más y mejor a sus necesidades y, como consecuencia, se consiguió acortar el tiempo requerido por estos organismos para conceder la comercialización de un producto farmacéutico.

- Obtención de empleados más efectivos al disponer estos de un mayor conocimiento a nivel organizativo. No cabe duda de que cuanto más conocimiento manejen los empleados, éstos se volverán más efectivos. Como ejemplo se puede citar el Trainer's Trading Post implantado en HP [6] con la finalidad de crear un forum para que los formadores y educadores compartiesen ideas, material y métodos. De esta manera, los formadores podían aprender unos de otros.
- Mejora de la toma de decisiones, puesto que ésta se hace en base a un mayor y mejor conocimiento. Al tener un gran conocimiento acerca del entorno y de los clientes, se consigue identificar oportunidades de negocio, tal y como es el caso de Zara presentado anteriormente.

Por todo ello, se puede concluir que la GC es una práctica que busca maximizar el valor de una organización ayudando a su gente a innovar y adaptarse a los cambios.

Análisis de las principales propuestas para gestionar el conocimiento organizativo

Dentro de la extensa literatura existente acerca de la GC, han surgido numerosos intentos y aproximaciones para facilitar la compleja tarea de desarrollar e implantar esta disciplina en una organización. Entre ellas, cabe destacar el modelo propuesto de T. Beckman y J. Liebowitz [9] [10], el marco metodológico de K. Wiig, R. De Hoog y R. Van der Spek [11], el

proyecto WORKS [12], la metodología de B. Junnarkar [13], la metodología de S. Stabb y H. P. Schnurr [14], la propuesta de A. Tiwana [15] y la metodología MEGICO [16].

Todas estas propuestas han sido analizadas respecto a las fases que, idealmente, debiera de contemplar toda aproximación para la construcción de sistemas de estas características, a saber:

- Adquisición del conocimiento. Toda aproximación debe guiar al equipo de GC en la identificación del conocimiento necesario a capturar. Idealmente, la propuesta debiera indicar qué tipos de conocimientos se deben buscar, así como las características a considerar para cada activo de conocimiento identificado.
- Asimilación del conocimiento. Una vez capturado el conocimiento, es necesario asimilar aquel que se considere relevante para el funcionamiento de la organización. Para ello es necesario entenderlo, es decir, conceptualizarlo, y posteriormente representarlo en la forma que mejor convenga; esto es, formalizarlo.
- Consolidación del conocimiento. Se debe contrastar la adecuación del conocimiento ya asimilado con el objetivo de evitar incorporar conocimiento no deseado al sistema de GC. Esta comprobación ha de realizarse a dos niveles. En primer lugar, será necesario chequear la validez del conocimiento, ya que la inclusión de conocimientos incorrectos en el sistema de GC tendrá graves consecuencias a nivel organizativo. En segundo lugar, ha de

asegurarse que el conocimiento incorporado no sólo sea correcto, sino que también sea relevante para la organización. Es de suma importancia no sobrecargar el sistema de GC con conocimientos no relevantes.

- Creación del sistema de soporte a la GC. Toda aproximación debería guiar al equipo de GC a la hora de determinar la forma en la que se incorporará nuevo conocimiento al sistema y su posterior distribución.
- Creación del entorno de colaboración. Dado que el éxito de un sistema de GC depende, en gran medida, de la forma en la que los empleados acepten y usen el sistema, es preciso definir mecanismos que contemplen la forma de mitigar el rechazo al proyecto de GC.

Sin embargo, no es suficiente con determinar si las aproximaciones estudiadas contemplan o no cada una de estas fases, sino que es necesario determinar el grado de detalle con el que las acometen. Por ello, cada una de estas fases será analizada en función de los siguientes criterios:

- Proceso. Especifica si la propuesta contempla, o no, la fase analizada y, en caso afirmativo, indica su grado de explicitación. La Tabla 1 muestra los posibles valores que puede tomar este atributo.
- Soporte. Indica si la propuesta propone, o no, técnicas y herramientas que den soporte a la fase analizada.

Valor	Significado
No	No se contempla la fase.
General	Se contempla la fase pero no de forma detallada.
Guías	Se contempla la fase y, además, se proponen mecanismos para su acometimiento, aunque éstos no se encuentren explicitados completamente.
Formal	El proceso está completamente formalizado, es decir, se explicita qué hay que hacer, cuándo hay que hacerlo, cómo ha de hacerse y quién ha de hacerlo.

Tabla 1: Posibles valores del atributo Proceso

La Tabla 2 recoge, de forma resumida, los resultados de dicho estudio, junto con los valores que debiera de tener una aproximación ideal para el desarrollo de este tipo de sistemas.

	Adquisición		Asimilación		Consolidación		Creación del sistema de soporte		Creación del entorno de colaboración	
	Proceso	Soporte	Proceso	Soporte	Proceso	Soporte	Proceso	Soporte	Proceso	Soporte
Beckman y Liebowitz	Guías	Si	Guías	Si	Guías	Si	Guías	Si	No	-
Wiig, De Hoog, y Van der Spek	Guías	Si	Guías	Si	No	-	Guías	Si	No	-
Proyecto WORKS	General	No	General	No	No	-	General	No	No	-
B. Junnarkar	Guías	Si	Guías	Si	No	-	Guías	Si	No	-
Stabb y Schnurr	General	Si	General	Si	No	-	General	Si	No	-
A. Tiwana	Guías	Si	Guías	Si	No	-	Guías	Si	Guías	Si
MEGICO	¿Formal?	Si	¿Formal?	Si	No	-	¿Formal?	Si	¿Formal?	Si
Ideal	Formal	Si	Formal	Si	Formal	Si	Formal	Si	Formal	Si

Tabla 2: Evaluación de las aproximaciones principales en la GC

Analizando en detalle estos resultados, se pueden obtener las siguientes conclusiones:

- La mayoría de las aproximaciones se centran en las fases de adquisición, asimilación, e incorporación y distribución del conocimiento, dejando de lado actividades tan importantes como la verificación y validación, o la definición del entorno de colaboración.
- Salvo en el caso de MEGICO, que dado su carácter privado no se puede comprobar, se puede afirmar que las aproximaciones actuales no definen, de manera formal, las fases y actividades que acometen. En el mejor de los casos, estas aproximaciones se limitan a indicar guías generales para el acometimiento de las fases que definen.

En definitiva, y pese a la gran proliferación de aproximaciones para el desarrollo de sistemas de GC, se puede concluir que no existe una propuesta, lo suficientemente elaborada, que guíe al equipo de GC en el acometimiento de su labor en todos los aspectos que exige una GC global. Por tanto, se está ante una situación en la que cada organización se ve en la obligación de elaborar de manera *ad hoc* su propuesta.

Estrategia planteada

Los autores atribuyen los problemas anteriores a que todas estas propuestas se articulan únicamente entorno al proceso de gestión, dejando

de lado, quizás, al aspecto más importante: el objeto a gestionar, es decir, el conocimiento. Por tanto, es preciso centrar la atención sobre el estudio de este elemento con el fin de determinar:

- i. En qué tipo de conocimientos se está interesado.
- ii. Cuáles son las propiedades a identificar para activo de conocimiento en función del tipo al que pertenezca.

En lo que respecta a estos puntos, los autores han llevado a cabo un estudio riguroso de este elemento, que se encuentra detallado en [17]. Como resultado de este estudio, se concluye que todo conocimiento puede ser clasificado en los siguientes niveles:

- Estático, el cual está compuesto por el conocimiento estructural u operativo, es decir, los hechos que son ciertos en el dominio y que se pueden usar en las operaciones: conceptos, propiedades, relaciones y restricciones.
- Dinámico, el cual conforma el comportamiento que tiene lugar en el dominio, es decir, la funcionalidad, la acción, el proceso o el control: inferencias, cálculos y secuencia de pasos. A su vez, este nivel puede dividirse en estratégico (que incluye qué hacer, cuándo y en qué orden) y táctico (que especifica cómo obtener nuevo conocimiento operativo).

La Figura 1 muestra gráficamente estos niveles junto con sus interrelaciones. En ella se puede apreciar como el conocimiento estratégico maneja al táctico —(1) en la Figura—, ya que para cada paso de último

nivel se indican las inferencias y cálculos que en él se llevan a cabo. Por su parte, el conocimiento operativo es manejado por los otros dos, ya que especifica con qué se deciden las alternativas de ejecución en la secuencia de pasos (puntos de bifurcación) —(2) en la Figura— y en base a qué se hacen las inferencias y los cálculos —(3) en la Figura.

El recurso gráfico en forma de pirámide muestra el hecho de que los distintos niveles funcionales del conocimiento siguen una estructura de apoyo entre sí. El conocimiento estratégico se sitúa en la cima ya que controla el comportamiento que se produce en el dominio. Por su parte, el táctico precisa del operativo que, por otra parte, es el más cuantioso. Debido a todo ello, el conocimiento operativo se dispone como base de la pirámide, estando por encima de él el táctico.

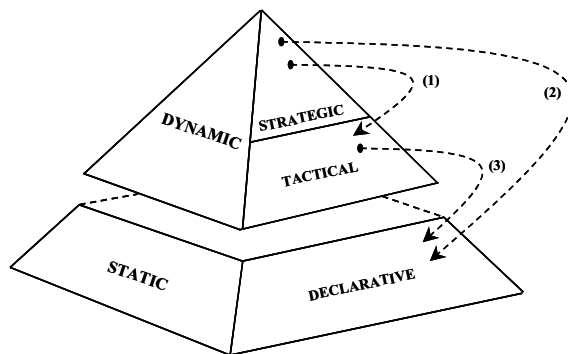


Figura 1: Niveles funcionales del conocimiento

Como resultado de este estudio, se identifican todos los elementos conceptuales que pueden ser necesarios manejar en cualquier dominio.

Lógicamente, estos elementos serán los que deben ser considerados en un sistema de GC y, por tanto, los que se deben formalizar en dichos sistemas.

Bibliografía

- [1] Drucker, P.F., et al., 1998. “*Harvard Business on Knowledge Management*”. Harvard Business School Press. Boston, USA.
- [2] Brooking, A., 1996. “*Intellectual Capital. Core Asset for the Third Millennium Enterprise*”. International Thomson Business Press. Londres, UK.
- [3] Davenport, T.H., Prusak, L., 2000. “*Working Knowledge: How Organizations Manage What They Know*”. Harvard Business School Press. Boston, USA.
- [4] Ruggles, R., 1997. “*Why Knowledge? Why Now?*”. Perspective on Business Processes. Centre of Business Innovation, Ernst and Young. nº 1, pp. 2-8.
- [5] Herreros, C., 2000. “*Zara: Un reto al pensamiento maduro*”. http://www.gestiondelconocimiento.com/documentos2/carlos_herreros/caso_zara.htm.
- [6] Davenport, T.H., 1997. “*If Only HP Knew What HP Knows...*”. Perspective on Business Processes. Centre of Business Innovation, Ernst and Young. nº 1, pp. 20-25.

- [7] Cohen, D., 1997. "*Knowing the Drill: Virtual TeamWork at BP*". Perspective on Business Processes. Centre of Business Innovation, Ernst and Young. n° 1, pp. 14-19.
- [8] Seemann, P., 1997. "*A Prescription for Knowledge Management: What Hoffmann-La Roche's Case Can Teach Others*". Perspective on Business Processes. Centre of Business Innovation, Ernst and Young. n° 1, pp. 26-33.
- [9] Beckman, T., 1997. "*A Methodology for Knowledge Management*". International Association of Science and Technology for Development (IASTED) AI and Soft Computing Conference. Banff, Canada.
- [10] Liebowitz, J., Beckman, T., 1998. "*Knowledge Organizations. What Every Manager Should Know*". CRC Press. Florida, USA.
- [11] Wiig, K., de Hoog, R., Van der Spek, R., 1997. "*Supporting Knowledge Management: A Selection of Methods and Techniques*". Expert Systems with Applications, vol. 13 (1), pp. 15-27.
- [12] Daniel, M., Decker, S., Domanetzki, A., Heimbrod-Habermann, E., Höhn, F., Hoffmann, A., Röstel, H., Studer, R., Wegner, R., 1997. "*ERBUS-Towards a Knowledge Management System for Designers*". Proceedings of the Knowledge Management Workshop at the 21st Annual German AI Conference. Freiburg, Alemania.

- [13] Junnarkar, B., 1997. “*Leveraging Collective Intellect by Building Organizational Capabilities*”. Expert Systems with Applications. vol. 13, nº 1, pp. 29-40.
- [14] Staab, S., Schnurr, H.P., 1999. “*Knowledge and Bussines Processes: Approaching and Integration*”. Workshop on Knowledge Management and Organizational Methods. IJCAI99. Estocolmo, Suecia.
- [15] Tiwana, A., 2000. “*The Knowledge Management Toolkit. Practical Techniques for Building a Knowledge Management System*”. Prentice-Hall. New Jersey, USA.
- [16] Paradela, L.F., 2001. “*Una Metodología para la Gestión del Conocimiento*”. Tesis Doctoral. Facultad de Informática. Universidad Politécnica de Madrid, España.
- [17] Andrade J., Ares J., García R., Rodríguez S., Silva A., 2005. “*Natural Language Analysis: Towards a More Human-Centred Conceptual Modelling in Software Engineering*”. En “*Encyclopaedia of Human Computer Interaction*”, Ghaoui, C. (ed.), Idea Group. Pendiente de publicar en 2005.

Técnicas de Inteligencia Artificial para el Tratamiento de Datos Astronómicos

*Alejandra Rodríguez¹, Carlos Dafonte¹, Bernardino Arcay¹,
Iciar Carricajo² y Minia Manteiga²*

*¹ Departamento de Tecnologías de la Información y de las Comunicaciones.
Universidade da Coruña*

Email: {arodriguezf, dafonte, cibarcay}@udc.es

*² Departamento de Ciencias de la Navegación y de la Tierra.
Universidade da Coruña*

Email: {iciar, manteiga}@udc.es

Introducción

La Astrofísica es una rama de la ciencia que se dedica principalmente al estudio de la distribución de energía de la radiación electromagnética a través del análisis de objetos observados.

Este estudio forma parte de un proyecto global dedicado al análisis de las últimas fases evolutivas de las estrellas. Los estudios evolutivos son una parte esencial de la Astrofísica ya que permiten analizar los cambios temporales que experimentan los cuerpos celestes (estrellas, nebulosas, planetas, etc.) en cuanto a su composición química y a sus propiedades físicas.

El estudio de la distribución espectral de las estrellas permite estudiar la evolución temporal de las mismas en la fase evolutiva que va desde la Rama Asintótica de Gigantes (AGB) hasta la fase de Nebulosa Planetaria, y su efecto en el medio interestelar galáctico. Para ello, es necesaria la clasificación espectroscópica de una amplia muestra de estrellas en las fases AGB y post-AGB.

La clasificación de espectros estelares se realiza actualmente en el sistema estándar Morgan-Keenan, que ordena las estrellas en función de su temperatura y gravedades efectivas, y resulta una herramienta de probada eficacia en:

- Clasificación de conjuntos extensos de objetos o catálogos estelares.
- Estudio de propiedades de estrellas peculiares.
- Obtención de parámetros estelares: distancia, extinción, metalicidad.
- Estudios evolutivos de estrellas.

El sistema de clasificación manual está basado en la comparación morfológica del espectro con espectros estándares a través de una inspección visual, procedimiento sujeto a un alto grado de subjetividad, que además consume una gran cantidad de tiempo y recursos humanos debido al gran volumen de información disponible hoy en día; todo ello sugiere la automatización por medio de la utilización de técnicas computacionales.

En concreto se propone un enfoque híbrido que permita la automatización de la clasificación espectral por medio de la combinación de varias técnicas de inteligencia artificial en una sola herramienta. Este enfoque resulta más eficiente que el de aplicar una única técnica pues se adapta mejor a las peculiaridades del proceso.

Los espectros se almacenan en una base de datos relacional, de donde son recuperados por un analizador espectral, que mediante técnicas de preprocesado y análisis morfológico de señales obtiene las medidas de los parámetros espectrales, tales como bandas y líneas de absorción y emisión, energía espectral, etc. Estos parámetros se utilizarán en el proceso de clasificación.

La clasificación se ha abordado desde dos grupos de técnicas de inteligencia artificial: sistemas expertos y redes de neuronas artificiales.

Los sistemas expertos utilizan los valores de los parámetros para comenzar a enlazar reglas de razonamiento (obtenidas de los espectroscopistas) con el objetivo de llegar a una conclusión. Las redes de neuronas utilizan los parámetros como patrones de la capa de entrada. El resultado de este proceso, o sea la clasificación del espectro, se introduce de nuevo en la base de datos.

Los sistemas expertos, en su primera versión, se basaron en las metodologías clásicas de sistemas basados en reglas de razonamiento [1]. Posteriormente se introdujeron variables difusas y funciones de pertenencia.

En cuanto a las redes de neuronas, se implementaron algoritmos basados en aprendizaje supervisado, en concreto backpropagation, redes competitivas (Kohonen) y algoritmos de aprendizaje híbridos (RBF).

Las secciones siguientes comienzan describiendo el sistema de clasificación estándar y las técnicas manuales de clasificación estelar. En segundo lugar se presentan los datos espectrales que se han utilizado para diseñar y validar las técnicas automáticas de clasificación. Posteriormente, se describen los algoritmos de análisis morfológico que se aplicaron a los espectros antes de ser tratados. Finalmente, se presentan los diferentes modelos de inteligencia artificial implementados, contrastando los resultados obtenidos con cada uno de ellos.

Sistema de clasificación estándar

La espectroscopia es una de las técnicas actuales más útiles y eficientes para el estudio de las estrellas, y en particular de sus condiciones físicas (temperatura, presión, densidad, etc.) y de su composición química (H, He, Ca, K, etc.).

El espectro de una estrella consiste en la distribución de luz del continuo de un cuerpo negro, distorsionada por la absorción interestelar y por la presencia de líneas y bandas de absorción y emisión de compuestos químicos [2].

Los espectros estelares se obtienen en los telescopios utilizando detectores y espectrógrafos apropiados. En las campañas astrofísicas los

observadores descomponen, utilizando espectrógrafos, la luz recogida de cada objeto en sus diferentes longitudes de onda, frecuencias o colores. A continuación, se unifican y reducen estos datos para obtener un espectro final calibrado en flujo energético ($\text{erg}^{-1}\text{cms}^{-2}\text{s}^{-1}\text{\AA}^{-1}$) y longitud de onda (\AA). La figura 1 muestra el espectro de una estrella de tipo G0 una vez reducida a su espectro unidimensional.

Como ya se ha mencionado, para estudiar los cambios temporales que experimentan las estrellas es necesario recoger y clasificar los espectros de una población de estrellas suficientemente representativa. El análisis de la distribución de los tipos espectrales dentro de la muestra recogida contribuye al estudio y comprensión de la evolución estelar, por ello, la clasificación espectral es una fase crítica que debe llevarse a cabo de una forma eficiente, rápida y apropiada.

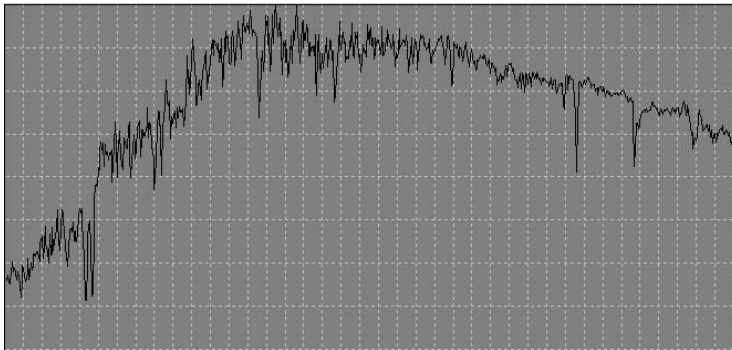


Figura 1: Espectro unidimensional de una estrella G0

El sistema actual más extendido y utilizado para clasificar los espectros estelares es el sistema MK, propuesto en 1943 por Morgan, Keenan y Kellman [3].

Una de las grandes ventajas de este esquema es que las clasificaciones MK de los espectros suelen ser estáticas, pues se basan en el estudio visual de los espectros y en la aplicación de un conjunto de criterios de clasificación; sin embargo, como contrapartida, diferentes expertos podrían llegar a clasificar el mismo espectro de formas sensiblemente diferentes.

Este sistema de clasificación cuantifica temperaturas estelares y niveles de luminosidad. El sistema MK se estableció utilizando como referencia observaciones espectroscópicas en el rango entre 3930 y 4860 Å. Las estrellas se dividieron en grupos o tipos espectrales cuya definición se basa principalmente en la intensidad de las líneas de hidrógeno y en la presencia o ausencia de algunas líneas significativas (Calcio, Hierro, Helio, etc.) y de bandas de absorción moleculares.

La temperatura de las estrellas se concreta en una secuencia denominada OBAFGKM, abarcando desde las estrellas más calientes (tipo O) a las más frías (tipo M). Cada uno de estos tipos espectrales se subdivide a su vez utilizando un sistema decimal (en el rango 0-9). De esta forma, la estrella O0 sería la estrella con mayor temperatura del sistema MK y la estrella M9 la de menor.

El sistema MK adjudica además a cada espectro una clase de luminosidad, desde el nivel I (supergigantes) hasta el V (enanas), que depende del brillo intrínseco de la estrella.

La tabla 1 muestra las principales propiedades de cada tipo espectral en el sistema estándar de clasificación MK.

Tipo	Color	Temperatura	Características	Ejemplos
O	Azul	> 25,000 K	Helio ionizado	10 Lacertra
B	Azul	11,000-25,000 K	Helio neutral Hidrógeno neutral	Rigel Spica
A	Azul	7,500-11,000 K	Hidrógeno neutral	Sirius Vega
F	Azul a Blanco	6,000-7,500 K	Hidrógeno neutral Calcio ionizado	Canopus Procyon
G	Blanco a Amarillo	5,000-6,000 K	Hidrógeno neutral Calcio fuertemente ionizado	El sol Capella
K	Naranja a Rojo	3,500-5,000 K	Metales neutrales Calcio ionizado	Arcturus Aldebaran
M	Rojo	< 3,500 K	Banda moleculares Metales neutrales	Betelgeuse Antares

Tabla 1: Características principales de los tipos espectrales en el sistema MK

Un sistema de clasificación estelar adecuado y eficiente debería implicar un compromiso entre mantener la información completa del espectro y la necesidad de un resumen compacto de dicha información. Este compromiso se alcanza en gran medida con las clasificaciones en el sistema MK, pues se obtiene una síntesis óptima del espectro mediante la determinación de su tipo espectral y luminosidad.

Actualmente, la obtención del tipo espectral y la luminosidad de una estrella en el sistema MK se realiza habitualmente de forma manual. Los espectroscopistas analizan el espectro basándose únicamente en su propia experiencia en el campo de la clasificación estelar.

El proceso de clasificación manual se basa frecuentemente en el estudio visual de los espectros y en un conjunto de criterios estándares que involucran características espectrales tales como líneas de emisión y bandas moleculares [3].

Los clasificadores seleccionan en catálogo de espectros de referencia que han sido previamente clasificados en el sistema MK. Este catálogo se utilizará como una guía durante todo el proceso de clasificación.

El primer paso de la clasificación manual es la normalización del espectro para aislar el componente continuo (afectado por el enrojecimiento interestelar), a partir de ese momento la comparación de los espectros con los del catálogo guía puede centrarse en la información aportada por las líneas espectrales. Asimismo, para facilitar la comparación entre los

espectros que se pretenden clasificar con los del catálogo seleccionado es necesario que todos estén en la misma escala.

El tipo espectral (o temperatura) de una estrella se obtiene calibrando la intensidad relativa de algunas líneas espectrales, en su mayoría pertenecientes a las transiciones de los átomos de hidrógeno y helio. Además, algunos grupos de líneas pueden utilizarse también para determinar densidad, presión y otras propiedades físicas. La anchura de las líneas refleja velocidades de rotación y los desplazamientos de las mismas, debidos al efecto Doppler, muestran la velocidad radial de la estrella.

Basándose en los espectros escalados y normalizados, los expertos tratan de determinar el tipo espectral y la luminosidad en el sistema MK. Esta segunda etapa consiste en el estudio de las relaciones de la intensidad de determinadas líneas de absorción (H, He, Ca, etc.) y la profundidad de determinadas bandas moleculares relevantes. De este modo los espectroscopistas obtienen una primera clasificación aproximada de cada espectro.

Esta clasificación MK inicial se refina comparando cada espectro sin clasificar con los del catálogo de referencia que corresponden al mismo tipo espectral, hasta que se logra aproximar el subtipo espectral.

Aunque este método manual de clasificación se ha utilizado con éxito a lo largo de los años, hoy en día no es muy viable puesto que las tecnologías actuales de obtención de espectros permiten recoger una cantidad enorme de datos espectrales en un tiempo relativamente corto.

Además, esta técnica presenta un alto grado de subjetividad ya que no existen criterios de clasificación definitivos y estos dependen, en gran medida, de la experiencia del espectroscopista en la tarea de clasificación estelar.

La clasificación manual de todos los espectros que actualmente se pueden obtener implicaría un aumento considerable del tiempo y los recursos humanos dedicados a esta fase del estudio de las estrellas. Por lo tanto, sería conveniente optimizar el procedimiento manual por medio de técnicas computacionales automáticas, eficientes y rápidas.

Datos astronómicos

Con el fin de diseñar herramientas automáticas que optimicen el costoso proceso de clasificación manual, hemos seleccionado una amplia muestra de espectros ópticos. En concreto, se ha elegido un conjunto de 258 espectros que comprenden todos los tipos y luminosidades posibles dentro del sistema MK.

Este conjunto de espectros seleccionados es suficientemente representativo, ya que ofrece una transición continua de las características espectrales entre cada tipo espectral y sus tipos adyacentes. Se han utilizado los catálogos públicos de Silva [4] - 28 espectros muestreados en el rango 3500 a 8900 Å con 5 Å de resolución espectral -, Pickles [5] - 97 espectros en el rango 1150 a 25000 Å con 5 Å de resolución - y Jacoby [6], 133 espectros en el rango 3510 a 7426 Å con 1.4 Å de resolución espectral.

Como paso previo a la utilización de este conjunto de espectros para el diseño de las técnicas automáticas de clasificación, los expertos que colaboran en este proyecto han analizado y corregido los 258 espectros estelares seleccionados utilizando para ello técnicas estadísticas de agrupamiento.

Técnicas de clasificación automática

En los últimos años la investigación en el campo de la clasificación espectral se ha centrado en la necesidad de desarrollar herramientas automáticas que asistan a los expertos, y en la revisión y mejora de las técnicas manuales.

En cuanto al uso de técnicas de inteligencia artificial aplicadas al diseño de sistemas de clasificación automáticos, algunos trabajos previos han utilizado también redes de neuronas artificiales [7] [8], obteniendo diferentes grados de resolución en sus clasificaciones.

Nuestro grupo de investigación ha contribuido a esta línea de investigación con el desarrollo de sistemas expertos para la clasificación de estrellas de luminosidades I, III y V [9]. Los resultados obtenidos nos han llevado a extender estos sistemas a estrellas de todos los niveles de luminosidad y a agregar nuevas técnicas, tales como redes neuronas artificiales y algoritmos estadísticos de agrupación, para refinar el tratamiento automático de los espectros.

Nuestro objetivo actual no es implementar de nuevo modelos o técnicas que ya han demostrado su conveniencia en este problema, sino combinar todas las técnicas en una herramienta única de clasificación. Proponemos la formalización de un sistema híbrido capaz de determinar el método de clasificación más apropiado para cada tipo de espectro, y de obtener clasificaciones MK *on-line* a través de una base de datos estelar en Internet (<http://starmind.tic.udc.es>).

Esquema general del sistema

El sistema de clasificación automática que se propone incluye dos módulos diferentes: un analizador espectral y un clasificador automático. En la figura 2 se muestran los diferentes componentes del sistema para la clasificación automática de espectros estelares.

El analizador espectral (módulo de análisis) lleva a cabo un análisis morfológico exhaustivo de los espectros, usando técnicas de procesamiento de señales, obteniendo como resultado una parametrización numérica de cada espectro. Este analizador se ha desarrollado en Builder C++ [10] e integra componentes ActiveX *ad hoc* para la visualización de los espectros.

El módulo de análisis recupera los datos astronómicos de una base de datos relacional que almacena y estructura toda la información disponible. Esta base de datos se ha implementado mediante el gestor PostgreSQL bajo Linux [11] Actualmente, hay aproximadamente 400 espectros almacenados en la base de datos, y pronto serán accesibles a través de Internet.

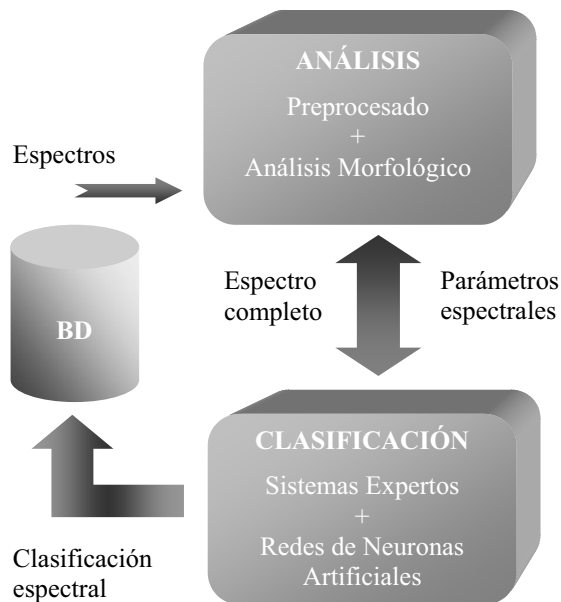


Figura 2: Esquema general del sistema de clasificación propuesto

El clasificador estelar se basa en el desarrollo de las diferentes técnicas de inteligencia artificial elegidas para esta aproximación a la clasificación MK de los espectros. Las entradas de este módulo están constituidas por los espectros completos que se pretenden clasificar y por los parámetros obtenidos mediante el analizador espectral. Las redes de neuronas artificiales se implementaron utilizando el simulador SNNS [12] y los sistemas expertos se desarrollaron utilizando OPS/R2 [13].

Actualmente estamos desarrollando una aplicación para permitir que el clasificador estelar pueda utilizarse a través de Internet, ya que nuestro

propósito principal es permitir a todos los usuarios que lo deseen clasificar sus espectros *on-line* de forma rápida, eficiente y cómoda.

Las secciones siguientes describen los diferentes módulos que integran el sistema final de clasificación automática de espectros estelares.

Análisis morfológico

El módulo de análisis realiza el preprocesado y análisis morfológico exhaustivo de los espectros estelares mediante técnicas de procesado de señales, ya que se considera para estos efectos que los espectros son señales que relacionan longitud de onda y flujo.

En este estudio morfológico de los espectros, se detectan y miden los parámetros espectrales que se utilizan en los procesos de clasificación manual. Estos parámetros pueden agruparse en tres tipos generales:

- Líneas de absorción y emisión: incluyendo hidrógeno, helio y algunas líneas metálicas (Ca, K, Fe, etc.)
- Bandas moleculares: bandas de absorción de hidrógeno y carbono, titanio, etc.
- Relaciones entre intensidades de líneas: ratios CH-K, He-H, etc.

La tabla 2 muestra las 25 características espectrales que se detectan y miden a través de los algoritmos implementados en el módulo de análisis morfológico.

Parámetro espectral	Descripción
Banda 1	5005 ± 055 Å
Banda 2	6225 ± 150 Å
Banda 3	4435 ± 070 Å
Banda4	5622 ± 180 Å
Banda 5	5940 ± 135 Å
Banda 6	6245 ± 040 Å
Banda 7	6262 ± 130 Å
Banda 8	6745 ± 100 Å
Banda 9	7100 ± 050 Å
Linea Ca II (K)	3933 Å
Linea Ca II (H)	3968 Å
Linea CH band	4300 Å
Linea H Iγ	4340 Å
Linea H Iδ	4102 Å
Linea He I	4026 Å
Linea He II	4471 Å
Linea H Iβ	4861 Å
Linea H Iα	6563 Å
Bandas principales	$\sum_{i=1}^{i=2} \text{Banda}_i$
Bandas Secundarias	$\sum_{i=3}^{i=9} \text{Banda}_i$
Ratio K-H	Ca II K / Ca II H
Ratio CH- H Iγ	CH band / H I γ
Ratio H Iδ - HeI	H I δ / He I
Ratio H Iδ - HeII	H I δ / He II
Energía espectral	Integral del flujo

Tabla 2: Parámetros de la clasificación espectral

Los parámetros espectrales descritos en la tabla anterior se obtienen mediante algoritmos de procesamiento de señales, basados fundamentalmente en la estimación del continuo local de las líneas de absorción / emisión, y en el cálculo de la energía de las bandas moleculares.

Una línea de absorción es un pico descendente (ascendente si se trata de líneas de emisión) que aparece en una zona de longitud de onda preestablecida en los espectros. Con el objetivo de calcular apropiadamente la intensidad de cada línea, se ha realizado una estimación del continuo local para cada una de ellas; para ello, se suaviza la señal con un filtro paso bajo, excluyendo los picos o desviaciones en un intervalo alrededor de la longitud de onda en la que se espera encontrar el pico de absorción o emisión. Este filtro se ha implementado mediante el método de media móvil de 5 puntos, seleccionando los 5 flujos más estables del intervalo. Esto es,

$$C_j = \left(\frac{\sum_{i=j-n}^{j+n} E_i * X_i}{N} \right), \quad (1)$$

dónde C_j es la estimación del continuo para la muestra j , E_i es el flujo en la muestra i , N es el número de valores utilizados en el método de medias móviles para calcular el continuo local, y X es un vector binario que indica los flujos representativos del continuo local de la zona espectral tratada. Por lo tanto, si E_i es un valor de flujo representativo del continuo local X_i será 1, y si E_i es un pico el valor de X_i será 0.

La intensidad de cada línea de absorción se obtiene calculando la desviación del flujo con respecto al continuo local estimado. De esta forma, la intensidad será positiva para las líneas de absorción y negativa para las de emisión.

Desde un punto de vista morfológico, una banda molecular es una zona espectral donde el flujo energético decrece repentinamente con respecto al continuo local y se mantiene durante un intervalo de longitud de onda amplio. Para tratar morfológicamente las bandas moleculares solamente es necesario medir su energía para decidir si son lo suficientemente significativas en cada espectro. En este caso, se calcula el área que comprende cada banda mediante la integral discreta del flujo entre los extremos de la misma. Esto es,

$$B_{lr} = \int_l^r L(\lambda_i) - \int_l^r E(\lambda_i), \quad (2)$$

dónde B_{lr} es el área de la banda entre las muestras l y r , L es la línea de proyección de la banda sobre el eje de abscisas, E es la función del flujo, λ la longitud de onda, l el límite izquierdo de la banda y r el límite derecho.

Sistemas expertos

La primera estrategia de clasificación de espectros estelares de forma automática consiste en la implementación de un sistema experto en el que se han integrado técnicas de sistemas basados en el conocimiento y lógica

difusa, obteniendo una emulación satisfactoria del actual proceso de clasificación manual descrito previamente. El sistema final clasifica espectros estelares con una tasa de acierto similar al porcentaje de acuerdo entre los expertos humanos que han clasificado manualmente los espectros (aproximadamente 80%).

Previamente al diseño del sistema experto, se ha realizado un estudio de la sensibilidad de los parámetros espectrales elegidos para la clasificación, con el fin de definir y delimitar correctamente los diferentes conjuntos difusos, variables y funciones de pertenencia. En este estudio se han analizado los parámetros espectrales de los catálogos utilizados como referencia en la clasificación manual, utilizando los algoritmos de análisis morfológico descritos para medir y detectar líneas de absorción y bandas moleculares.

Los resultados de este análisis nos llevaron a determinar los diferentes tipos espectrales que cada parámetro es capaz de discriminar. Algunos parámetros, que a priori parecían ser adecuados, se descartaron debido a su pobre capacidad para discriminar entre tipos. Asimismo, se incluyeron otros parámetros que no se tratan explícitamente en la clasificación manual, como por ejemplo la suma de la energía de las bandas.

Finalmente, el análisis de sensibilidad de los parámetros de clasificación determinó las variables difusas que se utilizarían en el diseño del sistema experto (clasificación global, tipo espectral y subtipo espectral) para cada clase de luminosidad. Además, se definieron también en esta fase

los diferentes conjuntos difusos y funciones de pertenencia obtenidos del análisis morfológico de los espectros del catálogo de referencia para la clasificación manual.

El sistema experto desarrollado almacena en la base de hechos la información necesaria para iniciar el proceso de razonamiento. Este conocimiento descriptivo (nombre del espectro, medidas de los parámetros espectrales, etc.) se representa por medio de *frames* [14], i. e. objetos y propiedades estructurados por niveles. Se eligió este modelo de representación debido a que es el más sencillo y adecuado para transferir los resultados del análisis morfológico al módulo de clasificación, permitiendo además realizar una equivalencia entre datos y conocimiento.

Los parámetros reales de clasificación y los valores limítrofes de cada tipo o subtipo espectral se incluyeron en el sistema experto en forma de reglas difusas. La base de reglas es la parte del sistema en la que se reproducen los criterios procedentes de la experiencia de los expertos humanos en el campo de la clasificación espectral.

En cuanto al tipo de reglas, hemos seleccionado reglas de producción *IF-THEN*, ya que permiten manejar la imprecisión e incertidumbre que caracterizan al razonamiento humano en este campo.

Las condiciones de estas reglas hacen referencia a los parámetros espectrales almacenados en la memoria de trabajo (instancia en cada momento de la base de hechos).

Las conclusiones aluden a los tres niveles de clasificación espectral deseados: global (estrellas tempranas, intermedias o tardías), tipo espectral y luminosidad.

La estrategia de selección de reglas elegida es la *Means-End Analysis* (MEA) [15]. Esta estrategia selecciona entre todas las reglas posibles aquellas que se han incorporado más recientemente a la memoria de trabajo y que contienen un número de patrones más elevado.

Las reglas de producción se encadenan utilizando razonamiento hacia delante y con una estrategia guiada por los objetivos que se desean alcanzar (los tres niveles de clasificación sucesivos). Se combinan métodos de razonamiento guiado con un método basado en valores de verdad, puesto que cada regla lleva asociado un factor de credibilidad que se asignó de acuerdo a la experiencia de los expertos que realizan el proceso de clasificación espectral de forma manual.

Utilizando como punto de partida la metodología clásica de Shortliffe y Buchanan [1], se ha llevado a cabo una evolución que incluye conjuntos difusos y funciones de pertenencia contextualizados para cada tipo espectral, permitiendo la superposición entre ellos. En cuanto a la inferencia, se utiliza el método *Max-product*, que combina la influencia de todas las reglas activas y proporciona una salida continua. En nuestra estrategia, los factores de credibilidad de cada regla se han considerado como valores de verdad. Finalmente, se utiliza el método del centroide difuso para obtener las salidas de cada conjunto de reglas activo. La

clasificación de las estrellas se obtiene con un grado de confianza, determinado por la probabilidad de cada salida.

Esta estrategia mixta, que combina reglas de producción clásicas con conjuntos difusos y funciones de pertenencia, ha logrado adaptarse de forma más adecuada al tipo de razonamiento humano en este campo, permitiéndonos tratar de forma más adecuada la imprecisión e incertidumbre implícitas en el proceso de clasificación manual.

Esta primera técnica del módulo de clasificación se desarrolló utilizando OPS/R2 [13], integrándolo con el analizador espectral mediante librerías de enlace dinámicas (DLL).

Redes de neuronas artificiales

La segunda técnica implementada en el módulo de clasificación es redes de neuronas artificiales, pertenecientes tanto a modelos de aprendizaje supervisado como no supervisado. En concreto se utilizaron redes *Backpropagation*, *Kohonen* y *RBF (Radial Basis Functions)* [16].

Los patrones de entrenamiento, validación y prueba de estas redes se obtuvieron de forma automática añadiendo las funciones necesarias al analizador espectral. Las entradas de las redes están formadas tanto por los valores de los parámetros de clasificación (líneas y bandas principalmente), como por los valores de flujo de zonas espectrales completas.

Las redes de neuronas se diseñaron, implementaron y validaron utilizando el simulador SNNs [12].

Tipo de red	Patrones de entrada	Capas ocultas
BP Tipo espectral	Parámetros espectrales	10
BP Tipo espectral	Parámetros espectrales	5x5
BP Tipo espectral	Parámetros espectrales	10x10
BP Tipo espectral	Parámetros espectrales	10x5x3*
BP Tipo espectral	659 valores de flujo	100x50x10x3
BP Luminosidad	Parámetros espectrales	10x10
BP Luminosidad	659 valores de flujo	100x50x10x3
RBF Tipo espectral	Parámetros espectrales	16
RBF Tipo espectral	Parámetros espectrales	8*
RBF Tipo espectral	Parámetros espectrales	4
RBF Tipo espectral	659 valores de flujo	124
RBF Luminosidad	Spectral parameters	8
RBF Luminosidad	659 valores de flujo	124
Kohonen Tipo espectral	Parámetros espectrales	2x2
Kohonen Tipo espectral	Parámetros espectrales	12x12*
Kohonen Tipo espectral	Parámetros espectrales	24x24
Kohonen Luminosidad	Parámetros espectrales	2x2

Tabla 3: Topologías implementadas para las redes Backpropagation (BP), Kohonen y RBF

En la tabla 3 se muestran las diferentes topologías implementadas para los tres modelos de redes elegidas.

La red *backpropagation* que alcanzó unos resultados más satisfactorios se corresponde con una red entrenada con 25 parámetros espectrales en la capa de entrada y tres capas ocultas de 10, 5 y 3 neuronas. Los mejores resultados para las redes *Kohonen* se alcanzaron con un mapa bidimensional de 12x12 unidades. Las redes RBF que alcanzaron un rendimiento mayor se corresponden con una topología formada con 25 neuronas en la capa de entrada (parámetros espectrales) y una capa oculta de 8 neuronas. En la tabla anterior las topologías que obtuvieron un mejor rendimiento se muestran marcadas con *.

Una vez implementadas ambas técnicas de clasificación espectral, se realizó un estudio adicional consistente en optimizar el sistema experto implementado utilizando los resultados obtenidos por las redes de neuronas en su mejor implementación. Para ello, se analizaron los pesos de las neuronas de la capa de salida con el fin de determinar para cada tipo espectral los parámetros de entrada que presentaban una mayor influencia en la salida. Los valores normalizados de estos pesos se incluyeron en el sistema experto en forma de factores de credibilidad de las reglas afectadas para cada tipo y subtipo espectral.

Con esta modificación del sistema de reglas se alcanzó un aumento significativo del rendimiento del sistema experto original.

Resultados

Para realizar un estudio comparativo de las técnicas de inteligencia artificial seleccionadas para abordar el problema de la clasificación espectral, se seleccionaron 100 espectros de diferentes tipos y luminosidades. La figura 3 muestra el comportamiento de las diferentes técnicas y el de dos de los expertos humanos en espectroscopia que colaboran en este proyecto.

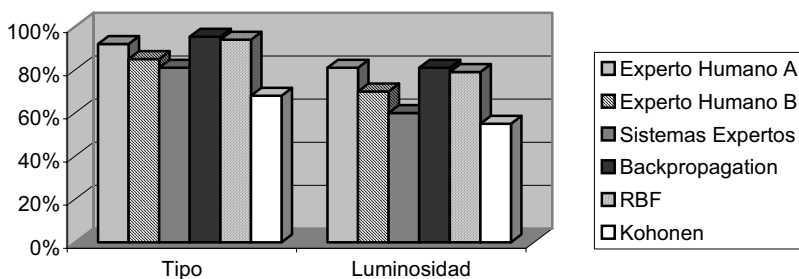


Figura 3: Rendimiento final para 100 espectros

Las redes *backpropagation* y *RBF* han obtenido una tasa de éxito de aproximadamente el 95% para los tipos espectrales. El modelo de *Kohonen* obtuvo un bajo rendimiento en todas sus implementaciones, lo cual puede deberse al tamaño insuficiente del conjunto de entrenamiento puesto que este tipo de red necesita agrupar los datos por sí misma extrayendo similitudes entre ellos.

Los sistemas expertos, si bien han resultado de gran utilidad para realizar la clasificación inicial de los espectros, obtienen unas tasas de acierto inferiores a las de las redes de neuronas.

Enfoque híbrido

En las secciones anteriores se han tratado dos métodos diferentes para obtener clasificaciones MK de los espectros estelares. Nuestro enfoque híbrido consiste en seleccionar aquellas técnicas que han obtenido un mejor rendimiento para cada nivel de clasificación e integrarlas en una única herramienta capaz de clasificar espectros estelares automáticamente y de forma eficiente.

El sistema final propuesto se basa principalmente en un sistema experto inicial que determina la clasificación global de cada estrella (temprana, intermedia o tardía) y que, de acuerdo a ella, envía los espectros a las redes de neuronas específicas más adecuadas para obtener el tipo espectral y la luminosidad. La clasificación del sistema experto se utiliza, como información adicional, en aquellos casos en los que los otros métodos no son capaces de discriminar.

En la figura 4 se muestra la interfaz gráfica de este enfoque híbrido. A través de esta herramienta de clasificación única, los usuarios pueden seleccionar, visualizar, analizar y clasificar tantos espectros como deseen de forma rápida, cómoda y sencilla.

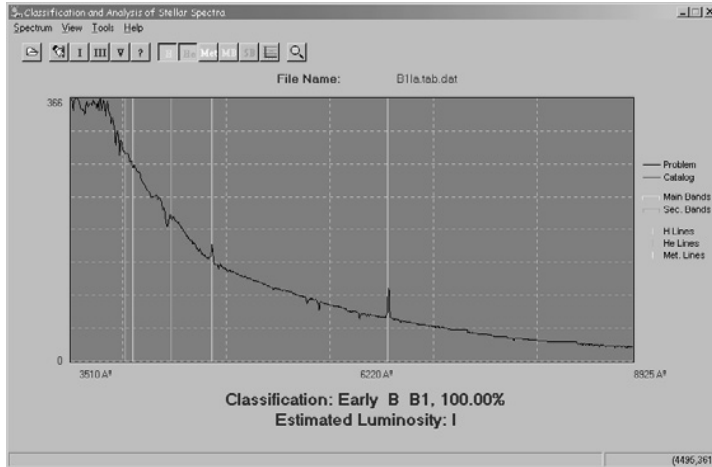


Figura 4: Interfaz del sistema final

Conclusiones

En este capítulo se ha propuesto una estrategia híbrida para abordar el problema de la clasificación MK de las estrellas. Se ha desarrollado un sistema capaz de analizar morfológicamente y clasificar automáticamente los espectros ópticos de las estrellas.

El sistema implementado combina técnicas de procesamiento de señales con sistemas expertos y redes de neuronas artificiales.

Las técnicas de clasificación descritas han obtenido, en sus mejores versiones, una tasa de acierto de aproximadamente el 95% para un conjunto

de prueba de 100 espectros. Este rendimiento significa un aumento del 10% respecto a las clasificaciones manuales realizadas por expertos del campo de la astrofísica.

Finalmente, las técnicas de inteligencia artificial implementadas se han integrado en un sistema híbrido capaz de seleccionar el mejor método de clasificación para cada tipo de espectro, lo cual permite un mayor grado de adaptación al problema.

Actualmente estamos analizando los modelos de redes funcionales para determinar si son aplicables a la clasificación MK de espectros estelares. Asimismo, estamos desarrollando un *web site* para permitir que las clasificaciones automáticas se realicen on-line de forma rápida y eficiente.

Bibliografía

- [1] Buchanan, B., Shortliffe, E., 1984. “*Ruled-based Expert Systems*”. Addison-Wesley.
- [2] Zombeck, M.V., 1990. “*Handbook of Astronomy and Astrophysics*”. 2nd. Ed. Cambridge University Press.
- [3] Morgan, W.W., Keenan, P.C., Kellman, E., 1943. “*An Atlas of Stellar Spectra with an outline of Spectral Classification*”. University of Chicago Press.

- [4] Silva, D.R., Cornell, M.E., 1992. “*A New Library of Stellar Optical Spectra*”. The Astrophysical Journal Suppl., vol 81 (2), pp. 865-881.
- [5] Pickles, A.J., 1998. “*A Stellar Spectral Flux Library. 1150-25000 Å*”, Publications of the Astronomical Society of the Pacific., vol. 110, pp. 863-878.
- [6] Jacoby, G.H., Hunter, D.A., Christian, C.A., 1984. “*A Library of Stellar Spectra*”. The Astrophysical Journal Suppl., vol. 56, pp. 257-281.
- [7] Weaber, W., Torres-Dodgen, A., 1995. “*Neural Networks Classification of the near-infrared spectra of A-type Stars*”. The Astrophysical Journal, vol. 446, pp. 300-317.
- [8] Gulati, R.K., Singh, H.P., 1998. “*Stellar Classification using Principal Component Analysis and Artificial Neural Networks*”. Monthly Notices Royal Astronomical Society, vol. 295 (2), pp. 312-318.
- [9] Rodriguez, A., Arcay, B., Dafonte, C., Manteiga, M., Carricajo, I., 2004. “*Automated knowledge-based analysis and classification of stellar spectra using fuzzy reasoning*”. Expert Systems with Applications, vol. 27 (2), pp. 237-244.

- [10] Hollingworth, J., Butterfield, D., Swart, B., Allsop, J., 2000. “*C++ Builder 5.0. Developer's Guide*”. SAMS.
- [11] Momjiam, B., 2000. “*PostgreSQL, Introduction and Concepts*”. Adisson-Wesley.
- [12] SNNS, 2001. Stuttgart Neural Network Simulator, Web: <http://www-ra.informatik.uni-tuebingen.de/SNNS/>.
- [13] Forgy, C.L., 1986. “*The OPS User's Manual, System Version 2.2*”. Production Systems Technologies Inc.
- [14] Sowa, J.F., 1999. “*Knowledge Representation: Logical and Computational*”. Brooks Cole Publishing Co.
- [15] Valette-Florence, P., 1994. “*Introduction to Means-End Chain Analysis*”. Rech. Appl. Mark, vol. 9, pp. 93-117.
- [16] Haykin, S., 1994. “*Neural Networks. A Comprehensive Foundation*”. MacMillan College Pub.



BIOINFORMÁTICA: Genética e Informática

José M^a Barreiro, Juan Pazos¹, Alfonso Rodríguez-Patón y Andrés Silva

Facultad de Informática

Universidad Politécnica de Madrid

Email: jpazos@fi.upm.es¹

Introducción

Las relaciones, mutuas y de doble dirección, entre la genética y la informática, vienen de antiguo y son, como se muestra en la tabla 1, más variadas, estrechas y profundas de lo que a primera vista pudiera parecer. De la genética a la informática, quizás la aportación más conocida sea la computación evolutiva cuyos representantes más conspicuos son los algoritmos y programas genéticos así como las estrategias evolutivas, algoritmos y los programas genéticos como medios de búsqueda. De la informática a la genética están los autómatas reproductores y el descifrado del código genético, como más relevantes. Por ello, en el apartado 3, se van a describir los autómatas reproductores de von Neumann. A continuación, en los puntos 4 y 5 se van a considerar, respectivamente, el código genético y el enfoque de Deutsch. Posteriormente, en el punto 6, se introducirá el concepto de prueba de χ^2 que se utilizará para, en el punto 8, establecer la conjetura o hipótesis de trabajo conclusión central del trabajo.

INFORMÁTICA VERSUS GENÉTICA	
De la Genética a la Informática	De la Informática a la Genética
<p>Natural Computation (NC) \equiv Evolutionary Computation (EC) [Genetics Algorithms (GA) + Evolution Strategies (ES) + Evolutionary Programming (EP)] + Neural Networks (NN) + Genetic Programming</p> <p>Fogel, Owens y Walsh, establecen como pueden evolucionar autómatas de estado finito mediante transformaciones unitarias y dos operadores genéticos: selección y mutación.</p> <p>Rechenberg define las estrategias evolutivas de poblaciones de máquinas de estados finitas.</p> <p>Holland y discípulos definen los algoritmos genéticos.</p> <p>1992 Koza propone usar la técnica de computación evolutiva par encontrar el mejor procedimiento para resolver problemas generando así la programación genética.</p> <p>Michalewicz establece los programas evolutivos como una forma de representación natural de los AA. GG. Y operadores genéticos sensibles al contexto.</p>	<p>1940 Shannon presenta una disertación doctoral titulada "Un álgebra para la Genética Teórica".</p> <p>1944 Erwin Schrödinger, conjetura la existencia del código genético.</p> <p>John Von Neumann, estableció los principios a partir de los cuales una máquina podía autorreproducirse.</p> <p>Crick, dio afortunada pero erróneamente, el nombre de código genético al pequeño diccionario que muestra la relación entre las cuatro bases que componen el ADN y los veinte aminoácidos que son las letras del lenguaje de las proteínas.</p> <p>John G. Kemeny, define las características de la reproducción de máquinas y como podría realizarse.</p> <p>Lionel S. Penrose and Roger Penrose, basándose en los trabajos de Homer Jacobson y Kemeny, abordaron los problemas mecánicos de la autorreproducción.</p> <p>1982 Tipler dio una justificación para el uso de autómatas autorreproductores.</p>

Tabla 1: Relación entre la informática y la genética

Computación evolutiva

La computación evolutiva forma parte de un conjunto de métodos de resolución de problemas que remedan, con mayor o menor exactitud, procesos naturales, como las Redes de Neuronas o la solidificación simulada. Todas ellas se engloban bajo el epígrafe de “Computación Natural” (CN), y que, en forma de ecuación, puede expresarse como sigue:

$$\text{CN (Computación Natural)} = \text{CE (Computación evolutiva)} + \text{RR.NN (Redes de Neuronas)} + \text{Programación Genética}$$
$$\text{CE (Computación Evolutiva)} = \text{AA.GG (Algoritmos Genéticos)} + \text{EE (Estrategias Evolutivas)} + \text{PE (Programación Evolutiva)}$$

Y que se consideraran más en detalle a continuación.

Pese a lo que pudiera parecer, al menos a simple vista, el desarrollo de la CN no se debió tanto a discusiones académicas, cuanto a la necesidad de resolver problemas cada vez más complejos en múltiples campos del conocimiento, para los que las técnicas clásicas de búsqueda determinista o analíticas no suelen ser de gran utilidad. Esto obligó, a desarrollar nuevos enfoques menos analíticos y más sintéticos.

En este sentido, la “Computación Evolutiva” (CE) es un enfoque alternativo dentro de la CN, para abordar problemas complejos de “búsqueda”, “optimización” y “aprendizaje” a través de modelos computacionales, cuya implantación concreta recibe el nombre de

“Algoritmo(s) Evolutivo(s)” (AE), de procesos evolutivos. Dicho de otro modo, la CE trata de desarrollar mecanismos estocásticos de búsqueda en paralelo con los que sustituir o mejorar las técnicas clásicas de búsqueda determinista cuando éstas no existen o no son buenas.

Los términos “búsqueda” y “optimización” son dos aspectos del mismo concepto o, si se prefiere, son como las dos caras de la misma moneda. La diferencia entre ellos estriba en que con el término “búsqueda” se enfatiza el proceso en sí, mientras que al decir “optimización” se hace hincapié en el resultado del proceso.

Quede claro que no se persigue una “emulación” de los procesos naturales; pues, a lo sumo, lo que se pretende es una simulación de los mismos o, más estrictamente, llevar a cabo, por analogía, una metáfora de los mismos. Por eso un AE será tanto mejor cuanto mejores resultados proporcione en la resolución del problema planteado, independientemente de su fidelidad con la biología. De hecho, la mayoría de estos algoritmos son excesivamente simplistas desde un punto de vista biológico, pero lo suficientemente complejos como para proporcionar mecanismos de búsqueda potentes y robustos.

Más precisamente, se denomina AE a cualquier procedimiento estocástico de búsqueda basado en el principio de evolución, cuya finalidad última es “la supervivencia del más apto” y la forma de conseguirla es mediante la “adaptación al entorno”. Para llevarlo a cabo un AE debe disponer, como se muestra en la figura 1, de:

Una población de posibles soluciones debidamente representados a través de “individuos”.

Un procedimiento de “solución” basado en la “aptitud de los individuos”.

Un método de “transformación”; esto es, de construcción de nuevas soluciones a partir de las actualmente disponibles.

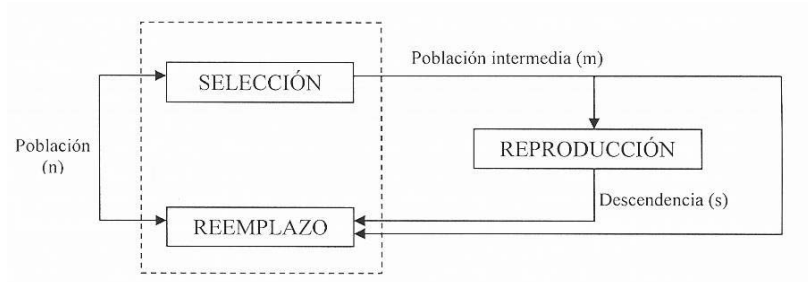


Figura 1: Esquema general de un AE

Más concretamente, al ejecutar un AE una población de individuos, que representa a un conjunto de candidatos a soluciones de un problema, es sometida a una serie de transformaciones con las que se actualiza la búsqueda y después a un proceso de selección que favorece a los mejores individuos. Cada ciclo de “transformación-selección” constituye una generación. Se espera que el AE, tras un cierto número de iteraciones; es decir, generaciones, consiga el mejor individuo que esté razonablemente próximo a la solución buscada.

Sobre este esquema general se han desarrollado varias implantaciones fundamentales siguientes:

Algoritmos Genéticos. Los algoritmos genéticos, en adelante AA.GG en plural y AG en singular, son una familia de modelos computacionales inspirados en la evolución. Estos algoritmos codifican una solución potencial para un problema específico en una estructura de datos sencilla similar a un cromosoma y aplicando operadores de recombinación sobre esa estructura para preservar información crítica. Los AA.GG son a menudo vistos como funciones de optimización aunque el rango de problemas en los que son aplicados es bastante amplio. Una implementación de un AG comienza con una población (normalmente aleatoria) de cromosomas. Entonces se evalúan estas estructuras y se localizan oportunidades de reproducción de forma que los cromosomas, que representan una mejor solución para el problema etiquetado, consiguen más cambios en su reproducción que aquellos cromosomas que tienen soluciones peores. La "bondad" de una solución se define típicamente en relación con la población actual.

Esta particular descripción de un AG es intencionadamente abstracta porque en el mismo sentido, el término AG tiene dos significados. En un sentido estricto, el algoritmo genético se refiere al modelo introducido e investigado por Holland (1975) y por sus alumnos (Dejong, 1975). Este es todavía el caso de la mayoría de teorías existentes de los AA.GG aplicados, aisladamente o primariamente, sobre el modelo introducido por Holland, así como sus variantes, que han sido denominados, AA.GG canónicos. En los avances teóricos recientes sobre modelos de AA.GG también, se aplican

principalmente AA.GG canónicos. En un uso más amplio del término, un AG es un modelo basado en poblaciones que usan operadores de selección y recombinación para generar nuevos ejemplos de puntos en el espacio de búsqueda. Desde una perspectiva computacional, los AA.GG, hacen evolucionar una población de enteros binarios sometiéndolos a transformaciones unitarias y binarias genéricas y a un proceso de selección.

En general, se puede decir que los AA.GG son métodos estocásticos (referidos tanto a las fases de selección como de transformación), de búsqueda ciega (no disponen de conocimiento específico del problema), codificada (no trabajan directamente sobre el dominio del problema, sino sobre representaciones de sus elementos), múltiple y paralela (esto es, buscan simultáneamente entre un conjunto de candidatos). Todo ello (Golberg, 1989) para proporcionar una robustez a la búsqueda; es decir, para darle más eficiencia sin perder generalidad y viceversa. Tradicionalmente, la característica global más importante de un método de búsqueda es su “fortaleza”. Ésta se mide en términos de eficiencia: Un método es tanto más fuerte cuanto más eficientemente lleve a cabo su tarea. Ahora bien, un postulado fundamental de la teoría de la optimización, afirma que la eficiencia y la generalidad son objetivos contrapuestos; es decir, cuanto más eficiente sea un método, menor número de problemas será capaz de resolver y viceversa. Los métodos deterministas y heurísticos son los más eficientes, pero los menos generales. Esto llevó a plantear la siguiente cuestión: Admitiendo que para los procedimientos tradicionales de búsqueda la relación eficiencia “versus” generalidad es inevitablemente constante, ¿existen otros procedimientos de búsqueda para los que dicha

relación sea mayor?. Esto llevó a cuestionar el papel fundamental otorgado a la "fortaleza" para evaluar la calidad de un método de búsqueda e introducir una nueva "ratio" para los métodos de búsqueda denominada "robustez". De este modo, se dice que un método es tanto más robusto cuanto mayor sea la relación eficiencia \times generalidad. Para lo cual ha sido necesario relativizar las exigencias de fortaleza y aprovechar las buenas cualidades en términos de robustez que proporcionan el no-determinismo y la "opacidad", o sea, la no-utilización de conocimiento específico.

Sin embargo, la característica esencial de los AA.GG, que no se observa directamente, es su capacidad de intercambio estructurado de información en paralelo o abreviadamente su "paralelismo implícito". De modo intuitivo, esta característica se refiere a que los AA.GG procesan externamente cadenas de códigos; sin embargo, lo que se está procesando internamente son similitudes entre cadenas y de manera tal que al procesar cada una de las cadenas de la población se están procesando a la vez todos los patrones de similitud que contienen, que son muchos más. Es precisamente por esta propiedad por lo que los AA.GG son mucho más eficientes que otros métodos de búsqueda ciega, y por tanto son más robustos con independencia de otras consideraciones.

Programas Evolutivos. Aquí se hace evolucionar una población de estructuras de datos sometiénolas a una serie de transformaciones específicas y a un proceso de selección. En efecto, ante la debilidad intrínseca de los AA.GG, básicamente debidos a su ceguera o codificación para la búsqueda, Michalewicz (1994) propuso una estrategia de

incorporación directa del conocimiento específico del problema por su base: La representación. En efecto, con el fin de obtener la máxima generalidad y el máximo grado de paralelismo implícito, se exige a los AA.GG que efectúen la representación de la población a través de cadenas binarias. Sin embargo, esta representación es tan rígida en la práctica que merece la pena considerar otras más flexibles aún a costa de perder algo de generalidad y paralelismo implícito. A fin de cuentas, una “representación directa” tan solo precisa información genérica, por lo que las consecuencias prácticas de pérdida de generalidad no serán muy grandes; por otra parte, existe la posibilidad de compensar la pérdida de paralelismo implícito procesando más información útil; es decir, de compensar la violación del principio del “alfabeto de símbolos mínimo” dando más importancia al principio de los “bloques constructivos con significado”.

Justamente a las implantaciones que resultan de ese enfoque se les denomina “Programas Evolutivos” que comparten punto por punto la estructura básica de los AA.GG, distinguiéndose de éstos solamente en el nivel interno por el uso de, por una parte, una representación lo más natural posible; esto es, próxima al dominio del problema que, además, sea útil a lo que es lo mismo significativa para el problema que se va a resolver; y, por otra, operadores genéticos específicos “sensibles al contexto”; esto es, adaptables al dominio, y lo más cerrados posibles.

Estrategias Evolutivas (EE). En este caso se hace evolucionar una población de máquinas de estados finitos someténdolas a transformaciones unitarias. Dichas estrategias, surgieron, a partir de los trabajos de Rechenberg (1973) y Schwefel (1975), como métodos estocásticos de

escalada con paso adaptativo, específicamente diseñados para resolver problemas de optimización paramétrica. Con el paso del tiempo, fueron incorporando procedimientos propios de la computación evolutiva hasta convertirse en un paradigma más de dicho enfoque. Actualmente, las EE son algoritmos evolutivos enfocados preferentemente hacia la optimización paramétrica que emplean una representación a través de vectores reales, selección determinista y operadores específicos de mutación y cruce, el cual ocupa ahora un lugar secundario respecto a la mutación. Las EE son el enfoque más fuerte, es decir, más específico, de la CE; sin embargo, mantienen la filosofía de las “cajas negras” y operadores ciegos.

Dos ejemplos de EE, son el $(\mu + \lambda)$ - EE y (μ, λ) EE. En el primero, μ padres producen λ descendientes; entonces la población se reduce de nuevo a μ padres al seleccionar las mejores soluciones entre ambos, los padres u y descendientes. De este modo, los padres sobreviven hasta que son reemplazados por mejores soluciones. El segundo, es decir, (μ, λ) EE, es más próximo al modelo generacional usado en AA.GG canónicos; la descendencia reemplaza a los padres y luego se realiza la selección. Los operadores de recombinación para estrategias evolutivas también tienden a diferir del estilo Holland, permitiendo operaciones tales como ponderación de parámetros, por ejemplo, crear un descendiente.

Las EE pueden ser simples; esto es de dos miembros, que en pureza no son métodos evolutivos propiamente dichos, y múltiples que sí son métodos evolutivos.

Aunque en su origen las EE y los AA.GG surgieron, como acaba de indicarse, de enfoques distintos y distantes, los sucesivos desarrollos de ambos han hecho que convergieran hasta tal punto que actualmente una EE puede considerarse como un tipo especial de AG, y viceversa, distinguiéndose sólo en el uso de unos u otros procedimientos por defecto. Con todo, algunas diferencias entre ambos son claras, como son las siguientes:

Las EE son intrínsecamente más fuertes que los AA.GG; es decir, requieren para su funcionamiento más conocimiento específico que los AA.GG, los cuales son, por construcción, algoritmos de búsqueda de propósito general.

El campo natural de aplicación de las EE son los problemas de optimización paramétrica, el de los AA.GG la optimización de atributos. Ello no quiere decir que no se puedan usar las EE para resolver problemas combinatorios; pues siempre se pueden resolver dichos problemas indirectamente, utilizando una codificación como la de las clases aleatorias o, directamente, mediante EE específicas.

En los AA.GG, la clase de búsqueda está en los operadores genéticos de recombinación (cruce), en las EE en los de alteración (mutación).

Por defecto, el criterio de reemplazo de los AA.GG es estocástico, en las EE determinista. En ambos casos, se trata de mantener un nivel óptimo de aleatoriedad frente al determinismo.

En las EE el tamaño de la descendencia está fijado de antemano, en los AA.GG sólo está fijado su promedio a través de la tasa de cruce. Además, en las EE todos los individuos son sometidos a mutación, la cual es más drástica para unos individuos y menos para otros, según como sea el vector de dispersiones; en los AA.GG, sólo unos pocos individuos son sometidos a mutación, la cual es igual de drástica para ellos.

En las EE el tratamiento de los puntos no factibles se lleva a cabo implícitamente mediante filtrado; en los AA.GG, requiere un tratamiento específico. Además, en los AA.GG la técnica de filtrado de las soluciones no factibles no es usual ni provechosa. De hecho, se usa muy poco, en las EE da buenos resultados debido a la capacidad de autoajuste de los parámetros evolutivos.

Los AA.GG son mecanismos de búsqueda preferentemente global; es decir, tienen dificultades para el ajuste fino, con las EE sucede justo al contrario.

Programación Evolutiva. La Programación Evolutiva está basada en el primer libro de Fogel, Owens y Walsh (1966) titulada "Inteligencia Artificial hacia la evolución simulada". Los individuos u "organismos", en este estudio son máquinas de estado finitas. Los organismos resuelven alguna función objetivo obteniendo la oportunidad de reproducirse. Los padres son mutados para crear hijos. Esto adquirió un interés renovado en Programación Evolutiva como se reflejó en 1992 en la Primera Conferencia Anual sobre Programación Evolutiva (Fogel and Atmar, 1992).

En este enfoque, se hace evolucionar una población de máquinas de estados finitos sometiénolas a transformaciones unitarias, haciendo uso de dos operadores genéticos: la selección y la mutación. Ésta realiza cambios más o menos severos en la estructura de las máquinas de acuerdo con cierta distribución. Aquella; esto es, la selección se hace por torneo o de modo determinista. No se exige que el tamaño de la población sea constante, tan sólo que cada individuo genere a lo sumo un descendiente para la próxima generación.

Programación Genética. A principios de la década de los noventa del pasado siglo, Koza (1992) propuso utilizar la técnica de la CE, no para encontrar soluciones a los problemas, sino para encontrar el mejor procedimiento con que resolverlos. Dicho de otro modo, las estructuras que se someten a evolución no codifican las posibles soluciones a un problema, sino algoritmos o programas que, al ejecutarse, determinan dichas soluciones. Este enfoque se le conoce como “Programación Genética” (PG).

Para llevar acabo esta idea, el enfoque más adecuado es el de los programas evolutivos. La representación de los individuos se hace mediante unas estructuras de datos jerárquicamente enlazadas denominadas, árboles de análisis, en inglés “parse trees”; la codificación utiliza un conjunto de símbolos de función y otro de símbolos terminales adecuados para la solución de problemas en el dominio de interés. La función de aptitud mide la precisión de cada individuo al ejecutar una serie de ejemplos para los que la solución es conocida de antemano. En cuanto a los operadores genéticos, se usan los clásicos, adaptados para trabajar eficientemente con árboles de análisis y otras representaciones específicas del enfoque. La PG se usa

principalmente para resolver problemas de aprendizaje artificial, aunque también ha dado buenos resultados en la solución de ciertos problemas combinatorios.

Autómatas reproductores

La primera aportación de las ciencias a la genética fue la disertación doctoral de Claude E. Shannon, el padre de la teoría de la información, en 1940, cuyo título fue "An Algebra for Theoretical Genetics" (Shannon, 1940). En dicha disertación, presentaba un álgebra especialmente deseable para estudiar los problemas de las dinámicas de las poblaciones mendelianas. Esta álgebra, trataba dichos problemas sólo a nivel de gen, lo que, como se mostrará en este trabajo, no resulta suficiente, para entender el código genético.

De la informática a la genética, la sin duda aportación más espectacular, por su carácter premonitorio, fue la teoría de autómatas reproductores de von Neuman; es decir, el construir modelos formales de autómatas capaces de reproducirse. En efecto, von Neumann en el año 1948 pronunció una conferencia titulada "La teoría Lógica y General de los Autómatas" (Von Neumann, 1951, 1963) donde estableció los principios de cómo una máquina podría autorreproducirse. El procedimiento sugerido por von Neumann se consideró en un primer momento más que nada como una curiosa especulación lógico-matemática. Sin embargo, la visión de von Neumann sobre la forma en que se reproducen los seres vivos; por otra parte, en abstracto, más sencilla de los que pueda parecer, recibió un

espaldarazo, al confirmarse, cinco años después de su conferencia, cuando Watson y Crick (1953(a)) encontraron el modelo de ADN.

Fue a partir de los años cincuenta cuando la Teoría de la Información (en adelante TI), ejerció una notable influencia en biología, como por otra parte, en otros muchos campos alejados del ámbito estrictamente matemático. Y es justo a partir de esta fecha cuando una parte importante de las ciencias de la vida incorpora en su acervo conceptos propios de la TI. En efecto, toda la información necesaria para la génesis y desarrollo de la vida de los organismos se encuentra en la secuencia de bases de las largas cadenas de ADN. Sus instrucciones aparecen codificadas según un alfabeto de cuatro letras A, T, C y G. Un texto formado por palabras escritas con esas cuatro letras constituye la información genética de cada organismo vivo. La existencia del código genético, que fue conjeturada en 1944 por el premio Nobel de física Erwing Schrödinger (1944), fue puesta de manifiesto nueve años más tarde, por James Watson y Francis Crick (1953(a), (b)), ambos premios Nobel por dicho descubrimiento. En el intermedio, en 1948, es cuando von Neumann estableció como una máquina podía reproducirse.

Autorreproducción

Butler en su novela utópica "Erewhon", palabra que, en inglés, es la imagen especular de "nowhere" que significa en ninguna parte, plantea la posibilidad de que las máquinas utilicen a los hombres como intermediarios para construir nuevas máquinas. En el siglo pasado, hubo muchos casos de

máquinas construidas por máquinas. Se utilizaban motores de vapor para hacer otros motores de vapor, y las máquinas herramientas hacían todo tipo de aparatos. Mucho antes de la era de la computadora, había máquinas mecánicas y eléctricas que ayudaban a construir motores moldeando el metal. La Revolución Industrial fue en gran parte posible gracias a la intervención de máquinas-herramientas, máquinas, en otras palabras, que estaban concebidas con la única finalidad de construir otras máquinas. Sin embargo, los aparatos que sirven para hacer otros aparatos no son exactamente el tipo de máquinas-herramientas en que pensaba el primer ministro inglés Disraeli: "El misterio de los misterios es que las máquinas hagan máquinas". Es decir, máquinas para construir otras máquinas, en las que la intervención humana no existiera. Dicho de otro modo, máquinas que se reproduzcan, o, si se quiere emplear un término menos controvertido, se repliquen.

¿Cuál es el significado de la palabra reproducción? Como lo señaló Kemeny (1955), si se entiende por ello la creación de un objeto igual al de origen a partir de la nada, es evidente que una máquina no puede reproducirse, pero tampoco un hombre podría hacerlo. Para que la reproducción no viole el principio de conservación de la energía es preciso disponer de cierta materia prima. El hecho característico de la reproducción de la vida es que el organismo vivo es capaz de crear nuevos organismos semejantes a él a partir de la materia inerte que le rodea. Si se acepta que las máquinas no están vivas, y se insiste en el hecho de que la creación de vida es una característica fundamental de la reproducción, el problema está resuelto: una máquina es incapaz de reproducirse. Se reformula, pues, el

problema de modo que la reproducción de las máquinas no resulte lógicamente imposible. Para ello es preciso suprimir la palabra "viviente". Así, pues, se exigirá que la máquina sea capaz de crear un organismo nuevo, semejante a ella, a partir de componentes sencillos que existan en su entorno.

Desde 1948, los científicos sueñan con crear máquinas programadas para producir réplicas de sí mismas, estas réplicas producirían a su vez otras réplicas, y así sucesivamente, sin límite alguno. En aquel año, se inició la teoría que estableció los principios de cómo se podría conseguir semejante proeza. Esta teoría posee dos aspectos, que se podrían denominar lógico y mecánico, éste lo trataron, entre otros, Lionel y Roger Penrose (1974) y aquí no se va a considerar. La parte lógica, que es lo que aquí y ahora interesa, fue investigada en primer lugar por von Neumann en el Instituto for Advanced Study de Princeton. Allí y entonces, von Neumann indicó la posibilidad de construir un dispositivo que tuviese la propiedad de autorreproducirse. El método consistía en construir otra máquina descriptible, y de aquí se seguía lógicamente que esta máquina llevaría una especie de cola en la que iría incluido el código que describe la forma de reproducir el cuerpo de la máquina y la forma de reproducir el propio código. Según Kemeny (1955), colaborador de von Neumann, el cuerpo básico de la máquina consistiría en una caja conteniendo partes constituyentes, al que se le añadiría una "cola" en la que estarían almacenadas las unidades de información. Desde el punto de vista mecánico, se consideraba que las partes elementales, a partir de las que debería construirse la máquina, eran rollos de cinta, lápices, gomas de

borrar, tubos de vacío, cuadrantes, células fotoeléctricas, motores, baterías y otros dispositivos semejantes. La máquina montaría estas partes a partir de la materia prima en torno a ella, organizándolas y transformándolas en una reproducción de sí misma. Puesto que el propósito de von Neumann era resolver las condiciones lógicas del problema, las increíbles complicaciones materiales de éste se dejaban de lado por el momento.

Aspecto lógico de la autorreproducción

La producción masiva de un gran número de ejemplares de un mismo objeto por impresión o combinación de partes prefabricadas es algo tan corriente que inmediatamente se podrían citar muchos ejemplos de ello. El lector tiene ante sí mismo una prueba en el ejemplar del artículo en que está leyendo estas palabras. Sin embargo, la idea de un objeto que se reproduce a sí mismo está tan fuertemente asociada a los procesos fundamentales de la biología que trae consigo sugerencias mágicas. En efecto, la construcción de una máquina capaz de reproducirse a sí misma podría ser considerada como imposible y perteneciente a la categoría de objetos del tipo de los motores perpetuos. Sin embargo, en estos momentos, los adelantos de la genética han llevado rápidamente a los biólogos a centrar su atención en la estructura y función de las cadenas moleculares autorreproductoras que, desde el núcleo de las células vivas, presiden todas sus actividades funcionales. La estructura de estas cadenas de ácidos nucleicos está muy estudiada, pero no así la teoría general de la autorreproducción, de la que la duplicación de los ácidos nucleicos constituye un caso particular.

Von Neumann simplificó el problema haciendo una serie de hipótesis razonables. La primera de ellas es que no era importante que la máquina fuera capaz de moverse, siendo suficiente el que poseyera un mecanismo capaz de organizar su entorno empleando el control remoto. En segundo lugar, suponía que el espacio estaba dividido en celdas cúbicas, y que cada parte de la máquina y cada fragmento de materia ocupaba precisamente una celda. En tercer lugar, se suponía que el proceso estaba cuantificado no solamente en cuanto al espacio, sino también en cuanto al tiempo; es decir, que existían ciclos y que a lo largo de ellos tenían lugar todas las acciones. Ni siquiera resultaba necesario considerar tres dimensiones, una red bidimensional era tan útil como una red de cubos.

El espacio consistirá entonces en una hoja muy grande, en principio infinito, dividida en cuadrados. Una máquina ocupa un área conexa formada por un número elevado de cuadrados y puesto que cada cuadrado representa un elemento de la máquina, el número de cuadrados ocupados proporciona una medida de la complejidad de la máquina. La máquina está rodeada de celdas inertes, que deberá organizar. Para que esto sea factible, la máquina deberá ser una combinación de "cerebro" y "músculo", puesto que no sólo tiene que organizar la materia, sino también transformarla. De acuerdo con esto, la máquina de von Neumann está formada por tres clases diferentes de partes. Posee "neuronas", similares a las del sistema nervioso, que constituyen el control lógico. Existen también celdas de transmisión, que llevan mensajes a partir de los centros de control. Estas celdas poseen una entrada, a través de la que reciben los impulsos, y una salida, por la que pasa el impulso en el ciclo siguiente. Una serie de celdas de transmisión,

adecuadamente colocadas, forman la cadena a través de la que se transmiten los mensajes. Además, la máquina tiene "músculos", encargados de modificar las celdas que los rodean, bien para aumentar su grado de complejidad o bien para destruirlas. Los cambios que tienen lugar son análogos a los que tienen lugar en el cuerpo humano a causa de la combinación de las acciones químicas y musculares. Su fin principal es el de convertir las celdas inertes en partes de la máquina.

Al igual que en el sistema nervioso, la operación se desarrolla por etapas: el estado de cada una de las celdas está determinado por su propio estado y por el de las más próximas, ambos referidos al ciclo anterior. Las neuronas y las celdas de transmisión pueden permanecer en reposo, o bien enviar un impulso, caso de ser adecuadamente estimuladas. Las celdas "musculares" reciben órdenes enviadas por las neuronas a través de las celdas de transmisión, y reaccionan "matando" alguna parte indeseable, es decir, convirtiéndola en inerte, o bien transformando alguna celda inerte del entorno de la máquina en una parte específica de ella. Se ve que la estructura de esta máquina es bastante similar a la de un animal superior. El conjunto de sus neuronas forma el sistema nervioso central; las celdas de transmisión establecen contactos adecuados con los diversos órganos; y los órganos realizan las tareas que se les encomienda después de haber recibido la orden correspondiente.

Las instrucciones pueden ser muy extensas, lo que significa que deberán ser de alguna manera, externas. La máquina de von Neumann posee, tal como se ve en la figura 2, una "cola" que contiene el proyecto que tiene que llevar a cabo. Esta cola está formada por una banda muy larga,

compuesta de instrucciones codificadas. El cuerpo básico realiza dos tipos de funciones: sigue las instrucciones de la cola y, a la vez, es capaz de reproducirla. Supóngase que la cola contiene una descripción codificada del cuerpo básico. De este modo, el cuerpo, siguiendo las instrucciones, construirá otro cuerpo igual. Cuando haya terminado, empezará a reproducir su propia cola, uniéndola al nuevo cuerpo. De este modo, se habrá reproducido a sí mismo.

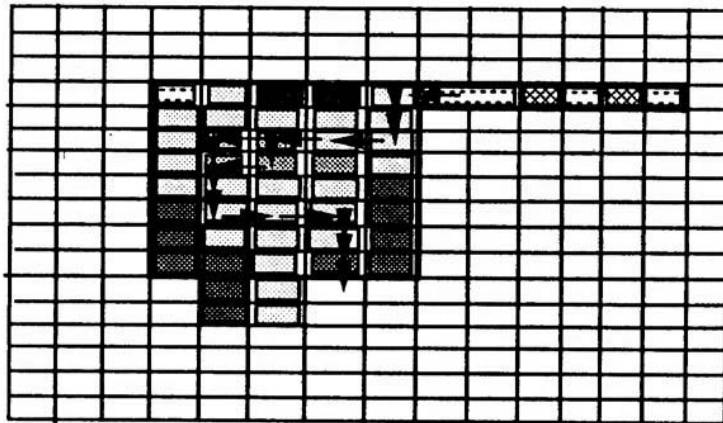


Figura 2: Esquema de la máquina autorreproducción de von Neumann.

El secreto de la máquina está en que no intenta autorreproducirse. Von Neumann diseñó una máquina que podría reproducir cualquier otra máquina a partir de una descripción de ella. Por tanto, tal máquina podrá autorreproducirse. Una vez hecho esto, resulta sencillo reproducir la cola y unirla a la descendencia. La nueva máquina podrá continuar

reproduciéndose hasta que se agote el material o hasta que estas nuevas máquinas entren en conflicto entre sí, imitando en esto a sus "padres" humanos.

Resulta asombroso el pequeño número de partes que posee una máquina de este tipo. En los proyectos de von Neumann se habla de un cuerpo básico de 80 por 400 cuadrados, al que se añade una cola de 150.000 cuadrados de longitud. El cuerpo básico posee las tres clases de celdas ya descritas: neuronas, celdas de transmisión y celdas musculares. Estos tres tipos de celdas se diferencian entre sí únicamente en su estado de excitación y en la forma en que están conectadas entre sí. La cola es todavía más sencilla; está formada por celdas en posición de "encendido" o "apagado", proponiendo el código. Se tiene, pues, unas 200.000 celdas, la mayor parte de las cuales son del tipo más sencillo posible, mientras que las neuronas encargadas del control lógico figuran en una proporción despreciable. Cualquiera que sea el criterio empleado para medir la complejidad, ésta es muchísimo menor que la de un ser humano, y, sin embargo, la máquina es capaz de reproducirse.

Continuando con esta analogía entre la máquina y el organismo humano, es posible comparar la cola con el conjunto de los cromosomas. La máquina reproduce su propia cola del mismo modo que las células "hijas" copian los cromosomas de la célula padre. Resulta sumamente significativo el hecho de que, mientras que los cromosomas ocupan solamente una parte muy pequeña del cuerpo, la cola es mayor que el cuerpo básico de la máquina. Esto indica que la codificación realizada por los cromosomas es enormemente eficiente. Pero, para decirlo todo, hay que reconocer que los

cromosomas desempeñan un papel menos importante que la cola, que contiene una descripción completa del cuerpo básico, en tanto que la descripción cromosómica es incompleta, puesto que los hijos se parecen a sus padres, pero no constituyen una reproducción de ellos.

Sería, por otra parte, muy interesante continuar los trabajos de von Neumann y construir una máquina que fuese capaz de manejar una descripción incompleta y construir otras máquinas más o menos semejantes a ella. ¿Son capaces estas máquinas de experimentar un proceso evolutivo? Cabría diseñar las colas de tal modo que en cada ciclo tenga lugar un pequeño número de cambios aleatorios, por ejemplo, basta cambiar "encendido" por "apagado" en el código. Esto sería algo semejante a las mutaciones: si la máquina siguiese siendo capaz de tener descendencia, dichas modificaciones pasarían a ésta. Cabría también disponer las cosas de tal modo que la provisión de materias primas sea limitada, de suerte que las máquinas deban luchar por el espacio vital, pudiendo llegar a matarse entre sí.

La propuesta de von Neumann, para construir máquinas que tuvieran la capacidad reproductora de los organismos vivos, se consideró, en un primer momento, más que nada una curiosa especulación matemática, especialmente si se tiene en cuenta que las computadoras de aquellas fechas eran gigantes de 30 o más toneladas, y eran poco más que aparatos para hacer operaciones matemáticas con rapidez. ¿Cómo podría conseguirse que una máquina produjera una copia de sí misma?. Una manera que resulta imposible es recibiendo la orden de su programador humano: "¡Reprodúctete!". La máquina sólo puede responder "No puedo

reproducirme, pues no sé quién soy". Este planteamiento sería tan absurdo como si un hombre le diera a su esposa una serie de botellas y matraces y le dijera que hiciera un niño. En opinión de von Neumann, el programador humano que se proponga crear una dinastía de máquinas debería realizar las tres sencillas acciones siguientes:

Dar a la máquina una descripción completa de sí misma.

Dar a la máquina una segunda descripción de sí misma, pero esta segunda descripción es de una máquina que ha recibido ya la primera descripción.

Finalmente, ordenar a la máquina que cree otra máquina que corresponda precisamente a la máquina de la segunda descripción, y ordenar a la primera máquina que copie y pase esta orden final a la segunda máquina.

Lo más notable de este procedimiento lógico es que, además de ser mucho más sencillo de lo que puede parecer, era la visión de von Neumann sobre la forma en que se reproducen las criaturas vivas. Unos años después de su conferencia, sus ideas se vieron confirmadas cuando los biólogos Francis Crick y James D. Watson (1953 (a) y (b)) encontraron la solución al código genético y descubrieron el secreto de la reproducción orgánica. Resultó ser esencialmente el mismo que la secuencia para la reproducción de la máquina propuesta por von Neumann. En los seres vivos, el ácido desoxirribonucleico (ADN) desempeña el papel de la primera máquina. El ADN da instrucciones al ácido ribonucleico (ARN) para la construcción de

proteínas; el ARN viene a ser así como un "auxiliar" del ADN. Mientras que el ARN realiza la aburrida tarea de elaborar proteínas para su organismo paterno y su descendencia, el ADN hace el papel brillante e imaginativo de programar sus genes, que, en el caso de un bebé humano, decidirán si tendrá pelo rubio o moreno y si será de temperamento excitable o flemático. En resumen, el ADN y el ARN realizan juntos todas las tareas que debe realizar la primera máquina von Neumann para crear la segunda máquina de la dinastía. Y por eso, si se decide construir máquinas autorreproductoras, hay pruebas biológicas importantes de que von Neumann dio, hace tiempo, con el procedimiento correcto que se debe seguir.

Pero se puede preguntar ¿por qué va a querer alguien construir computadoras que realicen réplicas de sí mismas?. En el mejor de los casos, el procedimiento puede resultar engorroso. Supóngase que alguien se acuesta después de pasarse la tarde trabajando con la computadora y, al levantarse al día siguiente, se encontrara con que había dos computadoras en vez de una. ¿Para qué servirán estas computadoras que se regeneran?. La respuesta es que se utilizarán en sitios remotos para realizar tareas difíciles y peligrosas que los hombres no pueden realizar con facilidad. En consecuencia, hay que considerar totalmente la posible localización de estos lugares. ¿Qué es lo que está reteniendo el desarrollo biológico humano? ¿Por qué más de treinta y cinco años después de que el hombre pusiera el pie por primera vez en la Luna, no hay todavía una colonia lunar permanente?. Han pasado más de tres cuartos de siglo desde que el hombre consiguiera volar por primera vez, y todavía la mayoría de los seres

humanos están constreñidos a vivir en la superficie de la Tierra. ¿Por qué?. El astrónomo Tipler, (1980) de la Universidad de California en Berkeley, respondió a esta pregunta muy claramente cuando en 1980, declaró: "Es el retraso en la tecnología de las computadoras, no en la tecnología de los cohetes, lo que impide comenzar mañana mismo la exploración de la Galaxia". Es en el espacio, no en la Tierra, donde se harán valer las máquinas superinteligentes autorreproductoras y es en el espacio donde está el futuro a largo plazo de la humanidad. Es fascinante considerar la forma en que Tipler y otros que han estudiado el futuro remoto consideran cómo las máquinas de von Neumann harán posible, en primer lugar la colonización del sistema solar de planetas, y luego de la Vía Láctea con más de 150.000 millones de soles.

Como es natural, ninguna de las máquinas de las que aquí se habla ha sido construida, al menos que se sepa, pero nada impide que sea posible construirlas. Dicho con otras palabras, científicamente nada impide su construcción, otra cuestión y ésta sí que es peliaguda, es que tecnológicamente sea viable el realizar réplicas.

El código genético

Código versus Cifra

Técnicamente, se define un código como una sustitución al nivel de las palabras o las frases, mientras que una cifra se define como una sustitución al nivel de las letras. Por eso, cifrar significa ocultar un mensaje

utilizando una cifra, en tanto que codificar significa ocultar un mensaje empleando un código. De un modo similar, el término descifrar se aplica a la resolución de un mensaje cifrado; es decir, en cifra, mientras que el término descodificar se aplica a la resolución de un mensaje codificado. Como puede verse, los términos codificar y descodificar son más generales, y tienen relación tanto con códigos como con cifras. Por eso no hay que confundir, por un mal uso, estos términos. Así, por ejemplo, cuando se habla del código Morse, el que traduce las letras del alfabeto inglés y algunos signos de puntuación en forma de puntos y rayas, no es un código en el sentido de que no es una forma de criptografía porque no hay una ocultación del mensaje. Los puntos y las rayas son simplemente una forma conveniente de representar las letras para el medio telegráfico. En realidad, el código Morse no es otra cosa que un alfabeto alternativo.

Formalmente, se define código como sigue. Sea A^* un monoide libre (Hu, 1965) engendrado por el conjunto A . Es decir, A^* es el conjunto de palabras de longitud finita formadas con los símbolos de A , mediante la ley de composición interna asociativa de concatenación, y con un elemento neutro que es la palabra vacía. Entonces, un código $C = \{c_1, \dots, c_i\}$ es un subconjunto de A^* . Siendo los elementos de c_i las palabras del código. Se indica por n_i el tamaño de la palabra c_i .

Se dice que un código es binario, ternario, en general n -ario, cuando A está formado por, respectivamente, dos, tres, en general n símbolos. Si todas las palabras son de la misma longitud, se dice que C es un código de

longitud fija, o también un código de bloque. En otro caso, se dice que C es un código de longitud variable.

Sea σ el alfabeto de una fuente de información, que se define como aquella que emite símbolos de un alfabeto $\sigma = \{s_1, \dots, s_r\}$ con una probabilidad de aparición de cada uno de ellos dada por $\pi_i (1 \leq i \leq r)$. Entonces una codificación es una aplicación φ de σ en C , que se prolonga en una aplicación φ^* de σ^* en C^* . Y, naturalmente, una decodificación es una aplicación ψ de C^* en σ^* . Un código es de decodificación única, es decir, φ^* es inyectiva, y ψ o ψ^* es la identidad.

Código genético

Definición

La expresión "código genético" se utiliza actualmente con dos significados muy distintos. Por una parte, el público profano lo emplea a menudo para nombrar el mensaje genético completo de un organismo. Por otra, los biólogos moleculares aluden al pequeño diccionario que enseña cómo relacionar el lenguaje de cuatro letras de los ácidos nucleicos con el lenguaje de veinte letras de las proteínas, de la misma forma en que el "código" Morse relaciona el lenguaje de puntos y rayas con las veintiséis letras del alfabeto. Aquí se utilizará el término en este sentido. Sin embargo, el término técnico para una regla de traducción así no es, hablando con propiedad, "código" sino cifra como así lo llamó Erwin Schrödinger (1983).

De la misma forma en que el código Morse debería llamarse cifra Morse. En aquel tiempo, Crick o no lo sabía, o no hizo caso de esta distinción, lo que fue una suerte, ya que “código genético” suena mucho mejor que “cifra genética”. En efecto, Crick al conjunto de bases contenidas en los cromosomas, las denominó correctamente texto cifrado o clave, y añadió, pero el término “clave” o texto cifrado, es demasiado limitado. Las estructuras cromosómicas son al mismo tiempo los instrumentos que realizan el desarrollo que ellos mismos pronostican. Representan tanto el texto legal como el poder ejecutivo, o para usar otra comparación, son a la vez los planos del arquitecto y la mano de obra del constructor, o como se verá más adelante, son el "Knowledge y el Knowing".

El código se encuentra en el centro de la biología molecular de la misma forma que la tabla periódica de los elementos constituye el centro de la química, pero hay una diferencia profunda y trascendental, la tabla periódica es, con probabilidad rayana en la certeza, válida y cierta en cualquier lugar del universo. Sin embargo, si hay vida en otros mundos, y si ésta utiliza ácidos nucleicos y proteínas lo que no es seguro, parece muy probable que ahí el código fuera sustancialmente distinto. Existen incluso pequeñas variaciones con respecto a él, en algunos organismos terrestres. El código genético, como la vida misma, no es un aspecto de la naturaleza eterna de las cosas, sino, al menos en parte, producto de un accidente.

El lema central del código genético, es la relación entre la secuencia de bases del ADN o de su m-ARN transcrito y la secuencia de aminoácidos de las proteínas, del modo que la secuencia de bases de un gen es colineal con la secuencia de aminoácidos de su polipéptido producto. Este código,

que es el mismo en todos los seres vivos, es maravilloso por su sencillez. En efecto, el conjunto de tres bases o codón especifica un aminoácido. Los codones son leídos secuencialmente por las moléculas del t-ARN que sirven de adaptadoras en la síntesis de las proteínas.

Características del Código

En 1961, los experimentos de Crick y colegas (Crick, 1961) entre otros, establecieron como características del código genético las siguientes:

El alfabeto. Como ya se ha indicado se compone de cuatro elementos o bases; a saber: Adenina, Guanina, Citosina y Timina y se representan por sus iniciales A, G, C, T; es decir; $Al = \{ A, G, C, T \}$.

Relación de Codificación. Una sola base podría especificar solamente cuatro tipos de aminoácidos puesto que sólo hay cuatro tipos distintos de bases en el ADN. La primera cuestión que se puede plantear aquí se enunciaría como sigue: ¿Por qué se emplean cuatro bases en el ADN? Si bien no es posible dar una respuesta fiable a esta cuestión, si es posible acotar el espacio de posibilidades. En efecto, es bastante evidente que el número de bases distintas tiene que ser par, puesto que, en otro caso, cualquier esquema de emparejamiento de bases conllevaría una pérdida de información que haría imposible la replicación de moléculas de ARN tal y como se conoce hoy. Dos bases podrían especificar dieciséis tipos ($4 \times 4 = 16$) de aminoácidos y tres bases podrían determinar ($4 \times 4 \times 4 = 64$) sesenta y cuatro clases de aminoácidos. Puesto que las proteínas están formadas por un conjunto básico de veinte aminoácidos, resulta evidente que son

necesarias al menos tres bases para especificar los veinte aminoácidos. Esta evidencia teórica de teoría de la información y matemáticas, fue refrendada por experimentos genéticos que demostraron que, de hecho, un grupo de tres bases codifica un aminoácido. A este grupo de tres bases se le denomina, como acaba de decirse, codón o triplete.

Parece poco probable que la evolución haya atendido a los criterios de optimidad de la TI para la elección de codones de tres bases. El que se tenga un código tal, se debe, probablemente, más a criterios estructurales, electroquímicos y físicos aplicados sobre las moléculas que intervienen en los procesos descritos. También resulta difícil creer que pueda haber subyacente un criterio de optimidad del código. En resumen, la selección natural no va a actuar tanto sobre la optimidad del código que transmite la información, característica ésta que no aporta ningún beneficio físico, sino más bien sobre la eficiencia de las moléculas que realizan la transcripción; esta eficiencia sí se traduce en beneficios concretos, tales como mayores cantidades de moléculas en menos tiempo.

En efecto, por teoría de la información se sabe, que si se tiene un alfabeto A de b símbolos que genera mensajes de longitud n, de los que se tiene un desconocimiento total, la cantidad de información I recibida con cada mensaje viene dada por:

$$I = \log bn = n \log b \quad (1)$$

Es decir, la longitud del mensaje por el logaritmo de los estados posibles, en este caso, las variaciones con repetición de n elementos

tomadas de m en m . Naturalmente, es fácil observar que la cantidad de información aumenta al aumentar el número de mensajes entre los que la fuente puede escoger, y disminuye a medida que la libertad de elección y, por consiguiente, el grado de incertidumbre disminuye.

Convencionalmente se ha elegido como unidad de medida de la indeterminación, la de un experimento con dos posibles resultados equiprobables, ya que si sólo existiera un único resultado de indeterminación sería nula y no habría información acerca del fenómeno. Según esto, para que:

$$\log_2 = 1 \quad (2)$$

Se cumpliera, es necesario elegir como base de los logaritmos la 2, que es la única que verifica la ecuación (2).

Considérese ahora el alfabeto anterior A de b símbolos. El número de estados E necesarios para representar un mensaje de n símbolos es:

$$E = n.b \Rightarrow n = \frac{E}{b} \quad (3)$$

Es fácil de demostrar que esto es así como sigue: Supóngase que se quiere representar un mensaje de tres símbolos: x, y, z ; entonces, el mensaje

tendría que tener al menos tres elementos distintos capaz de almacenar cada uno de los símbolos, verbigracia:

x y z

Además, como se está en un sistema posicional, el mismo elemento es diferente dependiendo de la posición en que se encuentre. Por ejemplo el mensaje x x x también necesita los tres elementos y además sus valores son distintos. Entonces, se define como estados de un sistema de mensajes la cantidad de elementos diferentes necesarios para representar un mensaje y que viene dada por (3).

Sustituyendo el valor de n dado por la ecuación (3) en la (1), se tiene que:

$$I = E \cdot \frac{\log b}{b} \quad (4)$$

Puesto que se quiere que la cantidad de información con respecto a la base sea máxima, hay que calcular la primera derivada de I con respecto a b y hacerlas igual a cero; es decir:

$$\frac{\partial I}{\partial b} = E \cdot \frac{1/b \log e \cdot b - \log b \cdot 1}{b^2} = E \frac{\log e - \log b}{b^2} = E \cdot \frac{\log \frac{e}{b}}{b^2}$$

Y ahora igualando a cero esta derivada se obtiene que:

$$E. \frac{\log \frac{e}{b}}{b^2} = 0$$

De donde:

$$\log \frac{e}{b} = 0 \Rightarrow \frac{e}{b} = k^0 \Rightarrow \frac{e}{b} = 1 \Rightarrow b = e$$

Con lo que se ve que la función dada en 4 tiene un máximo o un mínimo para el valor $b=e$, para determinarlo, se calcula la derivada segunda:

$$\frac{dI}{db^2} = E. \frac{(0 - \frac{1}{b} \cdot \log e) \cdot b^2 - (\log e - \log b) \cdot 2b}{b^4} = E. \frac{-3 \log e + 2 \log b}{b^3}$$

cuyo valor para $b = e$ es:

$$E. \frac{-3 \log e + 2 \cdot \log e}{e^3} = E. \frac{-\log e}{e^3}$$

puesto que el número de estados e es necesariamente un valor positivo, el valor de la derivada segunda es negativo. Por lo tanto, se concluye que para $b = e$ la función I tiene un máximo.

La demostración es absolutamente general con independencia del tipo de logaritmos que se tomen pues el paso de un logaritmo a otro es isomórfico y el resultado es, en consecuencia, único. Pero la base e no es entera y, por lo tanto, poco práctica a efectos, en este caso, del mundo biológico. Como el número más próximo al e es el 3 se puede concluir que la base óptima es la 3 y consecuentemente el alfabeto óptimo es el que tiene tres símbolos.

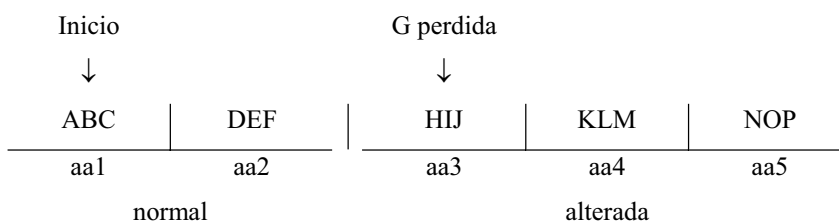
Solapamiento. En un código de ternas no superpuestas, cada codón especifica solamente un aminoácido, mientras que en otro de ternas superpuestas ABC especifica el primer aminoácido, BCD el segundo, CDE el tercero y así sucesivamente. Estas alternativas fueron discriminadas por estudios de la secuencia de aminoácidos en mutantes. Supóngase que la base C muta a C'. En un código sin superposición sólo se cambiará el aminoácido 1. En cambio, en un código completamente solapado por una mutación de C en C', se alterarán los aminoácidos 1, 2 y 3. Estudios de la secuencia de aminoácidos en la cubierta proteica de mutantes del virus del mosaico del tabaco, indicaron que, normalmente, sólo se alteraba un aminoácido. Llegándose a la conclusión de que el código genético no presenta solapamiento.

Secuencialidad en la lectura de los codones. La secuencia de bases se lee secuencialmente a partir de un punto fijo de partida. No hay comas. Esto es, el código genético no requiere puntuación ni señal alguna para indicar el final de un codón o el principio del siguiente. Existiendo sólo una "señal de iniciación" en el ARN indicativa de donde se debe iniciar la lectura. Esta señal es el codón AUG que codifica el aminoácido metionina. Esto, es

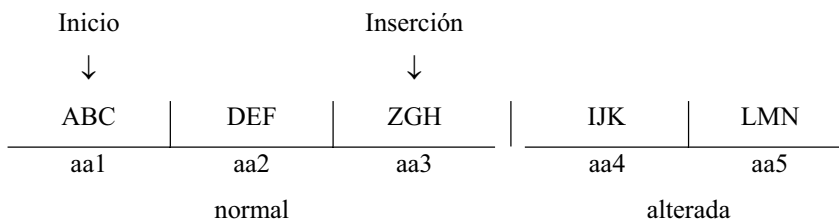
decir, la no-existencia de puntuación, presenta problemas. En efecto, supóngase que de la secuencia:

ABCDEFGHIJKL

se pierde, por mutación, la G, entonces la secuencia quedaría como sigue:



En esta cadena polipéptida los dos primeros aminoácidos serán normales, pero el resto de la secuencia de bases se leerá incorrectamente puesto que el patrón de lectura ha sido desplazado por el “borrado” de G. Supóngase ahora que se añade una base Z entre F y G, entonces la secuencia queda como sigue:



Esta mutación por adición también altera el patrón de lectura a partir del codón para el aminoácido. De hecho, los estudios genéticos de

recombinación de mutantes por adición y borrado, revelaron muchas de las características del código genético.

No-eficiencia o degeneración del código. Como ya se ha mencionado, hay sesenta y cuatro posibles ternas de bases y sólo veinte aminoácidos. La pregunta es inmediata. ¿Hay exactamente una terna para cada aminoácido o se codifican algunos aminoácidos por más de un codón? Los estudios genéticos revelaron que la mayoría de los sesenta y cuatro codones codifican aminoácidos. Para ser más exactos, sesenta y uno de los sesenta y cuatro ternas especifican un aminoácido particular. Así pues, para la mayoría de los aminoácidos, hay más de un “vocablo” o codón codificante. Esto, que matemáticamente quiere decir que existe una aplicación “sobreyectiva” entre los codones y los aminoácidos más las señales de empuje y terminación de cadena, es, para lo que trata este artículo, trascendental. En otras palabras, el código genético es degenerado, y, además, no es una degeneración uniforme. Ahora bien, el hecho de que el código genético sea degenerado no significa que sea imperfecto puesto que, como ya se ha dicho, no existe solapamiento.

Representaciones del código: matricial, codones generalizados y autómatas

Existen distintas maneras de representar el código genético, entre los más importantes están los siguientes: La más común en los libros de texto, tal y como se muestra en la tabla 2, es en forma de tabla o matriz (Ter-Avanesyan, 1988). También, en otras ocasiones, se usa una representación

lineal (Froelam, 1985). Sin embargo, las dos que presentan un enfoque más riguroso y formal desde una perspectiva informática son: los codones generalizados (Bozinovski, 1989) y los autómatas finitos, que se estudiarán en detalle más adelante.

1ª Posición (extremo 5')	2ª Posición				3ª Posición (extremo 3')
	U	C	A	G	
U	Fenil Fenil Leu Leu	Ser Ser Ser Ser	Tir Tir Parada Parada	Cis Cis Parada Trp	U C A G
C	Leu Leu Leu Leu	Pro Pro Pro Pro	His His Gln Gln	Arg Arg Arg Arg	U C A G
A	Ile Ile Ile Met (Inicio)	Tri Tri Tri Tri	Asn Asn Lis Lis	Ser Ser Arg Arg	U C A G
G	Val Val Val Val	Ala Ala Ala Ala	Asp Asp Glu Glu	Gli Gli Gli Gli	U C A G

Tabla 2: Código Genético en forma de Matriz de triple entrada

A) Matricial

Como puede verse en la tabla 2, el código puede representarse en forma de tabla. En ella, solamente el triptófano y la metionina están codificados por una única terna. Los otros dieciocho vienen codificados por dos o más ternas. De hecho la leucina, arginina y serina son especificados por seis codones cada una. Sin embargo, en condiciones fisiológicas normales, el código no es ambiguo: cada codón designa a un único aminoácido. En dicha tabla dada la posición de las bases en un codón es

posible encontrar el correspondiente aminoácido como sigue: Supóngase el codón 5' AUG 3' del m-ARN, entonces se entra por la A en la tabla 2, luego se va a la U y luego a la G y se encuentra la metionina.

B) Codones Generalizados

El proceso de biosíntesis de las proteínas habitualmente se describe mediante una transcripción y una traslación, nociones que, básicamente, tienen naturaleza lingüística. Para describir formalmente el proceso de biosíntesis de las proteínas, sea el lenguaje del ADN, A^* , un conjunto de todas las cadenas sobre el alfabeto del ADN, $D = \langle T, C, A, G \rangle$, y sea R^* un lenguaje m-ARN sobre el alfabeto del ARN $R = \langle U, C, A, G \rangle$. Entonces, se definen:

a) Una transcripción, como una función,

$$f: D \rightarrow R$$

dado por su grafo, definido por:

$$G(f) = \langle (T,U), (C,C), (A,A), (G,G) \rangle$$

Que es composición de dos funciones, es decir:

$$f = f_2 \circ f_1$$

dadas por sus grafos:

$$G(f_1) = \langle (T,A), (C,G), (A,T), (G,C) \rangle$$

$$G(f_2) = \langle (T,A), (C,G), (A,U), (G,C) \rangle$$

Sea ahora, el lenguaje de las proteínas P^* el conjunto de todas las cadenas sobre el alfabeto de las proteínas $P = \langle \text{Ala, Arg, Asp, Asn, Cys, Gln, Glu, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val} \rangle$, formado por los distintos aminoácidos.

b) Una traslación, puede considerarse como una función parcial,

$$g: C \rightarrow P$$

con $\text{dom } \langle g \rangle = C - \langle \text{UAA, UAG, UGA} \rangle$

donde $C = R^*$ es un conjunto de todas las ternas; es decir, cadenas sobre R de tamaño 3. Los elementos de conjunto $\text{dom } \langle g \rangle$ se denominan codones. El conjunto $\{ \text{UAA, UAG, UGA} \}$ se denomina conjunto de signos o marcas de puntuación, ya que sus elementos sirven como puntos de finalización para el proceso de traslación efectuado sobre una cadena de C^* . La aplicación g es sobreyectiva, conocida como código genético; sin embargo, es posible construir una representación que proporciona una aplicación biyectiva. Para ello, considérese el conjunto potencia $2R$ del alfabeto R, e introdúzcanse las siguientes notaciones para algunos elementos del conjunto potencia:

$$U = \{ U \}, C = \{ C \}, A = \{ A \}, G = \{ G \}$$

$$Y = \{ U, C \}, \bar{Y} = \{ A, G \}$$

$$\bar{U} = \{ C, A, G \}, \bar{C} = \{ U, A, G \}, \bar{A} = \{ U, C, G \},$$

$$\bar{G} = \{ U, C, A \}$$

$$R = \{ U, C, A, G \}$$

Sobre estos alfabetos se puede aplicar al lenguaje las operaciones de unión y concatenación, siguientes:

$A + B = \text{Unión } (A, B)$ y $AB = \text{Concatenación } (A, B)$, entonces el código genético puede expresarse como se muestra en la tabla 3.

AUG → Met UGG → Trp AUG → Ile	CCR → Pro ACR → Thr GCR → Ala GUR → Val GGR → Gly	UUY → Phe UAY → Tyr UGY → Cys CAY → His AA Y → Asn GAY → Asp	CA \overline{Y} → Gln AA \overline{Y} → Lys GA \overline{Y} → Glu	AGY + UCR → Ser UU \overline{Y} + CUR → Leu AG \overline{Y} + CGR → Arg
-------------------------------------	---	---	---	---

Tabla 3: Representación de los codones generalizados

El conjunto puntuación puede escribirse como sigue:



La terna de inicio como:



y posiblemente



Aunque esto es aún una cuestión abierta.

El álgebra usada realmente proporciona expresiones para representar el código genético de una única manera. Las expresiones obtenidas se denominan codones generalizados. El código genético expresado en codones generalizados no es degenerado en el sentido de que cada aminoácido procede de un único codón generalizado.

c) Autómata de estados finitos

El proceso de traslación tiene lugar en los ribosomas, partículas que enlazan el m-RNA, esto es el ácido ribonucleico mensajero, y comenzando sobre unas ciertas partes de la cadena del m-RNA, leyéndolas, produciendo una cadena polipeptídica de acuerdo con el código genético, hasta que se lee un elemento del conjunto puntuación, en cuyo caso la

traslación se completa y la cadena polipeptídica se libera. Este proceso puede representarse tal y como puede verse en la figura 3, mediante un autómata finito reconocedor del alfabeto del ARN, que traduce en aminoácidos los distintos codones.

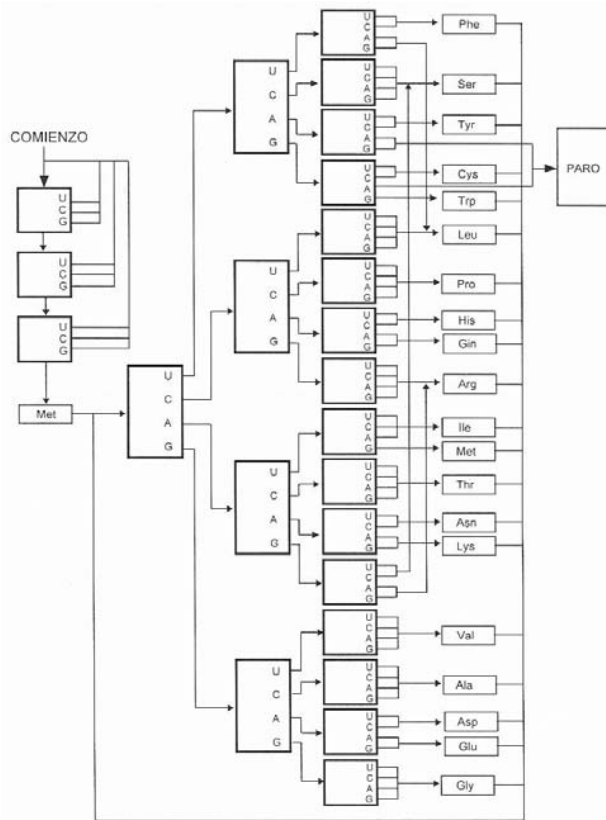


Figura 3: Una máquina de estado-finita traductora del código genético a aminoácidos

Importancia de las bases en el codón

Los codones que especifican el mismo aminoácido se denominan sinónimos, por ejemplo CAU y CAC son sinónimos para la histidina. Es de observar que los sinónimos no están distribuidos arbitrariamente en la tabla 1. Un aminoácido especificado por dos o más sinónimos ocupa una sola casilla, a no ser que sean más de cuatro los sinónimos. Los aminoácidos de cada casilla son especificados por codones que tienen las dos primeras bases iguales pero difieren en la tercera, como por ejemplo: GUU, GUC, GUA y GUG. Así pues, la mayoría de los sinónimos difieren únicamente en la última base de la terna. La revisión del código muestra que XYC y XYU codifican siempre el mismo aminoácido mientras que XYG y XYA lo hacen casi siempre. Es evidente, entonces, que las dos primeras letras de cada codón son las determinantes de relevancia, mientras que la tercera parece tener menor importancia y no encajar con tanta precisión como las otras. La base estructural para estas equivalencias es evidente debido a la naturaleza de los anticodones de las moléculas del tARN. Sin embargo, y es lo que en este trabajo se quiere enfatizar, la última base del codón es fundamental, desde un punto de vista funcional, pues es la que va a caracterizar el comportamiento de la proteína y del gen origen en el que se encuentre.

El significado biológico de la degeneración generalizada del código genético es doble. Por una parte, la degeneración reduce al mínimo los efectos deletéreos de las mutaciones. Si el código no fuera degenerado, veinte codones designarían los aminoácidos y los otros cuarenta y cuatro conducirían a la terminación de la cadena. La probabilidad de mutación que

originase la terminación de la cadena sería, por tanto, mucho mayor con un código no degenerado que con el real degenerado. Es importante consignar que las mutaciones de terminación de la cadena generalmente conducen a proteínas inactivas, mientras que las sustituciones de un aminoácido por otro son generalmente inocuas. Por otra parte, la degeneración del código también puede ser significativa en cuanto permite al ADN modificar ampliamente su composición de bases sin alterar la secuencia de aminoácidos codificados por él. El contenido de [G] + [C] podrían codificar las mismas proteínas si utilizaran sistemáticamente diferentes sinónimos.

Una respuesta plausible es la de que la redundancia proporciona un mecanismo de seguridad. En efecto, podría servir para absorber tasas de mutación excesivamente elevadas: un porcentaje de las mutaciones o sustituciones de nucleótidos no conllevaría modificación alguna en la secuencia de aminoácidos, por lo que estas mutaciones llevan el adjetivo de silenciosas. De nuevo parece que la mutación pierde interés como mecanismo de innovación.

Pero también es posible plantear un motivo de utilidad: parece bastante plausible que las mutaciones silenciosas pueden tener beneficios a largo plazo. Una mutación que conllevara una modificación inmediata en la proteína correspondiente, probablemente tendría un efecto poco beneficioso y, por lo tanto, reportaría pocas ventajas a un sistema que tolerase la existencia de éstas. Sin embargo, una mutación silenciosa no se traduce en un efecto visible, pero tiene dos consecuencias: Producir diversidad de secuencias nucleotídicas. Cada una de estas variantes puede estar más

próxima a una nueva variante más beneficiosa que la original. El resultado es que las mutaciones silenciosas probablemente permiten sustituir un aminoácido por otro con mayor flexibilidad. Lo que se quiere decir, es que en un código no redundante la alteración de un nucleótido permite sustituir un aminoácido por otro de entre un conjunto limitado de ellos. La existencia de redundancia y, en este caso, de las mutaciones silenciosas, puede provocar que el conjunto de aminoácidos alcanzables desde uno dado sea mayor. Quizás habría que hablar de estas mutaciones, en vista de lo descrito, no como silenciosas, sino como sigilosas o astutas. Se ha verificado que el porcentaje de mutaciones silenciosas que han quedado grabadas en el ADN de especies estudiadas es mucho mayor al que le correspondería estadísticamente, lo cual parece demostrar su importancia.

Se puede hacer también una analogía con lo que sucede con la transmisión de información genética en cualquier especie: un individuo puede ser portador de un gen que codifica una determinada característica sin que ésta se exprese. Es decir, que el individuo ni se va a beneficiar ni va a ser víctima de las propiedades implícitas en el gen; no obstante, puede transmitir ese gen a su descendencia entre los cuales surgirán individuos que sí expresen el gen. Las mutaciones silenciosas también producen variantes de información que permanecen latentes hasta que otra mutación genere una nueva proteína. Esta consideración conduce a una propiedad importante del ADN. Con esta molécula, la mutación pierde su importancia como generador de innovaciones. Ahora el principal mecanismo innovador es la mezcla de variantes de genes dentro de una misma especie.

Señales de Terminación

Finalmente, la última característica del código se refiere al hecho de que de los 64 ternas posibles, hay 3 que no codifican ningún aminoácido y estos son: UAG, UAA y UGA. Estas ternas constituyen las señales de terminación de las cadenas polipeptídicas. Estos codones no son leídos por moléculas de t-ARN, pero sí por proteínas específicas llamadas factores de liberación. La señal de iniciación de la síntesis proteica es mucho más compleja, pero en cualquier caso los codones AUG o GUC forman parte de ese inicio.

El enfoque de Deutsch

Cuando Aristóteles (1996) consideraba como la facultad más espectacular de la materia animada su capacidad de moverse voluntariamente, su contenido literal no es actualmente válido; sin embargo, el Estagirita tenía muy claro algo importante: Al tratar de asociar la vida con un concepto físico básico reconocía que ésta es un fenómeno fundamental de la naturaleza. Se dice que un fenómeno es "fundamental" cuando de su comprensión depende que se comprenda de una manera más profunda el mundo (Deutsch, 1999). La comprensión de la que aquí se habla, naturalmente hay otras como el amor, la religión, etc., se expresa en leyes naturales y principios de lógica y matemáticas. En este sentido, una comprensión más profunda es aquella que resulta más general, implica menos conexiones entre verdades superficialmente diversas, explica más y deja menos asunciones sin explicar. Los fenómenos más fundamentales

intervienen en la explicación de muchísimos fenómenos, pero son explicados por leyes y principios básicos, por ejemplo la gravitación.

Las características de las cosas vivas, meramente descriptivas que hacían referencia a conceptos fundamentales eran los siguientes: por supuesto el movimiento aristotélico, respiración, excreción, reproducción, crecimiento e irritabilidad en la expresión del Dr. Johnson para refutar el idealismo del Obispo Berkeley (Boswell, 1971) en el sentido de que si uno "golpea" algo vivo "devuelve" el golpe. Sin embargo, ambas explicaciones son falsas. La razón de que estas concepciones sean falsas es que ambas fracasaban en percatarse de qué es, en realidad una entidad viva, que trataban de definirla mediante algún atributo físico o alguna substancia, como el "élan" vital de Bergson. Hoy en día se cambia de conceptualización puesto que se sabe que la materia en forma de organismos vivos, no constituye la base de la vida. Es meramente uno de sus efectos, cuya base es molecular. Es un hecho que existen determinadas moléculas que hacen que determinados entornos se copien. Y esto lleva al concepto de "replicantes". En general se entiende por replicante una entidad que hace que ciertos entornos la copien, verbigracia los virus informáticos, los buenos chistes, los "meme" término acuñado por Richard Dawkins (1982) para designar ideas humanas que, como los chistes, provoca que los que los han escuchado lo mencionen para repetirlo en otras audiencias. Las teorías científicas son los "meme" más evolucionados técnica e intelectualmente.

Pues bien, toda la vida sobre la tierra se basa en "replicantes" que son moléculas y reciben el nombre de genes. Éstos, como ya se ha visto, son

verdaderos programas informáticos escritos como secuencias de símbolos en un alfabeto de cuatro letras, en un lenguaje estándar denominado "código genético", el cual, con muy ligeras variaciones, es común a toda la vida sobre la tierra. Algunos virus se basan en el ARN, mientras que los priones son en cierto sentido, proteínas moleculares autorreplicantes. Determinadas estructuras operan en el interior de las células como computadoras que ejecutan esos programas genéticos. Dicha ejecución consiste en la obtención de ciertas moléculas, proteínas, a partir de moléculas más simples, aminoácidos, en determinadas condiciones externas y éstos a partir de los codones. Por ejemplo, la secuencia ATG, es una instrucción para incorporar el aminoácido metionina a la molécula de proteína que se está obteniendo.

De un modo específico y característico, un gen, que en definitiva es un replicante molecular, es activado químicamente en determinadas células del cuerpo, a las que instruye para fabricar la correspondiente proteína. Por ejemplo, la hormona insulina, responsable del control del nivel de azúcar en sangre en los vertebrados, es una de esas proteínas. Si bien el gen para su fabricación está presente en casi todas las células del cuerpo, sólo se activa en determinadas células del páncreas, y ello únicamente cuando es necesario. A escala molecular, es eso todo lo que puede realizar un gen: programar a su computadora celular para fabricar un determinado compuesto químico. Sin embargo, los genes pueden actuar como replicantes porque esos programas químicos de bajo nivel se "suman" mediante capas y más capas de complejos controles y retroalimentaciones hasta convertirse en complicadísimas instrucciones de alto nivel. Así conjuntamente el gen de la insulina y los genes involucrados en su activación y desactivación

constituyen un completo programa para la regulación del nivel de azúcar en el torrente sanguíneo.

Del mismo modo, existen genes que contienen instrucciones específicas que determinan cómo y cuando deben ser copiados, ellos y otros genes, así como instrucciones para la fabricación de nuevos organismos de la misma especie, incluyendo las computadoras moleculares que ejecutarán todas esas instrucciones generación tras generación. Hay también instrucciones acerca de cómo debe responder el organismo en su totalidad ante los estímulos, externos, como verbigracia, cuando ha de cazar, huir, copular, luchar, comer, etc. Y así sucesivamente.

Ahora bien, un gen puede actuar como replicante tan solo en determinados entornos que, por analogía, con los nichos ecológicos, se les va a denominar nicho. Es decir, nicho es el término que se va a emplear aquí para referirse al conjunto de todos los posibles entornos en los que un determinado replicante podría hacer que lo copiasen. Y, para un organismo, conjunto de entornos y estilos de vida de los que podría vivir y reproducirse. Así, el nicho de un gen de la insulina incluye entornos en los que está ubicado en el núcleo de una célula en compañía de otros genes, y la célula, por su parte, se encuentra adecuadamente situada en el interior de un organismo funcional, en un hábitat adecuado para su vida y su reproducción. Pero hay también otros entornos capaces de copiar el gen de la insulina, como los laboratorios de biotecnología, en los que bacterias son alteradas genéticamente para incorporarlos al gen. Dichos entornos también forman parte del nicho del gen, al igual que otros posibles entornos,

infinitamente numerosos, todos ellos muy distintos de aquel en que evolucionó.

No obstante, no todo lo que puede ser copiado es un replicante. Éste "causa" que su entorno lo copie; es decir, contribuye causalmente a su propia copia. Dawkins a esto lo denomina replicante activo. Esto es, la presencia del replicante y su forma física específica "marcan la diferencia" entre si va a ser copiado o no. En otros términos, el replicante es copiado si está presente pero si fuera reemplazado por otro ente, incluso muy similar, no lo sería.

Junto con los genes, en el ADN de la mayoría de los organismos, están presentes secuencias aleatorias de A, C, G y T, denominadas secuencias "basura" o "chatarra" del ADN, que también son copiadas y transmitidas a la descendencia. Sin embargo, si semejante secuencia es sustituida por casi cualquier otra de parecida longitud sigue siendo copiada, por lo que se puede deducir que la copia de dichas secuencias no depende de su forma física específica. A diferencia de los genes, las secuencias basura no son programas. Por ello, si tuvieran alguna función, hasta hoy desconocida, no sería las de transmitir información de ninguna clase. Es decir, si bien son copiadas, no contribuye causalmente a su copia, por lo que no pueden ser consideradas replicantes. Naturalmente en realidad esto es una exageración pues, en puridad, todo lo que ha sido copiado debe haber hecho alguna contribución causal a su copia. Las secuencias basura, por ejemplo, están hechas de ADN lo que permite a la computadora celular copiarlas. Ésta no puede copiar moléculas que no sean ADN. Sin embargo, no suele resultar ilustrativo e interesante considerar replicante a algo que

tenga tan solo una pequeña contribución causal a su propia replica. Estrictamente hablando, ser replicante es cuestión de grado. En este sentido, se definirá "el grado de adaptación" de un replicante a un determinado entorno como el grado en que el replicante contribuye causalmente a su propia réplica en dicho entorno. Cuando un replicante está bien adaptado a la mayoría de los entornos de un nicho, se puede definir como bien adaptado al nicho. Así, por ejemplo, el gen de la insulina está altamente adaptado a su nicho, en tanto que las secuencias basura, tiene un grado despreciable de adaptación.

Para cuantificar grados de adaptación hay que tener en cuenta no sólo el replicante en cuestión sino también el abanico de sus posibles variantes. Cuanto más sensible a la estructura física precisa del replicante sea el proceso de copia en un determinado entorno, más adaptado estará dicho replicante a ese entorno. Para los replicantes altamente adaptados, se deben considerar sólo ligeras variaciones, puesto que si se dieran variaciones más amplias ya no serían replicantes. Esto es, para cuantificar el grado de adaptación de un replicante a un nicho, hay que tener en cuenta su grado de adaptación a todos los entornos del nicho. Es decir, hay que considerar tanto las variantes del entorno como del replicante. Si la mayoría de las variantes del replicante no consiguen causar que la mayoría de los entornos la copien, se desprende que la forma del replicante en cuestión es la causa determinante de su propia copia en aquel nicho, que si, por el contrario, la mayoría de las variantes del replicante son copiadas en la mayoría de los entornos del nicho, ello significa que la forma de ese replicante no incide en la copia, que se producirá en cualquier caso. En

consecuencia el replicante tiene escasa contribución causal a su propia copia y, por lo tanto, no está altamente adaptado a este nicho.

Así pues, el grado de adaptación de un replicante depende no sólo de lo que haga en su entorno concreto, sino también de lo que harán múltiples entes, la mayoría de los cuales no existen, en un amplio número de entornos distintos del concreto. Esta curiosa propiedad también se da en la realidad virtual, en adelante RV.

El factor más importante que determina el nicho de un gen es, normalmente, que su replicación depende de la presencia de otros genes, no sólo de sí mismo, sino externos al organismo que los contiene. Los que conviven unidos en largas cadenas de ADN internos en sí mismo en lo que se denomina organismo vivo. A menudo se dice que estos organismos se reproducen, esto es, se piensa que los organismos son replicantes, pero esto es un error. Los organismos "no son copiados" durante la reproducción, ni son los causantes de su propia copia. Son formados de nueva planta de acuerdo con planos incorporados al ADN de los organismos progenitores. Por lo tanto, el organismo en el entorno inmediato que copia a los auténticos replicantes: los genes de ese organismo. Desde esta perspectiva, el hocico de un oso y su madriguera son meras partes del entorno que manipulan los genes del oso durante el proceso de hacerse replicar.

Ahora bien, "los genes dan forma material al conocimiento sobre sus nichos". Y esto es lo trascendente pues todo lo que tiene importancia fundamental en el fenómeno de la vida depende de esa propiedad, y no de la replicación "per se". Como lo explica Deusch con un ejemplo con

diamantes, como decía Marilyn "Diamonds are forever"; si unos hipotéticos genes no replicantes compartieran todas las propiedades de los genes reales excepto la replicación, el denominador común de ambos tipos de genes replicantes o no, es la "supervivencia del conocimiento". Estrictamente hablando, pues, lo que está adaptado, o no, a un cierto nicho es, en realidad, un fragmento de conocimiento, más que un objeto físico o un ente material. Si está adaptado, tendrá la propiedad de que, una vez establecido en ese nicho, tenderá a permanecer en él. En el caso de un replicador, el material físico cambia, y con cada replicación se forma una nueva copia compuesta de partes no replicantes. El conocimiento no replicante puede también materializarse sucesivamente en "distintas" formas físicas como, por ejemplo, cuando se transfiere sonido de un disco analógico, primero a una cinta de audio y luego a un disco compacto. En suma, si bien toda la vida se basa en replicantes, de lo que realmente trata el fenómeno de la vida es del conocimiento. Es posible, pues, dar una definición de adaptación directamente en términos de conocimiento como sigue: "una entidad está adaptada a su nicho cuando da forma material al conocimiento que causa que ese nicho mantenga la existencia de dicho conocimiento". Curiosamente, el ADN; es un ejemplo, extraordinario de las dos facetas del conocimiento como objeto y como acción; pues no sólo transmite la información, sino que provoca cambios que generan la descendencia. Es decir, la vida trata de la materialización física del conocimiento, exactamente igual que el principio de Turing (1950) en cuanto dice que es posible dar forma material a las leyes de la física en cuanto son de aplicación a cualquier entorno físicamente posible, en un generador de RV. Los genes son programas así. Aun más, cualesquiera otros programas de

RV que existan o puedan llegar a existir físicamente, son efectos directos o indirectos de la vida humana. La vida es, pues, el medio presumiblemente necesario, mediante el cual los efectos a los que se refiere el principio de Turing son aplicados a la naturaleza.

¿Existe en el caso de la vida una diferencia básica entre los objetos portadores de conocimiento y los no portadores, una diferencia que no depende ni de los entornos del objeto ni de sus efectos en el futuro remoto, sino únicamente en sus atributos físicos inmediatos? La respuesta es sí. Y para ver en que consiste, hay que adoptar la postura del multiverso. En efecto, considérese el ADN de un organismo vivo, y supóngase que en alguna parte de uno de sus genes se encuentra la secuencia TCGTCGTTTC. Esta cadena concreta de diez moléculas, tanto en el nicho especial formado por el resto del gen como en su "propio" nicho, es un replicante. Es decir, incorpora una pequeña, pero importante, cantidad de conocimiento. Supóngase ahora, para completar la hipótesis, que se encuentra un segmento de ADN basura que también tiene la secuencia TCGTCGTTTC. Esta secuencia, sin embargo, no merece ser considerada replicante, puesto que apenas contribuye a su replicación y no es portadora de conocimiento alguno. Se trata de una secuencia aleatoria. Se tienen, pues, dos objetos físicos, segmentos ambos de la misma cadena de ADN, uno de los cuales es portador de conocimiento, mientras que el otro no. Sin embargo, son "físicamente idénticos". ¿Cómo puede ser esto así?. La respuesta es, como se muestra en la figura 4, que en realidad esos dos segmentos no son idénticos, sólo lo parecen cuando son vistos desde algunos universos como el que se palpa.

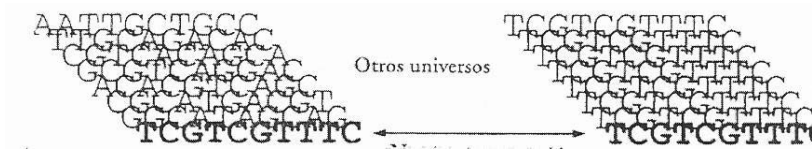


Figura 4: Vista desde el multiuniverso de dos segmentos de ADN que resultan ser idénticos en el universo “palpable”, uno aleatorio y el otro del interior de un gen

Supóngase que un rayo cósmico produce una mutación aleatoria que produce variaciones no sólo dentro de la población de organismos en un universo, sino también entre universos. Así, pues, cuando un rayo cósmico impacta en una cadena de ADN y produce una mutación, algunas de sus contrapartidas en otros universos se quedarán sin sus correspondientes copias de esa cadena de ADN, mientras que otras impactarán en distintas posiciones y originarán mutaciones diferentes. De este modo, un solo rayo cósmico que incidiera sobre una sola molécula de ADN causaría, en general, la aparición de gran número de mutaciones distintas en diferentes universos.

Cuando se considera la posible apariencia física de un determinado ente físico en otros universos, no se debe mirar tan lejos en el multiverso que resulte imposible identificar una contrapartida suya en el otro universo, de modo que se deba considerar únicamente universos lo bastante similares al conocido para que no surjan insalvables disparidades. Entonces el segmento del gen considerado debe tener la misma secuencia en casi todos los universos próximos al conocido. Ello se debe a que cabe presumir que

está altamente adaptado, lo que significa que la mayor parte de sus variantes no conseguirán hacerse copiar en la mayor parte de las variantes de sus entornos y no podrían, por tanto, aparecer en dicha localización en el ADN de un organismo vivo. En cambio, cuando el segmento del ADN no portador de conocimiento experimenta prácticamente cualquier mutación, la versión mutante seguirá siendo capaz de ser copiada. A lo largo de generaciones de realizaciones habrán ocurrido múltiples mutaciones, la mayoría de las cuales no habrán afectado a la replicación. Por consiguiente, el segmento de ADN basura, a diferencia de su contrapartida en el gen, será completamente heterogéneo en distintos universos. Podría muy bien suceder que cada posible variación de su secuencia estuviera representada por un igual en el universo, que es lo que debería entenderse cuando se dice que su secuencia es estrictamente aleatoria.

En consecuencia, la perspectiva del multiverso (Everett, 1957) revela una estructura física adicional el ADN de un organismo vivo. En el universo conocido, tal y como se muestra en la figura 4, contiene dos segmentos con la secuencia TCGTCGTTTC, uno de ellos es parte de un gen, mientras que el otro no. En casi todos los universos cercanos, el primer segmento conserva la secuencia que presenta en el universo conocido, pero el segundo varía enormemente de secuencia de un universo próximo a otro. Así pues, y desde la perspectiva del multiverso, los dos segmentos no son ni remotamente parecidos.

Hay que reconocer que la antigua idea de que la materia viva tiene propiedades especiales se aproximaba mucho a la verdad pues el hecho es

que no es la materia viva, sino la materia "portadora de conocimiento" la que es físicamente especial.

En suma, con un microscopio mágico capaz de ver en otros universos la tarea de distinguir las secuencias de gen de las de no-gen, así como de estimar el grado de adaptación de cada gen, que en cualquier universo considerado aisladamente, resultaría imposible pues se trata de una propiedad emergente, sería sencilla. Como se ve en la figura 4, los genes se distinguirán de los no genes del mismo modo que los campos cultivados se distinguen de los bosques en una fotografía aérea. Serían regulares en múltiples universos próximos, mientras que los no genes, serían irregulares. Y en cuanto a la adaptación, los genes mejor adaptados presentarían la misma estructura en un amplio abanico de universos.

Prueba de χ^2

Supóngase que en una determinada muestra se observan una serie de posibles sucesos S_1, S_2, \dots, S_j que ocurre con frecuencias o_1, o_2, \dots, o_j , denominadas "frecuencias observadas" y que, según las reglas de la probabilidad, se espera que ocurran con frecuencias e_1, e_2, \dots, e_j , llamadas frecuencias teóricas o esperadas, tal y como aparece en la tabla dada por la tabla 4.

Suceso	S ₁	S ₂	...	S _j
Frecuencia observada	o ₁	o ₂	...	o _j
Frecuencia esperada	e ₁	e ₂	...	e _j

Tabla 4: Ejemplo de muestra con frecuencias asociadas

Una medida de la discrepancia existente entre las frecuencias observadas y esperadas, viene dada por el estadístico χ^2 dado por:

$$\chi^2 = \frac{(o_1 - e_1)^2}{e_1} + \frac{(o_2 - e_2)^2}{e_2} + \dots + \frac{(o_j - e_j)^2}{e_j} = \sum_{i=1}^j \frac{(o_i - e_i)^2}{e_i}$$

donde si el total de frecuencia es N, $\sum o_i = \sum e_j = N$.

Si $\chi^2 = 0$, las frecuencias observadas y esperadas concuerdan exactamente; mientras que si $\chi^2 > 0$, no coinciden exactamente. A valores mayores de χ^2 , mayores son las discrepancias ente ambas frecuencias.

La distribución muestral de χ^2 se aproxima muy estrechamente a la distribución chi-cuadrado dada por:

$$Y = Y_0 (\chi^2)^{1/2} (v-2) e^{-1/2} \chi^2 = Y_0 \chi^{v-2} e^{1/2} \chi^2$$

Siendo v el número de grados de libertad dado por:

$v = k-1$, si las frecuencias esperadas pueden calcularse sin tener que estimar parámetros poblacionales con los estadísticos muestrales.

$v = k-1-m$, si las frecuencias esperadas sólo pueden calcularse estimando m parámetros de la población a partir de estadísticos muestrales.

La información genética contenida en el ADN y, en consecuencia, en el m-ARN está en forma de "codones" o ternas de tres nucleótidos, que se transcriben en aminoácidos, los cuales se combinan para formar proteínas. En ciertos lugares, en la estructura de las proteínas la composición del aminoácido no es crítica. Sin embargo, ciertos aminoácidos aparecen en tales sitios hasta seis veces más frecuentemente que otros. En los años 60, los biólogos moleculares buscan determinar si la composición de aminoácidos era un reflejo del código genético o si ciertos aminoácidos fueron seleccionados como óptimos.

La composición de los aminoácidos en lugares no críticos en la estructura de las proteínas pueden, o no, contribuir a la calidad total de una proteína. Si ciertos aminoácidos son óptimos para la estructura de una

proteína, la selección natural debería haber aclarado sobre el tiempo evolutivo para incrementar la frecuencia de esos aminoácidos. Una hipótesis alternativa es que la composición de aminoácidos, en un lugar es simplemente una permutación aleatoria del código genético.

Con estos antecedentes la cuestión que se plantea es la siguiente: ¿Son las frecuencias de aminoácidos particulares simplemente una consecuencia de las permutaciones aleatorias del código genético o, por el contrario, un producto de la selección natural?

Para resolver esta cuestión; es decir, para saber si algún aminoácido en concreto es de alguna forma adaptativo, entonces debería ocurrir más frecuentemente que lo esperado por el azar. Esto puede evaluarse fácilmente calculando las frecuencias esperadas de aminoácidos y comparándolas con las observadas. Los codones y las frecuencias esperadas y observadas de todos y cada uno de los aminoácidos vienen dadas en la tabla 5.

Las frecuencias de bases de ARN en la naturaleza son: 22% Uracilo, 30,3% Adenina, 21,7% Citosina y 26,1% la Guanina. La frecuencia esperada de un codón particular puede entonces calcularse multiplicando las frecuencias de cada base de ADN comprendida en el codón. Y la frecuencia esperada del aminoácido puede entonces calcularse por añadir las frecuencias de cada codón que codifica para un aminoácido. Por ejemplo, los codones de ARN para la Tirosina son: UAU y UAC, de este modo, la expectativa aleatoria para sus frecuencias es:

$$(0,22) (0,303) (0,22) + (0,22) (0,303) (0,217) = 0,0292$$

Ya que tres de los sesenta y cuatro codones son de parada o sin sentido, esta frecuencia para cada aminoácido debe multiplicarse por un factor de corrección de 1,057.

Excluyendo la Arginina, las correlaciones entre frecuencias observadas y esperadas fue de $r = 0,89$, que resulta ser altamente significativo. Sólo la Arginina resulta ser un producto de la selección natural actuando sobre uno más de los codones de Arginina.

ÁCIDOS AMINADOS	Codones	Número de apariciones	Frecuencia observada (%)	Número esperado	Frecuencia esperada (%)	$\frac{(O_i - E_i)^2}{E_i}$
Serina	UCU,UCA,UCC UCG,AGU,AGC	443	8,0	472	8,6	1,782
Leucina	CUU,CUA,CUC CUG,UUA,UUG	417	7,6	434	7,9	0,666
Arginina	CGU,CGA,CGC CGG,AGA,AGG	230	4,1	582	10,6	212,9
Glicina o glicocola	GGU,GGA,GGC GCG	406	7,4	390	7,1	0,656
Alanina	GCU,GCA,GCC GCG	406	7,4	395	7,2	0,306
Valina	GUU,GUA,GUC CUG	373	6,8	330	6,0	5,603
Treonina	ACU,ACA,ACC ACG	340	6,2	373	6,8	2,919
Prolina	CCU,CCA,CCC CCG	275	5,0	275	5,0	0
Isoleucina	AUU,AUA,AUC	209	3,8	280	5,1	18,0
Lisina	AAA,AAG	395	7,2	302	5,5	28,639
Acido glutámico	GAA,GAG	318	5,8	258	4,7	13,953
Acido aspártico	GAU,GAC	324	5,9	192	3,5	90,750
Fenilalanina	UUU,UUC	220	4,0	121	2,2	81
Asparragina	AAU,AAC	242	4,4	225	4,1	1,284
Glutamina	CAA,CAG	203	3,7	210	3,8	0,233
Tirosina	UAU,UAC	181	3,3	165	3,0	1,551
Cisteina	UGU,UGC	181	3,3	137	2,5	14,131
Histidina	CAU,CAC	159	2,9	164	3,0	0,152
Metionina	AUG	99	1,8	99	1,8	0
Triptófano	UGG	71	1,3	88	1,6	3,284
TOTALES		5.492	100,0	5.492	100,0	$\chi^2 = 477,809$

Tabla 5: Aminoácidos y tripletes asociados con sus frecuencias según King y Jukes (1969)

En conclusión, la composición promedio de los aminoácidos refleja pasivamente las permutaciones aleatorias del código genético. En otras palabras, en lugar de ser óptimos aminoácidos particulares, la cosa más importante que determina la frecuencia de los aminoácidos en una proteína es sencillamente el número de codones codificados por él.

En la práctica, las frecuencias esperadas se calculan de acuerdo con una hipótesis H_0 . Si bajo esta hipótesis el valor calculado χ^2 es mayor que algún valor crítico tal como $\chi_{0,95}^2$ o $\chi_{0,99}^2$, que son los valores críticos a los niveles de significación de 0,05 y 0,01 respectivamente, se deduce que las frecuencias observadas difieren significativamente de las esperadas y se rechaza H_0 al nivel de significación correspondiente. En caso contrario, se aceptará o al menos no se rechazará H_0 .

Pues bien, considerando la tabla 5 de King y Jukes (King, 1969), los autores realizaron la prueba de χ^2 a esos datos, observándose que el valor de χ^2 es: 477,809. Como este valor es mucho mayor que el $\chi_{0,995}^2$ esperado que para 19 grados de libertad, que vale 38,6, se deduce que las frecuencias observadas difieren muy significativamente de las esperadas rechazándose la hipótesis H_0 . En consecuencia, cabe esperar que las asociaciones de bases en ternas no se hace al azar, y por ello es necesario buscarle una explicación.

Descifrado del genoma

El problema

Como se ha dicho, de ordinario los genes están en los cromosomas: estructuras celulares éstas cuyo principal componente es el ácido desoxirribonucleico, abreviado ADN, que está formado por dos cadenas complementarias, integradas por largas secuencias de unidades nucleótidos. Cada nucleótido contiene una de las cuatro bases nitrogenadas posibles: Adenina (A), Citosina (C), Timina (T) y Guanina (G) que sólo se asocian, de acuerdo con la Ley de Chargaff, de dos modos posibles entre ellas: A con T y C con G. Lo común es que algunas porciones de ADN formen genes; otras, no. En el caso humano, las porciones que son genes constituyen tan sólo aproximadamente el 10% del ADN total. El resto parece que no tiene nada que ver con la síntesis de las proteínas, es, por así decirlo, hasta que se le encuentre una funcionalidad, basura genética. La lectura y significado de este conjunto de símbolos que forman el ADN puede compararse, "prima facie", con el desciframiento del jeroglífico de la vida. Si el desciframiento de la escritura jeroglífica, por Champolion, a partir de la piedra Roseta, fue ardua y difícil, que no lo será el de 3×10^9 símbolos a partir de un alfabeto de sólo cuatro símbolos. Para darse una idea de la magnitud de la empresa baste decir que, por ejemplo, la secuencia de nucleótidos del ADN escritos ocuparían un espacio equivalente a 150 tomos similares a la guía telefónica de una ciudad como Madrid. O, por establecer un ejemplo comparativo más gráfico, si un genoma vírico que tiene 3.000 pares de bases, ocupa una página de un libro que constara de 3.000 letras, y

otro bacteriano contiene tres millones de pares de bases, equivaldría a un libro de mil páginas, el genoma humano, formado por tres mil millones de bases, ocuparía una biblioteca de mil libros. Las características desde el punto de vista criptoanalítico de código genético son las siguientes:

Sólo se dispone del texto cifrado, que es el caso más difícil.

En general, el proceso de ataque a un texto cifrado se basa en el hecho de que se puede determinar si el resultado del proceso de descifrado se corresponde con un lenguaje determinado, o dicho de otro modo, es texto en claro. Además, en los problemas basados en métodos de criptoanálisis, se conoce normalmente el lenguaje utilizado para la redacción del texto claro. Sin embargo, en el caso del código genético no es así. Por un lado, es muy poco probable que el código genético sea un intento de ocultar algún mensaje en claro, por otro, no se conoce ninguna correspondencia con ningún mensaje actual. Sin embargo, existen precedentes en los que se han utilizado técnicas de criptoanálisis y otras técnicas semejantes en la comprensión de lenguajes, de los que se desconocían si se podían equiparar a los lenguajes actuales, como es el caso de los jeroglíficos egipcios. Aún en esos casos, sin embargo, se emplearon “puntales” o palabras de las que se conocía su escritura en más de un lenguaje o bien se buscaron analogías con respecto a los lenguajes conocidos en la actualidad. Quizás, el código genético en ese aspecto de no tener puntales, sea mejor compararlo con el lineal B (Singh, 2000).

El código genético no es, necesariamente un lenguaje. Con menor probabilidad, cualquier transformación producida sobre él, aún mediante el

uso de técnicas de criptoanálisis, dará lugar a un texto de cualquier lenguaje humano. Por ello, el uso de estas técnicas se muestra, en principio, inadecuado para la resolución de este problema. Sin embargo, las técnicas de criptoanálisis sí pueden aportar luz sobre ciertos aspectos del código genético. Por ejemplo, el método de Kasiski permite encontrar ciertos patrones que se repiten con una periodicidad, y el análisis de índices de coincidencia aporta información relevante acerca de la variación de las frecuencias del mensaje genético. Por lo anteriormente dicho, la obtención de resultados de este tipo, no presentarían una gran relevancia con respecto a la relación entre el código genético y un lenguaje claro, pero sí una gran relación con otro punto de vista del código genético, como es su interpretación como si fuese el código de un programa de computadora.

Tal vez una forma de descifrar el jeroglífico sea la que propone Lander del MIT cuando dice que del mismo modo que la organización de los elementos químicos en la tabla periódica dio coherencia a una masa de datos previamente sin relacionar, se acabará descubriendo que las decenas de miles de genes de los organismos actuales están hechos de combinaciones de un número mucho más reducido de módulos o elementos genéticos más simples o, por así decirlo, genes primordiales. Lo importante es que los genes humanos no deben ser considerados como completamente diferentes unos de otros. Antes bien, deberían verse como unas familias que mantienen toda clase de vida. Una vez completada la tabla se dará fin a la genética estructural para iniciar la genética funcional o uso práctico de la tabla. Por ejemplo, la diferencia entre dos personas no es más del 1% de las

bases. La mayoría de los genes sólo tendrán dos, tres o cuatro variantes; es decir, unas 300.000 variantes principales.

Retomando ahora el desciframiento del genoma, se denomina secuenciar el genoma a, únicamente, enunciar las bases A, T, C y G del mismo. Es decir, es leer sin comprender. Por lo tanto, es deletrear el genoma o, a lo sumo, balbucearlo. Por el momento, secuenciar el genoma consiste, simplemente, en determinar el encadenamiento de los miles de millones de bases que lo componen. Como lo señaló Cohen (1994), es fácil darse cuenta de la aridez de la tarea, ya que se trata de leer un texto que, poco más o menos, dice así:

TCATCGTCGGCTAGCTCATTGACCATCGTATGCATCACTATTAC
TGATCTTG...,

y que discurre a lo largo de millones de líneas y miles de páginas. Desde luego con una "lengua" que posee un alfabeto de sólo cuatro letras, era de esperar una monotonía mucho mayor que la que presentan las lenguas actuales cuyos alfabetos, latino, cirílico, etc., se componen de más de veinte letras.

Por importante que sea, sobre todo por su carácter de condición necesaria, secuenciar el genoma no basta, pues hay que descifrarlo; esto es, comprender su significado. En otros términos, es como aprender a leer letras y convertirlas en ideas. Y ahí la cuestión se vuelve peliaguda. De hecho, actualmente es imposible prever cuándo, cómo y en qué medida será posible descifrar el genoma una vez secuenciado. Sería maravilloso, pero

poco realista, el suponer que el aprendizaje de la lectura del genoma desemboque inopinadamente en su comprensión, del mismo modo que los niños que aprenden las letras, franquean, de repente, cierto misterioso umbral y empiezan a comprender lo que leen. La realidad es, sin embargo, más dura y el descifrado del genoma se presenta como una tarea ímproba e ingrata. Lo ingrato de la labor no se debe tanto a la pobreza del alfabeto cuánto a la ignorancia de la lengua.

Por poner un ejemplo ilustrativo, debido a Cohen, de lo que acaba de decirse, trátase de leer, aunque no sea más que durante veinte minutos, y en voz alta para no hacer trampas, una novela escrita con el alfabeto latino de veintiséis letras, pero en una lengua desconocida: turco, serbocroata, ... y se verá el dolor de cabeza que produce esto. Supóngase ahora que no hay elección, puesto que uno se encuentra en una isla desierta y el único libro que tiene está escrito en una de esas lenguas de las que no se tiene diccionario a mano. Por lo tanto, uno debería arreglárselas con lo poco que sepa y ejercitar la paciencia y la perseverancia. Si así se hace, uno acabará por familiarizarse con ciertas características de la lengua desconocida. Por ejemplo, localizará motivos recurrentes, establecerá analogías, descubrirá ciertas reglas, otros significados, similitudes interesantes, etcétera, etcétera. De hecho, en primer lugar, se observa que esta lengua tiene una peculiaridad que, al menos en principio, resulta desconcertante, no se conforma con presentar unas secuencias provistas de un significado preciso, las que interesan y se denominan genes, sino que también presenta todo unos farragosos párrafos situados tanto entre gen y gen, intergenes, como en el interior de los genes, intragenes, que dividen la secuencia que tiene

significado. Hasta hoy, nadie ha podido averiguar para qué sirve todo este relleno. Y, para colmo de la exasperación, estas extravagantes series de letras constituyen más del noventa por ciento del genoma. Por lo menos, del genoma humano, que se presenta como una interminable lista de genes e intergenes con intragenes no codificadores situados en el seno mismo de los genes.

Considérese ahora un tomo de las poesías de Quevedo (1995), pero escritas en lengua desconocida y con el siguiente estilo genómico, aunque para comodidad del lector se ha traducido el texto al español:

kvlseimkmifdsqmoieurniñaedpvaeighmlkehimdogcoleizoapglkchdjth
zkeivauozierdmofmoimthaaeokkkkkkghdmorsleidjtlidfifsfifesithgmelmilk
ajchwchmqurozqdaverirlmeoarusndorkejmtsimeormtlehoekdmzriglalmetho
slerthrosazlckthmekromlsdigquemelthnlsrejtstalkhlrjjsletehrejrozthiolalely
rletuolgdnartlgldrtlmaljkfsdñanaioemzlekthldiimsejrmkthomsdlkgldheoz
kelldesgyruriotpleghkdssseruieormdkfjhjkdddddgoghfdjsenbvxxwqsd ...

Actualmente, los biólogos empiezan a distinguir las secuencias características que comienzan y terminan las palabras del genoma y pueden identificar cada vez mejor los genes de los que no son genes. Por lo tanto, con un poco de entrenamiento, los genetistas consiguen leer, entre la maraña que precede, lo siguiente:

“Quitar codicia no añadir dinero hace a los hombres ricos Casimiro”

Pero, incluso después de identificar las palabras de la poesía, aún se estaría muy lejos de imaginar cómo era ese Casimiro a quien va dirigida la poesía.

Volviendo a la analogía con el lenguaje humano, la tarea que se han impuesto los que secuencian el genoma es el inventario de las miles de palabras del diccionario biológico de la humanidad. Sólo el inventario. La explicación de las palabras vendrá después. Para dar una idea de la empresa, este diccionario, a razón de una treintena de genes por página, y un millón de caracteres aproximadamente, tendrá tres mil páginas, dado que el genoma tiene tres mil millones de bases. El equivalente, en páginas y peso, a los dos tomos del Diccionario de la Real Academia de la Lengua.

Pero, a diferencia de los diccionarios corrientes, los vocablos no estarían dispuestos por orden alfabético, sino por orden cromosómico. Las palabras del diccionario genómico humano, en lugar de agruparse en las correspondientes secciones de la A a la Z, se ordenarían en 23 capítulos, uno por cada par de cromosomas debidamente numerados del 1 al 23, por el orden convencional que los biólogos le atribuyen. Los del primer par son los más largos y los del vigésimo primero y vigésimo segundo, los más cortos. Son, como ya se ha dicho, los cromosomas denominados "autosomas", es decir, no sexuales, del griego soma, cuerpo. El vigésimo tercer par corresponde a los cromosomas sexuales X e Y.

Así pues, el primer capítulo, un cromosoma de x millones de bases, ocuparía x páginas a razón de un millón de caracteres por página; el segundo, y..., el vigésimo primero, z. "Balizar" y cartografiar el genoma

equivale a paginar el diccionario: ordenación de las páginas 1, 2, 3, ... hasta tres mil, en 23 capítulos, pero sin llenarlos todavía. En este estadio, aún no se sabe que hay en cada una de las páginas, a excepción de un millar de palabras, repartidas aquí y allá. Una vez terminada la paginación o, si se prefiere, la cartografía, podrán los expertos dedicarse al secuenciado sistemático de los genes, esto es, a llenar de palabras, todavía sin la explicación, las páginas.

A continuación, dentro de unos años, cuando se tenga la secuencia del conjunto del genoma, se verá que en el capítulo primero, página 1, están las palabras, o, sus equivalentes genómicos, no por orden alfabético: a, abaca, abacena, abacial, ábaco, ..., abad, abadejo, ..., abadesa, ..., sino en el orden que les ha sido impuesto, por la naturaleza a lo largo de los cromosomas. Por ejemplo: gracia, desgracia, gracioso; después una palabra que no se parece a la anterior, como "cotidiano", para continuar con: agraciado, gracias, seguidas de unas "incongruentes" ciruela y reparto y, a continuación, desgraciadamente, en gracia, agraciar, Entonces, observándolas más atentamente, se encuentran que muchas de ellas tiene en su composición las letras "graci" y uno se pregunta: ¿Qué significa ese factor común? y, más allá, se observará que aparece el factor común "gen" en genético, que también aparece en otro sitio en otro capítulo, en génesis, genio, ingenioso, genial, Esto permitirá introducir todos los "graci" y todos los "gen" en una computadora para descubrir su frecuencia de aparición y examinar las secuencias que los acogen en todos los sentidos. Luego, se tratará de reclasificar el todo, después de haber anotado los datos estadísticos significativos. En otra ocasión, los investigadores se pueden

tropezar con una serie del estilo de: complacer, complacencia, complaciente, placentero, pláceme, placer, ... y pudiera darse el afortunado caso de que se conoce el significado de su factor común, en este caso "placer" porque está asociado a tal o cual particularidad hereditaria ya estudiada en profundidad. Ello, permitirá el aventurarse a ensayar nuevas reglas de construcción, nuevos acoplamientos, a partir de la misma raíz común. Todo ello de un modo similar a cómo los cartuchos ayudaron a descifrar los jeroglíficos. Es decir, los genes, al igual que las palabras del idioma, pueden agruparse por familias, los vocablos de las diferentes lenguas humanas que giran en torno al mismo concepto comportan una misma raíz. De igual manera, cabe suponer que los genes cuya secuencia se parezca poseen funciones similares. Lo cierto y verdad, al menos aquí y ahora, es que los genes que se parecen vienen de un mismo gen ancestral, y forman familias de genes con funciones similares. Se habla entonces de familias multigénicas, cuyos genes pueden estar diseminados en varios cromosomas. Con frecuencia, genes vecinos tienen una ortografía similar aún en el caso de que no empiecen por las mismas letras. Y los genes homónimos son generalmente sinónimos, genes que se escriben de forma diferente pero tienen significados parecidos. Ello permitirá usarlos, de un modo similar a cómo se usaron los puntales en el desciframiento de códigos secretos.

Poco a poco, se irá refinando y afinando el análisis. Al final se sabrá cómo distinguir los equivalentes de los sinónimos lingüísticos. Incluso, detectando las raíces comunes, se aprenderá a remontar la genealogía de ciertos genes; es decir, a seguir su linaje evolutivo. Las funciones biológicas

complejas, como: la respiración, la digestión o la reproducción, se asimilan a frases cuyas palabras están inscritas en páginas diferentes del diccionario genómico. Porque, si existe un primer libro, el diccionario genético escrito según el orden topográfico de los cromosomas, la evolución ha escrito otros mil después de aprenderse este diccionario: sobre la fisiología, el crecimiento, el envejecimiento, la inmunidad... hasta el libro del pensamiento que, sin duda, nunca se terminará. Incluso se podría imaginar que un mismo gen pudiera adquirir un significado diferente según el lugar que ocupa en tal o cual frase genética, al igual que una palabra en el lenguaje humano depende del contexto. Además, a medida que la evolución ha alcanzado nuevos niveles de complejidad, ha ido inventando funciones biológicas. Es probable que, paralelamente, la combinación genética se haya enriquecido con nuevas reglas, integradoras de los niveles precedentes, en otras tantas muñecas rusas de "algoritmos", colocadas unas dentro de otras en sutil jerarquía.

En resumen, la lengua genética ha debido perfilar progresivamente su sintaxis y su estilística y falta por encontrar su semántica y su pragmática. Los mensajes que determinan el color de los ojos, la textura de la piel o la masa muscular no son, sin duda, de la misma factura que los que inducen el sistema inmunitario, la diferenciación celular o el "cableado" cerebral. Obviamente, muchos conceptos fundamentales de esta lengua son desconocidos. Una vez hecho el secuenciado, queda aún mucho que investigar de modo que, para ver claro, tal vez sea necesario siglos de trabajo, si es que se acaba alguna vez. La cuestión es que el ADN no se ofrece como un libro abierto ni como una cinta de vídeo.

Conjetura o hipótesis de trabajo

El dogma científico clásico al que se le inculca o debería inculcar a todo universitario es que primero se formulan conjeturas o hipótesis y luego se comprueban. Pero, naturalmente, para la formulación de dichas conjeturas o hipótesis es necesario tener en cuenta los hechos, tal y como se presentan y estos hechos son:

Las proteínas deben su función a su estructura o plegado, esto es, a su forma que depende del orden en la secuencia de los aminoácidos que la componen. Y este orden, a su vez, viene determinado por la secuencia de bases del ADN.

De acuerdo con la tabla 4 de King y Jukes (King, 1969), el autor y colegas calcularon el χ^2 y observaron que el test de χ^2 que se obtiene, no da lugar a dudas de que la distribución de ternas y su traducción en aminoácidos no se debe al azar, sino todo lo contrario.

De acuerdo con el código genético se sabe que varias ternas dan el mismo aminoácido.

Esto lleva a plantear la Hipótesis de Trabajo o conjetura propuesta:

Dependiendo de que triplete y en que posición dé un aminoácido concreto, así será la "situación" del individuo.

Comprobación:

Para verificar esta conjetura hay que realizar, y en eso se está, los siguientes pasos:

- P1. Determinar una enfermedad genética de etiología única.
- P2. Intentar asociar dentro del gen causante de la enfermedad las posibilidades de las relaciones "tripletes versus aminoácidos obtenidos", para establecer, si es posible, una relación causal, y si no, una correlación. Para ello hay que definir el tamaño de la muestra adecuado.
- P3. Si se establece una causalidad, o una correlación superior a 0,8 ir a P1 con un nuevo caso. Si el número de casos es mayor que el tamaño de la muestra propuesto y se estableció para todos una causalidad o correlación alta, aceptar la hipótesis; si no, rechazarla.

Naturalmente, cuando el código genético sea descifrado en su totalidad, este tipo de hipótesis carecerá de sentido pues el propio ADN permitirá hacer explícito su mensaje y las consecuencias que de él emanan. Pero mientras tanto, es una forma de entender el por qué se producen ciertas enfermedades.

Bibliografía

- [1] Alonso, F., Pazos, J., 1992. "*La Superación de los Obstáculos para el Desarrollo de Inteligencia Artificial*". En Inteligencia Artificial:

Fundamentos Teóricos y Aplicaciones, Real Academia de Ciencias Exactas, Físicas y Naturales, pp: 219-256. Madrid.

- [2] Aristóteles, 1996. "*Investigación sobre los animales*". Circulo de lectores, S.A. Barcelona, España.
- [3] Boswell, J., 1791. "*Boswell's Life of Johnson*". (6 Agosto de 1763). Londres.
- [4] Cohen, D., 1994. "*Los Genes de la Esperanza*". Seix Barral, S.A. Barcelona.
- [5] Crick, F.H.G., Barnett, L., Brenner, S., Watts-Tobin, R.J., 1961. "*General Nature of the Genetic Code for Proteins*". Nature, vol. 192, pp. 1277-1232.
- [6] Dawkins, R., 1982. "*El Gen Egoísta*". Salvat, S.A. Barcelona. España. Edición Inglesa, The Selfish Gene. Oxford University Press. Oxford.
- [7] De Broglie, L., 1955. "*Por los Senderos de la Física*". Conferencia Pronunciada en el Museo Guimet el 25 de Febrero de 1955.
- [8] DeJong, K., 1975. "*An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*". PhD Dissertation. Dept. of Computer and Communication Sciences. University of Michigan. Ann Arbor, Michigan.

- [9] Deutsch, D., 1999. "*La Estructura de la Realidad*". Anagrama, S.A. Barcelona.
- [10] Einstein, A., 1995. "*Autobiografía y Escritos Científicos*". Círculo de Lectores, S.A. Barcelona, España. pp. 50-51.
- [11] Einstein, A., 1970 "*Autobiographical Notes*". In, Albert Einstein: Philosopher Scientist. P.H. Schilpp (Ed.). Open Court Publishing Company. La Salle, Illinois.
- [12] Everett, H., 1957. "*Relative State. Formulation of Quantum Mechanics*". Rev. of Mod. Physic, vol. 29, pp. 454-462.
- [13] Fogel, L., Atmar, J.W., 1992. Eds.: *Proceedings First Annual Conference on Evolutionary Programming*.
- [14] Fogel, L., Owens, A.J., Walsh, M.J., 1966. "*Artificial Intelligence Through Simulated Evolution*". John Wiley and Sons. New York.
- [15] Golberg, D.E., 1989. "*Genetic Algorithms in Search, Optimization and Machine Learning*". Addison-Wesley. Reading. Mass.
- [16] Holland, J.H., 1975. "*Adaptation in Natural and Artificial Systems*". University Michigan Press Ann Arbor, Michigan.
- [17] Holland, J.H., 1992. "*Algoritmos Genéticos*". Investigación y Ciencia. Septiembre.

- [18] Hu, S-T., 1965. "*Elements of Modern Algebra*". Holden-Day, Inc. San Francisco, Ca.
- [19] Kemeny, J.G., 1955. "*Man Viewed as a Machine*". Scientific American, vol. 192 (4), April, pp. 58-67.
- [20] King, J.L., Jukes, T.H., 1969. "*Non Darwinian Evolution*". Science, vol. 164, pp. 788-798.
- [21] Kipling, R., 1990. "*The Complet Verse; The Elephant's Child; Just so Verses*". Kyle Cathie Limited. London, pp. 497.
- [22] Koutnetsov, B., 1967. "*Einstein. Sa Vie. Sa Pensée. Ses Théories*". Marabout Universite Gerard & Co. Verviers, Belgique.
- [23] Koza, J.R., 1992. "*Genetic Programming*". The MIT Press. Reading, MA.
- [24] Lorenz, K., 1969. "*La Otra Cara del Espejo*". Plaza & Janés, S.A. Esplugas de Llobregat, España.
- [25] Michalewicz, Z., 1992. "*Genetic Algorithms + Data Structures = Evolucionary Programs*". Springer Verlag. New York.
- [26] Penrose, L. J., 1964. "*Máquinas que se Autorreproducen*". En R. Canap, et. Al.: *Matematicas en las Ciencias del Comportamiento*. Alianza Editorial, S.A. Madrid, pp. 270-289.

- [27] Poincaré, H., 1963. “*La Ciencia y la Hipótesis*”. Espasa Calpe, S.A. Madrid.
- [28] Poincaré, H., 1997. “*Sobre la Ciencia y su Método*”. Círculo de Lectores, S.A. Barcelona, España.
- [29] Quevedo, F. de., 1995. “*Poesía Completa P*”. Turner Libros, S.A. Madrid, pp. 43-44.
- [30] Rechenberg, I., 1973. “*Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*”. Fromman-Holzboog Verlag. Stuttgart. Deutschland.
- [31] Schrödinger, E., 1983. “*What if Life?*”. Cambridge Universities Press. Cambridge 1944. Traducción española ¿Qué es la Vida?. Tusquets Editores, S.A. pp. 40-42. Barcelona, España.
- [32] Schwefel, H.P., 1975. “*Evolutionsstrategie und Numerische Optimierung. Dissertation*”. Technische Universität Berlin. Berlin.
- [33] Sinden, R.R., 1994. “*ADN Structure and Function*”. Academic Press. New York.
- [34] Singh, S., 2000. “*Los Códigos Secretos*”. Editorial Debate, S. A. Madrid.
- [35] Shanonn, C.E., 1940. “*An Algebra for Theoretical Genetics*”. Ph. D. Dissertation. Massachusetts Institute of Technology (MIT).

- [36] Sófocles, 1996. *“Ajax Las Traquinianas. Filotecte”*. Planeta De Agostini. Madrid.
- [37] Ter-Avanesyan, M., Inge-Vechfomov, 1988. *“Genetical Control of the Protein Synthesis”*. Leningrad University Press. Leningrad.
- [38] Tipler, F.J., 1980. *“Extraterrestrial Intelligent Beings Do Not Exit”*. Quarterly Journal of the Royal Astronomical Society, vol. 21, pp. 267-281.
- [39] Turing, A.M., 1950. *“Computing Machinery and Intelligence”*. Mind, vol. 59, n° 236, pp. 433-460.
- [40] Turing, A.M., 1992. *“Intelligent Machinery”*. In, Collected Works of A.M., Turing: Mechanical Intelligence. Elsevier Science Publishers. New York.
- [41] Turing, A.M., 1936. *“On Computable Number, with an application to the Entscheidungsproblem”*. Proc. Lond. Math. Soc. (2), vol. 42, pp. 280-285.
- [42] Von Neumann, J., 1963. *“La Teoría Lógica y General de los Automatas”*. En Cerebral Mechanisms in Behaviour. L.A. Jeffres (Ed). John Willey and Sons. New York 1951, pp. 1-41. Y en, John von Neumann. “Collected Works”. Pergamon. Oxford, pp. 288-328.

- [43] Von Neumann, J., 1966. "*Theory of Self-Reproducing Automata*". Illinois University Press. Urbana. Illinois.
- [44] Vose, M., 1993. "*Modeling Simple Genetic Algorithms*". Foundation of Genetic Algorithms (2). D. Whitley (Ed.) Morgan Kauffman, pp. 63-73.
- [45] Watson, J.D., Crick, F.H.C., 1953. "*Molecular Structure of Nucleic Acid. A Structure for Deoxyribose Nucleic Acid*". Nature, vol. 171, 1953 (a), pp. 737-738.
- [46] Watson, J.D., Crick, F.H.C., 1953. "*Genetic Implications of the Structure of Deoxyribonucleic Acid*". Nature, vol. 171, 1953 (b), pp. 964-967.



El libro
Inteligencia Artificial y
Computación Avanzada
de Juan J. Romero, Carlos Dafonte, Ángel
Gómez y Fernando J. Penousal (Eds.),
editado por la
Fundación Alfredo Brañas
se terminó de imprimir en los talleres de
Tórculo, Artes Gráficas de A Coruña
en el mes de enero de 2007



PUBLICACIONES DE LA FUNDACION ALFREDO BRAÑAS

C/ Ourense nº 11 - bajo.
15701. Santiago de Compostela
www.fundacionbranas.org
Telf.: (981) 58.05.77 - Fax: (981) 58.01.63
CIF.: G-150.78.009

Distribuidor para Galicia de los Libros de la FAB
Editorial Galaxia, Travesía de Vigo, 71
36206. Vigo

Colección Informática

1. S. CARIDAD-SIMON (Ed.). *Los Medios Informáticos de Gestión Empresarial Pública y Privada*. 1991. 290 pág. I.S.B.N. 84-88051-01-8.
2. S. CARIDAD-SIMON (Ed.). *El Reto de la Informática en la Década de los 90*. 1991. 250 pág. I.S.B.N. 84-88051-02-6.
3. A. PAZOS SIERRA y N. EZQUERRA (Eds.). *Inteligencia Artificial en Medicina*. 1994. 164 pág. I.S.B.N. 84-88051-42-5.
4. A. PAZOS SIERRA y J.Mª BARREIRO (Eds.). *Teleinformática*. 1996. 160 pág. I.S.B.N. 84-88051-55-7.
5. A. PAZOS SIERRA y J.Mª BARREIRO (Eds.). *Realidad Virtual y Visualización Avanzada*. 1999. 320 pág. I.S.B.N. 84-88051-68-9.
6. A. PAZOS SIERRA, J.Mª BARREIRO, A. SANTOS DEL RIEGO y J. DORADO DE LA CALLE (Eds.). *Aplicaciones Avanzadas en Internet*. 1999. 222 pág. I.S.B.N. 84-88051-74-3.
7. A. PAZOS SIERRA, C. MONTES GRACIA, A. SANTOS DEL RIEGO y J.Mª BARREIRO (Eds.). *Nuevas Tecnologías y Computación Avanzada en la Ayuda a Discapacitados*. 1999. 348 pág. I.S.B.N. 84-88051-72-7.
8. J. ARES, M. LÓPEZ, A. MORENO y R. GARCÍA (Eds.). *Ingeniería del Software. Diferentes Tendencias*. 1999. 280 pág. I.S.B.N. 84-88051-75-1.
9. A. MORENO SÁNCHEZ-CAPUCHINO, J.Mª BARREIRO SORRIVAS, A. SANTOS DEL RIEGO y R. GARCÍA VÁZQUEZ (Eds.). *Ingeniería del Software en Aplicaciones 3D. Entornos Inmersivos e Interactivos*. 1999. 160 pág. I.S.B.N. 84-88051-92-1.
10. J. DORADO, J. RABUÑAL, J.J. ROMERO y A. PORTO (Eds.). *Protección y Seguridad de la Información*. 2002. 160 pág. I.S.B.N. 84-88051-98-0.
11. J. DORADO, N. PEDREIRA, J. PEREIRA y C. DAFONTE (Eds.). *El Futuro de Internet: Acceso y Teleservicios*. 2002. 258 pág. I.S.B.N. 84-88051-99-9.
12. J. DORADO, N. PEDREIRA, J. RABUÑAL y A. PAZOS (Eds.). *Bioinformática*. 2004. 228 pág. I.S.B.N. 84-88051-49-2.
13. J.J. ROMERO, C. DAFONTE, A. GÓMEZ y F.J. PENOUSAL (Eds.). *Inteligencia Artificial y Computación Avanzada*. 2007. 400 pág. I.S.B.N. 84-934497-0-9.





