Calibración multivariante de reacciones químicas utilizando EDAs paralelos *

A. Mendiburu, J. Miguel-Alonso
Dept. de Arquitectura y
Tecnología de Computadores
Universidad del País Vasco
{amendiburu,miguel}@si.ehu.es

J.A. Lozano
Dept. de C. de la
Computación e I.A.
Universidad del País Vasco
lozano@si.ehu.es

M. Ostra, C. Ubide
Dept. de Química Aplicada
Universidad del País Vasco
miren.ostra@ehu.es,
carlos.ubide@ehu.es

Resumen

El presente artículo describe la aplicación de diferentes técnicas de minería de datos para resolver un problema de calibración en un entorno de química cuantitativa. Datos obtenidos a partir de reacciones experimentales con concentraciones conocidas para dos o más componentes, son usados para entrenar un modelo que, posteriormente, será utilizado para predecir las concentraciones de dichos componentes (desconocidos) en una nueva reacción. El problema puede ser planteado como selección + predicción, donde el objetivo es conseguir buenos valores para las variables a predecir, a la vez que el número de variables de entrada es minimizado, seleccionando un pequeño conjunto de ellas (verdaderamente significativo). Se han completado aproximaciones iniciales utilizando Análisis de Componentes Principales y Filtrado. Posteriormente, se han aplicado otros métodos para reducir el número de variables seleccionadas, mediante el uso de Algoritmos Paralelos de Estimación de Distribuciones (EDAs), tratando de reducir el error medio de las predicciones.

*Este trabajo se ha realizado con la ayuda de: Ministerio de Educación y Ciencia (TIN2004-07440-C02-02 y PPQ2003-07318-C02-02), Diputación Foral de Gipuzkoa (OF-846/2004), Gobierno Vasco (ETORTEK-BIOLAN, SAIOTEK S-PE04UN25) y la Universidad del País Vasco (9/UPV 00140.226-15334/2003). Los autores agradecen al Dr. S. Maspoch (Universidad Autónoma de Barcelona) la cesión de los datos experimentales utilizados.

1. Introducción

En los laboratorios modernos, el desarrollo de instrumental químico, ha permitido la existencia de equipamiento capaz de adquirir grandes cantidades de información en cortos periodos de tiempo. Por ejemplo, todo el espectro Ultravioleta-visible (UV-Vis) puede ser obtenido a razón de varias muestras por segundo utilizando CCDs. Habitualmente, el número de puntos de datos en cada espectro varía entre 100 y 1000, y el número de espectros adquiridos entre 100 y 200. Toda esta cantidad de información se puede almacenar de manera eficaz en un ordenador personal, abriendo un abanico de nuevas posibilidades a la hora de estudiar dicha información. Todo tipo de técnicas de minería de datos pueden ser utilizadas para obtener conocimiento a partir de los datos crudos.

Son muchas las reacciones químicas que pueden ser controladas a través de su espectro UV-Vis. Cuando las condiciones de la reacción son controladas, se puede lograr que las tasas de cambios en el espectro UV-Vis sean exclusivamente dependientes de la concentración de analitos que forman parte de la reacción; de esta forma, es posible conocer la concentración de dichos componentes en la solución. Este tipo de procedimientos puede ser aplicado para determinar combinaciones de 2-3 componentes fuertemente relacionados. Para ello, son dos los pasos a completar: en el primero, obtener un volúmen suficiente de datos a partir de concentraciones conocidas de los analitos de

interés, creando un modelo que, en el segundo paso, es utilizado para predecir la concentración de los mismos en una mezcla desconocida.

Para construir el modelo se puede utilizar una red neuronal artificial, entrenándola con datos experimentales durante la fase de construcción del modelo o fase de calibración. Posteriormente, puede ser utilizada para predecir valores. Lógicamente, el objetivo es obtener un algoritmo capaz de predecir concentraciones con el mínimo error posible.

Debido a la gran cantidad de información disponible, el paso de calibración (en el que se busca el modelo óptimo) puede necesitar largo tiempo y resultar tedioso si se utilizan técnicas manuales (prueba y error). En este articulo se propone una solución automatizada, presentando distintas aproximaciones para el problema del calibrado, incluyendo el uso de Algoritmos de Estimación de Distribuciones (EDAs) paralelos.

Cuando se trabaja con problemas donde cada muestra contiene miles de variables (como es nuestro caso), es necesario completar una fase previa: reducir el número de variables, buscando aquellas que contienen la información más relevante. Existen distintas aproximaciones que puede ser utilizadas para este fin: Construcción de Características y Selección de Conjuntos de Características (FSS) [10].

Las técnicas de Construcción de Características buscan las posibles relaciones entre las variables y devuelven un nuevo conjunto de variables, combinando las originales. Por otro lado, FSS busca las variables más significativas, devolviendo un subconjunto del conjunto de variables originales.

En FSS podemos encontrar dos aproximaciones diferentes: "filter" y "wrapper". La aproximación "filter" selecciona las variables basándose en características intrínsecas del conjunto de datos. Sin embargo, la técnica "wrapper" tiene en cuenta el objetivo final de las variables seleccionadas; por ejemplo, en un problema de clasificación, esta técnica evalúa cada subconjunto de variables en función de la precisión del clasificador que es construido a partir del subconjunto de variables.

Como aproximación inicial se aplicaron dos técnicas: Análisis de Componentes Principales (PCA) y "filter". Como los resultados obtenidos mediante dichas aproximaciones no fueron satisfactorios, se estudió una solución más compleja, combinando técnicas de "filter" y "wrapper" en dos pasos consecutivos. Dentro de las distintas soluciones "wrapper" presentadas en los últimos años, hemos decidido utilizar EDAs debido a los prometedores resultados obtenidos en trabajos previos dentro del área de FSS [5, 6], aunque lógicamente, somos conscientes de la existencia de otros métodos igualmente eficaces [2].

Adicionalmente, como éste tipo de algoritmos (EDAs) requieren un tiempo computacional notable, presentamos una implementación paralela que posibilita el uso de estos algoritmos en problemas que eran inabordables utilizando un solo ordenador.

La organización del artículo es la siguiente: La sección 2 comienza con una descripción del problema y a continuación se introducen las dos aproximaciones iniciales. Posteriormente, Sección 3, se describen los EDAs así como una implementación paralela. La Sección 4 recoge los resultados obtenidos con las distintas aproximaciones para finalizar en la Sección 5 con las conclusiones y líneas de trabajo futuras.

2. Aproximaciones iniciales

En esta sección se presenta una descripción detallada del problema al que nos enfrentamos, así como dos técnicas que fueron utilizadas inicialmente para resolverlo.

2.1. Descripción del problema y del módulo de predicción

Se trata de crear un modelo capaz de predecir las concentraciones de tres analitos (formaldehído, glioxal y glutaral) dentro de una solución química. Para la fase de calibración, disponemos de 181 muestras. Las variables que forman parte de la muestra representan mediciones de la absorción de luz para un rango determinado de longitudes de onda, muestreadas a lo largo del tiempo. El vector de muestras está organizado de la siguiente forma: 61

bloques consecutivos, donde cada uno de ellos representa un intervalo de tiempo (desde 2 a 602, en lapsos de 10 segundos). Dentro de cada bloque, 91 valores representan la absorción de luz para un rango de longitudes de onda (entre 290 y 470nm, con distancias de 2nm). Por lo tanto, el fichero de casos almacena 181 experimentos, con 5551 características y 3 valores objetivo (aquellos a predecir), todos continuos. La concentración de las tres variables objetivo dentro del conjunto de entrenamiento ha sido normalizada a valores en el rango [0,1].

Como módulo de predicción hemos utilizado una red neuronal artificial (ANN) [11]. Básicamente, una ANN puede ser definida como un conjunto de unidades (nodos) simples de procesamiento que trabajan conjuntamente intercambiando información entre ellas. Cada unidad almacena la información que recibe y genera un valor de salida que depende de una función interna.

Dentro de la variedad de aproximaciones descritas en la literatura, nos hemos decantado por el modelo conocido como Perceptron Multicapa (MLP) [15]. En dicho modelo, los nodos son organizados en capas: una de entrada, una de salida, y varias capas intermedias (ocultas), siendo la comunicación posible sólo entre las capas adyacentes. Una vez que la estructura ha sido definida (número de capas y nodos por capa), el paso final requiere que los pesos de la red sean ajustados, de tal forma que produce el resultado deseado cuando es confrontado con unos determinados datos de entrada. A este proceso se le denomina entrenamiento. Como sucede con la estructura, existen distintas propuestas para completar el entrenamiento. Entre todas ellas, hemos seleccionado una aproximación clásica denominada Backpropagation [16].

Una vez que el algoritmo de predicción ha sido seleccionado, necesitamos considerar la forma de utilizarlo. Como el problema tiene más de cinco mil variables, alimentar directamente la red neuronal con todas ellas podría ser una propuesta inicial. Sin embargo, los resultados obtenidos son bastante pobres: utilizar tantas variables como entrada dificulta la labor de la red MLP.

Por lo tanto, el modelo presentado se compone de dos módulos. El primero, selección, coge como entrada todo el conjunto de datos y lo reduce considerando las variables más representativas o las componentes principales (en función de la técnica utilizada); y el segundo, la utilización de una MLP que toma como entrada las variables seleccionadas en el primer paso y devuelve valores para las tres variables que han de ser predichas.

Debido al reducido número de muestras disponibles, hemos decidido completar una validación cruzada de 5 partes (5-fold crossvalidation) para medir la precisión del modelo. En esta técnica, el conjunto de datos es dividido aleatoriamente en k partes, utilizando $k-1$ para entrenar el modelo y una para comprobar su bondad. El proceso es repetido k veces, variando la parte usada para la evaluación y por lo tanto, para el entrenamiento. Para medir la precisión, se ha utilizado un error global, calculando la media de los errores para cada objetivo. Concretamente, el error sobre cada objetivo se ha calculado mediante el cuadrado de la diferencia entre el valor predicho y el valor real.

En las siguiente secciones procederemos a explicar el uso de las técnicas PCA y "filter".

2.2. Aproximación PCA

Mediante esta técnica se extraen las características principales del conjunto de datos, y posteriormente, se entrena la red MLP para construir un módulo de predicción. PCA utiliza procesos matemáticos que transforman un número de variables (posiblemente) relacionadas en un conjunto (menor) de variables no relacionadas denominadas componentes principales [7].

Una vez que el conjunto de datos con las componentes principales está disponible, podemos probar diferentes configuraciones estructurales sobre MLP de cara a seleccionar la mejor. Para ello, hemos utilizado un procedimiento de fuerza bruta, probando diferentes configuraciones para las capas de entrada y ocultas. Evidentemente, era necesario fijar un límite para este proceso de prueba y error, debido a la enorme cantidad de configuraciones

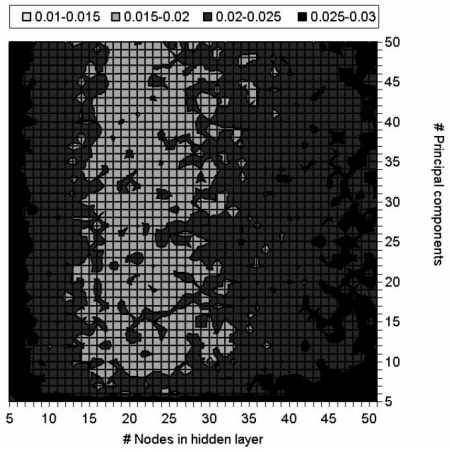

Figura 1: Error cuadrático medio para distintas configuraciones de MLP y PCA

posibles que se podrían probar. En concreto, hemos utilizado una única capa oculta. Como el número de nodos para la capa de salida está fijado (3, las variables a predecir), es necesario determinar el número de nodos a utilizar en la capa de entrada y en la oculta.

El número de nodos de la capa de entrada dependerá del número de componentes principales que queremos incorporar a nuestro modelo. A priori, no conocemos cuantas de ellas son realmente interesantes y tampoco el número apropiado de neuronas para la capa oculta. Por ello, siendo esta una aproximación inicial, se han realizado distintas pruebas variando el número de componentes principales entre 5 y 50 así como el número de nodos en la capa intermedia (también entre 5 y 50). Los resultados obtenidos se muestran en un mapa de error (Figura 1) donde cada punto (x, y) del mapa representa la media del error cuadrático para una configuración con x nodos ocultos e y nodos de entrada.

Puede observarse que las configuraciones con pocos (<15) o demasiados (>27) nodos intermedios producen grandes errores. Respecto al número de componentes principales, se necesitan más de 7 para conseguir configuraciones interesantes. El mejor modelo obtenido puede ser consultado en el Cuadro 2.

2.3. Aproximación Filter

En la literatura se han descrito distintos métodos para completar aproximaciones "filter". Generalmente, estas técnicas realizan una evaluación univariada para cada variable, asignándole un valor. Una vez que todas las variables han sido evaluadas, disponemos de una lista ordenada (ranking) basado en la relevancia de cada variable.

La aproximación utilizada en este articulo, es la denominada Correlation-based Feature Selection (CFS), presentada en [4]. CFS es una aproximación multivariada, es decir, es capaz de evaluar la bondad de subconjuntos de variables, devolviendo como resultado las variables más relevantes. Dicho método requiere que todos los datos sean discretos, y por lo tanto, ha sido necesario un proceso previo de discretización, completado aplicando uno de los algoritmos más utilizados, el denominado método de la Entropía o Fayyad-Irani [3].

Mediante el proceso de filtrado se han obtenido un total de 31 variables relevantes, con lo que la configuración de la capa de entrada de la red MLP queda fijada. Tal y como se ha hecho para el caso de PCA, 46 configuraciones diferentes han sido probadas, variando el número de nodos de la capa intermedia entre 5 y 50. Los errores obtenidos son mostrados en la Figura 2. Puede observarse que, en términos generales, cuantos más nodos se utilizan en la capa oculta, peores resultados se obtienen. Esta segunda aproximación (ver Cuadro 2), mejora los resultados obtenidos mediante la técnica PCA pero todavía resulta insuficiente para los usuarios de este modelo.

Por lo tanto, tratando de reducir aun más el error, hemos probado una aproximación más compleja: filter+wrapper, basada en técnicas de ordenación y EDAs paralelos.

3. EDAs y propuesta paralela

La estructura de los EDAs (introducidos en [13]) es similar a la de los Algoritmos Genéticos (AGs) pero, en vez de utilizar operadores de cruce y mutación para obtener la nueva población, ésta es obtenida muestreando una dis-

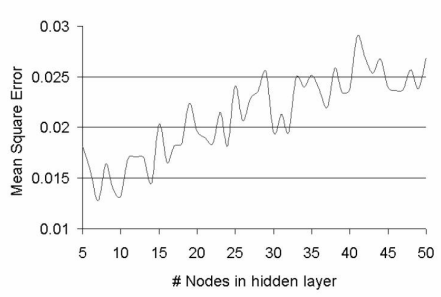

Figura 2: Error cuadrático medio para la aproximación "filter" utilizando diferentes configuraciones de MLP

tribución de probabilidad. Dicha distribución de probabilidad es estimada a partir de una selección de individuos de la población anterior. Por lo tanto, las relaciones entre las distintas variables que representan a los individuos están expresadas explícitamente a través de la distribución de probabilidad conjunta asociada a los individuos seleccionados en cada generación. Un esquema típico para los EDAs es el siguiente:

1. Generar la primera población con M individuos y evaluarlos.
2. N individuos son seleccionados de la población de M individuos, siguiendo un determinado método de selección.
3. Partiendo de N, se induce un modelo de probabilidad n-dimensional que muestra las interdependencias entre las variables, donde n es el tamaño del individuo.
4. Finalmente, una nueva población de M individuos es generada basada en el muestreo de la distribución de probabilidad obtenida en el paso anterior.

Los pasos 2, 3 y 4 se repiten hasta que se cumpla una condición de parada (por ejemplo, un número máximo de generaciones).

El paso más importante es el tercero, donde se detectan las dependencias entre las variables. Para completar este paso se han propuesto distintas aproximaciones, tanto en el domino discreto como en el continuo. Para obtener más información sobre EDAs y sus características consúltese [9, 14].

Nuestro interés por el uso de EDAs está motivado por resultados obtenidos en trabajos previos [5, 6]. No obstante, estos trabajos informan de la dificultad para aplicar EDAs multivariados (por ejemplo, Algoritmos de Estimación de Redes Bayesianas (EBNA)) en problemas con más de cien variables, debido al coste computacional.

Por lo tanto, seria interesante diseñar algoritmos paralelos que permitan reducir el tiempo de ejecución manteniendo la funcionalidad. Concretamente, nos hemos centrado en el algoritmo $EBNA_{BIC}$ [8] aunque el esquema se puede aplicar fácilmente a otros $EBNAs$.

La fase de aprendizaje del $EBNA_{BIC}$ implica el aprendizaje de una red Bayesiana a partir de los individuos seleccionados en cada generación, esto es, aprender la estructura (el grafo S) y los parámetros (las distribuciones de probabilidad local $\boldsymbol{\theta}$).

Existe además otra fase común para todos los EDAs, "muestreo y evaluación", que puede requerir largos tiempos de computo cuando la evaluación del individuo es compleja (tal y como sucede en el caso que nos ocupa).

Por lo tanto, en las siguientes secciones describiremos ambas fases así como sus aproximaciones paralelas.

3.1. La fase de aprendizaje

Durante esta fase, se aprende una red Bayesiana utilizando un algoritmo voraz. Cada posible red (estructura) tiene asignado un valor que representa su idoneidad para la presente población. La búsqueda se realiza añadiendo o borrando arcos de la red siempre y cuando se mejore el valor de la red actual.

Lógicamente, el criterio utilizado durante este proceso juega un papel muy importante dentro del algoritmo, pues influye directamente sobre la red Bayesiana obtenida. EBNA$_{BIC}$ utiliza el criterio conocido como BIC (Bayesian Information Criterion) [17]. Dada una estructura S y un conjunto D de individuos seleccionados, el criterio BIC se puede definir como:

$$BIC(S,D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \quad (1)$$
$$-\frac{1}{2}\log(N)\sum_{i=1}^{n} q_i(r_i-1)$$

donde:

- n es el número de nodos (variables) de la red Bayesiana (tamaño del individuo).
- r_i es el número de valores diferentes que cada variable X_i puede tomar.
- q_i es el número de valores diferentes que las variables padre de X_i, $\boldsymbol{Pa_i}$, pueden tomar.
- N_{ij} es el número de individuos en D en los que las variables $\boldsymbol{Pa_i}$ toman su valor j-ésimo.
- N_{ijk} es el número de individuos en los que la variable X_i toma su valor k-ésimo y las variable $\boldsymbol{Pa_i}$ su valor j-ésimo.

Una propiedad importante de este criterio es que se puede descomponer, es decir, puede ser calculado en base a la suma de los criterios locales de las variables. Cada variable X_i tiene un valor BIC asociado:

$$BIC(i,S,D) = \sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \quad (2)$$
$$-\frac{1}{2}\log(N)q_i(r_i-1)$$

En cada paso se completa una búsqueda exhaustiva sobre el conjunto de posibles modificaciones de arcos, seleccionando aquella que maximice el valor. Dicha modificación (borrado/adición) será posible siempre y cuando la red mantenga una estructura acíclica. El proceso continúa hasta que ninguna modificación consiga mejorar el valor de la red actual. Además, es importante tener en cuenta que si en la estructura S se modifica el arco (j,i), entonces solo será necesario calcular el criterio $BIC(i,S,D)$.

Para implementar la versión paralela nos hemos basado en el conocido esquema maestro-esclavo. El maestro ejecuta el programa y cuando llega a una fase costosa, reparte las tareas entre los esclavos. Después, recoge los resultados obtenidos y continúa con la ejecución secuencial. Dependiendo del problema, puede resultar interesante diseñar el maestro de forma que también se comporte como un esclavo cuando el trabajo es repartido (aprovechando así el tiempo de espera).

En relación al algoritmo, hemos explicado que la fase de aprendizaje consiste en un proceso en el que en cada paso se calculan las modificaciones de arcos que pueden mejorar el valor del criterio BIC. Por lo tanto, el conjunto de modificaciones a calcular será repartido entre los esclavos (maestro incluido), y estos devolverán los valores correspondientes a cada una de las modificaciones. Una vez recogidos todos los resultados, el maestro selecciona el mejor, aplica el cambio de arco correspondiente (informando a los esclavos para que hagan lo propio) y el proceso se repite. Para una información detallada sobre esta y otras implementaciones paralelas consúltese [12].

3.2. La fase de "muestreo y evaluación"

Una vez que la red Bayesiana ha sido creada, en esta fase se crean (muestrean) nuevos individuos. Una vez obtenidos, se procede a evaluar cada uno de ellos. Tal y como hemos comentado anteriormente, el hecho de utilizar una red MLP para calcular la idoneidad de cada individuo hace que la ejecución de la función de evaluación sea costosa.

Por lo tanto, se puede implementar una aproximación paralela sencilla, repartiendo el proceso de "muestreo y evaluación" entre el maestro y los esclavos. Al tratarse de máquinas de idénticas características, bastará con realizar un reparto homogéneo $NewPopSet/comp$, donde $NewPopSet$ es el número de individuos a crear y $comp$ el número de esclavos (maestro incluido) disponibles.

Finalmente, una vez que los individuos han sido creados y evaluados son enviados al maestro para continuar con el siguiente paso del algoritmo $EBNA_{BIC}$.

3.3. Adaptación del algoritmo para el problema de predicción

De cara a adaptar el algoritmo se ha fijado la siguiente configuración: cada variable X_i del individuo puede tomar dos valores: 0 ó 1 (1 implica que la i-ésima característica es seleccionada y 0 que no es seleccionada). Para evaluar la calidad de la solución (individuo), se completa una ejecución de la red MLP utilizando como entrada el subconjunto de características seleccionadas.

Una configuración típica para los EDAs es fijar un tamaño de población igual o mayor que el tamaño del individuo. Para este problema en particular, definir un individuo con más de cinco mil variables, implicaría que cada vez que una nueva población sea creada, más de cinco mil individuos deberían ser evaluados. Desgraciadamente, esto resulta excesivo incluso para la versión paralela.

Por lo tanto, es necesario aplicar previamente una técnica para reducir el número de variables sobre las que realizar la búsqueda "inteligente". Para completar este proceso, se han utilizado 6x2 rankings distintos, combinando seis métricas descritas en [1] (Información Mutua, Distancia Euclídea, Matusita, Kullback-Leibler modo 1 y 2, y Bhattacharyya) y dos discretizaciones con cada una de las variables a predecir. Finalmente, se ha creado una única lista ordenada utilizando un consenso de las anteriores.

Para completar los experimentos, el tamaño del individuo se ha fijado en 500 (se seleccionan las primeras 500 variables de la lista) y se ha limitado el número de generaciones a 80. Además, como nos interesa minimizar el número de variables seleccionadas, se ha establecido una probabilidad inicial de 0.05 (para cada variable) de ser seleccionada.

Tal y como ha ocurrido en los procesos previos, una vez que se ha fijado la capa de entrada, es necesario determinar el número de neuronas de la capa oculta. Debido a la alta demanda computacional de este algoritmo, sería costoso ejecutar un proceso de búsqueda y error, y por ello se ha decidido fijar el número de neuronas en 16 (este número produce resultados interesantes en las dos aproximaciones previas pero evidentemente se deberían probar más combinaciones). Por lo tanto, esta aproximación debe ser vista como un test inicial de cara a estudiar la viabilidad de los EDAs para resolver este tipo de problemas.

Cuadro 1: Tiempos de ejecución

Aprox.	*Etapa*	*#CPUs*	*Tiempo*
PCA	Obtener comps.	1	6h
	MLP	22	1h
Filter	Obtener vars.	1	1h
	MLP	22	5'
EBNA	Obtener vars.	1	4h
	EBNA+MLP	22	18h

Cuadro 2: Error mínimo total

Aprox.	*#Ipt*	*#Ocu*	*Error*	*Desv*
PCA	41	20	$13.958e^{-3}$	$1.786e^{-3}$
Filter	31	7	$12.811e^{-3}$	$2.227e^{-3}$
EBNA	60	16	$8.367e^{-3}$	$0.236e^{-3}$

4. Resultados de los experimentos

Para los experimentos se ha utilizado un cluster linux de 11 nodos biprocesador (AMD Athlon MP 2000). Los tiempos empleados por cada uno de los algoritmos está recogido en el Cuadro 1.

En el Cuadro 2 se recogen los mejores resultados para cada una de las aproximaciones, así como la desviación (de un total de 10 ejecuciones). Se puede observar que los resultados más prometedores son aquellos obtenidos utilizando la aproximación $EBNA_{BIC}$. Sin embargo, después de analizar los resultados de manera individual para cada variable, hemos observado que un error medio aceptable está ocultando un error alto en la tercera variable (ver Cuadro 3).

Cuadro 3: Error cuadrático medio por variable

Aprox.	*formald.*	*glioxal*	*glutaral*
PCA	$6.759e^{-3}$	$6.273e^{-3}$	$28.785e^{-3}$
Filter	$4.065e^{-3}$	$4.697e^{-3}$	$29.824e^{-3}$
EBNA	$2.358e^{-3}$	$3.576e^{-3}$	$20.025e^{-3}$

5. Conclusiones y líneas futuras

El presente artículo recoge algunas aproximaciones prometedoras para predecir valores de concentración en una reacción química. Como planteamiento inicial, se han utilizado las técnicas PCA y "filter" pero, debido a la baja calidad de los resultados obtenidos se optó, en una segunda fase, por combinar las técnicas "filter+wrapper" utilizando algoritmos evolutivos. Más concretamente, se ha presentado una implementación paralela de un EDA, $EBNA_{BIC}$, que permite el uso de este tipo de algoritmos para resolver problemas donde la aproximación secuencial resulta inviable.

En cuanto al problema, existen aspectos del planteamiento inicial que pueden ser mejorados. Por ejemplo, respecto a la estructura de la red MLP, hemos realizado una aproximación inicial basada en los resultados obtenidos con las técnicas anteriores, pero existen planteamientos mejores, tales como aplicar algoritmos que busquen configuraciones que mejoren el funcionamiento de la red de predicción.

Por otro lado, atendiendo a los resultados, se puede observar que existe un desequilibrio en el error obtenido para cada variable predicha. En esta situación, sería interesante valorar la posibilidad de utilizar una aproximación multi-objetivo de cara a presentar un conjunto de soluciones no dominadas que ofrecen un amplio abanico de posibilidades a fin de que se pueda seleccionar la aproximación más interesante en cada caso.

Referencias

[1] M. Ben-Bassat. Use of distance measures, information measures and error bounds in feature evaluation. In P. R. Krishnaiah and L.Ñ. Kanal, editors, *Handbook of Statistics*, volume 2, pages 773–791. North-Holland Publishing Company, 1982.

[2] E. Cantú-Paz, "Feature subset selection, class separability, and genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-04* (K. D. et al, ed.), vol. 2, (Berlin), pp. 959–970, Springer Verlag, 2004.

[3] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993.

[4] M. A. Hall and L. A. Smith. Feature subset selection: A correlation based filter approach. In N. K. et al., editor, *Proceedings of the Fourth International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858, Dunedin, 1997.

[5] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. Feature subset selection by Bayesian networks based optimization. *Artificial Intelligence*, 123(1-2):157–184, 2000.

[6] I. Inza, P. Larrañaga, and B. Sierra, "Feature Subset Selection by Estimation of Distribution Algorithms," in *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation* (P. Larrañaga and J. A. Lozano, eds.), pp. 269–294, Kluwer Academic Publishers, 2002.

[7] I. T. Jolliffe. Principal Component Analysis. Springer-Verlag, 1986.

[8] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Combinatorial optimization by learning and simulation of Bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI 2000*, pages 343–352, Stanford, CA, USA, 2000.

[9] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, 2002.

[10] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic Publishers, 1998.

[11] J. L. McClelland and D. E. Rumelhart. Explorations in the microstructure of cognition. In *Parallel Distributed Processing.* The MIT Press, 1986.

[12] A. Mendiburu, J. Miguel-Alonso and J. A. Lozano, "Parallel implementation of EDAs based on probabilistic graphical models" *IEEE Transactions on Evolutionary Computation.* In Press.

[13] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions i. binary parameters. In *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, 1996.

[14] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[15] F. Rosenblatt. *Principles of Neurodynamics.* Spartan Books, New York, 1962.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by backpropagating errors. *Nature*, 323:533–536, 1986.

[17] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 7(2):461–464, 1978.