Universidad Autónoma de Madrid
Escuela Politécnica Superior
Departamento de Ingeniería Informática

Selección de variables mediante programación cuadrática

Quadratic Programming Feature Selection (QPFS)

Trabajo de Fin de Máster presentado para la obtención del título
Máster en Ingenieria Informática y de Telecomunicaciones
(*major* en Inteligencia Computacional)

Autor
Irene Rodríguez Luján

Director
Carlos Santa Cruz Fernández

Madrid, Febrero 2009

Resumen

El paradigma de la selección de variables dentro del campo del reconocimiento de patrones ha adquirido especial importancia en los últimos años debido a la aparición de problemas de clasificación de alta dimensionalidad. Existen diferentes aproximaciones en la literatura para abordar este problema, siendo uno de los métodos más recientes y de mejores resultados el denominado criterio de mínima-redundancia-máxima-relevancia (mRMR), basado en el cálculo de la información mutua intervariable y variable-clase. Una de las principales limitaciones de mRMR es su escalabilidad para problemas de altas dimensiones debido, principalmente, a su dependencia cuadrática en la dimensión del problema. Es por ello que se propone un nuevo algoritmo de selección de variables, Quadratic Programming Feature Selection (QPFS), inspirado en la optimización media-varianza de carteras de inversión y que utiliza el método de Nyström de diagonalización de matrices para reducir la complejidad temporal de mRMR ofreciendo resultados similares en términos de acierto en clasificación.

Agradecimientos

En primer lugar, quiero agradecer a mi tutor, Carlos Santa Cruz Fernández, su interés, apoyo, orientación y confianza durante estos meses. Gracias por dar rápida respuesta a mis dudas, por buscar huecos siempre que lo necesitaba, por hacerme sentir agusto desde el primer momento y haberme dado la posibilidad de pasar unos meses en la Universidad de San Diego California, una experiencia totalmente enriquecedora tanto profesional como personalmente. Sin duda, este trabajo hubiera sido imposible sin él.

Agradecer también a Ramón Huerta su acogida en la Universidad de San Diego, su entusiasmo y positivismo, sus grandes ideas y las chocolatinas a media tarde. Gracias al Profesor Charles Elkan por su tiempo y disponibilidad, haciendo de cada reunión una clase magistral. Y gracias a Eduardo Serrano por ser un grandísimo apoyo en la "aventura" americana.

Gracias a mi familia: mis padres y mi hermano Luis, que no me han fallado nunca y siempre me han dado ánimos para seguir adelante, sobre todo en los momentos más difíciles. Gracias a Mary y Adel, inolvidables. Gracias a mis amigos: Javi, Natalia y Lauri (mi apoyo incondicional en los últimos meses), Noe, Garcés y Sku (a mi lado desde distintas partes del mundo) y a mis amigos de la uni, especialmente Mariaje, Ángela, María, Elena, Edu, Jorge y Juanda. Gracias también a Carlos por su humor "mágico" y dosis de optimismo. Gracias a mis compañeros del Instituto de Ingeniería del Conocimiento, y sobre todo a Álvaro, por su paciencia y ayuda infinita.

Mi agradecimiento a todos aquellos profesores que han sabido transmitirme su entusiasmo e ilusión durante estos años, entre ellos Antonio, Mari Nieves, Eugenio y Jose.

Y por último, agraceder el apoyo económico de la beca de Formación de Personal Universitario de la Universidad Autónoma de Madrid y de la Cátedra de análisis de patrones de comportamiento del Instituto de Ingeniería del Conocimiento, UAM.

Índice general

Capítulo 1

Introducción

En los últimos años, han aparecido numerosas aplicaciones en el campo del reconocimiento de patrones donde la dimensión del problema de clasificación es elevada. Entre estos campos se encuentra la clasificación de diferentes patologías en función de microarrays de expresión genética en los que, por lo general, el número de variables es muy alto en comparación al número de patrones de que se dispone [11, 31, 52]. La clasificación de documentos es otro caso donde, tanto el número de patrones como la dimensión del problema son elevados siendo, además, los datos por lo general muy *sparse* [16]. Otras aplicaciones pueden ser la predicción del retorno de un activo del stock market en función de un gran número de características del mismo o la detección de fraude en tarjetas de crédito.

En muchos de estos casos, la selección de un subconjunto de variables se convierte en tarea fundamental para evitar el sobreajuste de los datos de entrenamiento [13] y determinar el conjunto de variables más significativas para la clasificación, pudiendo dar así una mejor interpretación a los resultados.

El objetivo de todo problema de selección de variables es encontrar un subconjunto de ellas que mejor *expliquen* la clase a la que pertenece cada patrón. No es objetivo de este trabajo determinar el tamaño óptimo de tal subconjunto, sino dar una *buena* ordenación de las variables del problema. Desde el punto de vista del diseño del clasificador, un algoritmo de selección de variables forma parte del proceso de clasificación, donde a la selección de variables le sigue la aplicación de un clasificador estándar sobre el subconjunto escogido.

Múltiples métodos se han desarrollado para tratar de resolver el paradigma de la selección de variables. Estas técnicas se pueden dividir en tres grandes grupos en función del papel que juega el clasificador en el proceso de elección [21]. Como sistemas independientes del clasificador se encuentran los métodos *de filtro*, las técnicas *wrapper* realizan una búsqueda guiada de posibles subconjuntos y utilizan un clasificador para

evaluar la *calidad* de cada uno de ellos y, en último lugar, los métodos *embebidos* realizan la selección de variables dentro del propio proceso de entrenamiento del clasificador, incluyéndola en la función objetivo a optimizar. Estas técnicas se explican con más detalle en el capítulo 2.

Los métodos de filtro parecen los más adecuados para problemas de alta dimensionalidad [21, 50] por su sencillez y mayor rapidez, pero uno de sus puntos débiles es que no tienen en cuenta la interdependencia entre variables, aspecto importante para una buena selección tal y como se ha reconocido en diversos estudios [8, 9, 26]: la combinación de variables buenas individualmente no implica necesariamente un buen error en clasificación. Para tratar de solventar esta carencia se han propuesto algoritmos como *Maximal Dependency (MaxDep)* o *minimal-redundancy-maximum-relevance (mRMR)* basados en criterios de información mutua [37]. MaxDep es muy costoso computacionalmente y requiere una gran cantidad de datos de entrenamiento para estimar funciones de densidad conjuntas (sección 2.1.1). mRMR es una aproximación iterativa a MaxDep de coste computacional lineal en el número de patrones y cuadrático en la dimensión (sección 3.3.1), convirtiéndose en prohibitivo para problemas de gran número de patrones y, especialmente, de alta dimensionalidad.

Como alternativa al algoritmo mRMR, se proprone el método de selección de variables *Quadratic Programming Feature Selection (QPFS)* en el que las variables se seleccionan en función a su dependencia con la clase y su *independencia* con otras variables, basándose en la idea de optimización media-varianza de carteras de activos financieros [33] (sección 2.2.1). A diferencia de mRMR, QPFS es un algoritmo *de un paso* que tiene en cuenta todas las variables para proceder a la optimización de una función cuadrática (sección 2.2 y capítulo 3) y es capaz de reducir considerablemente el coste temporal de mRMR trasladando el problema de optimización a un subespacio de menor dimensión (sección 3.2) empleando para ello el método de Nyström de diagonalización de matrices [17] (secciones 2.3 y 3.2.2). El análsis teórico y los resultados obtenidos al comparar los algoritmos mRMR y QPFS en distintos conjuntos de datos (capítulo 4), permiten concluir que la complejidad temporal del método propuesto es inferior a la de mRMR, ofreciendo a su vez, resultados similares en términos de acierto en clasificación.

El trabajo se estructura como sigue: en el siguiente capítulo se abordan los conceptos fundamentales para comprender el método de selección de variables QPFS desarrollado en el capítulo 3. El capítulo 4 describe la implementación, metodología, conjuntos de datos de prueba, resultados y conclusiones de los experimentos llevados a cabo para comparar mRMR frente a QPFS. Finalmente, el capítulo 5 expone las principales líneas de investigación que quedan abiertas.

Capítulo 2

Conceptos fundamentales

A lo largo de este capítulo se exponen los conceptos básicos para comprender la motivación y fundamentos que han guiado el método QPFS propuesto en el capítulo 3. Se explica en qué consiste el problema de selección de variables y los distintos planteamientos existentes para abordarlo, el concepto de programación cuadrática y su aplicación a la optimización de carteras de activos financieros y, el método de Nyström para resolver problemas de diagonalización en altas dimensiones submuestreando la matriz original.

2.1. Métodos de selección de variables

El problema de selección de variables ha adquirido especial interés en diversas áreas de aplicación donde el número de variables para un problema de clasificación es superior al millar y muchas de ellas son redundantes o irrelevantes; entre estas áreas se encuentran la clasificación de documentos de la red o el análisis de arrays de expresión genética. El paradigma de la selección de variables trata de cubrir tres aspectos: mejorar el error de predicción de los clasificadores, proporcionar clasificadores más rápidos y eficientes y, facilitar la comprensión del proceso subyacente generador de los datos.

Es posible clasificar las distintas técnicas de selección de variables en tres grandes grupos:

- **Métodos de filtro**: La selección de variables forma parte del preprocesado de los datos y es independiente del predictor [4, 15, 16]. Suele ser preferible a otros métodos de selección de variables debido a su escalabilidad. Computacionalmente es eficiente porque sólo requiere el cómputo y ordenación de la relevancia de cada variable respecto a la clase; estadísticamente es robusto frente a problemas de sobreajuste [24].
- **Métodos *wrapper***: Se utiliza un clasificador como caja negra para asignar *puntuaciones* a subconjuntos de variables de acuerdo a su poder de clasificación

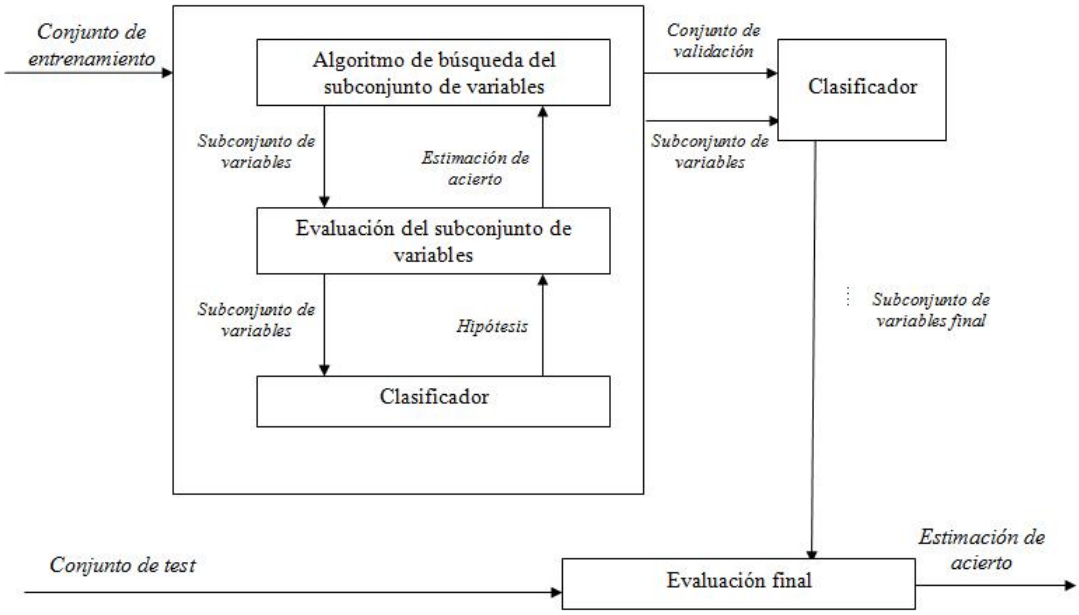

Figura 2.1: Esquema general de un algoritmo *wrapper* de selección de variables. El clasificador actúa como *caja negra* en el proceso de selección.

[27, 29, 30]. Es necesario definir cómo realizar la búsqueda en el espacio de todos los posibles subconjuntos de variables, cómo evaluar el error de predicción para guiar la búsqueda y, finalmente, qué clasificador usar. La figura 2.1 muestra un esquema general de un algoritmo *wrapper* de selección de variables. Este tipo de técnicas son habitualmente criticadas por parecer métodos de fuerza bruta que suponen una alta carga computacional, pero existen estrategias que reducen la complejidad sin sacrificar necesariamente la eficiencia predictiva, entre ellas, *best-first* [18, 42, 49], *simulated annealing* [29] o algoritmos genéticos [3, 45, 46]. Estas búsquedas se pueden utilizar para realizar una selección de variables *forward* (las variables se van añadiendo progresivamente al subconjunto) o *backward* (se parte del conjunto total de variables y se van eliminando aquellas que se consideran menos representativas). Se suelen utilizar como clasificadores naïve Bayes o SVMs.

- **Métodos embebidos**: La selección de variables se realiza en el proceso de entrenamiento del clasificador a partir del valor de la función objetivo a optimizar para distintos subconjuntos de datos. Presentan algunas ventajas como un mejor aprovechamiento del conjunto de datos (no es necesario subconjunto de validación) y mayor rapidez que los métodos *wrapper*, dado que no precisan la evaluación de un clasificador para cada uno de los subconjuntos candidatos. Los árboles de de-

cisión [5] o la selección de variables en SVMs [47] son un ejemplo de este tipo de selectores.

Los métodos de filtro parecen los más adecuados para problemas de alta dimensionalidad, fundamentalmente por su mayor rapidez, su independencia respecto al clasificador subyacente, y su aplicación previa a la fase de entrenamiento del clasificador que permite reducir la dimensión del problema y prevenir el sobreajuste [21, 50].

Dentro de los métodos de filtro se pueden utilizar distintas medidas para determinar el poder discriminativo de cada una de las variables. La mayoría de ellas son de naturaleza estadística: la correlación de Pearson [22], el T-test [24] o la información mutua [10, 37], entre otras, pero también existen medidas *empíricas* como el error de predicción o el área de la curva ROC que se obtienen al tratar de clasificar el problema utilizando únicamente cada una de las variables [15].

La gran limitación de estos algoritmos es que no tienen en cuenta la dependencia entre variables y, por tanto, están perdiendo de vista el objetivo de determinar cuál es la mejor combinación de ellas [6, 21]. Con fin de cubrir estas deficiencias se idearon algoritmos como *Max-Dependency (MaxDep)* y *minimal-redundancy-maximal-relevance (mRMR)* [37] en los que, no sólo se considera la *similitud* de la variable con la clase sino también la dependencia intervariable. Estos algoritmos, estudiados con detalle en la sección 2.1.1, son el punto de referencia del método propuesto en este trabajo.

2.1.1. Criterios basados en información mutua

En este capítulo se describirán los algoritmos de selección de variables *Max-Dependency (MaxDep)* y su aproximación *minimal-redundancy-maximum-relevance (mRMR)* [11, 37, 52], fundamentados en el criterio de máxima dependencia estadística basado en información mutua.

En lo que sigue, supondremos un problema de clasificación en C clases formado por N patrones. Cada patrón se representa por el vector formado por el conjunto de M variables y un valor adicional que indica la clase a la que pertenece ésto es, $x = \{v_i, c\}$ con $i = 1, \ldots, M$ y $c = 0, \ldots, C-1$. El objetivo de la selección de variables consiste en encontrar un subespacio de m variables ($m < M$) que permita clasificar correctamente el problema.

La información mutua es la medida más general de independencia estadística entre dos variables aleatorias v_i y v_j, cuantifica la información que v_i y v_j comparten es decir, cómo afecta el conocimiento de una de ellas en la incertidumbre sobre la otra. Formalmente, se define la información mutua entre dos variables aleatorias v_i y v_j a

partir de las funciones de densidad marginales $p(v_i)$ y $p(v_j)$, y la función de densidad conjunta $p(v_i, v_j)$ como,

$$I(v_i; v_j) = \int \int p(v_i, v_j) \log \frac{p(v_i, v_j)}{p(v_i)p(v_j)} dv_i dv_j \tag{2.1}$$

Esta definición verifica,

$$I(v_i; v_j) \geq 0 \tag{2.2}$$

$$I(v_i; v_j) = I(v_j; v_i) \tag{2.3}$$

Si v_i y v_j son independientes, el conocimiento de v_i no da ninguna información acerca de v_j y, por tanto, su información mutua es cero (la implicación contraria también es cierta, es decir, la información mutua es nula únicamente en el caso de variables independientes [32]). En el otro extremo, si v_i y v_j son idénticamente distribuidas, el conocimiento de v_i implica un total conocimiento sobre v_j y viceversa.

La principal dificultad para el cálculo de (2.1) radica en estimar las densidades $p(v_i, v_j)$,$p(v_i)$ y $p(v_j)$ a partir de los datos. Para variables categóricas, la ecuación (2.1) se reduce a un sumatorio y el cálculo de la información mutua es inmediato realizando un simple conteo.

$$I(v_i; v_j) = \sum_{w_i} \sum_{w_j} P(v_i = w_i, v_j = w_j) \log \frac{P(v_i = w_i, v_j = w_j)}{P(v_i = w_i)P(v_j = w_j)} \tag{2.4}$$

En cambio, la evaluación de la integral (2.1) para variables continuas es mucho más costosa. Una posible solución es incorporar un paso de discretización en el preprocesado de los datos pero, en algunos casos, determinar cuál es la discretización adecuada no es tarea fácil puediéndo recurrir entonces a otros métodos de estimación de funciones de densidad como máxima verosimilitud, estimación bayesiana o técnicas no paramétricas [13].

Así pues, el objetivo de un método de selección de variables basado en información mutua es encontrar un subconjunto S de m variables $S = \{v_1, \ldots, v_m\}$ que conjuntamente tenga la máxima dependencia con la variable objetivo c,

$$\text{máx}\, D(S, c) \ , \ D = I(\{v_i, i = 1, \ldots, m\}\, ; c) \tag{2.5}$$

A pesar de que MaxDep es un criterio óptimo desde un punto de vista teórico, la estimación de $p(v_i, \ldots, v_m)$ y $p(v_1, \ldots, v_m, c)$ es prohibitiva por dos motivos: el número

de ejemplos de que se dispone suele ser insuficiente y es computacionalmente costosa. Únicamente es posible utilizar MaxDep cuando se desea seleccionar un número pequeño de variables y el número de patrones N es grande; de esta manera, la insuficiencia de datos para la estimación de las funciones de densidad quedaría cubierta y su cálculo se realizaría en espacios de muy pocas dimensiones donde el coste computacional es asumible.

Como alternativa a MaxDep se puede seleccionar el conjunto de variables basándose en la dependencia con la clase (*relevancia*). El criterio de *Máxima Relevancia* consiste en encontrar el subconjunto de variables S que maximicen $D(S, c)$, donde D se ha aproximado como el valor medio de la información mutua entre cada una de las variables que componen el subconjunto S y la clase, esto es

$$D = \frac{1}{|S|} \sum_{v_i \in S} I(v_i; c) \tag{2.6}$$

Por tanto, el subconjunto *óptimo* seleccionado por el algoritmo serán las m variables con mayor información mutua con la clase c. Este criterio se enmarca dentro de los métodos de filtro descritos anteriormente.

Criterio de mínima Redundancia y Máxima Relevancia (mRMR)

Es posible que las variables seleccionadas por MaxRel sean bastante redundantes, debido a una gran dependencia entre ellas. Si la dependencia entre dos variables es alta, su poder discriminativo no se ve muy afectado al eliminar una de ellas. Por tanto, no sólo es deseable maximizar la relevancia de las variables con la clase, sino también minimizar la redundancia entre ellas. Con tal fin, el criterio de *minima redundancia* se define como

$$\text{mín}\, R(S)\ ,\ R = \frac{1}{|S|^2} \sum_{v_i \in S} I(v_i; v_j) \tag{2.7}$$

El algoritmo *minimal-redundancy-maximal-relevance (mRMR)* combina (2.6) y (2.7) de acuerdo a la expresión,

$$\text{máx}\, \Phi(D, R)\ ,\ \Phi = D - R \tag{2.8}$$

Los autores del algoritmo [37] sugieren un método iterativo para encontrar un conjunto de variables cercano al óptimo ya que, en la práctica, es inviable tratar de evaluar la expresión (2.8) para todos los posibles subconjuntos de m variables en el espacio original. Supuesto encontrado el conjunto subóptimo S_{m-1} de $m-1$ variables, la siguiente variable a añadir será áquella que verifique

$$\max_{v_j \in X \setminus S_{m-1}} \left\{ I(v_j; c) - \frac{1}{m-1} \sum_{v_i \in S_{m-1}} I(v_j; v_i) \right\} \tag{2.9}$$

En la primera iteración del algoritmo el término de redundancia es nulo, entonces la primera variable seleccionada por mRMR es la misma que la escogida por MaxRel: áquella que tenga la máxima relevancia con la clase.

En la siguiente sección se desarrolla el concepto de programación cuadrática como problema de optimización y su aplicación a la selección de portfolios de activos financieros. La idea en la que se basa la selección de la cartera óptima es una de la principal motivación del método QPFS desarrollado en el capítulo 3.

2.2. Programación cuadrática

Se denomina ***programa cuadrático (QP)*** a todo problema de optimización de una función cuadrática en un espacio multidimensional ($x \in \mathbb{R}^M$) sujeto a restricciones lineales sobre las variables. Más concretamente,

$$\min f(x) \ , \ f(x) = \frac{1}{2} x^T Q x - F^T x \tag{2.10}$$

Sujeto a inecuaciones de la forma:

$$Gx \leq b \tag{2.11}$$

Y restricciones de igualdad:

$$Hx = d \tag{2.12}$$

Donde F es un vector M-dimensional que representa los coeficientes del término lineal, y Q es una matriz simétrica en $\mathbb{R}^{M \times M}$ asociada a los coeficientes de los términos cuadráticos. Respecto a las restricciones, $G \in \mathbb{R}^{q \times M}$, $b \in \mathbb{R}^q$, $H \in \mathbb{R}^{r \times M}$ y $d \in \mathbb{R}^r$.

Existen diversos métodos para resolver el problema QP [20]. No es objeto de estudio de este trabajo profundizar en ellos y se ha optado por el algoritmo de Goldfarb e Idnani [19] utilizado en el paquete *quadprog* de R [44].

2.2.1. Optimización media-varianza para selección de carteras

Una de las aplicaciones de la programación cuadrática es la selección óptima de carteras de activos financieros [38]. El problema de optimización media-varianza fue propuesto originalmente por Markowitz [33] y supone un mercado de M activos, cada uno de los cuales tiene un retorno R_i definido en este modelo como el promedio de porcentaje de crecimiento anual. Estos retornos tienen una distribución conjunta de media $\mu \in \mathbb{R}^M$ y matriz de covarianzas $Q \in \mathbb{R}^{M \times M}$.

Un ***portfolio*** $x \in \mathbb{R}^M$ es un vector de inversiones en los activos cuyas componentes suman la unidad y cuyo retorno esperado y varianza (comúnmente denominada ***riesgo***) vienen dados por $E = \mu^T x$ y $V = x^T Q x$ respectivamente. Es decir,

$$E = \sum_{i=1}^{M} x_i \mu_i \tag{2.13}$$

$$V = \sum_{i=1}^{M} \sum_{j=1}^{M} \sigma_{ij} x_i x_j \tag{2.14}$$

Siendo μ_i el valor esperado del retorno del i-ésimo activo y σ_{ij}, la covarianza entre los retornos del i-ésimo y j-ésimo activo. Así pues, E representa el retorno esperado del portfolio y V su riesgo o volatilidad: los activos presentes en el mercado no se comportan de igual modo, los activos pueden tener tendencias similares, opuestas o ningún tipo de relación entre ellos. Esta relación entre el movimiento en el mercado de los retornos de dos activos R_i y R_j puede representarse mediante su covarianza σ_{ij} (o correlación).

Por tanto, asumiendo que los inversores tratan de obtener un portfolio *eficiente* (entendido como áquel tal que no existe otra cartera con mayor retorno y menor o igual riesgo o, de menor riesgo con igual o mayor retorno esperado), la inversión *óptima* x_i para el activo i-ésimo vendrá dada por la solución del problema de optimización cuadrática,

$$\min_x \left\{ \frac{1}{2} x^T Q x - \mu^T x \right\} \tag{2.15}$$

Sujeto a las restricciones:

$$x_i \geq 0 \ , \ i = 1, \ldots, M \tag{2.16}$$

$$\sum_{i=1}^{M} x_i = 1 \tag{2.17}$$

2.3. El método de Nyström

El método de Nyström para matrices [17], permite aproximar los autovectores de una matriz de Gram $M \times M$, con M grande, resolviendo un problema de diagonalización de una matriz $k \times k$ con $k \ll M$ y utilizando la denominada *extensión de Nyström*.

El método de Nyström surgió originalmente como técnica para aproximar numéricamente las funciones propias de un problema de diagonalización de la forma,

$$\int_a^b K(x,y)\phi(y)dy = \lambda\phi(x) \tag{2.18}$$

Realizando una partición del intervalo $[a,b]$ en n puntos equidistantes $\xi_1, \xi_2, \ldots, \xi_n$ y aproximando la integral anterior por cuadratura simple [39] se tiene,

$$\frac{(b-a)}{n}\sum_{j=1}^{n} K(x,\xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(x) \tag{2.19}$$

Donde $\hat{\phi}(x)$ es una aproximación de $\phi(x)$. Igualando $x = \xi_i$ en (2.19),

$$\frac{(b-a)}{n}\sum_{j=1}^{n} K(\xi_i,\xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(\xi_i) \text{ , para } i = 1, \ldots, n \tag{2.20}$$

Sin pérdida de generalidad, sea $a = 0$ y $b = 1$ entonces, el sistema de ecuaciones descrito en (2.20) se puede expresar como,

$$K\hat{\Phi} = n\hat{\Phi}\Lambda \tag{2.21}$$

Donde $K_{ij} = K(\xi_i, \xi_j)$ es la matriz de Gram, Λ una matriz diagonal con los autovalores aproximados $\lambda_1, \lambda_2, \ldots, \lambda_n$ y $\Phi = [\phi_1, \phi_2, \ldots, \phi_n]$ los autovectores asociados. Sustituyendo la expresión anterior en la ecuación (2.19) se obtiene la denominada ***extensión de Nyström*** para cada $\hat{\phi}_i$:

$$\hat{\phi}_i(x) = \frac{1}{n\lambda_i}\sum_{j=1}^{n} K(x,\xi_j)\hat{\phi}_i(\xi_j) \tag{2.22}$$

A partir del resultado anterior es posible definir un método de Nyström para matrices que permite aproximar los autovectores de una matriz de Gram resolviendo un problema

de diagonalización mucho menor. Sea la matriz $Q \in \mathbb{R}^{M\times M}$ simétrica y semidefinida positiva expresada como,

$$Q = \begin{pmatrix} A & B \\ B^T & E \end{pmatrix} \tag{2.23}$$

Donde $A \in \mathbb{R}^{k\times k}$, $B \in \mathbb{R}^{k\times m}$, y $E \in \mathbb{R}^{m\times m}$ siendo $k \ll m$ y $k+m=M$. Dado que Q es simétrica, puede escribirse como $Q = Z^T Z$. Si el rango de Q es k y las filas de la submatriz $[A \quad B]$ son linealmente independientes, Z se puede definir en función de A y B.

Sea $Z = [X \quad Y]$ con $X \in \mathbb{R}^{M\times k}$ y $Y \in \mathbb{R}^{M\times m}$. Reescribiendo Q,

$$Q = Z^T Z = \begin{pmatrix} X^T X & X^T Y \\ Y^T X & Y^T Y \end{pmatrix} \tag{2.24}$$

Igualando bloque a bloque en (2.23) y (2.24) se tiene $A = X^T X$ y $B = X^T Y$. Diagonalizando $A = U\Lambda U^T$, con $U^T U = I_M$ se tiene

$$\bar{X} = \Lambda^{\frac{1}{2}} U^T \tag{2.25}$$

$$\hat{Y} = (\bar{X}^T)^{-1} B = \Lambda^{-\frac{1}{2}} U^T B \tag{2.26}$$

Con $\bar{X} \in \mathbb{R}^{k\times k}$ y $\hat{Y} \in \mathbb{R}^{k\times m}$. Entonces $\hat{Z} = [\bar{X} \quad \hat{Y}] \in \mathbb{R}^{k\times M}$ y, por tanto,

$$\hat{Q} = \hat{Z}^T \hat{Z} = \begin{pmatrix} \bar{X}^T \bar{X} & \bar{X}^T \hat{Y} \\ \hat{Y}^T \bar{X} & \hat{Y}^T \hat{Y} \end{pmatrix} \tag{2.27}$$

$$= \begin{pmatrix} \bar{X}^T \bar{X} & \bar{X}^T \Lambda^{-\frac{1}{2}} U^T B \\ (\Lambda^{-\frac{1}{2}} U^T B)^T \bar{X} & (\Lambda^{-\frac{1}{2}} U^T B)^T \Lambda^{-\frac{1}{2}} U^T B \end{pmatrix} \tag{2.28}$$

$$= \begin{pmatrix} A & B \\ B^T & B^T A^{-1} B \end{pmatrix} \tag{2.29}$$

Si el rango de Q es mayor que k o las filas de la submatriz $[A \quad B]$ no son linealmente independientes, entonces $\hat{Q}$ es una aproximación de Q cuantificada como $\|E - B^T A^{-1} B\|$.

Dada la aproximación anterior de $\hat{Q}$, es posible calcular sus autovectores $\hat{Q} = \bar{U}\Lambda\bar{U}^T$ conocida la diagonalización $A = U\Lambda U^T$ y a partir de la extensión de Nyström (2.19) adaptada a matrices: la coordenada j-ésima del autovector i-ésimo de $\hat{Q}$ vendrá dada por,

$$\bar{U}_{ji} = \frac{1}{\lambda_i} \sum_{n=1}^{k} Q_{nk} U_{ni} \tag{2.30}$$

Siendo $\lambda_1, \lambda_2, \ldots, \lambda_k$ los autovalores de A, Q_{jn} la fila j-ésima y la columna n-ésima de Q y U_{ni} la n-ésima coordenada del autovector i-ésimo de A. Entonces, el i-ésimo autovector será,

$$\bar{U}_i = \begin{pmatrix} \frac{1}{\lambda_i} \sum_{n=1}^{k} Q_{1n} U_{ni} \\ \vdots \\ \frac{1}{\lambda_i} \sum_{n=1}^{k} Q_{Mn} U_{ni} \end{pmatrix} = \frac{1}{\lambda_i} \begin{pmatrix} A \\ B^T \end{pmatrix} U \tag{2.31}$$

Y, por tanto,

$$\bar{U} = \begin{pmatrix} A \\ B^T \end{pmatrix} U\Lambda^{-1} = \begin{pmatrix} AU\Lambda^{-1} \\ B^T U\Lambda^{-1} \end{pmatrix} = \begin{pmatrix} U \\ B^T U\Lambda^{-1} \end{pmatrix} \tag{2.32}$$

Finalmente se ortogonalizan las columnas de $\bar{U}$. Sea $A^{\frac{1}{2}}$ la raíz cuadrada definida positiva de A, sea $S = A + A^{-\frac{1}{2}} B B^T A^{-\frac{1}{2}}$ y su diagonalización $S = R\hat{\Lambda}R^T$. Se define la matriz $\hat{V}$ como,

$$\hat{V} = \begin{pmatrix} A \\ B^T \end{pmatrix} A^{-\frac{1}{2}} R \hat{\Lambda}^{-\frac{1}{2}} \tag{2.33}$$

Se puede probar que $\hat{V}$ y $\hat{\Lambda}$ diagonalizan $\hat{Q}$, es decir $\hat{Q} = \hat{V}\hat{\Lambda}\hat{V}^T$ y $\hat{V}^T\hat{V} = I_k$.

Capítulo 3

Método propuesto

En este trabajo se propone un nuevo método de selección de variables, *Quadratic Programming Feature Selection* (QPFS), para problemas de clasificación multiclase mediante optimización cuadrática. El principal objetivo del algoritmo es proporcionar un procedimiento de complejidad temporal menor que en mRMR para problemas de clasificación de alta dimensionalidad ofreciendo, a su vez, resultados similares en términos de acierto en clasificación.

Uno de los algoritmos de selección más eficientes desde el punto de vista de acierto en clasificación es mRMR, pero su escalabilidad para problemas de cientos de miles de variables es costosa computacionalmente por su dependencia cuadrática en la dimensión del problema (sección 3.3.1). Además, las soluciones propuestas por los algoritmos MaxDep, mRMR o MaxRel van construyendo secuencialmente el subconjunto de variables, con lo que la elección de la primera variable, en función únicamente de la dependencia con la clase, pudiera ser crítica a la hora de encontrar un subconjunto poco redundante. En el método que proponemos se consideran todas las variables de una sola vez y se minimiza la función objetivo.

Una de las principales ideas subyacentes en este nuevo método se encuentra en el problema de selección de carteras dentro del mercado económico tal y como se introdujo en la sección 2.2.1. En el contexto de la selección de variables, proponemos considerar el vector solución $x \in \mathbb{R}^M$ como una medida de la importancia de cada variable y proporcionar una ordenación de las mismas en función de este valor; con ello, trataremos de hallar un subconjunto de variables que, por un lado minimicen la interdependencia entre ellas (término cuadrático) y por otro, maximicen la relevancia con la clase (término lineal).

Nuevamente supondremos un problema de clasificación de C clases formado por N patrones, cada uno de ellos representado como un vector en $\mathbb{R}^{M+1}$ formado por el con-

junto de M variables y un valor adicional que indica la clase a la que pertenece y que denotaremos por c con $0 \leq c \leq C-1$. Como se expuso en la sección 2.2, el problema de la optimización cuadrática consiste en la minimización de una función cuadrática en un espacio multidimensional sujeta a restricciones lineales sobre las variables. En nuestro caso, la matriz Q representará la similitud entre pares de variables (redundancia) y el vector F una medida de dependencia entre cada variable y la clase c.

En ocasiones, puede ser interesante introducir un parámetro de ponderación α para controlar la importancia del término lineal (relevancia) frente al término cuadrático (redundancia). En aquellos tipos de problemas donde las variables no sean especialmente redundantes potenciar aquellas más relevantes respecto a la clase puede dar mejores resultados. En cambio, si el problema tiene un alto nivel de redundancia (por ejemplo, tareas de selección de genes), favorecer a las variables relevantes puede provocar una selección de variables redundantes menos beneficiosa en términos de acierto que aquella que hubiera premiado la independencia, es en estos casos donde un valor de alfa más próximo a cero puede ser lo más acertado.

El problema de optimización se reformula entonces del siguiente modo:

$$\min_x \left\{ \frac{1}{2}(1-\alpha)x^T Q x - \alpha F^T x \right\} \tag{3.1}$$

Donde x, Q y F se definen como antes y $\alpha \in [0,1]$. Si $\alpha = 1$, únicamente se considera la relevancia de las variables con la variable objetivo y, por tanto, QPFS es equivalente a maxRel cuando F representa la información mutua variable-clase. Por otra parte, si $\alpha = 0$, las variables con mayor peso son las que minimizan la interdependencia en conjunto, sin importar su dependencia con la clase. La elección de un valor apropiado para el parámetro α depende del tipo de problema de clasificación o del término de la ecuacion (3.1) que se desee potenciar más.

En cuanto a las restricciones del problema impondremos que cada uno de los coeficientes de la solución sea positivo y que su suma sea unitaria,

$$x_i \geq 0, \forall i = 1, \ldots, M \tag{3.2}$$

$$\sum_{i=1}^{M} x_i = 1 \tag{3.3}$$

El objetivo de estas restricciones es establecer un *mapa* entre las soluciones del problema y el intervalo $[0,1]$ de manera que si se realiza la selección de variables sobre dos subconjuntos distintos procedentes de la misma fuente de datos, las soluciones obtenidas

en cada caso sean *comparables*.

La salida del algoritmo será un ranking de todas las variables de acuerdo a su peso en el vector solución x: las variables con mayor peso serán las más deseables en el problema de clasificación en términos de independencia y relevancia. En caso de que el peso de dos variables sea el mismo ($x_i = x_j$), se considerará *mejor* aquella variable v_k cuya relevancia F_k con la clase sea mayor.

3.1. Medidas de similitud propuestas

El método de selección de variables descrito en la ecuación (3.1) admite cualquier medida de similitud intra-variable siempre y cuando la matriz Q sea simétrica. No existen restricciones respecto a la similitud variable-clase. En este trabajo, se han probado como medidas de similitud la correlación de Pearson (analizada en la siguiente sección) y la información mutua (sección 2.1.1).

3.1.1. Correlación de Pearson

Como primera aproximación, se propone el coeficiente de correlación de Pearson como medida de similitud entre variables debido a su simplicidad de cálculo y su demostrada utildad en otros métodos de selección (*Correlation Feature Selection, CFS* [22]) o de generación de variables (*Principal Component Analysis,PCA* [13]).

La correlación entre dos variables aleatorias v_i y v_j es una medida estadística que cuantifica la dependencia **lineal** entre ambas y la dirección de esta dependencia. Existen varios coeficientes para medir la correlación entre dos variables, siendo el más conocido el coeficiente de correlación de Pearson.

El coeficiente de correlación de Pearson se define en el intervalo $[-1, 1]$. Un coeficiente de correlación de Pearson entre dos variables v_i y v_j toma el valor 1 si existe una dependencia lineal directa entre ambas variables, el valor -1 es alcanzado cuando esta dependencia es indirecta. Cuanto más se aproxime el coeficiente de Pearson a los valores extremos -1 y 1, mayor es la correlación entre las variables. En el caso de que dos variables sean independientes, la correlación entre ambas es 0, pero el recíproco no es cierto: la correlación detecta sólo dependencias lineales entre variables. Únicamente correlación nula es equivalente a independencia en el caso de que la distribución conjunta de v_i y v_j sea gausiana.

Formalmente, el coeficiente de correlación de Pearson entre dos variables aleatorias v_i y v_j se define como,

$$\rho_{ij} = \frac{\mathrm{cov}(v_i, v_j)}{\sqrt{\mathrm{var}(v_i)\mathrm{var}(v_j)}} \tag{3.4}$$

Donde, *cov* denota la covarianza entre las variables y *var* la varianza de cada una de ellas. La estimación muestral de la correlación viene dada por,

$$\hat{\rho}_{ij} = \frac{\sum_{k=1}^{N}(v_{ki} - \bar{v_i})(v_{kj} - \bar{v_j})}{\sqrt{\sum_{k=1}^{N}(v_{ki} - \bar{v_i})^2 \sum_{k=1}^{N}(v_{kj} - \bar{v_j})^2}} \tag{3.5}$$

Donde N es el número de muestras de que se dispone, v_{ki} representa la muestra k-ésima de la variable aleatoria v_i y $\bar{v_i}$ la media muestral de la variable aleatoria v_i.

Para el algoritmo QPFS, tomaremos como $Q = (q_{ij})$ la matriz del valor absoluto del coeficiente de correlación de Pearson entre las variables v_i y v_j, es decir $(q_{ij}) = |\hat{\rho}_{ij}|$, asumiendo que valores próximos a ± 1 representan redundancia entre las variables.

Para problemas multiclase (no binarios), la definición anterior de la correlación variable-clase no es suficiente, dado que la asignación arbitraria de una etiqueta a cada una de las clases implica una ordenación topológica de las mismas carente de sentido. Así pues, para problemas multiclase, se sugiere la utilización del denominado *Coeficiente de Correlación Ponderado de Pearson* [23]. Se define la dependencia entre la variable v_i y la variable objetivo c en un problema de C clases como,

$$\hat{\rho}_{ic} = \sum_{j=1}^{C} \hat{p}(c = c_j)|\hat{\rho}_{iC_j}| \tag{3.6}$$

Donde C_j es una variable binaria auxiliar que toma el valor 1 cuando $c = c_j$ y 0 en otro caso.

3.1.2. Información Mutua

Una de las principales limitaciones de la correlación de Pearson es que solo capta dependencias lineales entre las variables. En muchos de los problemas reales de clasificación una dependencia de este tipo no es suficiente y es necesario extender la medida

de similitud para que sea capaz de detectar otro tipo de relaciones. Frente a esta carencia se propone la utilización de la información mutua como medida de similitud tal y como se expuso en la sección 2.1.1. Si bien la información mutua es una medida más robusta que la correlación, precisa de una mayor cantidad de datos para ofrecer una estimación fiable y requiere la aproximación de las correspondientes funciones de densidad.

3.2. Aproximación al problema en un subespacio

En problemas de alta dimensión es muy probable que exista una alta dependencia entre las variables, en estos casos la matriz de similitud intravariable Q será singular y el problema planteado en la ecuación (3.1) se puede reformular en un espacio de menor dimensión, reduciendo así el coste computacional del algoritmo.

Dada la diagonalización de la matriz simétrica $Q = U\Lambda U^T$, el problema (3.1) es equivalente a,

$$\min_x \left\{ \frac{1}{2}(1-\alpha)x^T U\Lambda U^T x - \alpha F^T x \right\} \tag{3.7}$$

Supongamos ahora que el rango de la matriz Q es $k < M$ entonces, consideremos los primeros k autovalores de la matriz diagonal Λ y sus autovectores, entonces el problema se reduce a resolver un problema de optimización en un espacio k-dimensional,

$$\min_y \left\{ \frac{1}{2}(1-\alpha)y^T \tilde{\Lambda} y - \alpha F^T \tilde{U} y \right\} \tag{3.8}$$

Sujeto a $M+1$ restricciones,

$$\tilde{U}y \geq \overrightarrow{0} \tag{3.9}$$

$$\sum_{j=1}^{k}\sum_{i=1}^{M} u_{ij}y_j = 1 \tag{3.10}$$

Donde $\tilde{\Lambda}$ es la matriz cuadrada formada por las k primeras filas y columnas de Λ, $\tilde{U}$ es una matriz $M \times k$ con los primeros k autovectores de Q, u_{ij} la i-ésima coordenada del j-ésimo autovector y $y = \tilde{U}^T x$ un vector en $\mathbb{R}^k$. Deshaciendo el cambio de variable, se recupera el vector x original: $x = \widetilde{U}y$.

En la figura 3.1 se muestra el esquema del algoritmo propuesto,

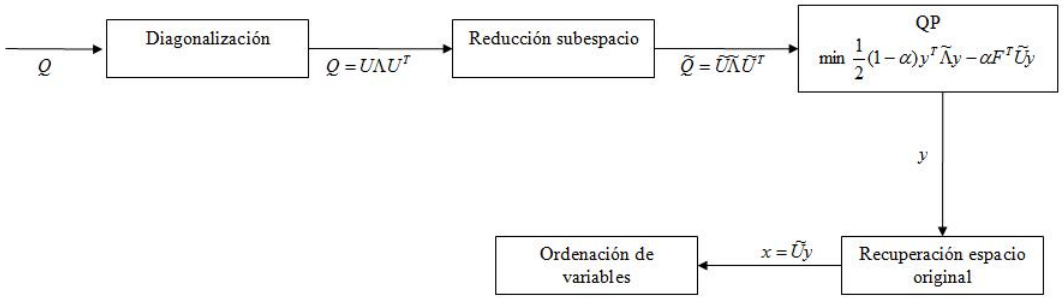

Figura 3.1: Esquema del algoritmo QPFS

3.2.1. Estimación del error

En los conjuntos de datos reales es inusual tener una matriz estrictamente singular pero, si el problema es redundante Q puede ser razonablemente bien aproximada por una matriz de menor rango usando su descomposición diagonal y estimando el rango $\tilde{k}$ de la aproximación como el número de autovalores mayores a un umbral prefijado $\delta \geq 0$. Entonces, se define la matriz aproximada $\tilde{Q}$ como $\tilde{Q} = U\Gamma U^T$, donde Γ es la matriz Λ en la que los autovalores menores que δ se han igualado a cero [14].

Sea x^* la solución óptima del problema original (3.1) y sea $\tilde{x}^*$ la solución del problema aproximado (3.8). Definamos también las funciones objetivo,

$$f(x) = \frac{1}{2}(1-\alpha)x^T Q x - \alpha F^T x \tag{3.11}$$

$$\tilde{f}(x) = \frac{1}{2}(1-\alpha)y^T \tilde{\Lambda} y - \alpha F^T \tilde{U} y \tag{3.12}$$

Entonces, podemos estimar el error cometido al resolver el problema aproximado en lugar del original a partir del teorema enunciado en [14]:

Teorema 3.2.1 *Si $(Q - \tilde{Q})$ es semidefinida positiva y traza$(Q - \tilde{Q}) \leq \epsilon$, entonces $f(x^*) - f(\tilde{x}^*) \leq \frac{d^2 l \epsilon}{2}$, donde l es el número de restricciones activas en el problema aproximado y d una cota superior para cada uno de los elementos del vector solución.*

En nuestro caso, $0 \leq x_i \leq 1$, por tanto, $d = 1$ y, efectivamente, $(Q - \tilde{Q})$ es semidefinida positiva, ya que $(Q - \tilde{Q}) = U(\Lambda - \Gamma)U^T$ y $(\Lambda - \Gamma)$ es una matriz diagonal con autovalores positivos acotados superiormente por δ. También se verifica $\epsilon \leq (M - k)\delta$. Por tanto,

$$f(x^*) - f(\tilde{x}^*) \leq \frac{l(M-k)\delta}{2} \tag{3.13}$$

3.2.2. QPFS y el método Nyström

La reducción del problema de optimización cuadrática en un espacio de menor dimensión se fundamenta en la esperable redundancia en los datos que hace que la matriz de término cuadrático sea quasi-singular. Este hecho puede aprovecharse para reducir el coste de la diagonalización de Q utilizando el método de Nyström, donde los autovalores y autovectores de Q se aproximan resolviendo un problema de diagonalización en un espacio de menor dimensión a partir un submuestreo de las variables (sección 2.3). El proceso de selección de variables con esta modificación se muestra en la figura 3.2.

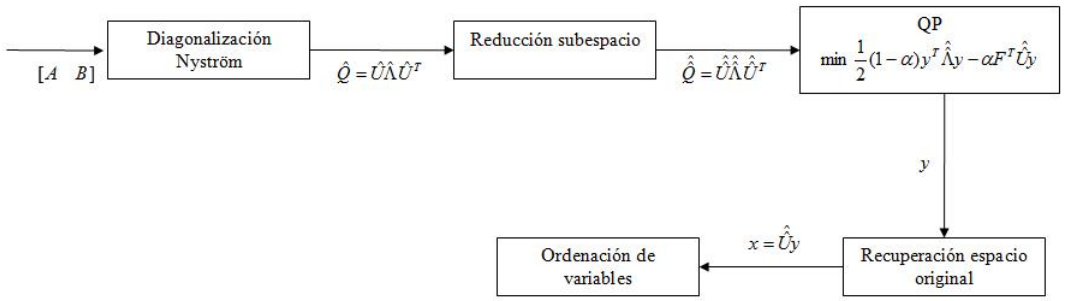

Figura 3.2: Esquema del algoritmo QPFS utilizando la aproximación de Nyström.

Uno de los puntos críticos del método de Nyström es la elección del subconjunto de variables a considerar. No existe un procedimiento eficiente para submuestrear los datos, aunque se han iniciado varias investigaciones al respecto [12, 36, 43, 51]. En la mayoría de las aplicaciones se realizan submuestreos aleatorios de los datos [17, 48] en los que se puede estimar la calidad de la aproximación mediante la norma $\|E - B^T A^{-1} B\|$ (sección 2.3).

Al utilizar el método de Nyström para realizar la selección de variables existen por tanto dos fuentes de error: la debida a la aproximación de Nyström y la procedente de la resolución del problema cuadrático en un subespacio de menor dimensión que el original.

3.3. Análisis de complejidad

Una de las principales ventajas de QPFS frente a mRMR es la reducción de la complejidad temporal. Dado un problema de clasificación con N patrones de entrenamiento y M variables, tratamos de proporcionar una permutación de las variables de acuerdo a cierto criterio de relevancia. En cada uno de los casos, analizaremos el coste temporal que supone el cálculo de la medida de similitud y la complejidad del algoritmo de selección

en sí.

En este trabajo hemos optado por la discretización previa de las variables continuas para estimar sus funciones de densidad. Es importante notar que, bajo este supuesto, el coste temporal de calcular la información mutua es igual al de la correlación, dado que el proceso de discretización tiene coste lineal en el número de patrones y en la dimensión del problema, mientras que el cálculo de la información mutua y la correlación es lineal en N pero cuadrático en M. Así pues, el análisis de rendimiento que sigue es el mismo independientemente de la medida de simililtud a utilizar.

3.3.1. mRMR

En este caso, para obtener una permutación de las variables originales del problema, es necesario en primer lugar, calcular la similitud entre cada una de las variables y la clase, así como la similitud entre cada par de variables. Es decir, serán necesarias del orden de N operaciones para el cálculo de similitud clase-variable y del orden de NM^2 operaciones para el cómputo intervariable. En resumen, el coste del cálculo de la similitud para el algoritmo mRMR será $O(NM^2)$.

En segundo lugar, el algoritmo de selección consite en elegir iterativamente la *mejor* variable teniendo en cuenta las elegidas en los pasos anteriores. Dado que buscamos una permutación de las variables, serán necesarias M iteraciones para, realizar en cada una de ellas una búsqueda lineal. Por tanto, el coste de la obtención de la lista ordenada será $O(M^2)$.

Se concluye, por tanto, que el coste temporal del algoritmo mRMR es $O(NM^2)$ esto es, lineal en el número de patrones y cuadrático para la dimensión.

3.3.2. QPFS sin la aproximación de Nyström

Al igual que sucede con mRMR, la obtención de una permutación de las variables requiere, en primer lugar, el cálculo de la medida de similitud intervariable y variable-clase, cuya complejidad temporal es $O(NM^2)$.

El procedimiento de ordenación de las variables consiste en la resolución de un problema de programación cuadrática, cuyo coste temporal es cúbico en el espacio de variables $O(M^3)$ [48]. No obstante, esta estimación es una cota superior que, difícilmente llegará a alcanzarse en el modelo propuesto por dos motivos: el algoritmo resolverá generalmente el problema cuadrático en un subespacio de menor dimensión y la matriz del término cuadrático es diagonal. La diagonalización de la matriz Q puede realizarse mediante

algoritmos como SVD o Jacobi [39], de coste $O(M^3)$.

Por tanto, el coste temporal del algoritmo QPFS sin utilizar la aproximación de Nyström es máx$(O(M^3), O(NM^2))$; es decir, si $N \gg M$, la complejidad es lineal en el número de patrones y cuadrática en el espacio de variables mientras que, en caso contrario, el rendimiento es cúbico para la dimensión del problema e *independiente* del número de muestras.

3.3.3. QPFS + Nyström

Sea $p \in (0, 1]$ la proporción de variables muestreadas en el método Nyström y $k = pM$ el número de submuestreos. Entonces, la matriz del término cuadrático Q queda descompuesta del siguiente modo,

$$Q = \begin{pmatrix} A & B \\ B^T & E \end{pmatrix}$$

Y será suficiente con aproximar la submatriz $[A \quad B] \in \mathbb{R}^{k \times M}$, lo que supone un coste de $O(NpM^2)$.

La complejidad de la diagonalización de la matriz del término cuadrático mediante el método de Nyström es $O(k^2M)$ [48], equivalentemente $O(p^2M^3)$. La utilización de tal procedimiento con k submuestreos, implica que el rango de la matriz será a lo sumo k y, por tanto, la complejidad temporal de la optimización cuadrática para este caso $O(k^3)$, esto es $O(p^3M^3)$. Obviamente $p^2M^3 > p^3M^3$, con lo que se concluye que el coste del procedimiento de selección de variables en este caso será $O(p^2M^3)$.
La inclusión del parámetro p dentro del análisis temporal anterior queda justificada por tratarse de un parámetro del algoritmo cuya diferencia en maginitud con la dimensión M afecta a la complejidad del algoritmo, tal y como se muestra en los resultados experimentales (sección 4.2).

En síntesis, la complejidad temporal del algoritmo QPFS utilizando la aproximación de Nyström con una proporción de submuestreos p es máx$(O(NpM^2), O(p^2M^3))$. Es decir, para $N \gg pM = k$ se tiene una complejidad temporal $O(NpM^2)$ y, en caso contrario $O(p^2M^3)$.

3.3.4. Análisis compartativo mRMR vs. QPFS

En esta sección se realizará un estudio sobre en qué tipos de problemas es más conveniente utilizar QPFS frente a mRMR en términos de coste temporal; para ello, se utilizan

los resultados obtenidos en los tres apartados anteriores y que se resumen en la tabla 3.1.

	mRMR	QPFS	QPFS+Nyström
$N \ll pM$	$O(NM^2)$	$O(M^3)$	$O(p^2M^3)$
$N \gg pM$	$O(NM^2)$	$O(M^3)$	$O(NpM^2)$
$N \gg M$	$O(NM^2)$	$O(NM^2)$	$O(NpM^2)$

Tabla 3.1: Complejidad temporal de los algoritmos mRMR y QPFS en función del número de patrones N, la dimensión del problema M, y la proporción de submuestreos p en el método de Nsytröm.

El coste temporal del QPFS sin Nyström supera o iguala al de mRMR. En cambio, para QPFS con Nyström la complejidad temporal del algoritmo propuesto es inferior a la de mRMR para $N \gg p^2M$. Tomando $p^2 \approx 10^{-2}$, se tiene que QPFS+Nyström es más eficiente en tiempo que mRMR si $N \gg 10^{-2}M$. Es decir, el número de patrones es menor que 10^{-2} veces la dimensión del problema.

Capítulo 4

Experimentos

Este capítulo se centra en comparar empíricamente los algoritmos maxRel (*maximal relevance*), mRMR (*minimal-redundancy-maximal-relevance*) y QPFS. En primer lugar se abordan distintos aspectos relacionados con la implementación de los algoritmos y el clasificador con el que se evaluará el nivel de acierto de cada uno de ellos. Además, se proporciona una descripción detallada de los conjuntos de datos utilizados y el método de evaluación empleado en cada uno de ellos para, finalmente, exponer y analizar los resultados obtenidos en términos de acierto en clasificación y coste computacional.

4.1. Implementación

La mayor parte del código se ha implementado en C utilizando el paquete estándar *Linear Algebra PACKage (LAPACK)* [1] para las operaciones básicas con matrices: diagonalización, SVD, multiplicación e inversión. La optimización cuadrática se ha llevado a cabo mediante el algoritmo de Goldfarb e Idnani [19] utilizado en el paquete *quadprog* [44] de R e implementado en Fortran.

En la sección 2.1.1 vimos que el cálculo de la información mutua presentaba el inconveniente de la estimación de las funciones de densidad de cada una de las variables del problema, especialmente en el caso de variables continuas, presentes en todos los conjuntos de datos utilizados. Se ha optado por discretizar tales variables en 3 segmentos, correspondientes a las posiciones $(-\infty, \mu - \sigma]$, $(\mu - \sigma, \mu + \sigma]$,$(\mu + \sigma, +\infty)$, siendo μ la media muestral de la variable y σ su desviación típica. Esta aproximación es muy sencilla, poco costosa y suficiente para comprobar la eficiencia del algoritmo propuesto y su comparativa frente a mRMR dado que en ambos casos la discretización de las variables es idéntica y, con ello, la aproximación de la información mutua.

Los algoritmos a comparar en este trabajo no requieren de un clasificador subyacente en particular. Al tratarse de métodos de selección *de filtro*, es de esperar que su com-

portamiento sea similar independientemente del clasificador utilizado. Por simplicidad se ha optado por el SVM con kernel lineal implementado en el paquete LIBSVM 2.6 [7] válido para problemas binarios y multiclase.

4.1.1. Conjuntos de datos

Los cinco conjuntos de datos utilizados en este trabajo se muestran en la tabla 4.1 junto con el promedio del porcentaje de error sin realizar selección de variables utilizando como clasificador un SVM lineal, y referencias a trabajos anteriores en los que se han empleado para evaluar los métodos de selección de variables .

Dataset	N	M	C	Error de referencia	Referencias
ARR	422	278	2	21.81 %	[37, 52]
NCI60	60	1123	9	38.67 %	[31, 52, 53]
SRBCT	83	2308	4	0.22 %	[31, 53]
GCM	198	16063	14	33.85 %	[31, 52, 53]
RAT	181	8460	2	8.61 %	[25]

Tabla 4.1: Descripción de los conjuntos de datos. N representa el número de patrones, M la dimensión del problema y C el número de clases. Se incluyen referencias a otros trabajos en los que se han utilizado estos conjuntos de datos para la selección de variables.

- **ARR**: Conjunto de datos de arritmias cardíacas disponible en el repositorio UCI Machine Learning [2]. Se trata de un problema binario (presencia/ausencia de arritmia cardíaca) con 422 patrones y 478 variables.

- **NCI60**: Conjunto de datos estudiado inicialmente en [41]. Se utilizan microarrays cDNA para examinar la variación en la expresión genética de 1123 genes en 60 *cell lines* para distinguir entre 9 tipos distintos de cáncer. El número de muestras de cada tipo es como mínimo 2 y como máximo 9. Los datos están disponibles en `http://users.cis.fiu.edu/ yzhan004/genesel.html`.

- **SRBCT**: Conjunto de datos *Small Round Blue Cell Tumors* en niños, formado por 83 patrones de 4 clases (se exluyendo muestras no-SRBCT) y 2308 marcadores. Disponible en `http://research.nhgri.nih.gov/microarray/Supplement/`.

- **GCM**: Conjunto de datos formado por 198 muestras de tumores humanos de 15 tipos distintos [40]. Se dispone de información sobre 16063 marcadores (variables). El dataset puede obtenerse en `http://users.cis.fiu.edu/ yzhan004/genesel.html`.

- **RAT**: Dataset obtenido del estudio de respuesta genética en ratas a diferentes medicamentos y tóxicos [34]. Se trata de un problema binario de identificación de sustancias tóxicas. El dataset se ha formado a partir de chips de microarrays de cRNA con 8565 marcadores(variables) y ha sido preprocesado eliminando aquellas variables con más de un 10 % de missing values y sustituyendo el resto de missing values por la media aritmética de la variable sobre los ejemplos disponibles, lo que reduce la dimensión del problema a 8460. Se dispone de 61 muestras de tóxicos y 120 de otras sustancias. Conjunto de datos disponible en *NIH Gene Expression Omnibus (GEO)*, número de acceso GSE2187.

Las principales características de estos conjuntos son que el número de ejemplos de cada clase es generalmente pequeño y desbalanceado, todo ello unido al alto número de clases como en NCI60 o GCM, dificulta la tarea de clasificación, teniéndose en la mayoría de los casos un alto porcentaje de error.

4.1.2. Metodología

Para estimar el error de clasificación para los datasets ARR, NCI60, SRBCT y GCM se ha utilizado como método de validación cruzada 10CV [13] con 100 ejecuciones distintas, con lo que los resultados obtenidos son comparables a los descritos en la bibliografía [31, 37, 52, 53]. En el caso del conjunto de datos RAT se ha seguido la misma metodología que en [25], utilizando 120 patrones para entrenamiento (61 para test) y 300 permutaciones.

En cuanto a la complejidad temporal, se han realizado dos tipos de pruebas para comprobar que el comportamiento empírico de los algoritmos coincide con lo que cabría esperar del análisis teórico expuesto en la tabla 3.1. En ambos casos se han medido los tiempos de ejecución para los algoritmos mRMR, QPFS sin Nyström y QPFS+Nyström y distinta proporción de submuestreos para 50 permutaciones distintas de los datos. Las medidas han sido realizadas en un equipo con doble procesador Intel(R) Core(TM) CPU 4300 a 1.80GHz y han consistido en:

- **Comportamiento en función del número de patrones** N. Para las pruebas se ha utilizado en dataset SRBCT modificado: se ha fijado la dimensión en $M = 140$ y se ha incrementado artificialmente el número de patrones en un factor de 4, esto es, $N = 332$.

- **Comportamiento en función de la dimensión** M. En este caso se ha utilizado el conjunto de datos SRBCT sin modificar, $N = 83$ y $M = 2308$.

4.2. Discusión y conclusiones

El principal objetivo de los experimentos realizados es probar que el algoritmo QPFS+Nyström es considerablemente más rápido que mRMR para problemas de alta dimensionalidad con tasas de acierto similares a las de mRMR. Como objetivo secundario se tiene la comparación de la correlación de Pearson y la información mutua como medidas de similitud.

En la figura 4.1 se ha representado el coste temporal promedio en 50 iteraciones de los algoritmos mRMR y QPFS en función del número de patrones N del problema de clasificación y dejando fija la dimensión M. Para el algoritmo mRMR se observa claramente una dependencia lineal en N, tal y como se analizó teóricamente en la sección 3.3. En cuanto al algoritmo QPFS sin utilizar la aproximación de Nyström, se observa un comportamiento lineal, tendencia esperada de acuerdo al análisis teórico. Finalmente, en el caso de QPFS+Nyström, el algoritmo presenta un comportamiento lineal en N, aumentando la pendiente de la recta con el porcentaje de submuestreos p tomados para Nyström. Nuevamente, estos resultados coinciden con los que se obtuvieron teóricamente en la sección 3.3.

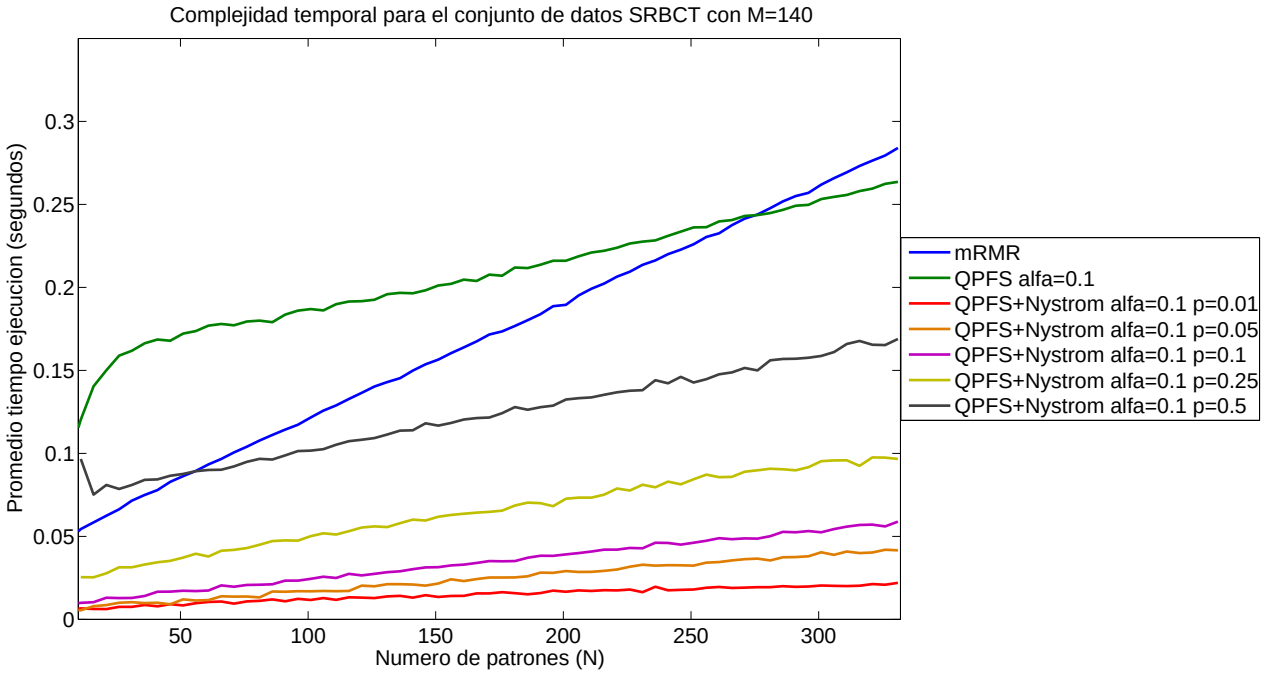

Figura 4.1: Complejidad temporal de los algoritmos mRMR y QPFS (con y sin Nyström) en función del número de patrones N del problema.

Como conclusión podemos decir entonces que el algoritmo QPFS+Nyström presenta una menor complejidad temporal que mRMR cuando el número de patrones del problema es grande (en la gráfica 4.1, únicamente mRMR tiene menor coste que QPFS+Nyström cuando el número de patrones es muy bajo en comparación a la dimensión y la proporción de submuestreos es muy elevada).

La gráfica 4.2 muestra el promedio en 50 iteraciones del coste temporal de los algoritmos mRMR y QPFS en función de la dimensión M del problema, dejando constante el número de patrones N. Para el algoritmo mRMR se observa una dependencia cuadrática en la dimensión coincidiendo con el resultado teórico formulado en la tabla 3.1. Para QPFS sin Nyström, se tiene una tendencia cúbica en la dimensión del problema debida al coste de la diagonalización y resolución del problema de optimización cuadrática, encontrándose siempre por encima de mRMR. Finalmente, QPFS+Nyström tiene una complejidad temporal inferior a la de los dos algoritmos anteriores, de comportamiento cuadrático y directamente dependiente del número de submuestreos p considerados en el método de Nyström. Para comprobar el orden de la tendencia en cada caso se ha realizado un *t-test* de significancia estadística sobre los coeficientes de los polinomios que ajustan cada una de las curvas por mínimos cuadrados [35].

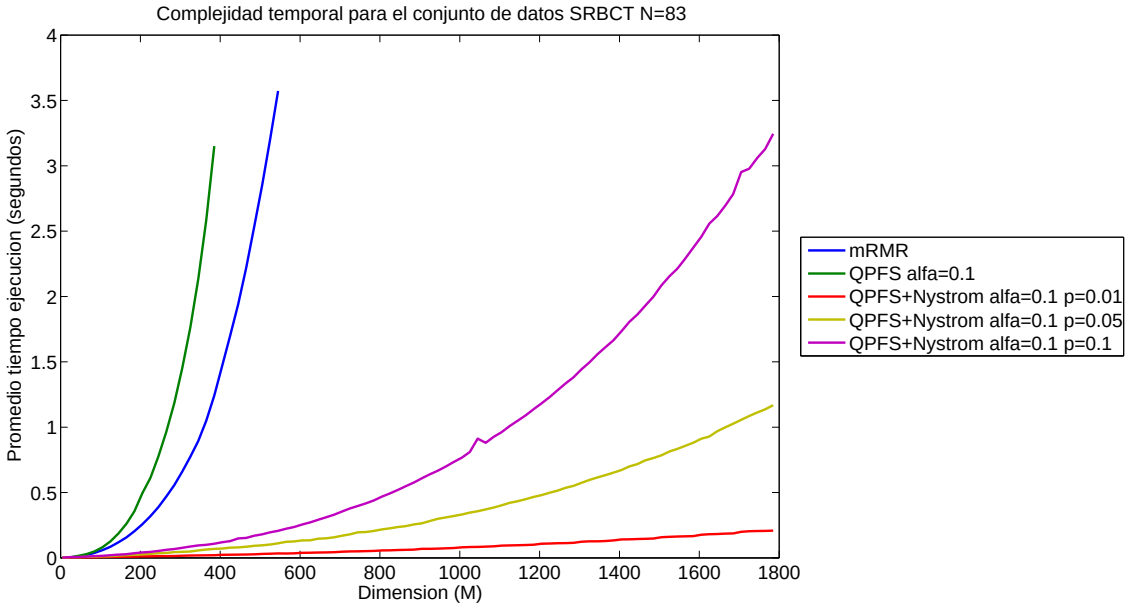

Figura 4.2: Coste temporal (segundos) de los algoritmos mRMR y QPFS en función de la dimensión M del problema.

En resumen, el algoritmo propuesto QPFS+Nyström presenta tiempos de ejecución

inferiores tanto a mRMR como a QPFS sin Nyström, acentuándose esta diferencia a medida que aumenta la dimensión del problema.

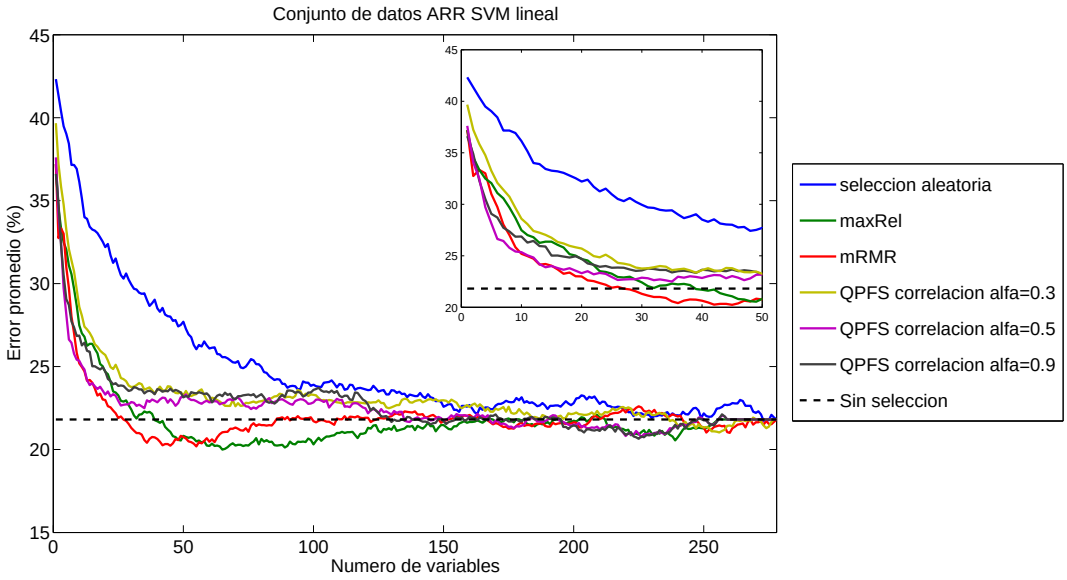

Figura 4.3: Comparación del error de clasificación para los algoritmos mRMR y QPFS con correlación para el conjunto de datos ARR.

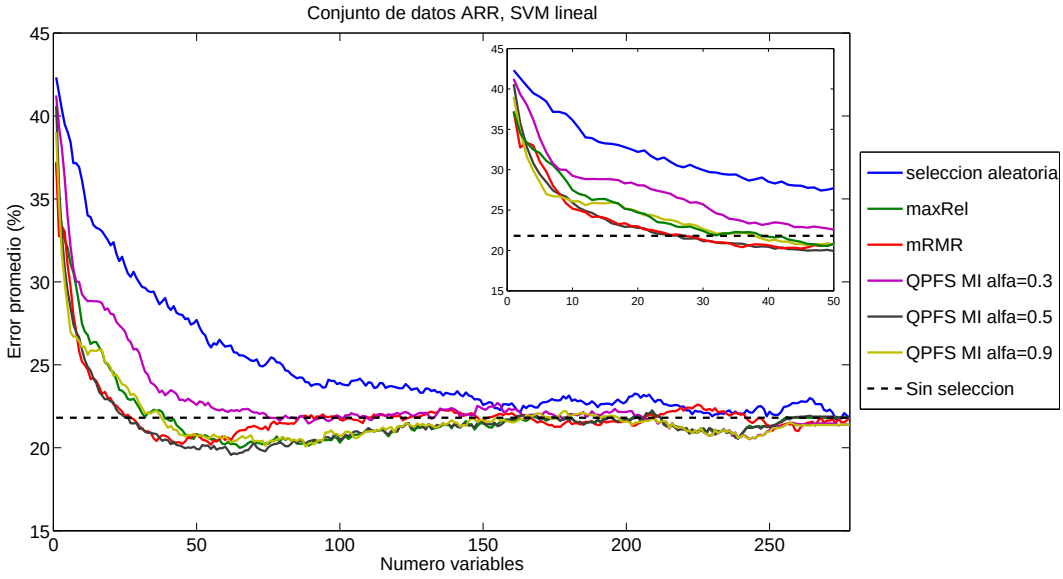

Figura 4.4: Comparación del error de clasificación para los algoritmos mRMR y QPFS con información mutua para el conjunto de datos ARR.

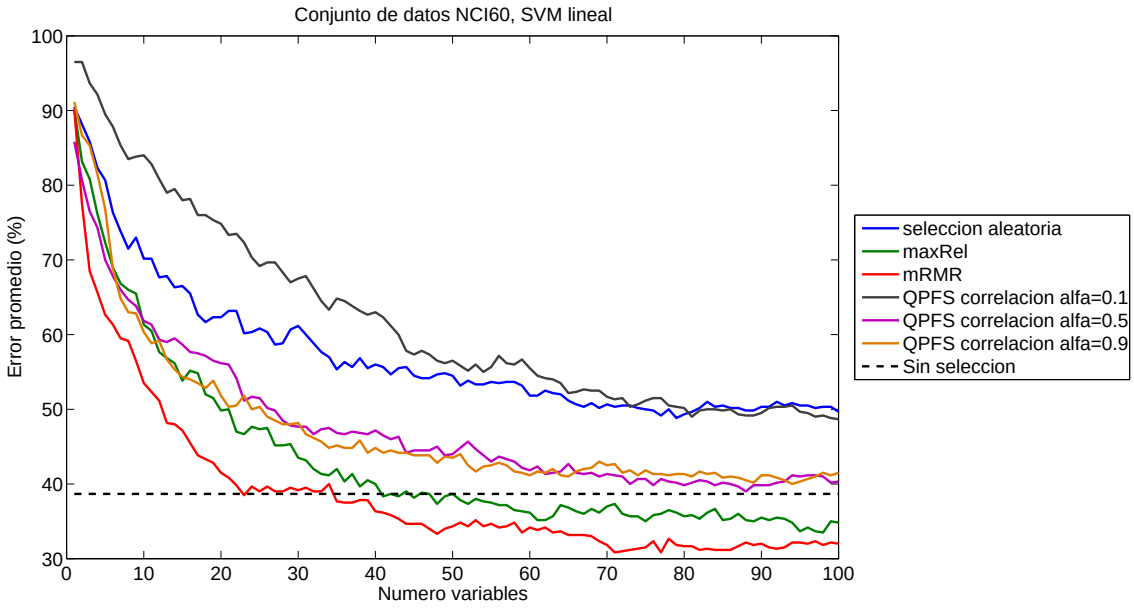

Figura 4.5: Comparación del error de clasificación para los algoritmos mRMR y QPFS con correlación para el conjunto de datos NCI60.

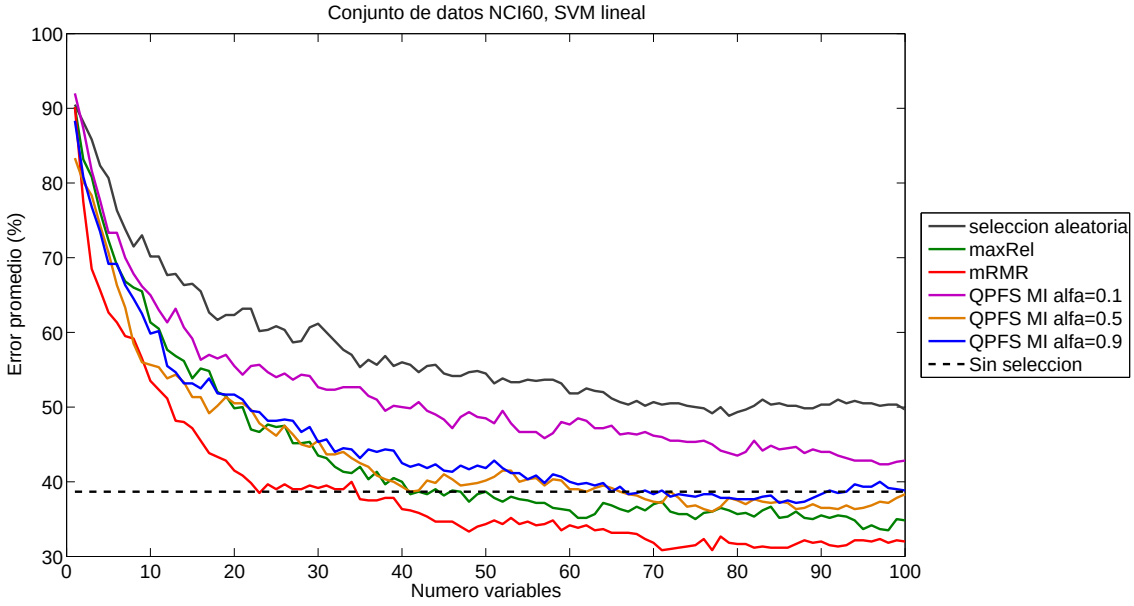

Figura 4.6: Comparación del error de clasificación para los algoritmos mRMR y QPFS con información mutua para el conjunto de datos NCI60.

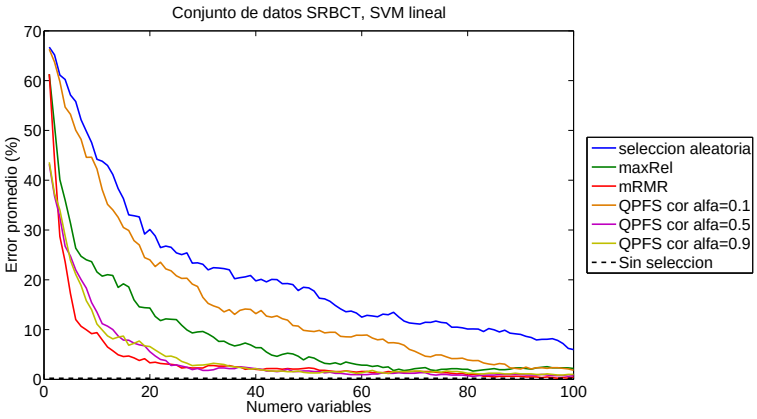

Figura 4.7: Comparación del error de clasificación para los algoritmos mRMR y QPFS con correlación para el conjunto de datos SRBCT.

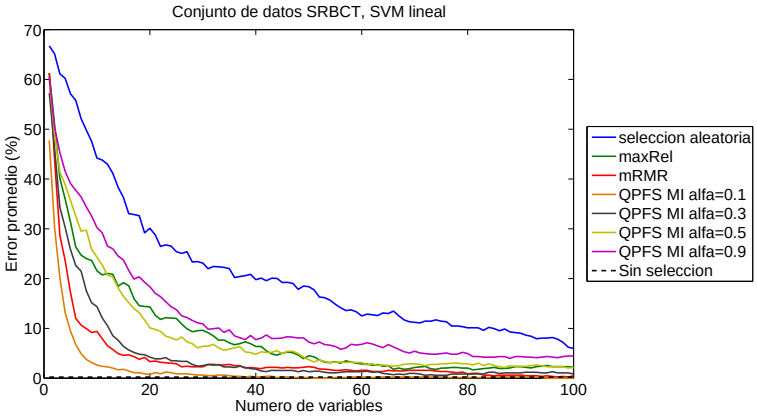

Figura 4.8: Comparación del error de clasificación para los algoritmos mRMR y QPFS con información mutua para el conjunto de datos SRBCT.

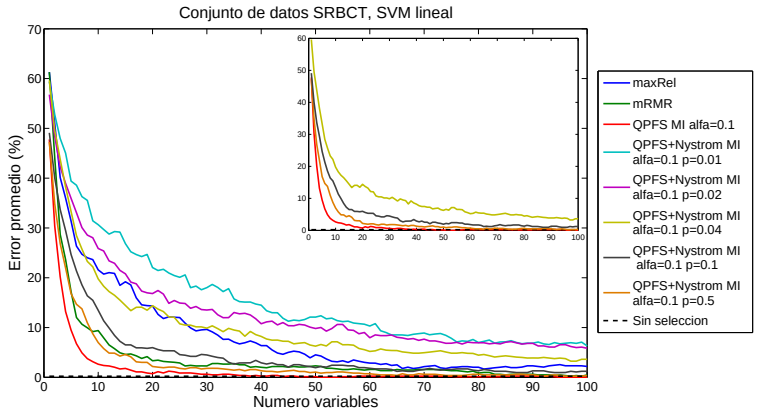

Figura 4.9: Comparación del error de clasificación para los algoritmos mRMR y QPFS+Nyström con información mutua para el conjunto de datos SRBCT.

La figuras 4.3 y 4.4 presentan el porcentaje promedio de error de clasificación (SVM lineal) para el problema ARR en función del tamaño del subconjunto de variables seleccionado por cada método y utilizando como medidas de similitud para QPFS la correlación y la información mutua, respectivamente. El empleo de la correlación como medida de similitud presenta resultados ligeramente peores que los métodos maxRel y mRMR, mientras que la tasa de aciertos al usar la información mutua es muy pareja a la de mRMR e incluso ligeramente mejor cuando el número de variables del subconjunto es superior a 60; en ambos casos, un valor $\alpha = 0{,}5$ sería adecuado. Para este conjunto de datos se ha representado la permutación completa de las variables y, parece que la elección de un tamaño aproximado de 50 sería la más acertada. En la gráfica también se muestra el rendimiento que se obtendría al realizar una selección aleatoria de variables, por debajo de cualquiera de los métodos de selección presentados.

Las figuras 4.5 y 4.6 muestran el porcentaje promedio de error de clasificación para el problema NCI60 en función del tamaño del subconjunto de variables empleando como medidas de similitud la correlación (figura 4.5) y la información mutua (figura 4.6). Nuevamente se observa que la correlación no llega a alcanzar las tasas de acierto de maxRel o mRMR. Cabe destacar que para $\alpha = 0{,}1$, los resultados que se obtienen son peores que para una selección aleatoria de variables, dando a entender la importancia de la relevancia de las variables para este conjunto de datos. La información mutua presenta resultados ligeramente mejores (figura 4.6), sin llegar a alcanzar el rendimiento de mRMR, pero bastante competitivos para $\alpha = 0{,}5$.

Las gráficas 4.7, 4.8 y 4.9 presentan el porcentaje promedio de error de clasificación para el conjunto de datos SRBCT en función del subconjunto de variables utilizando la correlación e información mutua, respectivamente, como medidas de similitud. Este problema es más *sencillo* que los anteriores, presentando tasas de acierto superiores al 90 %. Para la correlación, QPFS ($\alpha = 0{,}5$ y $\alpha = 0{,}9$) se encuentra entre maxRel y mRMR cuando el tamaño del subconjunto es menor de 20 variables; a partir de ese momento, QPFS y mRMR muestran comportamientos similares. En la figura 4.8 se observa como la información mutua cambia radicalmente el comportamiento del algoritmo QPFS en función de α, ahora los mejores resultados se obtienen para valores pequeños de α, en particular, para $\alpha = 0{,}1$, la tasa de acierto es superior a la de mRMR. Este hecho puede deberse a que el problema SRBCT sea altamente redundante pero la dependencia entre las variables no sea lineal y, por tanto, la correlación no determina correctamente el subconjunto de variables más *independientes*, mientras que la información mutua es capaz de detectar cualquier tipo de dependencia estadística. Finalmente, la gráfica 4.9 realiza una comparación entre la solución de QPFS para $\alpha = 0{,}1$ sin utilizar la aproximación de Nyström y el rendimiento de QPFS+Nyström para distinta proporción p de submuestreos: a medida que se aumenta el número de muestras, la solución obtenida se aproxima cada vez más a la solución sin la aproximación de Nyström. En este caso

particular, tomando un 50 % de submuestreos se alcanza prácticamente el mismo rendimiento que sin Nyström, aunque podría considerarse que un porcentaje del 10 % sería suficiente para tamaños superiores a 40 variables.

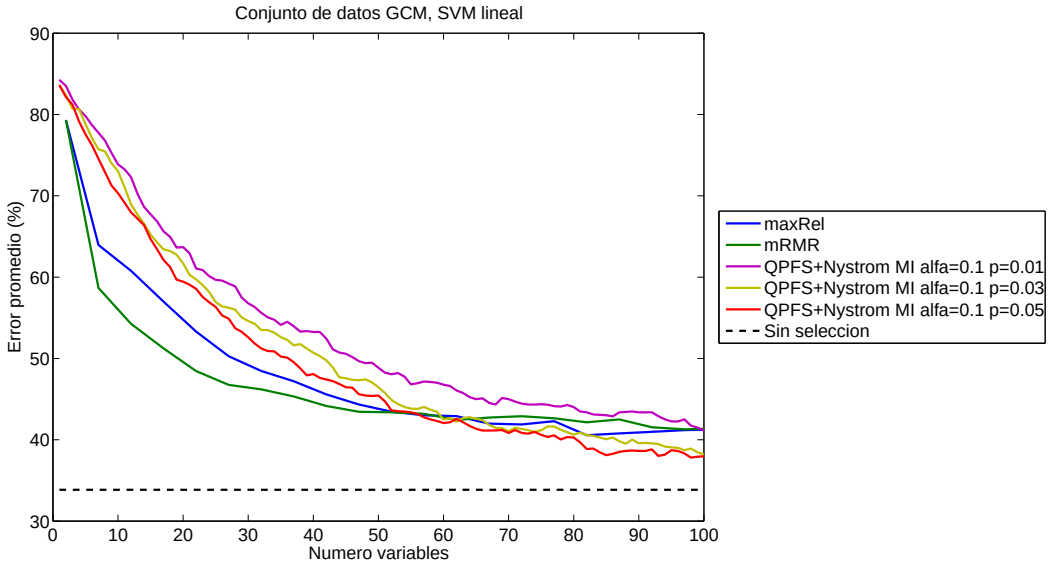

Figura 4.10: Comparación del error de clasificación para los algoritmos mRMR y QPFS+Nyström con información mutua para el conjunto de datos GCM.

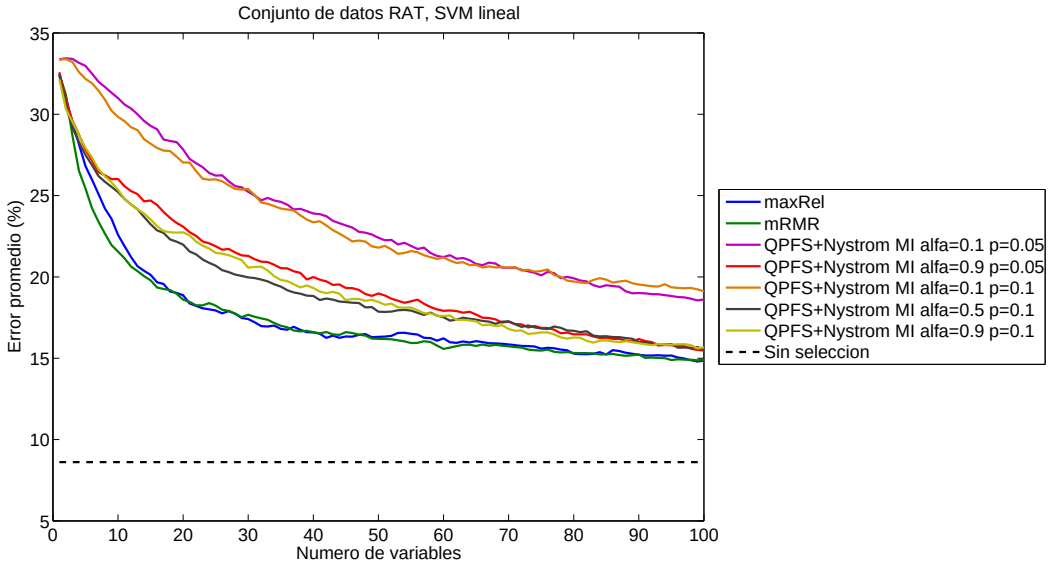

Figura 4.11: Comparación del error de clasificación para los algoritmos mRMR y QPFS+Nyström con información mutua para el conjunto de datos RAT.

La figura 4.10 representa el porcentaje promedio de error de clasificación para el problema GCM en función del tamaño del subconjunto de variables para maxRel, mRMR y QPFS+Nyström con $\alpha = 0{,}1$ e información mutua. Se observa que el acierto de maxRel y mRMR para subconjuntos de 60 variables o menos es considerablemente mejor que el obtenido para QPFS+Nyström; en cambio, si el tamaño del subconjunto es superior, ambos algoritmos presentan un error similar, siendo incluso el de QPFS ligeramente mejor. En cuanto al valor del parámetro p, parece que es suficiente tomar un 3 % de submuestreos, lo que reducirá considerablemente el tiempo de ejecución teniendo en cuenta que se trata de un problema de 16063 dimensiones.

Finalmente, los resultados para el conjunto de datos RAT se encuentran en la figura 4.11. La representación del error de clasificación frente al tamaño del subconjunto de variables muestra que, aunque QPFS+Nyström (información mutua) no llega a alcanzar a maxRel y mRMR en las primeras 100 variables, ofrece una tasa de acierto muy competitiva, especialmente para tamaños superiores a 80, necesitando únicamente un 5 % de submuestreos ($M = 8460$). La similitud entre los resultados de maxRel y mRMR hace pensar que para este problema, la utilización de algoritmos de filtro como maxRel sea la opción más acertada por su bajo coste y buenos resultados.

4.2.1. Conclusiones generales

El algoritmo propuesto QPFS+Nyström presenta un menor coste computacional que el método mRMR, acentuándose esta reducción en la complejidad cuando el número de patrones N y/o la dimensión M del problema es grande. A su vez, los resultados obtenidos en términos de acierto de clasificación son competitivos con los de mRMR, mejorándolos levemente en algunos casos (SRBCT) o experimentando un ligero empeoramiento en otros (NCI60); en cualquier caso, el orden de magnitud de la diferencia en el error de predicción es mucho menor que las diferencias en términos de complejidad temporal.

Se ha comprobado experimentalmente que el uso de la información mutua proporciona mejores resultados que la correlación. En todos los conjuntos de datos utilizados, los resultados obtenidos al utilizar la información mutua mejoran a sus equivalentes con correlación, siendo la mejora más significativa en unos casos que en otros.

Respecto al parámetro α (ponderación de la relevancia frente a la redundancia de variables), se ha observado que en función del problema, ciertos valores de α pueden proporcionar resultados significativamente mejores que otros. En cuanto al parámetro p (proporción de submuestreos en el método Nyström), a medida que éste aumenta, la solución proporcionada por QPFS se aproxima a la solución obtenida utilizando un método

de diagonalización *completa*; este hecho indica que tomando un número de submuestreos adecuado se puede llegar a obtener un rendimiento muy similar al que proporcionaría QPFS sin Nyström pero reduciendo considerablemente el coste temporal.

Capítulo 5

Trabajo futuro

El algoritmo de selección de variables QPFS, deja abiertas algunas líneas de investigación para sucesivas mejoras o aplicaciones a distintos campos del aprendizaje automático. A continuación se describen brevemente algunas de las principales direcciones en las que puede ir orientada la investigación basada en los conceptos desarrollados y/o resultados obtenidos a lo largo de este trabajo.

Los conjuntos de datos en los que se ha probado el algoritmo QPFS son de alta dimensionalidad pero reducido número de patrones. Sería deseable analizar los resultados experimentales del método propuesto cuando el número de patrones del problema y la dimensión son altos, casos en los que la eficiencia temporal de QPFS frente a mRMR se hace más significativa.

Uno de los problemas asociados a la selección de variables es determinar la dimensión del subconjunto candidato. Una de las técnicas más utilizadas en este sentido es la evaluación del error de clasificación en función del número de variables elegidas y considerar que se ha alcanzado un *buen* tamaño cuando la magnitud del error apenas varía; a ella recurren los métodos *wrapper* [27] o los autores de mRMR [37]. Este método requiere un conjunto de validación (no muy deseable en problemas de escaso número de patrones), es costoso y convierte la selección de variables dependiente del clasificador. Analizando la solución obtenida en la optimización cuadrática para distintos conjuntos de datos se ha observado que en algunos casos una gran parte de los coeficientes del vector solución eran nulos. Este hecho sugiere que quizás un estudio de la distribución de los coeficientes del vector solución del problema de optimización pueda dar una idea sobre el tamaño del subconjunto a seleccionar.

Una de las ventajas del algoritmo QPFS es que permite la utilización de cualquier medida de similitud siempre y cuando la matriz del término cuadrático sea simétrica. El uso de otro tipo de medidas especializadas (por ejemplo, en el ámbito de la clasifica-

ción de documentos [16]) o menos costosas computacionalmente que la correlación o la información mutua podría aportar resultados interesantes.

A lo largo de este trabajo, la selección de los parámetros α (factor de ponderación de la relevancia de las variables frente a su redundancia) y p (proporción de submuestreos en el método de Nyström), se ha llevado a cabo probando dentro de una rejilla establecida que se consideraba *razonable*. Por tanto, no podemos garantizar que los parámetros elegidos sean los *óptimos* y para problemas de grandes magnitudes esta técnica es muy costosa; así pues, una de las líneas de investigación que se desprenden de este trabajo consiste estimar *inteligentemente* cuáles serían buenos valores para α y p. En relación al valor de p, existen varios trabajos recientes que tratan de determinar el número de submuestreos necesarios para el método Nyström y cómo seleccionarlos [12, 36, 43, 51]. Estas aproximaciones no son del todo concluyentes y aún queda un amplio campo de investigación en este sentido.

Aprovechando las ventajas en términos de complejidad temporal que proporciona el algoritmo QPFS para problemas de alta dimensionalidad, se está trabajando en la generación de gran cantidad de variables *aleatorias* a partir de una gramática siguiendo la técnica de *Evolución Gramatical (GE)* [28] basada en algoritmos genéticos para, seguidamente, aplicar QPFS. Con ello se espera que la generación explosiva de variables aleatorias pueda proporcionar combinaciones de las variables originales del problema que proporcionen mejores resultados en términos de acierto de clasificación para predictores simples (lineales).

Bibliografía

[1] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. Lapack: a portable linear algebra library for high-performance computers. In *Supercomputing '90: Proceedings of the 1990 conference on Supercomputing*, pages 2–11, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[2] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[3] Jerzy Bala, J. Huang, Haleh Vafaie, Kenneth Dejong, and Harry Wechsler. Hybrid learning using genetic algorithms and decision trees for pattern classification. In *IJCAI (1)*, pages 719–724, 1995.

[4] Ron Bekkerman, Naftali Tishby, Yoad Winter, Isabelle Guyon, and Andre Elisseeff. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

[5] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J.Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, January 1984.

[6] Rich Caruana, Virginia R. De Sa, Isabelle Guyon, and Andre Elisseeff. Benefitting from the variables that variable selection discards. jmlr, 3: 1245–1264 (this issue. In *Journal of Machine Learning Research, 3:1245–1264 (this issue*, pages 200–3, 2003.

[7] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[9] T.M. Cover. The best two independent measurements are not the two best. *IEEE Trans. Systems, Man, and Cybernetics*, 4:116–117, 1974.

[10] Inderjit S. Dhillon, Subramanyam Mallela, Isabelle Guyon, and Andre Elisseeff. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:2003, 2003.

[11] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *J Bioinform Comput Biol*, 3(2):185–205, April 2005.

[12] Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2005, 2005.

[13] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[14] Shai Fine, Katya Scheinberg, Nello Cristianini, John Shawe-taylor, and Bob Williamson. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

[15] George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

[16] George Forman. Bns feature scaling: an improved representation over tf-idf for svm text classification. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, pages 263–270, New York, NY, USA, 2008. ACM.

[17] Charless Fowlkes, Serge Belongie, and Jitendra Malik. Efficient spatiotemporal grouping using the nyström method. In *In Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 231–238, 2001.

[18] Matt Ginsberg. *Essentials of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[19] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, 1983.

[20] Nicholas I. M. Gould and Philippe L. Toint. A quadratic programming bibliography.

[21] Isabelle Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[22] Mark A. Hall. Correlation-based feature selection for machine learning. Technical report, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 1998.

[23] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *International Congress of Machine Learning*, pages 359–366. Morgan Kaufmann, 2000.

[24] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001.

[25] Jianping Hua, Waibhav D. Tembe, and Edward R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recogn.*, 42(3):409–424, 2009.

[26] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):4–37, January 2000.

[27] George John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *In Machine Learning: Proceedings of the Eleventh International Conference*, San Francisco, 1994.

[28] Kaboudan and Mak. Biologically inspired algorithms for financial modelling: Published by: Springer, a. brabazon and m. oneill, 2006, isbn 3-540-26252-0, $85. *Genetic Programming and Evolvable Machines*, 7(3):79–88, October 2006.

[29] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[30] Pat Langley. Selection of relevant features in machine learning. In *In Proceedings of the AAAI Fall symposium on relevance*, pages 140–144. AAAI Press, 1994.

[31] Tao Li, Chengliang Zhang, and Mitsunori Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429–2437, 2004.

[32] David J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, June 2002.

[33] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[34] G.Ñatsoulis, L. El Ghaoui, G. R. Lanckriet, A. M. Tolley, F. Leroy, S. Dunlea, B. P. Eynon, C. I. Pearson, S. Tugendreich, and K. Jarnagin. Classification of a large microarray data set: algorithm comparison and analysis of drug signatures. *Genome Res*, 15(5):724–736, May 2005.

[35] John Neter and William Wasserman. *Applied Linear Statistical Models*. Richard D. Irwin, INC., 1974.

[36] M. Ouimet and Y. Bengio. Greedy spectral embedding. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 253–260, 2005.

[37] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell*, 27:1226–1238, 2005.

[38] Marina Potaptchik, Levent Tuncel, and Henry Wolkowicz. Large scale portfolio optimization with piecewise linear transaction costs. *Optimization Methods Software*, 23(6):929–952, 2008.

[39] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, September 2007.

[40] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci U S A*, 98(26):15149–15154, December 2001.

[41] Douglas T. Ross, Uwe Scherf, Michael B. Eisen, Charles M. Perou, Christian Rees, Paul Spellman, Vishwanath Iyer, Stefanie S. Jeffrey, Matt Van de Rijn, Mark Waltham, Alexander Pergamenschikov, Jeffrey C. Lee, Deval Lashkari, Dari Shalon, Timothy G. Myers, JohnÑ. Weinstein, David Botstein, and Patrick O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nat Genet*, 24(3):227–235, March 2000.

[42] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[43] Alex J. Smola. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann, 2000.

[44] Berwing A. Turlach and Andreas Weingessel. The quadprog package.

[45] H. Vafaie and K. DeJong. Robust feature selection algorithms. *Proc. 5th Intl. Conf. on Tools with Artifical Intelligence*, pages 356–363, 1993.

[46] Haleh Vafaie and Kenneth De Jong. Genetic algorithms as a tool for feature selection in machine learning. In *in Machine Learning. In Proceedings of the 1992 IEEE Int. Conf. on Tools with AI*, pages 200–204. Society Press, 1992.

[47] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. In *NIPS*, pages 668–674, 2000.

[48] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[49] Lei Xu, Pingfan Yan, and Tong Chang. Best first strategy for feature selection. In *9th International Conference on Pattern Recognition*, volume 2, pages 706–708, 1988.

[50] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. *In Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03)*, 2003.

[51] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved nyström low-rank approximation and error analysis. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1232–1239, New York, NY, USA, 2008. ACM.

[52] Yi Zhang, Chris Ding, and Tao Li. Gene selection algorithm by combining relieff and mrmr. *BMC Genomics*, 9(Suppl 2), 2008.

[53] Shenghuo Zhu, Dingding Wang, and Tao Li. Feature selection for gene expression using model-based entropy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(1), 5555.

Índice de tablas

Índice de figuras