REDUCCIÓN DE LA DIMENSIONALIDAD PARA OPTIMIZAR LA CLASIFICACIÓN DE DATOS FUNCIONALES

Por

Shirley Yohany Huanca Ochoa

Tesis sometida en cumplimiento parcial de los requerimientos para el grado de

MAESTRÍA EN CIENCIAS

en

MATEMÁTICA(ESTADÍSTICA)

UNIVERSIDAD DE PUERTO RICO
RECINTO UNIVERSITARIO DE MAYAGÜEZ

2015

Aprobada por:

Dámaris Santana Morant, Ph.D. Miembro, Comité Graduado	Fecha
Wolfgang Rolke, Ph.D. Miembro, Comité Graduado	Fecha
Edgar Acuña Fernández, Ph.D. Presidente, Comité Graduado	Fecha
Héctor Méndez Mella, Ph.D. Representante de Estudios Graduados	Fecha
Olgamary Rivera Marrero, Ph.D. Directora Interina del Departamento	Fecha

Abstract of Disertación Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Science

DIMENSIONALITY REDUCTION TO OPTIMIZE FUNCTIONAL CLASSIFICATION DATA

By

Shirley Yohany Huanca Ochoa

2015

Chair: Edgar Acuña Fernández
Major Department: Mathematical Sciences

Nowdays throw due to the continuous advance of technology, statisticians have been facing the need to develop new methods to extract meaningful information quickly and efficiently in large data sets, such as functional data. This type of data corresponds to a random observation over an interval. This data are treated theoretically using the definitions and properties of curves; as well as computationally through data mining techniques treating them as high-dimensional vectors. It is in this sense that the application of some methods of feature selection will be advisable prior to any analysis of such data. In this technique a representation of finite size for each curve is used, thus overcoming the problem of high dimensionality.

In this work we compare three feature selection procedures with a commonly used reduction dimensionality method for functional data. Results will be presented using two real datasets. This is done in order to compare the effectiveness to minimize the error rate of misclassification in the datasets. The results obtained in this thesis

show that the dimensionality reduction using B-Splines yields a better performance that feature selection.

Resumen de Disertación Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Maestría en Ciencias

REDUCCIÓN DE LA DIMENSIONALIDAD PARA OPTIMIZAR LA CLASIFICACIÓN DE DATOS FUNCIONALES

Por

Shirley Yohany Huanca Ochoa

2015

Consejero: Edgar Acuña Fernández
Departamento: Ciencias Matemáticas

Con el paso del tiempo y con el continuo avance de la tecnología, los estadísticos se han enfrentado a la necesidad de desarrollar nuevos métodos para extraer información significativa de forma rápida y eficiente en grandes conjuntos de datos, tal como son los datos funcionales. Este tipo de datos corresponden a una observación aleatoria en un intervalo; y por tanto son tratados teóricamente utilizando las definiciones y propiedades de curvas; así como computacionalmente a través de técnicas de minería de datos, considerándolos como vectores de alta dimensión. Es en este sentido que, la aplicación de algunos de los métodos de selección de variables será recomendable antes de cualquier análisis sobre estos datos. En esta técnica una representación de tamaño finito es usada para cada curva, superando así el problema de la alta dimensionalidad.

En este trabajo se comparan tres procedimientos de selección de variables con un método comúnmente usado de reducción de dimensionalidad para datos funcionales. Los resultados se presentarán utilizando dos conjuntos de datos reales. Esto se hace

con el fin de comparar la efectividad para reducir la tasa de error de mala clasificación en los conjuntos de datos. Los resultados obtenidos en esta tesis, muestran que la reducción de la dimensionalidad usando B-Splines tiene un mejor rendimiento que la selección de variables.

A mis abuelos, padres y hermanos por ser mi ejemplo de vida y principal fuente de inspiración para ser mejor cada día.

AGRADECIMIENTOS

A *Dios* por ser el motor de nuestras vidas, por la familia que me dio y porque siempre guiará mi vida en el cumplimiento de mis metas.

A mis padres *Félix Huanca Yanarico* y *Teresa Ochoa Quispitupa* por la familia que formaron, por su apoyo, confianza, amor y dedicación durante todas las etapas de mi vida. Ustedes siempre serán mi mejor ejemplo a seguir.

A mis hermanos *Diana* y *Cristian* por su presencia, amistad y cariño, por darme fuerzas cuando las necesitaba y por la admiración que en mí generan.

A mis abuelitos *Carmen, Mariano y Juana* por todo el esfuerzo que pusieron en la formación de sus hijos y el cariño a sus nietos.

Al *Dr. Edgar Acuña* por estar presto a atender mis consultas durante la realización de este trabajo, por sus aportes, consejos y el apoyo que recibí de su parte.

Al *Departamento de Ciencias Matemáticas* por la oportunidad que me brindó para realizar mis estudios de maestría, a los *profesores* por compartir sus conocimientos y a todo el *personal administrativo* por su atención y amable colaboración.

A *Roberto Trespalacios* no sólo por sus aportes en el mejoramiento de este trabajo, su disposición y apoyo en todo momento, sino también por su compañia, su tiempo y dedicación. Gracias por tu confianza, consideración, cariño y por junto a mi familia, inspirarme a ser mejor y hacer las cosas bien siempre!.

A *John Villavicencio* por su orientación y ayuda para mi llegada a Puerto Rico.

A *Edwin Flórez* por sus aportes en el restablecimiento de este trabajo.

A *mis amigos y compañeros* porque su presencia y el haberlos conocido hizo de esta experiencia más bonita y enriquecedora, en especial a Charlie, Fabián, Ricela, Einstein, Daiver, Elluz, Lucho, Javier Moyano y a todos mis amigos en Puerto Rico.

Índice general

página

Índice de cuadros

Tabla pagina

Índice de figuras

Figura pagina

LISTA DE ABREVIATURAS

ADF	Análisis de Datos Funcionales.
mRMR	Método de Mínima Redundancia y Máxima Relevancia.
MaxRel	Método de Máxima Relevancia.
B-Splines	Splines Básicos.
MID	Diferencia de Información Mutua.
MID1	Diferencia de Información Mutua con threslhold $\alpha = 1$.
MID2	Diferencia de Información Mutua con threslhold $\alpha = 0{,}5$.
MID3	Diferencia de Información Mutua con threslhold $\alpha = 0$.
MIQ	Cociente de Información Mutua.
MIQ1	Cociente de Información Mutua con threslhold $\alpha = 1$.
MIQ2	Cociente de Información Mutua con threslhold $\alpha = 0{,}5$.
MIQ3	Cociente de Información Mutua con threslhold $\alpha = 0$.
CART	Classification and Regression Trees.
k-NN	k- Nearest Neighbors.
Rpart	Recursive Partitioning and Regression Trees.
CV	Cross Validation.
nm	Nanómetros.

Capítulo 1
INTRODUCION

El Análisis de Datos Funcionales (ADF) se refiere al análisis estadístico de muestras de datos que consisten en funciones aleatorias o superficies, en las que cada función representa un elemento de la muestra. Típicamente, las funciones aleatorias contenidas en la muestra se considera que son independientes y suaves. La metodología del ADF es esencialmente no paramétrica, se utilizan métodos de suavizamiento los cuales permiten el modelado flexible de las curvas.

El ADF ha cobrado importancia recientemente debido a que los datos funcionales aparecen en ramas como la metereología, la economía, la demografía, la genética, la química, entre otras. En ciencias del comportamiento, permite estudiar las funciones psicobiológicas tales como las emociones o el número e intensidad de episodios depresivos. Del mismo modo se puede evaluar su importancia en el estudio de la comunicación humana, incluyendo lenguaje verbal, no verbal, gráfico, música y percepción.

En la actualidad el ADF abarca una amplia variedad de métodos estadísticos que se ajustan a las curvas de la muestra. Así mismo, el volumen actual de la estadística computacional dedicada al estudio del ADF agrupa una serie de contribuciones a la investigación, sobretodo en las áreas principales de este tema. Una de ellas consiste en aplicar técnicas de reducción de dimensionalidad a datos funcionales, antes de usarlos para una toma predictiva como es la clasificación supervisada.

1.1. Antecedentes

El ADF es un campo de investigación reciente, cuyo estudio ha sido atractivo y de gran interés para muchos autores, quienes han facilitado las principales herramientas de análisis para trabajar con este tipo de datos. Dos de los principales autores de este campo son: Ramsay y Silverman, 2005 [1], quienes introducen el estudio del ADF, comenzando desde la definición de un dato funcional hasta la aplicación de diferentes técnicas de la estadística clásica a los datos funcionales.

Debido a las características comunes que poseen, y tal como lo afirma M. Valderrama [2], las aproximaciones al ADF se han hecho principalmente mediante la extensión de las técnicas multivariantes de vectores a las curvas; y llevando procesos estocásticos al mundo real. Desde un punto de vista conceptual los datos podrían ser considerados como muestras de trayectorias de un proceso estocástico de tiempo continuo, cuya representación gráfica es un conjunto de curvas definidas en el espacio de los parámetros del proceso. Sin embargo, en el ADF, no hay ninguna hipótesis sobre la distribución de probabilidad del proceso estocástico subyacente a los datos, sólo la información de la muestra [2]. En este sentido, de un tiempo a esta parte, algunos investigadores vienen estudiando este tipo de datos; uno de ellos es Levitin, 2007 [3], que presenta en aspectos claves del ADF utilizando datos reales de un estudio de la emoción musical; por su parte Navarro, 2004 [4], hace una introducción tanto al ADF descriptivo como al análisis de componentes principales funcionales. Del mismo modo, discute parte de la teoría de splines para transformar datos discretos en funciones, implementa algunos algoritmos en lenguaje R y usa todo lo anterior en la aplicación de dos ejemplos; uno relacionado con pirámides de población de 225 países y el segundo proveniente de datos quimiométricos que se administran simultáneamente en dos esquemas de tratamientos distintos.

Por otro lado, en cuanto a la selección de variables aplicado a datos funcionales, hasta el momento no se cuenta con un método que nos permita realizarla de manera óptima. No obstante, cabe mencionar que en Fraiman, 2015 [5], se introduce un procedimiento general para capturar la información relevante de un conjunto de datos funcionales establecidos en relación a un método estadístico utilizado para analizar los datos, como la clasificación, regresión o componentes principales. El objetivo que tienen los autores es identificar un pequeño subconjunto de funciones que pueden "explicar mejor" el modelo, destacando sus características más importantes. Finalmente muestran la aplicación sobre algunos conjuntos de datos, tales como las mediciones de altura de 54 niñas y 39 niños, tomadas en 31 puntos entre 1 a 18 años (ver Ramsay, 2005 [1]); a partir de ellos buscan encontrar los instantes que determinan los patrones de crecimiento en niños y niñas, también trabajan sobre el conjunto de datos phoneme [6] y con los datos de temperatura de Canadá [1].

En lo que respecta a la clasificación como un proceso posterior a la reducción de la dimensionalidad se puede citar al artículo de Fraiman [7], en él se presentan dos procedimientos para la selección de variables en el análisis de conglomerados "cluster" y algunas reglas de clasificación. El primer procedimiento de selección está orientado a detectar las variables no informativas, las cuales luego de ser detectadas serán sustituidas por su media marginal; mientras que el segundo trabaja la multicolinealidad a través de la media condicional. Se describe un algoritmo de "forward-backward" (selección hacia adelante y hacia atrás, respectivamente) para buscar el subconjunto mínimo de variables que explica un porcentaje fijo de la asignación de los datos a los cluster. Se lleva a cabo una simulación y se analizan dos conjuntos de datos reales: uno de ellos referido a la calidad de educación buscando encontrar grupos homogéneos de escuelas, así como la caracterización de los grupos; mientras que el otro se orienta a encontrar patrones de comportamiento de consumidores de una compañia

eléctrica. En ambos casos; se observa que pese al esfuerzo computacional en tiempo requerido, los resultados son positivos.

En el ámbito de la clasificación supervisada aplicado a una muestra de curvas, se puede citar a Ferraty y Vieu, 2003 [6], quienes proponen una herramienta no paramétrica para el estudio de la relación entre una curva (considerada como un predictor funcional) y una respuesta categórica. Su estudio tiene el propósito de desarrollar una discriminación no paramétrica de curvas (NPCD) utilizando un enfoque funcional, es decir, teniendo en cuenta las trayectorias continuas x_i en lugar de los vectores discretizados; para ello se estima la probabilidad posterior de que una curva (de entrada), pertenezca a una clase dada; la estimación se deriva del estimador kernel introducido por Ferraty y Vieu [8] (2002) en el contexto general de la regresión en que la respuesta es real mientras que el regresor, funcional. El rendimiento práctico se muestra por medio de un estudio de simulación sobre los conjuntos de datos Tecator y Phoneme (descritos en el Capítulo 4), y se concluye el buen desempeño de las estimaciones en términos de la tasa de error de mala clasificación del método que proponen en comparación con otros métodos como Análisis Discriminante Penalizado Ridge (PDA/Ridge), Regresión por Mínimos Cuadrados Parciales Multivariados (MPLSR), Árboles de Regresión y Clasificación (CART), entre otros.

Finalmente, Ivanov, 2010 [9] y Torrecilla, 2010 [10] muestran dos métodos diferentes de selección de variables para clasificación supervisada que mejor se adecúan del caso multivariado al funcional. El primero es un método computacional basado en información mutua que utiliza dos métodos no paramétricos para la estimación de la entropía (árboles entrópicos mínimos y vecinos próximos). Mientras que el segundo, muestra la eficiencia del método de Mínima Redundancia y Máxima Relevancia (mRMR) propuesto por C. Ding y H. Peng, 2003 [11] para luego contrastar

los resultados obtenidos con el método de mínimos cuadrados parciales (PLS por sus siglas en inglés). El mRMR es uno de los métodos de selección que serán evaluados en el presente estudio. Con respecto a las definiciones y principales características que poseen los datos funcionales, para el desarrollo del presente trabajo se tuvo como referencia bibliográfica principal a [1], [8] [12], [3], [4].

1.2. Justificación

En muchos estudios de la estadística convencional es habitual que el número de datos u observaciones sea mayor al número de variables; sin embargo en el análisis funcional ocurre lo contrario, pues una de las características más relevantes de estos datos, es que representan las mediciones de procesos de suavizamiento tomadas a través del tiempo, muchas veces incluso medido en intervalos desiguales. Es decir, el número de puntos donde se hacen las mediciones es mayor al número de observaciones (curvas). Es por esta razón que tanto para su estudio, como para su tratamiento se requiere aplicar, previamente a cualquier análisis algunas técnicas de reducción de dimensionalidad sobre el continuo (conjunto de puntos donde se hacen las mediciones).

El análisis de componentes principales funcionales ha sido bastante estudiado; y si bien es cierto, la aplicación de este método es útil cuando existe alto grado de correlación entre los múltiples atributos pues proporciona una representación de dimensión finita para cada curva y con ello reduce el problema de alta dimensionalidad; las componentes resultantes son funciones que representan el conjunto completo de curvas de la muestra; y por ello este método no será usado en el presente estudio, pues dificultaría realizar uno de los fines de nuestro estudio como es la clasificación supervisada.

Por lo anterior, uno de los propósitos que persigue nuestro trabajo es aplicar algunos métodos de selección de variables aplicados del caso multivariado al funcional, los cuales permitan realizar discretizaciones más finas, eliminando menos información sobre la curva y buscando obtener el subconjunto de puntos observados a lo largo de las curvas, que expliquen y representen al conjunto completo; y en los cuales las clases de las muestras sean fácilmente identificables [10].

Del mismo modo se usarán los B-Splines como método de reducción en los conjuntos de datos estudiados. Finalmente, se usará la clasificación supervisada mediante el uso de árboles de decisión y vecinos más cercanos como clasificadores para validar el método que estamos proponiendo, de manera que se minimize el error de mala clasificación.

1.3. Motivación

Además de la escasa literatura en métodos funcionales de selección de variables; el motivo principal que induce aplicar algunas técnicas de reducción de dimensionalidad en datos funcionales como paso previo a la clasificación supervisada, se debe a los buenos resultados que tanto la selección de variables, como el suavizamiento por splines han demostrado tener en estudios en los cuales han sido aplicados al caso multivariado.

1.4. Objetivos

1.4.1. Objetivo General

Comparar el efecto de aplicar cuatro métodos de reducción de dimensionalidad en la estimación de la tasa de error de mala clasificación en datos funcionales.

1.4.2. Objetivos Específicos

- Aplicar y comparar diferentes métodos de selección de variables del caso multivariado al funcional, eligiendo el que produce mejores resultados para clasificación supervisada.
- Obtener suavizadores B-Splines y comparar su efecto en la clasificación supervisada.
- Comparar la tasa de error de mala clasificación del método de reducción de dimensionalidad propuesto, con el encontrado de la aplicación de selección de variables.

1.5. Limitaciones

El estudio y uso del ADF presenta las siguientes limitaciones:

1. El eje "continuo" puede estar sujeto a distorsiones aleatorias, por ello en ocasiones es necesario reflejar parte de la curva con modelos funcionales adecuados.
2. Para cada función en la muestra se tienen mediciones disponibles sobre una base continua que puede variar.
3. Algunas veces los tiempos de medición son dispersos y distribuidos aleatoriamente, sobretodo cuando se trata de estudios longitudinales.
4. El error en las mediciones de los niveles de la trayectoria es también común.

Capítulo 2
ANÁLISIS DE DATOS FUNCIONALES (ADF)

En los últimos años, diversas áreas vienen trabajando con grandes bases de datos, las cuales con mayor frecuencia corresponden a observaciones de una variable aleatoria estudiada a lo largo de un intervalo de tiempo, espacio, o cualquier otra medida continua (o a discretizaciones extensas de ellos). En ambos casos, todas las técnicas incluidas están restringidas al espacio de funciones L_2, el cual es un espacio con características específicas que lo hacen especialmente tratable. El ADF se refiere a los datos que proporcionan información sobre las curvas, superficies o cualquier otra cosa variable en un continuo. El continuo es a menudo el tiempo, pero también puede ser la localización espacial, la longitud de onda, la probabilidad, etc. Por otro lado, los datos pueden ser tan precisos que en ocasiones el error es ignorado, pero a la vez éstos podrían ser producto de errores de medición sustancial o incluso tener una compleja relación indirecta con la curva que definen.

2.1. Conceptos Teóricos

Definition 2.1.1. *Una variable aleatoria $\boldsymbol{\chi}$ se dice que es una variable funcional si toma valores en un espacio infinito dimensional o espacio funcional E (Espacio normado o seminormado completo). Una observación χ de $\boldsymbol{\chi}$ se llama dato funcional, el cual puede ser una curva, superficie, o cualquier otro objeto matemático infinito dimensional.*

Definition 2.1.2. *Un conjunto de datos funcionales $\chi_1, \chi_2, \ldots, \chi_n$ es la observación de n variables funcionales $\boldsymbol{\chi_1}, \boldsymbol{\chi_2}, \ldots, \boldsymbol{\chi_n}$ idénticamente distribuidas.*

La primera dificultad que tendremos siempre al analizar datos funcionales, es encontrar una representación adecuada para ellos y determinar en que espacio funcional vamos a trabajar.

Un dato funcional $\chi_i(t) \in T$, se representa generalmente como un conjunto finito de pares (t_j, y_{ij}), $t_j \in T, j = 1, \cdots, M$ e $y_{ij} = \chi_i(t_j)$ (si no hay ruido blanco) y $y_{ij} = \chi_i(t_j) + \epsilon_j$ (si hay ruido blanco), donde ϵ_j tiene media cero. El conjunto de puntos $\{t_j\}_{j=1}^{M} \subset T$ puede ser considerado el mismo para todas las funciones en un conjunto de datos funcionales.

De otro lado, el espacio comúnmente usado cuando se habla de datos funcionales es el espacio $L_2[S]$; esto es, las funciones de cuadrado integrable en el intervalo $S = [a, b] \subset R$. Luego:

Sea $S = [a, b] \subseteq \mathbb{R}$. Normalmente se asume que se tienen elementos de:

$L_2(S) = \{f : S \to \mathbb{R}, \text{ tal que } \int f(t)^2 dt < \infty\}$

$L_2[S]$ con el producto interno usual $< f, g >= \int_S f(t)g(t)dt$ es un espacio Euclidiano.

Desde un punto de vista más general podemos tener datos funcionales en la familia: $L_p[S, \mu] = \{f : S \to \mathbb{R}, \text{ tal que } \int |f(t)|^p d\mu < \infty\}$ donde (S, μ) es un espacio de medida y $1 < p < \infty$. Estos espacios son semi-normados, salvo el caso $p = 2$ que es el único en esta familia que es un espacio de Hilbert separable [12].

En general, la representación de un dato funcional en una base ortonormal proporcionará ventajas tanto desde el punto de vista teórico como práctico, sirviendo de conexión entre la inevitable discretización del dato funcional y su verdadera forma funcional [12].

Básicamente, la idea clave cuando se pueden usar bases ortonormales, es representar cada dato funcional en la base usando aquellas coordenadas que son más significativas [12]. Debido a la alta dimensión de los datos funcionales, se elige en general un número K para representar los datos en el subespacio, convirtiendo el

problema de dimensión infinita en un problema multidimensional. El parámetro K es, en cierto modo, un parámetro de suavización de los datos funcionales. Si K es pequeño, tendremos un modelo muy manejable, pero posiblemente habremos perdido información relevante; por otro lado con un valor de K alto, representaremos muy bien los datos, pero el problema de la dimensión será mayor también. La elección y decisión sobre la base a usar, depende del objetivo y sobretodo del tipo de datos que se estén estudiando; es así que, para datos periódicos se suele emplear bases de Fourier, mientras que para datos no periódicos se usan bases B-Splines o Wavelets [12].

2.2. ADF frente a otras técnicas estadísticas multivariantes

Debido a ciertas características del ADF, es posible compararlo con la estadística clásica multivariante. Es en este sentido, que una muestra de puntos discretos se toma de una población; y ella puede contener muchas fuentes de variación, tales como: variabilidad inicial a través de las unidades objeto de medición, variabilidad debido al error de medición, variabilidad por muestreo, debido a la incapacidad de reproducir las mismas condiciones de tratamiento de una unidad a otra, y la variabilidad sistemática que es el objetivo del análisis. Este ejemplo funcional contiene las mismas fuentes de variación como lo hace una muestra discreta; sin embargo, existe el problema de la cuantificación de la variabilidad dentro y entre las curvas [3]. La relación entre el ADF y los modelos para medidas repetidas o datos longitudinales es aún más estrecha. Sin embargo el ADF permite la estimación de curvas más complejas en ventaja a esos dos métodos.

2.3. Recolección de datos

Se pueden hacer repeticiones múltiples de este proceso, obteniendo muestras ya sea del mismo proceso (mejorando nuestras estimaciones de la forma real de la curva que representa el proceso subyacente), o el suministro de datos a varios participantes en la misma variable dependiente s. Anteriormente, hemos señalado que

ADF está destinado a ser utilizado en los datos que provienen de un proceso suave y continuo. Sin embargo, prácticamente toda la recopilación de los datos que conocemos no producen observaciones continuas, sino que son tomadas como muestras en puntos discretos en el tiempo; y cuando esto ocurre se supone que existan suficientes observaciones para modelar el proceso, pues las grandes ventajas del ADF sobre los métodos tradicionales de análisis ocurren en conjuntos de datos de gran tamaño.

La replicación de orden i del proceso subyacente, se escribe como $x_i(t)$; y los valores de datos correspondientes a los puntos de tiempo t_{ij} se escriben como y_{ij} con $t \in \mathbb{R}$. A menudo se realizan las observaciones en los mismos argumentos t_j; por tanto para cada repetición en este trabajo, se considerará por simplicidad notacional que esto se cumple, es decir; que $t_{ij} = t_j$ para cada i. Cabe aclarar que en muchas aplicaciones reales los lugares e incluso los números de puntos de tiempo pueden variar de replicación en replicación; esto podría ocurrir por ejemplo, en un experimento en el que una variable dependiente se tenía que observar en puntos precisos en el tiempo, pero una o más mediciones se tomaron algo fuera de horario (unos minutos antes o después debido a factores externos). Sin embargo, dicha variabilidad puede ser manejada con algunas técnicas descritas a continuación (véase también Ramsay y Silverman [1] 2005). Aunque por lo general existe error de medición en las observaciones y_j, vamos a suponer que las t_j se miden sin error. De esta manera podemos relacionar las observaciones discretas con el proceso liso subyacente a través de la ecuación $y_{ij} = x_i t_j + \epsilon_{ij}$, donde cada ϵ_{ij} es el término de error, que se supone será distribuido con media $\mu = 0$ y varianza σ^2 finita.

2.4. Tipos de variabilidad entre las curvas

En el ADF es común hablar de dos tipos de variabilidad entre las curvas (y superficies): la primera de ellas está asociada a la *variación de la amplitud*, que se ocupa de las diferencias en altura entre las curvas; y la segunda se refiere a la *variación de fase*, que se ocupa de las diferencias en el tiempo (momentos). Es decir,

se refiere a la ubicación en la base continua de características más destacadas en las curvas observadas. Esto ha generado algunos inconvenientes en la adaptación y uso de estadísticas descriptivas tales como: medias, varianzas y correlaciones en el ámbito funcional y del mismo modo en la aplicación de técnicas como análisis de componentes principales o análisis de regresión, pues todas estas herramientas están diseñadas para describir sólo variación en la amplitud. La posibilidad de variación de fase es una característica propia de datos funcionales, por tanto; en el registro de curvatura temporal, la idea es ampliar por contracción o por expansión a nivel local, la hora del reloj; es decir ajustar los datos a un tiempo biológico estandarizado.

En algunas situaciones simples, esto puede ser cuestión de una transformación lineal simple de las temporizaciones para algunos individuos. Sin embargo, en un proceso más complejo de la función de curvatura temporal se requiere que se extienda a lo largo del tiempo algunos intervalos y se comprima sobre los demás; cuando esto ocurre, llamamos a nuestro tiempo *función de deformación* $h(t)$, de tal manera cuando un proceso se ejecuta más rápido que el tiempo estandarizado, se tiene que $h(t) > t$ y si se está ejecutando más lentamente, entonces $h(t) < t$. Luego, si nuestra curva original es $x(t)$, entonces la curva registrada será $x[h(t)]$. Normalmente se espera que la función de curvatura temporal quede intacta en los comienzos y finales de las curvas, es decir: $h(t_0) = t_0$ y $h(t_{end}) = t_{end}$. De otro lado, se impone un supuesto de monotonicidad en que se requiere que los eventos en las curvas registradas se produzcan en el mismo orden que el de las curvas no registradas; en términos matemáticos, se requeriría entonces que: $h(t_2) > h(t_1)$, si $t_2 > t_1$ lo que implica que la función de curva temporal es monotónicamente creciente. Por último, tenemos el objetivo de que nuestras curvas registradas poseen todas la misma forma que las curvas originales; es decir, sólo las amplitudes son diferentes pero los picos y valles ocurren al mismo tiempo; esto significaría que las curvas registradas en el

conjunto de datos son proporcionales entre sí; y matemáticamente se dice que, si las curvas de x_1 y x_2 son proporcionales, entonces $x_1(h(t)) = ax_2(t)$ para una constante a positiva. De todo esto, concluimos que obtener la variación de amplitud en ADF es de interés primordial, ya que ésta describe cómo las alturas o intensidades de las curvas varían entre los participantes (o entre cualesquiera grupos que puedan existir), y en este caso puede ser crítico que la variación de fase se elimine primero.

2.5. Construcción de curvas suaves a partir de datos discretos

Para hacer un ADF, el primer paso es convertir los datos originales en objetos funcionales. Para ello, se ajusta una curva a las observaciones discretas para aproximarlas al proceso continuo y luego se trabajan con estos nuevos objetos para su análisis posterior. Un procedimiento común en la estadística, matemáticas y la ingeniería para la representación de datos discretos como una función suave, es el uso de un expansión de base, tal como:

$$x_i(t) = c_{i1}\phi_1(t) + c_{i2}\phi_2(t) + \cdots + c_{ik}\phi_k(t) + \cdots + c_{iK}\phi_K(t), \tag{2.1}$$

Donde: K indica el número de funciones de la base y $f_k(t)$ es el valor de la k-ésima función base con valor t en el argumento. La funciones base $f_k(t)$ son un sistema de funciones elegidas especialmente para usarlas en la construcción de bloques para representar una curva suave.

2.6. Bases para datos funcionales

Una base es un conjunto de funciones conocidas e independientes $\{\phi_k\}_{k\in\mathbb{N}}$ tales que cualquier función puede ser aproximada, tan bien como se quiera, mediante una combinación lineal de K de ellas (con K grande). De esta forma, la observación funcional puede aproximarse como: $\boldsymbol{\chi(t)} = \sum_{k\in\mathbb{N}} c_k\phi_k(t) \approx \sum_{k=1}^{K} c_k\phi_k(t)$ [12]. Si los elementos de la base son fácilmente diferenciables hasta orden q, tenemos: $\boldsymbol{\chi^{(q)}(t)} = \sum_{k\in\mathbb{N}} c_k\phi_k^{(q)}(t) \approx \sum_{k=1}^{K} c_k\phi_k^{(q)}(t)$.

Hay diferentes tipos de sistemas de funciones de base, de las cuales las más conocidas son: las bases de Fourier, recomendadas si los datos son periódicos; las bases B-Splines que permiten hacer cálculos rápidos y flexibles; y las bases de Wavelets apropiadas para modelizar discontinuidades: exponencial, potencial, polinomial, etc. Los coeficientes c_{ik} determinan los pesos relativos de cada función de base en la construcción de la curva χ_i. La estimación de una curva, que se expresa como una expansión de función de base, se reduce básicamente al problema de estimación de parámetros multivariados de estos coeficientes. Cuando las curvas no son periódicas y además son complejas, de tal modo que un polinomio de bajo orden no puede capturar sus rasgos, se usan las funciones B-Splines. Los pesos de los coeficientes de base se eligen de tal forma que la curva se ajuste de manera óptima a los datos, para un cierto grado de suavizado. En este sentido, un gran número de funciones de base dará lugar a una curva que es más fiel a los datos observados (sesgo bajo), pero a menudo es menos suave (alta varianza). Contrariamente, el uso de un pequeño número de funciones de base, producirá una curva que da menos importancia en la interpolación de los puntos (sesgo alto) pero más importancia a la suavidad (baja varianza).

2.6.1. Bases de Fourier

Una base de Fourier es una base periódica de período $\frac{2\pi}{\omega}$ que cuando se seleccionan datos $\{t_j\}$ equiespaciados en $T = [0, T]$ y $\omega = \frac{2\pi}{T}$ está formada por las siguientes funciones ortonormales:

$$\phi_0(t) = \frac{1}{\sqrt{T}} \tag{2.2}$$

$$\phi_{2r-1}(t) = \frac{sen(r\omega t)}{\sqrt{\frac{T}{2}}} \tag{2.3}$$

$$\phi_{2r}(t) = \frac{cos(r\omega t)}{\sqrt{\frac{T}{2}}} \tag{2.4}$$

2.6.2. Bases de B-Splines

Un spline es un conjunto de polinomios (de orden m) definidos en subintervalos construidos de tal modo que el final del polinomio en un subintervalo coincida con el inicio del polinomio en el siguiente intervalo (hasta la derivada $m-2$). El conjunto de los puntos de corte de los subintervalos (límites de intervalos) son denominados "nodos" y se denota por $\tau = \{\eta\}_{l=0}^{L}$.

Los B-Splines (Bases de Splines) son polinomios unidos de extremo a extremo en los nodos, se calculan fácilmente con el algoritmo de Boor; entre ellos los más utilizados son los cúbicos. En general, esta estructura permite diferentes cantidades de suavidad o aspereza en varios lugares de la curva; el número de parámetros para definir una función spline es igual al número de nodos interiores $(L-1)$ más el orden del polinomio (m). Se representan de la siguiente manera:

$$S(t) = \sum_{k=1}^{m+L-1} c_k B_k(t, \tau) \tag{2.5}$$

Para la elección de los nodos se considera que estén igualmente espaciados; también se puede seleccionar los puntos del momento exacto en que los datos fueron observados o los puntos de tiempo de interés predefinido. El orden de los B-Splines es igual al grado de los polinomios a partir de los cuales se construyen, más uno. Por ejemplo, un B-Spline de orden 3 comprende curvas cuadráticas unidas en los nodos, y un B-Spline de orden 2 es construida a partir de líneas a trozos (es decir, segmentos de línea). En muchos conjuntos de datos, se ha encontrado que los B-Splines de orden 4 ajustan curvas suaves y por lo tanto, son muy usados en los paquetes de suavizado estándar. Los B-Splines se normalizan de modo que para cualquier valor fijo de t, la suma de todas las funciones de base B-Splines en ese punto sea uno. Esta normalización significa que un coeficiente c_{ik} es aproximadamente el valor de la curva que se forma en el lugar en el que la k-ésima función de base B-Splines alcanza su máximo.

2.6.3. Bases Wavelets

La base de Wavelets se construye a partir de dos funciones. El wavelet padre ϕ que verifica que $\int \phi(t)dt = 1$ y el wavelet madre ψ que verifica que $\int \psi(t)dt = 0$. Los elementos de la base se obtienen a partir de estas dos funciones ortogonales por traslación y cambio de escala.

$$\phi_{j,k}(t) = 2^{-\frac{j}{2}}\phi(2^{-j}t - k), \tag{2.6}$$

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{-j}t - k), \tag{2.7}$$

$$\int \phi_{j,k}(t)\phi_{j,k'}(t)dt = \delta_{k,k'}, \tag{2.8}$$

$$\int \psi_{j,k}(t)\phi_{j',k'}(t)dt = 0, \tag{2.9}$$

$$\int \psi_{j,k}(t)\psi_{j',k'}(t)dt = \delta_{j,j'}\delta_{k,k'}, \tag{2.10}$$

Elegida la base, la aproximación ortogonal wavelet de una función $f(t)$ viene dada por:

$$f(t) \approx \sum_k S_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + \cdots + \sum_k d_{1,k}\psi_{1,k}(t)$$

Llamando:

$$S_J(t) = \sum_k S_{J,k}\phi_{J,k}(t)$$

$$D_J(t) = \sum_k d_{J,k}\psi_{J,k}(t)$$

$$D_{J-1}(t) = \sum_k d_{J-1,k}\psi_{J-1,k}(t)$$

$$\vdots$$

$$D_1(t) = \sum_k d_{1,k}\psi_{1,k}(t)$$

A la función $S_J(t)$ se la conoce como señal suave y a las funciones $D_J(t)$ como las funciones detalle. A esta descomposición se la llama descomposición multiresolución.

Capítulo 3
REDUCCIÓN DE LA DIMENSIONALIDAD Y CLASIFICACIÓN

3.1. Selección de Variables

La selección de variables es una de las tareas principales de reducción de dimensionalidad, cuyo estudio y aplicación se encuentran en continuo crecimiento en los últimos años; esto debido a la irrupción de datos extremadamente grandes en diferentes áreas. La importancia del uso de estos métodos radica en que, en lugar de utilizar todas las variables disponibles (características o atributos) en los datos, se elige selectivamente un subconjunto de las mismas, las cuales son utilizadas en el sistema discriminante. En este proceso no se altera la representación original de las variables, además que el conjunto resultante suele ser más informativo. Existe un número de ventajas que inducen al uso de selección de variables, y de ellas resaltan las siguientes:

a. Reducción de la dimensión para disminuir del costo computacional.
b. Mejorar la exactitud de la clasificación.
c. Encontrar las características más interpretables o características que pueden ayudar a identificar y controlar los tipos de funciones.

Por otro lado; existen dos enfoques generales para realizar la selección de variables, éstos son: los métodos de filtro y métodos wrapper; a continuación se describen las principales características de ambos.

3.1.1. Métodos de Filtro

En estos métodos, la selección de variables se puede realizar de dos formas, la primera es considerando las características propias que determinan su relevancia, o en base a los poderes discriminantes que poseen con respecto a las clases; son fácilmente calculables y muy eficientes, además la selección de variables no está correlacionada con la de los métodos de clasificación, y por lo tanto tienen una mejor propiedad de generalización. Sin embargo, debemos resaltar que una deficiencia que presentan los métodos de filtro, es que las variables seleccionadas podrían estar correlacionadas entre sí, y esto generaría un problema de redundancia en el conjunto. El problema fundamental de la existencia de redundancia entre las variables, es que, el conjunto de variables no es una representación completa, resumida e independiente de las mismas; mientras que si se eliminan variables altamente correlacionadas se puede reducir el número de variables seleccionadas sin alterar en nada la eficiencia del conjunto restante, en el rendimiento de la predicción.

Ahora describiremos los métodos de filtro usados en esta tesis.

Relief

El algoritmo de selección de variables *ReliefF*, fue presentado por Kira y Rendell (1992) [13] inicialmente para un problema de dos clases. Este algoritmo se basa en el hallazgo de las variables que distinguen más entre las dos clases, para ello se toma una muestra aleatoria del conjunto original y los valores de los atributos de las instancias incluidas en esta muestra, son usadas para calcular el peso de cada atributo. Estos pesos se calculan de forma iterativa y las mejores variables son las de mayor peso. En 1994, este algoritmo fue ampliado por Kokonenko [14], quien extendió el algoritmo ReliefF a un problema multi-clase y también consideró el cálculo de los pesos cuando se presentan valores perdidos.

A continuación se muestra el pseudocódigo del Algoritmo 1 de Relief para el proceso de selección de variables.

Pseudocódigo Algoritmo 1 : ReliefF

Input: Un conjunto de datos de N instancias, consistente en M atributos de predicción y un atributo de clase.
m: Tamaño de la muestra aleatoria que se extrae de la base de datos.
Output: Un vector W de pesos para cada uno de los atributos p.

1: Inicializar en 0 el vector de peso del atributo $W[A_j] = 0$, $j = 1, 2, \ldots, M$
2: **for** $i \leftarrow 1$ to m **do**
3: Elegir al azar una instancia R_i
4: Encuentra la **NearHit** H_i y la **NearMiss** M_i de R_i
5: **end for**
6: **for** $j \leftarrow 1$ to M **do**
7: $W[A_j] = W[A_j] - diff(A_j, H_i, R_i)/m + diff(A_j, M_i, R_i)/m$
8: **end for**

Decisión: Si $W_j \geqslant \tau$ (un valor preestablecido) entonces se selecciona la variable f_j.

NearHit representa la instancia más cercana a la variable y que pertenece a su misma clase. Mientras que **NearMiss** la instancia más cercana que pertenece a una clase diferente.

Por otro lado, sea A un atributo y sean I_1 y I_2 dos instancias del conjunto de datos, entonces la función $diff(A, I_1, I_2) = |I_1 - I_2|/rango(A)$ donde: $rango(A) = max(A) - min(A)$. Si el atributo A es categórico, $diff(A, I_1, I_2) = 0$ si los valores de I_1 y I_2 son iguales y $diff(A, I_1, I_2) = 1$ en otros casos.

Método de mínima Redundancia y Máxima Relevancia (mRMR)

Es un método de reducción de dimensionalidad, que permite ampliar el poder de representación del conjunto de variables, al exigir que éstas sean más disímiles entre sí. Este método fue desarrollado por Chris Ding y Hanchuan Peng (2003) y desde ese momento ha sido usado en diferentes estudios (tal como [10]).

Otros estudios donde se trabaja la selección de variables basados en el método mRMR son [15], [11]. En el primero de ellos se demuestra la relación de cuatro esquemas de selección: máxima dependencia, mRMR, la máxima relevancia y la redundancia mínima; del mismo modo se muestra la combinación de selección de "mRMR + wrapper" para variables continuas e híbridas. El segundo estudio [11] presenta diferentes definiciones de los términos: relevancia y redundancia, asi como un primer conjunto de resultados del método mRMR. Por otro lado en el 2005, Ding y Peng demuestran la importancia de reducir la redundancia en la función de selección, ver [16], además ese mismo año estos autores presentaron un conjunto completo de los resultados experimentales de mRMR para la selección de genes de microarrays en diferentes condiciones [17], siendo éste último, una versión extendida de [11].

De acuerdo al tipo de variables, existen dos variantes del método mRMR. los cuales son descritos a continuación:

1. Criterio del método mRMR para Variables Categóricas y Discretas

   En [11] tanto para las variables discretas, como categóricas, se usa la información mutua como una medida de relevancia de las variables. La información mutua de dos variables x e y es definida en función de su distribución probabilística conjunta $p(x, y)$ y las respectivas probabilidades marginales $p(x)$ y $p(y)$, de la siguiente manera:

   $$I(x, y) = \sum_{i,j} p(x_i, y_j) log \frac{p(x_i, y_j)}{p(x_i), p(y_j)} \tag{3.1}$$

   En las variables categóricas la información mutua nos permite medir el nivel de similitud entre las variables.

Por otro lado, la idea de redundancia mínima nos permite seleccionar las variables que sean mutuamente diferentes entre sí y por tanto al mismo tiempo una mejor representación del conjunto original de datos.

Sea S el subconjunto de variables que estamos buscando; la condición de mínima redundancia es:

$$minW_I, \qquad W_I = \frac{1}{|S|^2} \sum_{i,j \in S} I(i,j), \tag{3.2}$$

donde: $I(x,y)$ se usa para representar $I(g_i, g_j)$ y $|S|$ es el número de variables en S.

Para medir el nivel del poder de discriminación de las variables cuando ellas son diferencialmente expresadas para todas las clases objetivo, volvemos a utilizar la información mutua $I(h, g_i)$ entre las clases específicas $h = h_1, h_2, \cdots, h_k$ (llamamos h a la variable de clasificación y g_i a la i-ésima variable). $I(h, g_i)$ cuantifica la importancia de g_i para la prueba de clasificación. Así, la condición de máxima relevancia es el de maximizar la relevancia total de todas las variables en S:

$$maxV_I, \qquad V_I = \frac{1}{|S|} \sum_{i \in S} I(h,i), \tag{3.3}$$

donde: $I(h,i)$ denota $I(h, g_i)$.

El conjunto de variables mMRM se obtiene mediante la optimización de las condiciones en las ecuaciones 3.2 y 3.3 simultáneamente. Para la optimización de ambas condiciones, se requiere la combinación de ellas en función de un criterio único. En [11] se trata las dos condiciones igualmente importantes, y se considera los siguientes dos criterios simples de combinación:

$$max(V_I - W_I) \tag{3.4}$$

$$max(\frac{V_I}{W_I}) \tag{3.5}$$

Una vez seleccionada una variable, existen diferentes esquemas para buscar la siguiente, a partir de las condiciones de optimización del mRMR [11], resumidas en la Tabla 3–1.

Tipo	Siglas	Nombre Completo	Fórmula
Discreta	MID	Diferencia de Información Mutua	$\max_{i\in\Omega_S}[I(i,h) - \frac{1}{\lvert S\rvert}\sum_{j\in S} I(i,j)]$
	MIQ	Cociente de Información Mutua	$\max_{i\in\Omega_S}\{I(i,h)/[\frac{1}{\lvert S\rvert}\sum_{j\in S} I(i,j)]\}$
Continua	FCD	Prueba F de la diferencia de correlación	$\max_{i\in\Omega_S}[F(i,h) - \frac{1}{\lvert S\rvert}\sum_{j\in S} \lvert c(i,j)\rvert]$
	FCQ	Prueba F del cociente de correlación	$\max_{i\in\Omega_S}\{F(i,h)/[\frac{1}{\lvert S\rvert}\sum_{j\in S} \lvert c(i,j)\rvert]\}$
	FDM	Prueba F de la distancia multiplicativa	$\max_{i\in\Omega_S}[F(i,h)\cdot\frac{1}{\lvert S\rvert}\sum_{j\in S} d(i,j)]$
	FSQ	Prueba F del cociente de similaridad	$\max_{i\in\Omega_S}\{F(i,h)/[\frac{1}{\lvert S\rvert}\sum_{j\in S}\frac{1}{d(i,j)}]\}$

Cuadro 3–1: Diferentes esquemas de búsqueda de la siguiente variable con las condiciones de optimización del mRMR

En el algoritmo usado por Ding and Peng [11], se selecciona la primera variable de acuerdo con la Ecuación 3.3; es decir, la variable con la más alta $I(h,i)$. El resto de las variables se seleccionan de forma incremental entre las restantes. Supongamos que m variables ya fueron seleccionadas para el conjunto S, entonces seleccionaremos funciones adicionales del conjunto $\Omega_S = \Omega - S$. Se busca optimizar las siguientes dos condiciones:

$$\max_{i\in\Omega_S} I(h,i) \tag{3.6}$$

$$\min_{i\in\Omega_S} \frac{1}{S}\sum_{j\in S} I(i,j) \tag{3.7}$$

2. Criterio del método mRMR para Variables Continuas

   Para datos de variables continuas, podemos elegir el estadístico F como la puntuación de máxima relevancia entre las variables y la variable de clasificación h. Luego, según [11], el valor de la prueba F de la variable g_i en la clase K (denotado por h), tiene la siguiente forma:

$$F(g_i, h) = [\sum_k n_k(\overline{g_i} - \overline{g})/(K-1)]/\sigma^2 \tag{3.8}$$

   donde:

   - $\overline{g}$: es el valor de la media de los g_i en las muestras.
   - $\overline{g_k}$: es el valor de la media de los g_i sin la k-ésima clase.
   - $\sigma^2 = [\sum_k (n_k - 1)\sigma_k^2]/(n-K)$ es la varianza agrupada (donde n_k y σ_k son el tamaño y la varianza de las k clases).

   La prueba F, se reduce a la prueba t cuando se trabaja con 2 clases de clasificación, mediante la relación $F = t^2$. Por lo tanto, para el conjunto de variables S, la máxima relevancia puede ser escrita como:

$$\text{máx}\, V_F, \qquad V_F = \frac{1}{|S|}\sum_{i \in S} F(i, h). \tag{3.9}$$

   La condición de mínima redundancia puede ser especificada de varias maneras. En el caso de que utilizemos el coeficiente de correlación de Pearson, $c(g_i, g_j) = c(i, j)$, la condición es:

$$\text{mín}\, W_c, \qquad W_c = \frac{1}{|S|^2}\sum_{i,j} |c(i, j)|, \tag{3.10}$$

   donde: Se considera como redundancia a la alta correlación positiva y a la negativa, de manera que se toma en cuenta, el valor absoluto de las mismas.

   En [11], se usa el mismo algoritmo de búsqueda incremental lineal, que para una variable discreta.

3.1.2. Métodos "Wrapper"

La selección de variables está relacionada con el método de clasificación usado. Por tanto la utilidad de una variable está directamente juzgada por la precisión estimada del método de clasificación. Desde el punto de vista computacional, estos métodos son mucho más lentos que los de Filtro. Los métodos "Wrapper" no han sido considerados en este trabajo de tesis.

3.2. Splines

Dado un conjunto de puntos observados a través de un continuo, podemos ajustar una curva que pase por la totalidad o la mayoría de ellos; de modo tal que, entre cada par de puntos se establece una sección de esta curva. La curva obtenida, se puede expresar como una función polinómica, cuya primera y segunda derivada son continuas en todas las secciones de la curva.

Las funciones spline se forman mediante la unión de polinomios en puntos fijos llamados "nodos", los cuales dividen al intervalo en subintervalos de igual o diferente longitud; es decir el spline es en cada intervalo un polinomio de grado específico.

A manera de ejemplo, si consideramos el caso más sencillo, en el que se tiene un punto de ruptura que divide al intervalo $[t_L, t_U]$ en 2 subintervalos y en cada uno de ellos se forma una recta, entonces el primer polinomio tendrá 2 grados de libertad (pendiente e intercepto); mientras que el segundo tendrá sólo un grado de libertad (correspondiente a la pendiente, pues coincidirá en el mismo intercepto); por tanto la línea poligonal total tendrá 3 grados de libertad.

En general, si m es el orden del polinomio que se forma en cada uno de los $L-1$ subintervalos delimitados por los "L" nodos, entonces el polinomio será de grado $m-1$ sobre cada uno; además, a diferencia del primer segmento que tendrá "m" grados de libertad (uno por parámetro), los siguientes, tendrán sólo 1 grado de libertad y la función spline tendrá por tanto un total de $m+L$ grados de libertad.

3.2.1. B-Splines

Dado m valores reales t_i, llamados nodos, con $t_0 \leqslant t_1 \leqslant \cdots \leqslant t_{m-1}$, una B-Spline de grado n es una curva paramétrica $\mathbf{S} : [t_0, t_{m-1}] \to \mathbb{R}^2$, compuesta por una combinación lineal de B-Splines básicas $b_{i,n}$ de grado n,

$$\mathbf{S}(t) = \sum_{i=0}^{m-n-2} \mathbf{P}_i b_{i,n}(t) \ , \ t \in [t_{n-1}, t_{m-n}] \tag{3.11}$$

Los $\mathbf{P}_i$ se llaman puntos de control o puntos de *Boor*. Hay $m - (n+1)$ puntos de control que forman una envoltura convexa. Las $m - (n+1)$ B-Splines básicas de grado n se pueden definir mediante la siguiente fórmula de recursión llamada *Cox-de Boor*.

$$b_{j,0}(t) := \begin{cases} 1 & \text{si} \quad t_j \leq t < t_{j+1} \\ 0 & \text{De otra forma} \end{cases} \tag{3.12}$$

$$b_{j,n}(t) := \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t). \tag{3.13}$$

Cuando los nodos son equidistantes, la B-Spline se dice que es uniforme, de otro modo sería no uniforme. Si dos nodos t_j son idénticos, cualquiera de las posibles formas indeterminadas $0/0$ se consideran 0. Nótese que $j + n + 1$ no puede exceder de $m - 1$, lo que limita tanto a j como a n.

Una formulación B-Spline para un solo segmento puede ser escrita como:

$$\mathbf{S}_i(t) = \sum_{k=0}^{3} \mathbf{P}_{i-3+k} b_{i-3+k,3}(t) \ ; \ t \in [0, 1], \tag{3.14}$$

Donde:

$\mathbf{S}_i$ es el i-ésimo segmento B-Spline.

$\mathbf{P}$ es el conjunto de puntos de control.

k es el índice del punto de control local.

Un conjunto de puntos de control sería $\mathbf{P}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$ donde w_i es el peso que tira de la curva hacia el punto de control $\mathbf{P}_i$ mientras que aumenta o se desplazan fuera de la curva, a la vez que disminuye. Toda una serie de segmentos, las $m-2$ curvas $(S_3, S_4, ..., S_m)$ definidas por $m+1$ puntos de control $(P_0, P_1, ..., P_m, m \geqslant 3)$ como un B-Spline en t, se definiría como:

$$\mathbf{S}(t) = \sum_{i=0}^{m-1} \mathbf{P}_i b_{i,3}(t), \tag{3.15}$$

para i que es el número de puntos de control y t es un parámetro global dados los valores de los nodos. Esta formulación expresa una curva B-Spline como una combinación lineal de funciones B-Splines básicas, de ahí el nombre.

Hay dos tipos de B-Splines - uniforme y no uniforme. Una B-Spline no uniforme es una curva donde los intervalos entre los puntos sucesivos de control no son, o no necesariamente son, iguales (el vector de nodos de espacios de nodo interiores no son iguales). Una forma común es donde los intervalos se reducen sucesivamente a cero, interpolando los puntos de control.

Como un ejemplo de un caso particular, podemos usar B-Splines cúbicos uniformes, en vista de que es la forma más usual de B-Splines. La función base puede ser fácilmente calculada, y en este caso, es igual para cada segmento. Puesto en forma de matriz, sería:

$$\mathbf{S}_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{i-1} \\ \mathbf{P}_i \\ \mathbf{P}_{i+1} \\ \mathbf{P}_{i+2} \end{bmatrix} \text{ para t } \in [0,1]. \tag{3.16}$$

El Algoritmo de *De Boor*, es usado para encontrar las bases de los splines con los cuales se construirán las curvas, para luego clasificarlas. A continuación describimos el algoritmo.

Algoritmo de *De Boor*

Con este algoritmo, dado los puntos $u_0, \ldots, u_{p-1}$ y $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_{p-1}$, se encuentra el valor de la curva B-Spline $\mathbf{S}(x)$ en un punto x. Utiliza $O(n^2) + O(n+p)$ operaciones, donde n es el grado y p el número de puntos de control de $\mathbf{S}$.

Pseudocódigo Algoritmo 2 : De Boor

Input: $\bar{u}_k$, $\bar{t} \in [\bar{u}_k, \bar{u}_{k+1})$
Output: $S(u_k)$
1: $I = max\{k|\bar{u}_k \leqslant \bar{t} < \bar{u}_{k+1}\}$;
2: **if** $\bar{t}_I = \bar{u}_I$ **then**
3: $r :=$ multiplicidad$(\bar{u}_I)$;
4: **else**
5: $r = 0$
6: **end if**
7: **for** i:=I-m+1 to I+1-r **do**
8: $P_{i,0} := P_{i-1}$;
9: **end for**
10: **for** j:=1 to m-r **do**
11: **for** i:=I-m+1 to I+1-r **do**
12: $P_{i,j} := \left(\frac{u_{m+i-j}-t}{u_{m+i-j}-u_{i-1}}\right) P_{i-1,j-1} + \left(\frac{t-u_{i-1}}{u_{m+i-j}-u_{i-1}}\right) P_{i,j-1}$;
13: **end for**
14: **end for**
15: $S(\bar{t}) := P_{I+1-r,m-r}$ Punto S sobre la curva

3.3. Análisis de Componentes Principales Funcionales

El análisis de componentes principales funcional, puede ser visto como un método para construir una base óptima ortogonal de dimensión fija, y esto se refieren a menudo a funciones empíricas básicas. Los objetivos del análisis de componentes principales funcionales (ACPF) se basan en los del análisis de componentes principales clásico (ACP), por tanto la idea es extraer y caracterizar los principales tipos de variación en los datos, es decir, entre las curvas observadas.

En este sentido, el primer ACP proporciona información sobre la forma más importante en que un conjunto de curvas varía. El segundo, ortogonal al primero en un sentido apropiado, da el segundo modo más importante de variación y así sucesivamente. A menudo, como en el caso multivariado, la variación total es principalmente explicada por dos o tres componentes principales funcionales.

Para los datos funcionales, por otra parte, los componentes principales están definidos por funciones de ponderación PC diferentes sobre el mismo intervalo de t como los datos funcionales. Estos componentes también pueden ser rotados para mejorar la interpretación. Sin embargo a diferencia del análisis multivariante, no existe el requisito de mantener la ortogonalidad en la rotación.

El único cambio crítico en el movimiento de multivariante para PC's funcionales es que las componentes son funciones (curvas) en lugar de vectores, sin embargo; la interpretación es muy similar: los componentes principales funcionales resultantes destacan las direcciones en las que el conjunto de datos más varía.

PCA es un método en el que un análisis multivariado de versiones discretizadas de los datos de la curva en realidad le da esencialmente los resultados funcionales; el suavizamiento puede ser añadido pero es importante sólo cuando las curvas son muy ásperas. Según el texto de Ramsay y Silverman [1] el ACP puede definirse iterativamente de la siguiente forma:

1. Se encuentra el vector de pesos $\epsilon = (\epsilon_{11}, \cdots, \epsilon_{p1})^t$ para el cual los valores *"scores"* del componente principal $f_{i1} = \sum_j \epsilon_{j1} x_{ij}$ maximizan $\sum_i f_{i1}^2$ sujeto a $\sum_j \epsilon_{ji}^2 = |\boldsymbol{\epsilon_1}|^2 = \mathbf{1}$.
2. Se lleva a cabo una segunda y hasta p subsecuentes etapas. En la $m-$ésima etapa se calcula un nuevo vector de pesos ϵ_m, con componentes ϵ_{jm}, tal que $\sum_j f_{jm}^2$ es máxima, sujeto a las restricciones $|\epsilon_m|^2 = 1$ y $\sum_j \epsilon_{jk}\epsilon_{jm} = \epsilon_k^t \epsilon_m = 0, k < m$.

3. Los componentes se obtienen resolviendo la ecuación propia $V\epsilon = \rho\epsilon$, donde V es una matriz de covarianzas o de correlación, $\boldsymbol{\epsilon}$ es un vector propio de V y ρ es un valor propio de V.

En el caso del ACPF, los valores de las variables se reemplazan por los valores de las funciones $\chi_i(t)$, tal que el índice discreto j del contexto multivariado, se sustituye por un índice continuo t. Las sumas sobre j se reemplazan por integrales sobre t. Por consiguiente el ACPF se encuentra como sigue:

1. Se halla la función de pesos $\epsilon_1(t)$ que maximiza $\sum_i f_{i1}^2$ sujeto a $\int_T \epsilon_1^2(t)dt = |\boldsymbol{\epsilon}_1|^2 = 1$. con $\int_T \chi_i(t)\epsilon_1(t)dt$.
2. Se realiza una segunda y hasta p nuevas etapas. En la $m-$ésima etapa se calcula un nuevo vector de pesos $\epsilon_m(t)$ y un nuevo componente principal ϵ_{jm}, tal que $\sum_j f_{jm}^2$ se maximiza sujeto a las restricciones $|\boldsymbol{\epsilon}_m|^2 = 1$ y $\int_T \epsilon_k(t)\epsilon_m(t)dt = 0$.
3. En la versión funcional se trabaja con la función de covarianza y no con la de la correlación (porque los valores de las funciones están en la misma escala). Por tanto se toma $\int_T \upsilon(s,t)\epsilon(t)dt = \rho\epsilon(t)$ como ecuación propia, donde $\upsilon(s,t) = \sum_{i=1}^{n} \chi_i(s)\chi_i(t)$ es la función de covarianza ($\chi_i(t)$ ha sido centrada), ρ es un valor propio y $\epsilon(t)$ es una función propia de la función de covarianza $\upsilon(s,t)$.

El ACPF pretende explicar el conjunto de funciones de la muestra a partir de unas cuantas, esto es:

$$\sum_{k=1}^{K} f_{i,k} \cdot \epsilon_k(t) \tag{3.17}$$

Cualquier función se puede representar como combinación lineal de las funciones propias; el problema de calcular valores y funciones propias, no es más que la aplicación del problema de buscar elementos propios en un espacio donde se trabaja con vectores en $\mathbb{R}^{\nvDash}$ o con funciones $\boldsymbol{L^2}$.

El problema de clasificación de Datos

La clasificación es una técnica muy útil, usada en diversos campos como el reconocimiento de patrones. En este trabajo se estudiarán algunas técnicas de uno de los dos tipos de clasificación existente, conocida como supervisada; como ejemplos de clasificación supervisada podríamos nombrar: el diagnóstico de enfermedades, la predicción de quiebra (o bancarrota) en empresas, el reconocimiento de caracteres escritos a mano, y muchas otras en la minería de datos, etc.

El objetivo de la clasificación supervisada dentro del aprendizaje automático consiste en la asignación de un objeto a una de las diversas categorías o clases especificadas, la clase es un sinónimo de categoría, es decir; una agrupación de objetos que tiene variables comunes.

3.4. Clasificación Supervisada

Este tipo de clasificación cuenta con un conocimiento a priori, es decir para la tarea de clasificar un objeto dentro de una categoría o clase contamos con modelos ya clasificados (objetos agrupados que tienen características comunes). Podemos diferenciar dos fases dentro de este tipo de clasificación: en la primera, tenemos un conjunto de entrenamiento (o de aprendizaje) para diseñar el clasificador y otro llamado de test o de validación para la clasificación, éstos nos servirán para construir un modelo o regla general para la clasificación. Por otro lado, la segunda fase consiste del proceso de clasificar en sí los objetos o muestras de las que se desconoce la clase a las que pertenecen.

A continuación se enumera algunas de las principales técnicas de clasificación supervisada:

- Análisis Discriminante Lineal.
- Métodos No Lineales: Discriminación Cuadrática, Regresión Logística, Projection Pursuit.

- Naive Bayes y Redes Bayesianas.
- Clasificadores basados en reglas.
- Árboles de Decisión.
- k vecinos más cercanos.
- Clasificadores basados en estimación de densidad por Kernel. Clasificadores que usan mezclas Gausianas.
- Redes Neurales: El perceptron de multicapas, Funciones bases radiales, mapas auto-organizantes de Kohonen.
- Máquinas de soporte vectorial.

Etapas de la Clasificación Supervisada

El problema general de clasificar N individuos procedentes de una muestra en un conjunto de $M < N$ clases en función de una serie de K variables $(X_1, X_2, \ldots, X_K)$, se compone de dos fases o etapas:

- Determinación del número de clases y de las propiedades de estas en relación a las K variables.
    1. Se asume que cada objeto pertenece a una clase predefinida, según es definido en la columna de clases.
    2. El modelo es representado por reglas de clasificación, árboles de decisión, o fórmulas matemáticas.
- Asignar cada uno de los N individuos a una de las M clases utilizando una regla de decisión basada en las propiedades de los individuos y las clases en relación a las K variables.
    1. Estimar la precisión del modelo
    2. La columna de clases de los objetos de la muestra de prueba es comparado con las clases predichas por el modelo. La muestra de prueba debe ser independiente de la muestra de entrenamiento de lo contrario puede ocurrir "sobreajuste".

3. La tasa de precisión es el porcentaje de objetos de la muestra de prueba que son correctamente clasificadas por el modelo.

Podemos anotar, que dentro de los usos de la Clasificación Supervisada encontramos entre otras cosas importantes, que ésta nos ayuda a predecir valores categóricos que representan clases y construir un modelo basado en la muestra de entrenamiento y posteriormente, lo usa para clasificar nuevos datos.

En este trabajo de tesis se ha usado sólo dos clasificadores k-NN y Árboles de Decisión.

3.4.1. k-NN

En el método de los k vecinos más cercanos ("k-NN: k Nearest Neighbors") ([18] Fix y Hodges, 1951) se estima la función de densidad de donde provienen un conjunto de datos. Aplicado a clasificación supervisada el método $k - NN$ se usa para estimar la función de densidad condicional $f(x/C_j)$, de las predictoras x para cada clase C_j. Este método de clasificación es "no paramétrico", ya que no se hace ninguna suposición distribucional acerca de las variables predictoras. A continuación damos una descripción del método

Los ejemplos de entrenamiento son vectores en un espacio característico multidimensional, cada ejemplo está descrito en términos de p atributos considerando q clases para la clasificación. Los valores de los atributos del i-ésimo ejemplo (donde $1 \leqslant i \leqslant n$) se representan por el vector p-dimensional $x_i = (x_{1i}, x_{2i}, ..., x_{pi}) \in X$.

El espacio es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un observación $\mathbf{x}$ es asignada a la clase C si esta es la clase

más frecuente entre los k ejemplos de la muestra de entrenamiento que están más cercanas de x. Generalmente se usa la distancia euclidiana para determinar la cercanía entre dos observaciones x_i y x_j

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{p}(x_{ri} - x_{rj})^2} \tag{3.18}$$

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación del ejemplo (del que no se conoce su clase) es representada por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los k ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados.

Este método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos; el problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes.

Para corregir el posible sesgo se puede asignar un peso a las distancias de cada atributo, dándole así mayor importancia a los atributos más relevantes. Otra posibilidad consiste en tratar de determinar o ajustar los pesos con ejemplos conocidos de entrenamiento. Finalmente, antes de asignar pesos es recomendable identificar y eliminar los atributos que se consideran irrelevantes.

Elección de la Métrica y el mejor k

Existen dos problemas fundamentales en el método $k - NN$, la elección de la distancia o métrica y la elección de k.

- La métrica más elemental que se puede elegir es la euclideana $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y})^t(\mathbf{x}-\mathbf{y})$. Esta métrica sin embargo, puede causar problemas si las variables predictoras han sido medidas en unidades muy distintas entre sí. Algunos prefieren rescalar los

datos antes de aplicar el método. Otra distancia bien usada es la distancia *Manhatan* definida por $d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$. Si hay predictoras de distinto tipo, esto es continuas, nominales y ordinales, entonces el uso de la distancia euclideana nos es un apropiada. Se pueden usar distancia para variables mixtas discretas son continuizadas, pero esto es muy complejo de hacer.

- La mejor elección de k depende fundamentalmente de los datos; generalmente, valores grandes de k reducen el efecto de ruido en la clasificación, pero crean límites entre clases parecidas. Un buen k puede ser seleccionado mediante una optimización de uso. El sesgo del error de clasificación aumenta a medida que k aumenta, en tanto que la varianza disminuye. Se ha demostrado [19](Cover y Hart, 1967) que la tasa de error del clasificador $k - NN$ es a lo más dos veces la tasa de error óptimo (error del clasificador Bayesiano donde las posteriores son conocidas.

El Clasificador k-NN

Desde el punto de vista de clasificacion supervisada el método $k - NN$ es muy facil de aplicar. En efecto, si las funciones de densidades condicionales $f(x/C_i)$ de la clase C_i que aparecen en la ecuación

$$P(C_i/x) = \frac{f(x/C_i)\pi_i}{f(x)}, \tag{3.19}$$

son estimadas por $k - NN$. Entonces, para clasificar un objeto, con mediciones dadas por el vector $\mathbf{x}$, en la clase C_i se debe cumplir que

$$\frac{k_i \pi_i}{n_i v_k(x)} \geqslant \frac{k_j \pi_j}{n_j v_k(x)}, \tag{3.20}$$

para $j \neq i$. Donde k_i y k_j son los k vecinos de x en las clase C_i y C_j respectivamente. Asumiendo priors proporcionales a los tamaños de las clases (n_i/n y n_j/n respectivamente) lo anterior es equivalente a: $k_i > k_j$ para $j \neq i$.

Luego, el procedimiento de clasificación sería en dos partes, así:

1. Hallar los k objetos que están a una distancia más cercana al objeto **x**; k, usualmente es un número impar.
2. Si la mayoría de esos k objetos pertenecen a la clase C_i entonces el objeto x es asignado a ella. **En caso de empate se clasifica al azar**. Las dos estapas descritas anteriormente se implementan en el pseudocódigo del Algortimo 3 de k-NN.

Pseudocódigo Algoritmo 3 : K Nearest Neighbor

```
for t = 1 to T do
    S_t ← S_{t-1}
    for s_q ∈ S_t do
        N_q ← k vecinos cercanos
        of s_q usando D(s_q, s_i)
        label(s_q)= argmax Σ_{s_i∈N_q} D(s_q, s_i);
        if label(s_q)≠ y_q then
            for s_i ∈ N_i do
                if y_i ≠ y_q then
                    w_i^t ← w_i^t − λ/d(x_q, x_i);
                else
                    w_i^t ← w_i^t + λ/d(x_q, x_i);
                end if
            end for
        end if
    end for
    if label(s_q)= y_q, ∀s_q then
        break
    end if
end for
```

3.4.2. Árboles de Clasificación

Un árbol de clasificación está basado en un paradigma de la Teoría de la Información. En éste, se realiza un particionamiento recursivo del dominio de definición de las variables predictoras, y se representa el conocimiento sobre el problema por medio de una estructura de árbol.

A continuación presentamos las ideas más relevantes de esta teoría que se fundamenta en el algoritmo básico de inducción del modelo a partir de los datos, para luego tratar los principales algoritmos para árboles.

El Algoritmo Básico

A continuación vamos a introducir las ideas fundamentales del denominado algoritmo TDIDT (Top Down Induction of Decision Trees) el cual puede ser contemplado como uniformizador de la mayoría de los algoritmos de inducción de árboles de clasificación a partir de un conjunto de datos conteniendo patrones etiqueados. La metodología a seguir puede resumirse en dos pasos, y se esquematiza en la Figura 3–1:

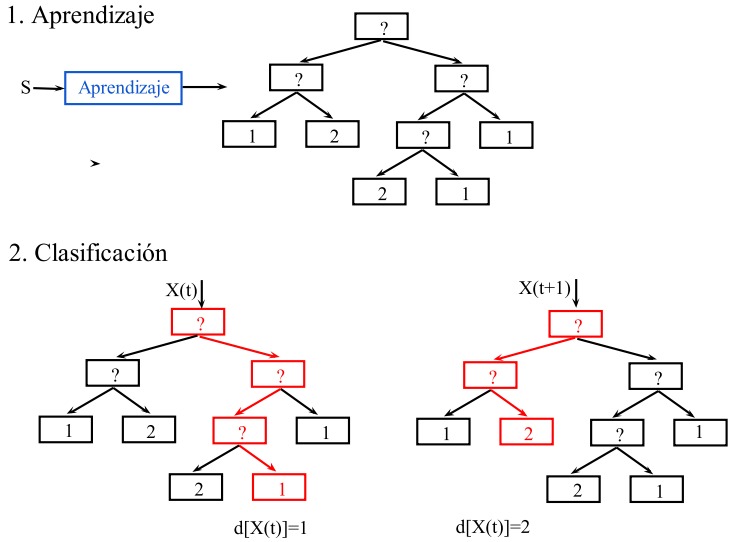

Figura 3–1: Etapas del árbol de clasificación

- **Aprendizaje**: Consiste en la construcción del árbol a partir de un conjunto de prototipos. Se constituye la fase más compleja y la que determina el resultado final. A esta fase dedicamos la mayor parte de nuestra atención.
- **Clasificación**: Consiste en el etiquetado de un patrón, X, independiente del conjunto de aprendizaje. Se trata de responder a las preguntas asociadas a los nodos

interiores utilizando los valores de los atributos del patrón X. Este proceso se repite desde el nodo raíz hasta alcanzar una hoja, siguiendo el camino impuesto por el resultado de cada evaluación.

Algoritmo ID3

Uno de los algoritmos de inducción de árboles de clasificación más populares es el denominado ID3 introducido por [20] (Quinlan, 1986). En él, el criterio escogido para seleccionar la variable más informativa está basado en el concepto de cantidad de información mutua (representadad por ganancia en información) entre dicha variable y la variable clase, esto debido a que $I(X_i, C) = H(C) - H(C|X_i)$ y lo que viene a representar dicha cantidad de información mutua entre Xi y C es la reducción en incertidumbre en C debida al conocimiento del valor de la variable X_i. Matemáticamente se demuestra que este criterio de selección de variables favorece la elección de variables con mayor número de valores. Además ID3 efectúa una selección de variables previa, consistente en efectuar un test de independencia entre cada variable predictora X_i y la variable clase C, de tal manera que para la inducción del árbol de clasificación tan sólo se van a considerar aquellas variables predictoras para las que se rechaza el test de hipótesis de independencia.

Algoritmo CART

En 1984, Breiman, Friedman, Oishen y Stone introdujeron un algoritmo para la construcción de árboles y los aplicaron a problemas de regresión y clasificación, el método se denomina CART (Classification and regression Trees) por sus siglas en inglés. Nosotros usaremos este algoritmo en la construcción de los árboles de clasificación. Las diferencias principales entre los algoritmos para cosntruir árboles se hallan en la regla para particionar los nodos, la estrategia para podar los árboles y el tratamiento de valores perdidos.

El pseudocódigo del Algoritmo 4 para árboles de clasificación, muestra que mientras todos los patrones que correspondan a una determinada rama del árbol de clasificación no pertenezcan a una misma clase, se seleccione la variable que de entre las no seleccionadas en esa rama sea la más informativa o la más idónea con respecto de un criterio previamente establecido. La elección de esta variable sirve para expandir el árbol en tantas ramas como posibles valores toma dicha variable.

Pseudocódigo Algoritmo 4 : Árbol de Clasificación

Input: D conjunto de N patrones etiquetados, los cuales están caracterizados por n variables predictoras $X_1, X2, \ldots, X_n$ y la variable clase C.
Output: Arbol de clasificación.

if todos los patrones de D pertenecen a la misma clase c **then**
  la inducción es un nodo simple (nodo hoja) etiquetado como c
else
  while $N \neq 0$ **do**
    1. Seleccionar la variable más informativa X_r con valores $x_r^1, \ldots, x_r^{nr}$
    2. Particionar D de acorde con los n_r valores de X_r en $D_1, \ldots, D_{nr}$
    3. Construir n_r subárboles $T_1, \ldots, T_{nr}$ para $D_1, \ldots, D_{nr}$
    4. Unir X_r y los n_r subárboles $T_1, \ldots, T_{nr}$ con los valores $x_r^1, \ldots, x_r^{nr}$
  end while
end if
return Tasa de error de clasificación

3.4.3. Validación cruzada

La validación cruzada (cross-validation) es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar cómo de preciso es un modelo que se llevará a cabo a la práctica. Es una técnica muy utilizada en proyectos de inteligencia artificial para validar modelos generados.

Supongamos que tenemos un modelo con uno o más parámetros desconocidos, y un conjunto de datos en el que el modelo puede estar entrenado (el conjunto de datos de entrenamiento). El proceso de adaptación optimiza los parámetros del modelo para que el modelo se ajuste a los datos de entrenamiento, tanto como sea posible. Si luego tomamos una muestra independiente de los datos de validación de la misma población que los datos de entrenamiento, por lo general resultan de que el modelo no se ajusta a los datos de validación, así como que se ajusta a los datos de entrenamiento. Esto se llama "*overfitting*", y es particularmente probable que ocurra cuando el tamaño del conjunto de datos de entrenamiento es pequeña, o cuando el número de parámetros en el modelo es grande. La validación cruzada es una manera de predecir el ajuste de un modelo a un conjunto de validación hipotética cuando un conjunto de validación explícita no está disponible. Ver [21] (Hjorth, 1993).

Estimación del error de mala clasificación

La evaluación de las diferentes validaciones cruzadas normalmente viene dada por el error obtenido en cada iteración, ahora bien, para cada uno de los métodos puede variar el número de iteraciones, según la elección del diseñador en función del número de datos total.

Estimación del error por validación cruzada de k iteraciones

En cada una de las k iteraciones de este tipo de validación se realiza un cálculo de error. El resultado final lo obtenemos a partir de realizar la media aritmética de los K valores de errores obtenidos, según la fórmula:

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i. \tag{3.21}$$

Donde:

E_i Representa el i-ésimo error. k Número de errores.

Estimación del error por validación cruzada aleatoria

En la validación cruzada aleatoria a diferencia del método anterior, cogemos muestras al azar durante k iteraciones, aunque de igual manera, se realiza un cálculo de error para cada iteración. El resultado final también lo obtenemos a partir de realizar la media aritmética de los K valores de errores obtenidos, según la misma fórmula:

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i. \tag{3.22}$$

Donde:

E_i Representa el i-ésimo error. k Número de errores.

Estimación del error por validación cruzada dejando uno fuera

En la validación cruzada dejando uno fuera se realizan tantas iteraciones como muestras (N) tenga el conjunto de datos. De forma que para cada una de las N iteraciones se realiza un cálculo de error. El resultado final lo obtenemos realizando la media aritmética de los N valores de errores obtenidos, según la fórmula:

$$E = \frac{1}{N} \sum_{i=1}^{N} E_i. \tag{3.23}$$

Donde:

E_i Representa el i-ésimo error. k Número de errores.

En nuestro trabajo, usamos este tipo de validación cruzada en la estimación de la tasa de error de mala clasificación.

Capítulo 4
METODOLOGÍA

En este capítulo se describirá los métodos y procedimientos que utilizaremos tanto para la reducción de la dimensionalidad como para la clasificación de los conjuntos de datos funcionales en estudio, la Figura 4–1 resume esquemáticamente la metodología de nuestro trabajo. La implementación de todas las funciones en ambos procesos se hará usando el lenguaje estadístico R.

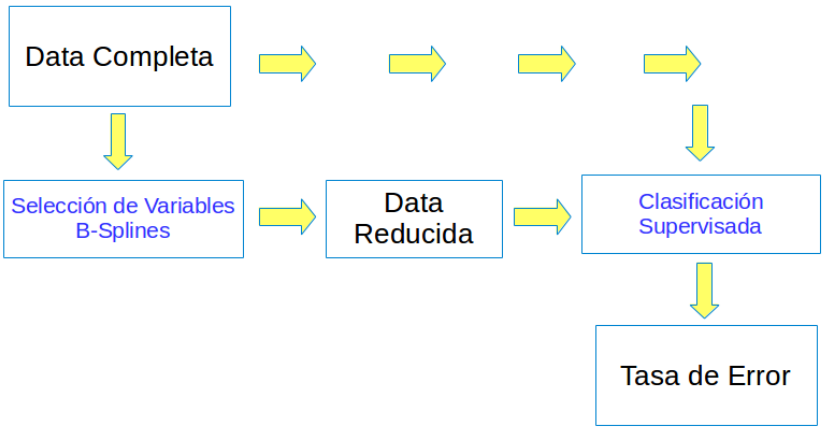

Figura 4–1: Flujograma de la Metodología.

4.1. Metodología usada para la Reducción de la Dimensionalidad

En relación a la reducción de la dimensionalidad, tal como muestra la Figura 4–2, se comparó los resultados de aplicar 3 métodos de selección de variables como son: el método de mRMR [11], el de Máxima Relevancia (MaxRel) y Relief; frente a la reducción de dimensión por B-Splines.

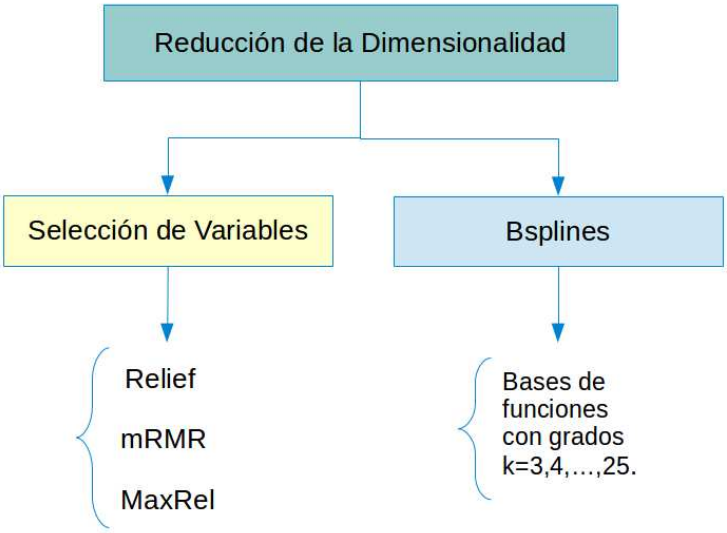

Figura 4–2: Métodos de Reducción de Dimensionalidad.

En el caso del método mRMR, se evaluó los dos esquemas de selección de variables propuestos para el caso Discreto, como son: la Diferencia de Información Mutua (MID) y el Cociente de Información Mutua (MIQ), ambos métodos combinan la relevancia y la redundancia de las variables utilizando información mutua, mediante las relaciones detalladas en la Tabla 3–1 del Capítulo anterior. Por otro lado, en vista de que estos métodos están propuestos para variables discretas, en casos como el nuestro, donde se trabaja con datos continuos, es necesario primero discretizar los conjuntos de datos; para ello se hará uso del método de discretización propuesto en [11], que establece la siguiente relación entre la media y la desviación estándar: $(\mu \pm \alpha * \sigma,\ \text{con } \alpha = 1, \frac{1}{2}, 0)$; por tanto a partir de este criterio, para $\alpha = 1$ ó $\alpha = \frac{1}{2}$ los valores del conjunto de datos asumirán 3 nuevos valores (-1,0,1), de acuerdo a:

$$Discretizacion = \begin{cases} -1 & \text{si } x < \mu - \alpha * \sigma \\ 0 & \text{si } \mu - \alpha * \sigma \leqslant x \leqslant \mu + \alpha * \sigma \\ 1 & \text{si } x > \mu + \alpha * \sigma \end{cases} \tag{4.1}$$

Mientras que si usamos $\alpha = 0$ obtendremos una binarización en el conjunto de datos que dependerá de si $x \leqslant \mu$ ó $x > \mu$.

De lo anterior, podemos ver que la elección de α tendrá alguna influencia en el conjunto de variables seleccionadas, sobretodo en el orden en que MID y MIQ, las seleccionen; pensando en eso en esta tesis se usó los tres valores de α para cada uno de los 2 métodos de mRMR tanto en el proceso de selección como en el proceso posterior de clasificación. En este sentido, sólo para fines de notación en el capítulo de Resultados se asumirán 6 métodos de mRMR (MID1, MID2, MID3, MIQ1, MIQ2 y MIQ3); donde la parte numeral del método indica los valores α : $1, 0{,}5 y 0$ respectivamente.

Tanto para la obtención de las variables seleccionadas por los métodos mRMR y MaxRel se usó un código implementado por Chris Ding y Hanchuan Peng [11] disponible en http://penglab.janelia.org/proj/mRMR/, en esta página los autores presentan el código de mRMR en distintos lenguajes y para diferentes entornos, la versión online permite trabajar con bases de datos de hasta 10 000 variables. Como resultado de la implementación obtenemos para ambos métodos, el ranking de variables seleccionadas y el peso de selección de cada una de ellas. Es importante resaltar que en MaxRel, al igual que para los métodos MID y MIQ de mRMR, se considera también el criterio de discretización previa a la selección de variables descrita en la ecuación (4.1); y para los 3 valores de α, se asumirán 3 métodos de MaxRel.

Con respecto a la obtención de variables seleccionadas mediante Relief, se hizo uso de la función attrEval de la librería CORElearn de R, que usa como criterio de selección, la relación que tienen todas las variables con respecto a la clase; esta función genera como resultado un peso para cada variable, el cual es usado para rankearlas y posteriormente reducir la dimensión del conjunto de datos original.

Por otro lado, en relación a la reducción por B-Splines, en vista de que para la construcción de una función spline debemos especificar un sistema de funciones de base y el grado de las mismas; se implementó un código en R que genere bases con grados entre 3 y 25. Así, luego de obtenida una base, el código implementado la

usa para realizar una regresión entre los valores de Y (todos los puntos que forman una curva) y ella como variable regresora. Finalmente, se obtiene para cada curva, los coeficientes de la ecuación de regresión que caracteriza la función suavizada, los cuales son usados posteriormente en la clasificación supervisada.

4.2. Metodología usada para la Clasificación Supervisada

En todos los casos, luego de obtener un conjunto de variables seleccionadas y/o reducir la dimensión de la data, se hizo uso de k-NN y Árboles de Decisión como clasificadores para obtener la tasa de error de mala clasificación. Para ello en los códigos de clasificación implementados se usaron funciones de R cuando no se consideraba validación cruzada; mientras que para el caso en que se consideraba la validación cruzada, se usó la función de clasificación "crossval", que hace parte de la librería dprep ([22]). En ambos clasificadores se trabajó con 10 particiones y 10 repeticiones; y se consideró 5 vecinos próximos para k-NN. La Figura 4–3 resume las características principales del proceso de clasificación.

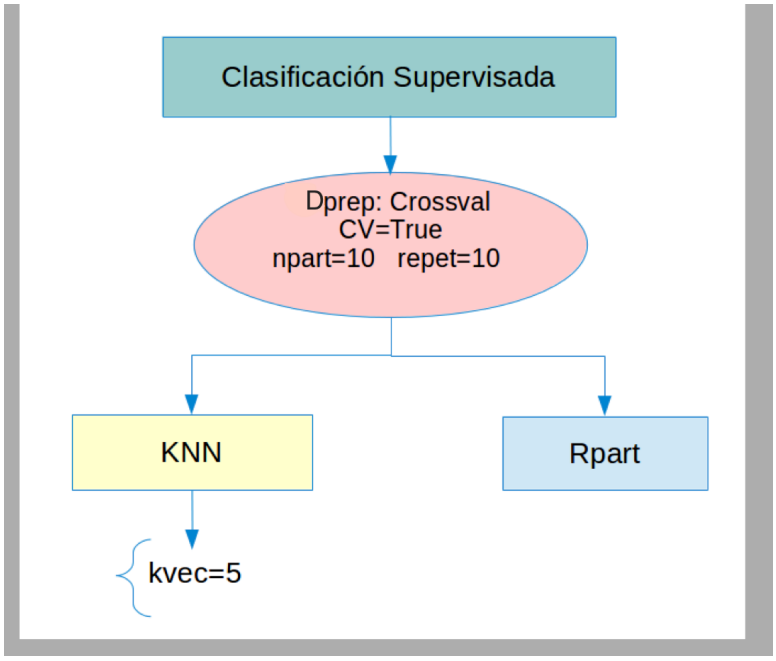

Figura 4–3: Clasificadores.

4.3. Conjunto de datos Tecator

Este conjunto de datos consiste de 215 piezas de carne finamente picada, para cada una de las cuales se observa una curva de espectrometría ($\mathbf{x_i},\ i = 1, 2, 3, ..., 215$) correspondiente a la absorbancia medida en 100 longitudes de onda (850-1050)nm esto quiere decir que $\mathbf{x_i} = (\chi_i(\lambda_1), ..., \chi_i(\lambda_{100}))$. La absorbancia es el $-log_{10}$ de la transmitancia medida por el espectrómetro.

Por otro lado, para la i - ésima muestra de carne ($\mathbf{x_i}$) se conoce también su contenido de humedad, grasa y proteínas; obtenidas mediante un proceso químico analítico; de éstas últimas se elige a la grasa (y_i) como variable dicotómica de clasificación. Debido a que en un proceso de discriminación, tenemos que considerar una respuesta categórica en lugar de una escalar, el conjunto de curvas queda dividido en dos grupos según el contenido de grasa en la pieza de carne sea menor o mayor al 20 %. Por tanto las $y_1, y_2, ..., y_{215}$ son reemplazadas por $y_1^*, y_2^*, ..., y_{215}^*$, esto es:

$$\forall i = 1, 2, ..., 215, \qquad y_i^* = \begin{cases} 0 & \text{si } y_i \leqslant 20 \\ 1 & \text{si } y_i > 20 \end{cases} \tag{4.2}$$

Este conjunto de datos se encuentra disponible en http://www.math.univ-toulouse.fr/~ferraty/SOFTWARES/NPFDA/npfda-datasets.html, y según su distribución, podemos representarla en forma tabular de la siguiente manera:

	Col 1	...	Col j	...	Col 100	Clases
Row 1	$\chi_1(\lambda_1)$	...	$\chi_1(\lambda_j)$	...	$\chi_1(\lambda_{100})$	y_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Row i	$\chi_i(\lambda_1)$	...	$\chi_i(\lambda_j)$	...	$\chi_i(\lambda_{100})$	y_i
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Row 215	$\chi_{215}(\lambda_1)$	...	$\chi_{215}(\lambda_j)$	...	$\chi_{215}(\lambda_{100})$	y_{215}

Cuadro 4–1: Distribución del conjunto de datos Tecator

Gráficamente, se tiene:

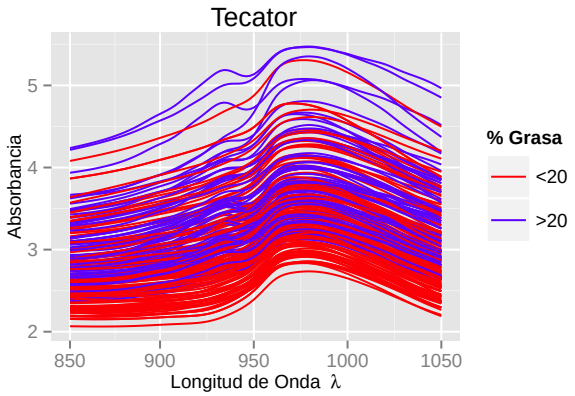

Figura 4–4: Curvas de Tecator

Tal como se observa en la Figura 4–4 una particularidad de este conjunto es que los datos en crudo son muy homogéneos y por tanto difíciles de clasificar, es por esta razón que en estudios donde trabajan con estos datos, se evalúa también el conjunto diferenciado ([10]), el cual es obtenido usando la función fdata.deriv de la librería fda.usc en R. En la Figura 4–5 se muestra la representación gráfica de las curvas diferenciadas Tecator de primer, segundo y tercer orden.

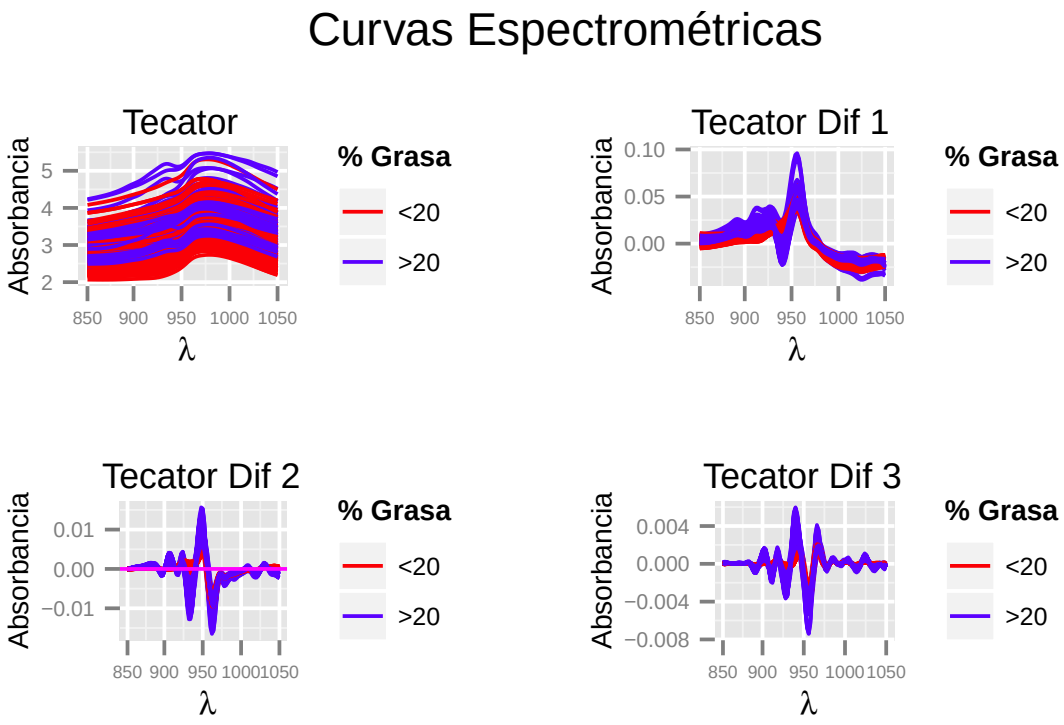

Figura 4–5: Curvas Diferenciadas Tecator

4.4. Conjunto de datos Phoneme

Este conjunto de datos es una parte del conjunto original, el cual se encuentra disponible en la dirección http://www-stat.stanford.edu/ElemStatLearn su estudio está relacionado al reconocimiento de voz, los datos representan log-periodograms correspondientes a las grabaciones de individuos durante 32 minutos, para esto se contó con la colaboración de Andreas Buja, Werner Stuetzle y Martin Maechler.

El estudio se refiere a cinco tramas de voz correspondientes a cinco fonemas transcritos; cada trama de voz está representada por 400 muestras a una velocidad de muestreo de 16 kHz; sólo las primeras 150 frecuencias de cada sujeto se retienen, y los datos constan de 2000 log-periodograms de longitud 150, con la pertenencia de clase de fonemas conocidos.

Por tanto, para $n = 2000$ se puede obervar los pares: $(\mathbf{x}_i, y_i)_{i=1,\ldots,n}$, donde $\mathbf{x}_i's$ corresponden a la discretización de "log-periodograms" (es decir $(\mathbf{x}_i = (\chi(f_1), \chi(f_2), \ldots \chi(f_{150}))$ es la i-ésima discretización de los datos) para cualquier miembro de la clase y_i que toma 5 fonemas diferentes como valores, distribuidos de la siguiente manera:

$$y_i \in \{1,2,3,4,5\} \text{ con } \begin{cases} 1 \longleftrightarrow \text{“sh”} \\ 2 \longleftrightarrow \text{“iy”} \\ 3 \longleftrightarrow \text{“dcl”} \\ 4 \longleftrightarrow \text{“aa”} \\ 5 \longleftrightarrow \text{“ao”} \end{cases}$$

Por otro lado, la distribución del conjunto de datos phoneme "npfda-phoneme.dat" consta de pares $(\mathbf{x}_i, y_i)_{i=1,\ldots,2000}$ que están ordenados como sigue:

	Col 1	...	Col j	...	Col 150	Clase
Row 1	$\chi_1(\lambda_1)$	...	$\chi_1(\lambda_j)$	...	$\chi_1(\lambda_{150})$	y_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Row i	$\chi_i(\lambda_1)$	...	$\chi_i(\lambda_j)$	...	$\chi_i(\lambda_{150})$	y_i
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Row 2000	$\chi_{2000}(\lambda_1)$	...	$\chi_{2000}(\lambda_j)$	...	$\chi_{2000}(\lambda_{150})$	y_{2000}

Cuadro 4–2: Distribución del conjunto de datos Phoneme

Las primeras 150 columnas corresponden a las 150 frecuencias y la última $(y_i's)$ contiene las respuestas categóricas (clases). El tamaño de cada clase es 400.

En la Figura 4–6 vemos el conjunto completo de datos Phoneme distinguidos por clase.

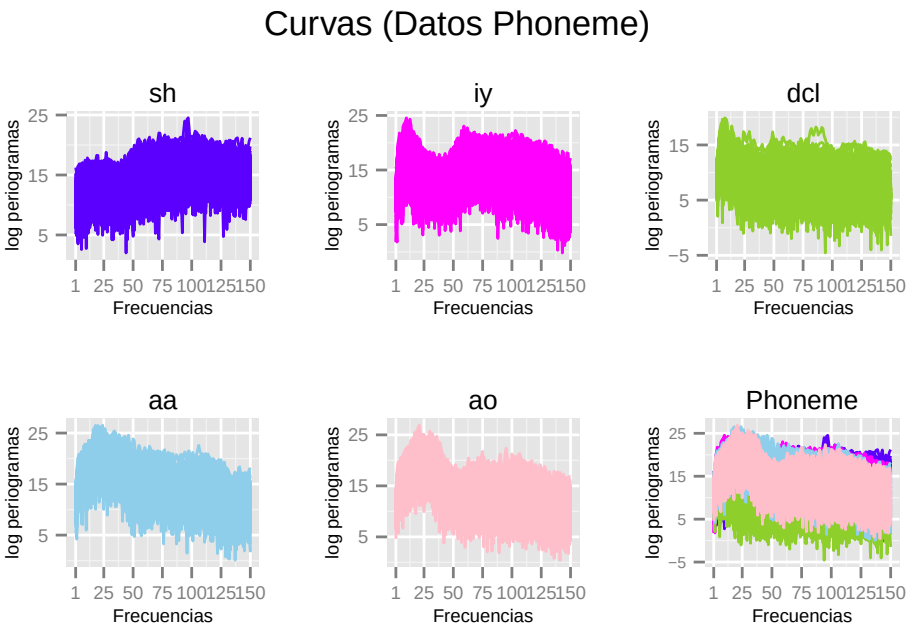

Figura 4–6: Curvas de Phoneme

Presentamos en la Figura 4–7 las primeras 5 curvas de cada clase de los datos Phoneme, en ésta última, se distingue mejor la variación y forma de las diferentes clases.

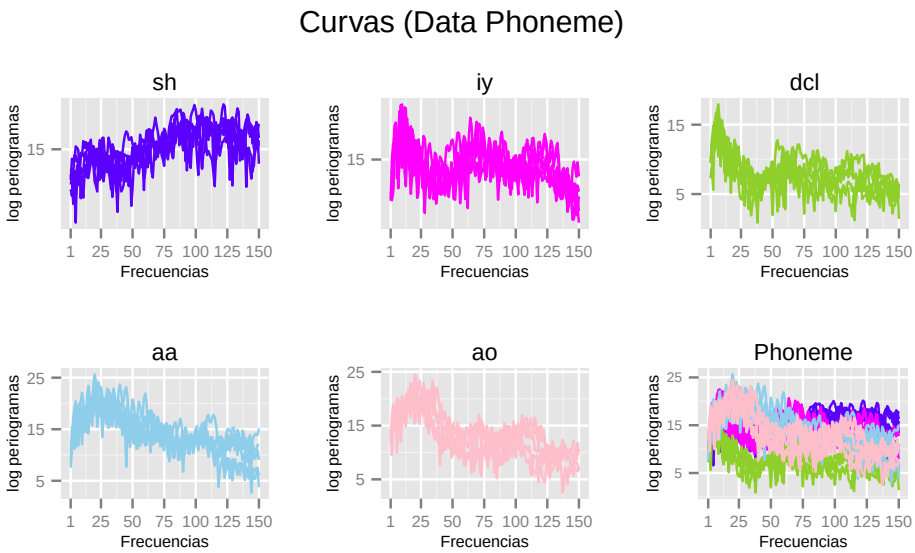

Figura 4–7: 5 Curvas de Phoneme

Capítulo 5
RESULTADOS

En este capítulo se mostrarán los resultados obtenidos durante las diversas pruebas de reducción y clasificación realizadas sobre los conjuntos de datos en estudio; tanto para la reducción como para la clasificación, los resultados se presentan divididos por métodos y por conjunto de datos.

5.1. Reducción de la Dimensionalidad

Como producto del proceso de selección de variables y reducción por B-Splines, para cada conjunto de datos se expondrá los resultados generados en dos etapas. A continuación se exhibe la primera de ellas, mediante las gráficas del puntaje de selección de cada método con respecto al número de variables seleccionadas.

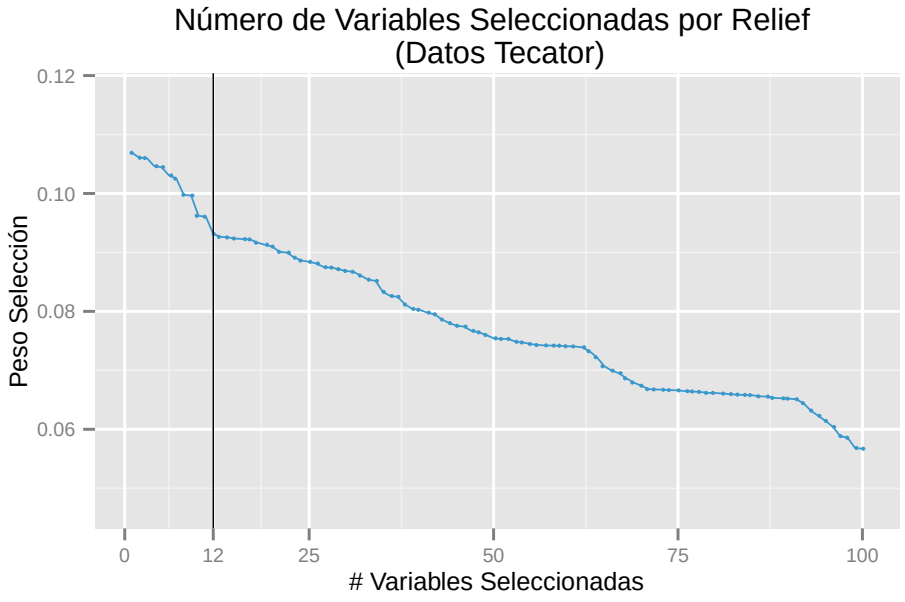

Figura 5–1: Puntaje de selección de Relief por número de variables seleccionadas.

En la Figura 5–1 observamos la curva formada con las puntuaciones del método Relief con respecto al número de variables a seleccionar en el conjunto Tecator; de ella se puede ver que este método induce a elegir 12 variables, pues aproximadamente en este valor, se presenta un punto de variación significativo en la curva.

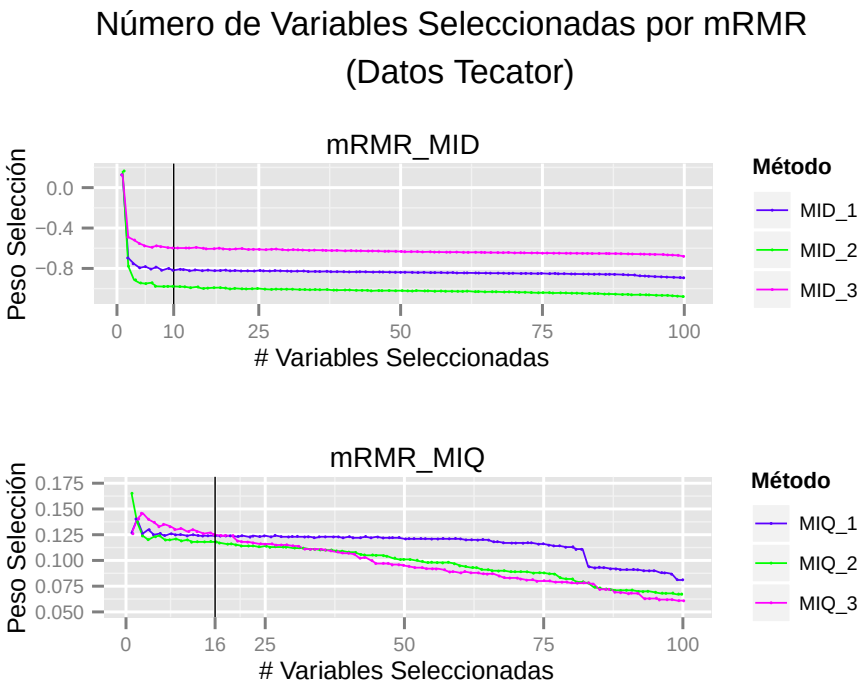

Figura 5–2: Puntaje de selección de los métodos mRMR por número de variables seleccionadas.

Las Figuras 5–2 y 5–3 muestran respectivamente las curvas de los puntajes de selección para los métodos mRMR y MaxRel, en la Figura 5–2 se observa que de las 3 curvas MID_1, MID_2 y MID_3 generadas por las variaciones de threshold ($\alpha = 1$, 0,5 y 0) del método mRMR(MID), la curva MID_3 es la que mayores puntajes de selección presenta, seguida por la curva MID_1 y finalmente MID_2, esto indica que la discretización binaria sobre el conjunto de datos Tecator, genera mayores puntajes en el proceso de selección del método MID; sin embargo de acuerdo a la ligera estabilización de las oscilaciones en las 3 curvas, se puede inferir que mRMR(MID) sugiere trabajar con las primeras 10 variables seleccionadas, mientras que el intercepto de las curvas mRMR(MIQ) induce a seleccionar 16 variables mediante este método.

En el caso de las puntuaciones de selección del método MaxRel (ver Figura 5–3), tenemos que las curvas generadas por los 3 valores de “threshold"no muestran una clara estabilidad en su distribución; del mismo modo, en ellas se ve que las correspondientes a MaxRel_2 y MaxRel_3 además de tener un comportamiento similar, presentan el primer cambio brusco en las puntuaciones con las primeras 8 a 10 variables seleccionadas, y un segundo en las 10 siguientes; sin embargo, la notoria reducción de las puntuaciones obtenidas si se selecciona más de 25 variables, define considerar ese valor, como un aproximado del número de variables a seleccionar al usar este método.

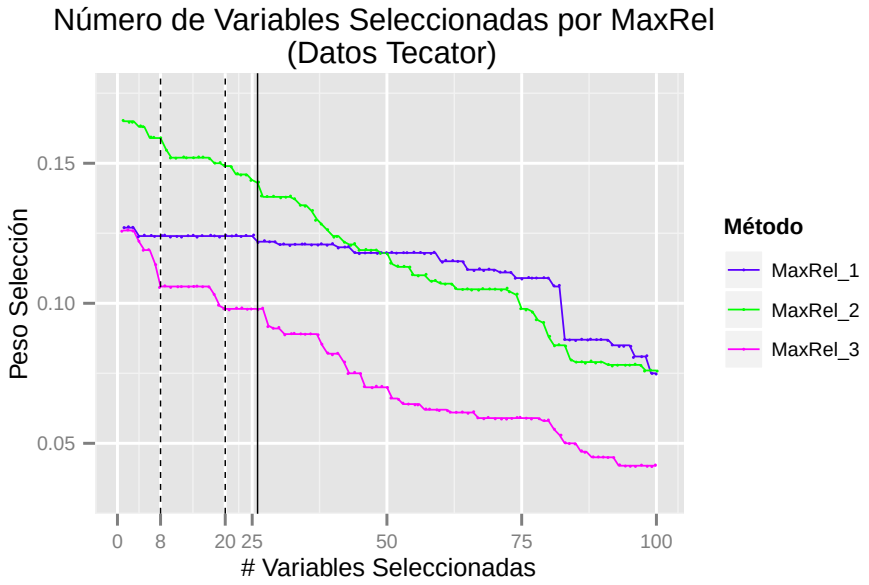

Figura 5–3: Puntaje de selección de los métodos MaxRel por número de variables seleccionadas.

Las siguientes gráficas muestran las principales variables que cada método selecciona en el conjunto de datos Tecator, el cual contiene 100 variables. Mostramos en la Figura 5–4 las puntuaciones de selección asociadas a cada variable por el método Relief; las líneas verticales trazadas señalan las dos primeras variables (100 y 41) que este método selecciona en el primer y segundo lugar respectivamente.

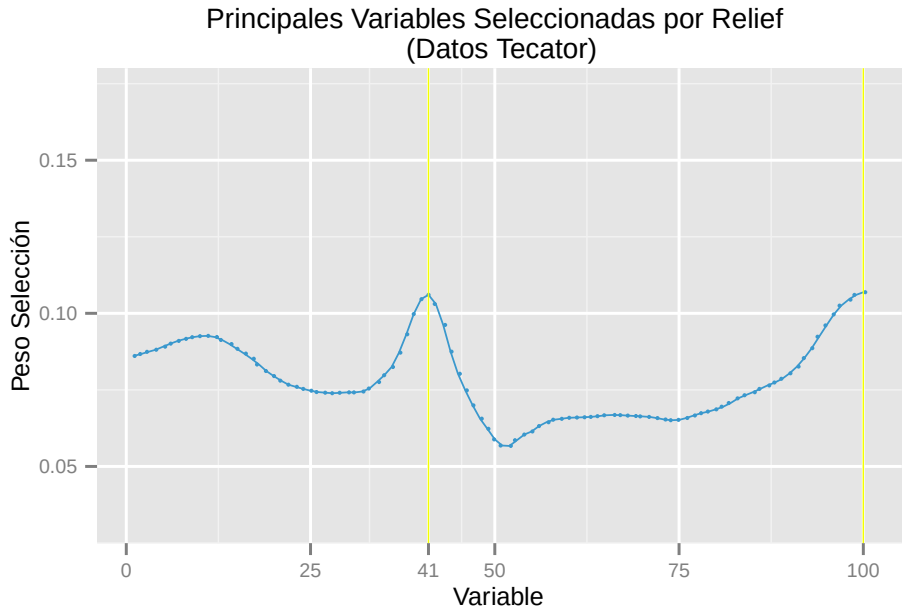

Figura 5–4: Puntaje de las mejores variables seleccionadas por Relief

La Figura 5–5 corresponde a los resultados de mRMR, en ella se puede distinguir no sólo la coincidencia con Relief en la selección de 2 de las mejores variables (100 y 41), sino también varias coincidencias entre los métodos (MID y MIQ) de mRMR.

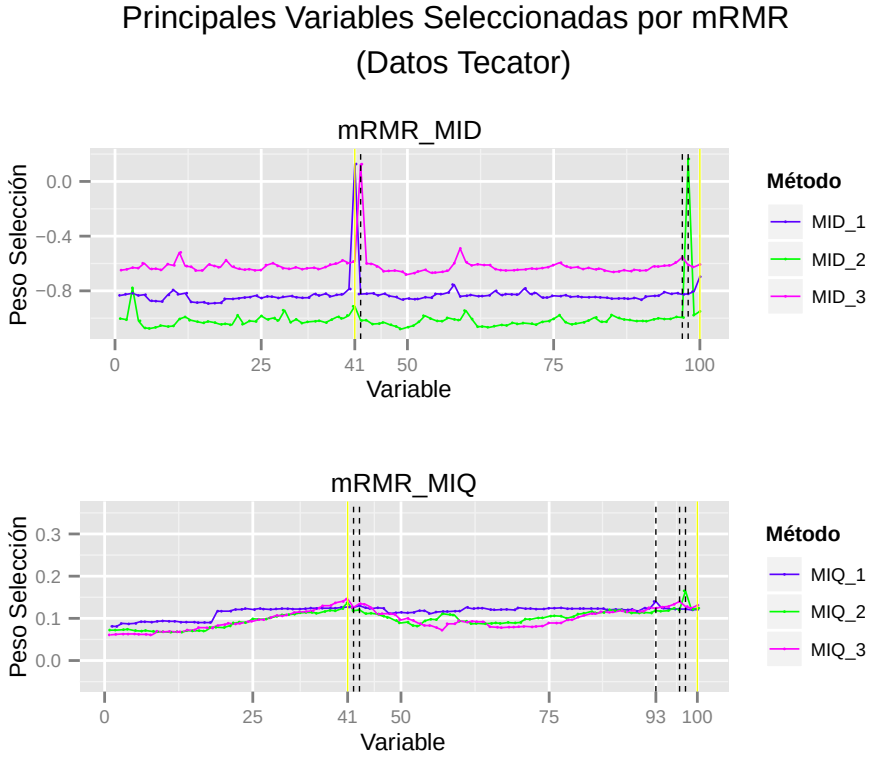

Figura 5–5: Puntaje de las mejores variables seleccionadas por mRMR

Todas estas coincidencias son distinguibles mediante los picos más altos de sus respectivas curvas. En este sentido se puede observar que las puntuaciones más altas del método MID son para las variables 41, 98, 97, 100 y 42; mientras que en el caso de MIQ se incluyen además de las anteriores, las variables 42 y 93.

Por otro lado, la Figura 5–6 expone las curvas de las puntuaciones que el método MaxRel otorga a cada una de las 100 variables de Tecator, nuevamente en ella se divisa la predominancia de las variables 100 y 41 como las mejores y algunas otras coincidencias tales como las variables 98, 97, 42 y 93.

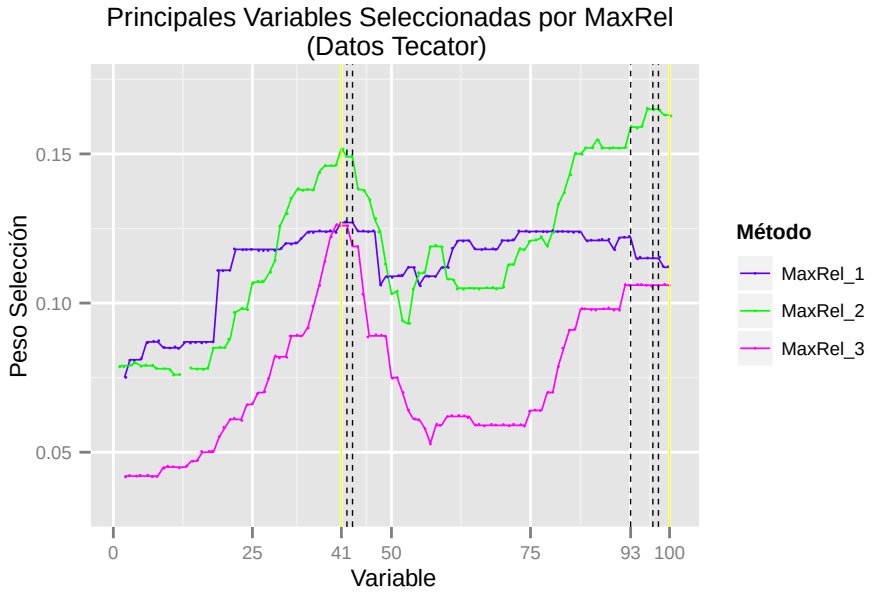

Figura 5–6: Puntaje de las mejores variables seleccionadas por MaxRel

Las figuras anteriores se resumen en las siguientes dos tablas, las cuales describen el listado de las primeras 5 y 10 variables seleccionadas, con respecto al número de métodos que las seleccionan.

Del conjunto de variables mostrados en la Tabla 5–1 , la 41, 42, 100 y 98 fueron elegidas por al menos 3 métodos para ocupar los 3 primeros lugares en la selección de variables; sin embargo de ellas destaca aún más la variable 41, no sólo por liderar el ranking al ser elegida por 8 métodos, sino porque además 3 de ellos la eligieron como la mejor variable (puesto 1) y otros 3, como la segunda mejor variable.

Variable	Longitud de Onda	Número Métodos	Relief	MID			MIQ			MaxRel		
				1	2	3	1	2	3	1	2	3
41	930.808	8	x	x	x		x	x	x	x		x
42	932.828	4				x			x	x		x
40	928.788	4	x					x	x	x		
100	1050	3	x	x				x				
98	1045.960	3			x			x			x	
39	926.768	3				x			x			x
43	934.848	3					x			x		x
97	1043.939	3				x			x		x	
99	1047.979798	2	x								x	

Cuadro 5–1: Primeras 5 variables seleccionadas del conjunto de datos Tecator.

Por otro lado a pesar de que las variables 98 y 42 son seleccionadas en el puesto 1 por 3 métodos cada una, ambas son superadas por las variables 100 y 40 debido a que éstas además de ser seleccionadas por mRMR y MaxRel, también fueron elegidas por Relief entre las mejores variables para distinguir entre una clase y otra.

La Tabla 5–2 nos muestra un resumen de las primeras 10 variables seleccionadas del conjunto de datos tecator. Es importante resaltar que en la Tabla 5–2 se enumera sólo aquellas variables que fueron elegidas por al menos dos métodos. Podemos ver que la distribución de esta tabla presenta una ligera variación con respecto a la anterior (ver Tabla 5–1), tanto en la aparición de nuevas variables elegidas, como en el incremento de la cantidad de métodos que habían seleccionado algunas de las variables entre las 5 primeras; y este hecho genera que éstas permanezcan liderando el conjunto de variables que permite distinguir de mejor manera entre las clases.

Variable	Longitud de Onda	Número Métodos
41	930.808	8
100	1050.000	7
40	928.788	7
42	932.828	6
97	1043.939	6
43	934.848	6
39	926.768	5
99	1047.980	5
98	1045.960	4
96	1041.919	4
44	936.869	4
38	924.747	4
94	1037.879	2
85	1019.697	2
76	1001.515	2

Cuadro 5–2: Primeras 10 variables seleccionadas en el conjunto de datos Tecator

La obtención de las derivadas de la Absorbancia (A) respecto a la Longitud de Onda (λ) es un método usado para determinar mejor la correlación entre los datos debido a que la derivada resalta las diferencias en curvas de absorción muy próximas y del mismo modo nos permite distinguir los puntos de máxima absorción.

Primera derivada de A respecto a λ

Representa la pendiente de cada punto de la banda de absorción, $\delta A/\delta\lambda = 0$ en el máximo de la banda de absorción, obteniéndose con gran exactitud el valor de la longitud de onda donde ocurre la mayor absorción.

Segunda derivada de A respecto a λ

$\delta^2 A/\delta\lambda^2$=0 en los puntos de inflexión de la banda de absorción, y en la posición de λ máximo se obtiene un valor mínimo que también permite establecer con gran exactitud el máximo de absorción del espectro.

La Figura 5–7 muestra los intervalos de longitud de onda que contienen a las primeras variables que fueron seleccionadas como las mejores por los distintos métodos de selección; en ellas se puede observar que a pesar de que en la gráfica original de los datos Tecator, éstas variables no distinguen bien entre las clases, en las curvas diferenciadas sí lo hacen, y de éstas la gráfica de la segunda derivada de Tecator exhibe claramente que los intervalos formados por las variables [V38,V44] y [V94,V100] que representan respectivamente los intervalos de longitudes de onda [924,936]nm (franja celeste) y [1037,1050]nm (franja rosada) permiten separar los cortes finos de carne que contienen 20 o menos porciento de grasa de los que contienen más del 20 %.

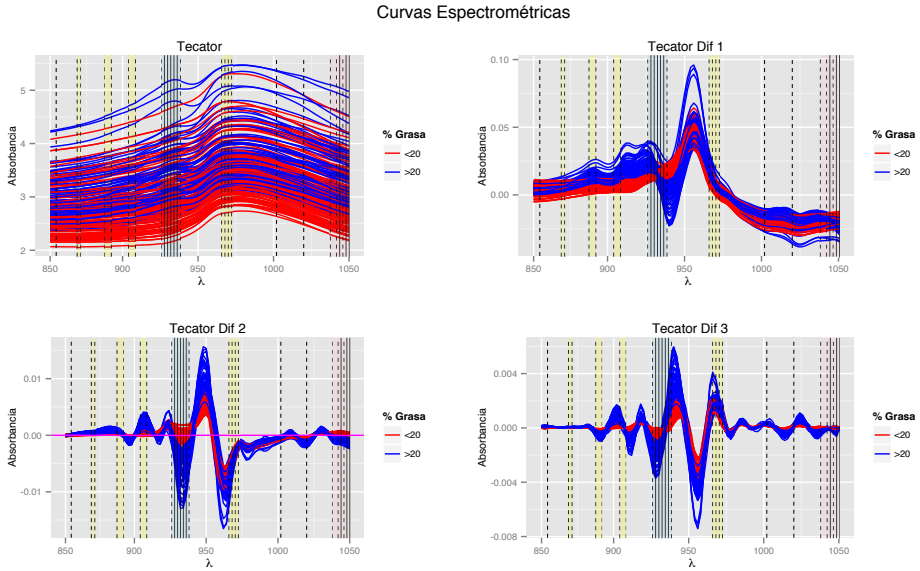

Figura 5–7: Curvas diferenciadas del conjunto de datos Tecator.

A continuación se muestran los resultados obtenidos al aplicar los métodos de reducción sobre el conjunto de datos Phoneme, similarmente a lo ocurrido en una primera parte de este capítulo con el conjunto de datos Tecator, aquí se exhibirán los gráficos del número de variables que cada uno de los métodos sugiere seleccionar, y el subconjunto de variables mejor puntuadas por los métodos en estudio, las cuales son eligas en las primeras posiciones. Finalmente estos últimos resultados se sintetizarán en dos tablas que muestren el ranking de las 5 y 10 mejores variables seleccionadas para el conjunto de datos Phoneme, que contiene un total de 150 variables.

La Figura 5–8 induce a inferir que el método Relief sugiere seleccionar entre 10 y 17 variables, debido a que tal como lo muestran las líneas verticales trazadas en esos y otros puntos del dominio, la curva tiende a tener cambios notorios en su comportamiento a partir de esos valores.

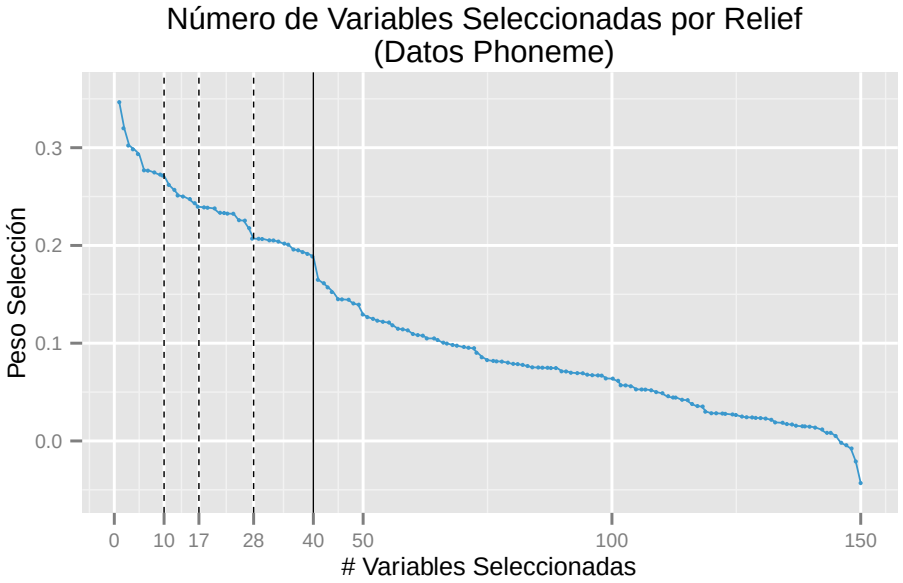

Figura 5–8: Número de Variables Seleccionadas por Relief

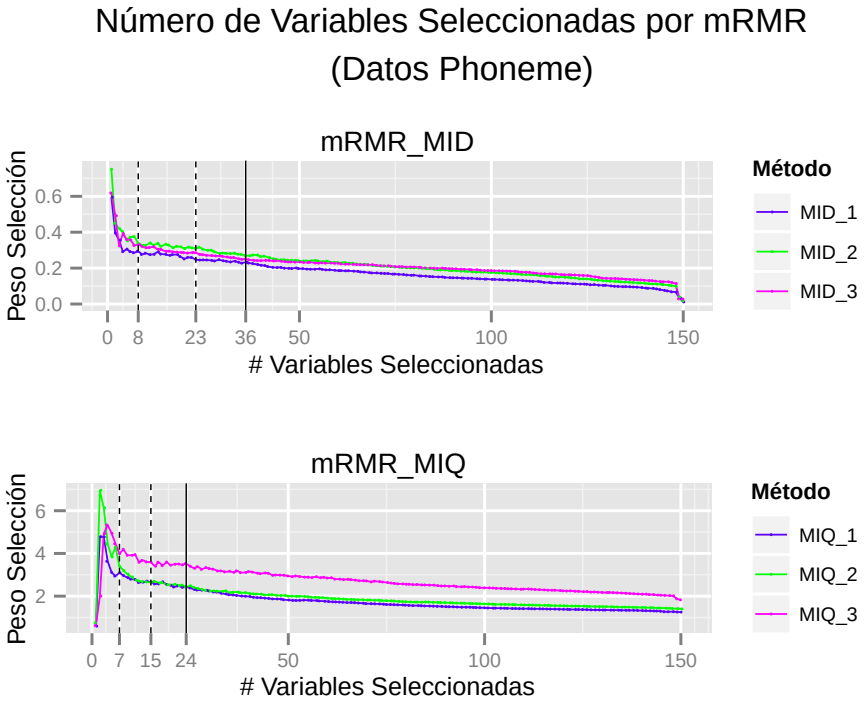

Figura 5–9: Número de Variables Seleccionadas por mRMR

Tanto las gráficas de los métodos MID y MIQ de mRMR (Figura 5–9) sugieren trabajar con pocas variables seleccionadas (ambas oscilan entre 7 y 25), pero de ellas las cruvas de MID presentan cierta estabilidad.

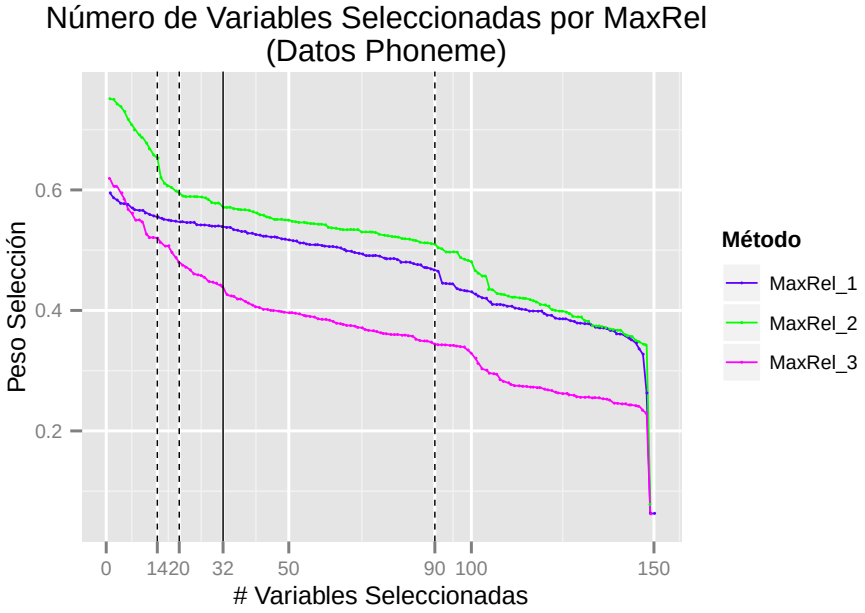

Figura 5–10: Número de Variables Seleccionadas por MaxRel

El gráfico correspondiente a las curvas del método MaxRel (Figura 5–10) señalan que el threshold correspondiente a $\alpha = 0{,}5$ genera mayores puntuaciones en la selección, mientras que la binarización en el proceso de discretización genera puntajes más bajos; sin embargo las 3 curvas muestran similitud en su forma y nos permiten deducir que este método sugiere elegir entre las primeras 15 y 20 variables, incluso podría resultar bueno en el proceso de clasificación si elegimos más de 32 variables.

En este punto se muestran las gráficas y tablas correspondientes a las puntuaciones más altas obtenidas en el proceso de reducción y que son asignadas a las variables primeramente seleccionadas del conjunto de datos Phoneme.

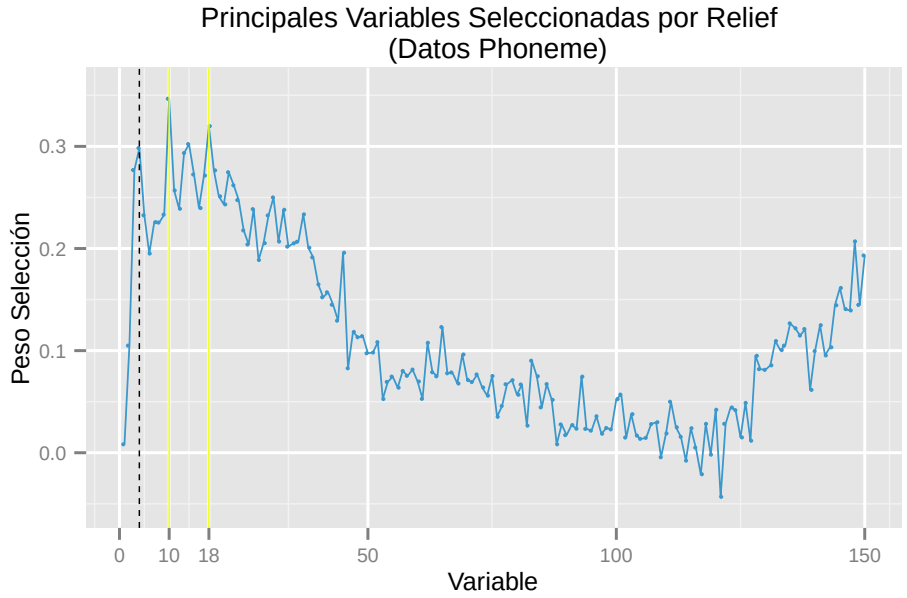

Figura 5–11: Puntaje de las mejores variables seleccionadas por Relief

El primero de este conjunto de gráficos (Figura 5–11), muestra las oscilaciones de las puntuaciones de Relief para cada una de las 150 variables de Phoneme, en él se trazó un par de líneas verticales en los dos puntos más altos observados, los cuales corresponden a la primera (10) y segunda (18) variables seleccionadas.

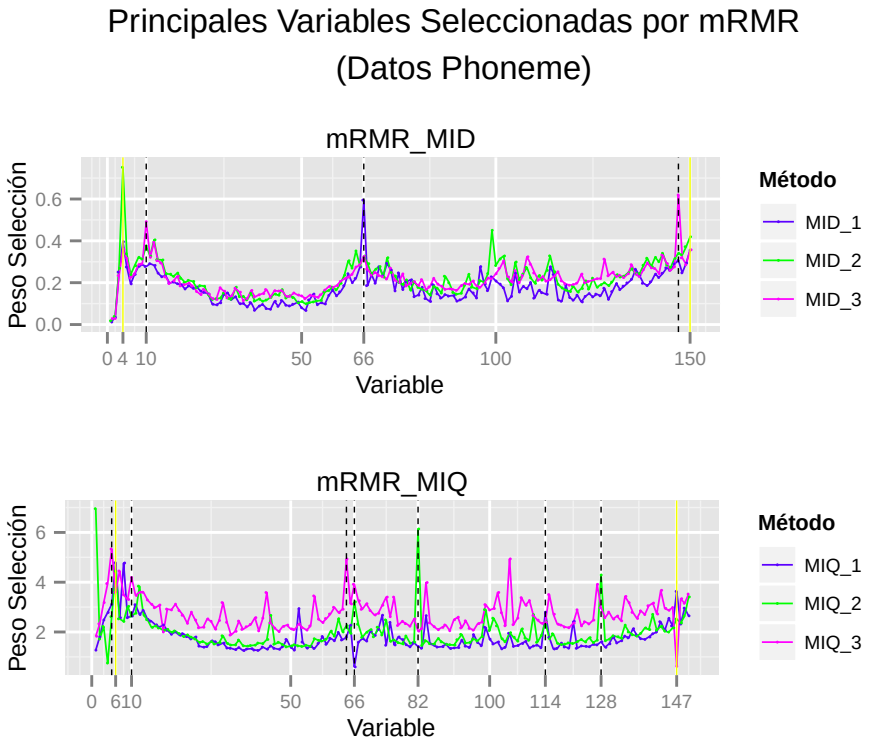

Figura 5–12: Puntaje de las mejores variables seleccionadas por mRMR

Las gráficas de las puntuaciones de los métodos mRMR(MID), mRMR(MIQ) y MaxRel, dos de ellas mostradas en la Figura 5–12 y la otra expuesta en la Figura 5–13 dejan ver las coincidencias en los picos más altos de las 9 curvas y con ello la sugerencia de estos método en la selección conjunta de las variable 4, 10, 66, 147 y 150.

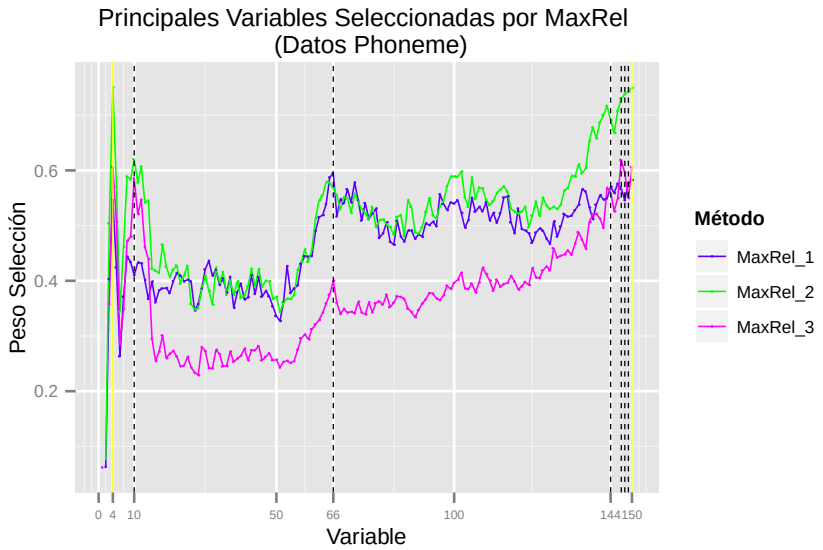

Figura 5–13: Puntaje de las mejores variables seleccionadas por MaxRel

Por otro lado a decir de las curvas de los métodos mRMR, se observa que en el caso de MID las oscilaciones de las puntuaciones son menores que las de MIQ; mientras que éstas a su vez muestran mayor estabilidad que las establecidas por las puntuaciones de selección de MaxRel.

En la Tabla 5–3 podemos observar de forma concisa la lista de mejores variables seleccionadas por los métodos, junto al número de métodos que las selecciona.

De este primer ranking podemos aducir que las variables 4, 147 y 150 lideran el ranking de las primeras variables seleccionadas; lo cual indicaría que pudieran ser las favoritas en el proceso de clasificación.

Variable	Número Métodos	Relief	MID			MIQ			MaxRel		
			1	2	3	1	2	3	1	2	3
4	7	x	x	x	x		x			x	x
147	6		x		x	x		x		x	x
150	5		x	x					x	x	x
66	3		x			x			x		
10	3	x			x						x
12	3			x	x		x				
18	2	x									x
5	2					x		x			
6	2					x	x				
64	2			x				x			
149	2								x	x	
148	2									x	x

Cuadro 5–3: Primeras 5 variables seleccionadas del conjunto de datos Phoneme.

De otro lado, debemos resaltar que las variables 10 y 18 cobran significativa importancia al ser seleccionadas por Relief, y por su parte, la última variable del conjunto de datos (150) destaca por ser elegida por 5 métodos entre las 3 mejores variables.

La Tabla 5–4 presentada a continuación confirma la información tanto gráfica como escrita previamente y del mismo modo a partir de ella se muestra la Figura 5–14 que exhibe los intervalos de frecuencias que incluyen a las mejores variables seleccionadas (franjas púrpura y verde), así como otras variables importantes (franjas amarillas) del conjunto Phoneme. En todos los intervalos formados, se distingue la diferencia y/o similitud en la altura y forma de las curvas correspondientes a las 5 diferentes clases.

Variable	Número Métodos
4	7
66	7
150	7
147	6
149	6
10	5
12	5
143	4
11	4
144	3
146	3
18	2
5	2
6	2
64	2
148	2
9	2
114	2
128	2

Cuadro 5–4: Primeras 10 variables seleccionadas en el conjunto de datos Phoneme

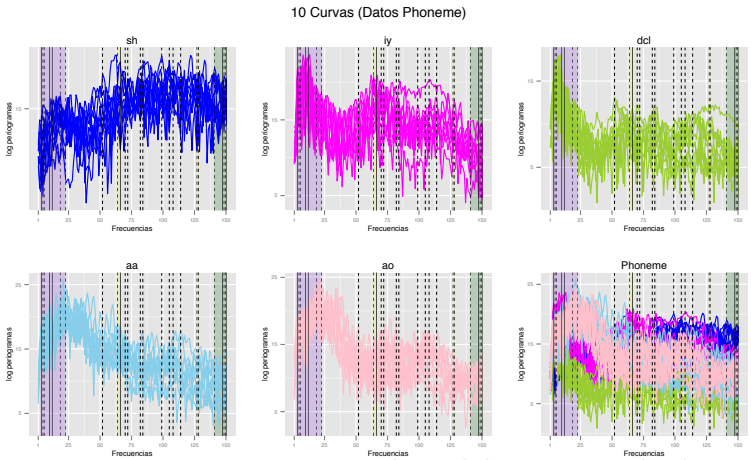

Figura 5–14: 10 Primeras curvas por clase del conjunto de datos Phoneme.

5.2. Clasificación Supervisada

Como se indicó en el capítulo anterior, para la aplicación de la clasificación supervisada se hizo uso de k-NN y árboles como clasificadores; a continuación se mostrarán algunos gráficos y tablas de la tasa de error de mala clasificación encontradas luego de aplicar los diferentes métodos de reducción de dimensionalidad a cada uno de los conjuntos de datos en estudio.

Estos resultados se expondrán en una primera parte de acuerdo al método de reducción y de forma general para ambos conjuntos de datos, para luego presentar los resultados especificos de clasificación para cada conjunto de datos. Es importante resaltar que en la mayoría de los casos mientras no se indique lo contrario, los resultados mostrados consideran el uso de validación cruzada en el proceso de clasificación.

La Figura presentada a continuación 5–15 representa para cada uno de los conjuntos de datos, las curvas de la tasa de error obtenidas usando Relief como método de selección, de ellas se puede ver en el primer caso, que aunque las curvas tienen similar comportamiento, las puntuaciones de la tasa de error para Tecator, presentan mayor oscilación que las correspondientes a la data Phoneme. De otro lado en ambas gráficas se distingue la variación significativa al usar validación cruzada en el proceso de clasificación, respecto a si no se usa.

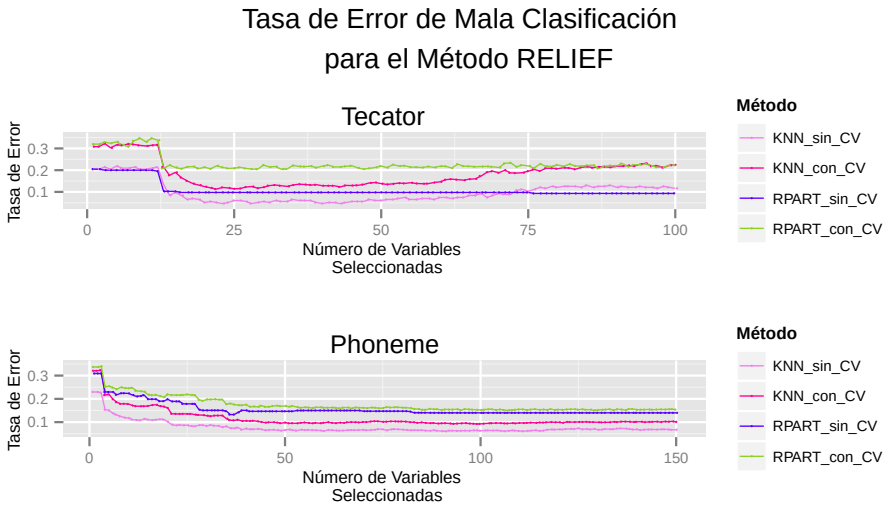

Figura 5–15: Tasa de Error de Mala Clasificación del método Relief

Las Figuras 5–16 y 5–17 muestran las curvas de estimación de la tasa de error de acuerdo al número de variables que los métodos mRMR seleccionan en cada conjunto de datos; en ellas se observa para todas las curvas que k-NN genera menores tasas de error de mala clasificación con respecto a Rpart (sobretodo cuando el conjunto de variables seleccionadas es pequeño), pues luego los valores de ambos clasificadores tienden a estabilizarse.

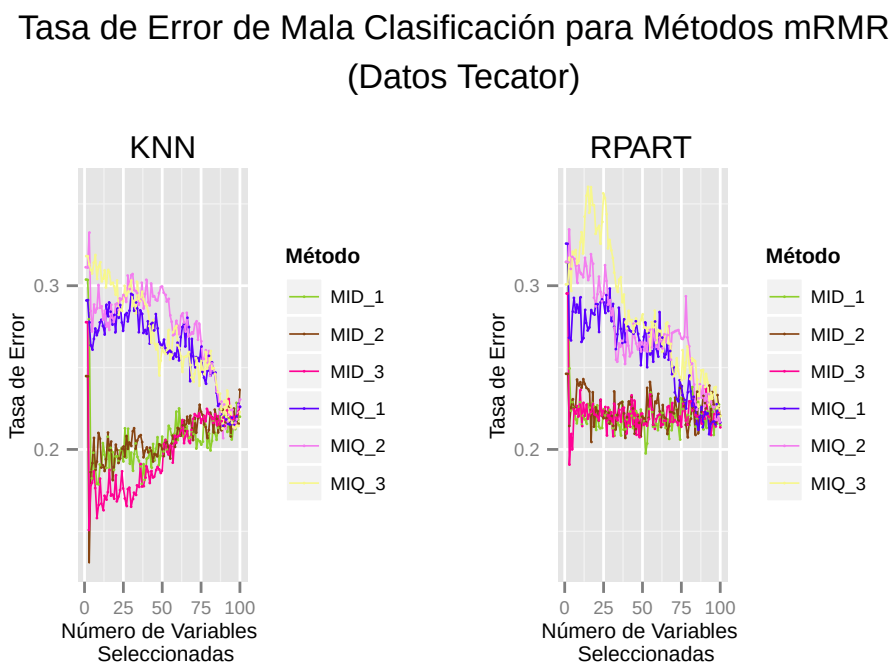

Figura 5–16: Tasa de Error de Mala Clasificación de mRMR para Tecator

En el caso del conjunto Phoneme (Figura 5–17), vemos que la distribución de las 6 curvas (3 de MID y 3 de MIQ) es similar entre los clasificadores, pero se observa también que para este conjunto de datos, el método MIQ genera valores ligeramente menores de tasas de error que los correspondientes a MID, para los 3 diferentes valores de threshold, anteriormente definidos.

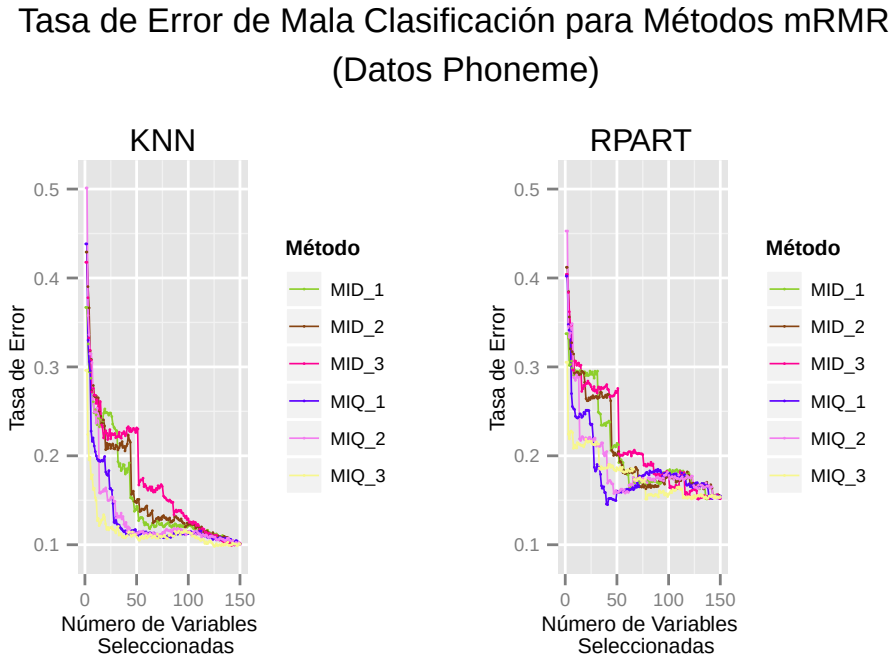

Figura 5–17: Tasa de Error de Mala Clasificación de mRMR para Phoneme

De lo anterior, se puede deducir que en Phoneme destacamos el método MIQ frente a MID, mientras que en Tecator resaltamos la supremacía de MID sobre MIQ, pues además de presentar valores diferenciados de tasa de error, se observa la estabilización de ellos con pocas variables seleccionadas.Por tanto a partir de estos resultados, se usarán más adelante el MID (para Tecator) y el MIQ (para Phoneme) como representantes del método mRMR para comparar simultáneamente las tasas de error de mala clasificación de todos los métodos de reducción estudiados aplicados a los conjuntos de datos Tecator y Phoneme, respectivamente.

De otra parte tenemos las gráficas de las tasas de error de mala clasificación aplicadas sobre los conjuntos de datos reducidos que contenían las variables seleccionadas por el método MaxRel.

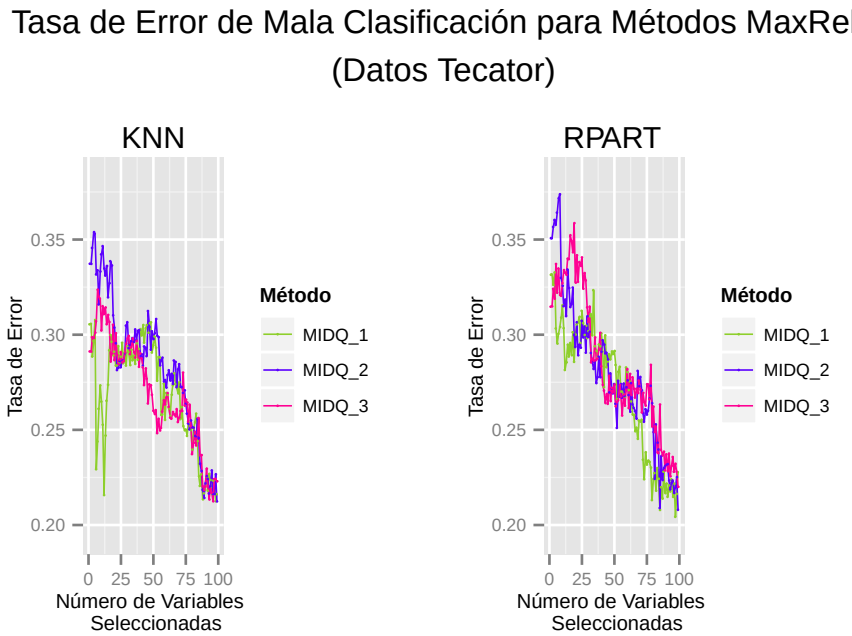

Figura 5–18: Tasa de Error de Mala Clasificación de MaxRel para Tecator

De estas dos figuras podemos observar que la asociada al conjunto de datos Tecator (Figura 5–18) muestra mayor variación entre los puntajes, pues se ve que luego de seleccionar las primeras 20 variables, existen tramos donde dichas curvas disminuyen intempestivamente. Por su parte la gráfica para Phoneme (Figura 5–19) muestra mayor estabilidad en los resultados de ambos clasificadores, sobretodo luego de trabajar con las primeras 50 variables.

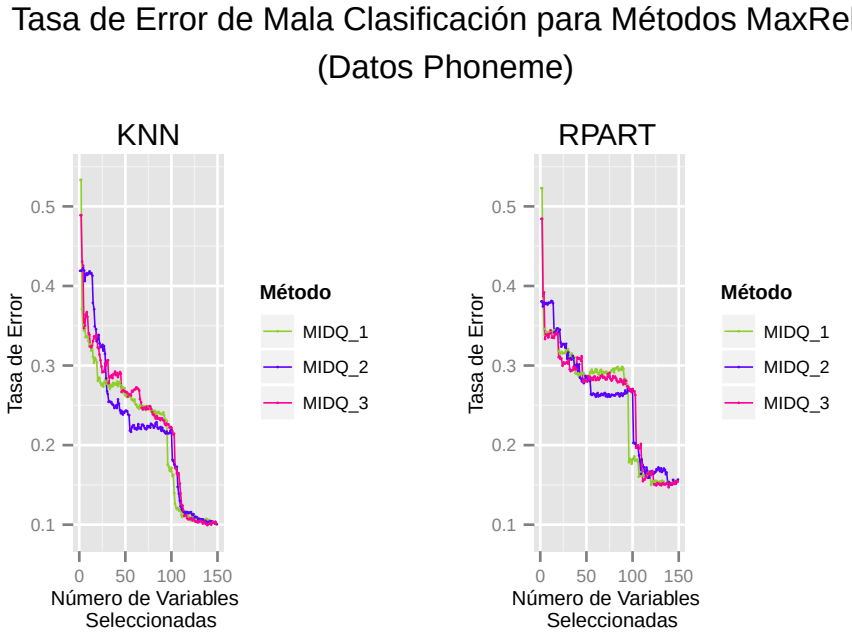

Figura 5–19: Tasa de Error de Mala Clasificación de MaxRel para Phoneme

Finalmente se observan las gráficas de las tasas de error de mala clasificación asociadas a la reducción de la dimensionalidad por B-Splines; en el caso del conjunto de datos Tecator (primera gráfica de la Figura 5–20) observamos un comportamiento bueno y homogéneo de las tasas en los B-Splines de grado medio, pues los valores más altos de la tasa de error se observan en los extremos, es decir tanto si se decide trabajar con B-Splines de grado pequeño (menor a 5) como si lo hacemos con uno de grado alto (mayor a 12). Por otra parte, si evaluamos el comportamiento de la variación de las tasas de error obtenidas como producto de la reducción por B-Splines en el conjunto Phoneme (segunda gráfica de la Figura 5–20), veremos que éste nos indica trabajar con polinomios de grado menor o igual a 10.

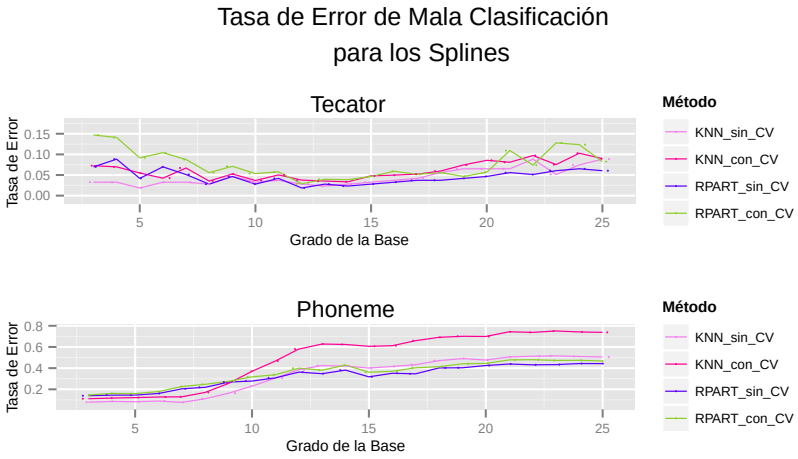

Figura 5–20: Tasa de Error de Mala Clasificación de los B-Splines

Con el propósito de verificar si se podía optimizar aún más la clasificación de los conjuntos de datos, se evaluó el desempeño de los B-Splines cuando se determinaba previamente un conjunto de nodos interiores. Para ello, tomando en cuenta los resultados obtenidos en los B-Splines sin nodos interiores (Figura 5–20), se consideró generar B-Splines de grados 3 a 8 y definir en ellos conjuntos con entre 1 y 5 nodos interiores; en este sentido para cada grado, se realizó la clasificación en tres grupos con distintas variables elegidas como nodos. Los resultados se muestran a continuación.

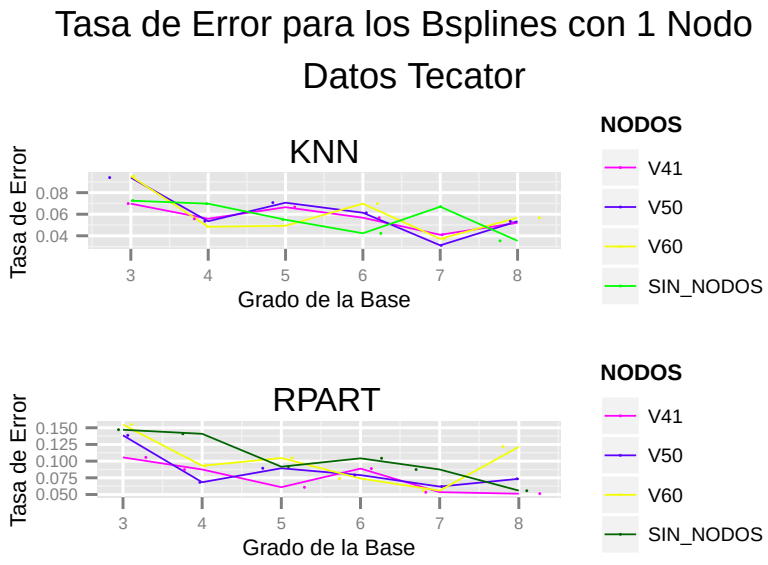

Figura 5–21: Tasas de error de mala clasificación para B-Splines con 1 Nodo.

Las Figuras 5–21 y 5–22 muestran para el conjunto de datos Tecator que las tasas de error de mala clasificación de las curvas B-Splines en las que se considera nodos interiores son mejores respecto a las obtenidas usando B-Splines sin nodos.

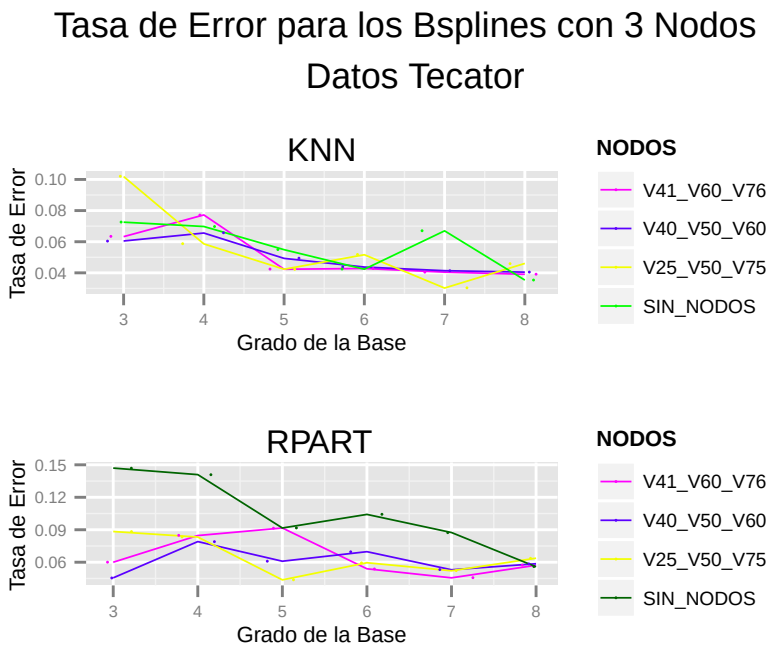

Figura 5–22: Tasas de error de mala clasificación para B-Splines con 3 Nodos.

Se pudo observar también que el ajuste de polinomios impares (preponderando grado 3 y 5) se desempeñan de mejor manera que ajustes pares, y que a medida que se incrementa el número de nodos interiores las tasas de error disminuían ligeramente.

Finalmente es importante resaltar que cuando se trabaja con 1 nodo interior (ver Figura 5–21) la variable 41 (**V41**) elegida como la más importante por los métodos de selección para Tecator, presenta las tasas de error más pequeñas comparada con la mediana (V50) y a la variable 60 (V60) que parecía ser determinante en la distribución de las absorbancias; por su parte la Figura 5–22 muestra que si consideramos 3 nodos interiores, los grupos de variables (**V40**, V50, V60) y el de los cuartiles (V25, V50, V75), se despempeñan muy bien.

En el caso del conjunto de datos Phoneme (Figuras 5–23 y 5–24) se observó que independientemente del número de nodos interiores, en todos los casos (excepto si usamos B-Splines de grado 3), el ajuste de B-Splines sin nodos (Figura 5–20) muestra valores más pequeños de tasas de error de mala clasificación.

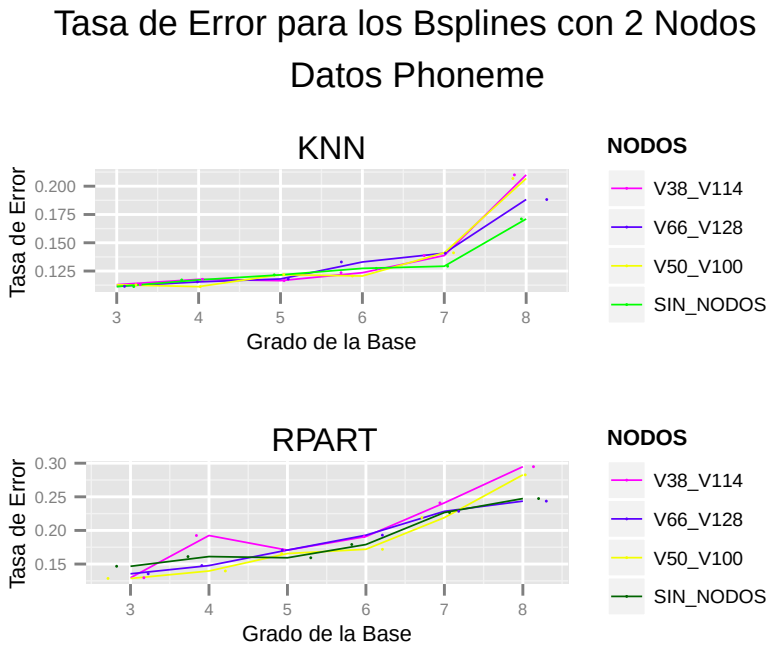

Figura 5–23: Tasas de error de mala clasificación para B-Splines con 2 Nodos.

Sin embargo, al igual que en Tecator, estas figuras nos muestran que el usar como nodos interiores algunas de las variables elegidas por los métodos de selección, disminuye la tasa de error frente a la elección de los percentiles, tal es el caso de la variable 66 (**V66**) al especificar 2 nodos (ver Figura 5–23), así como de los conjuntos de variables: (**V18**,V52,V82,V108,**143**) y (**V11**,V38,**V66**,V90,**128**).

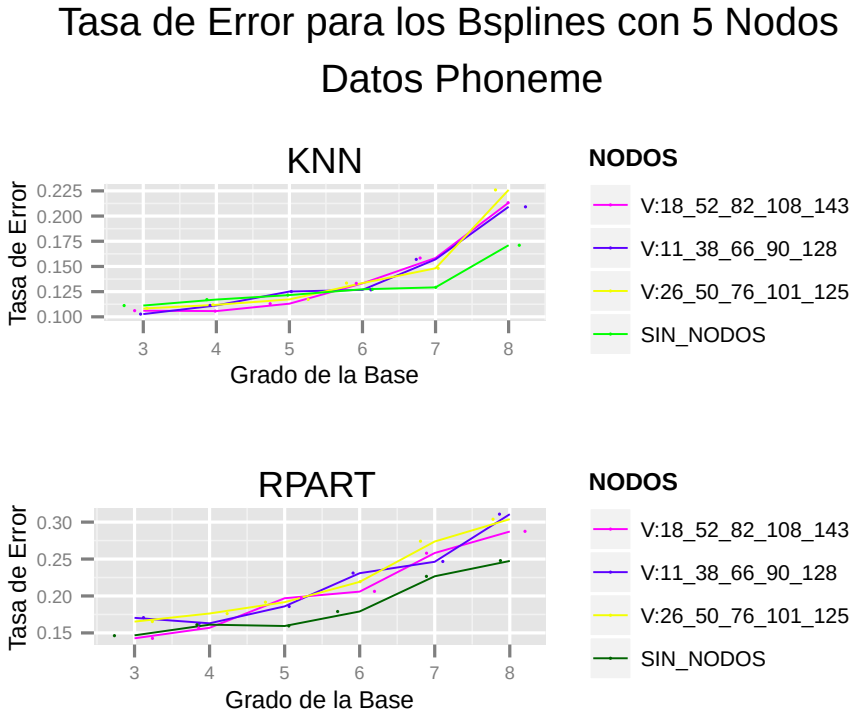

Figura 5–24: Tasas de error de mala clasificación para B-Splines con 5 Nodos.

A continuación presentamos los resultados de clasificación más resaltantes de forma especifíca para cada conjunto de datos en estudio.

5.2.1. Conjunto de datos Tecator

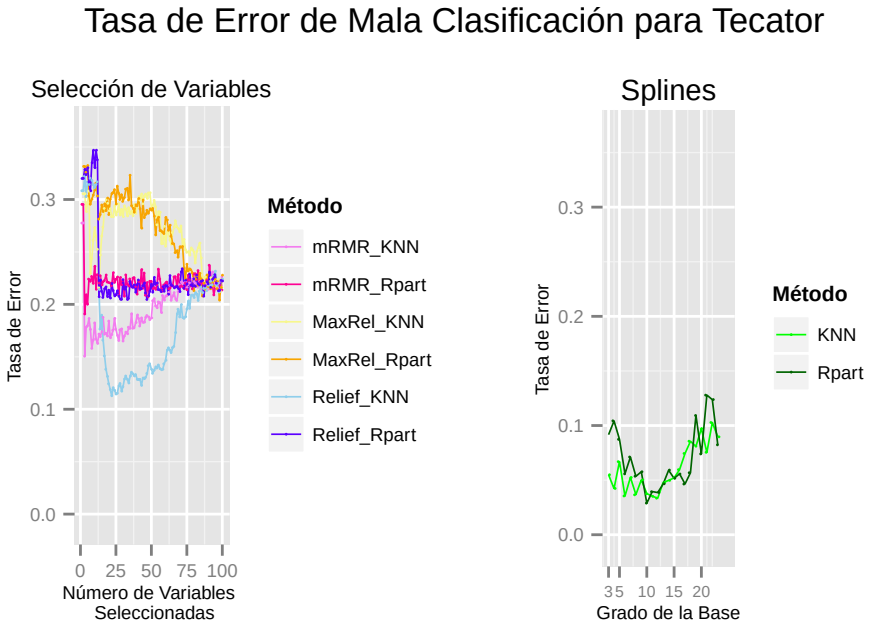

Figura 5–25: Tasa de Error de Mala Clasificación para Tecator

En la primera gráfica de la Figura 5–25 se presenta un resumen de la tasa de error obtenida por todos los métodos de reducción para los datos Tecator, mientras que la segunda corresponde al uso de los B-Splines, en estas gráficas se puede distinguir la superioridad de los B-Splines frente a todos los métodos de selección de variables, pues sus valores son menores en todo su dominio. Por otro lado, con respecto a los métodos de selección vemos que a pesar de que todas las tasas convergen hacia un mismo punto, tanto los resultados de Relief como los de mRMR son más óptimos que los asociados a MaxRel.

La Tabla 5–5 resume las mejores tasas de error obtenidas de los métodos de reducción, considerando el número de variables seleccionadas, en ella al igual que en la Figura 5–25 se puede ver que los B-Splines muestran una tasa de error de mala clasificación menor al 10 % trabajando con bases de grados entre 5 y 8; mientras que mRMR y Relief sugieren trabajar con las primeras 5 y 15 variables seleccionadas respectivamente para obtener una tasa de error similar a la encontrada al clasificar el conjunto de datos completo.

Método de Reducción de Dimensionalidad		k-NN		rpart	
		Sin CV	Con CV	Sin CV	Con CV
B-Splines	Nro de Variables	k=5:8	k=5:8	k=5:8	k=5:8
	Tasa Error	2.3 %-3.3 %	3.9 %-7 %	2.8 %-5 %	5 %-10 %
mRMR	Nro de Variables	3 a 5	3 a 5	3 a 5	3 a 5
	Tasa Error	1 %-1.4 %	1.4 %-3 %	1.9 %-2.8 %	3.4 %-3.9 %
Relief	Nro de Variables	15	15	15	15
	Tasa Error	11.2 %	18 %	10 %	23 %
Completa	Nro de Variables	100	100	100	100
	Tasa Error	12 %	23 %	10 %	23 %

Cuadro 5–5: Tasa de Error de Mala Clasificación para los datos Tecator

Para corroborar la información de la tabla anterior con respecto al buen trabajo que realizan los B-Splines, se presenta la gráfica 5–26. Como se explicó en el Capítulo de Metodología, en el caso de los B-Splines se generó bases con grados 3 a 25 y consecutivamente a ello, se realizó una regresión tomando a las bases generadas como variables independientes y al vector fila de la data como variable explicada, la Figura 5–26 muestra las 215 curvas de tecator estimadas por los B-Splines como resultado de este proceso, y en ellas se observa que la estimación de la base B-Splines con grado $k = 7$ se aproxima bastante a las curvas iniciales del conjunto de datos Tecator.

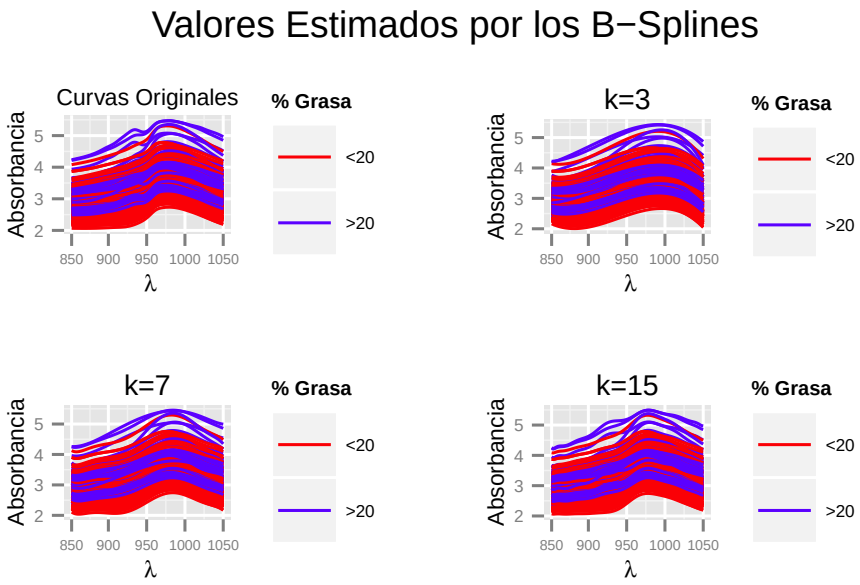

Figura 5–26: Curvas Espectrométricas estimadas por B-Splines de diferentes grados.

5.2.2. Conjunto de datos Tecator Diferenciados

En el caso del conjunto Tecator diferenciado, se presenta sólo los resultados asociados a la tasa de error de mala clasificación para la primera derivada del conjunto Tecator, debido a que con este nuevo conjunto de datos se obtienen valores pequeños de tasa de error de mala clasificación. La Tabla 5–6 resume los resultados obtenidos en la clasificación de este conjunto de datos y en ella se observa que por ejemplo el Conjunto diferenciado completo presenta una tasa de error de mala clasificación de sólo el 1.9 % al usar k-NN como clasificador y 4.5 % si usamos árboles.

Método de Reducción de Dimensionalidad		k-NN		rpart	
		Sin CV	Con CV	Sin CV	Con CV
MaxRel	Nro de Variables	2 a 5	1 a 5	1 a 5	1 a 5
	Tasa Error	0.5 %-1 %	1.3 %-3.3 %	1.9 %-2.8 %	3.3 %-3.8 %
mRMR	Nro de Variables	2 a 5	1 a 5	1 a 5	1 a 5
	Tasa Error	1 %-1.4 %	1.4 %-3 %	1.9 %-2.8 %	3.4 %-3.9 %
Relief	Nro de Variables	15	15	15	15
	Tasa Error	1.9 %	1.9 %	1.9 %	4.5 %
B-Splines	Nro de Variables	k=5	k=5	k=5	k=5
	Tasa Error	3.3 %	5 %	3.3 %	4 %
Completa	Nro de Variables	100	100	100	100
	Tasa Error	1.9 %	1.9 %	1.4 %	4.5 %

Cuadro 5–6: Tasa de Error de Mala Clasificación para Tecator Diferenciado

5.2.3. Conjunto de datos Phoneme

En el caso de este conjunto de datos, observamos un comportamiento similar al obtenido en Tecator, con respecto a la tasa de error de mala clasificación de los métodos de selección, siendo Relief y mRMR (en ese orden) los que mejor se desempeñan en comparación con MaxRel. Del mismo modo se puede observar en la Figura 5–27 que para todos los métodos de selección, el clasificador k-NN presenta menores tasas de error que el correspondiente a árboles (Rpart), mientras que en el caso de los B-Splines, se oberva un comportamiento diferente al de Tecator.

Tanto en la Figura 5–27 como en la Tabla 5–7 se observa un comportamiento monótono creciente para los B-Splines asociados a este conjunto de datos, mostrando que los B-Splines trabajan bien si es que se consideran bases de grado pequeño (en este caso entre 3 y 6).

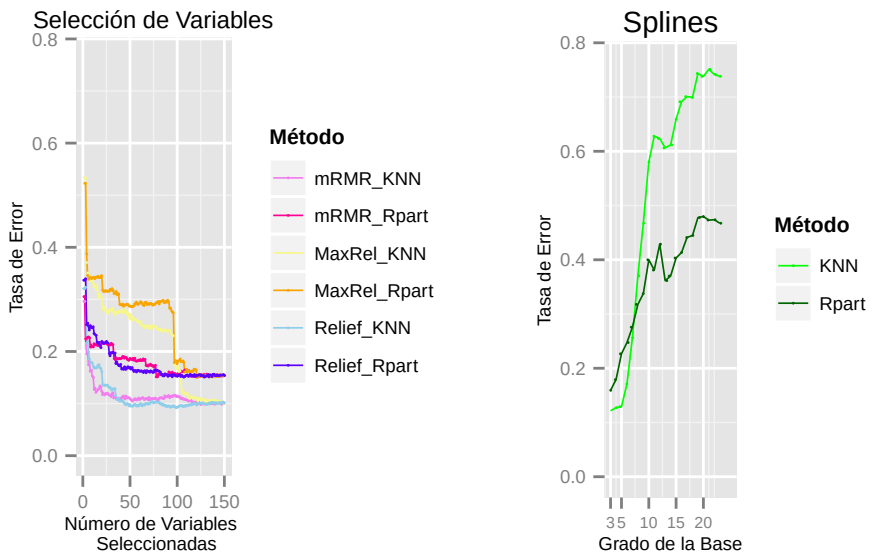

Figura 5–27: Tasa de Error de Mala Clasificación para Phoneme

Lo anterior confirma el hecho de que en muchas investigaciones que involucran el estudio y aplicación de B-Splines o Splines en general, se haga uso de Splines cúbicos y se garantice un buen suavizamiento si se trabaja con Splines de grados 4 o 5.

Método de Reducción de Dimensionalidad		k-NN		rpart	
		Sin CV	Con CV	Sin CV	Con CV
B-Splines	Nro de Variables	k=3:6	k=3:6	k=3:6	k=3:6
	Tasa Error	7 %-8 %	11 %-12 %	14 %-16 %	15 %-17 %
mRMR	Nro de Variables	3 a 5	3 a 5	3 a 5	3 a 5
	Tasa Error	14 %-16 %	19 %-21 %	19 %-20 %	20 %-22 %
Relief	Nro de Variables	7 a 10	7 a 10	7 a 10	7 a 10
	Tasa Error	11 %-13 %	17 %-18 %	21 %-22 %	24 %
Completa	Nro de Variables	150	150	150	150
	Tasa Error	6.8 %	10 %	14 %	16 %

Cuadro 5–7: Tasa de Error de Mala Clasificación para los datos Phoneme

Capítulo 6
CONCLUSIONES Y TRABAJOS FUTUROS

Los aspectos fundamentales estudiados en el presente trabajo, fueron la reducción de la dimensionalidad y la clasificación supervisada, ambos aplicados sobre datos funcionales. Mediante la combinación de éstos se encontró que algunos métodos de reducción en estudio generan menor tasa de error de mala clasificación en datos funcionales que cuando se usan todas las variables. Esto hace que nuestro objetivo principal se cumpla.

A continuación se presentan las conclusiones del trabajo realizado y seguidamente los trabajos futuros.

6.1. CONCLUSIONES

1. En el caso de la reducción de la dimensionalidad se observó que los métodos multivariados de selección de variables: Relief, mRMR y MaxRel analizados en nuestro estudio, trabajan muy bien con datos funcionales, tomándolos como vectores de alta dimensión. De los tres métodos, se vió que Relief y mRMR seleccionan un número similar de variables.

   Por otro lado, de acuerdo a las características del método mRMR se concluye que éste es más exigente en el proceso iterativo de selección, al elegir variables poco correlacionadas entre sí, lo cual lo hace altamente superior con respecto a MaxRel que evalúa sólo la relevancia sin considerar la redundancia entre las variables.

2. En segundo lugar se debe resaltar que aunque como se indicó anteriormente, el desempeño de los métodos multivariados de selección de variables es bueno, éstos son superados por los B-Splines, que por su naturaleza, se adaptan con mayor facilidad a los datos funcionales reduciendo considerablemente su dimensión, el cual es uno de los principales problemas al que nos enfrentamos al estudiar este tipo de datos.
3. Finalmente, con respecto a la clasificación, los resultados nos mostraron nuevamente la superioridad de los B-Splines frente a los 3 métodos de selección, pues en todos los casos la tasa de error de mala clasificación asociada a los B-Splines fue menor con dimensión pequeña, los resultados óptimos se obtuvieron al considerar bases con grados entre 5 y 8. Los B-Splines son seguidos en desempeño por el método Relief, que demostró obtener buenos resultados de clasificación trabajando con el conjunto de variables iniciales reducido entre el 12 y 16 %. Por su parte, los métodos mRMR y MaxRel se posicionan en el tercer y cuarto lugar respectivamente; y de los dos, mRMR muestra mejores resultados.

6.2. TRABAJOS FUTUROS

- Aplicar otros suavizadores a datos funcionales en sustitución de B-Splines, usar Wavelets podría ser una buena opción.
- Adaptar otros métodos de selección de variables para datos de clasificación a datos funcionales.
- Evaluar el efecto de reducción de la dimensionalidad aplicada a datos funcionales con otros clasificadores tales como regresión logística.
- Analizar otros conjuntos de datos funcionales, para ver si los resultados experimentales obtenidos en este trabajo se siguen sosteniendo.
- Utilizar simulación para explorar la validación de los métodos descritos.

APENDICES

Apéndice A
REDUCCION DE LA DIMENSIONALIDAD

A.1. Algoritmos para Seleccionar Variables

A.1.1. Función de selección de variables usando RELIEF

```
1
2  ###### Funcion de seleccion de variables usando la libreria CORElearn ######
3  ############################## y el metodo RELIEF ##############################
4
5  #Cargar la libreria CORElearn:
6  library(CORElearn)
7
8  #Leer la data.
9
10 #Usar attrEval y Relief para seleccionar el mejor subconjunto de variables:
11 selec_relief<-function(data,j){
12 COREdata=attrEval(formula=Class~.,data=data,estimator="Relief")
13 CORE_index=order(COREdata,decreasing=TRUE)
14 CORE_index=order(COREdata,decreasing=TRUE)[1:j]
15 CORE_index=CORE_index+1
16 CORE_index=c(1,CORE_index)
17 data_selec_Relief=data[,CORE_index]     #Data con las j variables seleccionadas
18 return(data_selec_Relief)
19 }
```

A.1.2. Función de selección de variables usando mRMR y MaxRel

```
1
2  ######### Extrayendo de la data, el conjunto de variables #########
3  ########### seleccionadas por los metodos mRMR y MaxRel ###########
4
5  #Leer la data.
6
7  #Asignar el subconjunto de variables seleccionadas a la data:
8
9  Selec_Var<-function(data,data_index,index){
10 var=data_index$Fea[1:index]
11 var=var+1
12 var=c(1,var)
13 dataselec=data[,var]
14 return(dataselec)
15 }
16
17 #Leer la data_index que contiene las ''j'' variables seleccionadas por mRMR o MaxRel.
18 funcion_index<-function(data,index,metodo,n){
19 data_index
20 dataselec_metodo=Selec_Var(data,data_index,index)
21 return(dataselec_metodo)
22 }
```

A.2. Algoritmos para Reducir Dimensionalidad

A.2.1. Función que encuentra los B-Splines

```
1
2  ######## Funcion de reduccion de la dimensionalidad ########
3  ############## usando B-Splines con grado ''k'' ##############
4
5  #Cargar la libreria splines:
6  library(splines)
7
8  #Leer la data.
9  #Definiendo las variables:
10 curvas=as.matrix(data[,2:dim(data)[2]])
11 #Definir la variable ''x''(continua) con la que se crearan las bases.
12
13 Bspline<-function(data,longitudesonda,k){
14 x=as.numeric(longitudesonda[,1])
15 curvas=as.matrix(data[,2:dim(data)[2]])
16 coeficientesk=matrix(0,dim(curvas)[1],k+1)#Matriz de 215*(k+1)
17 sumresidcuadk=matrix(0,dim(curvas)[1],1)  #Matriz de n*1
18 yhatk=matrix(0,dim(curvas)[1],length(x))  #Matriz de 215*100
19 for(i in 1:dim(curvas)[1]) {
20     y=curvas[i,]
21     base=bs(x,degree=k)
22     z=lm(y~base)
23     yhatk[i,]=z$fit
24     sumresidcuadk[i]=sum((z$residuals)^2)
25     coeficientesk[i,]=z$coeff
26                                }
27     data_Bspline=matrix(0,dim(coeficientesk)[1],dim(coeficientesk)[2]+1)
28     colnames(data_Bspline)=c("Class","Beta0","Beta1","Beta2","Beta3","Beta4",
29                 "Beta5","Beta6","Beta7",...,"Betak")
30     data_Bspline[,1]=as.matrix(data[,1])
31     data_Bspline[,2:(dim(coeficientesk)[2]+1)]=coeficientesk
32     lista<-list("coeficientes_de_los_Bspline"=coeficientesk,
33                 "valores_estimados_con_el_spline"=yhatk,
34                 "suma_de_residuales"=sumresidcuadk,"data_Bspline"=data_Bspline,
35                 "El promedio de la suma de los residuales
36 ~~~~~~~~~~~~~~~~~cuadrados usando splines de grado k es:"= mean(sumresidcuadk))
37     return(lista)
```

A.2.2. Función que determina el grado de los B-Splines que ajustan mejor el modelo

```
1
2  #### Determinando el grado de los B-splines que ajustan mejor el modelo ####
3  ################## (Prueba para k entre 3 y 20) ##################
4
5  medias=c(1:18)
6  x=as.numeric(longitudesonda[,1])
7  maxpar=20
8  coeficientesk=array(0,c(dim(curvas)[1],maxpar+1,dim(curvas)[1]))#Matriz de 215*(k+1)
9  sumresidcuadk=matrix(0,dim(curvas)[1],maxpar-2)                 #Matriz de 215*1
10          yhatk=matrix(0,dim(curvas)[1],length(x))               #Matriz de 215*100
11
12 for(k in 3:maxpar){
13 for(i in 1:dim(curvas)[1]){
14     y=curvas[i,]
15     base=bs(x,degree=k)
16     z=lm(y~base)
17     yhatk[i,]=z$fit
18     sumresidcuadk[i,k-2]=sum((z$residuals)^2)
19     coeficientesk[i,1:(k+1),i]=z$coeff
20     medias[k-2]=mean(sumresidcuadk[,k-2])
21                                 }
22                   }
23     print(sumresidcuadk)
24     print (medias)
25     print(coeficientesk)
26     print(yhatk)
```

A.3. Algoritmo para encontrar la derivada de n-ésimo orden

A.3.1. Función que calcula la derivada de la data Funcional

```
1
2  #### Funcion que calcula la derivada de los datos funcionales ####
3
4  #Cargar las siguientes librerias:
5  library(splines)
6  library(Matrix)
7  library(MASS)
8  library(nlme)
9  library(mgcv)
10 library(fda)
11 library(fda.usc)
12
13 #Leer la data:
14 data
15
16 #Creando la funcion que calcula la n-esima derivada de la data:
17 data_diferenciada=function(data,n){
18 data_dif1=data
19 data_dif1[,1]=data[,1]
20 data_dif1[,2:101]=fdata.deriv(data[,2:101],nderiv=n,method="bspline")$data
21 return(data_dif1)
22 }
```

Apéndice B
CLASIFICACIÓN SUPERVISADA

B.1. Algoritmos de Clasificación Supervisada

B.1.1. Función de Clasificación usando R

```
1
2  ######## Funcion de clasificacion de variables usando Knn y Arboles #########
3
4  #Cargar las librerias:
5  library(splines)
6  library(Matrix)
7  library(MASS)
8  library(nlme)
9  library(mgcv)
10 library(fda)
11 library(fda.usc)
12
13 #Leer la data.
14
15 clasificacion<-function(data,clasificador=c("rpart","knn"),cv){
16
17 #---- Usando rpart ----#
18 #----------------------#
19 if(clasificador=="rpart"){
20
21 #Aplicando el Metodo de Arboles
22 arbol=rpart(Class~.,data,method="class")
23 plot(arbol,margin=.02)
24 text(arbol,use.n=T)
25
26 #Encontrando las nuevas clases con rpart sin validacion cruzada
27 if(cv==F){
28 prediccion=predict(arbol,data,type="class")
29 tabla_clasificacion=table(data$Class,prediccion)
30 dimnames(tabla_clasificacion)=list(Clase_actual=c("0", "1"),
31                          "Nueva_Clase_rpart_sin_cv"=c("0","1"))
32 print("Tabla de Clasificacion rpart sin validacion cruzada")
33 print(tabla_clasificacion)
34
35 #Comparar las clases nuevas con las clases iniciales
36 errores_arbol=sum(prediccion!=data$Class)
37 cat("El numero de datos mal clasificados usando arboles es:",errores_arbol, "\n")
38
39 #Tasa de error de mala clasificacion
40 cat("La tasa de error de mala clasificacion usando
41 rpart sin cv es:", errores_arbol/215)
42    }
43 #Encontrando las nuevas clases con rpart usando validacion cruzada
44 if(cv==T){
45 cvprediccion=xpred.rpart(arbol,xval=10)
46 ncol=dim(cvprediccion)[2]
47
```

```
#Tasa de error global de mala clasificacion
error_cvprediccion=mean((cvprediccion-1)!=as.numeric(data$Class))
cat("La tasa de error global usando rpart con cv: ", error_cvprediccion,"\n")

#Tasa de error de mala clasificacion por columnas errores_cvcolumnas=c(0,ncol)
for(i in 1:ncol){
        errores_cvcolumnas[i]=sum(data$Class!=(cvprediccion[,i]-1))
                }
cat("El numero de datos mal clasificados de xpred es:", errores_cvcolumnas,"\n");
cat("El # promedio de datos mal clasificados de xpred es:", mean(errores_cvcolumnas))
cat("Equivalente a una tasa de error de:", mean(errores_cvcolumnas/215))
                    }
                                                    }
#---- Usando knn ----#
#--------------------#
if(clasificador=="knn"){
#Encontrando las nuevas clases con rpart sin validacion cruzada
if(cv==F){
prediccion=knn(data[,-1],data[,-1],data[,1],k=5)
tabla_clasificacion=table(data$Class,prediccion)
dimnames(tabla_clasificacion)=list(Clase_actual=c("0", "1"),
"Nueva_Clase_knn_sin_cv"=c("0","1"))
print("Tabla de Clasificacion knn sin validacion cruzada")
print(tabla_clasificacion)

#Comparar las clases nuevas con las clases iniciales
errores_knn=sum(prediccion!=data$Class)
cat("El numero de datos mal clasificados usando knn sin cv es:",errores_knn, "\n")

#Tasa de error de mala clasificacion por resustitucion
error_prediccion=mean(knn(data[,-1],data[,-1],data[,1],k=5)!=data[,1])
cat("La tasa de error de mala clasificacion usando knn sin cv es: ",error_prediccion)
       }

#Encontrando las nuevas clases con knn y validacion cruzada

if(cv==T){
cvprediccion=knn.cv(data[,-1],data[,1],k=5)
tabla_clasificacion=table(data$Class,cvprediccion)
dimnames(tabla_clasificacion)=list(Clase_actual=c("0", "1"),
"Nueva_Clase_knn_con_cv"=c("0","1"))
print("Tabla de Clasificacion knn con cv")
print(tabla_clasificacion)

#Comparar las clases nuevas con las clases iniciales
errores_cvknn=sum(cvprediccion!=data$Class)
cat("El numero de datos mal clasificados de knn
con cv es:",errores_cvknn, "\n")

#Tasa de error de mala clasificacion por resustitucion
error_prediccion=mean(cvprediccion!=data[,1])
cat("La tasa de error de usando knn sin validacion cruzada es: ",error_prediccion)
          }
                                                }
}
```

Bibliografía

[1] Ramsay J.O. y Silverman B.W. *Functional Data Analysis*. Springer-Verlag, second edition, 2005.

[2] Valderrama M.J. An overview to modelling functional data. *Computational Statistics*, 22:331–334, 2007.

[3] Levitin D.J. Nuzzo R.L. Vines B.W. y Ramsay J.O. Introduction to functional data analysis. *Canadian Psychology*, 48(3):135–155, 2007.

[4] Navarro P.V. Análisis de datos funcionales, implementación y aplicaciones, Universidad Politécnica de Catalunya, Julio 2004.

[5] Fraiman R. Gimenez Y. y Svarc M. Feature selection for functional data. *arXiv:1502.02123v1 [stat.ME]*, 2015.

[6] Ferraty F. y Vieu P. Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1-2):161–173, 2003.

[7] Fraiman R. Justel A. y Svarc M. Selection of variables for cluster analysis and classification rules. *Journal of the American Statistical Association*, 103(483):1294–1303, 2008.

[8] Ferraty F. y Vieu P. *Nonparametric Functional Data Analysis Theory and Practice*. Springer-Verlag, first edition, 2006.

[9] Bonev B.I. *Feature selection based on Information Theory*. PhD thesis, University of Alicante, June 2010.

[10] Torrecilla N.J. Análisis de datos funcionales, clasificación y selección de variables. Master's thesis, Universidad Autónoma de Madrid, Septiembre 2010.

[11] Ding C. y Peng H. Minimum redundancy feature selection from microarray gene expression data. *Proc.2nd IEEE Computational Systems Bioinformatics*

Conference (CSB 2003), Stanford, CA, pages 523–528, Aug. 2003.

[12] Febrero M. A present overview on functional data analysis. *Boletín de Estadística e Investigación Operativa*, 24(1):06–12, 2008.

[13] Kira K. y Rendell L. The feature selection problem: Traditional methods and a new algorithm. *AAAI-92 Proceedings*, 1992.

[14] Kokonenko I. Estimating attributes: Analysis and extensions of relief. *Proceedings of the European Conference on Machine Learning (ECML94), Secaucus, NJ. USA*, 1994.

[15] Peng H. Long F. y Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[16] Peng H. Ding C. y Long F. Minimum redundancy maximum relevance feature selection. *IEEE Intelligent Systems*, 20(6):70–71, 2005.

[17] Peng H. y Ding C. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2):185–205, 2005.

[18] Silverman B.W. y Jones M.C. E. fix y j.l. hodges: An important contribution to nonparametric discriminant analysis and density estimation (commentary on fix and hodges (1951)). *International Statistical Review*, pages 233–247, 1989.

[19] Cover T. y Hart P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, (1):21–27, 1967.

[20] Quinlan J.R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, first edition, October,1992.

[21] Hjorth J.S. *Computer Intensive Statistical Methods: Validation model selection and bootstrap*. Chapman & Hall/CRC, first edition, 1994.

[22] Acuna E. Dprep; data preprocessing and visualization functions for classification r package 2.1.1. http:// academic.uprm.edu/eacuna/dprep2.1.zip, 2011.