USO DE BÚSQUEDA TABÚ PARA SELECCIÓN DE VARIABLES EN CLASIFICACIÓN

Casado Yusta Silvia
Departamento de Economía Aplicada
Universidad de Burgos
e-mail: scasado@ubu.es

Gomez Palacios Olga
Departamento de Economía Aplicada
Universidad de Burgos
e-mail: olga_gomezpalacios@yahoo.es

Núñez Letamendia Laura
Departamento de Finanzas
Instituto de Empresa
e-mail: laura.nunez@ie.edu

Pacheco Bonrostro, Joaquín
Departamento de Economía Aplicada
Universidad de Burgos
e-mail: jpacheco@ubu.es

Resumen

En este trabajo se propone un método de selección de variables para su uso posterior en el análisis discriminante. El objetivo consiste en encontrar de entre un conjunto de *m* variables, un pequeño subconjunto de éstas que nos permitan clasificar de forma eficaz obteniendo un porcentaje de aciertos elevado. Reducir el número de variables conlleva una serie de ventajas como reducción en el coste de

adquisición de datos, mejora en la comprensión del modelo final del clasificador e incremento de la eficiencia y eficacia del clasificador . Dado que el objetivo principal es reducir el número de variables manteniendo un porcentaje de aciertos elevado, el problema concreto consiste en encontrar, para un valor entero p pequeño, el subconjunto de tamaño p de variables originales que dan lugar al mayor porcentaje de aciertos en el análisis discriminante. Para la resolución de este problema se propone una técnica basada en la estrategia metaheurística Búsqueda Tabú y se comprueba que obtiene significativamente mejores resultados que los métodos tradicionalmente usados por los paquetes estadísticos clásicos como Stepwise, Backward y Forward.

Palabras clave: Selección de variables, análisis discriminante, estrategias metaheurísticas, Búsqueda Tabú.

Area temática: métodos cuantitativos.

1.- Introducción

En el problema de clasificación el objetivo es determinar la clase a la que pertenecen una serie de individuos caracterizados por atributos o variables. A través de un conjunto de ejemplos (de los que se conoce su clase) se pretende diseñar y generalizar el conjunto de reglas que permita clasificar, con la mayor precisión posible el conjunto de individuos.

Existen varias metodologías para abordar este problema: Análisis Discriminante clásico, Regresión Logística, Redes Neuronales, Árboles de decisión, Instance-Based Learning, etc. En la mayor parte de los métodos de análisis discriminante se buscan los hiperplanos en el espacio de las variables que separan mejor a las clases de individuos. Esto se traduce en buscar funciones lineales a partir de las cuales realizar la clasificación (Wald, Fisher,etc). El uso de funciones lineales permite una mejor interpretación de los resultados (por ejemplo importancia y/o significación de cada variable en la clasificación) al analizar el valor de los coeficientes obtenidos. No todos los métodos de clasificación resultan tan cómodos para realizar este tipo de análisis, (a algunos incluso se les clasifica como modelos de "caja negra"). Por esta razón el análisis discriminante clásico sigue siendo una metodología interesante.

Cuando se tienen muchas variables, antes del diseño de cualquier método de clasificación, es necesario seleccionar de las variables originales aquellas que son realmente necesarias. Es decir eliminar del análisis aquellas menos significantes. Así, el problema consiste en encontrar un subconjunto de variables con las que se pueda llevar a cabo la tarea de clasificar de forma óptima. Este problema es conocido como problema de selección de variables. La investigación en este campo empezó a principios de la década de los sesenta, Lewis (1962) y Sebestyen (1962). Según Liu y Motoda (1998) la selección de variables conlleva diversas ventajas como la reducción del coste en la adquisición de datos, mejora en la comprensión del modelo final del clasificador, incremento en la eficiencia del clasificador y mejora en la eficacia del clasificador. Durante las cuatro últimas décadas se ha investigado mucho en este problema. Muchos trabajos sobre selección de variables están relacionados con la medicina y la biología, tales como Sierra y otros (2001), Ganster y otros (2001), Inza y otros (2002), Lee y otros (2003), Shy y Suganthan (2003) y Tamoto y otros (2004).

Desde un punto de vista computacional la búsqueda del subconjunto de variables es un problema NP-Hard, (Kohavi (1995) y Cotta y otros (2004)), es decir que no existe ningún método que garantice la solución óptima en un tiempo polinomial en el tamaño de problema, (NP = *No polinomic*). Esto significa que, en la practica, encontrar la solución óptima cuando el tamaño del problema es grande es inviable. Para este tipo de problemas se desarrollan dos tipos de algoritmos o métodos: los métodos exactos u óptimos, que encuentran la solución óptima pero que solo son aplicables cuando el tamaño del problema es pequeño; y los métodos aproximados o heurísticos que, aunque no garantizan el óptimo, encuentran buenas soluciones en un tiempo razonable. Entre los métodos exactos el más conocido es el de Narendra y Fukunaga (1977), pero como señala Jain y Zongker (1997) el algoritmo es inviable para problemas con un número elevado de variables. Por otra parte la calidad de los soluciones "heurísticas" difiere mucho según los métodos empleados. Como en otros problemas de optimización las estrategias metaheurísticas están demostrando ser metodologías superiores al resto. Así destacan los trabajos de Bala y otros (1996), Jourdan y otros (2001), Oliveira y otros (2003), Inza y otros (2001a, 2001b) y Wong

y Nandi (2004) que desarrollan algoritmos genéticos y el más reciente de Garcia y otros (2003) que presenta un método basado en Búsqueda Dispersa.

Estos métodos buscan subconjuntos de mayor capacidad clasificatoria según diferentes criterios, pero ninguno de ellos está enfocado al uso posterior de las variables seleccionadas en análisis discriminante. En este trabajo se propone el diseño "ad-hoc" y comparación de un método de selección de variables para análisis discriminante. En la literatura para este específico propósito solo existe el conocido método *Stepwise* y variantes como O'Gorman (2004), así como los métodos *Backward* y *Forward*. Estos son sencillos procedimientos de selección basados en criterios estadísticos (Landa de Wilks, F de Fisher, etc.) y que han sido incorporados en algunos de los paquetes estadísticos más conocidos como SPSS, BMDP, etc. Como destaca Huberty (1989) y Salvador (2000) estos métodos no son muy eficaces, y cuando las variables originales son muchas raramente alcanzan el óptimo. El método que proponemos en este trabajo consigue, como se verá posteriormente, significativamente mejores resultados.

En este trabajo se diseña un método basado en la estrategia metaheurística Búsqueda Tabú que consta de un método para encontrar la solución inicial (método constructivo) y un procedimiento básico de búsqueda tabú que se complementa con Intensificación y Diversificación. Posteriormente se analizan sus componentes y se estudia su eficacia comparándolo con los métodos anteriores (*Stepwise*, *Backward* y *Forward*). Este trabajo esta enfocado al análisis discriminante de 2 grupos. En trabajos posteriores se adaptarán los métodos al caso de más grupos.

El trabajo se estructura de la siguiente forma: En la sección 2 se modeliza el problema y en la sección 3 se describe el método usado para generar la solución inicial, constituido por un método constructivo y un procedimiento de Búsqueda Local. En la sección 4 se describe el algoritmo de Búsqueda tabú básico y en la 5 se describe el algoritmo de Búsqueda Tabú completo, es decir, el básico reforzado con estrategias de intensificación y diversificación. Posteriormente en la penúltima sección se muestran los resultados de las experiencias computacionales realizadas y finalmente en la sección 7 se exponen las principales conclusiones obtenidas.

2.- Modelización del problema

El problema de seleccionar el subconjunto de variables con más capacidad clasificatoria se puede formular de la forma siguiente: Considérese V un conjunto de m variables, simplificando $V = \{1, 2,..., m\}$, y A un conjunto de n casos. Para cada caso se conoce además la clase a la que pertenece. Considérese un valor predeterminado $p \in N$, $p < m$. Hay que encontrar el subconjunto $S \subset V$, de tamaño p con la mayor capacidad clasificatoria en análisis discriminante, $f(S)$.

Más concretamente la función $f(S)$ se define como el porcentaje de aciertos en A del clasificador de Wald (Hair y otros 2001) obtenido a partir de las variables de S. Obsérvese que este clasificador se obtiene a partir de la distancia de Mahalonobis a los centroides de cada grupo. Por tanto la evaluación de $f(S)$ se puede hacer en $\theta(p^2)$ + $\theta(n)$ operaciones: el primer sumando corresponde al calculo de la inversa de la matriz de varianzas-covarianzas de S, y el segundo a la clasificación de los casos. En definitiva, que a la vez que se calcula $f(S)$ se obtiene como paso previo el clasificador lineal de Wald.

3. Generación de la solución inicial

La solución inicial se genera mediante un método constructivo y un método de mejora, es decir cada solución S generada por el método constructivo es mejorada posteriormente por un procedimiento de búsqueda local.

El funcionamiento del método constructivo es el siguiente: Se parte de solución inicial vacía ($S = \emptyset$); en cada paso o iteración se elige un elemento que no esté en la solución y se añade; el proceso finaliza cuando la solución esta completa ($|S| = p$). Para determinar que elemento se añade se hace uso de una función guía (función *voraz*) R_j, j=1..m, que sirve como indicador de la bondad o conveniencia de cada

elemento candidato a entrar en la solución, eligiendo el elemento de mayor valor según esa función guía.

El funcionamiento del método constructivo en seudocódigo es el siguiente:

Iniciar $S = \emptyset$

Repetir

Calcular R_j, $\forall j \in V - S$

Determinar $Rmax = max \{ R_{j*} / j* \in V - S \}$

Make $S = S \cup \{j*\}$

hasta $|S| = p$

En este trabajo se van a contrastar 3 funciones voraces o formas de calcular R_j diferentes, basadas en la F de Snedecor, λ de Wilks y la propia función objetivo *f*. Concretamente, sea una solución parcial *S*, $\forall j \in V–S$, R_j se puede calcular como sigue:

a) se calculan los residuos en un modelo de regresión lineal donde *j* es la variable dependiente y las independientes son los elementos de *S*. El valor R_j va a ser el cociente varianza entre grupos/varianza intragrupos de estos residuos. El objeto de la regresión es eliminar información redundante. En la iteración inicial ($S = \emptyset$) R_j va a ser el cociente varianza entre grupos/varianza intragrupos de la variable original *j*.

b) Se determinan las matrices de varianzas-covarianzas entre grupos e intragrupos del conjunto $S \cup \{j\}$, que denotamos por B_j y W_j . R_j se calcula como el determinante de $W_j^{-1} \cdot B_j$.

c) R_j se calcula como $f(S \cup \{j\})$.

Los 2 primeros funciones se corresponden con la F de Snedecor y la inversa de la λ de Wilks, 2 criterios muy usados por los paquetes estadísticos en los métodos de selección que incorporan. El tercero evalúa directamente la capacidad clasificatoria

según se definió en 2. En el apartado 5 se comprueba como en efecto este último criterio es el más adecuado.

Tal y como se comenta al principio de esta sección, a cada solución completa *S* generada por el método constructivo se la mejora posteriormente por un sencillo procedimiento de búsqueda local. En este caso, cada paso de la búsqueda local va a consistir en intercambiar una variable que esté dentro de la solución por otra que este fuera. Más concretamente, sea *S* una solución se define

$$N(S) = \{ S' / S' = S \cup \{j'\} - \{j\} \ , \forall \ j \in S, j' \notin S \}$$

El procedimiento de búsqueda local se puede describir como sigue

Leer Solución Inicial S
Repetir
    Hacer valor_ant = f(S)
    Buscar f(S) = max { f(S') / S' ∈ N(S) }*
    Si f(S) > f(S) entonces hacer S = S**
hasta f(S) ≤ valor_ant*

Como se observa el procedimiento finaliza cuando ningún intercambio produce mejora.

4.- Algoritmo "básico" de Búsqueda Tabú

Los primeros trabajos sobre *Búsqueda Tabú* con tal nombre se deben a Glover (1989) y (1990), aunque los principios básicos ya se explicaron en un trabajo anterior del mismo autor en 1986 (Glover, 1986). Esta estrategia está teniendo grandes éxitos y mucha aceptación en los últimos años. Según su creador, es un procedimiento que "*explora el espacio de soluciones más allá del óptimo local*", (Glover y Laguna, 1993). Se permiten cambios *hacia arriba* o que empeoran la solución, una vez que se llega a un óptimo local. Simultáneamente los últimos movimientos se califican como *tabús* durante las siguientes iteraciones para evitar que se vuelva a soluciones anteriores y el algoritmo cicle. El termino tabú hace referencia a "*un tipo de*

inhibición a algo debido a connotaciones culturales o históricas y que puede ser superada en determinadas condiciones...". (Glover, 1996). Recientes y amplios tutoriales sobre Búsqueda Tabú, que incluyen todo tipo de aplicaciones, pueden encontrarse en Glover y Laguna (1997) y (2002).

A continuación se describe un algoritmo básico de Búsqueda Tabú que usa los mismos movimientos vecinales que el procedimiento de búsqueda local descrito en el apartado 3. Estos movimientos consisten en intercambiar en cada paso un elemento que esta en la solución *S* por otro de fuera. Para evitar que el algoritmo cicle, cuando se lleva a cabo un movimiento, que consiste en intercambiar *j* de *S* por *j'* de *V-S* , se impide que el elemento *j* vuelva a *S* durante un número prefijado de iteraciones. Concretamente, se define

vector_tabu (*j*) = número de la iteración en la que el elemento *j* salio de *S* .

Algunos movimientos "tabús" pueden ser permitidos en determinadas condiciones ("criterio de aspiración"), por ejemplo cuando mejora la mejor solución encontrada. El procedimiento de Búsqueda Tabú se describe a continuación, donde *S* es la solución actual y *S** la mejor solución encontrada. El paramétro *Tabu_Tenure* indica el número de iteraciones durante las cuales no se permite volver a la solución a un elemento que salio de la misma anteriormente. Tras realizar una serie de pruebas, el valor de *Tabu_Tenure* fijado coincide con *p*.

Procedimiento básico de Búsqueda Tabú

Leer solucion inicial S

Hacer *vector_tabu(j) = - Tabu_Tenure, j =1..m; niter =0, and S* = S*

Repetir

niter = niter+1

Calcular $v_{jj'} = f(S \cup \{j'\} - \{j\})$

Determinar $v_{j^*j'^*} = min \{v_{jj'} / \forall\ j \in S,\ j' \notin S$ *verificando:*

niter > vector_tabu(j') + Tabu_Tenure or

$v_{jj'} > f(S^*)$ *('criterio de aspiración')}*

Hacer $S = S \cup \{j'^*\} - \{j^*\}$ *and vector_tabu (j*) = niter*

Si *f (S) < f(S*) entonces hacer: S* = S, f* = f y iter_better = niter;*

hasta niter > iter_better+2·m

El procedimiento básico finaliza cuando trascurren $2 \cdot m$ iteraciones sin mejora.

5. Algoritmo "completo" de Búsqueda Tabú: Intensificación y Diversificación

En muchas aplicaciones de búsqueda tabú el procedimiento básico descrito en el apartado anterior puede ser reforzado con estrategias de *Intensificación* y *Diversificación*. La *Intensificación* consiste en explorar con mas detalle las regiones donde se encuentran las mejores soluciones encontradas hasta ese momento. La *Diversificación* se realiza después de la *Intensificación* y consiste en dirigir la búsqueda a regiones no exploradas. Un esquema del funcionamiento del algoritmo de Búsqueda Tabú completo es el siguiente.

Procedimiento completo de Búsqueda Tabú

Repetir

Ejecutar Búsqueda Tabú Básico

Ejecutar Intensificación

Ejecutar Diversificación

Hasta alcanzar una condición de parada

5.1 Intensificación

En nuestro caso la intensificación se realiza de la forma siguiente: se diseña una lista con los mejores óptimos locales diferentes encontrados hasta ese momento; cada par de soluciones diferentes de la lista se combinan para generar una nueva solución. Para combinar las soluciones usamos una estrategia denominada Path Relinking. La idea básica es la siguiente: entre las dos soluciones iniciales se construye un "camino" que las une. Un número de soluciones de ese "camino" o "cadena" son seleccionadas como nuevas soluciones. Estas soluciones intermedias se procura que

estén equidistantes entre sí. Posteriormente a esas soluciones intermedias se las aplica el método de mejora descrito en la sección 3. La figura 1 ilustra esta idea.

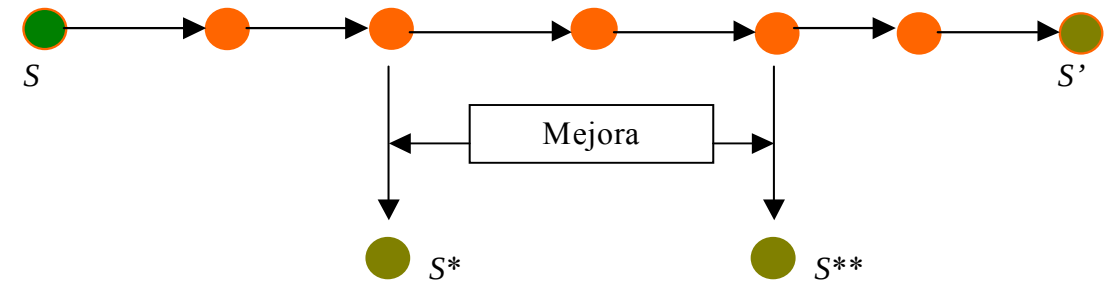

Figura 1.- *Generación de nuevas soluciones usando Path Relinking*

Desde cada par de soluciones, en la figura *S* y *S'* , se construye un camino que las une. Soluciones en posiciones intermedias en ese camino se seleccionan y se mejoran. De esta forma se generan las nuevas soluciones, (en la figura *S** y *S***).

Path Relinking es una estrategia que tradicionalmente se trató como un método de intensificación dentro de la Búsqueda Tabu aunque en los últimos años ha adquirido identidad propia. La idea que subyace es que en el camino entre dos buenas soluciones hay soluciones de igual o mejor calidad que las de partida. Ver Glover, Laguna and Martí (2000) para mayores detalles.

El número de óptimos locales de la lista es 6. Inicialmente se crea con los 6 mejores óptimos locales diferentes obtenidos en la primera fase básica. Esta lista se actualiza después de cada fase básica y después de cada intensificación. Una vez actualizada tras cada fase de intensificación se eliminan los 3 peores óptimos locales de la lista. De esta manera en la siguiente actualización entraran en la lista óptimos locales (al menos 3) de otras regiones del conjunto de soluciones. El objeto es asegurar en las siguientes intensificaciones combinar soluciones de diferentes regiones.

5.2 Diversificación

La diversificación se realiza de la manera siguiente: se construye una nueva solución inicial. Concretamente se usa el método constructivo descrito en el apartado 3 con la función guía modificada R'_j que se define como

$$R'_j = R_j - \beta \cdot R_{\max} \frac{freq(j)}{freq_{\max}}$$

donde:

$freq(j)$ = numero de veces que el elemento j aparece en las soluciones visitadas en las fases básicas hasta el momento;

$freq_{\max} = \max \{ freq(j) / j \in V\text{-}S \}$.

De esta manera se penaliza la elección de elementos que más han aparecido y se fuerza la elección de otros. En este trabajo se ha tomado el valor $\beta = 1$.

6.- Resultados computacionales

Para chequear y comparar la eficacia del método diseñado y sus componentes se han hecho una serie de pruebas con diferentes problemas tests. Para ello se hace uso de una tabla de datos financieros: las variables son ratios financieros y los casos son empresas españolas. La tabla consta de 141 ratios financieros para un total de 198

empresas (que se describe con detalle en Gómez y otros, 2004). El número de casos (empresas) que se consideran es 198, divididos en 2 clases (solventes y no solventes), con 131 y 67 elementos respectivamente.

Las pruebas se van a dividir en 3 grupos. En el primero se examinan las diferentes funciones guías propuestas en el apartado 3 y en el segundo se analizan y comparan las diferentes componentes de nuestro procedimiento de Búsqueda Tabú. Finalmente en el tercer grupo se compara el Tabú completo con los métodos tradicionales *Stepwise*, *Forward* y *Backward*.

6.1. Análisis de funciones guías

En estas pruebas se va a comparar la eficacia de las 3 funciones guías que se proponen para el constructivo descrito en la sección 3: F Snedecor, λ de Wilks y capacidad clasificatoria *f*. Para estas pruebas se usa la tabla de 141 ratios financieros para un total de 198 empresas. De esta tabla se consideran tablas de menores dimensiones, con un número *m* de ratios financieros para las 198 empresas. Así se consideran los siguientes valores de *m*, *m* = 40 (correspondientes a los primeros 40 ratios financieros), 65, 90, 105 y 120.

Para cada valor de *m* se consideran diferentes valores de *p*. Para cada problema se ejecuta el constructivo propuesto en la sección 3 usando cada una de las funciones guías. En la tabla 1 se muestran la capacidad clasificatoria *f* final obtenida.

m	*P*	F Snedecor	Inversa λ	*f*
40	4	0,71212121	0,71212121	0,77777778
40	5	0,72727273	0,72727273	0,77777778
40	6	0,73232323	0,73232323	0,78282828
40	7	0,73232323	0,73232323	0,76767677
40	8	0,72222222	0,72222222	0,76767677

m	*P*	F Snedecor	Inversa λ	*f*
65	6	0,74242424	0,74242424	0,78787879
65	7	0,76767677	0,76767677	0,78282828
65	8	0,77272727	0,75757576	0,77272727
65	9	0,78282828	0,76262626	0,76767677
65	10	0,77272727	0,78787879	0,76767677
90	8	0,75757576	0,75757576	0,83838384
90	9	0,77777778	0,77777778	0,79292929
90	10	0,77777778	0,77777778	0,80808081
90	11	0,78282828	0,78282828	0,7979798
90	12	0,77777778	0,77777778	0,84343434
105	10	0,76767677	0,76767677	0,81818182
105	11	0,76767677	0,76767677	0,79292929
105	12	0,77777778	0,77777778	0,82323232
105	13	0,78282828	0,78282828	0,81818182
105	14	0,78282828	0,78282828	0,85353535
120	12	0,81313131	0,78282828	0,82828283
120	13	0,81818182	0,81313131	0,82828283
120	14	0,7979798	0,79292929	0,84848485
120	15	0,81313131	0,78787879	0,82828283
120	16	0,81313131	0,78787879	0,82828283

Tabla 1. Comparación de funciones guía

En 22 de los 25 casos la propia capacidad clasificatoria *f* da lugar a mejores resultados que los otros 2 métodos más tradicionales. Estos obtienen resultados muy parecidos, aunque quizás los de la F de Snedecor sean ligeramente mejores. Aunque en este caso la propia función objetivo ha resultado ser la mejor función guía hay que hacer notar que en muchos problemas esto no siempre es así.

6.2. Análisis de los componentes del Algoritmo de Búsqueda Tabú completo

En este segundo grupo de pruebas se comparan las diferentes partes de nuestro procedimiento de Búsqueda Tabú. Es decir, se trata de comprobar como evolucionan las soluciones obtenidas cuando se ejecutan cada uno de estas componentes. Para ello se ejecuta el algoritmo completo con los mismos problemas que el subapartado anterior. Las fases o componentes de nuestro algoritmo de Búsqueda Tabú son:

- Obtención de una solución inicial (constructivo y búsqueda local)
- Ejecutar el procedimiento de Búsqueda Tabú básico a partir de la solución inicial
- Exploración de las regiones en torno a las mejores soluciones encontradas (*Intensificación*) y exploración de nuevas regiones (*Diversificación*).

En la tabla 2 se muestran los resultados obtenidos

M	p	Constructivo	Búsqueda local	T.Básico	Inten.+Diver
40	4	0,77777778	0,77777778	0,78282828	0,78282828
40	5	0,77272727	0,77777778	0,78282828	0,78787879
40	6	0,77777778	0,77777778	0,79292929	0,7979798
40	7	0,76767677	0,76767677	0,79292929	0,80808081
40	8	0,76767677	0,76767677	0,80808081	0,80808081
65	6	0,78787879	0,78787879	0,79292929	0,8030303
65	7	0,78282828	0,78282828	0,7979798	0,8030303
65	8	0,77777778	0,77777778	0,8030303	0,80808081
65	9	0,76767677	0,77272727	0,8030303	0,82828283
65	10	0,76262626	0,78282828	0,82828283	0,82828283
90	8	0,84343434	0,84343434	0,84343434	0,84848485
90	9	0,84343434	0,84848485	0,84848485	0,85858586
90	10	0,84343434	0,84343434	0,85858586	0,85858586
90	11	0,84343434	0,84343434	0,86868687	0,86868687
90	12	0,84343434	0,84343434	0,86363636	0,86363636
105	10	0,81313131	0,82323232	0,83333333	0,85858586
105	11	0,81818182	0,82323232	0,83333333	0,85858586
105	12	0,82323232	0,82323232	0,86868687	0,86868687
105	13	0,82323232	0,83333333	0,83838384	0,86363636
105	14	0,82323232	0,83333333	0,84343434	0,87373737
120	12	0,81313131	0,81313131	0,84848485	0,86363636

			Búsqueda		
M	**p**	**Constructivo**	**local**	**T.Básico**	**Inten.+Diver**
120	13	0,81313131	0,84848485	0,85858586	0,87878788
120	14	0,81313131	0,84848485	0,85858586	0,86363636
120	15	0,81313131	0,84848485	0,87373737	0,87373737
120	16	0,81313131	0,81313131	0,85858586	0,87373737

Tabla 2: Análisis de los componentes del algoritmo de Búsqueda Tabú completo

Obsérvese como en cada fase se mejora ligeramente lo obtenido en la fase anterior. Esto se hace más significativa a medida que aumenta el tamaño del problema (valores de p y m). Por ejemplo para $m = 120$, y $p = 13$ se pasa del 81.3% de aciertos en la solución obtenida por el método constructivo a 87.8 % obtenida por el algoritmo completo. Parece claro que todas las componentes juegan un papel importante.

6.3.- Busqueda tabú frente a métodos tradicionales

Finalmente se compara el algoritmo de Búsqueda Tabú con los procedimientos tradicionales *Stepwise*, *Backward* y *Forward* que usan algunos de los paquetes estadísticos más conocidos como SPSS, BMDP, etc. Para ello se usa la tabla original entera (es decir $m = 141$ ratios financieros).

El método *Forward* (o de *selección hacia delante*) comienzan eligiendo la variable que más discrimina según algún criterio. A continuación selecciona la segunda más discriminante y así sucesivamente. El algoritmo finaliza cuando entre las variables no seleccionadas ninguna discrimina de forma significativa.

El método *Backward* (o *de eliminación hacia detrás*) actúa de forma inversa. Se comienza seleccionando todas las variables. En cada paso se elimina la menos discriminante. El algoritmo finaliza cuando todas las variables que permanecen discriminan significativamente.

El método *Stepwise* o (*regresión por pasos*) utiliza una combinación de los dos algoritmos anteriores: en cada paso se introducen o eliminan variables dependiendo de la significación de su capacidad discriminatoria. Permite además la posibilidad de "arrepentirse" de decisiones tomadas en pasos anteriores, bien sea eliminando del conjunto seleccionado una variable introducida en un paso anterior del algoritmo, bien sea seleccionando una variable previamente eliminada.

Quizás, uno de los criterios más usado para la selección de variables sea el de la λ de Wilks, según se comentó en el apartado 3. La significación de la capacidad discriminante se mide con el estadístico F obtenido a partir de esta λ. Una explicación más detallada se puede encontrar en Salvador (2000). En la tabla 3 se muestran los resultados de ejecutar nuestro algoritmo de Búsqueda Tabú para valores de *p* desde 2 a 13. Así mismo también se han ejecutado los métodos *Backward*, *Forward* y *Stepwise*. También en la tabla 3 se muestran los resultados de las soluciones intermedias, correspondientes a esos mismos valores de *p*, obtenidos por la ejecución de los métodos *Backward* y *Forward*. El método *Stepwise*, en este caso, ha coincidido con los 6 primeros pasos del método *Forward*, es decir no ha habido eliminación de variables.

			Búsqueda Tabú Completo			
P	***Forward***	***Backward***	**Constructivo**	**Blocal**	**Básico**	**Intensific +Diversif.**
2	0,68181818	0,67676768	0,78787879	0,78787879	0,78787879	0,78787879
3	0,71212121	0,69191919	0,79292929	0,79292929	0,7979798	0,7979798
4	0,72222222	0,69191919	0,8030303	0,8030303	0,8030303	0,8030303
5	0,74242424	0,70707071	0,8030303	0,8030303	0,81313131	0,83333333
6	0,75757576*	0,71717172	0,8030303	0,8030303	0,81313131	0,84343434
7	0,77272727	0,72222222	0,8030303	0,8030303	0,84848485	0,84848485
8	0,76262626	0,71717172	0,8030303	0,8030303	0,82323232	0,85353535
9	0,77777778	0,72222222	0,8030303	0,8030303	0,82323232	0,85858586
10	0,78282828	0,72727273	0,8030303	0,8030303	0,84848485	0,85858586
11	0,79292929	0,72727273	0,8030303	0,8030303	0,86868687	0,86868687
12	0,76767677	0,73232323	0,7979798	0,82828283	0,85858586	0,85858586

			Búsqueda Tabú Completo			
P	***Forward***	***Backward***	**Constructivo**	**Blocal**	**Básico**	**Intensific +Diversif.**
13	0,7979798	0,72222222	0,7979798	0,84848485	0,87878788	0,87878788

- solución final obtenida por el método *Stepwise*

Tabla 3. Comparación de búsqueda tabú con los métodos tradicionales *Forward* y *Backward*.

De la tabla 3 se pueden obtener las siguientes conclusiones:

- El método *Backward* parece funcionar claramente peor que el método *Forward* y *Stepwise*, al menos para este problema

- Nuestro método mejora significativamente las soluciones de los métodos tradicionales para cualquier valor de *p*. Ya no solo el algoritmo completo, sino cada una de las componentes. Obsérvese como con el método *Forward* apenas se consigue llegar al 79% de aciertos con 13 variables, mientras que con nuestro tabú se supera el 80% con 4 variables.

Considerando el doble objetivo de minimizar el número de variables y maximizar el porcentaje de aciertos es claro que nuestras soluciones dominan a las de estos métodos tradicionales. Así se aprecia en el gráfico 1.

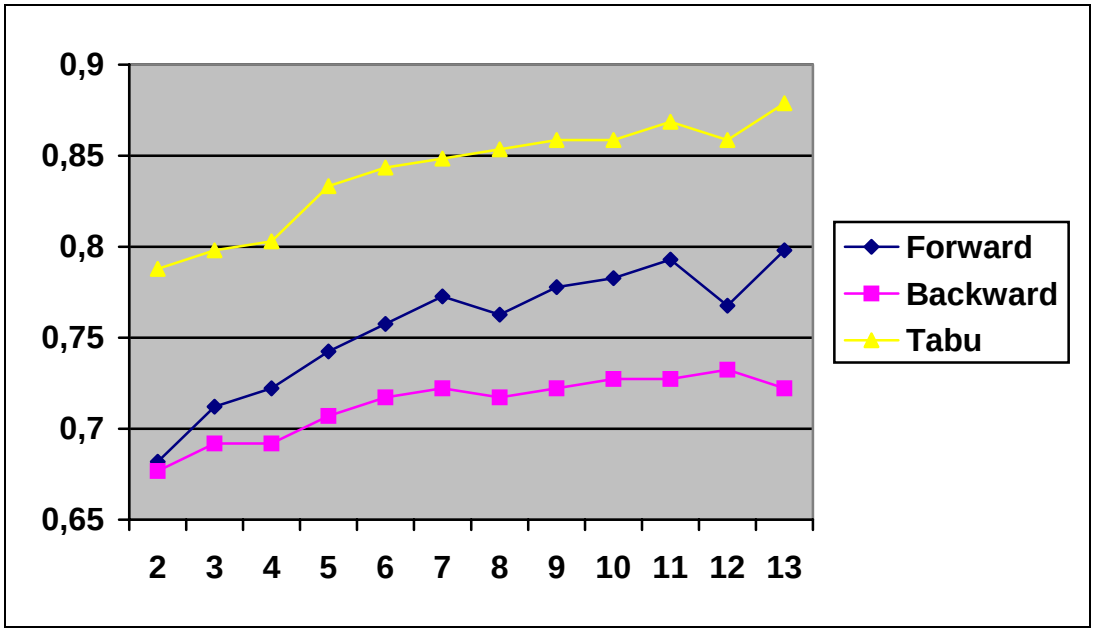

Gráfico 1: Comparación de Búsqueda Tabú con métodos tradicionales *Forward* y *Backward*.

7. Conclusiones

En este trabajo se propone un método de selección de variables para su uso posterior en el análisis discriminante de dos grupos. Concretamente se trata de un algoritmo basado en la estrategia metaheurística Búsqueda Tabu que consta de un método constructivo (para generar la solución inicial) y un procedimiento básico de búsqueda tabú que se complementa con Intensificación y Diversificación. El objetivo principal es reducir el número de variables manteniendo un porcentaje de aciertos elevado en el análisis discriminante. Para estudiar la eficacia del método diseñado y sus componentes se han hecho una serie de pruebas con diferentes problemas tests. De los resultados obtenidos se puede concluir que el método propuesto en este trabajo consigue significativamente mejores resultados que los métodos que sen han usado tradicionalmente como *Stepwise*, *Backward* y *Forward*.

Bibliografía

1. Bala J., Dejong K., Huang J., Vafaie H. y Wechsler H. (1996): *Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts*, Evolutionary Computation, 4, 3, 297-311.

2. Cotta C., Sloper C. y Moscato P. (2004): *Evolutionary Search of Thresholds for Robust Feature Set Selection: Application to the Analysis of Microarray Data*, Lecture Notes In Computer Science 3005: 21-30.

3. Ganster H., Pinz A., Rohrer R., Wildling E., Binder M. y Kittler H.(2001): *Automated Melanoma Recognition*, IEEE Transactions On Medical Imaging 20 (3): 233-239.

4. García, F.C., García-Torres, M., Moreno Pérez, J.M. and Moreno-Vega, J.M. (2003): *Búsqueda Dispersa para el Problema de la Selección de Variables, CAEPIA-2003*

5. Glover F. (1986): Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research, Vol. 13, pp. 533-549.

6. Glover F. (1989): *Tabu Search: Part I*, ORSA Journal on Computing. Vol. 1, pp. 190-206.

7. Glover F. (1990): *Tabu Search: Part II*, ORSA Journal on Computing. Vol. 2, pp. 4-32.

8. Glover F. y Laguna M. (1993): *Tabu Search in Modern Heuristic Techniques for Combinatorial Problems*, C. Reeves ed., Blackwell Scientific Publishing, pp. 70-141.

9. Glover F. Y Laguna M. (1997): *Tabu Search*, Kluwer Academic Publishers, Boston.

10. Glover, F. y Laguna, M. (2002): *Tabu Search*, in Handbook of Applied Optimization, P. M. Pardalos and M. G. C. Resende (Eds.), Oxford University Press, pp. 194-208.

11. Glover F., Laguna M. y Marti R. (2000): *Fundamentals of Scatter Search and Path Relinking*, Control and Cybernetics, vol. 29, pp.653-684.

12. Gómez O., Casado S., Nuñez L., y Pacheco J. (2004): *Resolución del Problema de Selección de Variables Cuantitativas mediante GRASP. Aplicación a Ratios Financieros*, Actas XII Congreso Asepuma celebrado en Septiembre de 2004 en la Universidad de Murcia.

13. Hair, J., Anderson, R., Tatham, R. y Black, W. (1999): *Análisis Multivariante.* 5ª Edición. Prentice Hall.

14. Huberty, C.J.(1994): *Applied Discriminant Analysis*. Wiley. Interscience.

15. Inza I., Larrañaga P., Etxeberria R. y Sierra B. (2000): *Feature Subset Selection by Bayesian networks based optimization*, Artificial Intelligence, 123, 157-184.

16. Inza I., Merino M., Larranaga P., Quiroga J., Sierra B. y Girala M.(2001a): *Feature Subset Selection by Genetic Algorithms and Estimation of Distribution Algorithms - A Case Study in the Survival of Cirrhotic Patients Treated with TIPS*, Artificial Intelligence In Medicine 23 (2): 187-205.

17. Inza I., Larranaga P. y Sierra B. (2001b): *Feature Subset Selection by Bayesian Networks: A Comparison with Genetic and Sequential Algorithms*, International Journal of Approximate Reasoning 27 (2): 143-164.

18. Jain A. And Zongker D. (1997): Feature *Selection: Evaluation, Application, and Small Sample Performance*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 153-158.

19. Jourdan L., Dhaenens C. y Talbi E. (2001): *A Genetic Algorithm for Feature Subset Selection in Data-Mining for Genetics*, MIC 2001 Proceedings, 4th Metaheuristics Internationl Conference, 29-34.

20. Kohavi R. (1995): *Wrappers for Performance Enhancement and Oblivious Decision Graphs*, Stanford University, Computer Science Department.

21. Lee S., Yang J. y Oh K.W. (2003): *Prediction of Molecular Bioactivity for Drug Design Using a Decision Tree Algorithm*, Lecture Notes In Artificial Intelligence 2843: 344-351.

22. Lewis P.M. (1962): *The Characteristic Selection Problem in Recognition Systems*, IEEE Trans. Information Theory, vol. 8: 171-178.

23. Liu H. Y Motoda H.(1998): *Feature Selection for Knowledge Discovery and Data Mining*, Boston: Kluwer Academic.

24. Narendra P.M. y Fukunaga K. (1977): *A Branch and Bound Algorithm for Feature Subset Selection*, IEEE Trans. Computers, vol. 26, no. 9: 917-922.

25. O'Gorman T.W. (2004): *Using adaptive Methods to Select Variables in Case-Control Studies*, Biometrical Journal 46,5, pp.595-605.

26. Oliveira L.S., Sabourin R., Bortolozzi F., y otros (2003): *A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition*, International Journal Of Pattern Recognition And Artificial Intelligence 17 (6): 903-929.

27. Salvador Figueras M. (2000): *Análisis Discriminante,*[en línea] 5campus.com, Estadísticahttp://5campus.com/lección/discri[10 de febrero de 2005]

28. Sebestyen, G. (1962): *Decision-Making Processes* in Pattern Recognition. New York: MacMillan.

29. Shy S. y Suganthan P.N. (2003): *Feature Analysis and Clasification of Protein Secondary Structure Data*, Lecture Notes in Computer Science 2714: 1151-1158.

30. Sierra B., Lazkano E., Inza I., Merino M., Larrañaga P. y Quiroga J. (2001): *Prototype Selection and Feature Subset Selection by Estimation of Distribution Algorithms. A Case Study in the Survival of Cirrhotic Patients Treated with TIPS*. Lecture Notes in Artificial Intelligence 2101:20-29.

31. Tamoto E., Tada M., Murakawa K., Takada M., Shindo G., Teramoto K., Matsunaga A., Komuro K., Kanai M., Kawakami A., Fujiwara Y., Kobayashi N., Shirata K., Nishimura N., Okushiba S.I., Kondo S., Hamada J., Yoshiki T., Moriuchi T. y Katoh H.(2004): *Gene expression Profile Changes Correlated with Tumor Progression and Lymph Node Metastasis in Esophageal Cancer*. Clinical Cancer Research 10(11):3629-3638.

32. Wong M.L.D. y Nandi A.K. (2004): *Automatic Digital Modulation Recognition Using Artificial Neural Network and Genetic Algorithm*. Signal Processing 84 (2): 351-365.