CLASIFICACIÓN EN DOS ETAPAS DE DATOS DE EXPRESIÓN GENÉTICA DE ADN

MARTÍN ENRIQUE RUBIO

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
MAESTRÍA INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
BOGOTA D.C.
2006

Resumen

Con el actual rápido desarrollo de los *microarrays* de expresión genética, que genera una tendencia a aumentar la densidad de integración de los ensayos, se están generando más y más *bytes* de datos de medidas de expresión, pero la obtención de la información adicional de grupos funcionales sigue siendo una labor compleja y dispendiosa, lo que ocasiona que la utilización de algoritmos de clasificación supervisada se limite a una reducida proporción de los datos. Este trabajo propone utilizar una metodología que combina algoritmos no supervisados y supervisados, ambos implementados aplicando el truco del kernel, utilizando los primeros para realizar un modelaje inicial, a partir del cual se apliquen los segundos. La metodología se implementa para clasificación funcional de genes y de tipos de tejido, se validan los resultados con estudios previos, encontrándose resultados comparables con mejorar significativas solo para la clasificación funcional.

Tabla de contenido

Lista de tablas

Lista de figuras

1.Introducción

1.1 Motivación

En las últimas décadas buena parte de las investigaciones en ciencias biológicas han consistido en el estudio del funcionamiento de los seres vivos a nivel molecular y genético, con lo que se han desarrollado novedosas técnicas de extracción de información, capaces de dar pistas sobre el funcionamiento y desarrollo de la vida a este nivel. Uno de estos nuevos desarrollos son los *microarrays*, que se definen básicamente como [20] "una matriz bidimensional de material genético, que permite la automatización simultánea de miles de ensayos encaminados a conocer en profundidad la estructura y funcionamiento de nuestra dotación genética, tanto en los distintos estados de desarrollo como patológicos del paciente". Esta tecnología esta compuesta por *arrays* de ADN, *arrays* de proteína, *arrays* de tejidos y *arrays* de química combinatoria [2]. En particular los *microarrays* de expresión genética de ADN, usados para el estudio de patrones en la expresión genética a lo largo de todo el genoma, permiten a los investigadores en un solo experimento obtener la caracterización de una gran cantidad de genes, lo que finalmente se traduce como gran cantidad de datos esperando a ser procesados.

Los datos de expresión genética obtenidos con *microarrays,* son analizados principalmente utilizando dos acercamientos de aprendizaje computacional, el supervisado y el no supervisado. En el segundo de estos se realiza un modelaje descriptivo que intenta dilucidar patrones con significancia biológica en una colección de medidas de expresión. Por su parte el acercamiento supervisado, intenta crear clasificadores que asignen los datos a grupos funcionales previamente conocidos por lo que requiere información adicional a las medidas de expresión. Con el actual rápido desarrollo de los *microarrays*, que tienden a aumentar la densidad de integración de los ensayos, se están generando más y más *bytes* de datos de medidas de expresión, pero la obtención de la información adicional de grupos funcionales sigue siendo una labor compleja y dispendiosa, lo que ocasiona que la utilización del acercamiento supervisado se limite a una reducida proporción de los datos.

Esta tesis propone utilizar una metodología que combine técnicas no supervisadas y supervisadas, utilizando la primera para realizar un modelaje inicial, a partir del cual se implemente la segunda. Para desarrollar esta metodología se propone la utilización de un algoritmo de clasificación en especial, el algoritmo de support vector machines (SVM). Se piensa en este algoritmo como el principal candidato para desarrollar la metodología, en base a que una de sus características más notables es la de realizar una buena clasificación incluso a partir de una serie de etiquetas no muy confiables, además de que puede detectar ejemplos mal clasificados, proporcionando así, herramientas para mejorar la clasificación inicial.

1.2 Genómica funcional

Las investigaciones biológicas y biomédicas están en medio de una transición importante que esta siendo impulsada por dos factores fundamentales como, el aumento masivo de la información de secuencias de ADN y el desarrollo de tecnologías que explotan su uso. En los últimos años, se ha completado la secuencia de los genomas de más de 60 organismos, con otros 170 o más en progreso. La secuencia del genoma humano ha sido descifrada, en esfuerzos tanto públicos como privados [18].

Por el contrario a los esfuerzos de secuenciación del genoma, la genómica funcional es un proyecto menos específico que está en el medio de un acercamiento general a los problemas. La meta no es simplemente proveer un catalogo de todos los genes e información sobre sus funciones, sino también entender cómo trabajan juntos los componentes para abarcar también el funcionamiento de la célula y organismos. Básicamente la genómica funcional busca contribuir a la elucidación de algunas preguntas fundamentales tales como [12]: ¿Cómo la secuencia exacta del ADN humano difiere entre individuos? ¿Cuáles son las diferencias que

resultan en una enfermedad o predisposición a una enfermedad? ¿Cuál es el papel específico de cada proteína sintetizada por un patógeno bacterial? ¿Cómo colaboran las proteínas para realizar las labores requeridas para la vida? ¿Por qué no todos lo genes están activos (se expresan) en un momento determinado en una célula determinada? ¿Cuáles genes son utilizados bajo una circunstancia determinada?, ¿Cómo la expresión genética diferencial resulta en diferentes tipos de células y tejidos en un organismo multicelular?

Los organismos son complejos y sus genomas pueden ser inmensos (demandando el estudio de un gran número de genes y proteínas) por lo que su análisis requiere nuevas y poderosas tecnologías que están siendo desarrolladas como complemento de las metodologías tradicionales que trabajan a pequeña escala [18]. Ejemplo de una de estas tecnologías aplicadas en la genómica funcional son los denominados microarreglos o arreglos de ADN, que han permitido el análisis simultáneo de la expresión de miles de genes, lo que esta cambiando el modelo reduccionista científico hacia un enfoque más amplio y directo de la interrelación de los componentes que constituyen las células de diferentes organismos, hacia hipótesis dirigidas al entendimiento del funcionamiento global de un sistema celular [21].

1.3 Arreglos de ADN

Los arreglos de ADN se definen como una distribución ordenada de cientos a miles de moléculas de ADN [3]. Esta distribución (con forma de matriz) se realiza comunmente sobre material sólido de diversa índole, principalmente placas de cristal, plástico o nylon, de pocos centímetros cuadrados de superficie. El principio básico de funcionamiento de los arreglos es la hibridación (enlace químico no covalente) entre ácidos nucleicos conocidos que son fijados químicamente a la superficie sólida (en posiciones específicas de la matriz) y fragmentos de ácidos nucleicos complementarios presentes en las muestras en estudio [21], las reacciones de hibridación se ejecutan en paralelo para todas las muestras en un arreglo.

Los arreglos pueden contener ácidos nucleicos fundamentalmente de dos tipos: ADN complementario (ADNc) y oligonucleótidos (filamentos cortos de ácido nucleicos) sintéticos, los que definen dos tipos de arreglos de ADN dominantes. La tecnología de los arreglos de ADN complementario fue desarrollada por grupos en la universidad de Stanford liderados por Patrick Brown y Ronald Davis [24]. El sistema esta basado en la impresión de pequeñas muestras (en el rango de los nanos a los pico litros) de ADNc sobre una superficie de cristal, de 25x76x1 mm, previamente revestida con poli linazas o poli aminas para facilitar la absorción electrostática de las muestras. Dependiendo del tipo de tecnología de impresión utilizada, en un área de un centímetro cuadrado se pueden obtener arreglos con un rango de puntos de 200 a 10000, con el tamaño de cada punto oscilando entre los 500 y los 75 µm. Los ADN complementarios empleados en la construcción de estos arreglos se obtienen a partir de librerías de ADNc (usualmente obtenidas mediante la clonación) que contienen fragmentos representativos de genes en el transcriptoma[1], los cuales se conocen como "secuencias de expresión cortas" (Expresed Séquense Tags - ETSs) y que son identificadas con un número de referencia o acceso de fácil identificación en una base de datos. Los arreglos de oligonucleótidos sintéticos desarrollados por Fodor [8] y comercializados por Affymetrix Inc (Santa Clara, CA) se elaboran en una fase sólida similar a la anterior solo que aquí las muestras, ahora de oligonucleótidos, son directamente sintetizadas en la superficie sólida y la implantación de las muestras se realiza utilizando procesos de fotolitografia extraídos de la fabricación de circuitos integrados.

Para determinar la expresión global de genes por medio de los arreglos, en el caso de ADN complementario, se requiere la obtención de ARN mensajero (ARNm) de las células a estudiar (células con una característica especial o que han sido estimuladas) y de células control (células normales o que no han sido estimuladas). Seguidamente, este ARNm se convierte en ADN complementario (ADNc) por medio de una reacción de trascripción reversa en la que simultáneamente se agregan nucleótidos marcados con fluorocromos tales como la Cyanina-3 (Cy-3) y la Cyanina-5 (Cy-5); alternativamente el marcaje del ADNc se puede realizar con un isótopo radioactivo. En el proceso tradicional, para la muestra control se utiliza la Cy-3 y para la

[1] El termino transcriptoma se refiere al estudio de los perfiles de expresión de todos los genes del genoma.

muestra problema se usa la Cy-5. La Cy-3 emite una fluorescencia verde, mientras que la Cy-5 emite una fluorescencia roja. Este marcaje diferencial permite no sólo localizar las señales fluorescentes en el arreglo, facilitando la identificación de los genes, sino que también permite la cuantificación de las señales correspondientes a cada uno de ellos como una medida directa del grado de expresión del ARNm correspondiente [21].

Luego de adicionar el ADNc marcado a la matriz se debe esperar un tiempo prudencial (toda la noche) para propiciar la hibridización. En el caso de las matrices de vidrio o de plástico, los resultados son analizados en un fluorómetro el cual cuantifica la intensidad de la fluorescencia en cada punto de la matriz, lo que corresponde a un gen. La intensidad para cada fluorocromo reportada por el equipo (el fluorómetro) en cada punto de la matriz es la resultante de la relación Cy5/Cy3 (verde/rojo), la cual se expresa como intensidad de fluorescencia en escala logarítmica y en números absolutos: si ésta es 1 (color amarillo) significa que el gen se expresa de manera similar en ambas muestras (control y problema), si es menor que 1 (color verde) significa que el gen se expresó más en las células control y si la relación es mayor de 1 (color rojo) significa que el gen se sobre expresó en la muestra problema. Convencionalmente se emplea como punto de corte de significancia el valor de dos en la escala logarítmica. Finalmente empleando algoritmos de asociación computacionales y con base en la regulación positiva y/o negativa de los genes bajo estudio, es posible establecer relaciones entre los patrones de expresión de los genes y diversas funciones celulares [21]. El análisis de los datos de arreglos comienza con una imagen escaneada de las intensidades de las fluorescencias. Cada uno de los puntos de muestra establecidos en el arreglo es identificado mediante la utilización de algoritmos de alineación de la grilla inicialmente establecida. Luego las intensidades son convertidas en valores numéricos, que se normalizan, de forma que experimentos de diferentes muestras y diferentes arreglos puedan ser comparados. El resultado final es una matriz de niveles de expresión genética para cada condición. Los datos resultantes pueden ser procesados de diversas formas dependiendo del propósito del experimento [13].

1.4 Tareas en las que se utilizan los datos de expresión genética

Utilizando herramientas de aprendizaje computacional, los datos de expresión genética pueden ser utilizados principalmente para dos tareas: Identificación de genes relacionados funcionalmente [3] y la identificación de tipos de tejido [14].

1.4.1 Descubrimiento y predicción de clases de cáncer mediante el monitoreo de la expresión genética

Uno de los grandes retos en el tratamiento del cáncer resulta ser la utilización de la terapia adecuada para cada uno de los diversos tipos de patología de los tumores, esto con el propósito final de maximizar la eficacia y minimizar la toxicidad. Por lo tanto la búsqueda de mejoras en la distinción de tipos y subtipos de cáncer ha sido un factor primordial en los avances del tratamiento del cáncer [16]. El progreso en la tecnología de arreglos de ADN, esta permitiendo la medición simultanea de la expresión de miles de genes en especimenes clínicos en forma sencilla. Se ha demostrado [16] que utilizando los perfiles de expresión genética se puede lograr una distinción entre ciertos tipos de patologías de una forma relativamente sencilla y con alta precisión. El análisis simultaneo de varios miles de genes y el establecimiento adecuado de una relación entre éstos y alguna condición clínica en particular, requiere que los estudiosos del tema tengan a una colaboración cercana con las personas con experiencia en la producción de modelos a partir de datos.

La clasificación de cáncer mediante el monitoreo de la expresión genética se ha dividido en dos retos fundamentales [16]: el descubrimiento de clases, y la clasificación de las mismas. El descubrimiento de clases se refiere al reconocimiento y definición de tipos de tumor previamente no definidos, esto realizado de forma automática a partir de las mediciones de expresión sin usar, *a priori,* ningún conocimiento biológico o opinión de experto, es decir lo que

se quiere es realizar un análisis exploratorio de los datos que dilucide patrones dentro de los mismos, lo que se acomoda a una tarea de aprendizaje no supervisado. La predicción de clases hace referencia a la asignación de muestras particulares de tumor a clases previamente definidas, las cuales pueden reflejar estados actuales o desarrollos futuros de las mismas. Este último reto consiste básicamente en la construcción de modelos de funciones, basados en datos, que discriminen entre clases. Esta metodología se adapta al concepto de aprender a partir de ejemplos, que es una tarea de aprendizaje supervisado.

Características del experimento

En este caso la matriz de datos de expresión se ensambla de la siguiente forma: Cada una de las muestras de tejido, que se quieren analizar, representa una columna cuyas componentes son los valores de expresión de cierta cantidad (en el orden de los miles) de genes. Con lo que se monta una matriz de n columnas (tejidos) y m filas (genes). En este tipo de experimentos el numero de ejemplos es limitado y muy pequeño en comparación con su dimensionalidad, por lo que el ensayo además de buscar la construcción de un modelo que discrimine entre tipos de tejidos, también busca la identificación de genes cuyas expresiones sean buenos indicadores de diagnóstico.

1.4.2 Agrupamiento y clasificación de genes utilizando datos de expresión genética

Los esfuerzos de secuenciación de los genomas de diferentes organismos han permitido conocer las secuencias de nucleótidos que lo conforman, mas la identificación de las partes de esas secuencias que componen genes, es una labor un poco más complicada y más lo es, la identificación funcional de cada uno de esos genes o su regulación conjunta. Se ha encontrado [11] que grupos de genes con similitudes de su patrón de expresión en un conjunto limitado de experimentos tienen mayor tendencia a presentar un comportamiento biológico similar. El agrupamiento y clasificación de genes dentro de grupos funcionales se efectúa de forma similar al planteado anteriormente para las clases de cáncer. Aquí se emplean mecanismos no supervisados para tomar un gran conjunto de genes de función desconocida y crear subgrupos con una posible funcionalidad en común. Las metodologías supervisadas se utilizan para clasificar genes de función no conocida dentro de grupos funcionales previamente conocidos.

Características del experimento

Se monta el arreglo con cierto número de genes que se quieren caracterizar y se obtienen datos de expresión para m condiciones, sobre cada uno de los n genes de interés. Con esto se genera una matriz de $n \times m$ de datos de expresión genética, en donde cada columna identifica expresiones para un mismo gen en diferentes condiciones de experimentación. El objetivo al analizar estos datos, como ya se menciono, es identificar clases funcionales basándose en su vector de expresión.

1.5 Desarrollo de la tesis

1.5.1 Metodología de dos etapas

El capitulo dos presenta la metodología de clasificación en dos etapas, los desarrollos propuestos para su implementación, aspectos generales de la kernelización de algoritmos y el algoritmo de support vector machines. Además se detalla la implementación del algoritmo de aprendizaje no supervisado para operar en espacios proyectados de muy alta dimensión, se hacen pruebas de funcionamiento del mismo en datos de juguete, a partir de las cuales se proponen estrategias para ajustes y selección de parámetros del mismo.

1.5.2 Implementación y comparación de la metodología tanto para clasificación funcional como para fines de diagnósticos.

En el capitulo 3 se implementa la metodología del capitulo 2 tanto para clasificaron funcional como para fines diagnósticos en diversos conjuntos de datos obtenidos tanto de arrays de oligonucleótidos como de ADN complementario. Se comparan los resultados de los algoritmos de aprendizaje no supervisado propuestos con los algoritmos tradicionales utilizando diferentes medidas de comparación. Además se comparan los resultados de la metodología con los de otros estudios realizados sobre los mismos datos.

2. Metodología de dos etapas

En este capitulo se presenta la estrategia planteada para la clasificación en dos etapas, lo que incluye un desarrollo algorítmico en cada una de ellas así como ajustes y pruebas realizadas antes de la etapas de pruebas en los datos de interés.

2.1 Metodología planteada

Para alcanzar el objetivo de generar un clasificador de datos de expresión genética de ADN se busca seguir una metodología de dos etapas, que aplica estrategias de aprendizaje no supervisado, en la primera de ellas, y supervisado en la segunda. Estudios previos han demostrado que herramientas de *clustering* [11, 16, 14, 1, 3] son capaces de identificar clusters con validez biológica en datos de expresión, es decir identifican clases no conocidas en los datos. Luego estas clases "descubiertas" pueden ser asignadas a nuevos datos mediante la utilización de un clasificador generado a partir de un algoritmo supervisado.

Tomando como base el hecho de que en la actualidad metodologías de aprendizaje que hacen uso del truco del kernel, toman algoritmos relativamente simples y potencializan su desempeño, además de que algoritmos de este tipo ya han sido implementados con éxito en datos de expresión, la metodología aquí planteada se basa totalmente en la búsqueda de patrones en los datos en espacios proyectados. Si se tiene en cuenta que al realizar esta potencialización la complejidad en la utilización de los algoritmos se incrementa, se plantea utilizar un algoritmo de clustering relativamente sencillo, que además es de los mas utilizados para realizar en el procesamiento de datos de expresión, como es el algoritmo de K-means.

2.2 Algoritmos

2.2.1 El truco del Kernel

Gran parte de las aplicaciones de los algoritmos de aprendizaje requieren soluciones capaces de distinguir modelos más complejos que los lineales. Expresando esto de otra forma se puede decir que el proceso generador de los datos no se puede modelar simplemente a partir de una combinación lineal de los atributos, sino, que en general se requiere la utilización de características abstractas de los datos.

Una estrategia comúnmente usada en el ámbito del aprendizaje computacional es el cambio de la representación de los datos:

$$X = (x_1, ..., x_n) \mapsto \phi(X) = (\phi_1(X), ..., \phi_n(X)) \tag{2.1}$$

Lo cual es equivalente a realizar un mapeo del espacio de entrada X a un nuevo espacio $F = \{\phi(X) \mid X \in \mathrm{X}\}$.El hecho de que un simple mapeo de los datos hacia otro espacio puede, de gran forma simplificar la tarea de aprendizaje ha sido conocido por largo tiempo en el ámbito del aprendizaje computacional, y ha dado pie a una variedad de técnicas para la selección de la mejor representación de los datos. Las cantidades introducidas para describir los datos son usualmente llamadas características, mientras que las cantidades originales son llamadas atributos. La tarea de escoger la mejor representación para algún propósito en particular es conocida como selección de características, y comúnmente el espacio X se denomina como el de entrada, mientras que $F = \{\phi(X) \mid X \in \mathrm{X}\}$ es llamado el espacio de características.

Un ejemplo de mapeo de un espacio de entrada de dos dimensiones a uno de características con la mismas dos dimensiones, se visualiza en la figura 2.1, donde los datos no pueden ser separados con un modelo lineal en el espacio de entrada pero si en el de características.

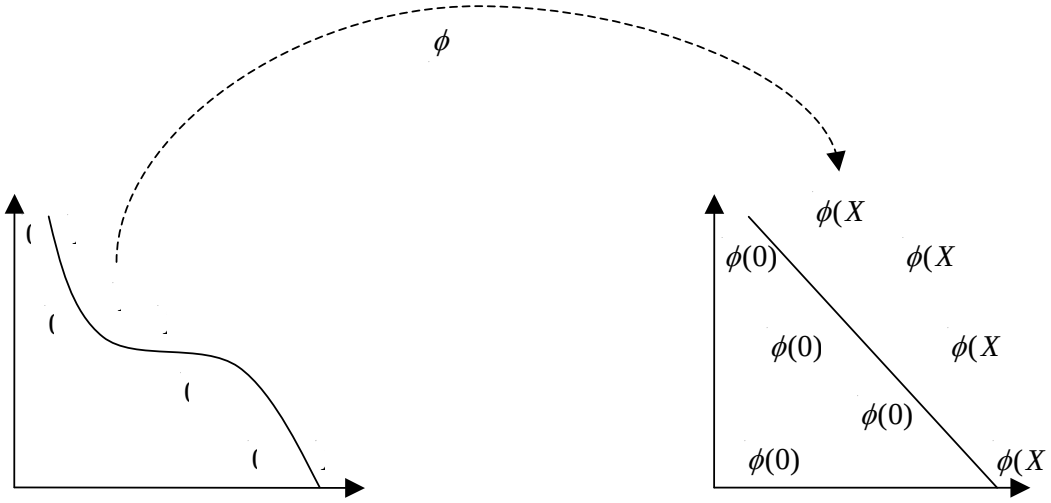

Figura 2.1. Utilización de proyecciones para lograr separación de los datos con modelos lineales

Lo que se quiere mostrar es que mediante mapeos a espacios de muy alta dimensionalidad la utilización de modelos lineales si es adecuada incluso para procesos generadores no lineales. El problema con este tipo de proyecciones es que la gran cantidad de dimensiones de un espacio de características dado, puede convertir el problema en uno intratable computacionalmente al tratar de definir explícitamente el espacio de características. Teniendo en cuenta lo anterior una metodología que defina implícitamente el espacio de características se torna como lo más deseada, por lo que se define lo siguiente:

Kernel: Un kernel es una función k, tal que para todo $X, Z \in \mathrm{X}$

$$k(X, Z) = \langle \phi(X) \cdot \phi(Z) \rangle \tag{2.2}$$

Donde ϕ es una mapeo desde X a un espacio de características y la matriz conformada aplicando la función k a todos los puntos en el conjunto de entrenamiento, denominada matriz de kernels o de Grand, es positiva definida.

Esta función nos calcula el producto punto de los datos en el espacio de características, sin tener que definir explícitamente los datos en ese espacio. Ahora si podemos definir un algoritmo de aprendizaje de forma que solo se empleen funciones de productos puntos entre los datos, se puede operar el algoritmo en un espacio de características, solo con la utilización de un kernel adecuado [25].

Algunos de los kenels más comúnmente utilizados se resumen en la tabla 2.1.

Kernel lineal	$k(x, z) = \langle x \cdot z \rangle$
Kernel Polinomial	$k(x, z) = \left(\langle x \cdot z \rangle + c\right)^d$
Kernel Gaussiano	$k(a, b) = \exp\left(\frac{-\lVert a - b \rVert^2}{2\sigma^2}\right)$
Kernel sigmoide	$k(a, b) = \tanh\left(c\langle a \cdot b \rangle + \theta\right)$

Tabla 2.1. Ejemplos de kernels comúnmente usados

Finalmente se define el truco del kernel como: Dado un algoritmo que se formula en términos de un kernel k, es posible construir un algoritmo alternativo mediante el reemplazo de k por otro kernel k.

2.2.2 K-means [19]

Este algoritmo tiene como meta colocar un conjunto de datos N, en un espacio I – dimensional, dentro de K subconjuntos (clusters) donde cada cluster es parametrizado por un vector m_K denominado su media.

Cada punto de los datos se denota como x_i con $(1 \le i \le N)$ donde N es el número de puntos. Cada uno de los x_i es un vector en el espacio I – dimensional. Se asume que el espacio I es un espacio donde se puede definir una métrica de distancia entre puntos $D(x_i, x_j)$.

Este algoritmo pretende obtener óptimos locales de una función objetivo cuadrática mediante la utilización de una reasignación iterativa de los puntos al cluster más cercano y el cálculo de las distancias de los puntos a cada uno de los clusters como se muestra en el algoritmo 2.1.

1. Seleccionar los K centros iniciales $m_1, m_2, ..., m_K$
2. Asignar cada muestra $x_i \ (1 \le i \le N)$ al centro mas cercano, formando los K clusters; Que se logra realizando calculando el valor de la función indicadora: $\delta(x_i, C_k)$ con $(1 \le k \le K)$ $\delta(x_i, C_k) = \begin{cases} 1 \ D(x_i, m_k) < D(x_i, m_j) \forall j \ne k \\ 0 \qquad\qquad otro \end{cases}$
3. Calcular el nuevo centro m_K para cada cluster C_k $m_K = \frac{1}{\lvert C_k \rvert} \sum_{i=1}^{N} \delta(x_i, C_k)$
4. Repetir 2 y 3 hasta convergencia
5. $C_k, m_K \ (1 \le k \le K)$

Algoritmo 2.1. Algoritmo de clustering *k-means*

Para inicializar el algoritmo (algoritmo 2.1), las K medias $\{m_K\}$ son inicializadas de alguna forma, la más común es con elementos del conjunto tomados aleatoriamente. De aquí en adelante el algoritmo de k-means es un procedimiento iterativo de dos etapas. En la etapa de asignación (2 en el algoritmo 2.1), cada punto i es asignado a la media mas cercana. En la etapa de refresco (3 en el algoritmo 2.1) las medias son ajustadas de forma que concuerden con la media de la muestra de datos de la cual son responsables.

Varias desventajas le son acreditadas a este algoritmo, una de estas consiste en que la asignación de cada elemento del conjunto de datos se realiza a un único centro (algoritmo hard), además de que todos los integrantes de un cluster se toman como iguales dentro de ese cluster, cuando se argumenta que los puntos localizados cerca de la frontera entre dos o mas clusters deberían jugar un rol parcial en la determinación de la localización de todos los clusters a los que ellos podrían ser asignados. Pero en K-means, cada punto de frontera es asignado a un único cluster y tiene igual peso que todos los puntos en el cluster y ninguno en otros clusters [19].

Otra de las desventajas más destacables, es que k-means no permite la división de clusters que no son linealmente separables en el espacio de entrada. Aproximaciones recientes que sugieren que la utilización del truco del kernel, donde los datos son inicialmente mapeados a un espacio de mayor dimensión utilizando una función no lineal, en el cual el algoritmo de k-means

realiza separaciones lineales que corresponde a particiones no lineales en el espacio de entrada.

Para solucionar la primera inquietud se plantea el siguiente algoritmo.

Soft K-means [19]

Buscando adaptar al algoritmo de k-means para manejar asignaciones múltiples de un elemento a diferentes clusters se emplea el soft K-means (algoritmo 2.2).

1. Seleccionar los K centros iniciales $m_1, m_2, ..., m_K$
2. Asignar cada ejemplo, $x_i \ (1 \le i \le N)$, un nivel de pertenencia a cada uno de los centros. Esto se logra calculando el valor de la función de pertenencia: $r(x_i, C_k)$ con $(1 \le k \le K)$ $$r(x_i, C_k) = \frac{\exp(-\beta D(x_i, m_k))}{\sum_{k'} \exp(-\beta D(x_i, m_{k'}))}$$ La sumatoria de las K pertenencias para un punto n es igual a 1
3. Calcular el nuevo centro m_K para cada cluster C_k $$m_K = \frac{1}{R_k} \sum_{i=1}^{N} r(x_i, C_k) x_i \text{ donde } R_k \equiv \sum_n r(x_i, C_k)$$
4. Repetir 2 y 3 hasta convergencia
5. $C_k, m_K, r(x_i, C_k), (1 \le k \le K)$ (Salidas)

Algoritmo 2.2. Algoritmo de clustering soft *k-means*

Para las separaciones no lineales se plantea

Kernelización (Kernel K-means)

Para poder aplicar el truco del kernel hay que expresar el algoritmo 2.1 de forma conveniente, donde el aspecto clave para *K-means* es el calculo de las distancias (euclidiana en este caso) en el nuevo espacio.
Denotando la transformación de x_i como $u_i = \phi(x_i)$. La distancia euclidiana entre u_i y u_j se expresa como:

$$D^2(u_i, u_j) = \left\| \phi(x_i) - \phi(x_j) \right\|^2 = \phi^2(x_i) - 2\phi(x_i) \cdot \phi(x_j) + \phi^2(x_j)$$
$$D^2(u_i, u_j) = k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j) \quad (2.3)$$

Ahora teniendo que lo que se requiere principalmente es calcular la distancia entre puntos y centros se sigue el siguiente procedimiento:

Denotemos al centro de un cluster en el espacio de características como Z_k tal que

$$Z_k = \frac{1}{|C_k|} \sum_{i=1}^{N} \delta(u_i, C_k) u_i \quad (2.4)$$

Donde $\delta(u_i, C_k)$ es una función indicadora. La distancia entre u_i y Z_k se expresa como

$$D^2(U_i, Z_k) = \left\| U_i - \frac{1}{|C_k|} \sum_{i=1}^{N} \delta(u_i, C_k) u_i \right\|^2 \quad (2.5)$$
$$= K(x_i, x_i) + f(x_i, C_k) - g(C_k)$$

Donde

$$f(x_i, C_k) = -\frac{2}{|C_k|}\sum_{j=1}^{N}\delta(u_j, C_k)K(x_i, x_j) \tag{2.6}$$

$$g(C_k) = \frac{1}{|C_k|^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\delta(u_j, C_k)\delta(u_i, C_k)K(x_j, x_i) \tag{2.7}$$

El algoritmo kernelizado (algoritmo 2.3) se obtiene aplicando la ecuación (2.5) al algoritmo 2.1 y se denomina Kernel K-means.

1. Inicializar $\delta(x_i, C_k)$ con valores iniciales, formando k clusters iniciales

K

$r(x_i, C_k) = \frac{\exp\left(-\beta D^2(U_i, Z_k)\right)}{\sum_{k'} \exp\left(-\beta D^2(U_i, Z_{k'})\right)}$ Con D^2 calculado con la ecuación 2.8.
3. Repetir 2 y 3 hasta convergencia
4. $r(x_i, C_k), (1 \le k \le K)$ (Salidas)

Algoritmo 2.4. Kernel sotf K-means.

En el algoritmo 2.4 la inicialización se realiza escogiendo k centros iniciales aleatoriamente del conjunto de datos y calculando $r(x_i, C_k)$ utilizando la ecuación 2.3 para calcular las distancias.

2.2.3 Pruebas de funcionamiento y ajustes de parámetros

Buscando tener un resultado visible del compartimiento de los algoritmos de *clustering* anteriormente presentados, se implementaron los algoritmos sobre conjuntos de datos simples (de juguete) en $\Re^2$. Los cuales son los ampliamente conocidos *dos círculos* y *XOR*.

- Dos círculos

Estos son datos que consisten de 120 puntos distribuidos en dos círculos concéntricos en el espacio bidimensional. Cada círculo corresponde a una clase. En la figura 2.2 se puede apreciar la distribución de los datos en la cual cada clase esta representada por puntos y cruces respectivamente.

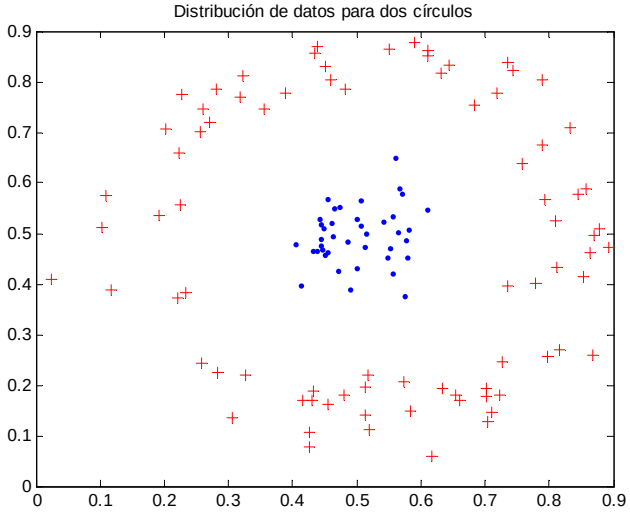

Figura 2.2 Distribución de datos para dos círculos

- XOR

Aquí se tienen nuevamente 120 puntos en un espacio bidimensional. Los datos se distribuyen en cuatro subespacios. Se tiene que los espacios diagonales forman un cluster. En la figura 2.3 se aprecia la distribución con los datos en los diferentes clusters diferenciados por símbolos.

Para evaluar la correcta implementación de los algoritmos se comparan resultados de implementaciones tradicionales de K-means con el kernel k-means utilizando el kernel lineal.

Se observo que ambos presentaron convergencia hacia el mismo punto y obtenían los mismos clusters (figura 2.4 y 2.6). La única diferencia radica en los centros arrojados por cada algoritmo, esto como consecuencia de que los centros en el espacio de características no pueden ser explícitamente expresados y calculan de forma aproximada (paso 5, algoritmo 2.3). Con estos resultados se da un primer paso en la validación de la implementación realizada.

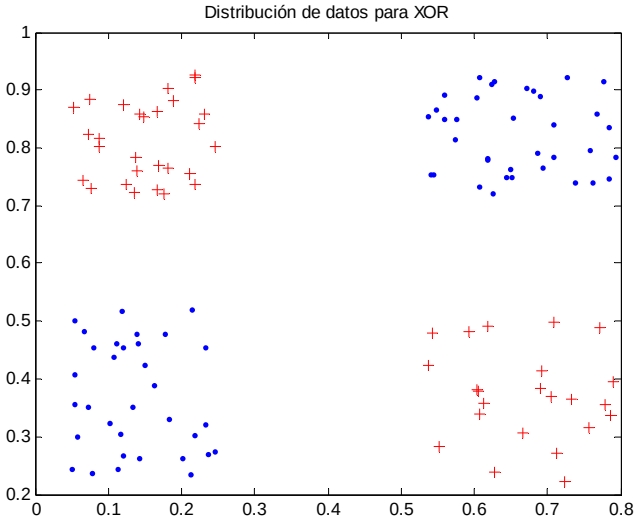

Figura 2.3 Distribución de datos para XOR

Luego se inicio un proceso iterativo en búsqueda del kernel que mejor modelara los datos, encontrándose que con el kernel gaussiano ($k(a,b)=\exp\left(\frac{-\|a-b\|^2}{2\sigma^2}\right)$ con $\sigma=0.11$), se obtenían los mejores resultados para dos círculos (figura 2.5), y con el kernel polinomial, ($k(x,z)=\left(\langle x\cdot z\rangle+c\right)^d$ con $d=2$ y $c=0$) para XOR. En el caso de este ultimo conjunto, los datos son inicialmente normalizados para tener media cero en cada dimensión a través de todos los datos.

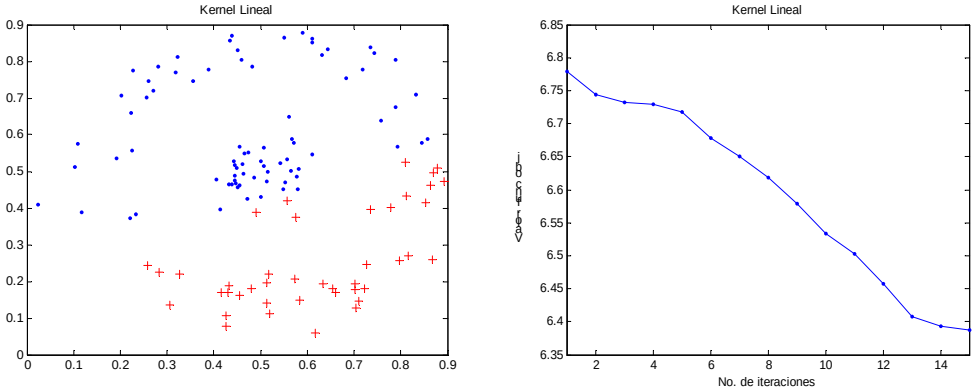

Figura 2.4. Resultados: (Al mejor caso encontrado) Kernel lineal y k-means

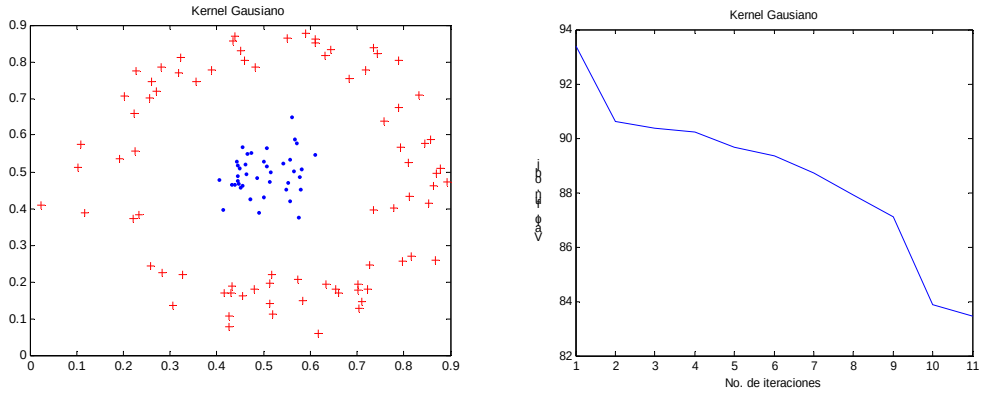

Figura 2.5. Resultados con el kernel gaussiano con $\sigma = 0.11$

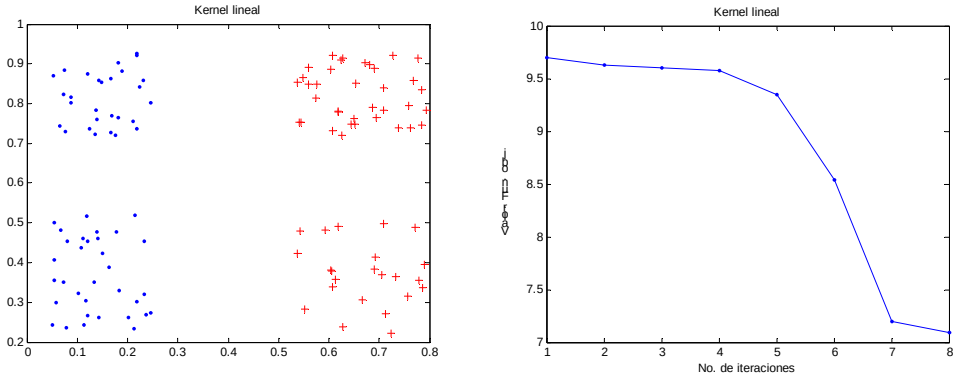

Figura 2.6. Resultados con el kernel lineal para XOR

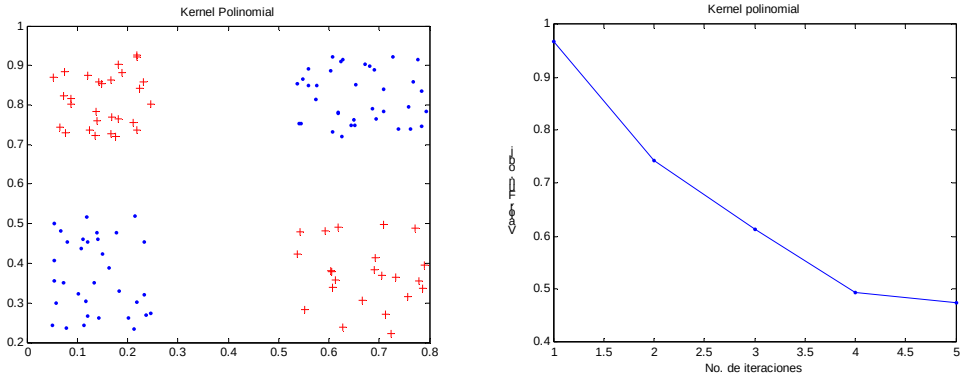

Figura 2.7. Resultados con el kernel polinomial ($d = 2$ y $c = 0$) para XOR

- Observaciones

Se logra identificar la distribución no lineal de los clusters utilizando el algoritmo kernelizado con el costo del aumento en la complejidad del mismo (esperado) ya que se crea la necesidad de escoger el kernel apropiado y sus parámetros de ser necesario. En el caso del gaussiano la selección del parámetro resulto de suma importancia, teniendo que con valores significativamente alejados del "ideal" encontrado, se generan agrupamientos significativamente desviados del modelo planteado. Para el caso del polinomial en XOR, teniendo que intuitivamente al normalizar los datos, el producto de las dimensiones en el espacio de entrada contiene toda la información necesaria para separar los clusters, se piensa en un kernel segundo orden, pero se presenta una dificultad en la escogencia de c, ya que solo utilizando el kenel homogeneo (c=0) se obtienen los resultados buscados.

En ambos casos para valores relativamente altos de σ (que originan valores medios de matriz de kernels cercanos a uno), los agrupamientos obtenidos resultaron ser exactamente los mismos que los logrados con el kernel lineal. Esto se presenta como un posible limite superior para el valor de σ, es decir una ves alcanzada la agrupación del lineal, por mas que se aumente el valor de sigma no se obtendrán resultados diferentes.

La tendencia natural del algoritmo original, de converger a mínimos locales, se conserva, por lo que se debe tener mucho cuidado al escoger los resultados finales, los cuales deben ser el resultado de múltiples ejecuciones del algoritmo, donde finalmente se escoja un punto de convergencia mínimo y consistente en las múltiples ejecuciones.

En cuanto a la utilización del algoritmo soft, los resultados son los mismos que los presentados, para los diferentes kernels, con el agregado del escogencia del valor de β, que presenta el mismo comportamiento que el algoritmo tradicional, donde al cuando su valor tiende a infinito se tiene a un agrupamiento hard. Un comportamiento que cave anotar es que para valores de β suficientemente altos, en ambos conjuntos de datos la convergencia del algoritmo presenta un mejor comportamiento que el algoritmo hard. Esto en el sentido de que tiende a estancarse con menor frecuencia en óptimos locales no deseados.

2.2.4 Support vector machines

En los problemas de clasificación binaria se tienen n experimentos $\{(x_1, y_1),....,(x_n, y_n)\}$, que se denomina como el conjunto de entrenamiento donde x_i es un vector que corresponde a las medidas de expresión del i-esimo experimento o ejemplo. Este vector tiene m componentes y cada componente corresponde a una medida de expresión de un gen o una condición de experimentación, dependiendo del caso, y y_i es una etiqueta binaria, la cual es ± 1. Lo que se quiere es estimar una función multivariada a partir de un conjunto de entrenamiento, de forma que esta función sea capaz de predecir adecuadamente la etiqueta de una nueva muestra.

Una interpretación geométrica de SVM, nos presenta la idea de una cantidad geométrica, llamada el margen, la cual es una medida de que tan bien pueden ser separadas dos clases. SVM realiza tal separación utilizando una función lineal

$$f(x) = w \cdot x = \sum_{i=1}^{m} w_i x_i \tag{2.11}$$

Donde x_i y w_i so las i-esimas componentes de los vectores x y w respectivamente. La etiqueta de un nuevo ejemplo x_{new} es el signo de (2.11), $y_{new} = sign\left[f(x_{new)} \right]$. La frontera de clasificación, es decir todos los valores de x para los cuales $f(x) = 0$, es un hiperplano definido por su vector normal w.

El objetivo de SVM es maximizar la distancia entre el hiperplano y el punto más cercano a este, con la restricción de que todos los puntos de las diferentes clases estén ubicados en lados opuestos del hiperplano. Lo cual se satisface con el siguiente problema de optimización:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$
$$\text{Sujeto a } y_i \left(\langle w_i \cdot x_i \rangle + b \right) \geq 1 \tag{2.12}$$

Donde b es el parámetro que traslada el hiperplano optimo con relación al origen. La distancia desde el hiperplano a los puntos mas cercanos de las dos clases es llamado el margen y se

define como $\frac{1}{\|w\|^2}$. Finalmente, apreciando la ecuación 2.12, lo que hace SVM es encontrar el hiperplano que maximice le margen.

El planteamiento hasta ahora presentado supone que los datos son linealmente separables, pero en la mayoría de los casos prácticos esto no se cumple, por lo que el problema de optimización planteado se modifica agregando variables de holgura [25] con lo que queda:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

Sujeto a $y_i\left(\langle w_i \cdot x_i \rangle + b\right) \geq 1 - \xi_i$

Y si se plantea el problema dual [18] se obtiene:

$$\max_{\alpha}\left(\sum_i \alpha_i \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle\right) \quad \text{(2.13)}$$

Sujeto a

$$0 \leq \alpha_i \geq C$$

$$\sum_i \alpha_i y_i = 0$$

En la etapa de evaluación

$$sign\left[\langle w \cdot x \rangle + b\right] = sign\left(\sum_i \alpha_i y_i \langle x, x_i \rangle + b\right) \quad \text{(2.14)}$$

Como se observa en las ecuaciones 2.13 y 2.14 el algoritmo de SVM tanto en el problema de optimización como en su evaluación esta dado en términos de un kernel (productos puntos de los datos en el espacio de entrada), por lo que se puede aplicar el truco del kernel para operar con proyecciones no lineales de los datos.

3. Implementación y comparación de la metodología tanto para clasificación funcional como para fines diagnósticos

En este capitulo se presentan los resultados de la implementación de la metodología planteada. Para la parte de agrupamientos funcionales de genes de utilizan bases de datos de genes de la levadura, que incluyen tanto experimentos en y fuera de ciclos celulares. La verificación completa de las dos etapas solo se utiliza un solo conjunto de datos, que es para el que se tienen etiquetas con que validar los resultados. También se valida la metodología utilizando conjuntos de datos de cáncer, que incluyen leucemia, colon, próstata y cerebro.

3.1 Descubrimiento y clasificación de grupos funcionales de genes

3.1.1 Datos

Datos	Fuente	Genes	Arreglos
Levadura 1	Eisen et al., 1998	6070	79
Levadura 2	SMD	6112	441
Levadura 3	Cho et al	6220	17
Levadura 4	Chu et al	3020	7

Tabla 3.1. La columna de genes indica el número total de genes incluidos en el arreglo.

Con el fin de que el estudio realizado tenga la validez necesaria se precisa una escogencia adecuada de los datos, de modo que los resultados obtenidos a lo largo del mismo sean realmente prácticos y se puedan comparar con otros de investigaciones similares. Razón por la cual se escogieron múltiples conjuntos de datos, que tienen como característica principal, que han sido utilizados en estudios anteriores resultando en datos con un alto valor comparativo y de análisis tanto numéricos como de la significancía biológica o clínica de los mismos. Las características de cada uno de los conjuntos se especifican a continuación y un resumen es presentado en la tabla 3.1.

- Eisen [11]

Este consiste de los datos de la expresión de 6070 genes medidos en 79 diferentes experimentos de hibridación con arreglos de ADN con condiciones variantes, entre las que se incluyen el *diauxic shift* , el ciclo mitótico de la división de la célula , la esporulación , cambios en temperatura y choques reductivos. Se toma la misma selección de 2467 de estos genes hecha por [11], en la cual se identifican cinco clases funcionales.

- Levadura (*S. cerevisiae*) [SMD]

Este segundo conjunto es una versión expandida del primero, donde los datos fueron generados a partir de un conjunto 441 experimentos y fueron tomados de la base de datos de microarrays de la universidad de *Stanford* (http://genome-www5.stanford.edu). Este conjunto esta conformado por 6112 genes de los cuales se toman 2404 genes como en [19]. Cabe agregar que el mayor número de experimentos para este conjunto de datos se refleja en un aumento en la dimensión del mismo, es decir mayor numero de variables por gen.

- Cho [4]

Este conjunto consiste en 17 medidas de mRNA transcripto en células de levadura sincronizadas en intervalos regulares de tiempo que cubrieron hasta dos ciclos celulares. Se utilizaron arrays de oligonucleótidos y se tienen medidas para 6220 genes en los 15 puntos seleccionados por [4]. Solo se emplean los 2945 genes seleccionados por [4].

- Chu [5]

Aquí se tienen medidas en 7 diferentes instantes durante la meiosis y la formación de esporas. Para genes que comprenden casi el 97% de genoma total, conocido o predicho, de la levadura (Saccharomy cescerevisiae). Se emplean los mismos 3020 utilizados en [27].

3.1.2 Comparación del desempeño

Clustering (Etapa 1)

Teniendo en cuenta que el objetivo del clustering en esta parte es agrupar genes de función similar, se piensa que el mejor método para esa tarea es aquel que tenga la mayor tendencia a agruparlos en conjuntos de función similar. Por lo tanto se escoge una medida de validación, denominada el índice z, que se fundamenta en las bases de datos de anotaciones funcionales actuales, que representan el mejor resumen con el que se puede trabajar, del estado del conocimiento en la actualidad. El índice z se basa en medidas de información mutua entre el resultado de un algoritmo de clustering, las anotaciones hechas en SGD (la base de datos del genoma de la levadura de la universidad de Stanford) y la ontología de los genes desarrollada por el consorcio de ontología de genes. Este índice indica la relación entre el agrupamiento realizado y las anotaciones, relativo a un método de clustering que realiza asignaciones aleatorias de genes a clusters. Entre mayor sea el valor de z indica que el resultado de un algoritmo en particular se aleja más del agrupamiento aleatorio [15].

Las figuras 3.1, 3.2, 3.3 y 3.4 presentan los resultados del algoritmo de kernel k-means para el kernel lineal (kl), el polinomial (kp) y el gaussiano (kg). Se grafica el valor de de z contra el numero de clusters, para $k = 1,2,.....,10$. Cada una de las curvas presentadas es un promedio móvil de tres valores de los datos originales, tal como se realizo en [15].Teniendo en cuenta que k-means presenta [15] resultados significativamente superiores a otras estrategias de clustering tales como Bagclust [9] y CLICK [23], la comparación contra este algoritmo presenta una buena medida frente a el estado del arte actual.

En el conjunto de datos de Cho (figura 3.1), todos los métodos presentan valores de z mayores que 25 para todos los valores de k, indicando que los diversos agrupamientos exhiben un nivel de validez biológica. La utilización de kernels no lineales claramente mejora el desempeño del algoritmo en este conjunto, los tres presentados en la grafica (dos polinomiales y un gaussiano) presentan valores de z superiores para todos los valores k. La tendencia a través de los diferentes valores de k es la misma para todos, con valores picos entre k=5 y k=9 (ver tabla 3.1). Los mejores resultados son arrojados utilizando el polinomial de orden 3 y el gaussiano con $\sigma = 2$, que presentan curvas similares, así como valores picos cercanos y agrupados entre 5 y 6 clusters, que es la tendencia general para los diferentes métodos probados.

Método	z	k
K p2	68,1	5
K lineal	47,9	9
K p3	65,9	6
K g3	58,3	8
K g2	**70,9**	**6**
K g1	45	8
K g1,5	53,5	7
K g1,8	60,1	6

Tabla 3.2. Valores absolutos pico de z entre todos los k.

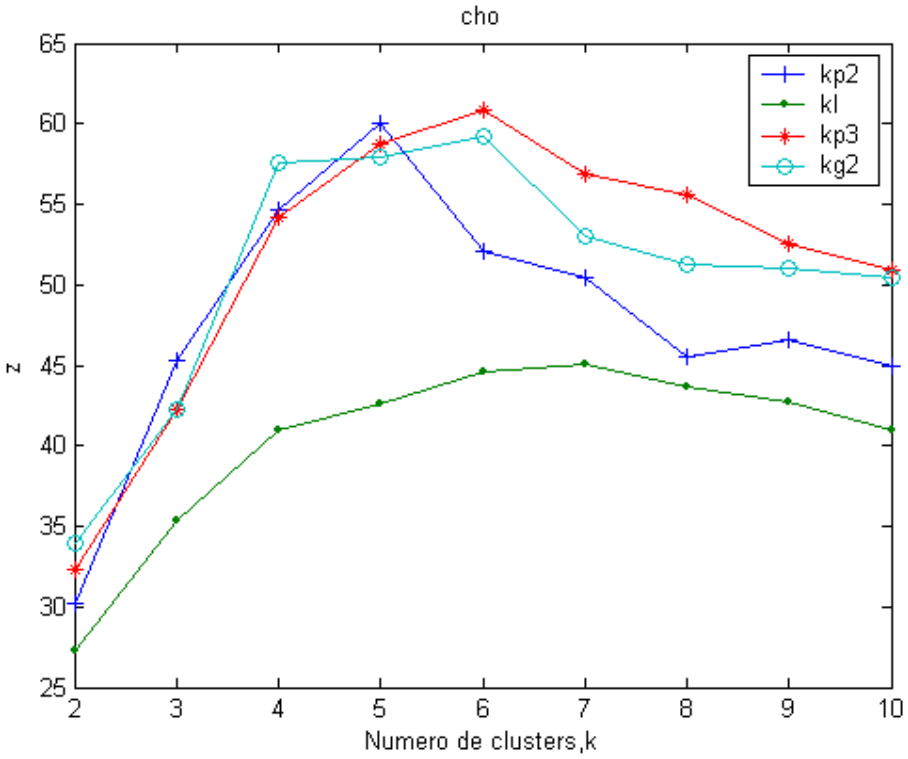

Figura 3.1: Curvas de k contra z para los diferentes kernels en los datos de Cho.

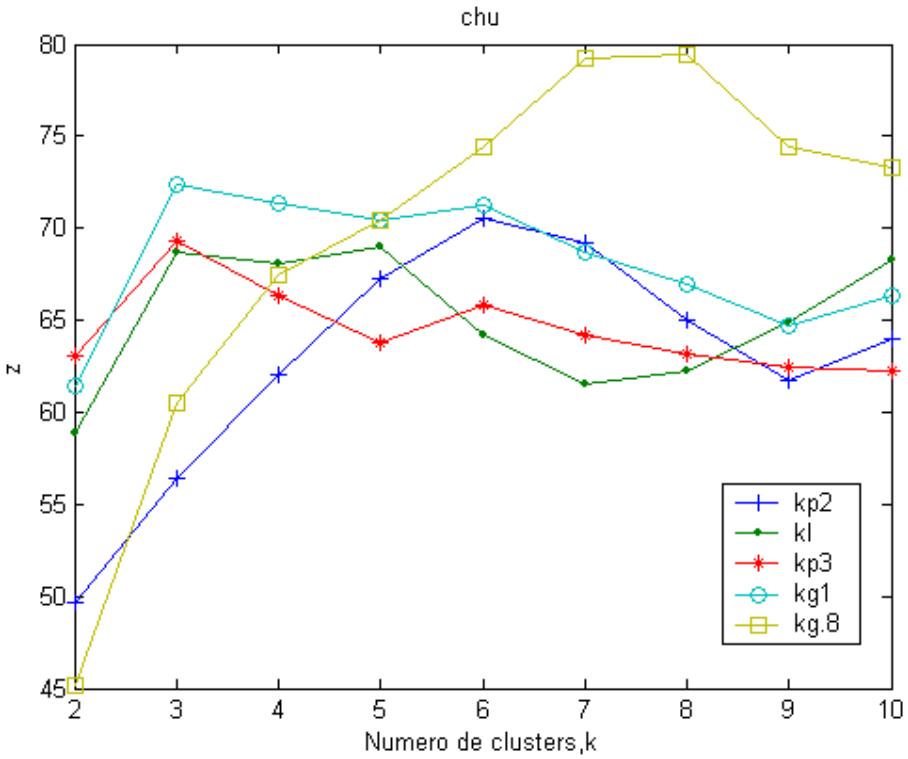

Figura 3.2. Comparación de los resultados de clustering para diferentes kernels en los datos de Chu.

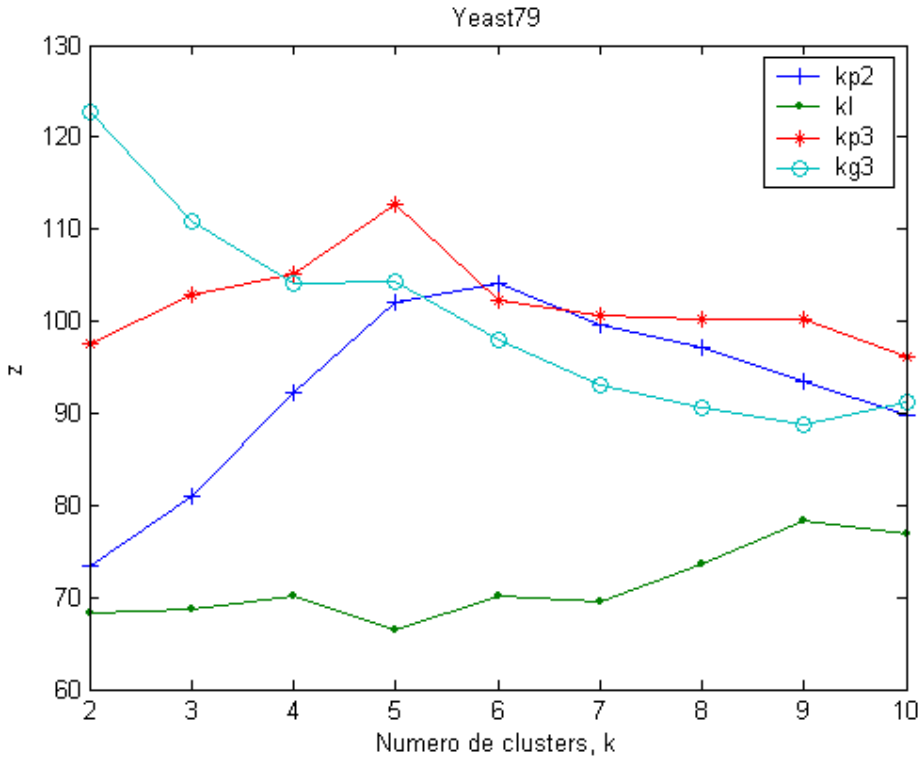

Figura 3.3. Comparación de resultados para los datos de Eisen.

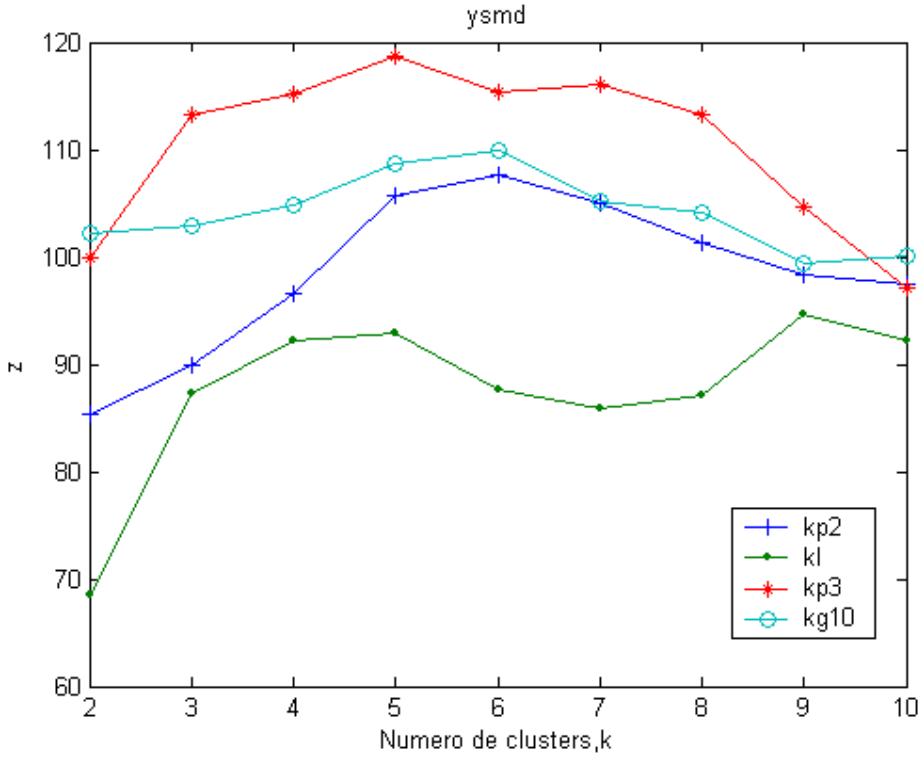

Figura 3.4. Comparación de resultados para los datos de SMD.

Los resultados en los datos de Chu, presentan una visible diferencia frente a los otros conjuntos de datos, como se puede ver en la figura 3.2. Para este caso los kernel polinomiales

utilizados no revelan mejoras significativas, y a pesar de que su valor pico es superior al del lineal, no esta lo suficientemente alejado, lo que teniendo en cuenta las características del índice z se podrían considerar como de desempeño muy similar. Mejoras mas visibles se obtienen con el kernel gaussiano que presenta una leve tendencia a estar por encima del lineal para los diversos valores de k, con $\sigma = 1$, y un valor pico significativamente superior al utilizar $\sigma = 0.8$ (ver tabla 3.3).

Método	z	k
K p2	74,3	7
K lineal	71,7	12
K p3	76,2	4
K g3	70,6	8
K g2	73,2	3
K g1	75,6	3
K g0,5	73,3	5
K g0,8	**85,6**	**8**

Tabla 3.3. Resultados de los valores absolutos picos de z para varios kernels, en los datos de Chu.

Los otros dos conjuntos de datos presentan resultados similares entre si. En ambos casos obtienen mejoras significativas con la no linealidad, utilizando tanto los kernels polinomiales como los gaussianos. Como se reporta en las tablas 3.4 y 3.5 los valores picos de z están por encima de las 100 unidades y hay una tendencia marcada a un mejor desempeño del polinomial de orden 3 con respecto al de segundo orden.

Como se percibe en los resultados para los diferentes conjuntos de datos, la tendencia general es que para $k = 2$ se presenten valores no muy altos de z, por lo que el funcionamiento obtenido para el conjunto de datos de *Eisen* (figura 3.3 y tabla 3.4), donde al utilizar el kernel gaussiano con $\sigma = 3$, se presenta un pico altamente pronunciado en el valor de z para $k = 2$, se torna como un caso patológico. Ahora, para indagar a fondo en la patología presentada se realiza un análisis mas profundo del caso presentado.

Método	z	k
K p2	109	6
K lineal	78	10
K p3	123	5
K g3	**150**	**2**

Tabla 3.4. Valores absolutos pico de z para los datos de Eisen.

Método	z	k
K p2	110	6
K lineal	105	5
K p3	**128**	**5**
K g3	65,2	6
K g10	114	6
K g15	115	6
K g20	112	10
K g0,8	127	3

Tabla 3.5. Valores absolutos pico de z para los datos de SMD.

Este análisis consiste en hallar las categorías funcionales enriquecidas identificadas en cada cluster. Estas categorías son definidas por el consorcio de ontología genética, las mismas utilizadas en el cálculo de z. El nivel de enriquecimiento de de cada categoría en cada uno de los clusters es calculado por su p-value. Para calcular este p-value se utiliza la distribución hipergeometrica que fue utilizada en [8] y en [17]. Esto con el fin de obtener la probabilidad de observar el número de genes de una categoría del GO en particular, dentro de cada cluster si

se utiliza un agrupamiento aleatorio. Valores pequeños del p-value indican que los genes pertenecientes a la categoría funcional enriquecida son biológicamente significativos dentro del cluster dado. Utilizando el mismo acercamiento propuesto en [17] solo se reportan clases funcionales con p-values menores que 5×10^{-7}.

Cluster	Categoría GO	Numero GO	p-value	Agrupamiento total
C1	biopolymer metabolism	GO:0043283	6,29E-15	31,89
	DNA binding	GO:0003677	9,50E-08	8,18
	response to endogenous stimulus	GO:0009719	6,30E-10	7,42
	cell proliferation	GO:0008283	4,85E-23	20,16
	M phase	GO:0000279	1,05E-13	7,48
	cell cycle	GO:0007049	8,89E-23	17,29
	DNA metabolism	GO:0006259	6,20E-16	15,94
	DNA strand elongation	GO:0006271	6,73E-08	2,19
	DNA replication and chromosome cycle	GO:0000067	6,86E-16	8,43
	modification-dependent protein catabolism	GO:0019941	1,60E-10	5,48
C2	cytoplasm organization and biogenesis	GO:0007028	2,67E-09	6,64
	group transfer coenzyme metabolism	GO:0006752	4,92E-07	1,71
	ribosome assembly	GO:0042255	4,31E-11	2,81
	structural constituent of ribosome	GO:0003735	6,32E-38	12,81
	protein biosynthesis	GO:0006412	1,18E-25	21,4
	cellular biosynthesis	GO:0044249	2,49E-24	32,03

Tabla 3.6. Categorías del GO enriquecidas en cada uno de los clusters obtenidos utilizando el kernel lineal con $k=2$, sobre los datos de Eisen. La columna de agrupamiento total indica el porcentaje de genes de cada clase dentro del cluster.

Cluster	Categoría GO	Numero GO	p-value	Agrupamiento total
C1	transferase activity	GO:0016740	1,60E-08	14,92
	cell proliferation	GO:0008283	5,40E-08	13,9
	localization	GO:0051179	1,04E-10	21,13
	nucleobase\, nucleoside	GO:0006139	3,78E-14	33,41
	biopolymer metabolism	GO:0043283	2,60E-10	25,97
	hydrolase activity	GO:0016787	4,54E-08	16,45
C2	cytoplasm organization and biogenesis	GO:0007028	5,44E-09	18,6
	structural constituent of ribosome	GO:0003735	0,00E+00	92,03
	protein biosynthesis	**GO:0006412**	**0,00E+00**	**100**

Tabla 3.7. Categorías del GO enriquecidas en cada uno de los clusters obtenidos utilizando el kernel gaussiano ($\sigma=30$), para $k=2$, sobre los datos de Eisen.

Los resultados del análisis se presentan en la tabla 3.7. Para tener un punto de comparación también se realiza este análisis para el kernel lineal (tabla 3.6) y para el kernel polinomial de orden 3 (tabla 3.8), pero en este caso para $k=10$. La columna de agrupamiento total indica el porcentaje de genes dentro del cluster que pertenecen a la categoría funcional dada. Para el kernel lineal se presentan un número mas amplio de categorías funcionales con enriquecimiento válido dentro de los clusters, pero con bajos porcentajes del total del cluster, lo que indica que, si bien se obtienen agrupamientos con significancia biológica, estos son muy generales, y no implican una identificación funcional clara. Ahora para la patología en estudio (caso del kernel gaussiano) se agrupan en general menos categorías con alta validez, pero en términos de porcentuales el rango de integrantes por cluster es mas alto en términos de escala, sobresaliendo el caso del segundo cluster, donde el 92.03% y el 100% de los genes dentro del cluster pertenecen a las categorías funcionales GO:0003735 y GO:0044249 respectivamente, lo que es indicativo de un alto nivel de discriminación arrojada este agrupamiento, lo que explica en gran parte el por qué del alto valor de z. Pero teniendo en cuenta que, con mayores valores k (tabla 3.8), también se identifica de forma clara ese cluster (cluster 8), con porcentajes similares de integrantes del mismo perteneciendo a cada categoría funcional,

entonces sigue la interrogante del por qué la marcada diferencia en el valor de z para el caso del kernel gaussiano con $k = 2$. La respuesta esta en recordar que este valor es el resultado de una resta contra resultados de un agrupamiento aleatorio y en el cual para $k = 2$ es ínfimamente probable que identifique el cluster en cuestión, de esto la superioridad en z.

Cluster	Categoría GO	Numero GO	p-value	Agrupamiento Total
C1	microtubule-based process	GO:0007017	2,23E-12	15,7
	structural constituent of cytoskeleton	GO:0005200	1,53E-07	8,26
	sister chromatid segregation	GO:0000819	5,00E-09	7,43
	spindle elongation	GO:0051231	1,17E-09	6,61
	mitotic spindle organization and biogenesis	GO:0007052	6,18E-10	9,09
	DNA replication and chromosome cycle	GO:0000067	4,27E-12	23,31
C3	group transfer coenzyme metabolism	GO:0006752	1,36E-07	4,39
	carbohydrate metabolism	GO:0005975	1,10E-08	12,83
	alcohol metabolism	GO:0006066	1,51E-11	12,16
	coenzyme metabolism	GO:0006732	1,10E-08	8,1
	hexose metabolism	GO:0019318	7,77E-11	8,78
	generation of precursor metabolites and energy	GO:0006091	1,68E-11	16,21
	purine nucleotide biosynthesis	GO:0006164	4,67E-07	5,06
	nucleotide metabolism	GO:0009117	4,71E-07	7,09
	alcohol catabolism	GO:0046164	1,48E-10	6,08
C4	translation factor activity\, nucleic acid binding	GO:0008135	2,43E-12	12
	nucleobase\, nucleoside	GO:0006139	1,34E-11	57,33
	RNA helicase activity	GO:0003724	4,80E-08	8
	ribosomal large subunit biogenesis	GO:0042273	1,50E-10	5,33
	translation initiation factor activity	GO:0003743	1,16E-11	9,33
	DNA-directed RNA polymerase activity	GO:0003899	1,39E-09	8,66
	RNA modification	GO:0009451	9,90E-14	13,33
	rRNA modification	GO:0000154	1,27E-09	5,33
	ribosome biogenesis	GO:0007046	1,10E-33	28,66
	translation	GO:0043037	3,97E-14	19,33
	purine nucleoside monophosphate metabolism	GO:0009126	2,60E-07	4,66
	cytoplasm organization and biogenesis	GO:0007028	1,66E-31	31,33
	RNA metabolism	GO:0016070	9,52E-22	37,33
	processing of 20S pre-Rrna	GO:0030490	4,21E-10	6
	primary metabolism	GO:0044238	7,66E-08	84
C5	generation of precursor metabolites and energy	GO:0006091	1,07E-09	11,44
C8	structural constituent of ribosome	GO:0003735	0,00E+00	86,26
	protein biosynthesis	GO:0006412	0,00E+00	95,42
	cytoplasm organization and biogenesis	GO:0007028	1,59E-08	16,79
C9	cell proliferation	GO:0008283	3,25E-15	31,64
	response to DNA damage stimulus	GO:0006974	2,66E-10	14,34
	DNA metabolism	GO:0006259	9,69E-12	25,31
	meiosis	GO:0007126	1,20E-08	8,01
	telomere maintenance	GO:0000723	3,66E-07	4,64
	fidelity during DNA-dependent DNA replication	GO:0045005	1,23E-08	5,06
	DNA replication	GO:0006260	3,87E-10	11,39
	DNA strand elongation	GO:0006271	1,51E-10	6,75
	biopolymer metabolism	GO:0043283	1,36E-12	44,72
	DNA recombination	GO:0006310	4,63E-07	6,75
	nuclear division	GO:0000280	7,77E-09	12,65
	DNA replication and chromosome cycle	GO:0000067	3,58E-10	14,76

Tabla 3.8. Categorías del GO enriquecidas en cada uno de los clusters obtenidos utilizando el kernel polinomial de orden 3, para $k = 10$, sobre los datos de *Eisen*.

Con este ultimo análisis realizado, también se aprecia el enriquecimiento funcional de los diversos clusters obtenidos para los tres resultados presentados. En los diferentes agrupamientos realizados siempre se encontraron categorías funcionales con alta significancia estadística, pero no en todos los clusters, pero si en la mayoría de ellos con el caso de la tabla 3.8.

Con el algoritmo de soft k-means se obtienen resultados similares a los obtenidos con la versión hard, solo se destaca un aspecto, que se refiere a la diferencia de los agrupamientos entre los algoritmos hard y este caso donde se tienen resultados de clustering con agrupaciones difusas, lo que nos brinda la posibilidad de poder filtrar aquellos genes que muestren un asociación pobre con el cluster al que fueron asignados, lo que si es un aporte adicional a lo ya presentado y vale la pena reportarlo.

El proceso de filtraje de genes a partir de un agrupamiento difuso, ha sido utilizado por [8] utilizando fuzzy c-means, y por [17] con el algoritmo de Gustafson-Kessel. Ahora se quiere mostrar que la aplicación de este procedimiento también es valido para los algoritmos kernelizados. Al igual que en [8] se seleccionan los genes cuyo nivel de pertenencia máximo es superior a la media de este valor a través de todos los genes. Para ilustrar este proceso se utiliza el cluster 4 de la tabla 3.8, y se comparan los porcentajes por clase del agrupamiento con todos los genes contra el agrupamiento con los genes filtrados. El resultado se aprecia en la tabla 3.9, donde a pesar de disminuir en numero total de genes dentro del cluster el porcentaje de genes pertenecientes a la mayoría de las clases aumenta, lo que es indicativo de que los genes asociados a una clase funcional en particular, presentan una asociación mas fuerte al cluster, lo que al final indica una relación biológica de los valores de pertenencia arrojados por el algoritmo.

Cluster	Categoría GO	Agrupamiento total	Agrupamiento parcial
C4	translation factor activity\, nucleic acid binding	12	14,51
	nucleobase\, nucleoside\	57,33	61,29
	RNA helicase activity	8	9,67
	ribosomal large subunit biogenesis	5,33	5,64
	translation initiation factor activity	9,33	11,29
	DNA-directed RNA polymerase activity	8,66	9,67
	RNA modification	13,33	14,51
	rRNA modification	5,33	6,45
	ribosome biogenesis	28,66	33,06
	translation	19,33	20,16
	cytoplasm organization and biogenesis	31,33	36,29
	RNA metabolism	37,33	41,12
	primary metabolism	84	86,29

Tabla 3.9. Categorías del GO enriquecidas en el clusters 4 de la tabla 3.8, en la columna de agrupadito parcial se coloca el porcentaje corregido al eliminar los genes con bajo nivel de pertenencia.

Se observa una marcada mejoría en los resultados del clustering, para todos los métodos, al utilizar conjunto de datos con más variables (experimentos) por gen. Para los datos de *Cho* y *Chu* se obtienen máximos en el valor de z en un rango de 70 a 80 unidades, mientras que para los conjuntos mas complejos los resultados se mueven entre 120 y 150 unidades, lo cual es indicativo de que utilizar medidas de expresión en un mayor rango de condiciones genera mejores agrupamiento con respecto a características funcionales, lo cual se percibe como algo razonable y además a sido sugerido en otros estudios [11, 15].
Los conjuntos de *Cho* y *Chu* que representan medidas para un solo tipo de procedimiento, ciclo celular y esporulación respectivamente, donde las diferencias entre cada experimento en ambos caso radica en el tiempo, es decir se toman medidas en diferentes intervalos de tiempo

durante la ocurrencia de un proceso (ciclo celular o esporulación). De los resultados se revela que las correlaciones entre las medidas para diferentes instantes de tiempo proporcionan información que mejora el agrupamiento funcional de los genes sólo para el caso de medidas en ciclo celular (*Cho*), lo que podría ser indicativo de que un modelamiento autoregresivo del ciclo celular puede proporcionar mayores distinciones funcionales. No así en el caso de *Chu*, a pesar de que los datos también son generados a forma de serie temporal.

En términos generales se observa qué la utilización de proyecciones no lineales de los datos mejora el agrupamiento funcional de los genes, pero como es común en este tipo de acercamiento se debe tener muy presente los kernels (espacios) seleccionados, así como sus parámetros ya que como se observo en los resultados anteriores no todos las elecciones proporcionan mejoras, incluso en algunos casos se deterioran los resultados (caso del k g1.5 en la tabla 1 del y K g3 en la tabla 4).

Clasificación (etapa 2)

En esta sección se evalúa qué tan buen modelaje predictivo se puede lograr a partir de un modelo descriptivo inicial de los datos. Se sigue la metodología planteada en el capitulo anterior, ahora aplicada en el conjunto de datos de eisen, utilizando las siguientes 6 clases [3].

- Clase 1: Ciclo del acido tricarboxílico.

También conocida como el ciclo de Krebs, Los genes en este grupo codifican enzimas que rompen por oxidación el pirubato (producido a partir de glucosa). Esta es un ciclo clave para la producción de energía para la célula inicialmente en la forma de NADH y es importante también para la producción de intermediarios en la biosíntesis de aminoácidos y otros componentes.

- Clase 2: Complejos en la cadena de respiración.

Estos complejos llevan a cabo la reacción de oxidación – reducción que captura la energía presente en el NADH a través del transporte de electrones y la síntesis quimiosmótica de ATP. Estos incluyen el complejo NADH deidrogenasa, el complejo citocromo b – c y el complejo citocromo oxidasa, todos embebidos en la membrana mitocondrial.

- Clase 3: Proteínas ribosomales citoplasmáticas.

Esta es una clase de proteínas requeridas para fabricar el ribosoma, este es un complejo de proteínas ARN en el citoplasma, codificado por mARN. Esta categoría no incluye genes del ribosoma mitocondrial.

- Clase 4: Proteasoma.

Esta compuesta de proteínas que abarcan un complejo responsable de la degradación general de proteína y otros aspectos específicos del procesamiento de proteínas. La proteasoma usa *ubiquitin*, un pequeño péptido que marca una proteína que va a ser degradada.

- Clase 5: Histonas.

Estos interactúan con el ADN a fin de formar nucleosomas los cuales en conjunción con otras proteínas forman la cromatina de la célula.

- Clase 6: Hélice jiro hélice (Helix – turn – helix).

Esta no constituye una clase funcional, es incluida solo como un control al igual que en [4] teniendo en cuenta de que no hay razón para creer que los miembros de esta clase tengan patrones de expresión similares, por lo que se espera que ningún clasificador pueda aprender a reconocer los miembros de esta clase utilizando en la medidas de expresión.

Para evaluar la fiabilidad de los resultados se realizan validaciones de los agrupamientos realizados contra las etiquetas ya conocidas (denominadas gold). Lo que se quiere es tener un indicador que muestre qué tan bien el modelo es capaz de reconocer los ejemplos positivos y negativos del conjunto. De acuerdo a los resultados del modelo (clustering) y a las etiquetas (clase 1 a 6) dadas, cada gen puede ser clasificado de la siguiente forma: Falso positivo (FP), que son los genes en los que el modelo indica que pertenecen a cierta clase pero su etiqueta dada no lo asocia a ese grupo; los falsos negativos (FN) son aquellos para los que el modelo indica que no pertenecen a cierta clase pero su etiqueta dice que si; los verdaderos positivos

(TP), son los que el modelo asigna dentro de la misma clase de la que indica su etiqueta; finalmente los verdaderos negativos (TN) son aquellos que tanto el modelo como su etiqueta indican que no pertenece a cierta clase. Teniendo en consideración la naturaleza de la distribución de las etiquetas en este conjunto de datos, donde se tienen muy pocos ejemplos positivos por clase, se utiliza un índice que da mayor importancia a los errores del tipo FN [3], para evaluar los agrupamiento modelados.

$$\cos to = fp + 2\,fn$$

Donde fp es el nuecero de falsos positivos y fn el de falsos negativos.

Clase	Método	FP	FN	TP	TN	Costo
1	KL	35	9	8	2415	53
	KP2	36	10	7	2414	56
	KP3	17	10	7	2433	37
	KG3	33	7	10	2417	47
2	KL	27	14	16	2410	55
	KP2	26	13	17	2411	52
	KP3	13	19	11	2424	51
	KG3	32	19	11	2405	70
3	KL	11	15	106	2335	41
	KP2	14	20	101	2332	54
	KP3	4	29	92	2342	62
	KG3	7	22	99	2339	51
4	KL	18	5	30	2414	28
	KP2	9	6	29	2423	21
	KP3	6	14	21	2426	34
	KG3	1	13	22	2431	27
5	KL	22	2	9	2434	26
	KP2	0	2	9	2456	4
	KP3	0	2	9	2456	4
	KG3	26	2	9	2430	30
6	KL	44	14	2	2404	72
	KP2	36	13	3	2415	62
	KP3	24	14	2	2429	52
	KG3	39	14	2	2412	67

Tabla 3.10. Resultado del modelaje inicial (clustering) para los datos de Eisen

Para obtener las etiquetas iniciales se implementan los diferentes métodos de clustering ya presentados. Teniendo en cuenta que cada clase tiene un pequeño conjunto de datos en comparación con el número total de ejemplos, se toman valores de k altos, en el orden de 40 a 50, y cada clase se compara con cada cluster asignándolo al cluster con menor costo como el representante dicha clase, los resultados se presentan en la tabla 3.10.

Los resultados no son muy alentadores en este punto. Analizando los datos en la tabla 3.10, en algunos casos se podrían obtener mejores costos asignando a todos los datos a la clase negativa, pero se debe tener presente que lo que se requiere principalmente es identificar los elementos positivos, y en todas las clases a excepción de la sexta (como era de esperarse) se identifican el 50% o más de estos.

Teniendo las etiquetas iniciales, generadas con alguno de los métodos de modelaje inicial, se procede a entrenar un clasificador (SVM), y realizar una comparación de los resultados. Esta evaluación se realiza utilizando validación cruzada, en donde se divide el conjunto de datos en tres partes, se entrena el clasificador con dos y evalúa con la restante, esto se realiza 3 veces de forma que se evalué el funcionamiento para todos los elementos en el conjunto de datos. La comparación se realiza con respecto al algoritmo de SVM, dos árboles de decisión: C4.5 y

MOC; el discriminante lineal de fisher (FLD) y *Parzen windows*, todos estos entrenados con las clases de *Eisen* (gold). Cabe aclarar que las etiquetas obtenidas en la primera etapaza solo se usan para la parte de entrenamiento, la validación se realiza contra las clases de *Eisen*.

Clase	Método	Etiquetas	FP	FN	TP	TN	Costo
1	P2-SVM	Gold	5	8	9	2445	21
	Rbf-SVM	KL	12	11	6	2438	34
	Rbf-SVM	KG3	5	10	7	2445	25
	Parzen	Gold	4	12	5	2446	28
	FLD	Gold	9	10	7	2441	29
	C4,5	Gold	7	17	0	2443	41
	MOC	Gold	3	16	1	2446	35
2	Rbf-SVM	Gold	8	6	24	2429	20
	Rbf-SVM	KL	13	12	18	2424	37
	P2-SVM	KP2	13	13	17	2424	39
	Parzen	Gold	22	10	20	2415	42
	FLD	Gold	10	10	20	2427	30
	C4,5	Gold	18	17	13	2419	52
	MOC	Gold	12	26	4	2425	64
3	Rbf-SVM	Gold	11	4	117	2335	19
	P3-SVM	KL	5	14	107	2341	33
	Rbf-SVM	KG3	10	11	110	2336	32
	Parzen	Gold	6	8	113	2340	22
	FLD	Gold	15	5	116	2331	25
	C4.5	Gold	31	21	100	2315	73
	MOC	Gold	26	26	95	2320	78

Tabla 3.11. Resultados de la validación cruzada de las clases 1,2 y 3. La columna de método indica el clasificador utilizado y las etiquetas denotan con que se entrena el mismo. Los datos para parzen, FLD, C4.5 y MOC, son tomados de [3].

Como se observa en las tablas 3.11 y 3.12, la utilización de la estrategia de dos etapas propuesta en este trabajo, proporciona resultados competitivos con respecto a otros algoritmos de clasificación supervisada con los que se compara. Se evidencia una superioridad en el desempeño del algoritmo de SVM para todas las clases válidas (de la 1 a la 5). La utilización de etiquetas ruidosas como en este caso, proporciona una estrategia que supera claramente a los árboles de clasificación y que es competitiva con los algoritmos de FLD y Parzen. Como era de esperarse no llega a alcanzar resultados del todo igualables, a cuando se utilizan etiquetas menos ruidosas para entrenar SVM.

En este conjunto de datos la estrategia no iguala en forma general el rendimiento de SVM, pero el hecho de poder encontrar resultados competitivos con otras estrategias, sin tener un conocimiento inicial de las clases, es bastante alentador. Los resultados generales muestran, que para obtener buenos rendimientos se necesita una muestra lo suficientemente representativa de ambas clases, y que de ser así, en la segunda etapa se será capaz de identificarlas a pesar del ruido que se tenga. Un caso que muestra esta viabilidad es la clase 5, donde para el caso del kernel lineal se tienen 31 ejemplos positivos, 22 de los cuales no concuerdan con los resultados de las combinación SVM – gold, pero los otros nueve son precisamente los que ese clasificador discrimina como positivos, y son suficiente estos para obtener resultados en clasificación iguales a SVM.

De las observaciones realizadas se deduce una funcionalidad de la metodología planteada un poco más concreta. Se puede decir que la primera parte busca inferir el modelo de proceso generador de los datos, agrupa genes. La segunda parte más que generar un clasificador, valida los resultados iniciales y los depura. La estrategia a escoger dependería de lo que se quiere hacer, si se pretende hallar simplemente agrupamientos funcionales, se finaliza el proceso en la depuración, que se puede realizar utilizando validación cruzada, ya no con las clases previamente conocidas, como se presento aquí, sino que se realiza la validación con las

etiquetas creadas en la fase 1. Los datos que se indiquen como mal clasificados se excluyen del conjunto, o se les cambia la etiqueta dependiendo de la aplicación especifica, luego estos datos son utilizados para entrenar un clasificador si así se quiere.

Clase	Método	Etiquetas	FP	FN	TP	TN	Costo
4	Rbf-SVM	Gold	1	5	30	2431	11
	Rbf-SVM	KL	2	6	29	2430	14
	Rbf-SVM	KP2	1	7	28	2431	15
	Parzen	Gold	21	5	30	2411	31
	FLD	Gold	7	12	23	2425	31
	C4,5	Gold	17	10	25	2415	37
	MOC	Gold	10	17	18	2422	44
5	Rbf-SVM	Gold	0	2	9	2456	4
	Rbf-SVM	KL	0	2	9	2456	4
	P2-SVM	KP2	0	2	9	2456	4
	Parzen	Gold	2	3	8	2454	8
	FLD	Gold	0	3	8	2456	6
	C4,5	Gold	2	2	9	2454	6
	MOC	Gold	2	5	6	2454	12
6	P2-SVM	Gold	2	16	0	2449	34
	P2-SVM	KL	24	16	0	2427	56
	P2-SVM	KP3	11	16	0	2440	43
	Parzen	Gold	14	16	0	2437	46
	FLD	Gold	14	16	0	2437	46
	C4,5	Gold	2	16	0	2449	34
	MOC	Gold	6	16	0	2445	38

Tabla 3.12. . Resultados de la validación cruzada de las clases 4,5 y 6.

3.2 Descubrimiento y predicción de clases de cáncer

3.2.1 Datos

Leucemia

Este conjunto de datos contiene niveles de expresión genética de 72 pacientes divididos entre los que sufren de leucemia linfoblástica (ALL, 47 ejemplos) y los que sufren de leucemia mieloide (AML, 25 casos). Los datos fueron obtenidos utilizando microarrays de oligonucleótidos de Affimetrix. Una descripción mas detallada se encuentra en [16]; y están disponibles en [28]. Se sigue exactamente el mismo protocolo planteado en [10], donde los datos son preprocesados, primero aplicándoles un proceso de filtraje, luego realizando una transformación logarítmica y una estandarización de modo que al final se tienen medidas de expresión para 3571 genes por paciente.

Colon

En este conjunto se tienen niveles de expresión para 40 tumores y 22 tejidos de colon normal. Se tienen 6500 genes humanos por muestra, que son medidos utilizando la tecnología de Affimetrix. Se trabaja con una selección de 2000 genes con la máxima intensidad mínima a través de todas las muestras, tal como fue hecha en [1]. Los datos están disponibles en [6]. Aquí también se preprocesan los datos. Estandarizándolos de forma que cada muestra de tejido tenga media cero y varianza uno a través de los genes

Próstata

Consiste de medidas de expresión para 52 tumores de próstata y 50 muestras de tejido de próstata normal, obtenidos también utilizando la tecnología de Affimetrix. Se preprocesan los datos al igual que en [26], con lo que se obtiene niveles de expresión para 6034 genes. Luego se aplica transformación logarítmica de base 10 y normalización a media cero y varianza uno. Los datos están disponibles en [28].

Cerebro

Este conjunto consiste de medidas de expresión para 42 casos, que son presentados en [22]. Esta dividido en 5 clases de tumores del sistema nervioso central, así: 10 meduloblastomas, 10 gliomas malignos, 10 rabdoides – teratoides (AT/RTs) atípicos, 8 tumores neuroectodérmicos primitivos (PNETs) y 4 del cerebelo humano. Los datos se crean utilizando la tecnología de affimetrix y están disponibles al público en [28]. Los datos se preprocesaron siguiendo el procedimiento explicado en la información suplementaría de [22], donde después del filtraje quedan medidas de expresión para 5597 genes por muestra.

3.2.2 Análisis de resultados

La aplicación de la metodología en esta parte es más inmediata, teniendo presente que son conjuntos divididos en pocas clases no sobrelapantes y excluyentes. En la primera etapa se toman los valores de k como iguales al número de clases, a priori, que se conocen de los datos. Se asigna la clase que mejor se ajuste a los datos en términos del la tasa se error obtenida.

Para la segunda etapa se aplica validación cruzada al igual que en la sección anterior, con la diferencia de que en este caso, por el menor numero de ejemplos por conjunto, se aplica dejando por fuera solo un ejemplo (*leave one out* [25]) entrenando el clasificador con el resto de los datos.

Método	FP	FN	TP	TN	ER
KL	7	7	40	18	19,44
KP2	7	7	40	18	19,44
KP3	7	7	40	18	19,44
Kg30	7	7	40	18	19,44

Tabla 3.13. Resultados de los agrupamientos iniciales para los datos de leucemia

Método	Etiquetas	FP	FN	TP	TN	ER
L-SVM	Gold	7	6	41	18	18,06
L-SVM	KL	7	6	41	18	18,06

Tabla 3.14. Resultados en validación cruzada para los datos de leucemia, teniendo que las etiquetas generadas con los diferentes kernels son iguales solo se presentan resultados para una de ellas.

Se encuentra que para los datos de leucemia, los diferentes kernels utilizados obtienen las mismas agrupaciones, con una tasa de error (ER) exactamente igual, como se muestra en la tabla 3.13. Con el kernel gaussiano se puede obtener agrupaciones diferentes para diversos valores de σ, pero estas son de inferior desempeño a las reportadas en la tabla 3.13. Los resultados de la clasificación (tabla 3.14) evidencian que las etiquetas generadas mediante el clustering brindan un igual desempeño al obtenido cuando se utilizan las etiquetas gold.

Método	FP	FN	TP	TN	ER
KL	5	4	18	35	14,52
KP2	5	4	18	35	14,52
KP3	5	2	20	35	11,29
Kg30	5	2	20	35	11,29

Tabla 3.15. Resultados de los agrupamientos iniciales para los datos de colon.

Método	Etiquetas	FP	FN	TP	TN	ER
L-SVM	Gold	3	3	19	37	9,68
L-SVM	KL	5	4	18	35	14,52
L-SVM	KP3	5	2	20	35	11,29

Tabla 3.16. Resultados de validación cruzada para los datos de colon.

Para cáncer de colon los resultados de la primera etapa (tabla 3.15) si se muestran disímiles al utilizar diferente clusters, en general se obtiene dos agrupamientos diferentes, aunque algo similares, el obtenido con el KL y el KP2 y el obtenido con el KP3 y Kg30. Este ultimo muestra mejores resultados, con dos errores (FN) menos que la primera, donde cabe apuntar que los 5 ejemplos asignados como falsos positivos son siempre los mismos, y los dos FN en KP3 Y Kg30 están incluidos en los cuatro de KL y KP2.

La segunda fase muestra que en la agrupación inicial existe una estructura valida en los datos, teniendo que la validación cruzada verifica la asignación inicial. Ahora los resultados son altamente competitivos en comparación a los que utilizan las etiquetas gold, donde se encuentran diferencias de uno y dos errores por encima del gold, ver tabla 3.16.

Método	FP	FN	TP	TN	ER
KL	15	28	22	37	42,16
KP2	16	27	23	36	42,16
KP3	19	25	25	33	43,14
Kg20	21	18	32	31	38,24
Kg19	29	12	23	38	40,20

Tabla 3.17. Comparaciones de las etiquetas iniciales para los datos de próstata.

Método	Etiquetas	FP	FN	TP	TN	ER
L-SVM	Gold	6	2	48	46	7,84
L-SVM	KL	15	28	22	37	42,16
L-SVM	KP2	16	27	23	36	42,16
L-SVM	KP3	19	25	25	33	43,14
L-SVM	KG20	20	21	29	32	40,20
L-SVM	KG19	29	12	38	23	40,20

Tabla 3.18. Resultados de la fase 2 para los datos de próstata.

El siguiente conjunto evaluado, el de los datos de próstata, muestra los resultados menos alentadores. Aquí se presentan las agrupaciones más disímiles entre métodos, como puede apreciarse en la tabla 3.17. Se obtiene el mejor desempeño al utilizar el kernel gaussiano con $\sigma = 20$, con una tasa de error de casi 4 puntos porcentuales menos que el lineal. En la segunda fase los pobres resultados de la primera se mantienen, los modelos iniciales son ratificados por el clasificador en la validación cruzada. Al comparar los resultados en la tabla 3.17 con los de la 3.18, es muy poco o nada lo que se aporta con la utilización del clasificador. Por lo que en última instancia el modelo gold se muestra altamente alejado de los obtenidos con la metodología de dos etapas.

Clase	FP	FN	TP	TN	ER
1	2	1	9	30	7,14
2	0	4	6	32	9,52
3	3	0	10	29	7,14
4	1	0	4	37	2,38
5	3	4	4	31	16,67

Tabla 3.19. Resultados del algoritmo de clustering utilizando el kernel lineal (los mismos resultados se obtienen para los polinomiales de orden 2 y 3) para los datos de cáncer de cerebro.

Para los datos de cáncer de cerebro al utilizar los kernels polinomiales se obtienen los mismos agrupamientos que con el lineal, tal como se presenta en la tabla 3.19, donde se obtienen en total 9 errores. Con el kernel gaussiano se mejora un poco al reducir en dos el número total de errores (tabla 3.20). En general los dos agrupamientos encontrados (tablas 3.19 y 3.20) son bastante similares en cuanto a desempeño, la diferencia radica básicamente en el agrupamiento que se hace de las clases 1 y 3, que es donde se obtienen las mejoras.

Clase	FP	FN	TP	TN	ER
1	2	0	10	30	4,76
2	0	4	6	32	9,52
3	0	0	10	32	0
4	1	0	4	37	2,38
5	4	3	5	30	16,67

Tabla 3.20. Resultados del algoritmo de clustering utilizando el kenel gaussiano con $\sigma = 30$ para los datos de cáncer de cerebro.

Para clasificación se presentan las matrices de confusión en la tabla 3.21, para los tres casos, se obtienen tasas de error de 14.3%, 23.8% y 21.4%, utilizando las etiquetas originales, el kernel lineal, y el gaussiano respectivamente.

Etiquetas	Clase	Modelo				
		1	2	3	4	5
Gold	1	10	0	0	0	0
	2	0	9	1	0	0
	3	0	0	10	0	0
	4	0	0	0	3	1
	5	2	1	1	0	4
KL	1	10	0	0	0	0
	2	0	6	1	0	3
	3	0	0	10	0	0
	4	0	0	0	4	0
	5	2	0	3	1	2
Kg30	1	10	0	0	0	0
	2	0	6	1	0	3
	3	0	0	9	0	1
	4	0	0	0	4	0
	5	2	0	1	1	4

Tabla 3.21. Matrices de confusión para los resultados de clasificación de los datos de cáncer de cerebro.

Totalizando los resultados se encuentra que la metodología de dos etapas presenta resultados competitivos en tres de cuatro conjuntos evaluados, la utilización de kernels no lineales mejoran la el desempeño en tres casos, aunque la mejora es pequeña como para poder afirmar que las características no lineales en estos datos aporten información adicional que mejore significativamente la clasificación. Se podrían explicar estos resultado haciendo un paralelo con otros estudios en donde se concluye que la utilización de kernels no lineales no mejora significativamente el desempeño del algoritmo de SVM para este tipo de datos (concretamente en los datos de próstata y colon) [14]. Esto puede indicar que, como se hizo en los otros estudios [14], también aquí se puede lazar la predicción de que al aumentar el número de ejemplos disponibles por clase, la utilización de kernels de mayor complejidad o específicamente diseñados [25], si será de utilidad.

Ahora, buscamos un poco la explicación del porque la metodología arroja resultados tan pobres para los datos de próstata. Si se piensa en la primera etapa; en términos generales el numero de errores es grande, pero finalmente no lo suficientemente alto, como para no considerar que se logran identificar un numero significativo de representantes de cada clase, por lo que se

esperaría, a la luz de los resultados de la sección 3.12, que al correr la segunda etapa se logren filtrar buena parte de los errores, pero ese no es el caso, por el contrario la segunda etapa ratifica la primera, lo que nos dice que la estructura encontrada si es válida, algo que no es de esperarse siendo que conocemos previamente la estructura (deseada) de los datos y sabemos que la encontrada se aleja considerablemente de ésta.

Con la aplicación de las variantes soft, se obtienen los mismos resultados hasta aquí presentados para los diferentes conjuntos de datos, en rangos altos de valores de β. Conforme se hace más pequeño β, y se hacen más difusos los resultados, el redimiendo tiende a disminuir. Cuando se intentó filtrar los datos basándose en sus valores de pertenencia se encontró que se excluyen los errores, pero en la misma proporción también se excluyen datos bien clasificados.

Cabe agregar que el comportamiento evidenciado en el capitulo 2 (con los datos de juguete) por el algoritmo de kernel K-means al utilizar el kernel gaussiano, el cual se presentó como un limite superior para el valor de sigma se presenta nuevamente para los diferentes conjuntos de datos de esta sección que nuevamente encuentra las mismas agrupaciones que la variante lineal para valores altos del parámetro σ.

4. Conclusiones y sugerencias

4.1 Que deja este trabajo

Los resultados obtenidos dan evidencia para afirmar que la motivación principal de este estudio, de utilizar un algoritmo de aprendizaje supervisado a partir de unos conjuntos de datos sin etiquetas, encuentra una solución valida en la metodología planteada. En términos generales se pueden encontrar modelos que pueden ser validados por un clasificador como SVM, y alcanzar resultados en clasificación que son competitivos con el estado del arte. Ahora a fin de alcázar mejores resultados, como se predecía desde un principio, se necesita una primera fase fuerte. Aquí se planteo la utilización del truco del kernel para fortalecer los algoritmos de clustering tradicionales, los resultados mostraron que esto es posible, más exactamente mostraron que en conjuntos donde un algoritmo como SVM, incrementa su desempeño con kernels no lineales, las variantes de K-means también son capaces de mejorar su rendimiento utilizando el mismo acercamiento. Cuando SVM no necesita la utilización de kernels complejos la estrategia de clustering tampoco, o por lo menos la mejora no es tan significativa como en otros casos. Se predice que, la necesidad de la utilización de kernels más complejos se genera conforme los datos así lo van requiriendo.

La metodología planteada, mas que ser lo que finalmente dicta el titulo de este trabajo, un clasificador, se presenta mas bien como un explorador que busca dependiendo del caso, grupos funcionales de genes o diferentes características en los tejidos de un grupo de pacientes. Resulta al fin de cuentas una herramienta de modelaje descriptivo, que utiliza algoritmos supervisados para depurar sus resultados, y que con estos puede, de ser necesario y si así lo permiten los datos, generar un buen clasificador.

4.2 Trabajo futuro

La utilización del truco del kernel en el algoritmo de K-means, demostró ser de utilidad, en conjuntos de datos aquí presentados, utilizando la distancia euclidiana como métrica de comparación entre datos. También resulta de interés La utilización de otras métricas de distancia, como la correlación de pearson, la manhattan, la de Hausdorff o en general las diferentes variaciones de la norma – n, y la comparación del desempeño y aplicabilidad de cada una de ellas tal como se hizo en [15] para la variante lineal de k-means. Asi mismo se torna atrayente la exploración en la utilización de kernels condicionalmente positivos definidos como métricas de distancia directas en los espacios de características. O la kernelización de otros algoritmos como SOM.

En términos generales, siguiendo el enfoque de los problemas en la obtención de etiquetas confiables para la gran cantidad de datos generados en la actualidad, herramientas más directas de clasificación que hacen uso de conjuntos de entrenamiento con etiquetas para un numero limitado de elementos dentro del mismo, se sugieren como aproximaciones mas directas para la generación de clasificadores que aprovechan la gran cantidad de datos disponibles, requiriendo solo un numero limitado de etiquetas por clase.

La aplicaciones que involucran datos de expresión no son las únicas donde metodologías como las aquí planteadas pueden ser útiles. También es de gran interés en el campo de la bioinformática la búsqueda de patrones en conjuntos de datos de secuencias, a los que se podría aplicar un enfoque similar al aquí presentado, necesitando solo definir adicionalmente para su implementación medidas de similitud o disimilitud entre ejemplos.

Alternativamente, la utilización de kernels diseñados específicamente para el tipo de datos analizados puede resultar de gran utilidad [25]. Un ejemplo que puede servir de guía es la utilización de un modelo generativo [29], el cual incorpora un conocimiento a priori de los datos en la estructura del espacio de características utilizado.

Referencias

[1] Alon,U., Barkai,N., Notterman,D.A., Gish,K., Ybarra,S., Mack,D. and Levine,A.J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, **96**, 6745–6750.

[2] Baldi, P., Brunak, S. (2001) Bioinformatics, The machine learning approach. MIT Press, Cambridge.

[3] Brown, M., Grudy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., M. Ares, J. y Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97, 262-267.

[4] Cho, R.J., Cambell, M.J., Winzeler, E.a., Steiinmetz, L., Conway, A., Wodicka, L., y otros. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell 2: 65-73.

[5] Chu, S., DeRisi, J., Eisen, M., Mulholland, D., Botstein, P., Brown, P.O., Herskowitz, I. (1998). The Transcriptional Program of Sporulation in Budding Yeast. Science 282: 699-705.

[6] Colorectal Cancer Microarray Research [http://microarray.princeton.edu/oncology/]

[7] Cristianini, N., Shawe-Taylor, J. (2000) An introduction to Support Vector Machines. Cambridge University Press.

[8] Dembele, D., Kastner, P. (2003). Fuzzy c-means metho for clustering microarray data. Bioinformatics 19:973-980.

[9] Dudoit, S., Fridlyand, J. (2003). Bagging to imprive the accuracy of a clustering procedure. Bioinformatics 19: 1090-1099.

[10] Dudoit, S., Fridlyand, J., Speed T: (2002) Comparison of discrimination methods for the classification of tumors using gene expression data**.** *J Am Stat Assoc*, 97:77-87.

[11] Eisen, M., Spellman,P ., Brown,P. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. PNAS, 95, 14863 – 14868.

[12] Fiels, S., Kohara, Y., Lockhart, D. (1999) Functional genomics *Proc. Natl. Acad. Sci. USA,96, 8825-8826.*

[13] Fodor, S., Rava, R. Huang, X., Pease, A., Colmes, C., y Adams, C. (1993) Multiplexed biochemical assys with biological chips. *Nature* 364:555-556.

[14] Furey,T., Cristianini,N., Duffy,N., Bednarski,D., Schummer,M. and Haussler,D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16, 906–914.

[15] Gibbons, F., Roth. F. (2002). Judging the quality of gene expression-based clusteringmethos using gene annotation. Genome Res. 12: 1574-1581.

[16] Golub,T., Slonim,D., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J., Coller,H., oh,M., Downing,J., Caligiuri,M. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531–537.

[17] Kin, D., Lee, K. (2005). Detecting clusters of different geometrical shapes in microarray gene expression data. Bioinformatics 21, 1927-1934.

[18] Lockhart, D, Kimberly, J, McConnell, C. (2002) An introduction to DNA microarrays. En: Lin, Simon M.(Author). *Methods of Microarray Data Analysis II*. Hingham, MA, USA: Kluwer Academic Publishers.

[19] MacKay, D. (2003) Information Theory, Inference, and Learning Algorithms. Cambridge University Press

[20] Microarrays y Biochips de ADN, Informe de vigilancia tecnológica. (2002) *GENOMA ESPAÑA/CIBT-FGUAM*

[21] Orrego, J, Patiño, P, Franco, J. (2003) Micromatrices de ADN en el estudio del sistema inmune Revista de la Asociación Colombiana de Alergia Asma e Inmunología. Volumen 12. Número 3.

[22] Pomeroy S, Tamayo P, Gaasenbeek M, Sturla L, Angelo M, McLaughlin M, Kim J, Goumnerova L, Black P, Lau C, *et al.*: Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature* 2002, 415:436-442.

[23] Saran, R., Maron-katz, A., Shamir, R. (2003). Click and expander: a system for clustering and visualizing gene expresión data. Bioinformatics 19, 1787:1799.

[24] Schena, M., Shalon, D.,Davis, RW., Brown, PO. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*. 270:467-470.

[25] Schölkopf, B., Smola A. (2001) Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge.

[26] Singh D, Febbo P, Ross K, Jackson D, Manola J, Ladd C, Tamayo P, Renshaw A, D'Amico A, Richie J, *et al.*:(2002). Gene expression correlates of clinical prostate cancer behavior**.** *Cancer Cell* , 1:203-209.

[27] Stafford, W., Qin,J. y Lewis, D. (2003) Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19, 2097–92104.

[28] Whitehead Institute Center for Genomic Research: cancer genomics [http://www-genome.wi.mit.edu/cancer]

[29]