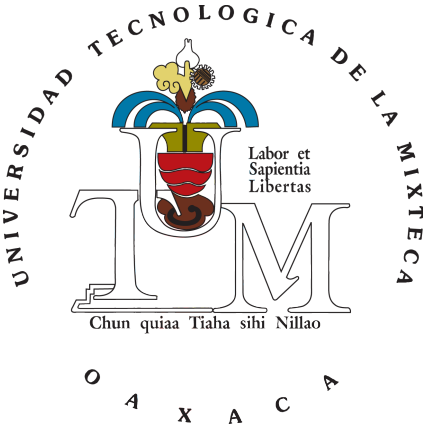


Universidad Tecnológica de la Mixteca

División de Estudios de Posgrado

Maestría en Tecnologías de Cómputo Aplicado

Análisis exploratorio de receptores acoplados a proteínas G mediante métodos de reducción de dimensionalidad

Tesis

Para obtener el grado de
Maestro en Tecnologías de Cómputo Aplicado

Presenta:
Ing. Omar Castellanos Santa Cruz

Director de tesis:
Dr. Raúl Cruz Barbosa
Co-director:
Dr. Johan Van Horebeek

Huajuapan de León, Oaxaca, México Diciembre de 2016

Dedicatoria

Quiero que seas eterna,
ajena a las reglas del tiempo y a sus condenas...

y es que para mi eres eterna,
pase lo que pase y sea lo que sea,
sé que no se extinguera jamás tu vela,
sé que aquí estarás aunque este mundo muera.

Agradecimientos

A la Universidad Tecnólogica de la Mixteca por abrirme sus puertas y ser mi segundo hogar durante el transcurso de mis estudios.

Al director del presente trabajo, por la orientación, motivación y esclarecimiento de ideas.

Al MC Pérez Ramos por facilitarme las transformaciones, que son la base de este trabajo.

Al cuerpo acádemico de la divisíon de estudio de posgrado, por los retos y el conocimiento brindado.

Índice general

Índice de figuras

RESUMEN

Los receptores acoplados a proteínas G (G protein-coupled receptors, GPCRs) constituyen una super-familia de receptores celulares. Dichas proteínas son actores clave en la comunicación celular debido a que su función es transducir una amplia gama de señales y regular funciones celulares. Debido a que existen diversas enfermedades asociadas al mal funcionamiento de este tipo de proteínas, los GPCRs, son el objeto de estudio de distintas instituciones de investigación y de empresas farmacéuticas.

La predicción de la función asociada a una proteína de forma automática mediante herramientas computacionales es uno de los objetivos de la bioinformática. Para poder procesar la información biológica de forma computacional el primer paso es modelar los datos. El problema del modelado de los datos de su forma biológica (alfabética) a su forma computacional (numérica) se ha atacado siguiendo el paradigma tradicional de estrutura-función, el cual indica que para cada proteína sólo corresponde una función. Este paradigma concuerda con el concepto tradicional de aprendizaje máquina, en el cual sólo es posible asociar una clase a cada muestra del conjunto de datos. En constraste, hace algunos años se descubrió que existen proteínas capaces de realizar más de una función, lo cual es un proceso bio-químico complejo que ha permitido encontrar una cantidad limitada de proteínas con dichas características. Por otro lado, recientemente ha surgido un nuevo enfoque de aprendizaje automático denominado aprendizaje multi-etiqueta, el cual es capaz de asociar instancias contenidas en el conjunto de datos a más de una clase. Dicho aprendizaje permite modelar problemas de una forma más realista que el aprendizaje tradicional.

En este trabajo, los datos para ser analizados son secuencias de GPCRs de la clase C. Cada secuencia de proteína representada en forma biológica es sometida a cuatro transformaciones diferentes para obtener vectores de características de longitud fija con información numérica. Dichos vectores resultantes son de alta dimensionalidad lo cual incrementa la complejidad de su manipulación y visualización. Es por ello que es necesario reducir el número de variables de los vectores transformados. Existen diferentes métodos para realizar esta tarea, en esta investigación se comparan métodos tradicionales y se propone un nuevo enfoque de modelado de los datos, el cual supone que los vectores que representan a las proteínas pertenecen a múltiples clases.

Antes de aplicar técnicas de aprendizaje automático se realiza un análisis exploratorio a las secuencias transformadas para observar de forma gráfica su disperción en el espacio

3D, y la correlación por pares de variables, entre otros datos estadísticos. Una vez que se estudia la estructura de los datos, se aplican técnicas de reducción de dimensionalidad y clasificacíon tradicional a éstos y se comparan los resultados con el estado del arte. Después, se obtienen resultados análogos mediante un modelado de las secuencias con un enfoque multi-funcional. También, a la representación multi-funcional de las transformaciones se le aplican técnicas de reducción y clasificación utilizando el enfoque de aprendizaje multi-etiqueta. Finalmente, se analizan, comparan y discuten los resultados de ambos enfoques.

Capítulo 1

Introducción

Normalmente, una computadora realiza una tarea determinada a través de una lista de órdenes a ejecutar, la cual recibe el nombre de algoritmo y este especifica paso a paso cómo realizar dicha tarea. Para esto, buscamos un algoritmo que sea eficiente, esto es, que logre el objetivo deseado y además lo haga optimizando la cantidad de recursos utilizados, comparado con otros algoritmos para resolver la misma tarea [Alpaydin, 2010]. Existen ciertas tareas donde los algoritmos aprenden una función o una regla de correspondencia a partir de un conjunto de ejemplos dado, para posteriormente, usando la función aprendida, poder tomar una decisión cuando se presente una situación nueva (ejemplo desconocido). Estos algoritmos forman parte del campo de Aprendizaje Supervisado, y pertenecen al área de la computación denominada Aprendizaje Automático (del inglés Machine Learning). En la actualidad, existen áreas de la ciencia que utilizan este tipo de algoritmos para resolver problemas específicos, en los cuales se manejan grandes volúmenes de información y donde se desconoce la función que asocia a los ejemplos o elementos del conjunto con su categoría o taxonomía correspondiente. Algunas aplicaciones de estas se encuentran en ciencias médicas, predicción del clima, geociencias, reconocimiento de formas, entre otras.

El manipular grandes cantidades de información hace tedioso el análisis y aumenta el tiempo de respuesta de los algoritmos, esta problemática puede deberse a tres motivos: al número de objetos en el conjunto de datos, a la gran cantidad de variables que describen a cada objeto o a una combinación de ambas situaciones. Una solución a este problema consiste en reducir el número de observaciones o elementos del conjunto de datos a través de prototipos o vectores referencia, asumiendo que estos conservan las propiedades de los vectores o elementos asociados. Otra solución consiste en reducir el número de características (dimensiones) que describen a los objetos en un conjunto menor que conserve la mayor información posible del conjunto original. La reducción de dimensionalidad (RD) se enfoca en reducir el número de variables de cada muestra, su objetivo es una representación más compacta de los datos y el mejoramiento en la exactitud del análisis de datos [Cunningham, 2007] .

Por otro lado, un problema específico que es de gran importancia en la actualidad pertenece al área de Bioinformática y Farmacología, el cual consiste en asociar una secuencia de proteína con su ligando correspondiente. Las representaciones que describen a las se-

cuencias de proteínas se presentan por lo general con vectores de alta dimensionalidad. El objetivo principal de esta tesis es reducir la dimension de la representación de la secuencia, de tal forma que se logre una mejor interpretación de su estructura y asociación con su ligando correspondiente. Para esto, se propone un enfoque de reducción de dimensionalidad suponiendo que una secuencia de proteína, puede estar asociada a más de un ligando. Las secuencias que se estudian son los receptores acoplados a proteínas G (del inglés G protein coupled receptors: GPCRs), que son de gran importancia en el área de farmacología para el desarrollo de medicamentos.

1.1. Planteamiento del problema

La Bioinformática trata de responder preguntas biológicas a través del análisis de grandes cantidades de datos utilizando herramientas computacionales. Uno de los problemas de la Biología Computacional y Bioinformática de gran auge en la actualidad es el estudio de los fármacos y las proteínas sobre los que estos actúan. Al lograr vincular secuencias de proteínas con un ligando específico, mediante técnicas de aprendizaje automático, se ayuda a comprender mejor su estructura y las funciones intracelulares a las que están asociadas [Pin et al., 2003]. Se denomina ligando a las macromoléculas que se unen al centro activo de la proteína para que ésta pueda realizar funciones determinadas. Estas funciones pueden ser de transporte, estimulación o inhibición de reacciones metabólicas. La unión ligando-proteína provoca un cambio conformacional en la proteína. La sustancia activa de los fármacos, la cual es un ligando, se adhiere a las moléculas celulares si su estructura molecular se lo permite. Al adherirse a la célula, lo hace a través de un receptor, existen varios tipos de receptores en la superficie celular los cuales se clasifican de a cuerdo a la función que realizan. Los receptores a los que se adhieren los ligandos generan algún tipo de reacción biológica, Por ejemplo: estimular el apetito, o en otro caso, inhibir el dolor. En este trabajo los datos para ser analizados son secuencias de GPCRs de la clase C. Las cuales son secuencias de proteínas que se encuentra dentro de las células, estas proteínas, que reciben el nombre de proteínas G, están ligadas a un tipo de receptor en la superficie celular. A estos receptores, ligados a proteínas G es a lo que la sustancia activa de los medicamentos se adhiere, desencadenando una respuesta celular. Estas secuencias de proteínas y los ligandos que las activan son el caso de estudio de esta investigación. Para el análisis estructural de estas proteínas las secuencias de aminoácidos que las representan deben ser modeladas de su forma biológica a su forma computacional. El resultado de este modelado es un vector de características de longitud fija, usualmente de alta dimensionalidad. Estos vectores de alta dimensión aumentan la complejidad de manipulación e interpretación de la información que almacenan. Es por ello que es recomendable reducir el número de variables de estos vectores.

Bajo el contexto de análisis exploratorio utilizando reducción de dimensionalidad, identificar y clasificar de forma automática las secuencias de los GPCRs puede realizarse de dos formas diferentes. La primera de ellas es mediante un análisis de visualización, lo cual consiste en aplicar un método de RD que logre reducir la dimensión de los datos a una dimensión inferior, en la cual se puedan representar los datos mediante un recurso gráfico

que manifieste visualmente la relación que existe entre las diferentes secuencias ya reducidas. En otras palabras, la dimensión inferior obtenida permite plasmar el nuevo conjunto de datos en el plano o en el espacio 3D para analizar o interpretar si existe similitud entre los datos. De esta forma es posible definir claramente a que clase pertenece cada muestra, delimitar las fronteras de clase, su distribución y el traslape entre ellas [Gao and Wang, 2006]. La segunda forma es a través de medidas de rendimiento de un clasificador: este proceso se lleva acabo cuando en la etapa de reducción, la dimensión resultante de los datos vuelve complicado el análisis visual, haciéndolo no objetivo. Debido a esto, es necesario auxiliarse de un clasificador.

En el estado del arte existen trabajos relacionados con el problema de clasificación de las diferentes familias de los GPCRs. En [Cruz-Barbosa et al., 2015] se realiza una comparación de métodos semi-supervisados (variantes de GTM y SVM) para la clasificación de subfamilias de GPCRs de la clase C. En [Karchin et al., 2002] utilizan SVM con una función kernel específica para el mismo propósito. [Bécu et al., 2013] utiliza funciones kernel con GTM, PCA y SOM para visualizar conjuntos de datos pertenecientes a la clase C de los GPCRs. De la misma forma se han utilizado otros métodos de RD para la visualización de datos pertenecientes a los GPCRs como lo muestra [Gao and Wang, 2006]. Otros algoritmos se han utilizado para la correcta clasificación de las distintas clases de GPCRs como en [Bakir and Sezerman, 2006, Peng et al., 2010].

En los trabajos citados anteriormente se ha utilizado clasificadores multi-clase con modelos supervisados y semi-supervisados para la clasificación y visualización de datos pertenecientes a secuencias de GPCRs. Todos estos enfoques tratan de asociar secuencias de proteínas a un sólo ligando, esto es, usan un modelo rígido, el cual puede disminuir la exactitud del análisis de los datos. En contraste, hace algunos años al lograr determinar la estructura $3D$ de las proteínas mediante cristalización se descubre una propiedad que es parte integral de la mayoría de las proteínas. Esta propiedad les da la capacidad de modificar su estructura funcional y poder interarticular con más de un ligando, pero debido a la complejidad de este proceso aún no se ha determinado las múltiples funciones asociadas a todas las familias de proteínas existentes. Entonces, lo que se propone en este trabajo es un modelado más realista (modelado suave), donde se asume que una secuencia puede pertenecer a más de una subfamilia. Por lo tanto, se pretende utilizar un enfoque de reducción de dimensionalidad multi-etiqueta para mejorar la visualización de los datos y/o el rendimiento de predicción mediante clasificadores. En consecuencia, al lograr comprender la estructura de las secuencias de proteínas y los ligandos que las activan es posible desarrollar mejores fármacos para el tratamiento de enfermedades.

1.2. Justificación

Los GPCRs son ampliamente investigados en la industria farmacéutica debido a su presencia y participación en gran número de funciones fisiológicas. En la actualidad, existe una gran variedad de fármacos que actúan sobre los GPCRs. La identificación, clasificación y agrupamiento de secuencias de las diferentes familias y subfamilias de los GPCRs ayudan a una mejor comprensión de sus funciones y en el desarrollo de nuevos fármacos.

Por lo tanto, es necesario contar con métodos automáticos para la exploración de los GPCRs que ayuden a discriminar mejor las familias y subfamilias y obtener la correlación entre sus diferentes características. Esta información es de gran importancia para la industria farmacéutica en la creación de mejores antialérgicos, anestésicos, antidepresivos, antipsicóticos, etc. Además, según nuestros conocimientos a la fecha de redacción, en la literatura no se ha realizado el estudio de esta familia de proteínas mediante un modelo suave. Debido a esto, una investigación mediante este enfoque es necesaria para observar el comportamiento de los GPCRs al ser analizados mediante algoritmos de aprendizaje multi-etiqueta.

1.3. Hipótesis

Utilizar un enfoque de reducción de dimensionalidad orientado a etiquetas múltiples permite una mejor discriminación, así como una mayor separabilidad entre las subfamilas de la clase C de los GPCRs.

1.4. Objetivos

1.4.1. Objetivo general

Analizar, explorar e implementar métodos de reducción de dimensionalidad basado en etiquetas múltiples para el problema del modelado de secuencias de GPCRs.

1.4.2. Objetivos específicos

- Investigar en la literatura métodos de reducción de dimensionalidad multi-etiqueta y convencionales.
- Investigar las funciones biológicas de los receptores acoplados a proteínas G y el problema de su modelado computacional.
- Seleccionar e implementar métodos de RD multi-etiqueta y métodos de RD convencionales para modelar el problema de las secuencias de GPCRs.
- Comparar los métodos de RD convencionales con los de métodos RD multi-etiqueta utilizados en el modelado de GPCRs mediante el desempeño de clasificadores y mediante herramientas de visualización gráficas.

1.5. Metas

- Reporte sobre métodos de reducción de dimensionalidad convencionales y multi-etiqueta.

- Reporte sobre las funciones biológicas y el modelado computacional de receptores acoplados a proteínas G.
- Implementación de los métodos de reducción de dimensionalidad multi-etiqueta y convencionales.
- Reporte comparativo de los métodos implementados sobre el modelado de GPCRs de la clase C.

1.6. Trabajos relacionados

Al representar secuencias de proteínas a una forma numérica mediante una transformación, se busca que dicha representación capture la mayor información de la secuencia original, y más aún, que capture las regiones clave dentro de la secuencia alfabética. Esto es, regiones biológicamente significativas que son la base de la función o funciones que realiza. De forma general, se pretende localizar el patrón subyacente de la función asociada de cada subfamilia de proteínas y para la familia en general. Al lograr reconocer la región clave en las secuencias podemos omitir en el análisis las secuencias residuales que no aportan información significativa para la clasificación de las proteínas de acuerdo a su rol o funciones asociadas. Al lograr agrupar a los GPCRs de acuerdo a su tipo o subtipo se logra un mejor diseño de fármacos y una mejor comprensión de los procesos moleculares implicados [Cruz-Barbosa et al., 2015] . En el estado del arte existen trabajos relacionados con el problema de clasificación de la diferentes familias de los GPCRs mediante diferentes métodos [Bakir and Sezerman, 2006, Gao and Wang, 2006, Peng et al., 2010] como son: máquinas de soporte vectorial, redes neuronales artificiales, modelos ocultos de Markov, sistemas difusos, búsqueda de vecinos más cercanos, técnicas de clasificación jerárquica. Para el enfoque de reducción de dimensionalidad se han explorado métodos como: PCA, Kernel PCA, MDS, GTM entre otros.

Específicamente, para la clase C de los GPCRs [Cruz-Barbosa et al., 2013, 2015] utiliza variantes semi-supervisadas de GTM y SVM para el problema de clasificación de estas subfamilias. [König et al., 2013] clasifica dichas subfamilias mediante una SVM, en [Cárdenas et al., 2016] mediante funciones kernel se compara GTM, PCA y SOM con el objetivo de explorar las secuencias en una representación de baja dimensión. En [Cárdenas et al., 2014, 2016] utilizan árboles filogenéticos y GTM para visualizar las secuencias de GPCRs de clase C. [König et al., 2014b, König et al., 2015] analizan esta misma familia mediante un enfoque de detección de etiquetas con ruido, para lo cual utiliza una SVM. En [König et al., 2014a] se utiliza la transformación de secuencias de n-gramas para la selección de características y posteriormente se utiliza una SVM para la clasificación. En [Ramos Pérez, 2016] utiliza aprendizaje profundo a través de una máquina de Boltzman restringida para obtener las representaciones de las secuencias de GPCRS de clase C y las compara con las representaciones tradicionales (AAC, PseAAC, Wavelet-PseAAC y ACC), obteniendo mayor rendimiento mediante la arquitectura profunda propuesta.

Existen algunos trabajos en la literatura que atacan el problema de clasificación de proteínas multifuncionales [Afzal et al., 2015, Sarlin and Peltonen, 2011, Wan et al., 2012].

En [Borroto-Escuela et al., 2011, Fuxe et al., 2014, Wieland and Mittmann, 2003] suponen que los GPCRs son multifuncionales, sin embargo, no se especifica que este supuesto es aplicable a todas las familias y subfamilias de dichas proteínas.

1.7. Metodología

Para afrontar el problema sobre el modelado de receptores acoplados a proteínas G se realiza un estudio detallado sobre el dominio de esta familia de proteínas y los métodos que se han utilizado para abordar este tema en el ámbito computacional. Se investiga en el estado del arte para conocer los trabajos relacionados con reducción de dimensionalidad y clasificación convencional (modelado rígido, asumiendo que una proteína se asocia con un sólo ligando) y multi-etiqueta (modelado suave, asumiendo que una proteína se puede asociar a varios ligandos) de esta familia de proteínas. Para esto se seleccionan e implementan algunos algoritmos de reducción de dimensionalidad convencional. Posteriormente, se le aplica a un conjunto de datos (de una base de datos pública) de receptores acoplados a proteínas G de la clase C, las diferentes técnicas de reducción implementadas. Estos algoritmos extraen las características relevantes que ayuden a asociar adecuadamente a los receptores con su ligando correspondiente. Después se realiza un análisis comparativo, el cual puede ser mediante un análisis de visualización, si la reducción de datos así lo permite, o de forma alternativa, mediante medidas de rendimiento de exactitud de un clasificador. Luego, se implementa un método de reducción de dimensionalidad que utilice un enfoque de etiquetas múltiples, asumiendo que el proceso de construcción de este algoritmo conlleva una investigación de diferentes formas de reducción de este tipo. Finalmente, se comparan los resultados arrojados por los métodos de reducción convencionales y multi-etiqueta antes mencionados con los resultados en el estado del arte.

La redacción de este trabajo se realiza durante el periodo de desarrollo de la tesis.

Capítulo 2

Marco teórico

Es este capítulo se describen los conceptos básicos del aprendizaje automático tradicional y multi-etiqueta. También, se explica brevemente los conceptos asociados al preprocesamiento de datos, reducción de dimensionalidad, clasificación y métricas de rendimiento, tanto para el aprendizaje tradicional como para el aprendizaje multi-etiqueta. Además, se expone el problema del modelado de GPCRs, así como la categorización de las familias de dichas proteínas.

2.1. Introducción

Actualmente la inteligencia artificial (IA) tiene una gran demanda en muchas áreas de la ciencia como son: ciencias sociales, química, biología, robótica, electrónica, computación, entre otras. El reconocimiento de patrones, aprender de la experiencia, tomar decisiones sin intervención humana y clasificar objetos de acuerdo a algún criterio, son tareas de algunas ramas de la IA. Disciplinas como: aprendizaje automático (machine learning, ML), minería de datos (data mining, DM), reconocimiento de patrones (pattern recognition, PR) pertenecen a IA y se dedican a estudiar y generar nuevos métodos que resulten eficientes para los problemas que nos enfrentamos en el mundo real [Puja and Neha, 2013].

Uno de los paradigmas de cómputo en IA es el paradigma de cómputo suave (o flexible), el cual trata de enfrentar los problemas cómo lo hace la mente humana. La mente humana es eficaz cuando se trata de razonar dando soluciones aproximadas y no exactas. Las respuestas humanas a los problemas cotidianos dependen del dominio del problema, razonamiento incierto, la adaptación a un entorno variable y ambigüo. La experiencia otorga la capacidad de hacer frente a la vaguedad inherente, la incertidumbre y a información incompleta de los problemas cotidianos. Las técnicas de soft computing aprovechan la imprecisión, la incertidumbre, la verdad parcial, y la aproximación para lograr viabilidad, robustez y bajo costo en las soluciones a problemas computacionales [Zadeh, 1994].

Por otro lado, las técnicas de hard computing tienen como principal objetivo la precisión, certeza y el rigor matemático. Además, es necesario para poder aplicarlas un modelo analítico previo. Muchos modelos matemáticos son productivos al aplicarse a conjuntos de

datos que cumple ciertos parámetros, pero para otros conjuntos de datos que no cumplen estas condiciones, el mismo modelo puede dar resultados no satisfactorios. En la práctica algunos problemas complejos se encuentran en las áreas como: biología, medicina, ciencias humanas y ciencias de la administración. En dichas áreas los métodos tradicionales basados en análisis matemáticos generalmente no ofrecen los mejores resultados [Kecman, 2001].

De acuerdo al contexto de esta investigación, las técnicas de soft computing se adaptan mejor a los problemas biológicos si los comparamos con las técnicas tradicionales. Estas últimas toman como punto de partida una respuesta lógica formal, que nos restringe a respuestas mutuamente excluyentes. En el área de soft computig los temas que se discuten son problemas relacionados con las siguientes vertientes de la IA [Mitra and Acharya, 2003].

- Sistemas basado en reglas difusas.
- Sistemas genéticos difusos.
- Redes neuronales.
- Agrupación.
- Aprendizaje semi-supervisado.
- Aprendizaje no supervisado.
- Aprendizaje multi-etiqueta y multi-instancia.
- Aprendizaje basado en ensambles.
- Clasificación desbalanceada.
- Clasificación basada en una clase.
- Clasificación con ruido.

Las siguientes secciones tratan sobre el problema de alta dimensionalidad en bioinformática y técnicas de reducción de sus dimensiones basado en aprendizaje tradicional y multi-etiqueta. Como última sección se incluyen los elementos básicos para el entendimiento del modelado y representación de secuencia de GPCRs así como el concepto de multifuncionalidad asociado a este tipo de proteínas.

2.2. Aprendizaje automático

Si observamos un evento n número de veces y notamos que para un determinado conjunto de condiciones iniciales el resultado siempre es el mismo, podemos crear una regla deductiva a base de esa experiencia. Si esta regla no varía en el tiempo podemos formalizarla mediante el método científico. Posteriormente podemos plasmar el proceso determinista

mediante un algoritmo, función, ecuación o ley matemática [Mitchell, 1997]. Desafortunadamente, no siempre es posible encontrar una relación o función explícita que asocie un evento con una determinada reacción. En el área de ciencias de la computación, a este fenómeno se le denomina evento no determinista. Para tratar de resolver este tipo de problemas debemos acudir a un área denominada aprendizaje automático. El Aprendizaje Automático es una disciplina que estudia cómo construir sistemas computacionales con base en el análisis de datos que en forma de ejemplos puede mejorar automáticamente la toma de decisiones mediante la experiencia [Bishop, 2006].

El resultado de ejecutar un algoritmo de ML puede ser expresado como una función que tiene una entrada y genera una salida. La forma precisa de la función se determina durante la fase de entrenamiento (también conocido como fase de aprendizaje) sobre los datos de ejemplo. A grandes rasgos, el entrenamiento es el proceso en el que el algoritmo de ML aprende de los datos de ejemplo. Una vez que extrajo cocimiento y logró generalizarlo es capaz de predecir que salida se obtendrá al someter una nueva muestra al sistema. Por otro lado, si para el problema en cuestión, existe un modelo matemático exacto que obtiene un valor invariante y correcto para cualquier ejemplo, esto quiere decir que no es necesario abordar el problema con técnicas de machine learning [Alpaydin, 2010]. La capacidad de clasificar correctamente los nuevos ejemplos que difieren de los utilizados para el entrenamiento/aprendizaje se conoce como clasificación. Las tareas de clasificación, predicción y reconocimiento son objetivos esenciales en el Reconocimiento de Patrones [Bishop, 2006].

La tarea de clasificación se puede resumir de la siguiente manera. Dado un conjunto de entrenamiento T, se construye un método para predecir, de acuerdo a lo aprendido con los ejemplos de T, la clase asociada a nuevos objetos. Esto es, reconocer alguna situación cuando ésta se presente y tomar una decisión con la estrategia aprendida. A partir de la información en T, se obtiene el modelo de clasificación, mediante un entrenamiento, una vez obtenido este modelo ya es posible categorizar nuevos objetos.

En general, las tareas de un proceso de aprendizaje pueden ser clasificadas en dos categorías: descriptivas y predictivas. El primer enfoque describe el conjunto de datos de una manera resumida y concisa, presentando propiedades generales e interesantes de los datos. Por otro lado, las tareas predictivas construyen uno o varios modelos que realizan inferencia sobre el conjunto de entrenamiento para intentar predecir el comportamiento de nuevos datos. En reconocimiento de patrones nos referimos a estas dos categorías como aprendizaje de tipo supervisado y no supervisado, añadiendo una tercera categoría, el aprendizaje semi-supervisado [Chapelle et al., 2006].

El **aprendizaje supervisado** consiste en categorizar nuevos objetos basados en un conjunto de entrenamiento del cual se conoce la clase de cada objeto. Por otra parte, el aprendizaje no supervisado se basa en conjuntos de entrenamiento en los que se desconoce la clase de los objetos, una de las tareas de éste tipo se lleva a cabo mediante técnicas de agrupamiento (clustering). En el enfoque **no supervisado** se agrupan objetos de acuerdo a una medida de similaridad, con cada grupo más o menos homogéneo y distinto de los demás. El **aprendizaje semi-supervisado** es una combinación de los dos enfoques anteriores. En este tipo de aprendizaje se cuenta con muy pocos ejemplos asociados a una etiqueta de clase y la mayoría de muestras del conjunto es de tipo no supervisado.

2.2.1. Pre-procesamiento de datos

En la mayoría de casos, los datos provienen de diversas fuentes, y puede contener valores impuros (incompletos o inconsistentes), rangos de valores muy dispersos u objetos superfluos (ruido, redundancia) que no son útiles para el proceso de clasificación. Debido a esto es necesario que los conjuntos de datos sean sometidos a una fase de pre-procesamiento para preparar los datos antes de someterlos a cualquier técnica de aprendizaje máquina. Los objetivos que persigue esta fase son: reducir el conjunto de datos (selección de características y de instancias), mejorar la eficiencia del proceso de minería de datos posterior, tratamiento de la información incompleta, eliminación de outliers y detección de posible dependencia entre las variables. Al lograr limpiar los datos durante este proceso se generan un conjunto de datos de calidad, que conducirá a mejores modelos [Bishop, 2006].

En la figura 2.1 podemos observar la arquitectura de un sistema de reconocimiento de patrones donde se observa que la fase de pre-procesamiento es de gran importancia. Esta fase ayuda en la reducción del tiempo de ejecución y la mejora del proceso de aprendizaje. La lista de tareas que se incluyen en esta fase se puede resumir en cuatro aspectos: análisis exploratorio, limpieza, transformación y reducción, no teniendo que aplicarse en este orden (ver figura 2.2).

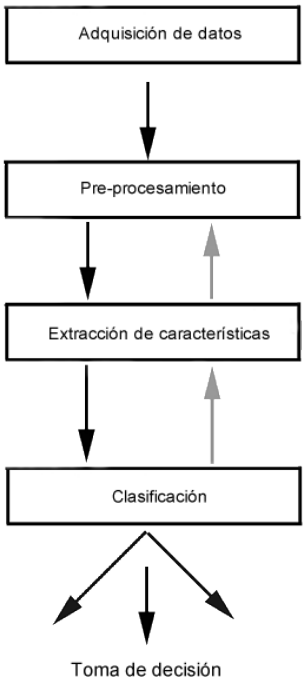


Figura 2.1: Arquitectura de un sistema de reconocimiento de patrones [Duda et al., 2000]

El **análisis exploratorio** tiene como objetivo permitir al investigador familiarizarse con la naturaleza de los datos. Esta parte del pre-procesamiento de datos utiliza generalmente un recurso gráfico para cumplir su labor, sin aplicar técnicas formales de análisis estadístico o de minería de datos. Esto es, durante el análisis exploratorio de los datos se examinan características que se pueden detectar fácilmente mediante algún tipo de gráfi-

co. Algunas de las características son: la distribución de las variables, las relaciones entre pares de variables o relaciones multivariantes, observar si existe una diferencia marcada entre los diferentes grupos del conjunto de datos, la existencia de valores atípicos, los rangos y la existencia de simetría en las diferentes variables. Las relaciones más evidentes que se detectan en esta fase son analizadas posteriormente con el rigor matemático correspondiente [Everitt and Hothorn, 2011].

El análisis exploratorio se enfoca en la detección, en cambio, la limpieza resuelve el problema con valores ausentes, datos atípicos y ruido, aunque este último proceso también los abarca la reducción de datos.

En lo que respecta a la transformación de datos, existen técnicas con las que se modifica los atributos del conjunto de entrenamiento. Ejemplos de este tipo de técnicas son: normalización y discretización. Mediante la normalización se elimina la dispersión excesiva de los datos, y con la discretización se delimitan valores por medio de umbrales a ciertos rangos determinados [Rencher, 2002].

Con respecto a la reducción de datos, las técnicas existentes para tal fin, son la selección y extracción de atributos e instancias. Estos dos enfoques se basan en un criterio de selección, descartando atributos irrelevantes y objetos innecesarios en un conjunto de entrenamiento, respectivamente. En la siguiente sección de describe la etapa de reducción de dimensionalidad que es la parte de la reducción de datos enfocada sólo en las características de los objetos.

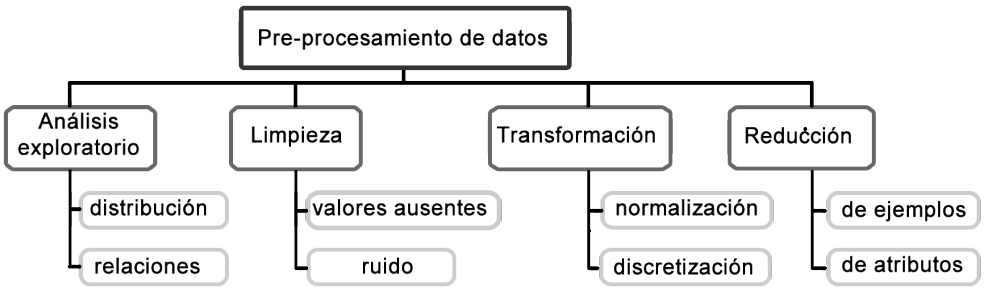


Figura 2.2: El diagrama muestra las diferentes técnicas a las que se puede someter el conjunto de datos en la etapa de pre-preocesamiento.

2.3. Reducción de dimensionalidad y conceptos asociados

Desde un punto de vista estadístico, vivimos en un mundo multidimensional, con una gran cantidad de características explicativas que modelan a cada objeto y evento en el universo. Problemas que involucran el concepto de multidimensionalidad son muy comunes en áreas como el Análisis de Datos Multivariantes y Bioinformática. En estadística, es

común escuchar sobre el problema que se presenta cuando la cantidad de datos a estudiar no es significativa. Este problema se complica si el número de variables de cada muestra es muy grande (pocos datos y muchos atributos). La peor situación surge cuando la cantidad de características que describen a cada instancia supera al número de elementos en el conjunto de datos. Estadísticamente es necesario que el número de elementos en el conjunto crezca de forma exponencial con respecto al número de variables. En la práctica, por lo general esta situación no se cumple [Cunningham, 2007]. Otro factor a tomar en cuenta es cuando el número de características es muy grande, entonces la probabilidad de que exista redundancia entre ellas es alta. Podemos entender como redundancia a la correlación entre estas características. Por lo general, se prefiere un conjunto de datos sin correlación entre sus variables, esto hace que exista menor o, en el caso óptimo, redundancia cero. Dicha eliminación de características redundantes se aplica a grandes conjuntos de objetos y se lleva a cabo mediante un análisis de los datos. En este análisis se extrae la información más relevante que se encuentra oculta entre sus variables descriptivas. Como consecuencia, en este proceso se detecta que variables son relevantes y la forma en que interactúan unas con otras. Al detectar que variables son relevantes se evita que en el proceso se pierda información importante. La extracción y selección de información en un conjunto de datos consiste en reformular el conjunto usando menos características, las cuales deben de ser capaces de conservar las propiedades del conjunto de datos original. A este proceso se le conoce como reducción de dimensionalidad [Venna et al., 2010].

Cuando se aplica un método de reducción de dimensionalidad a un conjunto de datos, la información descubierta y extraída también sirve de ayuda para interpretar y clasificar los datos existentes, de tal forma que dicha interpretación y clasificación sea similar o mejor que la correspondiente utilizando el conjunto original [Cunningham, 2007, Lee and Verleysen, 2007].

2.3.1. Maldición de la dimensionalidad

La maldición de la dimensionalidad [Bellman and Bellman, 1961] se refiere a una serie de dificultades que ocurren cuando se trata con datos de alta dimensión. Normalmente, cuando se tiene un conjunto de datos pequeños (pocas observaciones) con muchas características que lo describen, el espacio de alta dimensionalidad donde residen éstos es disperso. Lo anterior origina lo que se conoce como "fenómeno del espacio vacío"[Scott and Thompsonm, 1983], el cual se asocia con el hecho de que el número de muestras requeridas para estimar una función de varias variables con una exactitud dada sobre un dominio específico debe crecer exponencialmente de acuerdo al número de dimensiones. Esto ultimo es ampliamente conocido como maldición de la dimensionalidad, la cual aunado a espacios dispersos (fenómeno del espacio vacío) conduce a propiedades inesperadas del análisis de datos que residen en espacios de alta dimensión.

Cuando la dimensión crece, las propiedades de los espacios Euclidianos ya no son válidas, la longitud de los vectores tienden a normalizarse entre más se incremente el número de dimensiones en el que se plantea, este fenómeno se le conoce como el fenómeno de la concentración, el cual nos indica la falta de una métrica adecuada en espacios multidimensionales grandes. Dicho fenómeno hace que por ejemplo, el problema de clasificación

basado en la búsqueda del vecino más cercano (Nearest-Neighbor Search) sea un problema difícil de resolver en espacios altamente dimensionales, debido a que la distancia Euclidiana entre cualesquiera dos vectores es aproximadamente constante [Bellman and Bellman, 1961].

2.3.2. Dependencia y relevancia de variables

Una solución para atenuar o contrarrestar el problema antes mencionado en la sección 2.3.1 se logra encontrando las variables (características) que contienen dependencia entre ellas. Cuando dos o más variables están altamente correlacionadas, una de ellas contiene información sobre las otras, así, se puede conservar las variables que contienen la mayor información relevante y suprimir el resto de variables irrelevantes de las características del objeto. Sin embargo, hacer esto no garantiza conservar la información más relevante que se transmiten entre ellas. Por lo general, la correlación entre variables es muy compleja. Otra forma de reducir el número de variables es encontrar un conjunto transformado de estas que logre representar la información más relevante del conjunto original, con la menor perdida de información. Aquí, se busca que el conjunto transformado contenga un número menor de variables.

Al realizar una transformación se busca nuevas variables con propiedades bien definidas, las cuales aseguran que la transformación no altere la información contenida y transferida del conjunto original que ahora está representada de diferente forma. De acuerdo con el modelo de datos que se esté manejando, se debe de seleccionar una transformación o proyección adecuada. La proyección tiene como primer objetivo eliminar dependencia. Esto es, la reducción se lleva acabo con el fin de reducir el número de variables y trata de eliminar la redundancia en el conjunto original. El segundo y más complejo objetivo de una proyección, es recuperar la denominada variable latente, la que se esconde dentro de las variables originales y la cual no es posible medir directamente. Esta tarea recibe el nombre de separación de la variable latente, también es conocida como separación ciega de fuentes en el área de procesamiento de señales, o análisis de componentes independientes en análisis de datos multivariantes [Bellman and Bellman, 1961].

2.3.3. Aplicaciones de la reducción de dimensionalidad

El principal inconveniente cuando nos enfrentamos a datos con alta dimensionalidad es que esto afecta el rendimiento de los algoritmos de aprendizaje automático y reconocimiento de patrones. El número de variables presentes en una instancia determina el tamaño del espacio de hipótesis, esto es, conforme aumenta el número de características que describen a los objetos, el tamaño del espacio solución crece exponencialmente. Para solucionar este problema se aplican técnicas de RD, las cuales pueden lograr mejorar el rendimiento de un algoritmo, facilitar la interpretación y análisis de los resultados y reducir el tiempo de ejecución de dichos algoritmos [Cunningham, 2007].

La reducción de la dimensión se puede asociar a tres tareas:

1. La visualización y exploración.

2. La regresión.

3. La clasificación.

La visualización y exploración parte de un conjunto de datos residente en un espacio de alta dimensión y tiene como finalidad proyectarlo a un espacio de pocas dimensiones (usualmente dos o tres) mediante la preservación de sus propiedades intrínsecas. El conjunto resultante de baja dimensión debe poder representarse de forma gráfica para su exploración visual. En contraste, en una regresión, la reducción de dimensionalidad debe ayudar a que la regresión se comporte con un error mínimo cuando es necesario predecir nuevos puntos. De forma análoga, el objetivo de la clasificación es, después de aplicar la reducción a un conjunto, producir el mínimo error de clasificación. En [Lee and Verleysen, 2007] se afirma que los métodos de reducción de dimensionalidad pueden ser capaces de:

1. Estimar el número de variables latentes.

2. Reducir la dimensionalidad mediante una proyección de los datos.

3. Recuperar variables latentes mediante una proyección de los datos.

Las variables latentes son las variables que no se observan directamente sino que son inferidas a partir de otras variables. Para encontrar el número de variables latentes, es necesario realizar una estimación de la dimensión intrínseca. Sin embargo, no todos los métodos son capaces de poder realizar esta estimación. Durante la reducción de dimensionalidad interesa capturar las variables latentes, y descartar las variables que involucran ruido y otras imperfecciones. El segundo paso es proyectar los datos de alta dimensión en una dimensión inferior, con el objetivo de lograr una representación compacta y facilitar el posterior tratamiento de los datos. Dicho de otra forma, lo que se busca es la visualización y/o compresión de los datos. Por último, la tarea de separación de variables latentes también implica medios para recuperar las variables, con el fin de cumplir con un objetivo más allá de sólo una reducción de la dimensionalidad. En este proceso se imponen restricciones adicionales al nuevo espacio de baja dimensión. Por ejemplo, es común que los algoritmos de RD al realizar la tarea de recuperación de variables latentes modelen las nuevas variables del espacio de baja dimensión como una combinación lineal de las latentes para garantizar que los datos cumplan con cierto criterio, como puede ser independencia estadística.

2.3.4. Tipos de reducción de dimensionalidad

Existen dos principales enfoques para la reducción de la dimensionalidad, aunque ambos reducen el conjunto de características, el primero de ellos lo hace transformando el conjunto de variables originales y el segundo enfoque selecciona un subconjunto de estas variables sin alterarlas [Cunningham, 2007].

RD mediante transformación de características

Al utilizar este enfoque se transforman las características originales de tal forma que se encuentra un nuevo grupo de objetos con el mismo número de instancias, pero con menor número de características descriptivas. En dicha transformación se busca que el subconjunto de características (que es diferente al original), contenga la información más relevante que almacenan las variables iniciales. En consecuencia, el nuevo conjunto es una representación del conjunto original. Esta técnica puede dividirse en dos subcategorias [Cunningham, 2007].

Extracción de Características (EC): Consiste en producir un nuevo conjunto de características aplicando un mapeo a los datos originales. El análisis de componentes principales y análisis discriminante lineal son los dos algoritmos más conocidos que realizan extracción de características mediante aprendizaje no supervisado y supervisado, respectivamente.

Generación de Características (GC): Primero se encuentra la información oculta (o perdida) entre las características del conjunto y luego se incrementa el tamaño del espacio de hipótesis generando nuevas características, las cuales sólo almacenan datos que enfatizan la nueva información descubierta.

La EC es la técnica más utilizada, debido a su capacidad para reducir el conjunto de características. Es lógico pensar que la generación de nuevas características no es la mejor opción para reducir dimensionalidad, pues obliga al conjunto de variables a expandirse. Sin embargo, después de aplicar el método de GC podemos aplicar un método de extracción (EC) para obtener un subconjunto de características útil.

RD mediante selección de características

El objetivo de este enfoque es encontrar el mejor subconjunto (mínimo) de características, el cual debe ofrecer resultados similares o mejores dependiendo del problema al que se enfrente. La ventaja de este enfoque es que las características seleccionadas representan el concepto físico o abstracto original de su variable correspondiente, y pueden ser interpretadas directamente. Por el contrario, en las técnicas de transformación las nuevas variables, por lo general, no pueden ser interpretadas directamente ya que representa conceptos y métricas desconocidas. Usualmente, un modelo predictivo se utiliza para evaluar todas las posibles combinaciones de características y asignar una puntuación basada en la exactitud del modelo (aprendizaje supervisado). Los sistemas de evaluación pueden ser de carácter supervisado y no supervisado, en ambos casos se dividen en tres categorías. Para el aprendizaje supervisado estas categorías consisten en lo siguiente.

Enfoque de empaquetado (wrapper): El criterio de selección está basado en la exactitud dada por un clasificador.

Enfoque de filtro (filter): El criterio de selección está basado en una función distinta a la precisión de un clasificador, por ejemplo, una función ranking o criterio de separabilidad de clases.

Enfoque embebido (embedded): Utiliza un medida de perdida o ganancia de información para elegir las mejores características. Dicha medida se encuentra incorporada en

el método de aprendizaje utilizado.

Existen métodos de selección de características no supervisados, sin embargo es una zona menos explorada. La razón es que su objetivo es menos claro y es complicado encontrar un número reducido de características cuando el número de grupos (cluster) a crear es desconocido [Bellman and Bellman, 1961].

2.3.5. Métodos de reducción de dimensionalidad

[Sarlin and Peltonen, 2011] hacen una división de los métodos de reducción en dos generaciones. La primera generación se compone de los métodos clásicos que aún son ampliamente aceptados y difundidos en distintas áreas de las ciencias. Estos métodos tienen como objetivo, dado un conjunto de datos que reside en un espacio de alta dimensión, proyectar cada uno de los puntos (datos) a un espacio de menor dimensión basándose en la preservación de las distancias. La segunda generación es un grupo de métodos menos homogéneo que van desde las llamadas técnicas espectrales hasta las técnicas basadas en grafos. Debido a la gran cantidad de métodos, sólo se mencionan algunos de los más utilizados pertenecientes a ambas generaciones.

Análisis de componentes principales

El análisis de componentes principales (del inglés Principal Components Analysis, PCA) es uno de los métodos clásicos del análisis multivariante de datos para reducción de dimensionalidad. Cuando manejamos conjuntos de datos con demasiadas variables, los datos contenidos en dicho conjunto no son aptos para aplicarles técnicas que permitan visualizar dicho conjunto por medio de un recurso gráfico, su alta dimensionalidad hace complicado el análisis estadístico. Por lo general, PCA se utiliza para representar de forma gráfica conjuntos de datos reducidos para su análisis visual y exploratorio [Everitt and Hothorn, 2011].

Dado un conjunto de datos con p variable, PCA transforma dicho conjunto en uno nuevo con r variables, siendo $r << p$. Este nuevo conjunto debe cumplir ciertas restricciones: las nuevas variables, las cuales reciben el nombre de componentes principales, deben ser una combinación lineal de las variables de partida, además, cada nueva variable debe representar una parte de la variabilidad de los datos originales. Cuanto mayor sea la varianza de las componentes, mayor es la información que almacenan las nuevas variables, siguiendo esta lógica, la varianza debe ir decreciendo según se vaya obteniendo cada nueva variable. Por esta razón, la primera componente debe de ser aquella con mayor varianza, mientras que la última componente recogerá la menor variabilidad de los datos en el conjunto inicial. Cabe mencionar que la suma total de estas varianzas es igual a la suma de las varianzas del conjunto original [Bell, 2014]. Otra condición que debe cumplir el nuevo conjunto es que las nuevas variables no estén correlacionadas, ya que esta correlación impide apreciar de forma precisa el rol que juega cada variable en el problema bajo investigación [Rencher, 2002]. Dicho de otra forma, cualquier combinación de las nuevas variables no debe contener información en común o información que pueda deducirse a partir de una u otras variables.

El número de componentes principales es igual al número de variables del conjunto de partida, por este motivo, si utilizamos todas las componentes principales para generar el nuevo conjunto, obtendremos una proyección con las mismas dimensiones que el conjunto de partida. La única diferencia es que el conjunto generado con las componentes principales contiene datos no correlacionados. Si partimos de un conjunto de datos en el cual no existen variables dependientes, entonces no tiene caso aplicar PCA. Por el contrario, en conjuntos de datos en los cuales sus variables tienen alta correlación es posible encontrar un nuevo subconjunto de pocas variables que sustituya al original con la mínima pérdida de información [Jolliffe, 2002].

Análisis discriminante lineal

El análisis discriminante lineal (en inglés Linear Discriminant Analysis, LDA) junto a PCA son los métodos más conocidos para reducción de la dimensionalidad. LDA ademas de ser un método de reducción de la dimensión, también es un método de clasificación supervisada. LDA se utiliza cuando la variable dependiente en los datos es categórica. Este algoritmo busca una combinación lineal de las variables independientes que mejor discrimine la categoría de la variable dependiente. LDA crea un hiperplano discriminante mediante una combinación lineal. Para que este algoritmo funcione, los diferentes grupos contenidos en el conjunto de datos deben de tener una distribución normal, matrices de covarianza similares y medias diferentes. LDA también exige que el número de variables descriptivas sea mayor que el número de grupos y el número total de ejemplos existentes en la población [Bell, 2014].

Existen dos objetivos que persigue LDA, el primero es identificar la contribución que cada una de las variables aporta para la separación de los grupos mediante un análisis de varianzas. El segundo objetivo es encontrar una proyección basada en una combinación lineal de las variables en la cual los datos sea mejor discriminados de acuerdo a su etiqueta de clase [Gareth et al., 2014]. Los hiperplanos de separación son creados mediante la función discriminante de Fisher [Fisher, 1936]. El espacio en la que reside el nuevo conjunto de datos puede contener a los más $k-1$ dimensiones, esto se debe a que se crean $k-1$ hiperplanos discriminantes [Bell, 2014]. El análisis discriminante es óptimo cuando las variables provienen de una distribución normal multivariada con igual varianza dentro de cada grupo. Los resultados pueden no ser válidos ante la presencia de algunos pocos valores extremos [Khattree and Naik, 2000].

Regresión logística

El enfoque de regresión logística (RL) persigue dos objetivos: el primero de ellos es construir un modelo que permita predecir el valor de una nueva variable dependiente mediante estimaciones de probabilidad de pertenencia. El segundo objetivo es estimar la relación entre las variables independientes y la variable dependiente obteniendo con esto la influencia que cada variable aporta para la predicción de su categoría correspondiente. Por lo anterior, se puede deducir que RL se utiliza para problemas de clasificación en donde el problema es reducir el número de variables que operan en el conjunto de datos.

Se denomina a la probabilidad de éxito como p, la cual representa la probabilidad de pertenencia de una instancia a una determinada categoría y como q a la probabilidad de no pertenencia (o fracaso). El cociente p/q (llamado Odds) representa cuánto más o menos probable es el éxito que el fracaso [Hosmer and Lemeshow, 2000], siempre que $odds > 1$ el éxito tiene ventaja sobre el fracaso.

El modelo de regresión permite estudiar si la variable discreta depende o no de una o más variables independientes. Si los dicho anteriormente sucede, los coeficientes del modelo de regresión son los que dictan la relación de dependencia [Gareth et al., 2014]. El criterio para decidir si una variable contiene una aportación significativa para la categorización de las muestras es fijado por el valor-p [Izenman, 2006].

Si los valores-p asociados a los coeficientes δ_i (un coeficiente por variable) son inferiores a el intervalo de confianza establecido (usualmente 0.05) rechazaremos la hipótesis nula y aceptamos que el variable X_i aporta información al modelo y debe de estar contenido en él. El modelo logístico no es eficiente en el caso normal, dicho modelo puede ser más eficaz cuando los grupos en el conjunto de datos no tienen la misma matriz de covarianza, o su distribución en un gráfico de dispersión no se amolda a una elipse [Gareth et al., 2014].

2.4. Aprendizaje multi-etiqueta

Como se mencionó en la sección anterior, el universo es multidimensionalidad, todo aquello cuya existencia es perceptible, a lo cual denominamos objeto, contiene un número de características descriptivas. En esta sección se introduce un nuevo concepto que esta asociado a otra cualidad de estos objetos. Esta cualidad surge al clasificar los objetos de acuerdo a sus características y funciones. Siguiendo este criterio, un objeto puede cumplir más de una función y compartir características (no todas) similares con otros objetos existentes. Cuando esto sucede podemos asociar un objeto con diferentes etiquetas discriminantes, esta propiedad en los objetos se denomina multi-etiqueta, la cual nos permite modelar de forma más realista ciertos problemas del mundo real [Sun et al., 2013].

En los problemas clásicos de reconocimiento de patrones, las clases son mutuamente excluyentes por definición. Las etiquetas asignadas a cada objeto del conjunto de datos son previamente definidas y se utilizan para describir brevemente al objeto, asumiendo que cada ejemplo sólo puede pertenecer a una clase. Los posibles casos de pertenencia restantes son ignorados en la creación del modelo. Como consecuencia los errores de clasificación se producen cuando las clases se solapan en el espacio de características seleccionado. En los métodos de clasificación tradicional, los modelos son utilizados para predecir sólo una clase de pertenencia al clasificar nuevos objetos. Sin embargo, en algunas tareas de clasificación, es probable que algunos datos pertenezcan a múltiples clases, haciendo que las clases se sobrepongan desde el momento de crear el modelo de predicción. Por ejemplo: en clasificación de música una canción puede contener influencias de rock y de blues [McCallum, 1999, Schapire and Singer, 2000]. En el diagnóstico médico, una enfermedad puede pertenecer a múltiples categorías, y en el contexto biológico los genes pueden tener múltiples funciones [Clare and King, 2001]. Por lo anterior, podemos afirmar que existen

problemas de clasificación reales, en los cuales las clases no son mutuamente excluyentes y pueden compartir el espacio de hipótesis. Sabiendo esto, al momento de crear un modelo de clasificación, podemos tomar en cuenta lo antes mencionado y diseñar un clasificador que nos permita asignar una o más etiquetas a cada nuevo objeto a clasificar. El reto del aprendizaje multi-etiqueta es cómo utilizar eficazmente la correlación entre las diferentes etiquetas de los ejemplos del conjunto de datos.

De manera formal, sean X y Y dos conjuntos, los cuales denotan el espacio de instancias de entrada y el espacio de etiquetas de salida, respectivamente. Y tiene la forma $Y = \{0, 1\}^k$, donde k es el número de etiquetas. El j-ésimo componente del vector de etiquetas toma el valor 1 si la etiqueta es relevante y 0 de forma contraria. De forma similar a la clasificación tradicional el objetivo del aprendizaje multi-etiqueta es que un clasificador aprenda una función $f : X \to Y$ la cual asigna una etiqueta a cada instancia $x \in X$. Específicamente la salida del clasificador f para una instancia $x \in X$ es:

$$f(x) = [f_1(x), f_2(x), ...f_k(x)]^T \tag{2.1}$$

donde $f_j(x)(j = 1, 2, ..., k)$ sólo pueden tener el valor 1 o 0, lo cual indica la asociación de x con la j-ésima etiqueta. El conjunto de clases es denotado como $C = \{C_1, C_2, ..., C_k\}$.

Las principales aplicaciones del aprendizaje multi-etiqueta en problemas del mundo real que se han estudiado a la fecha son: clasificación de texto, clasificación de literatura, problemas de biología, ergonómica y clasificación semántica de imágenes.

En este contexto, al clasificar escenas podemos asociar el contenido de una imagen a diversas etiquetas, como: montaña, lago, verano, vacaciones. etc. Este problema se puede modelar como un problema de aprendizaje multi-etiqueta. El poder asociar etiquetas a las imágenes nos permite buscar imágenes similares, reduciendo el espacio de búsqueda y mejorando la exactitud en la recuperación.

La clasificación de texto puede ser modelada naturalmente como un problema de aprendizaje multi-etiqueta. La clasificación de texto incluye varios campos, como: páginas web bajo etiquetas jerárquicas, clasificación de hipertexto, detección de género en un texto, filtrado de contenidos, análisis y clasificación automática de correo electrónico, clasificación de sentimientos, estados afectivos u opiniones.

En el campo de la bioinformática, la clasificación de genes es de suma importancia. Los genes con funciones similares tienen perfiles de expresión similares. Cada gen puede estar asociado a múltiples funciones. Estas funciones se pueden considerar como etiquetas. La predicción de estas funciones se pueden modelar como un problema de aprendizaje multi-etiqueta. De forma análoga, las proteínas desempeñan varias funciones simultáneamente y se puede atacar el problema de la misma forma.

Dentro del campo de la medicina, la clasificación multi-etiqueta se ha utilizado para solucionar varios problemas. El más inmediato de ellos es el diagnóstico médico, el cual se puede modelar como multi-etiqueta, ya que hay enfermedades distintas que comparten síntomas.

2.4.1. Enfoques del aprendizaje multi-etiqueta

Los métodos existentes para el aprendizaje multi-etiqueta pueden dividirse en dos categorías: transformación del problema y adaptación del problema. El primer enfoque, transforma el problema de aprendizaje multi-etiqueta en una serie de problemas de aprendizaje de etiqueta simple. El segundo enfoque adapta los algoritmos de clasificación tradicionales en algoritmos de clasificación multi-etiqueta directamente [Sun et al., 2013]. Tres de los principales esquemas de transformación, los cuales recaen en el primer enfoque son: Copy Transformation (CT), Binary Relevance (BR) y Label Power Set (LPS).

Dado un conjunto de datos con etiquetas múltiples, CT ataca el problema de multi-etiqueta convirtiéndolo en un problema multi-clase, para ello realiza una transformación en el conjunto de datos. Específicamente, CT duplica cada ejemplo del conjunto original tantas veces como etiquetas asociadas tenga dicha instancia en el conjunto de datos. A cada réplica le asocia una de las etiquetas con las que esta reaccionado el objeto. BR es el enfoque más utilizado en la literatura, a diferencia de CT, crea una serie conjuntos de datos binarios, uno por cada etiqueta asociada al problema. Para cada etiqueta existente, toma todas las instancias con las que está asociada y las agrega al conjunto de entrenamiento como un ejemplo positivo de dicha clase, y las instancias restantes que no están asociadas a esta etiqueta son agregadas como ejemplos negativos de la etiqueta que se encuentre en proceso. Una vez creados los nuevos conjuntos, estos son utilizados para entrenar a un clasificador binario, un clasificador por cada nuevo conjunto generado. La clasificación multi-etiqueta se basa en la respuesta de todos los clasificadores binario para generar el conjunto de etiquetas que se le asociará a cada nueva instancia a clasificar. El principal inconveniente es que considera que las etiquetas son independientes entre sí. Otro inconveniente es que el nuevo conjunto puede sufrir de clases desbalanceadas, ya que el número de ejemplos positivos por lo general es significativamente mayor que el número de ejemplos negativos para algunas clases. Label Power Set por su parte crea tantas nuevas clases como combinaciones de las clases base haya en el conjunto de entrenamiento, para el peor caso se generarían 2^k (conjunto potencia) nuevas clases. El problema es que este enfoque se vuelve muy complejo al aumentar el número de clases [Weston, 2001]. Después de la transformación se pueden aplicar algunos algoritmos de clasificación de etiqueta simple. El rendimiento de diferentes algoritmos de clasificación de etiqueta-múltiple después de someter los datos a una transformación es comparado en [Read et al., 2012]. Un ejemplo de la transformación realizada por CT, BR, y PLS sobre un conjunto sencillo se muestra en la figura 2.3.

El segundo enfoque depende del algoritmo de clasificación que se necesite adaptar a un modelo de etiquetas múltiples. Algunos métodos propuestos toman como criterio la búsqueda de dependencia entre etiquetas, otros se enfocan en el tipo de correlación que existe entre ellas [Dembczyński et al., 2010]. Sobre la primer perspectiva existen tres opciones: buscar relación sólo entre pares de etiquetas [Fürnkranz et al., 2008, Weston, 2001], buscar correlación entre subconjuntos más amplios (más de dos etiquetas) [Godbole and Sarawagi, 2004, Maimon and Rokach, 2005], o encontrar la influencia del resto de la etiquetas al tratar de predecir una de ellas [Cheng and Hüllermeier, 2009, Godbole and Sarawagi, 2004]. Si nos enfocamos en búsqueda de correlación entre etiquetas, podemos

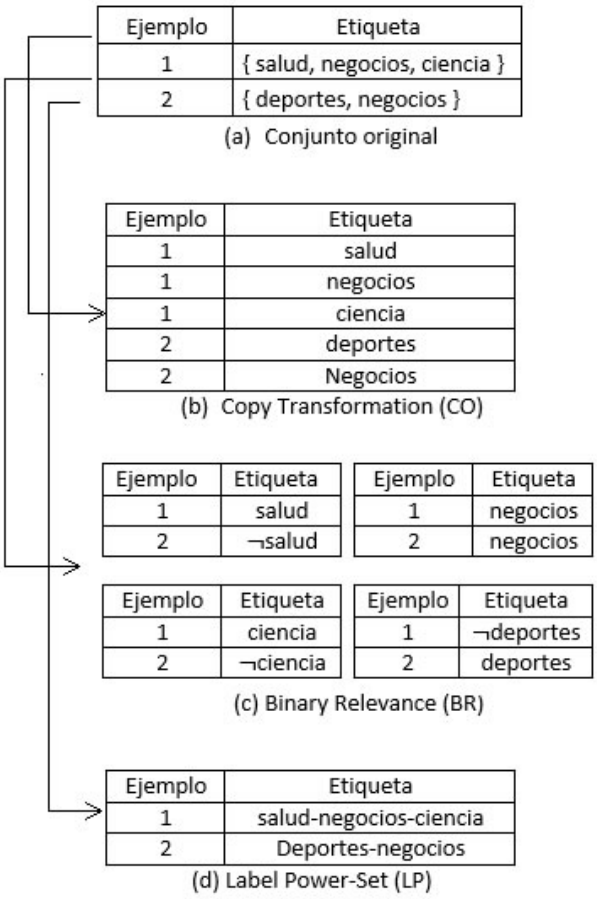


Figura 2.3: Un ejemplo que muestra las diferentes transformaciones de un problema multi-etiqueta

elegir entre los algoritmos diseñados para detectar dependencia condicional [Dembczyński et al., 2010, Godbole and Sarawagi, 2004, Read et al., 2011] y los que buscan dependencia incondicional [Cheng and Hüllermeier, 2009, Godbole and Sarawagi, 2004]. Dos ejemplos del enfoque adaptativo son: **k-NN** y **arboles de decisión**, para ambos métodos existe su versión multi-etiqueta.

2.4.2. Métodos de reducción de dimensionalidad multi-etiqueta

Reducción de la dimensionalidad multi-etiqueta basada en hipergrafos

Cuando hablamos de métodos de reducción de dimensionalidad espectral nos referimos a métodos que se basan en la descomposición espectral de la matriz S de características. La cual es simétrica, cuadrada y se obtienen a partir de un conjunto de datos X. Existen muchos métodos para la construcción de esta matriz, una vez construida S, todos los métodos obtienen sus valores y vectores propios, los cuales son utilizados posteriormente. Al construir la matriz de características el objetivo es recolectar de acuerdo a un crite-

rio, ciertas propiedades que cumple el subespacio o variedad donde residen los datos de origen. Posteriormente, al realizar la descomposición espectral de esta matriz se puede proyectar a un nuevo subespacio en base a los vectores y valores propios (espectro de la matriz) [Ng et al., 2001]. Los algoritmos más populares que recaen en este enfoque son PCA y MDS, por parte de los métodos lineales. La mayoría de conjuntos de datos que se enfrentan con métodos lineales son altamente no lineales, como consecuencia los nuevos datos proyectados pueden sufrir de distorsión debido al efecto de la curvatura de la variedad. La distancia Euclidiana por ejemplo, no respeta la geometría de la variedad y no es recomendable en espacios de alta dimensionalidad [Friedman, 1992].

Las técnicas de reducción de dimensionalidad espectrales no lineales buscan resolver este problema mediante el modelado de los datos suponiendo que no residen en un subespacio, sino en una variedad. Partiendo de que dicha variedad no es lineal, estos métodos tratan de mantener la estructura de los datos al ser proyectados a un espacio de baja dimensión. La proyección respeta la estructura de tal forma que los pares de datos que se encuentran cerca unos de otros en la variedad original también lo estarán en el espacio de baja dimensión. Lo mismo sucede para los pares de datos que se encuentran lejos entre si en la variedad. Los métodos espectrales no lineales de RD utilizan la teoría de grafos, en la cual el espectro y los vectores propios son obtenidos de la matriz creada a partir de un grafo [Rosenberg, 1997].

La construcción del grafo es un problema no trivial y existen varios enfoques para su implementación. A pesar de que los métodos para construir el grafo han demostrado eficacia, existen situaciones en las que su modelado no responde adecuadamente. Estos casos se dan cuando existe presencia de outliers o más de una variedad en el mismo conjunto de datos [Zien et al., 1999]. Los grafos pueden no ser capaces de representar estructuras complicadas, pero logran aproximarse a estas formas complejas de manera aceptable. La topología de un grafo representa su información estructural.

Dada una matriz de datos X una forma de construir un grafo es crear diferentes subconjuntos disjuntos mediante un criterio de similitud. Este criterio minimiza la similitud entre pares de puntos de diferentes subconjuntos mientras que para los pares de puntos dentro del mismo subconjunto debe de mantener un alto grado de similitud. Los tipos de criterio más populares para transformar un conjunto de datos basados en una función de similitud que responde a las relaciones de vecindad entre los puntos del conjunto de datos son:

Regla k: Para cada vértice se definen sus k-NNs utilizando la distancia Euclidiana, x_i y x_j estarán conectados si x_j esta dentro del conjunto de los k-NNs de x_i.

Regla ε: Se conecta x_i y x_j si $|x_i - x_j| < \epsilon$

Función kernel: Utilizando una función kernel también es posible definir un grafo partiendo de una matriz de datos. Mediante una función kernel se obtiene una función de similitud arbitraria teniendo en cuenta que esta no debe ser negativa. El inconveniente en

este enfoque es la forma en que esta función debe de captar la estructura local y global de los datos [Agarwal et al., 2006].

Al crearse el modelo que representa la estructura de los datos mediante un grafo, el paso siguiente es asignar los pesos a las aristas. Para la creación de la matriz de pesos W existen varios enfoques, entre los más conocidos esta los métodos basados en observación que crean los pesos de acuerdo a las conexiones entrantes y salientes de los nodos. Otro tipo de métodos se basan en similaridad, estos métodos utilizan una función kernel para la preservación de las conexiones.

Ya que se a construido el grafo y asignado su matriz de pesos debemos de elegir un método de aprendizaje (supervisado, no supervisado o semi-supervisado) de acuerdo al problema que se desea resolver. Existen diversos métodos de clasificación, agrupación y reducción de dimensionalidad basados en grafos [Zien et al., 1999]. Por ejemplo: MINCUT, kernel k-means, funciones armónicas, entre otros.

Hipergrafos Un hipergrafo es una generalización de un grafo tradicional. Un hipergrafo $G = (V, E)$ se define como un conjunto de vértices V y un conjunto de hiperaristas E, donde cada $e \in E$ es un subconjunto de V. Una hiperarista e es incidente en un vértice v, cuando v está contenida en e. El grado de un vértice en un grafo, es el número de aristas incidentes a él. En la teoría espectral el grado de un vértice $v \in V$ se define como:

$$d(v) = \sum_{v \in e, e \in E} w(e)$$

donde $w(e)$ es el peso asociado a la hiperarista $e \in E$. El grado de una hiperarista e, es el número de vértices en e

$$d(e) = |e|$$

Las matrices diagonales formadas por $d(e)$, $d(v)$, $w(e)$ se denota como D_e, D_v, W_H, respectivamente. La matriz de incidencia $J \in R^{|V| \times |E|}$ es definida como :

$$J(v, e) = \begin{cases} 1 \text{ si } v \in e \\ 0 \text{ en otro caso} \end{cases} \tag{2.2}$$

La matriz de adyacencia A es definida como :

$$A = JW_H J^T - D_v$$

El grafo Laplaciano se utiliza en el aprendizaje semi-supervisado [Zhou et al., 2006] y agrupación espectral, los vectores propios del grafo Laplaciano son indispensables para el aprendizaje. El grafo Laplaciano esta fuertemente correlacionado con la estructura de su hipergrafo.

Mediante hipergrafos podemos explorar la correlación existente entre las etiquetas de un conjunto de datos, este tipo de grafos extraen las relaciones que existen entre varias instancias que comparten una mista etiqueta. Podemos construir un hipergrafo que representa un conjunto de datos multi-etiqueta de la siguiente forma: Creamos una hiperarsita por cada etiqueta existente en los datos, posteriormente a cada una de las instancias del

conjunto las hacemos incidir en cada una de las hiperaristas con los que estén asociadas. En el aprendizaje espectral basado en grafo las etiquetas representan las hiperaristas y las instancias representan los vértices, al realizar la reducción de dimensionalidad la proyección se guía por la información que el hipergrafo extrae de las etiquetas [Sun et al., 2013].

En el contexto de aprendizaje automático el objetivo es aprender tomando en cuenta la información que se puede extraer de los vértices. De esta forma el aprendizaje se obtiene de la relación (conexión) que existe entre una arista y los vértices en los que incide. Los grafos tradicionales son llamados 2-grafos, debido a que una arista sólo puede conectar a dos vértices. En problemas de clasificación el interés recae en las etiquetas que cada vértice representa, debido a esto un grafo Laplaciano nos proporciona suficiente información al analizar sus vértices y aristas incidentes. Para poder capturar la información subyacente se han propuesto algunos métodos para crear hipergrafos Laplacianos a partir de un hipergrafo.

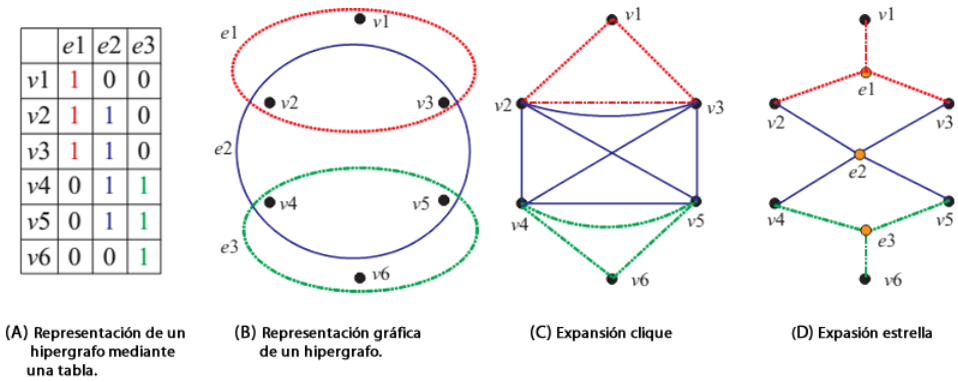
La matriz Laplaciana de un hipergrafo se puede construir mediante la expansión clique, la expansión estrella o puede ser definida directamente de manera análoga a como se hace en los grafos tradicionales.

Expansión clique

Un clique es un conjunto de vértices V tal que para todo par de vértices de V, existe una arista que las conecta. En otras palabras, un clique es un subgrafo en que cada vértice está conectado a todos los vértice del subgrafo, es decir, todos los vértices del subgrafo son adyacentes, como en un grafo totalmente conectado.

Expansión estrella

Un nuevo vértice es introducido por cada hiperarista, luego, este nuevo vértice es conectado a cada vértice en la hiperarista. Un ejemplo que ilustra la expansión clique y estrella se muestra en la figura 2.6.



	e1	e2	e3
v1	1	0	0
v2	1	1	0
v3	1	1	0
v4	0	1	1
v5	0	1	1
v6	0	0	1

Figura 2.4: Ejemplo de expasión estrella y clique apartir de un hipergrafo [Sun et al., 2013]

La forma directa de construir una matriz Laplaciana que representa un hipergrafo es

la siguiente: Sea

$$L_Z = I - S_z \tag{2.3}$$

la matriz Laplaciana normalizada, donde

$$S_z = D_v^{-1/2} J W_H D_e^{-1} J^T D_v^{-1/2} \tag{2.4}$$

Para poder realizar un aprendizaje espectral de un grafo con etiquetas múltiples se aprovecha las propiedades espectrales del hipergrafo en las cuales esta codificada la información de correlación entre las etiquetas. Mediante la construcción de un hipergrafo Laplaciano es posible aprender la forma de incrustar los datos a una dimensión inferior a través de una transformación lineal $W \in R^{d \times k}$ resolviendo el siguiente problema de optimización:

$$min_w \, Tr(W^T X L X^T W) \tag{2.5}$$

$$s.t. \; W^T X X^T W = I_k \tag{2.6}$$

Donde L representa la matriz del grafo Laplaciano normalizado. Esta ecuación (2.5) trata de preservar la relación existente en los datos de la matriz Laplaciana. De acuerdo con la teoría espectral de hipergrafos las instancias que comparten la misma etiqueta tienden a estar más cerca entre ellas en el espacio embebido.

Análisis discriminante multi-etiqueta

LDA tradicional (ver sección 2.3.5) es un método de reducción y clasificación ampliamente conocido en el área de aprendizaje automático. LDA fue diseñado originalmente para clasificación con etiquetas simples, debido a esto, no es posible utilizarlo en el aprendizaje multi-etiqueta directamente. Sin embargo, es posible replantear el algoritmo para que logre obtener la correlación entre las diversas etiquetas de cada muestra. Para ello, se debe buscar la forma de que al maximizar la razón de la variabilidad entre-grupos con respecto de la variabilidad intra-grupos, se tome en cuenta la correlación entre las etiquetas asociadas a cada instancia. Si se toman en cuenta estos dos aspectos al formar los ejes discriminantes, el algoritmo LDA ahora toma el nombre de análisis discriminante multi-etiqueta (Multi-label discriminant analysis, MLDA) [Wang et al., 2010].

Sea $X = \{x_i, y_i\}_{(i=1)}^n$ un conjunto de datos con n ejemplos y K clases, donde cada $x_i \in R^p$ y $y_i \in \{0,1\}^K$. Sea $y_i(k) = 1$ si x_i esta asociado a las k-ésima clase y $y_i(k) = 0$ en otro caso. Debido a que el conjunto contiene K clases entonces suponemos que el conjunto de datos se puede dividir en K grupos representados por $\{\pi_k\}_{k=1}^K$ donde π_k representa al subconjunto de datos de las k-ésima clase con n_k muestras. Denotaremos al conjunto $X = [x_1, ..., x_n]$ y $Y = [y_1, ..., y_n]^T = [y_{(1)}, , y_{(k)}]$, donde $y_{(k)} \in \{0,1\}^n$ es un vector que denota la asociación de todos los elementos del conjunto con la k-ésima clase.

Si partimos de la idea original de LDA, entonces se busca una transformación lineal $G \to R^p \times r$, la cual mapea a cada x_i que radica en un espacio p-dimensional a un nuevo

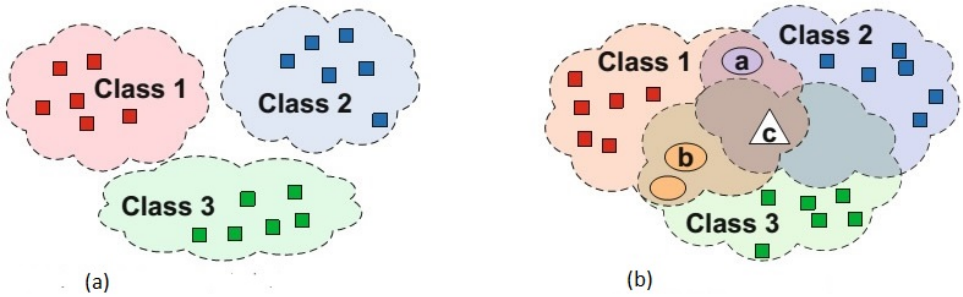


Figura 2.5: (a) Conjuntos de datos con etiqueta simple. (b) Conjunto de datos muti-etiqueta [Wang et al., 2010]

espacio r-dimensional con $r << p$, en el cual $x_i \rightarrow G^T x_i = q_i$. LDA tradicional se basa en la descomposición de la matriz de varianza $S_t = S_b + S_w$, donde S_t representa la matriz de varianza del conjunto, S_b representan las matriz de varianza entre-grupos y S_w representa la matriz de la varianza intra-grupos. Estas tres matrices se definen como:

$$S_b = \sum_{k=1}^{K} (m_k - m)(m_k - m)^T \tag{2.7}$$

$$S_w = \sum_{k=1}^{K} \sum_{x_i \in \pi_k} (x_i - m_k)(x_i - m_k)^T \tag{2.8}$$

$$S_t = \sum_{i=1}^{n} (x_i - m)(x_i - m)^T \tag{2.9}$$

donde $m_k = \frac{1}{n_k} \sum_{x_i \in \pi_k} x_i$ es la media de la k-ésima clase y $m = \frac{1}{n} \sum_{i=1}^{n} x_i$ es la media global del conjunto. Debido a que la obtención del nuevo espacio depende de que la transformación G logre maximizar la variabilidad entre-grupos y minimizar la variabilidad intra-grupos, el problema se reduce a encontrar los primero r valores propios de la siguiente ecuación:

$$S_w^{-1} S_b G = \lambda G \tag{2.10}$$

Las ecuaciones anteriores (2.7, 2.8, 2.9 y 2.10) no son adecuadas para el caso multi-etiqueta, ya que las particiones de las K clases se traslapan entre ellas haciendo que los hiperplanos que son los limites de decisión se vuelven inciertos. Además, no se ha definido en que medida un ejemplo con múltiples etiquetas debe contribuir a la varianza de los grupos a los que pertenece. La correlación entre etiquetas puede ser incorporada redefiniendo las ecuaciones 2.7, 2.8, 2.9 y 2.10 de la siguiente forma:

$$S_b = \sum_{k=1}^{K} S_b^{(k)}, S_b^{(k)} = (\sum_{i=1}^{n} Y_{ik})(m_k - m)(m_k - m)^T \tag{2.11}$$

$$S_w = \sum_{k=1}^{K} S_w^{(k)}, S_w^{(k)} = \sum_{i=1}^{n} Y_{ik}(x_i - m_k)(x_i - m_k)^T \tag{2.12}$$

$$S_t = \sum_{k=1}^{K} S_t^{(k)}, S_t^{(k)} = \sum_{i=1}^{n} Y_{ik}(x_i - m)(x_i - m)^T \tag{2.13}$$

donde m_k y m se definen como:

$$m_k = \frac{\sum_{i=1}^{n} Y_{ik} x_i}{\sum_{i=1}^{n} Y_{ik}}, m = \frac{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik} x_i}{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}} \tag{2.14}$$

Si consideramos que la correlación entre dos clases se formula así:

$$C_{kl} = cos(y_{(k)}, y_{(l)}) = \frac{\langle y_{(k)}, y_{(l)} \rangle}{\|y_{(k)}\| \|y_{(l)}\|} \tag{2.15}$$

De forma general $C \in \mathbb{R}^{k \times k}$ y es una matriz simétrica. Para un problema de etiquetas simple $C = I$, lo cual se interpreta como un caso particular en donde la correlación entre etiquetas no es utilizada en el aprendizaje. Debido a que en el aprendizaje multi-etiqueta un ejemplo puede pertenecer a más de una clase simultáneamente, el traslape entre dos clases nos indica que existe una correlación entre estas dos clases. Si el traslape entre clases es grande estadísticamente, podemos deducir que las clases están fuertemente relacionadas. Este fenómeno se puede utilizar en problemas de clasificación para inferir las clases de pertenencia de los nuevas instancias, si sustituimos Y por YC en las ecuaciones 2.11, 2.12 y 2.13 al formar las matrices se tomará en cuenta la correlación de clase.

En la figura 2.5 (b) se observa que la instancia a pertenece a la clase 1 y a la clase 2, por tal, es usada en $S_b^{(1)}$ y $S_b^{(2)}$ de la operación $S_b = S_b^{(1)} + S_b^{(2)} + S_b^{(3)}$. Lo mismo sucede para todas los ejemplos que pertenecen a más de una clase. El problema cuando ocurre esto es que existe un sobre conteo de los datos, lo que no sucede en LDA tradicional. Para corregir este problema se establece una matriz de normalización $Z = [Z_1, ..., Z_n]^T \in R^{n \times K} : z_i = \frac{y_i C}{\|y_i\|_{\ell 1}}$ donde $\|\|_{\ell 1}$ es la ℓ1-norma de un vector. Ahora, podemos reemplazar Z por Y en las ecuaciones 2.7, 2.8, 2.9 y 2.10 obteniendo las matrices finales de varianza para MLDA.

Podemos rescribir las matrices de varianza a una forma más compacta y resumida pero equivalente. Primero centramos los datos desde un enfoque multi-etiqueta:

$$\tilde{X} = X - me^T \tag{2.16}$$

donde $e = [1, ..., 1]^T$. Cabe hacer notar que la forma de centrar los datos en un enfoque de etiqueta simple sería:

$$\tilde{X} = X(I - \frac{mee^T}{n}) \tag{2.17}$$

También debemos definir $W = \text{diag}(w_1, ...w_k)$, donde $w_k = \sum_{i=1}^{n} Z_i k$ es el peso de la k-ésima clase. Para el enfoque de etiqueta simple $w_k = n_k$, que representa el número de ejemplos en la k-ésima clase. Ahora podemos definir S_b de la siguiente forma:

$$S_b = \tilde{X}\ ZW^{-1}Z^T\tilde{X}^T \tag{2.18}$$

Después redefinimos S_t de la forma siguiente. Sea $L = \text{diag}(l_1, ..., l_n)$ donde $l_i = \sum_{i=1}^{K} Z_{ik}$. La forma análoga de esta igualdad en el aprendizaje de etiqueta simple es $L = I$, dado que cada ejemplo sólo pertenece a una clase. De esta forma tenemos:

$$S_t = \tilde{X}\ L\tilde{X}^T \tag{2.19}$$

El objetivo de MLDA es similar al de LDA clásico, encontrar una transformación a un subespacio donde se maximice la razón de la variabilidad entre-grupos con respecto de la variabilidad intra-grupos (ecuación 2.20), la única diferencia es que MLDA agrega la correlación entre las etiquetas asociadas a cada instancia que pertenecen a un mismo grupo.

$$S_w^+ S_b G = \lambda G \tag{2.20}$$

Donde S_w^+ es la seudo-inversa de Sw.

Análisis de correlación canónica

Se puede ver al análisis de correlación canónica [Harold, 1936] (del inglés, Canonical-correlation analysis, CCA) como la generalización de la regresión múltiple (RM), en donde la variable dependiente Y ahora es representada por una matriz. El objetivo de CCA es encontrar la máxima correlación entre dos conjuntos de variables que son representadas por dos combinaciones lineal diferentes. Las nuevas variables recibe el nombre de variables canónicas y la relación entre los pares de variables es llamada correlación canónica.

Sea X y Y dos matrices, donde $X_{n\times m}$ representa las variables independientes, $Y_{n\times p}$ las variables dependientes y $U = Xa,\ V = Yb$ sus proyecciones lineales respectivamente. El objetivo de CCA es encontrar los coeficientes de las proyecciones tal que la correlación entre estos (a y b) sea máxima. Si S_x, S_y y S_{xy} representan las matriz de varianza-covarianza de los conjuntos X, Y y X con Y, respectivamente, entonces, las matrices análogas de varianza-covarianza en la proyección serán $S_u = a'S_x a = 1$, $S_v = b'S_y b = 1$, en las cuales se ha impuesto la restricción de varianza unitaria. Por lo anterior se puede obtener la correlación entre U y V como $a'S_{xy}b$, la cual se desprende de la formula para obtener el coeficiente de correlación de Pearson, el cual se define como: $\rho_{xy} = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}}$. Lo cual reduce el problema a maximizar $a'S_{xy}b$ con la restricción de varianza unitaria $a'S_x a = 1$, $b'S_y b = 1$. Este problema de optimización se resume a encontrar los r valores propios de la matriz a optimizar, donde r es el mín(m, p), de los cuales se toman los q valores propios más grandes. Para los r valores propios esta impuesta la restricción de ortogonalización.

Mínimos cuadrados parciales ortogonales

El método de mínimos cuadrados parciales ortogonales [Wold, 1966] (del inglés, Orthogonal Partial Least Squares, OPLS) es una generalización de PCA tradicional, el cual tiene la ventaja de ser supervisado y tomar en cuenta la relación del conjunto de variables predictoras X y el conjunto de variables de respuesta Y. En esta técnica se combina PCA y el análisis de regresión múltiple. A diferencia de PCA, OPLS modifica la dirección de máxima variabilidad de las variables latentes de X para maximizar la covarianza entre dichas variables y las variables latentes del conjunto Y. Del mismo modo que CCA, PLS crea dos combinaciones lineales diferentes $X = P_x a$ y $Y = P_y b$ con máxima varianza como lo hace PCA, posteriormente busca una nueva proyección $P_x = cP_y$ con máxima correlación mediante una regresión múltiple. Sin embargo, para el caso de OPLS la dirección de máxima varianza esta sesgada debido a la proyección de máxima correlación la afecta ya que la búsqueda de las tres proyecciones se realiza al mismo tiempo y dichas proyecciones están sujetas a la restricción de ortogonalización. La solución del problema se puede obtener mediante la descomposición de valores singulares de la matriz de covarianza S_{xy}, el número de valores propios es el mín(m, p), del cual podemos tomar los q valores más grandes. OPLS se usa cuando existe correlación entre las variables predictoras o existen más variable predictoras que observaciones.

La naturaleza de los modelos de CCA y PLS, permiten atacar problemas multi-etiqueta a pesar de no haber sido creados con este objetivo. Otra ventaja de los modelos que toman en cuenta algún tipo de relación entres dos conjuntos de variables para realizar la reducción de dimensionalidad, es que al seleccionar el número de variables latentes a tomar en cuenta en la nueva proyección de los datos, el máximo número de variables es el mín(m, p), donde p y q son el número de dimensiones de los conjuntos asociados al problema.

2.4.3. Métodos de clasificación multi-etiqueta

k vecinos más cercanos multi-etiqueta

ML-KNN (Multi-label k-nearest neighbor) es la forma análoga de k-NN en el enfoque de clasificación tradicional. La idea de ML-kNN es la siguiente [Zhang and Zhou, 2007]:

Sea χ un dominio de instancias y sea $\Psi = \{1, 2, ..., Q\}$ un conjunto de etiquetas asociadas a las instancias de χ de tal forma que existe un conjunto de entrenamiento $T = [(x_1, Y_1), (x_2, Y_2), ..., (x_m, Y_m)]$ tal que $(x_i \in \chi, Y_i \in \Psi)$.

Dada una instancia x y un conjunto de etiquetas asociadas a dicha instancia $Y \in \Psi$, donde $\vec{y}_x$ es el vector categórico de x. El l-ésimo componente de $\vec{y}_x(l)(l \in \Psi)$ toma el valor de 1 si $l \in Y$ y 0 en otro caso. Adicionalmente suponemos que $N(x)$ denota el conjunto de k vecinos más cercanos de la instancia x identificados del conjunto de entrenamiento T. De esta forma, basado en el conjunto de vecinos más cercanos, el vector que representa el conteo de miembros por clase se define como:

$$\vec{C}_x(l) = \sum_{a \in N(x)} \vec{y}_a(l), \text{con } \ l \in \Psi$$

donde $\vec{C}_x(l)$ cuenta el número de vecinos de x pertenecientes a la l-ésima clase. Para cada instancia t en el conjunto de prueba, ML-kNN primero obtiene los k vecinos más cercanos $N(t)$ del conjunto de entrenamiento T. Sea H_1^l el evento en el que t tiene etiqueta l, y sea H_0^l el evento en el cual t no esta asociado a la etiqueta l. También tenemos que E_j^l $(j \in 0, 1, ..., k)$ representa el evento que, entre los k vecinos más cercanos de t existen exactamente j instancias asociadas a la etiqueta l. Basados en el vector de conteo de pertenencia de clase $\vec{C}_x(l)$, el vector de categorías $\vec{y}_t$ es determinado utilizando el principio de máximo a posteriori (MAP):

$$\vec{y}_t(l) = max_{b \in \{0,1\}} P(H_b^l | E^l_{\vec{C}_t(l)}), l \in \Psi$$

donde $P(H_b^l | E^l_{\vec{C}_t(l)})$ es la probabilidad a posteriori de H_b^l, dado $E^l_{\vec{C}_t(l)}$. En otras palabras, esto quiere decir que, dada una instancia t, se exploran las probabilidades de pertenencia y no pertenencia de t con todas la etiquetas, dado que para t, existen j instancias en el conjunto de sus k vecinos más cercanos que están asociadas a la etiqueta l.

Así usando la regla de Bayes la ecuación MAP se puede escribir como:

$$\vec{y}_t(l) = max_{b \in \{0,1\}} P(H_b^l) P(E^l_{\vec{C}_t(l)} | H_b^l)$$

De esta manera, la información que se requiere para determinar el vector de pertenencia de clase son las probabilidades a priori y la probabilidad condicional. Ambas probabilidades se puede calcular directamente del conjunto de entrenamiento basándose en un conteo de frecuencia.

2.4.4. Métricas de evaluación

Las métricas existentes para evaluar el desempeño de los modelos de predicción se divide en dos categorías: basada en etiquetas y basadas en ejemplos. Al evaluar el rendimiento de un clasificador tradicional se utiliza un enfoque basado en etiquetas. Se evalúa cada instancia por separado y se promedia dividiendo por el número total de instancias, si la salida del modelo es la misma que la clase asociada a la instancia procesada, la predicción es correcta, e incorrecta en otro caso. Esta idea se puede utilizar para medir el desempeño de clasificadores multi-etiqueta, sin embargo, sólo basta con que una entrada del vector de etiquetas no coincida con el vector de clases real para que el ejemplo sea tomado como incorrecto. Lo anterior resulta en una métrica demasiado estricta [Godbole and Sarawagi, 2004, Schapire and Singer, 2000].

Las métricas diseñadas para modelos multi-etiquetas se encuentran en la segunda categoría, una medida evalúa la aportación de cada etiqueta por separado, posteriormente obtiene el promedio, ya sea siguiendo el criterio macro-averaging o micro-averaging.

En el criterio macro-averaging, la métrica se calcula para cada uno de los ejemplos y posteriormente se obtiene un promedio para obtener una medida global. En cambio, si se utiliza un criterio micro-averaging los resultados se dividen por etiquetas, y posterior se suman y se divide entre el número de etiquetas. En el aprendizaje multi-etiqueta, la respuesta de un clasificador puede ser un vector binario, en el cual se indica la asociación

de cada instancia con el conjunto de etiquetas. La otra forma que puede tomar el vector son números reales, los cuales indican el grado de relevancia de cada etiqueta de acuerdo a una función ranking o un conjunto de probabilidades de pertenencia. Para transformar esta salida a un vector binario se puede utilizar un umbral o función de corte, así, el resultado final mantiene el modelo multi-etiqueta. Si el vector de salida es binario, se puede utilizar alguna de las categorías anteriores, las cuales realizan un conteo de aciertos y fallos para lograr evaluar el rendimiento. Sin embargo, si lo que queremos es trabajar con un vector ranking, debemos de utilizar métodos que realicen los cálculos utilizando grados de relevancia para obtener una medida de desempeño equivalente [Ghamrawi and McCallum, 2005, Tsoumakas and Vlahavas, 2007, Tsoumakas et al., 2010].

Métricas binarias basadas en ejemplos

Perdida Hamming (Hamming loss): Es la medida más utilizada en la literatura multi-etiqueta. Hamming loss toman en cuenta los errores de clasificación y los errores de omisión, realiza una suma del número de veces que se presentan estos errores en cada instancia y posteriormente los divide entre el número de etiquetas m. Para obtener el cálculo de los errores utiliza la diferencia simétrica Δ (función XOR) del conjunto de etiquetas de salida O_i y el conjunto de etiquetas real Y_i. El mejor rendimiento es para resultados cercanos a cero.

$$Hamming\ loss = \frac{1}{m}\sum_{1}^{m}|O_i \Delta Y_i| \tag{2.21}$$

Precisión (Precision): Se define como el promedio de etiquetas acertadas con respecto de las predichas.

$$\text{Precision} = \frac{1}{m}\sum_{1}^{m}\frac{O_i \cap Y_i}{O_i} \tag{2.22}$$

Exactitud (Accuracy): se define como el promedio de la fracción de aciertos del clasificador frente a la unión de etiquetas reales y predichas.

$$\text{Accuracy} = \frac{1}{m}\sum_{1}^{m}\frac{O_i \cap Y_i}{Y_i \cup O_i} \tag{2.23}$$

Métricas de ranking

Error de máxima jerarquía (one error): Esta métrica determina cuantas veces la etiqueta λ que ocupa el primer puesto en el ranking $r(\lambda) = 1$ de la salida del clasificador

no se encuentra en el conjunto de etiquetas reales Y_i asociadas a la instancia x_i:

$$\text{One error} = \frac{1}{m}\sum_{1}^{m}\delta(\lambda_{max_{O_i}}, Y_{x_i}) \tag{2.24}$$

donde $\lambda_{max_{O_i}}$ es la etiqueta asociada a la posición 1 del ranking de salida O_{x_i} del clasificador al procesar la instancia x_i, la función δ se define de la siguiente forma:

$$\delta(\lambda, Y_i) = \begin{cases} 1 & \text{si } \lambda \in Y_i \\ 0 & \text{de otra forma.} \end{cases} \tag{2.25}$$

Cobertura (Coverage): Esta medida obtiene el número de pasos que se deben avanzar en la lista de ranking devuelta por el clasificador para cubrir todo el conjunto de etiquetas asociadas a cada instancia.

$$\text{Coverage} = \frac{1}{m}\sum_{1}^{m} max_{step}(O_i, Y_{x_i}) \tag{2.26}$$

Pérdida de jerarquía (Ranking loss) : Es el promedio de veces que al comparar pares de etiquetas $(y_1, y_2) \in Y_i \times \bar{Y_i}$ asociadas y no asociadas a una instancia, una etiqueta no relevante tiene mejor posición en el ranking que su par correspondiente (etiqueta asociada).

$$\text{Ranking} \quad \text{loss} = \frac{1}{m}\sum_{1}^{m} \left\| y_a, y_b : rank(x_i, y_a) > rank(x_i, y_b), (y_a, y_b) \in Y_i \times \bar{Y_i} \right\| \tag{2.27}$$

Precisión promedio (Average precision): Define la medida para cada etiqueta relevante de las instancias la proporción de etiquetas que están delante de ellas en el ranking, esto es, cuantas posiciones en promedio debemos avanzar en la lista del ranking para encontrar una etiqueta asociada a las instancias clasificadas.

$$\text{Average} \quad \text{precision} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{Y_i}\sum_{y \in Yi}\frac{\|y' | rank(x_i, y') \leq rank(x_i, y), y' \in Y_i\|}{rank(x_i, y)} \tag{2.28}$$

2.5. Receptores acoplados a proteínas G

En esta sección se describe de forma general la estructura celular, y los receptores celulares los que se unen a las proteínas en el interior de la célula. También, se explora la forma en que la célula se comunican y activan ciertas funciones al interactuar con algún tipo de ligando. Los receptores acoplados a proteínas G son los receptores celulares de interés dentro de esta sección.

2.5.1. Introducción

A cada instante de tiempo de nuestro ciclo biológico, nuestro cuerpo interactúa con el medio que lo rodea. Desde el acto más común y mecánico hasta el más complejo, nuestros sentidos reciben señales y generan respuestas. La forma en que se reciben estas señales pasa de forma desapercibida por ser un proceso a nivel celular. Miles de millones de células dentro de nuestro organismo trabajan de manera coordinada. Para lograr esta coordinación, las células envían señales que generan respuestas. De esta forma logran comunicarse entre ellas y/o hacia el mundo exterior [Katritch et al., 2014].

Las ciencias ómicas involucran diferentes disciplinas cuyo objeto de estudio son estructuras biológicas a nivel molecular, el modelar estas estructuras involucra una extensa cantidad de datos que son necesarios para entender como se interrelacionan sus diferentes componentes [Overington et al., 2006]. Una de estas disciplinas es la Proteómica, la cual estudia las formas de obtención de información funcional de las proteínas. Para el estudio de estos elementos se auxilia de una rama de la ciencia que responde al nombre de Bioinformática. El conocimiento que se genera de las investigaciones en esta área, nos permite el análisis y comprensión de las funciones de muchas macromoléculas y organismos de los cuales su rol no esta completamente definido [Kahn, 2011]. Una de las principales áreas de investigación en bioinformática trata sobre la creación de nuevos fármacos y el análisis de proteínas.

A	ALA	Alanina	M	MET	Metionina
C	CYS	Cisteína	N	ASN	Asparagina
D	ASP	Aspartato	P	PRO	Prolina
E	GLU	Glutamato	Q	GLN	Glutamina
F	PHE	Fenilalanina	R	ARG	Arginina
G	GLY	Glicina	S	SER	Serina
H	HIS	Histidina	T	THR	Treonina
I	ILE	Isoleucina	V	VAL	Valina
K	LYS	Lisina	W	TRP	Triptófano
L	LEU	Leucina	Y	TYR	Tirosina

Figura 2.6: Lista de los 20 aminoácidos nativos [Mathews et al., 2013]

Las proteínas, junto con los ácidos nucleicos y los polisacáridos son las tres biomoléculas en donde recae la mayoría de los procesos químicos necesarios para la vida. De la misma forma que se unen los nucleótidos (monómeros) para formar estructuras tridimensionales de ADN (polímeros), las proteínas son polímeros y los aminoácidos son los monómeros que se unen para formarlas. Existen miles de proteínas diferentes en cada ser vivo, formadas por una combinación de nueve aminoácidos esenciales y once no esenciales (ver figura 2.6). Los aminoácidos esenciales son aquellos que el organismo no puede sintetizar por si mismo y los adquiere por medio de la ingesta de alimentos. En cambio, los aminoácidos no esenciales son aquellos que el cuerpo puede sintetizar tomando como base a los aminoácidos esenciales. Los aminoácidos se unen entre ellos a través de enlaces peptídicos, a la unión de pocos (no más de 100) aminoácidos se les llama peptídos, cuando una cadena de aminoácidos contiene más de 100 elementos se les conoce como proteínas. Por ejemplo: la mioglobina, la proteína responsable del almacenamiento de oxígeno en los seres vivos contiene 153 aminoácidos. Es considerada una de las proteínas más pequeñas, debido a que existen proteínas con secuencias de miles de aminoácidos.

Al sintetizarse la secuencia de aminoácidos cada proteína en particular adquiere una secuencias específica. la posición de cada aminoácido debiera estar codificada según la función asociada a la proteína. Debido a la información oculta en las secuencias se puede considerar a las proteínas como codificadoras de información biológica. La secuencia de aminoácidos que forman a una proteína es denominada estructura primaria o secuencia nativa. A partir de la estructura primaria se pueden formar estructuras de orden superior como estructura secundaria (hélices o láminas), terciaria (doblaje de la estructura secundaria en el espacio 3D) y cuaternaria (agrupación de estructuras terciarias). La estructura tridimensional de cada proteína nos indica las funciones biológicas concretas que realiza. Cuando su estructura 3D no esta disponible se puede utilizar su estructura primaria [Bockaert and Pin, 2000].

Las proteínas G son una clase de proteínas dedicadas a la transmisión de señales, deben su nombre a la interacción que tienen con la Guanosina. Todas las células necesitan recibir señales para activar o desactivar sus proceso asociados, debido a esto, todas las células contienen proteínas G. Para que las proteínas G dentro de la célula puedan recibir señales del exterior, las células cuentan con receptores. Los receptores asociados (acoplados) a este tipo de proteínas se encuentran en la membrana celular, y su función es conectar el interior de la célula con el exterior.

Los receptores acoplados a proteínas G (G-Protein coupled receptors, GPCRs) son una súperfamilia de receptores intracelulares, una de las más grandes del genoma humano, estos receptores responden a estímulos de olor, sabor, luz, serotonina y adrenalina por citar algunos. En la figura 2.7(1) se puede observar una célula, a la derecha 2.7(2), la membrana celular, la cual contiene los receptores celulares (azul), a los cuales se adiheren los ligandos que activan a las proteinas G (rojo). El 50 % de los principios bioactivos presentes en los medicamentos que se comercializan hoy en día están dirigidos a estos receptores, por esta motivo su estudio es de gran importancia para la industria farmacéutica [Overington et al., 2006]. Estos receptores juegan un rol muy importante debido a que regulan y transducen una amplia cantidad de funciones y señales, respectivamente. Todos los GPCRs comparten una estructura común, su identificación, clasificación y separación por familias

y subfamilas de acuerdo a sus secuencias es una tarea principal en Bioinformática [Flower and Attwood, 2004].

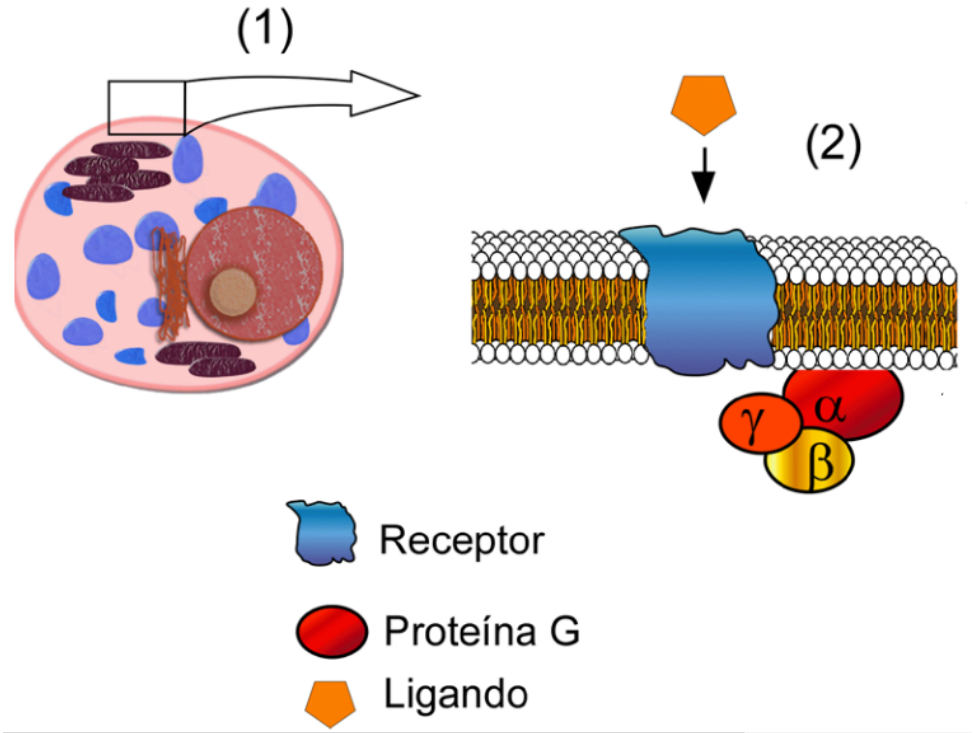


Figura 2.7: Célula y proteína G. (1) Célula. (2) Membrana celular, el ligando se muestra en amarillo, el receptor en azul y la proteína G en rojo (subunidad α), naranja (subunidad γ) y ambar (subunidad β) [Gonzales Gil, 2013].

Sin embargo, el estudio de estos receptores se complica por diversos factores como son:

- Las pequeñas cantidades en las que se encuentran en las células.
- El proceso de purificación necesario para su estudio.
- Las múltiples funciones que realizan.
- La divergencia de sus cadenas.[1]
- La variabilidad en la longitud de las secuencias.

Una amplia gama de enfermedades estan relacionadas con el funcionamiento anómalo de cierto tipos de proteínas. Este funcionamiento anómalo hace que las proteínas no cumplan su función correctamente, es aquí donde los fármacos juegan un rol esencial, ya que están diseñados para solucionar este tipo de problemas. La sustancia activa de los fármacos son

[1]La divergencia entres dos proteínas se entiende como el porcentaje de posiciones en las que poseen aminoácidos diferentes [Lewin et al., 2009].

macromoléculas (ligandos), las cuales se adhieren a los GPCRs de las células activándolas [2] o antagonizandolas [3] [Panettaand and Greenwood, 2008].

De la misma forma los GPCRs detectan y procesan señales del mundo exterior relacionadas con los sentidos (visión, gusto y olfato), los receptores al exterior de la célula, alteran moléculas intracelulares creando respuestas celulares diversas de acuerdo a un ligando específico. Por ejemplo, la hemoglobina, una proteína que se encuentra en la sangre, tiene como ligando las moléculas de oxígeno, cuando la hemoglobina se une a una molécula de oxígeno esta molécula la activa para que la proteína pueda transportar el oxígeno desde los pulmones hasta los tejidos. Si el ligando que activa a la hemoglobina es el dióxido de carbono, éste es transportado desde los tejidos a los pulmones para ser eliminado mediante la exhalación. A la transferencia de información al interior de la célula se lo conoce como transducción de señales [Palczewski, 2006, Stenkamp et al., 2002].

2.5.2. Clasificación de GPCRs

Los GPCRs se han clasificado de varias formas de acuerdo a ciertos criterios como: el sitio de unión de su ligando, características funcionales, estructurales o genéticas. El sistema de clasificación GRAFS (Glutamate, Rhodopsin, Adhesion, Frizzled/Taste2, Secretin) los divide en cinco familias de acuerdo a su filogenia (ver cuadro 2.1). Cada familia se divide en tipos y subtipos de acuerdo a agentes comunes [Kolakowski, 1994]. Los ligandos de los GPCRs son altamente variables, sin embargo, comparten características similares [Fredriksson and Schiöth, 2003].

Familia	Descripción
A	Familia de receptores de rodopsina
B	Familia de receptores de secretina
C	Familia de receptores de metabotrópicos de glutamato/feromonas
D	Familia de receptores de feromonas
E	Familia de receptores de AMP cíclico

Cuadro 2.1: Clasificación de los GPCRs

La familia/clase C de los GPCRs contiene los receptores de metabotrópicos de glutamato, los receptores $GABA-B$, los receptores vomeronasales, receptores de feromonas y los receptores del gusto (ver cuadro 2.2). Esta familia se divide en siete subtipos [Fredriksson and Schiöth, 2003], y debido a su relación con áreas como el dolor, ansiedad, trastornos neurodegenerativos, osteoporosis, autismo, depresión entre otras enfermedades, se han convertido en el objeto de estudio para nuevos fármacos [Barnes, 2006, Doré et al., 2014].

[2] Agonismo: ligando que activa un receptor.

[3] Antagonismo: ligando que bloquea un receptor impidiendo su activación.

Subfamilia	Descripción
Tipo 1	Sensores de calcio (Calcium sensing, CS)
Tipo 2	GABA-B (GB)
Tipo 3	Metabotrópicos de glutamato (Metabotropic glutamate, mG)
Tipo 4	Odorantes (Odorant, Od)
Tipo 5	Feromonas (Phermone, Ph)
Tipo 6	Receptores del gusto (Taste, Ta)
Tipo 7	Vomeronasal (Vomeronasal, VN)

Cuadro 2.2: Subfamilas de la clase C de los GPCRs.

2.5.3. Multifuncionalidad de las proteínas

A partir de la introducción de técnicas computacionales en biología molecular, es posible obtener secuencias de genes y proteínas de forma automática. Con estos estudios se ha logrado comprender mejor las relaciones que mantiene la estructura de ciertas biomoléculas con la función que realizan [Atchley et al., 2005]. Poder conocer la estructura de este tipo de moléculas no es un proceso trivial, actualmente la técnica más eficaz es por medio de difracción de rayos X sobre cristales (cristalización). Con este método, sumamente complicado, se logró obtener en el año 1962 la primera estructura $3D$ de una proteína [Orozco, 2014]. Algunas proteínas se cristalizan fácilmente, sin embargo, si las condiciones no son propicias esta tarea puede complicarse en gran medida. De forma general, es necesario experimentar cientos de condiciones para obtener cristales puros [Giacovazzo, 2011]. Inicialmente, al lograr obtener la estructura $3D$ de algunas proteínas mediante cristalización se asumió una estructura rígida para todas las secuencias existentes, de tal modo que se postuló que una proteína sólo podía ser funcional si su estructura tridimensional esta bien definida [Berg et al., 2002]. Esta idea se mantuvo por varios años en áreas afines de la ciencia, sin embargo, en investigaciones posteriores se observaron secuencias de proteínas de las cuales su estructurara terciaria era incapaz de explicar resultados experimentales. Con este descubrimiento, se llegó a la conclusión de que existen proteínas que no cuentan con una estructura estable. Al principio, debido a que la cantidad de proteínas de las cuales se había logrado obtener su estructura 3D eran escasas, estas regiones no estructuradas sólo se habían encontrado en regiones específicas, las cuales fueron consideradas como conectores entre regiones con estructura definida o dominios funcionales específicos. A medida que las investigaciones avanzaron en esta área, se logro la cristalización de un mayor número de proteínas, con lo cual se observó que estas áreas flexibles no sólo abarcaban pequeñas regiones de las secuencias, sino que existían secuencias las cuales contenían regiones con estructura no definida que variaban en tamaño y número [Tompa, 2002]. Posteriormente, se descubrieron secuencias para las cuales estas regiones abarcan la mayor parte de la longitud de la cadena. A partir de esos resultados, se derivó que el concepto clásico de estructura-función no era del todo cierto, ya que bajo este concepto, este tipo de proteínas no ejercían alguna función. Sin embargo, ya se había demostrado que proteínas con este tipo de estructura no definida están involucradas en diferentes vías

de señalización y algunas familias de proteínas con este tipo de estructura son abundantes en diferentes procesos bioquímicos vitales para diferentes organismos vivos. Este tipo de proteínas recibieron el nombre de proteínas intrínsecamente desordenadas (del inglés Intrinsically Disordered Proteins, IDP) [Tompa, 2002, Ward et al., 2004].

Una hipótesis que justifica la multifuncionalidad de las proteínas esta basada en la evolución de la célula, la cual plantea que al aumentar la complejidad de los organismos vivos para su adaptación evolutiva, aumenta la capacidad funcional de las proteínas. Este incremento funcional trae como consecuencia el incremento en el tamaño de las secuencias de aminoácidos o la creación de nuevas proteínas, lo cual probablemente afectaría algunas funciones celulares, pues disminuiría el agua disponible, entre otros efectos adversos. Debido a este hecho, la estructura funcional de las proteínas se vio obligada a tomar diversas estructuras dependiendo de las condiciones microambientales y de la presencia de diversos ligandos [Vogel and Chothia, 2006].

Esta evolución multifuncional de las proteínas incrementa la capacidad para almacenar información genética y aumenta la respuesta adaptativa de las células en un periodo de tiempo menor, lo que hace que los organismos vivos tengan una evolución más exitosa. Esta explicación, asume que el incremento de la complejidad funcional en una célula no esta dirigida a la creación de nuevos genes, sino en hacer que los genes codifiquen proteínas flexibles que permitan multifuncionalidad [Schad et al., 2011]. Las funciones secundarias asociadas a una proteína no necesariamente tienen que estar relacionadas con la función original. De igual forma, el orden histórico de la identificación en las funciones no tiene relevancia ni implica la perdida de alguna otra función.

Existen diferentes definiciones de multifuncionalidad en el estado del arte como son moonlighting protein [Jeffery, 1999, 2003], gene sharing [Piatigorsky, 2007], promiscuous protein [Nobeli et al., 2009], multitasking protein [Jeffery, 2004]. Estas definiciones no son sinónimos en si, ya que difieren en algunos aspectos, en esta investigación utilizaremos el concepto de multifuncionalidad (moonlighting) o multi-etiqueta como sinónimos.

Actualmente, se sabe que las proteínas multifuncionales están asociadas a receptores membranales, proteínas de andamiaje, proteínas del citoesqueleto, factores transcripcionales y receptores nucleares de hormonas, entre otras [Ward et al., 2004]. En la literatura existen referencias a la multifuncionalidad de los GPCRs [Borroto-Escuela et al., 2011, Fuxe et al., 2014, Wieland and Mittmann, 2003], sin embargo los artículos citados sólo hacen suposiciones ya que no se ha logrado comprobar estos supuestos mediante estudios formales. Identificar la función asociada a una proteína es un proceso complicado, pero más lo es demostrar una doble o triple función [Koonin and Galperin, 2003]. Existen métodos que sirven para indicar la propiedad multifuncional de una proteína, la mayoría de estos son procesos químicos complejos, sin embargo, también es posible hacerlo bionformáticamente mediante programas de predicción de dominios (ligandos) [Becker et al., 2012, Chapple et al., 2015a,b, Gómez et al., 2003, 2011, Khan and Kihara, 2014, Khan et al., 2012, 2014], que es la idea en la que se basa este trabajo.

Las proteínas multifuncionales están asociadas a enfermedades tan serias como el cáncer, desórdenes neurodegenerativos, entre otras enfermedades [Iakoucheva et al., 2002]. Además, la aplicación de la bioinformática para el estudio masivo de este tipo de datos ha permitido obtener información que indica la existencia de proteínas con este tipo de estructura.

2.5.4. Representación de información biológica

Un problema fundamental de la bioinformática esta relacionado en la forma de representar los datos en forma numérica, partiendo de modelos de representación alfabética, como lo es el caso de las secuencias de proteínas. El uso directo de códigos alfabéticos, los cuales no contiene una métrica implícita de comparación, trae como consecuencia que no se vea reflejada una relación directa entre la similitud de las estructuras. La perdida de información de las propiedades físico-químicas y relación de orden es notoria al ser comparadas dos secuencias [Atchley et al., 2005].

La conformación de largas secuencias alfabéticas y variabilidad del numero de aminoácidos que las componen, demanda el uso de sofisticados análisis estadísticos para el correcto estudio de las estructura y aspectos funcionales de las secuencias. A medida que se incrementa la calidad, como la cantidad de información en un modelado numérico, aumenta la precisión al comparar secuencias, lo que facilita el proceso de toma de decisiones.

El modelado de secuencias alfabéticas a una forma numérica se conoce como representación de secuencias de aminoácidos. Como ya se mencionó, cada proteína presenta una secuencia específica en orden y repetición de aminoácidos (secuencias nativas). Esta secuencia se llama estructura primaria y contiene la información necesaria para que las secuencia adquieran formas más complejas como son la estructura secundaria y tridimensional [Smith and Waterman, 1981].

```
TSDHGWALGEHGEWAKYSNFDVATHVPLIFYVPGRTASLPEAGEKLFPYLDPFD
SASQLMEPGRQSMDLVELVSLFPTLAGLAGLQVPPRCPVPSFHVELCREGKNLK
HFRFRDLEEDPYLPGNPRELIAYSQYPRPSDIPQWNSDKPSLKDIKIMGYSIRTIDY
RYTVWVGFNPDEFLANFSDIHAGELYFVDSDPLQDHNMYNDSQGGDLFQLLMP
```

Figura 2.8: Subcadena de una secuencia de aminoácidos

La figura 2.8 muestra un ejemplo de subcadena de una secuencia de aminoácidos. Para que los algoritmos de machine learning pueda realizar su labor de predicción, es necesario un modelado en el que sus componentes más importantes puedan ser representados de forma computacional. Una secuencia de aminoácidos puede ser representada por medio de un vector residente en un espacio n-dimensional, aunque también es posible representarla por medio de un grafo [**??**].

Los métodos de representación mediante vectores se dividen en vectores basados en características [**?**] y vectores basados en similitud [**??**]. Los vectores basados en extracción de características son libres de alineamiento, y la extracción de basa en algún criterio específico. El segundo método toma como base la comparación y alineamiento de las secuencias nativas mediante un criterio de similitud para extraer diversas propiedades físico-químicas, frecuencia de ocurrencia y correlación. Existen dos enfoques para lograr el alineamiento de secuencias: el primero de ellos utiliza programación dinámica para lograr una solución de alineación a través de una optimización global [Smith and Waterman, 1981]; el segundo enfoque, y el más utilizado para el alineamiento de secuencias, se basa en métodos heurísticos [**?**], los cuales, a pesar de no obtener soluciones necesariamente óptimas, ofrecen tiempos de cálculo aceptables, debido a que estos no crecen de manera exponencial conforme aumenta el tamaño de la secuencia.

2.5.5. Representaciones de secuencias de aminoácidos

Las diferentes formas de representación de secuencias de aminoácidos difieren en la cantidad de características que logran extraer de las secuencias nativas. Debido a esto, entre mejor poder discriminante o factor explicativo almacene la nueva forma de representación se obtendrá una clasificación más adecuada, que asocie a cada proteína con sus funciones biológicas.

Composición de aminoácidos

La composición de aminoácidos (AAC) es el método más simple y la forma más sencilla de representar a una proteína, tomando como base su secuencia nativa [Cruz-Barbosa et al., 2015]. Si partimos de una secuencia de aminoácidos S tal que:

$$S = [a_1, a_2, a_3, ..., a_n] \tag{2.29}$$

donde cada a_i con $i = 1, ..., n$ representa un aminoácido, la nueva representación se obtendrá al realizar un conteo de la cantidad de veces que aparece cada uno de los 20 aminoácidos existentes. La nueva representación queda de la siguiente forma:

$$AAC = [f_1, f_2, f_3, ..., f_{20}] \tag{2.30}$$

dondef_j representa la frecuencia relativa de ocurrencia del j-ésimo aminoácido. Al utilizar esta representación, la cual no toma en cuenta el orden de aparición de los aminoácidos en la secuencia, se pueden obtener resultados muy similares para secuencias muy diferentes.

Pseudo-composición de aminoácidos

La Pseudo-Composición de Aminoácidos (PseAAC) es ampliamente utilizada para predecir diversos atributos de las proteínas. Esta transformación representa mejor las característica de una secuencia de proteína a través de un modelo discreto. PseAAC toma en cuenta la información de orden de aparición de los aminoácidos en una cadena, incorporando esa información a la representación [Nanni et al., 2012].

PseAAC almacena en sus primeras 20 posiciones la información de frecuencia relativa, tal como los hace AAC, y en las posiciones restantes almacena factores adicionales que representan información de algún tipo de relación de orden entre los aminoácidos de la cadena. La representación de esta transformación puede verse de la siguiente manera:

$$PseAAC = [f_1, f_2, f_3, ..., f_{20}, ..., P_\alpha]^T \tag{2.31}$$

α, que es la longitud de la transformación, siempre es menor que la longitud real de la secuencia. El valor que toma α esta directamente relacionado con el número de niveles λ y con el número n de propiedades físico-químicas tomadas en cuenta a lo largo de los λ niveles [Nanni et al., 2012]. Para realizar las transformación es necesario capturar para cada aminoácido en la secuencia, n propiedades específicas de los λ aminoácidos contiguos (ver figura 2.9). La longitud α de una cadena se puede expresar como: $\alpha = 20 + n\lambda$

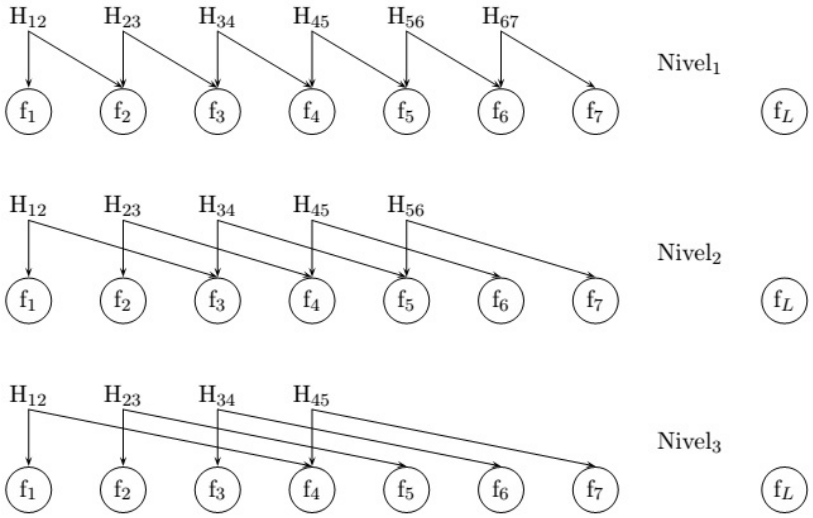


Figura 2.9: Niveles de correlación de orden de la secuencia de proteína [Nanni et al., 2012, Ramos Pérez, 2016].

Wavelet basado en energía multiescala y PseAAC

Esta forma de representación es un vector híbrido que contiene la concatenación de PseAAC y un vector de energía multiescala MSE. Para obtener este vector cada elemento de la secuencia de aminoácidos es remplazado por su valor hidrofóbico correspondiente en la escala FH. El vector resultante de esta sustitución es un vector numérico, análogo a una señal digital. A este nuevo vector se le aplica el algoritmo Mallat [Mallat, 1989] para obtener una transformación wavelet discreta (DWT). Este tipo de DWT se basa en la representación de señales aplicando bancos de filtros pasa bajas (coeficientes de aproximación) y filtros pasa altas (coeficientes de detalle) [Ur-Rehman and Khan, 2011]. Al aplicar el algoritmo Mallat al vector numérico de la secuencia se obtienen los coeficientes de aproximación y detalle, el vector MSE queda de la siguiente manera:

$$MSE(k) = [d_1^k, d_2^k, d_3^k, ..., d_m^k, a_m^k] \tag{2.32}$$

donde d_m^k, es la raíz cuadrada de la media de la energía de los coeficientes de detalle, a_m^k es la raíz cuadrada de la media de la energía de los coeficientes de aproximación. m representa la m-ésima escala, donde la escala significa el nivel de descomposición, el cual depende del tamaño de la secuencia y se obtiene al aplicar $\log_2$ a su longitud [Ur-Rehman and Khan, 2011].

Para el caso de las proteínas, los componentes de baja frecuencia son funcionalmente más importantes, al igual que PseAAC, esta transformación conserva información de orden en la secuencia. Al final se concatenan los resultados:

$$MSE - PseAAC = [f_1, f_2, f_3, ..., f_{20}, ..., P_\alpha, d_1^k, d_2^k, d_3^k, ..., d_m^k, a_m^k] \tag{2.33}$$

Auto-covarianza y covarianza cruzada

La idea esencial de esta transformación es convertir las secuencias primarias de aminoácidos a un vector de valores reales de forma similar a como lo hace DWT. Estos nuevos valores numéricos se denominan descriptores y representan propiedades físico-químicas [Smith and Waterman, 1981]. Posteriormente se realiza una nueva transformación de los descriptores a una matriz uniforme. Esta transformación contiene la auto covarianza (AC) que mide la correlación de un descriptor d con dos residuos separados en un intervalo lg y la covarianza cruzada (CC), la cual mide la correlación de dos descriptores diferentes entre dos residuos por otro intervalo a través de la secuencia [Cruz-Barbosa et al., 2015, Lapinsh et al., 2002]. La AC y CC son concatenados por medio de intervalos dando lugar a nuevas secuencias $C(lg)$, este nuevo conjuntos de secuencias son concatenados para un nuevo intervalo máximo lg_{max} dando como resultado un vector que tiene la siguiente forma:

$$ACC(lg_{max}) = [C(lg)C(lg), ..., C(lg_{max})] \tag{2.34}$$

esta transformación es libre de alineamiento y conserva mayor información presente en la secuencia, así como la dependencia de orden entre posiciones de aminoácidos vecinos.

Capítulo 3

Desarrollo del proyecto

Es este capítulo se describen las características de hardware, los entornos de desarrollo y la biblioteca de métodos multi-etiqueta desarrollada. También, se describe el esquema general de los experimentos y los módulos del proyecto utilizados para las pruebas de los diferentes algoritmos de clasificación, reducción de dimensionalidad y métricas de evaluación multi-etiqueta.

3.1. Esquema experimental

Para los experimentos se utilizó MATLAB R2010b, S.O. Ubuntu 14.04 a 64 bits. El hardware que se utiliza es una máquina de escritorio con procesador $i7$, 2.5 GHz, 14 Gb de RAM. Para las pruebas de exactitud se utiliza la estrategia de doble validación cruzada estratificada (ver 4.3.2) con 5 particiones externas y 3 particiones internas. Los resultados de las pruebas son el valor promedio de repetir los experimentos 10 veces. El conjunto de datos fue tomado de la base de datos pública GPCRDB [1] [Isberg et al., 2014]. La SVM utiliza un enfoque uno-contra-uno para construir el modelo general de los datos, la implemetanción del código en MATLAB es tomado de la libreria LIBSVM [2] [Chang and Lin, 2011]. La implemetación de los algoritmos de reducción de dimensionalidad tradicional forman parte del toolbox SIMFEAT (simple linear and kernel feature extraction) [3] [Arenas-García et al., 2013]. La versión multi-etiqueta de k-NN [Zhang and Zhou, 2007] es tomada de LAMDA (Learning And Mining from Data)[4]. Los códigos de los algoritmos de reducción de dimensionalidad multi-etiqueta forman parte del toolbox MLDR (Multi-Label Dimensionality Reduction) [5] [Sun et al., 2013].

Los gráficos son obtenidos mediante R v3.2.4 para Rstudio. La biblioteca de métodos de reducción de dimensionalidad y clasificación muti-etiqueta desarrollada en este trabajo se implementó en lenguaje C++ bajo la plataforma Ubuntu utilizando la biblioteca armadillo [Sanderson, 2010].

[1] http://www.gpcr.org/7tm
[2] https://www.csie.ntu.edu.tw/ cjlin/libsvm/
[3] http://isp.uv.es/soft_feature.html
[4] http://lamda.nju.edu.cn/Data.ashx
[5] http://www.public.asu.edu/ jye02/Software/MLDR

3.2. Esquema general

Como se mencionó en la sección 2.2.1, el primer paso para la implementación de un sistema de reconocimiento de patrones es la adquisión de los datos. Para este caso en particular, los datos se obtienen de una base de datos pública. Como dichos datos se encuentran en su forma alfabética, es necesario aplicarles alguna transfomación para obtener vectores númericos que los algoritmos de apredizaje máquina puedan analizar.

Una vez que contamos con información numérica, el siguiente paso es realizar un análisis exploratorio de los datos. El objetivo es observar mediante algún recurso gráfico ciertas propiedades estadísticas. Posteriormente, después de conocer la estructura de los datos se aplican técnicas de reducción de dimensionalidad utilizando un enfoque tradicional. Los datos que se obtienen al aplicar dichas técnicas son analizados de forma gráfica y comparados con los datos análogos de la tranformación sin reducción. Si el número de variables en la estructura de baja dimensión no permite un análisis objetivo de los datos, se utilizan técnicas de clasificación. Mediante estas técnicas se estima si la estructura en baja dimensión logra una mejor discriminación de los datos de acuerdo a su clase.

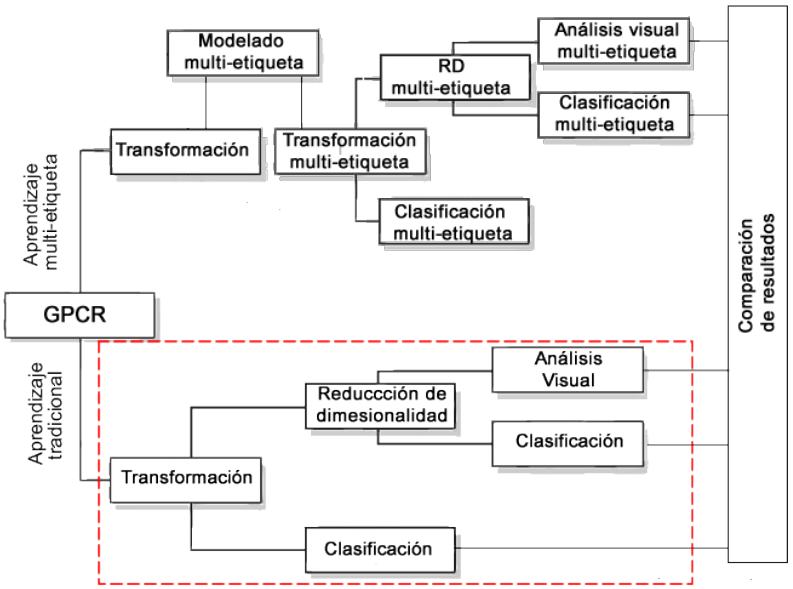


Figura 3.1: Esquema general de los experimentos.

El proceso descrito anteriomente culmina compararando los resultado obtenidos de los métodos utilizados con los reportados en el estado del arte siguiendo el enfoque de apredizaje de etiqueta simple.

En la figura 3.1 se muestra el esquema general de los experimentos realizados en este trabajo, en la parte inferior de la imagen se puede apreciar (contorno rojo) el procedimiento descrito anteriormente, el cual sigue la linea de aprendizaje tradicional. Por otro lado, partiendo de la transformación numérica también se puede realizar el proceso análogo utilizando el aprendizaje multi-etiqueta. La única diferencia estriba en que las transformaciones de secuencias de GPCRs que se encuentran expresadas mediante el modelo de

etiqueta simple se deben modelar a su versión multi-funcional para poder ser analizadas mediante aprendizaje multi-etiqueta.

El modelado multi-etiqueta se presenta de forma detallada en la sección 4.4. Aquí, se observa que el modelo multi-funcional de las secuencias de GPCRs de clase C se lleva acabo mediante el análisis de la matriz de confusión al aplicar una SVM a las transformaciones sin reducción. De dicho análisis se crea un conjunto de datos, en el cual las proteínas estan asociadas a más de una clase (función). Una vez obtenida la representación multi-funcional se puede aplicar técnicas de reducción y clasificación multi-etiqueta de la misma forma como se hizo con las transformaciones que contienen un modelado de etiqueta simple. Este proceso se ilustra en la figura 3.1, en la parte superior, siguiendo la linea de aprendizaje multi-etiqueta. Como parte final de los experimentos se compara el rendimiento de ambos enfoques y se discuten los resultados.

3.3. Módulos del proyecto

El proyecto tiene como objetivo la reducción de dimensionalidad de secuencias de aminoácidos, las cuales han sido transformadas de su representación alfabética a su representación numérica. Las representaciones a las que se han transformado las secuencias son: Composición de aminoácidos (AAC), Pseudo composición de aminoácidos (PseAAC), Wavelet basado en energía multiescala y pseudo composición de aminoácidos (Wavelet-PseAAC) y Auto-covarianza y covarianza cruzada (ACC) [Cruz-Barbosa et al., 2015, Lapinsh et al., 2002, Ramos Pérez, 2016, Ur-Rehman and Khan, 2011]. Las cuatro representaciones de secuencias de aminoácidos son vectores con diferente longitud (20, 62, 74 y 325 respectivamente). Al aplicar métodos de reducción de dimensionalidad se busca una estructura más compacta que realce la información discriminante en la cual se puedan analizar los datos mediante un recurso gráfico. Si los métodos aplicados a las transformaciones reducen el número de características a una dimensión en la que el análisis gráfico no sea objetivo, se utilizan clasificadores para obtener un medida de desempeño y poder comparar la estructura original y reducida de las secuencias. En la figura 3.2 se muestra el diagrama general de bloques de la primera etapa del proyecto, en la cual se utilizan técnicas de reducción y clasificación convencionales.

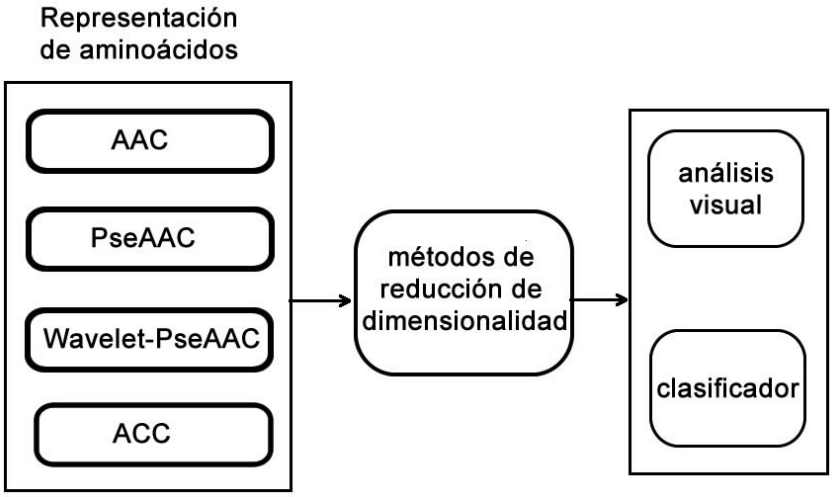


Figura 3.2: Diagrama de bloques para la reducción de secuencias de aminoácidos utilizando un modelado tradicional.

Las secuencias que se analizan en este trabajo forman parte de la familia C de los receptores acoplados a proteínas G, la cual esta categorizada en siete subfamilias. Debido a que existen secuencias de proteínas que pueden interactuar con más de un ligando, se propone un modelado multi-etiqueta para secuencias en las cuales al momento de clasificar o visualizar su distribución de clases exista un grado de certeza de su asociación a otra subfamilia de proteínas de la clase C. La figura 3.3 muestra el diagrama general de bloques del modelado de los datos de su forma convencional a su forma multi-etiqueta.

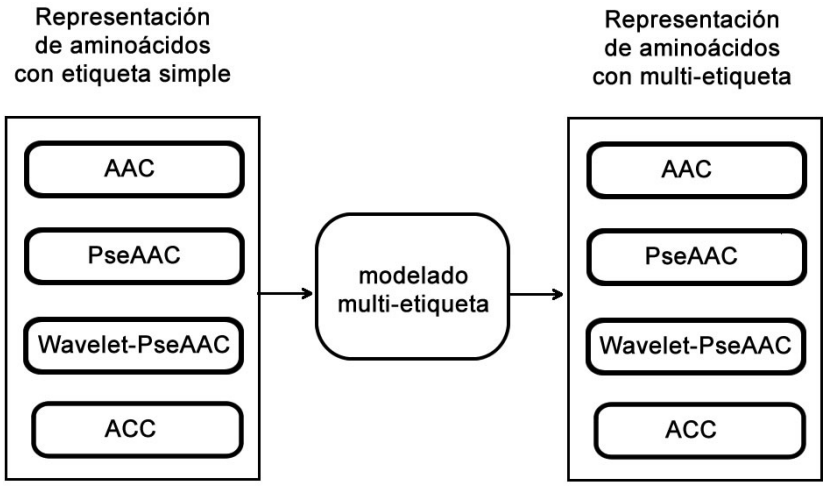


Figura 3.3: Diagrama de bloques del modelado tradicional al modelado multi-etiqueta

Las secuencias de aminoácidos modeladas de manera multi-etiqueta son analizadas con el mismo tipo de pruebas (de reducción de dimensionalidad y clasificación), pero con algoritmos basados en el enfoque multi-etiqueta. Como punto final se comparan los

resultados obtenidos para ambos modelos (simple y multi-etiqueta) y se resaltan los pros y contras de cada estrategia. En la figura 3.4 se observa el diagrama general de bloques del proceso de reducción y clasificación de los conjuntos modelados a su forma multi-etiqueta con algoritmos de clasificación y reducción que siguen el mismo enfoque.

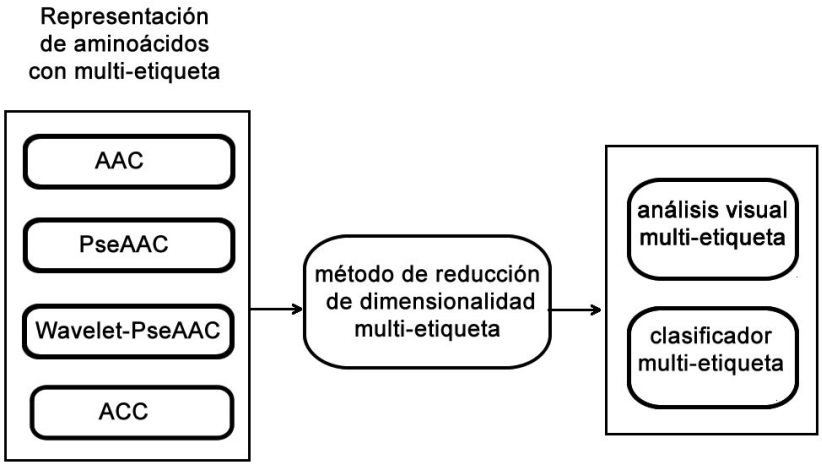


Figura 3.4: Diagrama de bloques para la reducción de secuencias de aminoácidos utilizando un modelado multi-etiqueta

3.3.1. k-NN multi-etiqueta

El aprendizaje basado en los k vecinos más cercanos de la versión multi-etiqueta es representado por la clase que se muestra en la figura 3.5. En el cuadro 3.1 se muestra una breve descripción de las variables y métodos asociados a dicha clase.

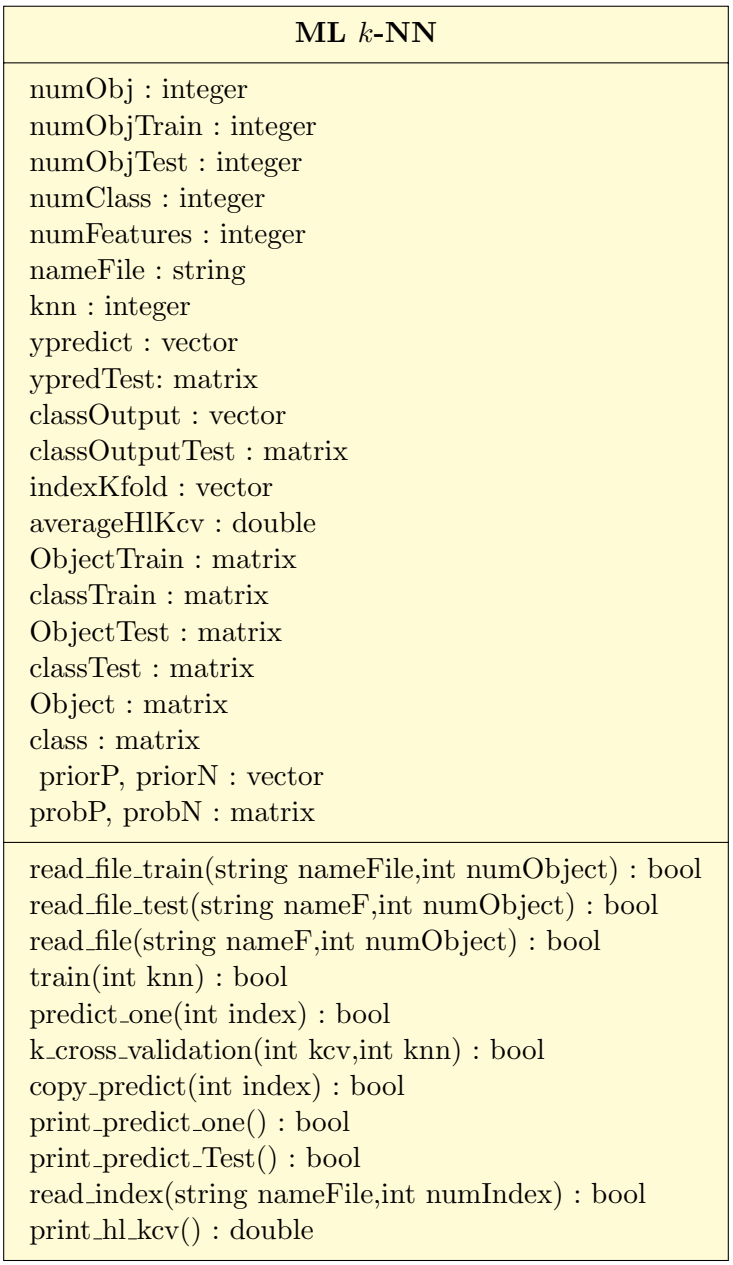


Figura 3.5: Diagrama de clases de k-NN multi-etiqueta

Clase MLkNN	
atributos	numObj: número de objetos del conjunto de datos numObjTrain: número de objetos en el conjunto de entrenamiento numObjTest : número de objetos en el conjunto de prueba numClass: número de clases numFeatures: número de características averageHlKcv: valor promedio de la métrica Hamming loss knn: número de vecinos a tomar en cuenta ypredict: vector de probabilidades de pertenecia ypredictTest: matriz de probabilidad de pertenencia classOutput: vector de etiquetas de salida de un ejemplo classOutputTest: matriz de etiquetas de salida del conjunto de entrenamiento indexKfold: vector de indices para la validación cruzada ObjectTrain: matriz de objetos del conjunto de entrenamiento classTrain: matriz de etiquetas del conjunto de entrenamiento ObjectTest: matriz de objetos del conjunto de prueba classTest: matriz de etiquetas del conjunto de prueba Object: matriz de objetos del conjunto de datos class: matriz de etiquetas del conjunto de datos priorP, priorN: vector de probabilidad a priori probP, probN: matriz de probabilidades condicionales
métodos	mlknn(): constructor read_file_train(...): leer subconjunto de entrenamiento read_file_test(...): leer conjunto de datos read_file(...): leer archivo train(...): entrenar el modelo predict_One(...): predecir un ejemplo del conjunto de prueba k_cross_validation(...): predecir mediante validación cruzada copy_predict(...): respaldar el vector de predicción print_predict_test(): imprimir las etiquetas asignadas al conjunto de prueba read_index(...): leer los índices para la validación cruzada print_hl_kcv(): imprimir el valor promedio de la k validación cruzada

Cuadro 3.1: Clase MLkNN con sus variables y métodos asociados

3.3.2. Análisis de correlación canónica

El algoritmo de análisis de correlación canónica utiliza la clase que se muestra en la figura 3.6. En el cuadro 3.2 se muestra una breve descripción de las variables y métodos asociados a dicha clase.

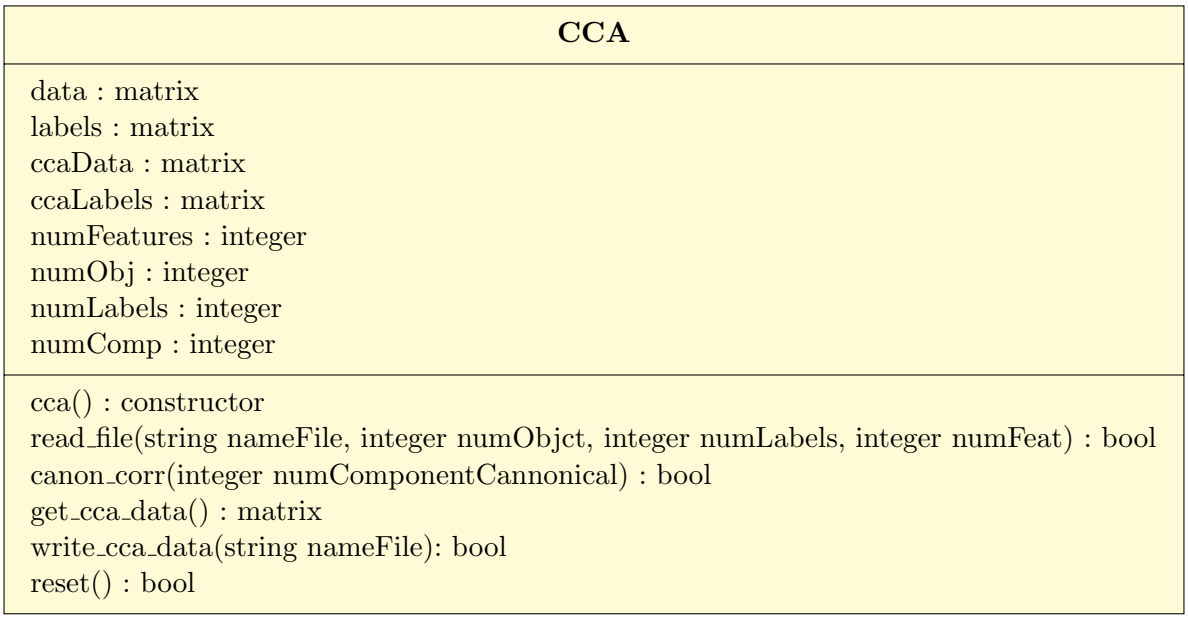


Figura 3.6: Diagrama de clases de análisis de correlación canónica.

Clase CCA	
atributos	data: matrix de características labels: matriz de etiquetas ccaData: matriz de proyección de las características ccaLabels : matriz de proyeccción de las etiquetas numFeatures: número de características numbj: número de objetos numLabels: número de etiquetas numComp: número de componentes canónicas
métodos	cca(): constructor read_file(...): leer matriz de datos canon_corr(...): ejecutar cca sobre los datos get_cca_data(): retorna la matriz de datos almacenada en l variable ccaData write_cca_data(string nameFile): escribe en un archivo el nuevo espacio reducido reset(): limpiar las variables

Cuadro 3.2: Clase CCA con sus variables y métodos asociados

3.3.3. Métricas de rendimiento

Para obtener el rendimiento de clasificación de ML k-NN se creó la clase que se muestra en la figura 3.7. En cuadro 3.3 se muestra una breve descripción de las variables y métodos asociados a dicha clase.

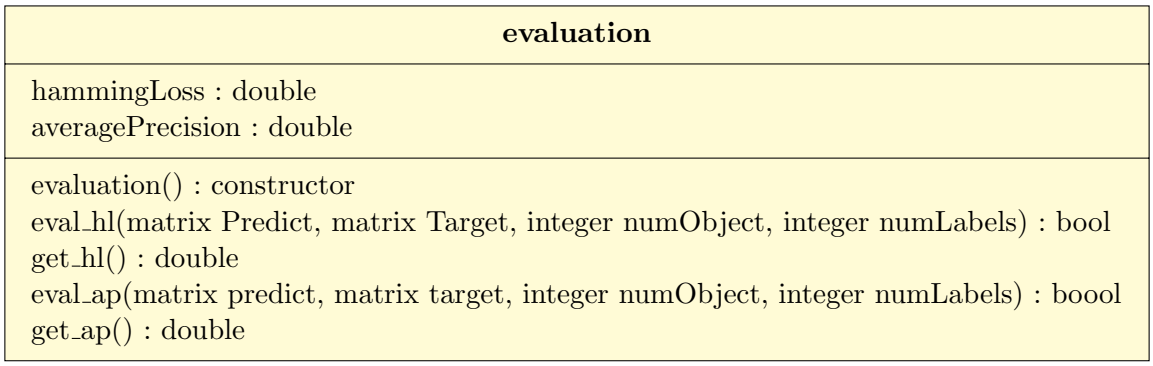


Figura 3.7: Diagrama de clases de las métricas de evaluación multi-etiqueta

Clase evaluation	
atributos	hammingLoss: almacena el valor de la métrica hamming loss averagePrecision: almacena el valor de la métrica average precision
métodos	evaluation(...): constructor eval_hl(...): obtiene el valor de la métrica hamming loss get_hl(): retorna el valor que almacena la variable hammingLoss eval_hp(...): obtiene el valor de la métricas avegare precision get_ap(): retorna el valor que almacena la variable averagePrecision

Cuadro 3.3: Clase evaluation con sus variables y métodos asociados

Capítulo 4

Resultados

Este capítulo presenta los resultados y discusión del análisis exploratorio, reducción de dimensionalidad y clasificación tradicional de las cuatro transformaciones aplicadas a los GPCRs de clase C. Posteriormente, se presenta el modelado multi-etiqueta de dichos conjuntos y los experimentos correspondientes a la reducción y clasificación multi-etiqueta de dicho modelado. Como punto final, se realiza una comparación de resultados del enfoque tradicional con el multi-etiqueta.

4.1. Análisis exploratorio

En esta sección se describe el conjunto de datos utilizado en los experimentos. Posteriormente, se realiza un análisis exploratorio a cuatro conjuntos de datos obtenidos al aplicar una serie de transformaciones a los datos de origen. El análisis se enfoca en visualizar la estructura de los datos mediante gráficas de dispersión en dos y tres dimensiones. También se exploran los datos para entender el tipo de distribución, los rangos de sus variables y la existencia de datos atípicos.

4.1.1. El conjunto de datos

Para las pruebas, se utiliza la base de datos pública GPCRDB [1] [Isberg et al., 2014]. La base de datos GPCRDB contiene las diferentes familias de GPCRs. Las secuencias utilizadas en estas pruebas corresponden a la familia C (versión 11.3.4 de marzo 2011), con un total de $1,392$ ejemplos de secuencias libres de alineamiento debidamente clasificadas en sus siete subfamilias (ver cuadro 4.1). A este conjunto de secuencias que están representadas en forma alfabética se le aplican cuatro transformaciones diferentes para obtener las representaciones numéricas correspondientes a dichas secuencias. La longitud de las secuencias alfabéticas es variable y va de 225 a $1,955$ aminoácidos por proteína.

Las transformaciones aplicadas a la base de datos de la familia C de GPCRs son: Composición de aminoácidos (AAC), PseAAC (Pseudo-composición de aminoácidos), Wavelet-PseAAC (basado en energía multiescala y PseAAC) y Auto-covarianza y covarianza cru-

[1] http://www.gpcr.org/7tm

Subfamilia	Descripción
Tipo 1	Receptores de calcio (Calcium sensing, CS)
Tipo 2	GABA-B (GB)
Tipo 3	Metabotrópicos de glutamato (Metabotropic glutamate, mG)
Tipo 4	Odorantes (Odorant, Od)
Tipo 5	Feromonas (Phermone, Ph)
Tipo 6	Receptores del gusto (Taste, Ta)
Tipo 7	Vomeronasal (Vomeronasal, VN)

Cuadro 4.1: Subfamilas de la clase C de los GPCRs.

zada (ACC). Para las trasformaciones PSeAAC y WaveletPseAAC se utilizan índices físico-químicos tomados de la base de datos AAIndex [2] [Kawashima and Kanehisa, 2000]. La transformación ACC utiliza los descriptores mencionados en [Lapinsh et al., 2002]. El cuadro 4.2 muestra el número de características de cada transformación. La figura **??** muestra el porcentaje que representa cada subfamilia dentro del conjunto de datos, siguiendo el orden y numeración específica en el cuadro 4.2.

Transformación	# de características
AAC	20
PseAAC	62
Wavelet-PseAAC	74
ACC	325

Cuadro 4.2: Transformaciones utilizadas y número de características.

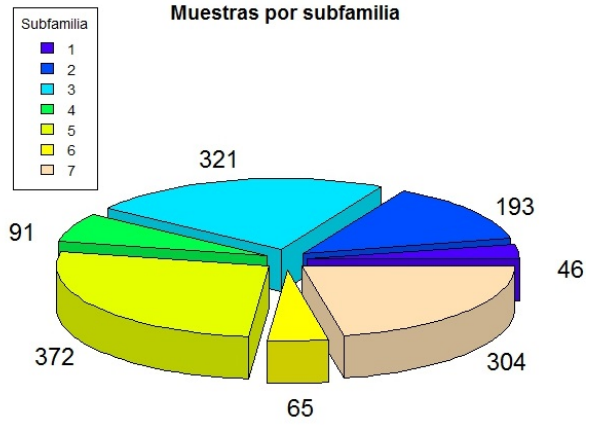


Figura 4.1: Número de muestras por subfamilia del conjunto de datos.

[2] http://www.genome.jp/aaindex/

4.1.2. Análisis previo de los datos

Antes de someter los conjuntos de datos a algún tipo de algoritmo, se lleva a cabo un análisis previo de los datos mediante una herramienta gráfica. Con el análisis exploratorio se busca entender la estructura y las propiedades que muestras los conjuntos de datos. El primer paso es examinar las variables, el tipo de distribución, observar si la distribución es simétrica y corroborar si la dispersión por pares de variables muestra algún tipo de relación.

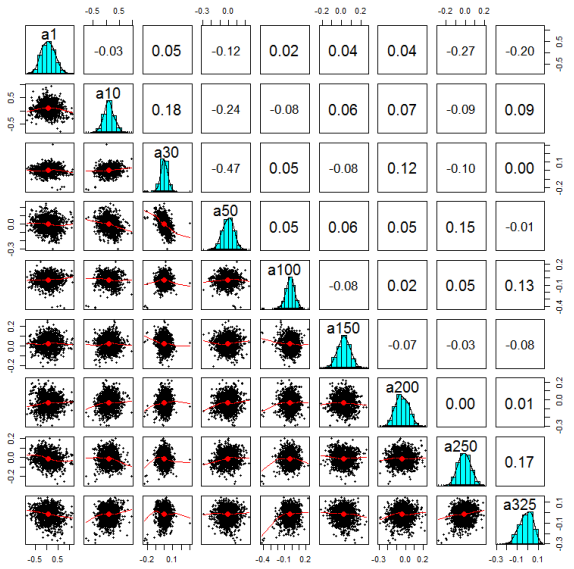
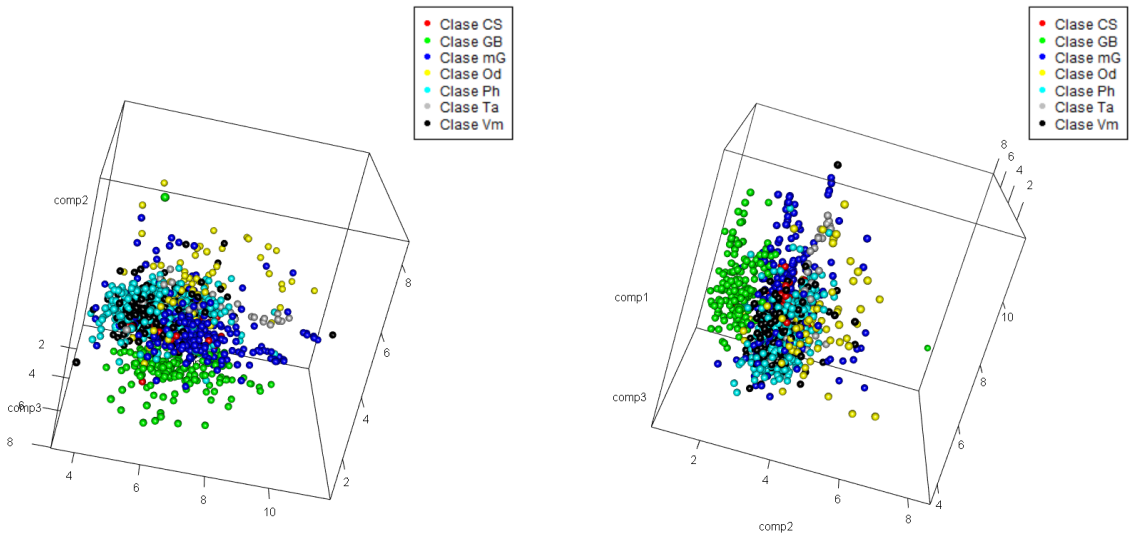


Figura 4.2: Gráficas de dispersión por pares para algunas variables de la transformación ACC.

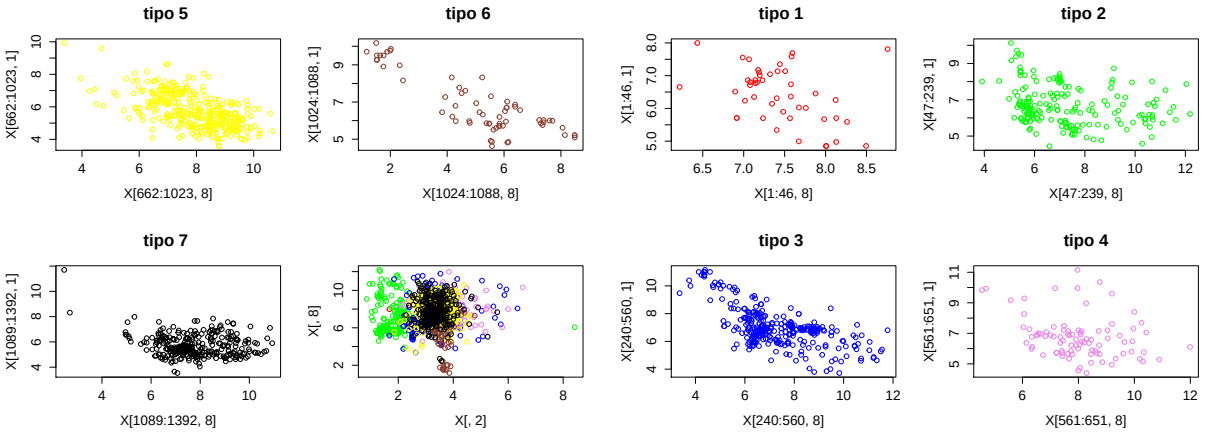
Después de la experimentación se observa que en las cuatro transformaciones las variables tienden a ser simétricas y con una distribución normal. La figura 4.2 muestra los diagramas de dispersión-correlación de algunas de las variables del conjunto de datos ACC. En general, los pares de variables no muestran una correlación lineal fuerte. Las gráficas de dispersión 3D muestran que en tres de los cuatro conjuntos (PseAAC, Wavelet-PseAAC y ACC) existe una separación lineal visible para al menos una de sus clases, y en el conjunto AAC se muestra una mayor mezcla de las muestras que las diferentes clases que el resto (ver figura 4.3) .

Las gráficas de dispersión vistas anteriormente representan a las variables sin distinción de acuerdo a su subfamilia discriminante. El saber la distribución de acuerdo a su etiqueta discriminante nos ayuda a comprender si la estructura de los datos al ser divididos de acuerdo a su clase es similar a la estructura general de la variable. Las figuras 4.3a y 4.3b contienen los diagramas de dispersión en 3D de acuerdo a la clase de pertenencia de cada muestra del conjunto AAC. En la figura 4.4a y 4.4b se muestra la nube de puntos de las variables 1 y 8 del conjunto ACC, la distribución de los datos en las subfamilias 3 (tipo 3) y 6 (tipo 6) indica un posible correlación lineal. En general bajo el análisis de dispersión por



(a) (b)

Figura 4.3: Grafica de dispersión en 3D de la transformación AAC.



(a) (b)

Figura 4.4: Gráficas de dispersión de acuerdo al tipo de subfamilia a la que pertenecen las muestras del conjunto AAC. Exceptuando a la transformación ACC, los demás conjuntos muestran que la clase 2 (color verde) se encuentra separada significativamente de las demás muestras.

pares y en el espacio $3D$ no muestra algún patrón, dependencia o distribución específica.

Por otro lado podrían existir observaciones sospechosas que se apartan de la media geométrica del grupo en todos los conjuntos. Para probar esto es necesario analizar con más detalle estos puntos, debido a que pueden representar datos atípicos, los cuales, como ya se mencionó afectan el análisis estadístico de la media, varianza, covarianza de los

datos y altera la correlación entre pares de variables.

Mediante gráficas de cajas es posible encontrar datos atípicos en las variables de los conjuntos de datos, sin embargo, para datos multivariantes es difícil comprobar que un dato es atípico de forma gráfica. Al ser un conjunto de datos multidimensional es difícil decidir con certeza que muestra es un dato atípico. Para todos los conjuntos, una o más variables contiene datos extremos en su dimensión correspondiente, mientras que en el resto de las dimensiones la misma muestra no se encuentra fuera del rango ordinario de los valores de la variable. Dicho de otra forma, esto quiere decir que al considerarse todas las dimensiones, los valores que toman las variables de una muestra sospechosa pueden caer fuera de la categoría de datos atípicos debido a la alta dimensionalidad. En estos casos se debe someter el conjunto a técnicas de análisis más robustas.

	Transformación			
Rango	AAC	PseAA	Wavelet-PseAAC	ACC
Mín	0.343643	-0.729	-0.729	-1.664
Máx	16	14.427	14.497	7

Cuadro 4.3: Rangos de las transformaciones

Los rangos de las variables varían para todos los conjuntos. Esta variación se observa en las gráficas de caja, lo que indica que se debe normalizar a los datos, ya que el no hacerlo puede afectar el resultado de las técnicas de análisis como por ejemplo PCA. En resumen, para los cuatro conjuntos de datos la dispersión tiende a ser normal. El cuadro 4.3 muestra los rangos mínimos y máximos de los cuatro conjuntos.

Las gráficas de caja también nos pueden decir si la media y la mediana se encuentran alejadas una de la otra, el cual ocurre en las cuatro transformaciones para la mayoría de las variables. De forma general, las cajas no son simétricas lo cual nos indica un leve sesgo en los datos, y que la distribución tiende a ser un poco asimétrica. En la figura 4.5 se aprecia como varían los rangos de las variables para el conjunto PseAAC.

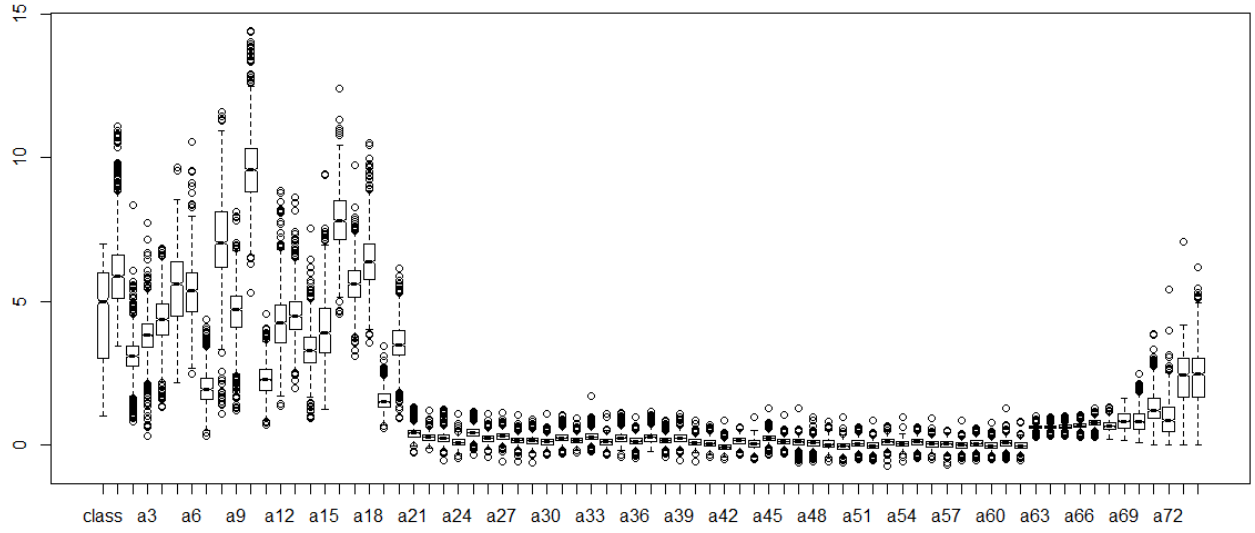


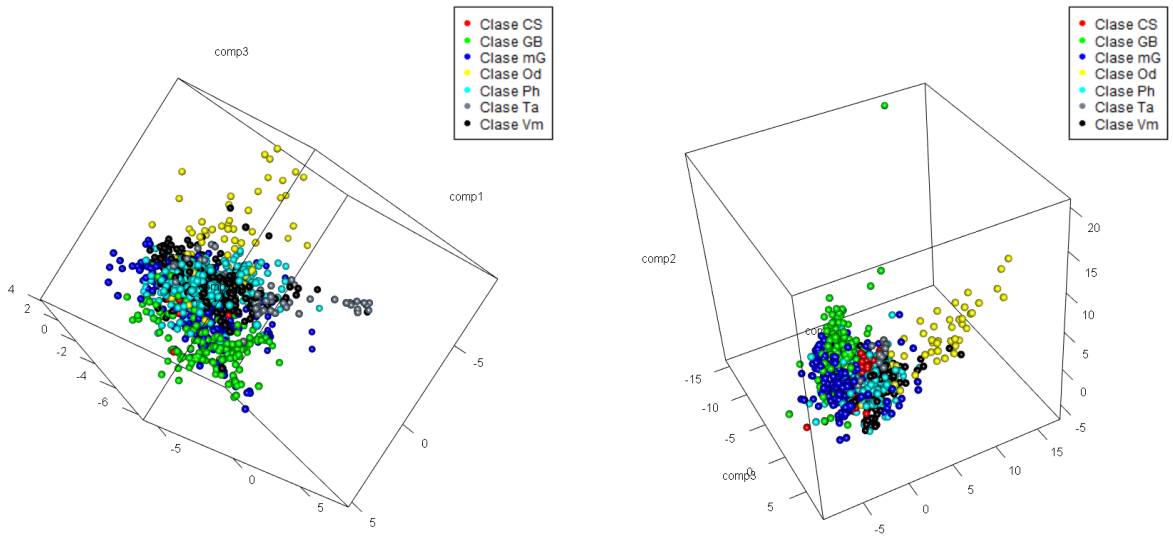
Figura 4.5: Gráfica de cajas del conjunto PseAAC.

4.2. Resultados de reducción de dimensionalidad

En esta sección se aplican métodos de reducción de dimensionalidad tradicionales a los conjuntos de datos en cuestión (ver cuadro 4.2). Las proyecciones, que son los nuevos conjunto de datos obtenidos al aplicar estos algoritmos de reducción, son analizadas de forma visual mediante gráficas de dispersión. Como parte final de estas pruebas se utiliza un algoritmo de cluster jerárquico y se generan dendrogramas con los grupos formados mediante el análisis jerárquico.

4.2.1. Pruebas de reducción

Después de identificar las principales características de los conjuntos de muestras mediante gráficos y el resumen de algunas cifras estadísticas de sus datos, se somete a estos conjuntos a algoritmos de reducción de dimensionalidad. El objetivo es obtener una mejor estructura de una baja dimensionalidad. Lo que se busca en esta nueva estructura es mejorar la relación y visualización existente entre las muestras de una misma subfamilia, y también que estas muestras (de una misma clase) se logren diferenciar del resto de las clases. El nuevo conjunto de datos reducido sera explorado de forma visual en dos y tres dimensiones. El resultado de esta exploración nos indicará si el algoritmo en cuestión logró encontrar una estructura en la cual los datos sean separables, o de forma contraria, si la reducción no logra la separación de las muestras de acuerdo a su categoría. Si esto último sucede, indica que el problema se debe atacar mediante otro enfoque.



(a) (b)

Figura 4.6: Gráficas de dispersión de las proyecciones obtenidas con PCA. (a) Conjunto AAC. (b) Conjunto PseAAC.

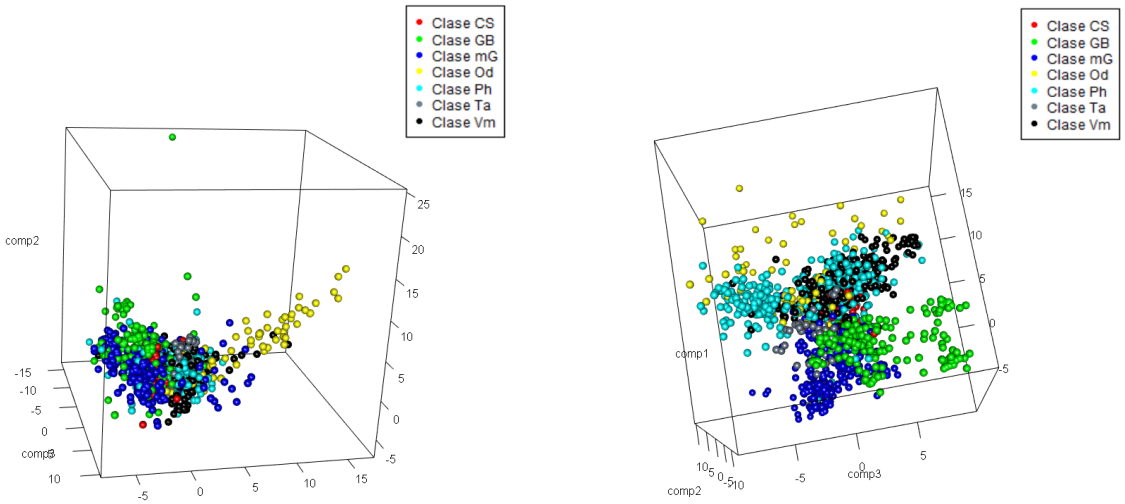
El primer algoritmo que se aplica es PCA, el cual supone que dentro del conjunto de variables existe un tipo de dependencia lineal. Si en una gráfica de dispersión por pares, se puede observar que la nube de datos se ajusta a una elipse, quiere decir que existe dependencia. PCA supone una distribución normal de los datos.

El resultado de aplicar PCA, sigue mostrando traslape al graficar sus primeras tres componentes como lo muestran las figuras 4.6 y 4.7. Sin embargo, son capaces de separar en forma mínima alguna de las clases. El conjunto ACC es el que mejor resultados de separación ofrece en las primeras tres componentes (ver figura 4.7b). Cabe mencionar que al aplicar PCA a todos los conjuntos se utilizó la matriz de correlación, ya que las variables de los conjuntos de datos tienen diferentes rangos.

Debido a que las variables originales no contenían una correlación fuerte, el número de componentes principales que logra explicar al menos el 80 % de la variabilidad de los datos son 10, 24, 28 y 97 para cada conjunto, respectivamente.

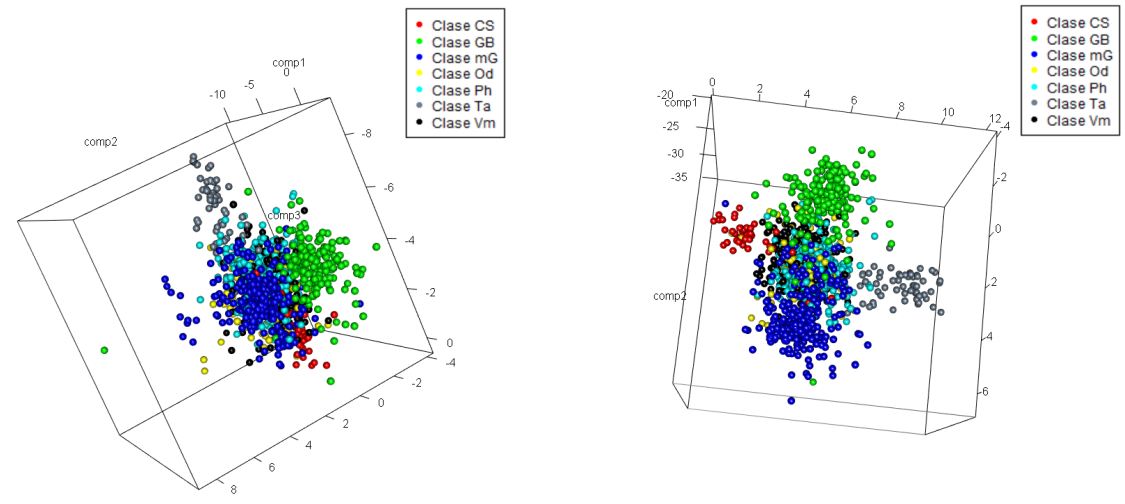
PCA se enfoca sólo en la reducción de dimensionalidad, al ser no supervisada, PCA no toma en cuenta la clase de pertenencia de cada muestra. Después de analizar el conjunto mediante PCA tratando de encontrar un correlación lineal entre variables que ayude a una mejor interpretación de su estructura, se analizan los datos con un algoritmo que se enfoca en la reducción de la dimensión pero buscando la mayor separabilidad entre los datos según la clase a la que pertenecen.

LDA basa su criterio en la búsqueda del mejor hiperplano que pueda categorizar mejor a las diferentes muestras según su clase. LDA es capaz de mapear los datos de la dimensión original, a una dimensión reducida igual a d-1, siendo d el número de clases en el conjunto. Esto sucede siempre y cuando el número de variables descriptivas sea superior al número



(a) (b)

Figura 4.7: Gráficas de dispersión de las proyecciones obtenidas con PCA. (a) Conjunto Wavelet-PseAAC. (b) Conjunto ACC.

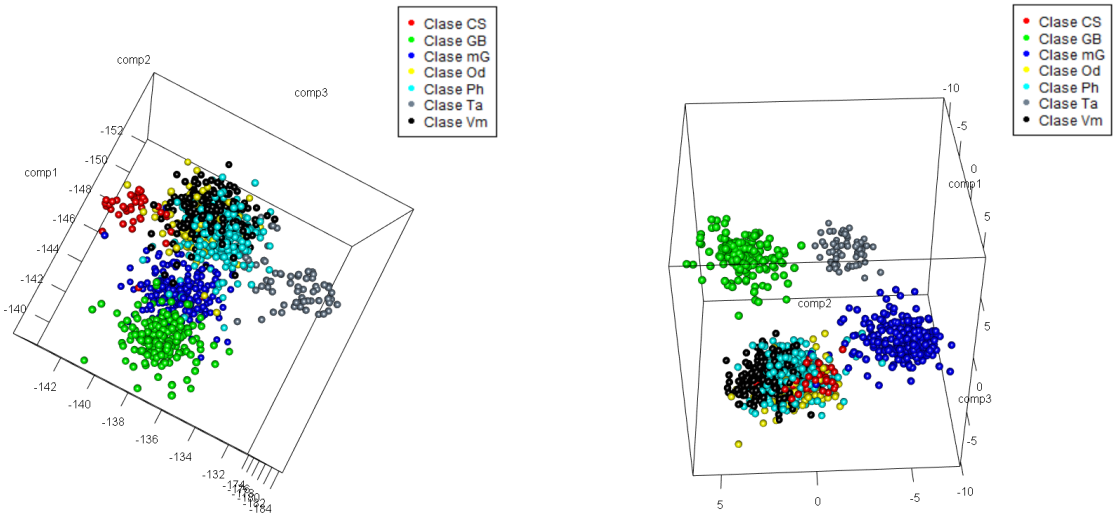


(a) (b)

Figura 4.8: Gráficas de dispersión de las proyecciones obtenidas con LDA. (a) Conjunto AAC. (b) Conjunto PseAAC.

de clases.

Los gráficos en tres dimensiones para las componentes de LDA muestran que la proyección logra separar la clase 2 (verde), 3 (azul) y 6 (gris) y parcialmente la clase 4 (amarillo)



(a) (b)

Figura 4.9: Gráficas de dispersión de las proyecciones obtenidas con LDA. (a) Conjunto Wavelet-PseAAC. (b) Conjunto ACC.

en el conjunto AAC (ver figura: 4.8a). En el conjunto PseAAC y Wavelet-PseAAC las clases 1 (rojo), 2 (verde), 3 (azul) y 6 (gris) se logran separar de los demás grupos del conjunto (ver figuras: 4.8b y 4.9a, respectivamente). Para el conjunto ACC sucede lo mismo que para PseAAC y Wavelet-PseAAC, sólo que para este último las clases 2, 3 y 6 logran una separabilidad mayor con respecto de los grupos restantes (ver figura: 4.9b). En resumen, el análisis gráfico de los conjuntos de datos en el nuevo espacio reducido muestra mayor separabilidad de acuerdo a su clase.

Para tener una perspectiva de cómo se agrupan los elementos de forma natural y verificar si las muestras de una misma clase son las más disimilares (cercanas) entre si, se utiliza el método de agrupamiento jerárquico. Este método agrupa las muestras utilizando un medida de disimilitud basada en distancias.

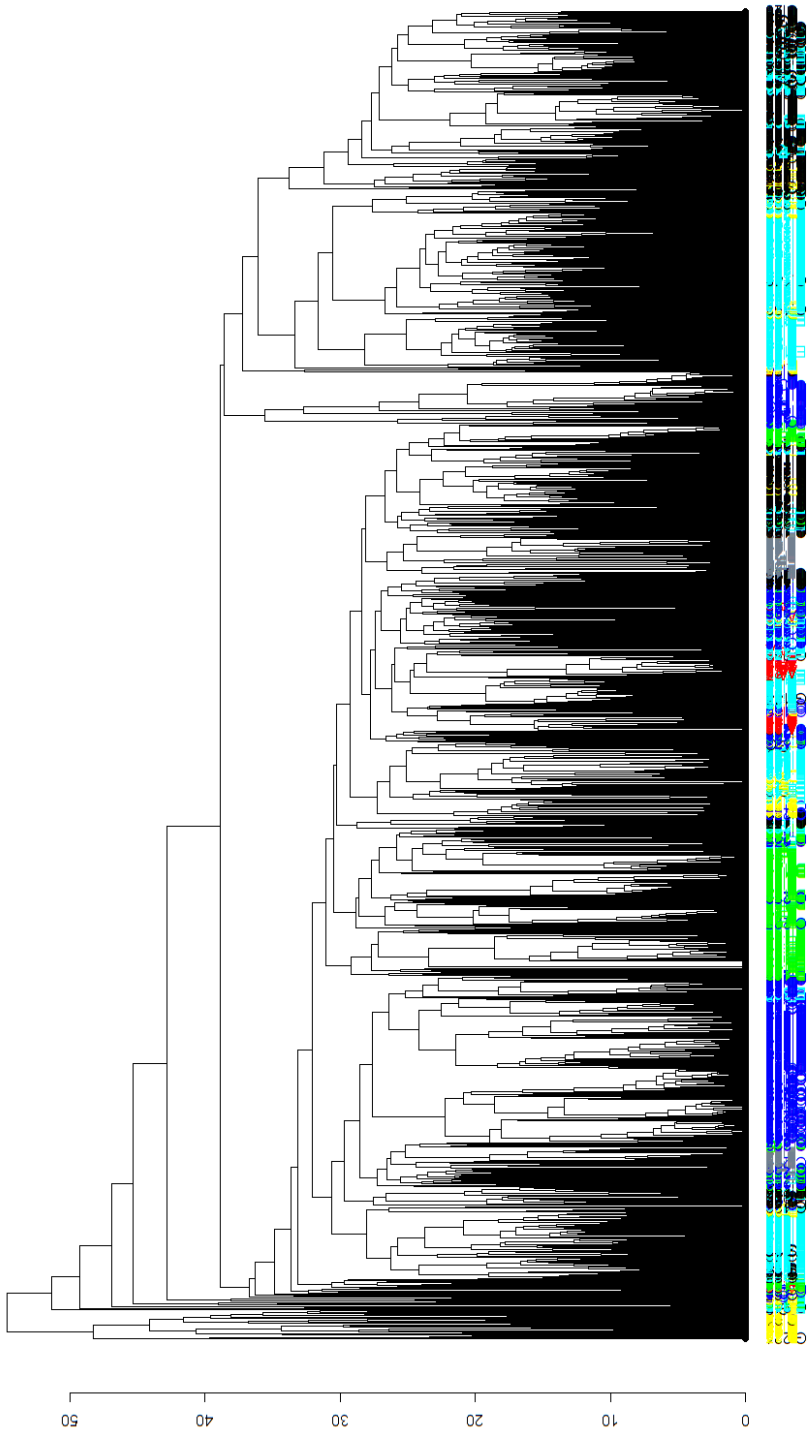


Figura 4.10: Dendrograma generado por el algoritmo de clúster jerárquico basado en la distancia media de los grupos del conjunto ACC. En las hojas, se muestra el número de instancia asociado con una letra como prefijo que es asignada de acuerdo a su clase, las marcas de color corresponde a la clase a la que pertenecen las muestras.

Al aplicar a los conjuntos de datos un análisis de conglomerados jerárquico se obtiene un árbol llamado dendrograma en el cual se puede apreciar claramente las relaciones de agrupación natural entre los datos. Este tipo de gráficos muestra la forma de agrupar utilizando la distancia media entre los grupos. Para los cuatro conjuntos se utilizó la matriz de distancias estandarizada, debido a los motivos antes expuestos.

El dendrograma de la figura 4.10 muestra la jerarquía de los elementos y los grupos que se forman al utilizar los datos del conjunto inicial (sin reducción de dimensionalidad) ACC. En dicha figura es notorio que para el conjunto no reducido el traslape de clases es significativo. Puesto que en las hojas de los árboles existen una mezcla de colores que indica que muchas ramas contiene elementos de distintas clases. Para este mismo conjunto de datos, pero utilizando los datos de la proyección de LDA se puede observar que los grupos que se forman en el dendrograma son más homogéneos (ver figura 4.11). Esto es, las hojas de una misma rama contienen en su mayoría objetos pertenecientes a la misma clase . Esta situación se repite para los tres conjuntos de datos restantes (AAC, PseAAC, Wavelet-PseAAC). Además estos resultados coinciden con las vistas en $3D$ de las proyecciones obtenidas con LDA en sus tres primeros componentes.

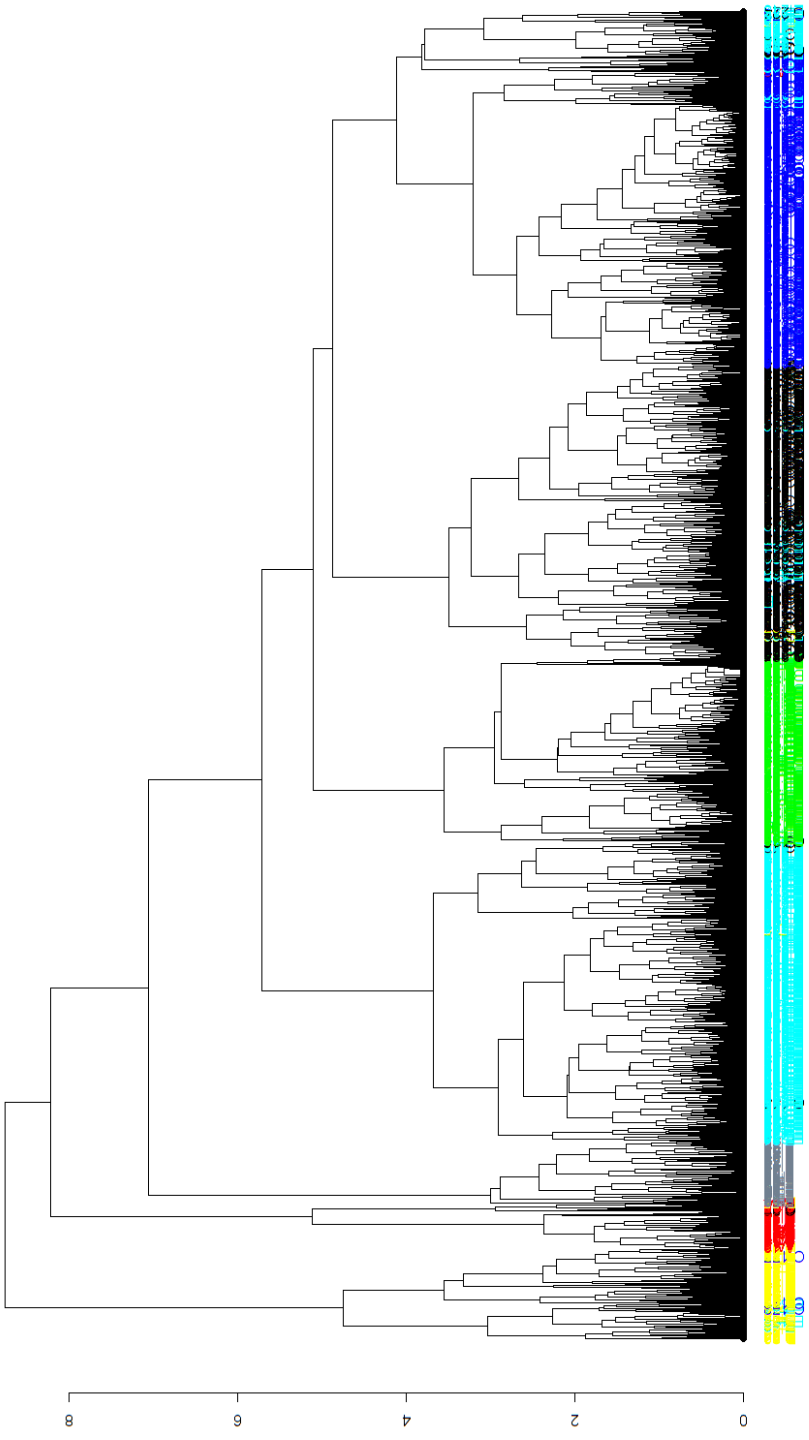


Figura 4.11: Dendrograma generado por el algoritmo de cluster jerárquico utilizando la distancia media de los grupos como medida de similaridad para la proyección creada por LDA para el conjunto de datos ACC.

4.3. Resultados de rendimiento

En esta sección se aplican algoritmos de clasificación a las transformaciones originales, posteriormente estos mismos métodos serán aplicados a los conjuntos reducidos. El objetivo es obtener el rendimiento de clasificación sobre cada conjunto. El rendimiento que muestran los cuadros a partir de esta sección es el valor promedio resultado de ejecutar el algoritmo correspondiente diez veces, utilizando validación cruzada estratificada con diez particiones.

4.3.1. Pruebas de exactitud utilizando clasificadores tradicionales

Los algoritmos de reducción de dimensionalidad como PCA, generalmente van separados de los algoritmos como LDA y los métodos de análisis de cluster. Estos dos últimos métodos regularmente se utilizan para clasificación y agrupamiento respectivamente. Como se mencionó en la sección anterior, el objetivo de la reducción de características en los conjuntos es encontrar una estructura más compacta de los datos que logre discriminar mejor nuevas observaciones. Por lo anterior, es necesario estimar el error de clasificación que sirva como referente para medir la efectividad de los métodos aplicados.

Transformación	% exactitud
AAC	81.25 (k=1)
PseAAC	87.42 (k=3)
Wavelet-PseAAC	87.57 (k=1)
ACC	87.5 (k=3)

Cuadro 4.4: Medida de exactitud promedio utilizando k-NN en las transformaciones originales.

En esta etapa se utilizan clasificadores sencillos para medir la eficiencia de los modelos obtenidos. El primer clasificador con el que se analizan los datos es k-NN (k nearest neighbors). Las variaciones de k en el rango $[1, 19]$, utilizadas en este algoritmo son las ayudan a obtener la configuración con mayor exactitud de clasificación, o dicho de otra forma, la configuración que mejor se ajuste al modelo cuando sea necesario predecir a que subfamila pertenece un nuevo dato. k-NN se aplica primero, al conjunto no reducido y, posteriormente al conjunto que representa la proyección obtenida al aplicar PCA, para ambos casos se utiliza validación cruzada estratificada con 10 particiones.

Los resultados de clasificación utilizando k-NN (con el mejor valor de k)y los conjuntos de datos originales se muestran en el cuadro 4.4. Para comparar, el cuadro 4.5 muestra los resultados de k-NN utilizando las proyecciones de los datos generadas por PCA. En estos cuadros (4.4 y 4.5) se puede observar que con excepción del conjunto AAC, la mayor exactitud de clasificación se obtiene con el conjunto original. Esto se puede justificar debido al hecho de que la distribución de los datos no se ajustan a una forma elíptica,

la cual indicaría dependencia lineal. PCA, que se enfoca en conservar la varianza de los datos no es lo suficientemente potente trabajando con datos que representan estructuras complejas.

Transformación	% exactitud
AAC	82.68 (k=1, 89)
PseAAC	82.32 (k=1, 100)
Wavelet-PseAAC	82.54 (k=1, 89)
ACC	85.41 (k=1, 80)

Cuadro 4.5: Medida de exactitud promedio utilizando k-NN y los conjuntos obtenidos con PCA. Entre paréntesis de izquierda a derecha se muestra el número de vecinos y la varianza acumulada en la reducción, respectivamente.

También se clasificaron los datos mediante el algoritmo LDA que basa su clasificación en los hiperplanos de separación que genera. Se utilizó validación cruzada estratificada con 10 particiones, el porcentaje de clasificación correcta se muestra en el cuadro 4.6.

Conjunto	Exactitud
AAC	77.29
PseAAC	86.78
Wavelet-PseAAC	86.56
ACC	96.69

Cuadro 4.6: Medida de exactitud promedio para las proyecciones obtenidas al aplicar LDA a las transformaciones originales y clasificando con los planos discriminantes que genera el algoritmo.

Otra alternativa es reducir la dimensionalidad de los conjuntos originales mediante LDA y clasificar los datos reducidos mediante k-NN. El cuadro 4.7 muestra los resultados de dicha alternativa. La exactitud de los conjuntos reducidos al utilizar los hiperplanos que genera LDA durante la reducción es muy similar a la obtenida mediante k-NN con los mismo conjuntos.

Transformación	Exactitud
AAC	78.02 (k=9)
PseAAC	84.84 (k=5)
Wavelet-PseAAC	86.70 (k=7)
ACC	96.26 (k=3)

Cuadro 4.7: Medida de exactitud promedio para las proyecciones obtenidas al aplicar LDA a las transformaciones originales y clasificando con k-NN.

CS	GB	mG	Od	Ph	Ta	VN	
41	1	2	0	1	0	1	CS
0	172	11	0	6	0	4	GB
2	3	286	6	21	1	2	mG
1	2	6	52	17	0	13	Od
2	2	13	6	321	1	27	Ph
0	1	0	2	2	58	2	Ta
4	0	6	4	35	2	253	VN

Cuadro 4.8: Matriz de confusión de la transformación AAC utilizando 1-NN para clasificar.

La matriz de confusión para el porcentaje de clasificación más bajo utilizando k-NN el cual corresponde al conjunto AAC se muestra en el cuadro 4.8. Por otra parte, el cuadro 4.9 muestra la matriz de confusión del conjunto con la mayor exactitud, la cual se obtiene al clasificar con LDA el conjunto ACC. Para ambos casos, la matriz de confusión muestra que el número de elementos mal clasificados es mayor para las clases 4, 5 y 7, lo cual sucede para todos los conjuntos. Además, estas tres clases son las que muestran estar más mezcladas en los gráficos de dispersión en dos y tres dimensiones (ver sección 4.2.1).

CS	GB	mG	Od	Ph	Ta	VN	
45	0	1	0	0	0	0	CS
0	193	0	0	0	0	0	GB
0	0	316	2	3	0	0	mG
1	0	0	81	4	0	5	Od
0	0	4	3	350	0	15	Ph
0	0	0	0	0	65	0	Ta
1	0	0	0	7	0	296	VN

Cuadro 4.9: Matriz de confusión de transformación ACC utilizando LDA para clasificar.

Lo anterior se confirma en el cuadro 4.10 que contiene las distancias entre los centroides obtenidos al aplicar LDA al conjunto AAC. Aquí las celdas resaltadas con negritas son los grupos más cercanos entre si. Este cuadro muestra que el centroide de la clase 4 es el más cercano a la clase 5 y 7 cuando se realizan los agrupamientos. En consecuencia, estas tres clases son las que contienen un mayor número de elementos mal clasificados, como muestran los cuadros 4.8 y 4.9.

0	1	2	3	4	5	6
2	4.80					
3	3.87	3.33				
4	3.98	5.58	4.08			
5	3.51	4.52	3.05	**2.67**		
6	5.67	5.81	4.65	4.37	3.31	
7	3.37	4.83	3.65	**2.52**	**1.26**	3.76

Cuadro 4.10: Matriz de distancias para los centroides obtenidos al aplicar LDA al conjunto AAC.

Debido a que las clases 4, 5 y 7 son las que afectan el porcentaje de clasificación correcta en mayor medida según los cuadros 4.6 y 4.7, en el siguiente experimento se fusionan estas tres clases, y, posteriormente se aplica k-NN para comprobar si se incrementa la exactitud de clasificación, con el objetivo de objetivo de evaluar si las clases son semejantes. De forma general, el rendimiento se incrementa para todos los conjuntos (AAC, PseAAC, Wavelet-PseAAC y ACC), logrando los siguientes resultados: 93.03 %, 93.68 %, 94.40 % y 93.39 %. Obteniendo mejor porcentaje de clasificación correcta el conjunto Wavelet-PseAAC.

Aunque el rendimiento aumenta con la fusión de las clases 4, 5 y 7 en los conjuntos originales, el número de ejemplos mal clasificados aún es significativo, sobre todo en la columna que representa a la fusión de las clases 4, 5 y 7. Además, la fusión de dichas clases asume que estas sólo realizan una función, lo cual contradice el conocimiento actual de que cada clase tiene una función distinta. Lo anterior permite inferir que estas subfamilias deben ser modeladas con un enfoque distinto.

Ahora, si se consideran sólo las muestras pertenecientes a las clases 5 y 7 y se aplica k-NN obtenemos una exactitud de clasificación de 88.31, 91.56, 91.42 y 93.63 por ciento, para cada conjunto, respectivamente. En este experimento se toman sólo las instancias pertenecientes a estas dos clases debido al hecho de que el mayor número de elementos mal clasificados que corresponden a la clase 5 son asignados a la clase 7 por LDA y k-NN. De forma análoga, la mayoría de muestras mal clasificadas pertenecientes a la clase 7 son asignadas a la clase 5 por los clasificadores utilizados, lo cual no sucede de forma recíproca si se toma en cuenta las instancias de la clase 4. El cuadro 4.11 muestra la matriz de confusión resultante para las muestras pertenecientes a las clases 5 y 7 del conjunto AAC al ser clasificadas con k-NN. Esta matriz contiene una cantidad considerable de ejemplos mal clasificados, lo cual indica que las dos clases contienen una estructura que es difícil de discriminar por los clasificadores utilizados. De igual forma que en el experimento anterior, dado que cada subfamilia tiene una función distinta, estas deben ser modeladas con un enfoque distinto.

a	b	
336	36	a = 5
43	261	b = 7

Cuadro 4.11: Matriz de confusión obtenida al clasificar mediante k-NN (k = 1) las muestras de las subfamilias 5 y 7 para el conjunto de datos AAC.

Por otro lado, la regresión logística (RL) es otro método tradicional se utiliza para clasificar objetos y selección de variables. Ante de atacar el problema de discriminación en cuestión con modelos más complejos, aplicaremos RL a las clases 5 y 7 asumiendo que un modelo lineal es capas de discriminarlas eficazmente. En el cuadro 4.12 los p-valores indican que en el subconjunto de datos de las familias 5 y 7 las variables no son significativamente importantes para el modelo, ya que el p-valor ($\Pr(> \|z\|)$) es muy elevado (> 0.05) para todas ellas. Con esta información no es posible realizar una interpretación objetiva del porque de este fenómeno.

	Estimate	Std. Error	z value	$\Pr(> \|z\|)$
(Intercept)	124908.2244	314917.4708	0.40	0.6916
a1	-1248.9351	3149.1787	-0.40	0.6917
a2	-1250.4035	3149.1678	-0.40	0.6913
a3	-1248.4876	3149.1766	-0.40	0.6918
a4	-1248.6197	3149.1710	-0.40	0.6917
a5	-1247.7501	3149.1763	-0.40	0.6919
a6	-1249.2277	3149.1696	-0.40	0.6916
a7	-1249.1490	3149.1891	-0.40	0.6916
a8	-1248.9927	3149.1751	-0.40	0.6917
a9	-1249.0761	3149.1714	-0.40	0.6916
a10	-1248.8284	3149.1718	-0.40	0.6917
a11	-1249.0197	3149.1729	-0.40	0.6916
a12	-1249.8282	3149.1717	-0.40	0.6915
a13	-1249.1792	3149.1750	-0.40	0.6916
a14	-1249.6769	3149.1699	-0.40	0.6915
a15	-1249.4926	3149.1832	-0.40	0.6915
a16	-1249.1391	3149.1751	-0.40	0.6916
a17	-1249.1615	3149.1745	-0.40	0.6916
a18	-1248.9310	3149.1730	-0.40	0.6917
a19	-1251.1359	3149.1946	-0.40	0.6912
a20	-1249.4858	3149.1782	-0.40	0.6915

Cuadro 4.12: Resumen de el conjunto de datos AAC para las clase 5 y 7 aplicando regresión logística.

A simple vista pareciera que ninguna variable es relevante. Supongamos que existen dos variables en un modelo y una de estas es una copia de la otra, los p-valores indican que ambas variables no son relevantes, si eliminamos una de ellas, entonces la variable que queda se vuelve significativamente relevante ya que sin esta última no se podría hablar de predicción. El modelo logístico no es eficiente cuando los datos mantienen una distribución normal, dicho modelo puede ser más eficaz cuando el conjunto de datos no tienen la misma matriz de covarianzas, lo cual contrasta con los parámetros que exige LDA.

	Estimate	Std. Error	z value	Pr($>\|z\|$)
(Intercept)	133.3053	14.7831	9.02	0.0000
a1	-1.1843	0.2089	-5.67	0.0000
a2	-2.6574	0.3209	-8.28	0.0000
a3	-0.7375	0.2317	-3.18	0.0015
a4	-0.8719	0.1961	-4.45	0.0000
a6	-1.4806	0.2220	-6.67	0.0000
a7	-1.3941	0.2496	-5.59	0.0000
a8	-1.2434	0.1744	-7.13	0.0000
a9	-1.3282	0.2290	-5.80	0.0000
a10	-1.0803	0.1807	-5.98	0.0000
a11	-1.2712	0.2864	-4.44	0.0000
a12	-2.0804	0.2520	-8.25	0.0000
a13	-1.4298	0.2324	-6.15	0.0000
a14	-1.9296	0.2486	-7.76	0.0000
a15	-1.7402	0.2328	-7.47	0.0000
a16	-1.3898	0.1977	-7.03	0.0000
a17	-1.4124	0.2140	-6.60	0.0000
a18	-1.1825	0.1892	-6.25	0.0000
a19	-3.3795	0.3991	-8.47	0.0000
a20	-1.7354	0.2343	-7.41	0.0000

Cuadro 4.13: Resumen de los conjunto de datos AAC para las clase 5 y 7 aplicando regresión logística eliminando la variable $a5$.

Mediante prueba y error se descartan variables para observar como se comporta el modelo RL al momento de predecir. En estos experimentos se descubrió que al eliminar una de las variables los p-valores cambian significativamente. El cuadro 4.13 contiene los nuevos índices de los p-valores al realizar regresión logística eliminado la variable $a5$ del proceso. Para todas las variable los p-valores cumple con la restricción antes mencionada, lo que indica que todas las componentes son esenciales para el modelo. Al utilizar este método como un clasificador utilizando validación cruzada el porcentaje de clasificación correcta es de 76.03 %, el cual resulta inferior al obtenido por $k-$NN tomando en consideración todas las variables. Sucede los mismo al analizar las subfamilias 5 y 7 de los conjuntos restantes con RL. RL no ofrece resultados superiores por tal, se descartan más experimentos.

4.3.2. Pruebas de exactitud utilizando clasificadores no lineales

En esta sección se aplica una máquina de soporte vectorial (SVM) con el kernel Gaussiano a las cuatro transformaciones sin reducción, así como también a las proyecciones obtenidas al someter dichos conjuntos a PCA y LDA. Para obtener un rendimiento de clasificación más preciso se emplea la técnica de doble validación cruzada.

Para que SVM ofrezca resultados óptimos, los parámetros del kernel a utilizar deben ser los que mejor generalicen al conjunto de datos. Para esto, se utiliza la estrategia de experimentación de diseño factorial [Alpaydin, 2010] (comúnmente denominada búsqueda de

malla) para ajustar los parámetros C y *gamma* en los rangos $[2^1, ..., 2^{10}]$ y $[2^{-1}, ..., 2^{-10}]$, respectivamente, del kernel Gaussiano. La figura 4.12 muestra el proceso general del esquema de doble validación cruzada.

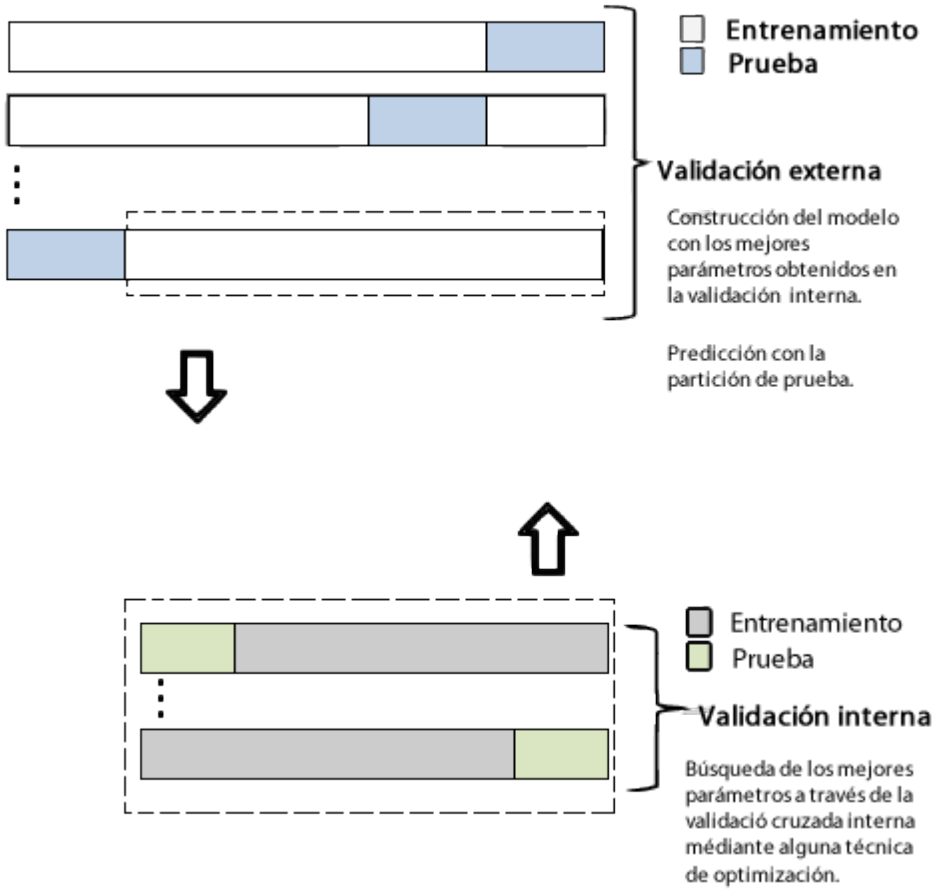


Figura 4.12: Esquema general de una doble validación cruzada

El procedimiento de doble validación cruzada consiste en los siguientes pasos [König et al., 2013, Ramos Pérez, 2016]:

1. Normalizar los datos.
2. Crear cinco particiones estratificadas para la validación cruzada externa.
   a) Utilizar la partición de entrenamiento para obtener los mejores parámetros del kernel Gaussiano.
      1) Para cada combinación de los parámetros C y γ crear un modelo y obtener su rendimiento utilizando validación cruzada interna con cinco particiones. Almacenar los parámetros que arrojen el mejor resultado.
      2) Crear un modelo utilizando los mejores parámetros obtenidos y entrenarlo con los datos de la partición de entrenamiento actual de la validación cruzada externa.

b) Clasificar el conjunto de prueba con el modelo entrenado en el paso $a2$) utilizando una medida de rendimiento.

3. Calcular el valor promedio del rendimiento de las cinco iteraciones de la validación cruzada externa.

El cuadro 4.14 muestra el rendimiento de las SVMs para las cuatro transformaciones sin utilizar RD y también de las proyecciones obtenidos al aplicar LDA y PCA.Aquí se observa que, SVM logra incrementar el rendimiento de clasificación de todos los conjuntos en forma general. Las transformaciones sin reducción AAC, ACC y las proyecciones obtenidas con PCA presentan un incremento significativo si las comparamos con los resultados de k-NN (ver sección 4.3.1). Para las proyecciones generadas por LDA sólo se observa un incremento considerable para la transformación PseAAC. Los mejores resultados para todos los conjuntos son obtenidas con la transformación ACC. A continuación comparamos los mejores resultados obtenidos con clasificadores convencionales en este trabajo con los reportados en la literatura.

	Resultados de la clasificación		
		Método de reducción	
Transformación	sin reducción	PCA	LDA
AAC	88.14	85.49 (100,20)	79.31
PseAAC	89.66	88.79 (81,24)	86.78
Wavelet-PseAAC	90.89	88.43 (89,37)	86.71
ACC	**92.53**	**92.10** (80,97)	**96.70**

Cuadro 4.14: Medida de exactitud promedio utilizando una SVM

Los resultados en [König et al., 2014b] reportan la mayor exactitud de clasificación para la transformación ACC utilizando la transformación de digramas (digram-frequency composition) con un 93 % de clasificación correcta utilizando una SVM. También es utilizado una SVM en [König et al., 2014a], en donde se obtiene una exactitud de 94.3 % con 585 características utilizando la transformación de secuencias de n-gramas sin selección de características, mientras que con una selección de 49 atributos logra una exactitud de 93.9 %. En [Ramos Pérez, 2016] se logra una exactitud de clasificación de 95.13 % utilizando aprendizaje profundo a través de una máquina de Boltzmann restringida. En este trabajo reduciendo la transformación ACC con LDA a seis características se logra una exactitud de 96.26 %, 96.69 % y 96.70 % con 3-NN, planos discriminantes de LDA y SVM, respectivamente.

Alternativamente a PCA se aplicó Kernel PCA (KPCA) y Spherical Principal Components (SPC) [Locantore et al., 1999], los resultados no se muestran en este documento ya que no mejoran el rendimiento mostrado por PCA. Aunque ACC-PCA con 80 componentes y un 90 % de retención de varianza iguala el rendimiento obtenido de ACC-SVM, el número de componentes necesarias es muy alto.

4.4. Modelado multi-etiqueta

De los experimentos que se realizaron en las secciones anteriores, los resultados de los puntos que se enlistan a continuación son la clave para el nuevo enfoque que se le dará al problema de inseparabilidad de algunas subfamilias pertenecientes a los GPCRs de la clase C:

- La exploración en $3D$ de las proyecciones generadas por LDA en sus 3 primeras componentes.
- El análisis de las matrices de confusión (ver cuadro 4.15).
- Los resultados de rendimiento al fusionar las clases 4, 5 y 7 (ver cuadro 4.16).

1	2	3	4	5	6	7	
45	0	1	0	0	0	0	a = 1
0	193	0	0	0	0	0	b = 2
0	0	316	2	3	0	0	c = 3
1	0	0	81	4	0	5	d = 4
0	0	4	3	350	0	**15**	e = 5
0	0	0	0	0	65	0	f = 6
1	0	0	0	7	0	296	g = 7

Cuadro 4.15: Matriz de confusión de la transformación ACC utilizando LDA para clasificar.

Transformación	Clasificador	
	1-NN	SVM
AAC	93.03	95.11
PseAAC	93.68	96.12
Wavelet-PseAAC	94.40	96.05
ACC	93.39	96.77

Cuadro 4.16: Cuadro de rendimiento promedio al fusionar las clase Od, Ph y Vn clasificando las transformaciones con k-NN y SVM de los conjuntos originales.

Las clases 4, 5 y 7 (Od, Ph y VN, respectivamente) están asociadas a funciones de receptores olfativos, y de acuerdo a la naturaleza multi-funcional de las proteínas (ver sección 2.5.3) podemos proponer que estas tres subfamilias comparten funciones. Si un conjunto de proteínas multifucionales pertenecientes a diferentes familias o subfamilias comparten al menos una función, sus secuencias nativas pueden ser similares. Las regiones biológicamente significativas, en las que se encuentran asociadas las funciones que realizan, son regiones flexibles, las cuales sólo toman una estructura definida en presencia

de cierto ligando. Cuando la estructura química de los ligandos presentes no permite una interacción, obtener un patrón preciso es improbable, lo que hace que estas regiones no aporten información relevante a los algoritmos de reconocimiento de patrones. Dicho de otra forma, la divergencia de las secuencias para este tipo de proteínas no es significativamente marcada, lo cual provoca que los algoritmos de aprendizaje automático no logren encontrar un patrón discriminativo y se confundan al momento de realizar la clasificación.

Si suponemos que existen proteínas que pueden ser activadas por más de un ligando, podemos atacar el problema de separabilidad, clasificación y reducción de dimensionalidad desde un enfoque multi-etiqueta. Como se mencionó en la sección 2.4, en el aprendizaje multi-etiqueta el espacio de hipótesis puede ser compartido por las instancias asociadas a múltiples clases. Para poder tomar un enfoque multi-funcional para los GPCRs de clase C, el primer paso es modelar las transformaciones con las que se han realizado los experimentos a su forma multi-etiqueta.

El modelado multi-etiqueta de las cuatro transformaciones se basa en las matrices de confusión que se obtienen al aplicar SVM a dichos conjuntos. De forma similar a como se realizaron las pruebas de rendimiento para las cuatro transformaciones con SVM utilizando doble validación cruzada y búsqueda de malla, se realiza nuevamente este proceso para separar las instancias asociadas a múltiples funciones del resto de las muestras. A cada transformación sin reducción se le aplicará dicho proceso, para obtener su modelo multi-etiqueta correspondiente. Para que el muestreo sea válido y se pueda omitir a las instancias que caen dentro del margen de error del modelo creado por SVM se repiten las pruebas 100 veces. La matriz de confusión es modificada para que en cada casilla almacene una lista que contenga a las instancias de acuerdo a la clase a la que las asignó el clasificador, y un contador que acumula el número de veces que cada instancia fue asignada a la misma clase durante las 100 iteraciones.

Por ejemplo: la casilla $(5, 7)$ de la matriz de confusión del cuadro 4.15 indica que 15 instancias que pertenecen a la clase 5 fueron asignadas a la clase 7. La matriz de confusión que se crea para el proceso iterativo ya mencionado, en lugar de almacenar el número de instancias mal clasificadas almacena una lista como la que se muestra en el cuadro 4.17. La primera posición en la lista almacena la dupla $(100, 898)$, lo cual se interpreta de la siguiente manera: la instancia número 898 fue asignada 100 veces a la clase 7, siendo la clase 5 su clase real.

Índice	(contador, núm. de instancia)	Índice	(contador, núm. de instancia)
1	100,898	21	46,662
2	100,969	22	25,929
3	100,970	23	23,715
4	100,971	24	19,991
5	100,985	25	17,927
6	100,986	26	17,990
7	100,987	27	15,655
8	100,993	28	15,800
9	100,995	29	15,980
10	100,1018	30	13,787
11	100,1020	31	13,989
12	98,967	32	12,923
13	98,972	33	12,1022
14	97,901	34	11,652
15	95,968	35	10,915
16	91,828	36	9,651
17	81,892	37	8,754
18	74,897	38	7,819
19	63,654	39	6,918
20	50,900	40	6,1021

Cuadro 4.17: Primeras 40 posiciones de la lista almacenada en la casilla $(5, 7)$ de la matriz de confusión al aplicar 100 repeticiones de SVM sobre el conjunto de la transformación ACC.

Al analizar las listas de las matrices de confusión de las cuatro transformaciones, se observa que las instancias mal clasificadas. Sólo alternan entre dos clases, la real y alguna otra que el clasificador asigna. Dicho de otra forma, el modelo multi-etiqueta sólo contendrá instancias asociadas a dos clases diferentes. Existen casos en los cuales alguna instancia es asignada a más de dos clases por el clasificador a lo largo del proceso iterativo, sin embargo el número de veces que esto sucede no es significativo y se puede considerar como un error de clasificación común. Dichas instancias no se toman en cuenta para el modelado multi-etiqueta de este trabajo.

Después de analizar las listas de las matrices de confusión se definen cuatro rangos a tomar en cuenta para la creación de la forma multi-etiqueta de las cuatro transformaciones. El primer rango contiene a las instancias que fueron clasificadas erróneamente las 100 veces a una clase determinada, esta versión del modelado multi-etiqueta es la más estricta, ya que sigue un punto de vista rígido. El segundo rango contempla a las instancias que fueron mal clasificadas de 90 a 100 veces ($[90, 100]$), y el tercer y cuarto rango contiene a las instancias mal clasificadas de 85 a 100 ($[85, 100]$) y de 75 a 100 ($[75, 100]$) veces, respectivamente.

Ahora existen cuatro modelos multi-etiqueta para cada una de las cuatro transformaciones, cada modelo con 1392 instancias. El cuadro 4.18 muestran el número de instancias multi-etiqueta que contiene cada transformación de acuerdo a los rangos establecidos(IA

	AAC		PseAAC		W-PseAAC		ACC	
Rango	IA	TI	IA	TI	IA	TI	IA	TI
[100]	61	61	75	75	60	60	65	65
$[90-100]$	39	100	53	128	40	100	36	101
$[85-100]$	12	112	15	159	11	111	9	110
$[75-100]$	9	121	16	175	24	135	11	121

Cuadro 4.18: Número de instancias multi-etiqueta de las transformaciones de acuerdo a su rango.

= instancias agregadas, IT = instancias totales). La segunda columna indica el número de instancias que se agregan al modelo al tomar en cuenta las instancias con menor frecuencia de error para la misma casilla de la matriz de confusión. Para las cuatro transformaciones aproximadamente la mitad de instancias multi-etiqueta caen dentro del primer rango ([100]).

4.5. Resultados utilizando clasificadores multi-etiqueta

En esta sección, los modelos multi-etiqueta de las cuatro transformaciones se someterán a un algoritmo de clasificación basado en el mismo tipo de aprendizaje. El clasificador seleccionado es k-NN multi-etiqueta (ML k-NN), los siguientes cuadros (4.19, 4.20, 4.21 y 4.22) muestran el rendimiento de las cuatro transformaciones para el mejor valor de $k \in [1, 15]$.

Rango	1-HammingLoss	1-RankingLoss	1-OneError	1-Coverage	Average_Precision
E.S.B.	**95.01**	95.26	82.72	**71.53**	90.03
[100]	94.77	95.30	83.89	66.45	90.42
$[90-100]$	94.72	95.51	84.99	64.24	90.93
$[85-100]$	94.69	95.52	85.19	63.36	91.00
$[75-100]$	94.62	**95.62**	**85.79**	62.92	**91.31**

Cuadro 4.19: Medida de exactitud promedio utilizando ML 11-NN para la transformación AAC multi-etiqueta.

En la primera columna de los cuadros, la casilla marcada como *Etiqueta Simple Binarizada* (*E.S.B.*), representa la transformación original, en la cual el vector Y de etiquetas esta representado por medio de una notación binarizada, conocida como 1 de m (por ejemplo $Y_i = 5 \rightarrow [0000100]$, donde Y_i representa la etiqueta de la i-ésima instancia x_i), la cual no contiene instancias multi-etiqueta y por tanto no existe una correlación entre la matriz de datos y la matriz de etiquetas que los algoritmos multi-etiqueta puedan explotar. La representación *E.S.B.* de cada transformación fungirá como control y servirá para poder comparar la exactitud de los modelos multi-etiqueta.

Cada cuadro contiene cinco métricas que evalúan el rendimiento del clasificador, a diferencia de la clasificación tradicional, algunas de las métricas de evaluación en el aprendizaje multi-etiqueta pueden no ser fáciles de interpretar.

Rango	1- HammingLoss	1- RankingLoss	1- OneError	1-Coverage	Average_Precision
E.S.B.	95.14	95.47	83.56	**72.81**	90.50
[100]	**95.50**	96.17	87.16	70.66	92.30
$[90-100]$	95.36	96.43	88.29	67.95	92.91
$[85-100]$	95.28	96.52	88.68	67.44	93.15
$[75-100]$	95.27	**96.59**	**89.03**	66.49	**93.35**

Cuadro 4.20: Medida de exactitud promedio utilizando ML 11-NN para la transformación PseAAC multi-etiqueta.

En la sección 2.4.4, se describieron las métricas utilizadas para medir el rendimiento de ML k-NN, sin embargo cuatro de estas métricas se enfocan en medir el error minimizando su función de evaluación correspondiente, por lo tanto, mientras más se acerquen a cero el rendimiento del clasificador es mejor. De forma contraria a lo expuesto, la métrica denominada *precisión media (Average_Precision)* maximiza su función de evaluación siendo óptima cuando el resultado es uno.

Para que los resultados se puedan comparar bajo el mismo rango se toma a las cuatro métricas (perdida Hamming (HammingLoss), perdida de jerarquía (RankingLoss), máximo error de jerarquía (OneError) y cobertura (Average) que minimiza su función de la siguiente manera: $1 - f(Y_i, O_i)$, siendo $(f(y, o))$ la función de evaluación, y y el vector de clases real, mientras que o es el vector que asigna el clasificador.

Rango	1- HammingLoss	1- RankingLoss	1-OneError	1- Coverage	Average_Precision
E.S.B.	95.22	95.53	83.47	**73.18**	90.52
[100]	**95.62**	96.30	86.83	72.59	92.22
$[90-100]$	95.49	96.59	88.28	70.82	92.96
$[85-100]$	95.55	96.64	88.63	70.04	93.17
$[75-100]$	95.29	**96.70**	**88.80**	68.09	**93.30**

Cuadro 4.21: Medida de exactitud utilizando ML 11-NN para la transformación Wavelet-PseAAC multi-etiqueta.

1- HammingLoss (*1-HL*) es equivalente a el concepto de exactitud que se maneja en el aprendizaje máquina tradicional. *1-RankingLoss* mide el rendimiento del clasificador mediante un conteo de las etiquetas que tiene mejor posición del ranking y no están asociadas a la instancia en proceso. *1-OneError* mide las veces que la etiqueta en la primera posición del ranking pertenece a las etiquetas reales de la instancia en proceso. *1-Coverage* mide el promedio de posiciones que existen en el nivel más alto del ranking que no están asociadas a etiquetas reales de cada instancia.

Average_precision (A_P) mide el número de posiciones que se tiene que avanzar en el ranking para encontrar todas las etiquetas asociadas a la instancia en cuestión, para que su rendimiento sea óptimo. Si una instancia esta asociada a k etiquetas, estas deben de estar en las primeras k posiciones en la lista del ranking devuelta por el clasificador.

Rango	1-HammingLoss	1-RankingLoss	1-OneError	1-Coverage	Average_Precision
E.S.B.	95.21	95.50	84.05	**73.02**	90.70
[100]	**95.62**	96.33	87.96	71.85	92.69
$[90-100]$	95.57	**96.54**	**89.02**	70.26	**93.20**
$[85-100]$	94.94	96.10	87.29	66.45	92.23
$[75-100]$	94.90	96.12	87.69	65.10	92.35

Cuadro 4.22: Medida de exactitud utilizando ML 11-NN para la transformación ACC multi-etiqueta.

El mayor rendimiento de los modelos multi-etiqueta para la transformación AAC con la métrica *1-HL* se obtiene con el conjunto *E.S.B.* y para las transformaciones restantes el conjunto con rango [100] es el que ofrece mejores resultados. Para la métrica A_P, en las transformaciones AAC, PseAAC y Wavelet-PseAAC el conjunto con rango $[75-100]$ es con el que se obtiene mayor rendimiento, mientras que para la transformación ACC es en el conjunto de rango $[90-100]$. Sólo se analizan estas dos métricas debido a que son las que se pueden comparar directamente con las métrica de rendimiento tradicional que se obtuvieron en la sección 4.3.

4.6. Resultados de reducción de dimensionalidad multi-etiqueta

Esta sección se enfoca en los resultados de rendimiento de clasificación para las cuatro transformaciones con sus respectivos rangos al aplicarles métodos de reducción de dimensionalidad multi-etiqueta.

	MLDA		HSL		CCA		OPLS	
	1-HL	AP	1-HL	AP	1-HL	AP	1-HL	AP
Rendimiento	98.86	98.85	98.86	98.90	98.76	98.90	98.85	98.80
Rango	E.S.B.	$[90-100]$	E.S.B.	$[90-100]$	[100]	$[90-100]$	E.S.B.	[100]
Transformación	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC

Cuadro 4.23: Mejor rendimiento con ML 3-NN para las cuatro transformaciones multi-etiqueta utilizando los diferentes algoritmos.

Debido a que los mejores resultados reportados en la sección 4.3.2 son con LDA tradicional, el primer método de reducción de dimensionalidad que se aplica a los conjuntos es la

versión multi-etiqueta de este método (MLDA, ver sección 2.4.2). Al igual que LDA tradicional el número de características en el espacio de baja dimensión al que son proyectados los datos al aplicar MLDA depende del número de clases del conjunto de datos. Debido a que los conjuntos de datos que se manejan en esta investigación contiene siete clases el nuevo espacio contendrá sólo seis dimensiones, ya que LDA obtiene k-1 hiperplanos para separar las instancias de las k clases. El cuadro 4.23 muestra el mejor rendimiento obtenido con ML k-NN aplicado a las proyecciones obtenidas por MLDA con la métrica 1-Hamming Loss y Average Precision.

Los resultados con MLDA superan los obtenidos con el enfoque tradicional. Para obtener resultados que permitan una comparación del rendimiento de algoritmos de RD multi-etiqueta se le aplica a los conjuntos de datos tres métodos más. Los métodos seleccionados son: aprendizaje espectral basado en hipergrafos (HSL), el cual es una técnica nativa de este tipo de aprendizaje, también se aplica análisis de correlación canónica (CCA) y mínimos cuadrados parciales ortogonales (OPLS).

De la misma forma que para MLDA sólo se presentan los mejores resultados para estas tres técnicas, el cuadro 4.23 muestran los resultados de rendimiento de cada algoritmo. Una ventaja de los métodos que toman en cuenta la correlación entre dos conjuntos de datos, en este caso la correlación entre la matriz de datos y la de etiquetas, es que el máximo número de características al aplicar la reducción de dimensionalidad, es el mínimo(q, p), donde q y p representan la dimensión de ambos conjuntos, datos y etiquetas, respectivamente. Los cuatro métodos de RD multi-etiqueta se basan en resolver un problema de optimización mediante un sistema propio (eigen system). Al obtener los siete valores propios de cada sistema, el valor propio con menor valor es muy pequeño comparado con los valores propios restantes y tiende a cero para todos los casos, es por ello que todos las proyecciones de baja dimensión sólo contienen seis características.

Si analizamos la información en los cuadros de rendimiento, podemos observar que el rendimiento del aprendiza multi-etiqueta obtenido para ML k-NN con los cuatro métodos de RD multi-etiqueta es muy similar. Además, para todas las pruebas el rendimiento supera el obtenido con k-NN y SVM al aplicarlos a los conjuntos sin reducción y los reducidos con los métodos de RD del aprendizaje tradicional. Cabe recalcar que aun para la versión *E.S.B.* se presenta esta situación. Lo anterior sugiere que las secuencias de GPCRs modeladas con el enfoque multi-etiqueta comparte funciones biológicas de tipo olfativo.

De la misma forma que en las pruebas de exactitud utilizando clasificadores tradicionales el rendimiento se incrementa para las transformacionales con mayor número de características descriptivas iniciales (transformación ACC). Esto indica que dicha transformación logra capturar la información relevante en la secuencias de GPCRs aún cuando el el conjunto es reducido De forma general, el mejor rendimiento para la métrica *HL* se mantiene en los conjuntos marcados con el rango *E.S.B* y [100], mientras que para la métrica *A_P* el mejor rendimiento los ofrecen los conjuntos con rango $[90 - 100]$ y $[85 - 100]$.

4.7. Comparación de resultados

La versión *E.S.B.* de los conjuntos multi-etiqueta es la forma equivalente del conjunto de datos en el aprendizaje tradicional. Como ya se mencionó anteriormente, el rendimiento de dicha versión supera al equivalente en la versión tradicional, por lo tanto, en esta sección se compara el rendimiento de las versiones *E.S.B.* con las versiones tradicionales de las diferentes transformaciones.

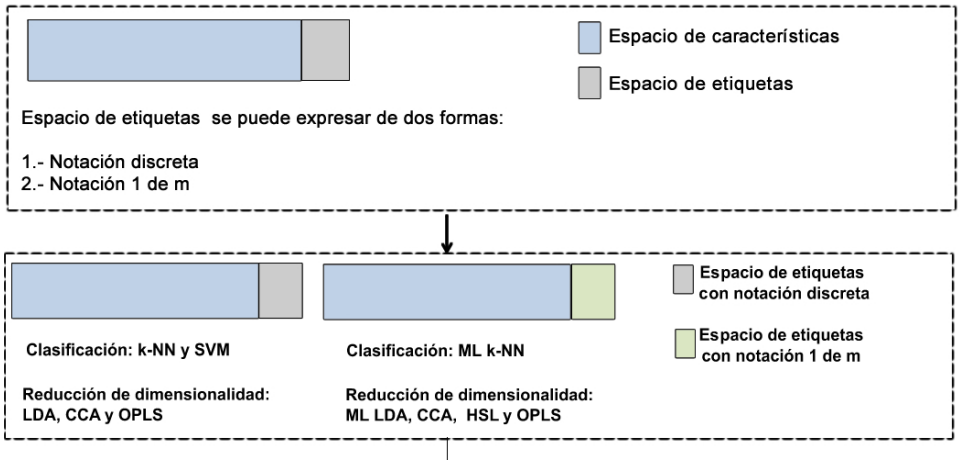


Figura 4.13: Representación de un conjunto de datos mediante un modelo de etiqueta simple y un modelo de etiquetas multiples, ambos modelos son equivalentes.

Para esta sección, al hablar de reducción y clasificación multi-etiqueta se asume que se habla de la versión tradicional del conjunto en la cual sus etiquetas están representadas en forma de un vector binario (*E.S.B.* o notación 1 de m). Por otro lado, si se hace referencia a algoritmos de clasificación y reducción tradicionales se asume que el conjunto de etiquetas es un vector que contiene números naturales (notación discreta). CCA y OPLS son capaces de trabajar con conjuntos de datos de etiqueta simple y multi-etiqueta. La figura 4.13 muestra como un conjunto de datos con un modelo de etiqueta simple puede ser representado para poder aplicar técnicas de apredizaje multi-etiqueta.

En el cuadro 4.24 se muestra el rendimiento de clasificación mediante la métrica 1-Hamming loss obtenida por ML k-NN comparado con el rendimiento de la métrica de exactitud obtenida por k-NN y SVM. El rendimiento que se obtiene mediante 1-Hammming loss para ML k-NN supera el rendimiento obtenido mediante k-NN y SVM. Esto se debe a que ML k-NN además de tomar en cuenta los k vecinos más cercanos incluye la correlación de cada etiqueta con la nueva instancia a clasificar basándose en los principios del aprendizaje bayesiano. Esto es, al agregar un enfoque probabilista que toma en cuenta tanto a las muestras de conjunto de entrenamiento como a los k-vecinos más cercanos logra optimizar los resultados de clasificación. El rendimiento de ML k-NN para las cuatro transformaciones sin reducción no varía significativamente, de forma contraria a lo que sucede en el enfoque tradicional, en el cual varía desde un 88.14 hasta un 92.53 para SVM, siendo este último método el que mejores resultados obtiene.

Transformación	ML 11-NN 1-Hamming loss	k-NN Exactitud	SVM
AAC	**95.01**	81.25 (k=1)	88.14
PseAAC	**95.14**	87.43 (k=3)	89.66
Wavelet-PseAAC	**95.22**	87.57 (k=1)	90.89
ACC	**95.21**	87.50 (k=3)	92.53

Cuadro 4.24: Rendimiento de las cuatro transformaciones sin reducción utilizando ML k-NN, k-NN y SVM

El rendimiento obtenido mediante la métrica 1-Hamming loss para ML k-NN en las transformaciones reducidas con los cuatro algoritmos multi-etiqueta (ver cuadros 4.25, 4.26, 4.27 y 4.28) es superior al rendimiento obtenido con la métrica de exactitud al reducir y clasificar con el enfoque tradicional. Para las proyecciones obtenidas al utilizar los diferentes algoritmos de RD multi-etiqueta el rendimiento con ML k-NN prácticamente es el mismo, si comparamos la misma transformación. Por ejemplo, para la transformación AAC, la exactitud de clasificación de ML k-NN es 94.01, 93.99, 93.94 y 94.06 con las proyecciones que se obtiene al plicarle MLDA, HSL, CCA y OPLS, respectivamente. Esto mismo sucede si se reduce la dimensión de ACC con las versiones tradicionales análogas y se clasifica con k-NN o SVM. Al igual que en la versión tradicional, los mejores resultados son obtenidos con la transformación ACC.

El cuadro 4.26 no muestra la comparación de rendimiento utilizando HSL en su versión tradicional, esto se debe a que HSL es un algoritmo que surge del aprendizaje multi-etiqueta y no existe una versión homóloga en el enfoque tradicional.

Los modelos multi-etiqueta con los que se trabajó hasta ahora, contienen conjuntos de etiquetas independientes para cada transformación. Dichos modelos se crearon de acuerdo a la matriz de confusión obtenida al ejecutar SVM sobre su transformación sin reducción correspondiente. Como última prueba, tomaremos la proyección obtenida de ACC-LDA, la cual representa el modelo con mejor rendimiento para el enfoque tradicional. De la misma forma que se crearon los modelos multi-etiqueta para cada transformación sin reducción se crea el modelo multi-etiqueta de la proyección ACC-LDA. A cada transformación sin reducción se le asocia el conjunto de etiquetas multiples antes mencionado y se ejecutan las pruebas de clasificación y reducción. La figura 4.14 muestra el proceso realizado.

Rango	Instancias agregadas	Total de instancias
$[100]$	32	32
$[90-100]$	7	39
$[85-100]$	1	40
$[75-100]$	2	43

Cuadro 4.29: Número de instancias multi-etiqueta de la transformación ACC-LDA de acuerdo a su rango.

Dicho modelado se puede aplicar a las cuatro transformaciones debido a que estas son

Transformación	MLDA ML 3-NN	MLDA k-NN	MLDA SVM	LDA k-NN	LDA SVM
AAC	**94.01**	78.23 (k=7)	79.53	78.45 (k=9)	79.31
PseAAC	**95.79**	85.85 (k=13)	86.78	85.78 (k=15)	86.78
Wavelet-PseAAC	**95.98**	86.85 (k=13)	86.71	86.64 (k=15)	86.71
ACC	**98.86**	96.48 (k=13)	96.70	96.48 (k=13)	96.70

Cuadro 4.25: Rendimiento de las cuatro transformaciones al ser reducidas con MLDA y LDA utilizando ML k-NN, k-NN y SVM para clasificar.

Transformación	ML 3-NN	k-NN	SVM
AAC	**93.99**	78.23 (k=7)	79.59
PseAAC	**95.90**	85.84 (k=9)	87.28
Wavelet-PseAAC	**95.90**	86.85 (k=13)	86.71
ACC	**98.86**	96.48 (k=13)	96.83

Cuadro 4.26: Rendimiento de las cuatro transformaciones al ser reducidas con HSL utilizando ML k-NN, k-NN y SVM para clasificar.

Transformación	ML CCA ML 3-NN	ML CCA k-NN	ML CCA SVM	CCA k-NN	CCA SVM
AAC	**93.94**	78.52 (k=9)	80.10	78.45 (k=9)	79.67
PseAAC	**95.71**	85.49 (k=13)	86.64	85.42 (k=11)	79.67
Wavelet-PseAAC	**96.01**	86.93 (k=15)	87.00	86.71 (k=15)	86.64
ACC	**98.84**	96.48 (k=13)	96.55	96.84 (k=9)	96.70

Cuadro 4.27: Rendimiento de las cuatro transformaciones al ser reducidas con ML CCA y CCA utilizando ML k-NN, k-NN y SVM para clasificar.

Transformación	ML OPLS ML 3-NN	ML OPLS k-NN	ML OPLS SVM	OPLS k-NN	OPLS SVM
AAC	**94.06**	85.85 (k=7)	79.31	85.70 (k=7)	79.96
PseAAC	**95.84**	85.85 (k=13)	87.28	85.70 (k=9)	86.35
Wavelet-PseAAC	**95.98**	86.85 (k=15)	86.93	86.57 (k=17)	86.57
ACC	**98.85**	96.34 (k=15)	97.05	96.26 (k=7)	96.70

Cuadro 4.28: Rendimiento de las cuatro transformaciones al ser reducidas con ML OPLS y OPLS utilizando ML k-NN, k-NN y SVM para clasificar.

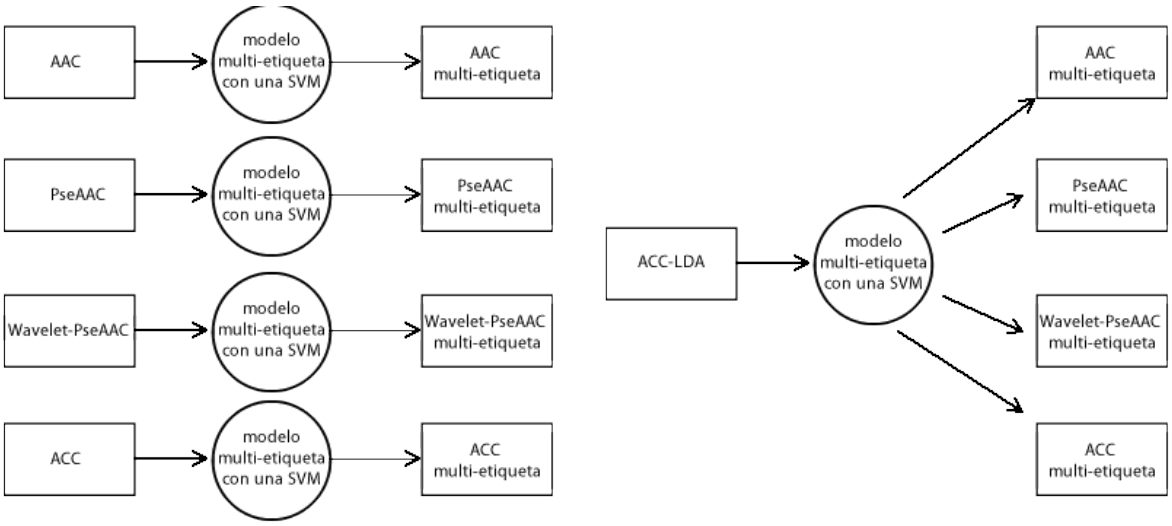


Figura 4.14: Modelados multi-etiqueta implementados para las pruebas de reducción y clasificación. (a) Primera versión del modelado multi-etiqueta, la cual modela el espacio de etiquetas de forma individual para cada transformación. (b) Segunda versión del modelado multi-etiqueta, para esta versión se modela el espacio de etiquetas de la proyección obtenida al aplicar LDA a la transformación ACC. El nuevo espacio de etiquetas múltiples obtenido es asignado a las cuatro transformaciones sin reducción.

equivalentes entre si, tanto las transformaciones de origen, como las proyecciones obtenidas al aplicar algún método de reducción. Esto se debe a que los algoritmos de reducción de dimensionalidad capturan la dimensión intrínseca de los datos de alta dimensión, conservado sólo las propiedades del conjunto de datos que más información aportan al criterio utilizado por el algoritmo correspondiente. Como consecuencia, el nuevo conjunto de datos no conserva características superfluas. Al modelar la forma multi-etiqueta partiendo de conjunto de datos ACC-LDA se espera un comportamiento similar al obtenido con los modelos creados con los conjuntos no reducidos. El cuadro 4.29 muestra el número de instancias que se agregan a los modelos al tomar en cuenta los diversos porcentajes que clasificación errónea en la matriz de confusión de las 100 iteraciones. Estos modelos contienen un menor número de instancias que los modelos multi-etiqueta anteriores ya que el rendimiento de ACC-LDA con SVM es significativamente mayor que el obtenido con SVM utilizando las transformaciones sin reducción.

Rango	1-HammingLoss	Average_Precision	Reducción
[100]	96.00	91.67	sin reducción
[100]	98.57	97.92	MLDA
[100]	**98.72**	**97.94**	HSL
[100]	98.60	97.70	ML CCA
[100]	98.53	97.70	ML OPLS

Cuadro 4.30: Rendimiento de los modelos multi-etiqueta de la transformación ACC con y sin reducción obtenidos de ACC-LDA

El cuadro 4.30 muestra el mejor rendimiento de los conjuntos reducidos con el nuevo modelo multi-etiqueta. Para todos los modelos los conjuntos con rango [100] son los que arrojan un mayor rendimiento, tanto para las versiones no reducidas, como para las versiones reducidas con los cuatro algoritmos de RD multi-etiqueta. El porcentaje de clasificación correcta es muy similar a los resultados obtenidos con los modelos multi-etiqueta anteriores.

4.8. Complejidad computacional

La complejidad computacional de las técnicas de reducción de dimensionalidad aplicadas a las transformaciones dependen de dos factores: el número de muestras (N) y la dimensión de los datos (D). La solución de las seis técnicas que muestra el cuadro 4.31 se basa en resolver una sistema propio mediante descomposición de valores singulares.

Por ejemplo, PCA genera una matriz de varianza-covarianza de DxD, la obtención de los valores y vectores propios de esta matriz exije una complejiad computacional $O(D^3)$ y un cantidad de espacio de orden $O(D^2)$. Dichas complejidades en tiempo y espacio pueden cambiar si $N < D$ tomando la siguiente forma: $O(N^3)$ y $O(N^2)$, respectivamente [Strange and Zwiggelaar, 2014]. De forma similar, en las técnicas de reducción que toman en cuenta tanto el espacio de características como el espacio de etiquetas, el mayor poder de cómputo recae en la resolución del sistema propio de las matrices que obtiene el algortimo en cuestión [Sun et al., 2009].

Técnica	Paramétrica	Parámetros	Complejidad en tiempo de cómputo
PCA	Si	No	$O(D^3)$
LDA	Si	No	$O(D^3)$
MLDA	No	No	$O(D^3)$
HSL	No	No	$O(D^3)$
CCA	Si	No	$O(D^3)$
OPLS	Si	No	$O(D^3)$

Cuadro 4.31: Análisis de la complejidad de los algoritmos utilizados en los experimentos

El cuadro 4.31 muestra un análisis de los diferentes algoritmos de reducción de dimen-

sionalidad. La segunda columna del cuadro 4.31 indica si el método es paramétrico o no lo es. Aquí se entiende que, las técnicas multivariantes que sólo funcionan adecuadamente bajo ciertas configuraciones geométricas de los datos, son denominadas métodos paramétricos. Si los parámetros en el conjunto de datos que se analiza no presentan las condiciones necesarias que requiere el método, estas técnicas no proporcionarán resultados adecuados. En particular, las técnicas de reducción de dimensión, y de categorización deben ser adecuadas para la tipología de los datos. Por ejemplo, LDA exige una distribución normal de los datos. Además los grupos dentro del conjunto de datos deben de tener varianza similares, medias distintas y no contener outliers. Es por ello que las técnicas paramétricas exigen un análisis previo de lo datos para observar si cumplen con los parámetros que exige el algoritmo. Por otro lado, cuando el algoritmo no es paramétrico no hay necesidad de un análisis estadístico previo, sin embargo, para estas técnicas no es posible medir la cantidad de información que es retenida en el espacio de baja dimensión [Van Der Maaten et al., 2008].

La tercera columna del cuadro muestra si el algoritmo necesita parámetros de entrada para realizar su función. Dicho cuadro muestra que los seis algoritmos utilizados no contienen parámetros que necesitan ser optimizados. El contener parámetros a optimizar influyen directamente en la función de coste y en el tiempo de ejecución del algoritmo. Las técnicas sin parámetros son flexibles al tratar con los datos, no obstante, si fuera necesario optimizar el algoritmo para un caso específico es necesario realizar cambios directamente en la forma de trabajar de dicho algortimo [Van Der Maaten et al., 2008].

La última columna muestra la complejidad computacional en tiempo de las diferentes técnicas. Conocer la complejidad computacional de una técnica de reducción de la dimensionalidad es de importancia para su aplicabilidad práctica. Como se mencionó en parrafos anteriores, la complejidad computacional de una técnica de reducción de la dimensionalidad está determinada por las propiedades del conjunto de datos. Además, tambien se debe de tomar en cuenta (si es el caso) los parámetros a optimizar, tales como la dimensión del nuevo espacio, k vecinos más cercanos (para técnicas basadas en grafos), el número de iteraciones (para técnicas iterativas), parámetros de regularización, etc.

Otro aspecto a tomar en cuenta es la cantidad de espacio en memoria que necesita el algoritmo. Si la memoria o los recursos computacionales requeridos son demasiado grandes, la aplicación se vuelve imposible. Para el caso de los GPCRs de clase C, el número de muestras es $1,392$, por lo tanto el espacio requerido no conlleva un problema que se deba analizar a fondo con las técnicas propuestas. De forma general, el espacio en memoria necesario es de orden $O(D^2)$, ya que los algoritmos sólo necesitan almacenar las matrices de dimensión D para el análisis del sistema propio. En [Cai et al., 2008, Ewerbring and Luk, 1989, Strange and Zwiggelaar, 2014, Sun et al., 2009, Van Der Maaten et al., 2008] se encuentra el análisis detallado de la complejidad de los algoritmos mencionados.

Otra característica de los métodos utilizados es que se basan en una función lineal para obtener las proyecciones de baja dimensión. Por ejemplo, LDA se basa en la suposición de que los datos son separables mediante un hiperplano lineal. PCA utiliza una combinación lineal de las variables que maximizan la variabilidad del sistema [Shawe-Taylor and Cristianini, 2004]. Por otro lado, cabe mencionar que el análisis de correlación canónica esta fuertemente relacionado con algunas técnicas de análisis multivariante como son: PCA,

LDA, PLS, regresión múltiple y análisis factorial. Bajo ciertas restricciones se puede considerar a CCA como un caso especial de las técnicas antes citadas, incluso del aprendizaje espectral basado en hipergrafos (HSL) [Sun et al., 2013].

Capítulo 5

Conclusiones y trabajo a futuro

Los resultados obtenidos para discriminar las subfamilias de la clase C de GPCRs con el enfoque tradicional indican que es necesario utilizar métodos más sofisticados si se pretende mejorar la exactitud de clasificación utilizando reducción de dimensionalidad.

El método tradicional que mejor resultados arroja es LDA aplicado a la transformación ACC, debido a que se enfoca en separar los ejemplos de acuerdo a su clase mediante una proyección en una dimension más baja. Esto aunado a que ACC contiene en sus variables descriptivas información de orden de las secuencias de aminoácidos. Lo anterior es debido a que LDA crea un hiperplano discriminante que maximiza la variabilidad entre instancias de diferentes clases al mismo tiempo que minimiza la varianza de las instancias asociadas a una misma clase. Al aplicar SVM a la proyección ACC-LDA, el rendimiento no se incrementa significativamente como lo hace con el resto de las transformaciones. Lo dicho anteriormente se debe a que las clases 4, 5 y 7 ya no pueden ser separadas con el kernel Gaussiano que utiliza SVM.

El suponer que las tres clases antes mencionadas comparten el espacio de hipótesis, nos lleva al enfoque multi-etiqueta, el cual ofrece resultados superiores a los reportados en la literatura, aun para el conjunto con el modelo *E.S.B.*. Lo anterior sugiere que las secuencias de GPCRs modeladas con dicho enfoque comparten funciones biológicas de tipo olfativo. Otro resultado de los experimentos, es que el rendimiento de los proyecciones obtenidas con los cuatro métodos de RD multi-etiqueta es consistente para los dos enfoques de modelado multi-etiqueta que se plantearon. Como punto final se puede decir que, aún si la forma de modelar las transformaciones a su versión multi-etiqueta puede llegar a ser difusa o incierta, se debe tomar en cuenta que seguimos un enfoque de cómputo suave. Por otro lado, los resultados de acuerdo a los rangos planteados son muy cercanos al conjunto definido como control (*etiqueta simple binarizada*), lo que nos lleva a concluir que el enfoque multi-etiqueta para los GPCRs de clase C es viable.

Como trabajo a futuro se pretende aplicar a las transformaciones variantes de LDA no lineales a la transformación ACC, principalmente, para observar si logran una mejor separación de las clases 4, 5 y 7. También, es necesario modelar las versiones multi-etiqueta de las cuatro transformaciones partiendo del mejor modelo del conjunto denominado *etiqueta simple binarizada* y observar el comportamiento de los algoritmos de clasificación y RD multi-etiqueta. De igual forma, los resultados obtenidos en este trabajo deben ser

validados por expertos del área de Biología y/o Bioquímica.

Bibliografía

A.M. Afzal, H.Y. Mussa, R.E. Turner, A. Bender, and R.C. Glen. A multi-label approach to target prediction taking ligand promiscuity into account. *J. Cheminformatics*, 7(24): 1–14, 2015.

S. Agarwal, K. Branson, and S. Belongie. Higher Order Learning with Graphs. *Proceedings of the 23rd International Conference on Machine Learning*, pages 17–24, 2006.

E. Alpaydin. *Introduction to Machine Learning*. MIT Press, second edition, 2010.

J. Arenas-García, K.B. Petersen, G. Camps-Valls, and L.K. Hansen. Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods. *CoRR*, 2013.

W.R. Atchley, J. Zhao, A.D. Fernandes, and T. Drüke. Solving the protein sequence metric problem. *Proceedings of the National Academy of Sciences*, 102(18):6395–6400, 2005.

B. Bakir and O.U. Sezerman. Functional Classification of G-Protein Coupled Receptors, Based on Their Specific Ligand Coupling Patterns. *Applications of Evolutionary Computing*, 3907:1–12, 2006.

P.J. Barnes. Receptor heterodimerization: a new level of cross-talk. *Journal of Clinical Investigation*, 116(5):1210–1212, 2006.

E. Becker, B. Robisson, Ch.E. Chapple, A. Guénoche, and Brun Ch. Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Biochemical Society Transactions*, 28(1):84–90, 2012.

J.M. Bécu, J. Pelé, P. Rodien, H. Abdi, and M. Chabbert. Structural evolution of G-protein-coupled receptors: a sequence space approach. *Methods in Enzymology*, 520: 49–66, 2013.

J. Bell. *Machine Learning: Hands-On for Developers and Technical Professionals*. Wiley, 1st edition, 2014.

R. Bellman and R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1st edition, 1961.

J.M. Berg, L. Stryer, J.L. Tymoczko, and N.D. Clarke. *Biochemistry*. W.H. Freeman, 2002.

C.M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 1st edition, 2006.

J. Bockaert and J.P. Pin. Molecular tinkering of G protein coupled receptors : an evolutionary success. *EMBO Journal*, 18(7):1723–1729, 2000.

D.O. Borroto-Escuela, A.O. Tarakanov, D. Guidolin, F. Ciruela, L.F. Agnati, and K. Fuxe. Moonlighting characteristics of G protein-coupled receptors: Focus on receptor heteromers and relevance for neurodegeneration. *International Union of Biochemistry and Molecular Biology life*, 63(7):463–472, 2011.

Deng Cai, Xiaofei He, and Jiawei Han. *Training linear discriminant analysis in linear time*, pages 209–217. 2008.

M.I. Cárdenas, A. Vellido, C. König, R. Alquézar, and J. Giraldo. Exploratory visualization of misclassified gpcrs from their transformed unaligned sequences using manifold learning techniques. *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 623–630, 2014.

M.I. Cárdenas, A. Vellido, and J. Giraldo. Visual exploratory assessment of class C GPCR extracellular domains discrimination capabilities. *10th International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 31–39, 2016.

M.I. Cárdenas, A. Vellido, I. Olier, X. Rovira, and J. Giraldo. Complementing kernel-based visualization of protein sequences with their phylogenetic tree. *J. Cheminformatics*, 7: 136–149, 2016.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 2011.

O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 1st edition, 2006.

Ch.E. Chapple, C. Herrmann, and C. Brun. PrOnto database: GO term functional dissimilarity inferred from biological data. *Frontiers Genet.*, 6(200), 2015a.

Ch.E. Chapple, B. Robisson, L. Spinelli, C. Guien, E. Becker, and C. Brun. Extreme multifunctional proteins identified from a human protein interaction network. *Nature Commun*, 6(7412), 2015b.

W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2):211–225, 2009.

A. Clare and R.D. King. Knowledge Discovery in Multi-Label Phenotype Data. *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53, 2001.

R. Cruz-Barbosa, A. Vellido, and J. Giraldo. Advances in Semi-Supervised Alignment-Free Classication of G Protein-Coupled Receptors. *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 759–766, 2013.

R. Cruz-Barbosa, A. Vellido, and J. Giraldo. The influence of alignment-free sequence representations on the semi-supervised classification of class C G protein-coupled receptors. *Med. Biol. Engineering and Computing*, 53(2):137–149, 2015.

P. Cunningham. Dimension Reduction. Technical report, University College Dublin, 2007.

K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. *Proceedings of the 27th International Conference on Machine Learning*, pages 279–286, 2010.

A.S. Doré, K. Okrasa, J. C. Patel, M. Serrano-Vega, K. Bennett, R.M. Cooke, J.C. Errey, A. Jazayeri, S. Khan, B. Tehan, M. Weir, G.R. Wiggin, and G.R. Marshall. Structure of class C GPCR metabotropic glutamate receptor 5 transmembrane domain. *Nature Reviews Drug Discovery*, 511:557–562, 2014.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.

B. Everitt and T. Hothorn. *Introduction to applied multivariate analysis with R.* Springer, 1st edition, 2011.

L. M. Ewerbring and F. T. Luk. Canonical correlations and generalized svd: Applications and new algorithms. *Proc. SPIE*, 0977:206–222, 1989.

R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(7):179–188, 1936.

D.R. Flower and T. K. Attwood. Integrative bioinformatics for functional genome annotation : trawling for G protein-coupled receptors. *Seminars in Cell & Developmental Biology*, 15(6):693–701, 2004.

R. Fredriksson and H.B. Schiöth. G Protein-coupled Receptors in the Human Genome. *Molecular Pharmacology*, 63(6):1256–72, 2003.

J. Friedman, editor. *Expanding Graphs*, volume 10, 1992.

J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label Ranking. *J. Machine Learning*, 73(2):133–153, 2008.

K. Fuxe, D.O. Borroto-Escuela, W. Romero-Fernandez, and M. Palkovits. Moonlighting proteins and protein-protein interactions as neurotherapeutic targets in the G protein-coupled receptor field. *Neuropsychopharmacology*, 39(1):131–155, 2014.

Q.-B. Gao and Z.-Z. Wang. Classification of G-protein coupled receptors at four levels. *Protein Engineering Design and Selection*, 19(11):511–516, 2006.

J. Gareth, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer, 1st edition, 2014.

N. Ghamrawi and A. McCallum. Collective multi-label classification. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 195–200, 2005.

C. Giacovazzo. *Fundamentals of crystallography*. Oxford University Press, 3rd edition, 2011.

S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled. *Advances in Knowledge Discovery and Data Mining*, 3056:22–30, 2004.

A. Gómez, N. Domedel, J. Cedano, J. Piñol, and E. Querol. Do current sequence analysis algorithms disclose multifunctional (moonlighting) proteins?. *Molecular BioSystems*, 19(7):895–896, 2003.

A. Gómez, S. Hernández, I. Amela, J. Piñol, J. Cedano, and E. Querol. Do protein–protein interaction databases identify moonlighting proteins?. *Molecular BioSystems*, 7(8):2379–2382, 2011.

P. Gonzales Gil. Receptores acoplados a proteínas G: Entendiendo cómo responde nuestro organismo a señales diversas. *Revista de Química*, 26:1–2, 2013.

H. Harold. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

D.M. Hosmer and S. Lemeshow. *Applied logistic regression*. John Wiley & Sons, 1st edition, 2000.

L.M. Iakoucheva, C.J. Brown, J.D. Lawson, Z. Obradovic, and A.K. Dunker. Intrinsic disorder in cell-signalling and cancer–associated proteins. *J. Molecular Biology*, 323: 573–584, 2002.

V. Isberg, Vroling B., R. Van der Kant, K. Li, G. Vriend, and D. Gloriam. GPCRDB: an information system for G protein-coupled receptors. *Nucleic Acids Research*, 42(7): 179–188, 2014.

A.J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer, 1st edition, 2006.

C.J. Jeffery. Moonlighting proteins. *Genome Biology*, 24:8–11, 1999.

C.J. Jeffery. Moonlighting proteins: old proteins learning new tricks. *Trends Genet*, 19: 415–418, 2003.

C.J. Jeffery. Molecular mechanisms for multitasking: recent crystal structures of moonlighting proteins. *Current Opinion in Structural Biology*, 14:663–668, 2004.

I.T. Jolliffe. *Principal Component Analysis*. Springer, 2da edition, 2002.

S.D. Kahn. On the future of genomic data. *Science*, 331(6016):728–729, 2011.

R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein coupled receptors with support vector machines. *Genomics*, 18(1), 2002.

V. Katritch, V. Cherezov, and R.C. Stevens. Structure-Function of the G-protein-Coupled Receptor Superfamily. *Annual Review of Pharmacology and Toxicology*, 53(6):531–556, 2014.

S. Kawashima and M. Kanehisa. AAindex: Amino acid index database. *Nucleic Acids Research*, 374(28):179–188, 2000.

V. Kecman. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press, 1st edition, 2001.

I.K. Khan and D. Kihara. Computational characterization of moonlighting proteins. *Biochemical Society Transactions*, 42(6):1780–1785, 2014.

I.K. Khan, M. Chitale, C. Rayon, and D. Kihara. Evaluation of function predictions by PFP, ESG, and PSI–BLAST for moonlighting proteins. *BMC Proceedings*, 6 (Supplement 7):S5, 2012.

I.K. Khan, Y. Chen, T. Dong, X. Hong, R. Tekeuchi, H. Mori, and D. Kihara. Genome–scale identification and characterization of moonlighting proteins. *Biol. Direct.*, 9(30): 1:30, 2014.

R. Khattree and D.N. Naik. *Multivariate data reduction and discrimination with SAS software*. John Wiley & Sons, 1st edition, 2000.

L.F. Jr. Kolakowski. GCRDb: a G-protein-coupled receptor database. *Receptors & channels*, 2(1):1–7, 1994.

C. König, R. Cruz-Barbosa, R. Alquézar, and A. Vellido. SVM-Based Classification of Class C GPCRs from Alignment-Free Physicochemical Transformations of their Sequences. pages 336–343, 2013.

C. König, R. Alquézar, A. Vellido, and J. Giraldo. Reducing the n-gram feature space of class C GPCRs to subtype-discriminating patterns. *Journal of integrative bioinformatics*, 11, 2014a.

C. König, A. Vellido, R. Alquézar, and J. Giraldo. Misclassification of class C G–protein-coupled receptors as a label noise problem. *ESANN 2014*, pages 695–700, 2014b.

C. König, M.I. Cárdenas, J. Giraldo, R. Alquézar, and A. Vellido. Label noise in subtype discrimination of class C G protein-coupled receptors: A systematic approach to the analysis of classification errors. 16:314, 2015.

E.V. Koonin and M.V. Galperin. *Sequence–Evolution–Function: Computational Approaches in Comparative Genomics.* Kluwer Academic, 2003.

M. Lapinsh, A. Gutcaits, P. Prusis, C. Post, T. Lundsted, and J. Wikberg. Classification of G-protein coupled receptors by alignment independent extraction of principal chemical properties of primary amino acid sequences. *Protein science*, 374(11):795–805, 2002.

J.A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 1st edition, 2007.

B. Lewin, J.E. Krebs, E.S. Goldstein, and S.T. Kilpatrick. *Lewin's Genes X*. Jones and Bartlett, 1st edition, 2009.

N. Locantore, J. Marron, and D. Simpson. Robust principal component analysis for functional data. *TEST– An Official Journal of the Spanish Society of Statistics and Operations Research*, (1):1–73, 1999.

O. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, 1st edition, 2005.

S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

C.K. Mathews, K.E. Van Holde, D.R. Appling, and S.J. Anthony-Cahill. *Biochemistry*. Prentice Hall, 4ta edition, 2013.

A.K. McCallum. Multi-label text classification with a mixture model trained by EM. *AAAI 99 Workshop on Text Learning*, 1999.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1st edition, 1997.

S. Mitra and T. Acharya. *Data mining - multimedia, soft computing, and bioinformatics*. Wiley, 1st edition, 2003.

L. Nanni, A. Lumini, D. Gupta, and A. Garg. Identifying Bacterial Virulent Proteins by Fusing a Set of Classifiers Based on Variants of Chou's Pseudo Amino Acid Composition and on Evolutionary Information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(2):467–475, 2012.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, pages 849–856, 2001.

I. Nobeli, A.D. Favia, and J.M. Thornton. Protein promiscuity and its implications for biotechnology. *Nature Biotechnology* , 27:157–167, 2009.

M. Orozco. A theoretical view of protein dynamics. *Chemical Society Reviews*, 43:5051–5066, 2014.

J.P. Overington, B. Al-Lazikani, and A.L. Hopkins. How many drug targets are there? *Nature Reviews Drug Discovery*, 5(12):993–996, 2006.

K. Palczewski. G protein-coupled receptor rhodopsin. *Annual Review of Biochemistry*, 75:743–767, 2006.

R. Panettaand and M.T. Greenwood. Physiological relevance of GPCR oligomerization and its impact on drug discovery. *Drug Discovery Today*, 13(23-24):1059–1066, 2008.

Z.-L. Peng, J.-Y. Yang, and X. Chen. An improved classification of G-protein-coupled receptors using sequence-derived features. *BMC Bioinformatics*, 11:420–433, 2010.

J. Piatigorsky. *Gene Sharing and Evolution: The Diversity of Protein Function.* Harward University Press, 1st edition, 2007.

J.P. Pin, T. Galvez, and L. Prézeau. Evolution, structure, and activation mechanism of family 3/C G-protein-coupled receptors. *Pharmacology and Therapeutics*, 98(3):325–354, 2003.

G. Puja and K. Neha. An Introduction of soft Computing approach over Hard Computing. *International Journal of Latest Trends in Engineering and Technology* , 3(1):254–258, 2013.

E.G. Ramos Pérez. Aprendizaje de representaciones de secuencia deaminoácidos utilizando arquitecturas profundas. Master's thesis, Universidad Tecnológica de la Mixteca, 2016.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2011.

J. Read, A. Bifet, G. Holmes, and B. Pfahringer. Scalable and Efficient Multi-label Classification for Evolving Data Streams. 88(1):243–272, 2012.

A.C Rencher. *Methods of Multivariate Analysis.* Wiley, 2da edition, 2002.

S. Rosenberg. *The Laplacian on a Riemannian manifold : an introduction to analysis on manifolds.* Cambridge University Press, 1st edition, 1997.

C. Sanderson. Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments. Technical report, NICTA, 2010.

P. Sarlin and T.A. Peltonen. Mapping the state of Financial Stability. *Working PaPer SerieS*, 1382:32–36, 2011.

E. Schad, P. Tompa, and H. Hegyi. The relationship between proteome size, structural disorder and organism complexity. *Genome Biology*, 12(12):R120, 2011.

R.E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2):135–168, 2000.

D.W. Scott and J.R. Thompsonm. Probability density estimation in higher dimensions. *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, (3):173–179, 1983.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 1st edition, 2004.

T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

R.E. Stenkamp, D.C. Teller, and K. Palczewski. Crystal Structure of Rhodopsin : A G-Protein-Coupled Receptor. *Science*, 6485(5480):963–967, 2002.

H. Strange and R. Zwiggelaar. *Open Problems in Spectral Dimensionality Reduction*. Springer, 2014.

L. Sun, S. Ji, and J. Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 977–984, 2009.

L. Sun, S. Ji, and J. Ye. *Multi-label dimensionality reduction*. Chapman and Hall/CRC, 1st edition, 2013.

P. Tompa. Intrinsically unstructured proteins. *Trends in Biochemical Sciences*, 27:527–533, 2002.

G. Tsoumakas and I. Vlahavas. Random k-Labelsets: An Ensemble Method for Multilabel Classification. *Proceedings of the 18th European Conference on Machine Learning*, 4701: 406–417, 2007.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *In Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

Z. Ur-Rehman and A. Khan. G-protein-coupled receptor prediction using pseudo-amino-acid composition and multiscale energy representation of different physiochemical properties. *Analytical biochemistry*, 412:173–182, 2011.

L.J.P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik. Dimensionality reduction: A comparative review, 2008.

J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.

C. Vogel and C. Chothia. Protein Family Expansions and Biological Complexity. *PLoS Comput Biol*, 2(5):48, 2006.

S. Wan, M.W. Mak, and S.Y. Kung. mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines. *BMC Bioinformatics*, 13: 290, 2012.

H. Wang, C. Ding, and H. Huang. Multi-Label Classification: Inconsistency and Class Balanced K-Nearest Neighbor. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010.

J.J. Ward, J.S. Sodhi, L.J. McGuffin, B.F. Buxton, and D.T. Jones. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *Trends in Biochemical Sciences*, 26:635–645, 2004.

J. Weston. A kernel method for multi-labelled classification. *In Advances in Neural Information Processing Systems 14*, 14:681—-687, 2001.

T. Wieland and C. Mittmann. Regulators of G-protein signalling: multifunctional proteins with impact on signalling in the cardiovascular system. *Pharmacol Ther*, 2(97):95–115, 2003.

H. Wold. Estimation of principal components and related models by iterative least squares. volume 59, pages 391–420. 1966.

L.A. Zadeh. Fuzzy Logic, Neural Networks, and Soft Computing. *Communications of the ACM*, (3):77–84, 1994.

M.L. Zhang and Z.H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Reconigtion*, 40:2038–2048, 2007.

D. Zhou, J. Huang, and B. Scholkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems 19*, 2006.

J.Y. Zien, M.D.F. Schlag, and P.K. Chan. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEEIEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9):1389–1399, 1999.

Anexos

Anexo A

Manual de usuario de la biblioteca desarrollada

En este anexo se detalla como utilizar las clases y métodos de la biblioteca desarrollada en el presente proyecto de tesis, también se ejemplifica como clasificar y reducir la dimensionalidad de la transformación ACC con MLkNN y CCA respectivamente.

A.1. Proceso de instalación

Para poder utilizar la biblioteca desarrollada (ml.h) es necesario instalar una biblioteca adicional. Armadillo [Sanderson, 2010] es una biblioteca de álgebra lineal utilizada en las áreas de aprendizaje automático, reconocimiento de patrones, visión por computadora, procesamiento de señales, bioinformática, entre otras. Armadillo es una biblioteca de código abierto con una sintaxis similar a la de Matlab y esta diseñada para plataformas Linux, Windows y Mac OS. Los pasos para instalar dicha biblioteca sobre la consola de ubuntu son los siguientes:

Código A.1: Comandos para instalar la biblioteca armadillo sobre consola

```
$sudo apt-add-repository ppa:comp-phys/stable
$sudo apt-get update
$sudo apt-get install libarmadillo-dev
```

Una vez instalada la biblioteca armadillo, sólo es necesario incluir el archivo ***ml.h*** desarrollado en este proyecto dentro del código fuente donde se desea utiliza MLkNN o CCA, cómo se ilustra en el código A.2.

Código A.2: Integración de la biblioteca ml.h

```
1 #include "ml.h"
```

A.2. Ejemplo práctico

El código A.3 ilustra la forma de utilizar los métodos de las clases mlknn y cca. En la linea 9 y 10 se instancían los objetos de la clase mlknn y cca, respectivamente. Posteriormente se lee el conjunto de datos (linea 12) y realiza el entrenamiento con k-NN multi-etiqueta utilizando validación cruzada de 10 iteraciones (linea 13). En la linea 14 se imprime el valor promedio de la métrica Hamming loss que se obtuvo en el la linea anterior.

En la linea 18 se aplica la reducción de dimensionalidad a través del método de análisis de correlación canónica al conjunto de datos de ACC, que se cargo en memoria en la linea anterior (linea 17). El método *canon_corr* toma como parámetro el número de componentes canónicas a tomar en cuenta, las cuales son sinónimo del número de características en el nuevo espacio reducido. Después de realizar la reducción, la proyección de datos obtenida se guarda en un archivo llamado *ACC_CCA.csv* (linea 21). En la linea 24 se carga en memoria dicho archivo para aplicar el algoritmo k-NN multi-etiqueta y obtener el rendimiento con el nuevo conjunto de datos mediante validación cruzada (linea 25). El método *reset* de ambas clases (lineas 22, 29 y 30) elimina el contenido que almacenan las variables dentro de la clase correspondiente para que puedan volver a ser utilizadas sin conflictos de memoria.

Código A.3: Ejemplo de uso de la biblioteca ml.h

```
1 #include <iostream>
2
3 #include "ml.h"
4 using namespace std;
5
6 int main()
7 {
8
9  mlknn knnml();
10  cca acc();
11
12  knnml.read_file("ACC.csv",1392,7,325);\\leer conjunto de datos
13  knnml.k_cross_validation(10,11); \\hacer validación cruzada de 10 iteraciones (10-VC) con 11-NN
14  knnml.print_hl_kcv(); \\imprimir el promedio de la validación
15                                \\ de la métrica hamming loss
16
17  acc.read_file("ACC.csv",1392,7,325); \\leer conjunto de datos
18  acc.canon_corr(6);  \\reducir utilizando las primeras 6
19                              \\componentes canónicas
20
21  acc.write_cca_data("ACC_CCA.csv"); \\escribir el conjunto reducido a un archivo
22  knnml.reset();
23
24  knnml.read_file("ACC_CCA.csv",1392,7,6);\\leer conjunto de datos
25  knnml.k_cross_validation(10,11); \\hacer 10-VC con 11-NN
26
27  knnml.print_hl_kcv(); \\imprimir el promedio de la validación
28                            \\ de la métrica hamming loss
29  knnml.reset();
30  acc.reset();
31
32 return 0;
33 }
```

Si el código anterior se guarda en un archivo llamado reducdim.cpp podemos compilarlo y posteriormente ejecutar el código bajo Linux o Mac OS ejecutando en la terminal las siguientes ordenes:

Código A.4: Compilar y ejecutar el código fuente sobre consola

```
$g++ reducdim.cpp -o reducdim -O2 -larmadillo
$./reducdim
```

Lo anterior produce la siguiente salida en la pantalla de la terminal:

Código A.5: Parte de la salida en la pantalla de la terminal al ejecutar el código A.4

```
Fold :::::::::::::::::: > > > > > > > > > 7 de 10
Training ....
Predict ....
Hamming Loss = 91.48
: : : : : : : : : :: : : : : : :: : : : : :: : :
Fold :::::::::::::::::: > > > > > > > > > 8 de 10
Training ....
Predict ....
Hamming Loss = 91.44
: : : : : : : : : :: : : : : : :: : : : : :: : :
Fold :::::::::::::::::: > > > > > > > > > 9 de 10
Training ....
Predict ....
Hamming Loss = 91.67
: : : : : : : : : :: : : : : : :: : : : : :: : :
Fold :::::::::::::::::: > > > > > > > > > 10 de 10
Training ....
Predict ....
Hamming Loss = 91.71
: : : : : : : : : :: : : : : : :: : : : : :: : :
Hamming loss 10 cross fold validation: 91.61
```