# Análisis multivariante: soluciones eficientes e interpretables


Tesis Doctoral

**Sergio Muñoz Romero**

**Departamento de Teoría de la Señal y Comunicaciones**
**Escuela Politécnica Superior**
**Universidad Carlos III de Madrid**

**2015**

Este documento está preparado para ser impreso a doble cara.

# Análisis multivariante: soluciones eficientes e interpretables

**TESIS DOCTORAL**

Autor:
**Sergio Muñoz Romero**

Directores:
**Dra. Vanessa Gómez Verdejo**
**Dr. Jerónimo Arenas García**

**Departamento de Teoría de la Señal y Comunicaciones**
**Escuela Politécnica Superior**
**Universidad Carlos III de Madrid**

**2015**

**Tesis Doctoral**

# Análisis multivariante: soluciones eficientes e interpretables


Autor: Sergio Muñoz Romero

Directores: Dra. Vanessa Gómez Verdejo
Dr. Jerónimo Arenas García


El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores

Presidente: Dr. José Luis Rojo Álvarez

Vocal: Dr. Steven Van Vaerenbergh

Secretario: Dr. José Miguel Leiva Murillo

acuerda otorgarle la calificación de

Leganés, a de de

*A mi familia*

# Resumen

*En dos palabras puedo resumir cuánto he aprendido acerca de la vida: Sigue adelante.*

Robert Lee Frost

En la actualidad, existe una tendencia creciente de almacenar ingentes cantidades de datos con el fin de analizar y extraer algún tipo de información útil de ellos. Sin embargo, el tratamiento de los mismos no resulta trivial y la aplicación de métodos de análisis de datos puede sufrir multitud de problemas tales como sobreajuste o problemas de multicolinealidades causados por la existencia de variables altamente correladas. Por ello, una etapa previa de extracción de características que permita reducir la dimensionalidad de los datos y eliminar dichas multicolinealidades perjudiciales entre variables es crucial para poder aplicar de manera adecuada y eficiente dichas técnicas de análisis de datos. En particular, los métodos de análisis multivariante (MVA) –que permiten extraer un nuevo conjunto de características representativas del problema– gozan de amplia popularidad y han sido aplicados con éxito en una gran cantidad de aplicaciones del mundo real. No obstante, cuando el objetivo consiste en obtener conocimiento de los datos capturados, no solo se requieren buenas prestaciones del sistema diseñado, sino también la capacidad de producir soluciones interpretables que permitan una mejor comprensión del problema. Por lo tanto, resulta deseable modificar estos métodos MVA aportándoles una especialización de las necesidades del problema con el fin de obtener dicha interpretabilidad.

En esta tesis doctoral, se estudian en detalle los métodos MVA y se presenta un marco general que engloba a dichos métodos MVA –en particular, a aquellos que obtienen características ortogonales entre sí–. Este estudio en profundidad permite una extensión de dicho marco general que facilita la inclusión de restricciones adicionales con el fin de proporcionarles habilidades adicionales, como, por ejemplo, la deseada capacidad de interpretabilidad. Para demostrar la versatilidad de este marco, se proponen soluciones MVA especializadas a cuatro casos particulares que requieren una interpretación completamente distinta del problema: soluciones MVA dispersas en las ca-

racterísticas extraídas; soluciones MVA dispersas en características extraídas a partir de relaciones no lineales entre variables; soluciones MVA que permiten la selección de las variables relevantes; y soluciones MVA no negativas para el diseño supervisado de bancos de filtros. Aunque en la literatura se pueden encontrar algunas soluciones especializadas, aquí se demuestra tanto teórica como experimentalmente que presentan graves problemas tanto de inicialización como de concepto en términos de poder ser considerados auténticos métodos MVA. La validez de las propuestas presentadas en esta tesis doctoral es certificada mediante una serie de experimentos que hacen uso de datos obtenidos del mundo real.

# Abstract

*In three words I can sum up everything*
*I've learned about life: it goes on.*

Robert Lee Frost

Currently, there is a growing tendency to store large amounts of data to analyze and extract any useful information from them. However, treating them is not trivial and application of data analysis methods can suffer several problems such as overfitting or multicollinearity problems caused by the existence of highly correlated variables. Therefore, a preliminar feature extraction stage that reduces the dimensionality of the data and eliminates these harmful multicollinearities between variables is crucial to apply these techniques for data analysis in an appropriate and efficient way. In particular, multivariate analysis methods (MVA) –which allow to extract a new set of representative features of the problem– enjoy wide popularity and have been successfully applied in a large number of real-world applications. However, when the aim is to obtain knowledge of the captured data, and not just good performance of the designed system, the ability to produce interpretable solutions for a better understanding of the problem is required. Therefore, it is desirable to modify these MVA methods to provide them with specialization of problem needs to obtain such interpretability.

In this thesis, we study in detail MVA methods and we present a general framework that encompasses them –in particular, those who obtain orthogonal features–. This in-depth study allows an extension of the general framework that facilitates the inclusion of additional constraints in order to provide additional properties, for example, the desired interpretability. To demonstrate the versatility of this framework, MVA specialized solutions to four particular cases that require completely different interpretation of the problem are proposed: sparse MVA solutions in the extracted features; sparse MVA solutions in extracted features from nonlinear relationships among variables; MVA solutions that allow the selection of the relevant variables; and non-negative MVA solutions for supervised design of filter banks. Although some specialized solutions can be found in the literature, here it is proven both theoretically and experimentally that they suffer serious pro-

blems of initialization and concept in terms of being considered authentic MVA methods. The legitimacy of the presented proposals in this thesis is certified through a series of experiments that use real-world data.

# Índice

# Índice de figuras

# Índice de Tablas

# Parte I

# Conocimientos preliminares

En esta primera parte de la tesis, se pretende motivar al lector y proporcionarle los conceptos necesarios con el fin de facilitar la lectura de las propuestas presentadas en la Parte II. Esta parte contiene un capítulo de introducción donde se presentan los objetivos que motivaron el presente trabajo, seguido de una revisión de las distintas técnicas existentes hasta el momento que han hecho posible la concepción de este estudio.

# Capítulo 1

# Introducción

*Lo último que uno sabe es por donde empezar.*

Blaise Pascal (1623-1662)

**Resumen:** En este primer capítulo, se pretende motivar al lector a que continúe con la lectura, a que encuentre los problemas que hay abiertos en la actualidad y que, con la ayuda o inspiración de este escrito, pueda incluso alcanzar algún tipo de provecho. La primera parte de este capítulo hace justo eso, identificar, en primer lugar, problemas manifiestos en el presente causados por la creciente generación de datos, para después proponer un camino de actuación y sacar así algún beneficio de ello. En el segundo y último apartado, se hace un repaso del trabajo más destacable realizado hasta el momento y relacionado con las contribuciones de esta tesis doctoral.

## 1.1. Motivación

La motivación del trabajo realizado en esta tesis doctoral proviene fundamentalmente de las necesidades surgidas a causa de la creciente explosión de datos acontecida en estos últimos años. Dichas necesidades son principalmente dos: el aprovechamiento de la información contenida en los datos disponibles y el rápido tratamiento de los mismos.

En la actualidad, se está viviendo una revolución tecnológica en prácticamente todos los campos de la ingeniería. A consecuencia de esta creciente generación de innovadores productos y servicios, se está capturando y almacenando una cantidad ingente de datos con el fin de poder ser aprovechados en un futuro. Por desgracia, esta recolección de datos se está realizando de

manera indiscriminada, sin tener en cuenta si dichos datos pueden ser útiles o no. Cabe decir que, en la mayoría de los casos, la no exclusión de datos poco o nada aprovechables no viene dada por falta de tiempo, ahorro económico, vagancia o ignorancia, sino porque la utilidad de dichos datos es difícilmente predecible. Para su aprovechamiento, por lo tanto, se está haciendo totalmente necesario el uso de técnicas de aprendizaje automático o máquina ("Machine Learning", ML) que permitan producir un conocimiento útil e interpretable a partir de los datos disponibles.

Es aquí donde la motivación de este trabajo reside, pues los métodos presentados en esta tesis pertenecen al ámbito del aprendizaje automático y tienen como fin obtener información útil e interpretable a partir de datos disponibles para, por ejemplo, una posterior toma de decisiones.

## 1.2. Revisión del estado del arte

### 1.2.1. Aprendizaje supervisado: problemas de clasificación y regresión

Antes de comenzar con la revisión del estado del arte del análisis multivariante, resulta interesante poder discernir entre los dos escenarios más usados en el aprendizaje máquina: el aprendizaje supervisado y el no supervisado. La diferencia entre ambas aproximaciones radica en la naturaleza de los datos disponibles o en el uso que se haga de ellos.

Para facilitar la aclaración de esta diferencia, se va a centrar la exposición en el siguiente ejemplo. Supóngase el hipotético caso de querer cuidar con mucho mimo una planta y que se dispone de un conjunto de medidas tanto de la temperatura de la tierra como de la cantidad de agua que recibe, pero no hay forma de saber la humedad de la tierra ni, por lo tanto, de saber si se tiene que regar o no. La tarea de estimar la humedad mediante las medidas de temperatura y cantidad de agua es conocida como regresión, mientras que el problema de determinar si hay que echar agua o no se conoce como clasificación. En este caso, el escenario deseable para poder predecir tanto la humedad de la tierra como si hay que regar o no sería a partir de ejemplos previamente etiquetados; es decir, suponiendo que se consiguió pedir prestado un sensor de humedad de suelo durante un tiempo finito y que se pudieron tomar las tres medidas simultáneamente, las *etiquetas* serían esas mediciones de humedad deseadas. Con estas medidas, se dice que se dispone de un conjunto de datos etiquetados y el hecho de usar estas etiquetas para aprender a predecir —o, dicho de otro modo, para entrenar el regresor o el clasificador— se conoce como *aprendizaje supervisado*. Por el contrario, si no se dispone de estas etiquetas o no se quiere hacer uso de ellas, se dice que el aprendizaje se realiza de modo no supervisado. En este ejemplo, el aprendizaje no supervisado no sería sencillo, no solo porque la regresión no

supervisada no tiene sentido, sino porque la tarea de decidir si hay que regar o no sin un historial de cuando se debió hacerlo a la vista de la temperatura del suelo y del agua recibida no parece muy viable.

Por ultimo, esas *máquinas* entrenadas (regresor/clasificador) requieren recibir unos datos de entrada y producir unos datos de salida. En el ejemplo de arriba, los datos de entrada serían las medidas de temperatura y de cantidad de agua caída —en este caso, se dice que el conjunto de datos de entrada tiene dos dimensiones—, mientras que los datos de salida o *etiquetas estimadas* serían, o bien las estimaciones de la humedad que habría en la tierra (problema de regresión), o bien las decisiones tomadas sobre si se riega o no la planta (problema de clasificación).

### 1.2.2. Métodos MVA

En los últimos años, los métodos de análisis de datos están siendo cada vez más utilizados con el fin de automatizar la extracción de información relevante de los datos disponibles, siendo usados eficientemente cuando se manejan datos con una o pocas dimensiones. Sin embargo, cuando las herramientas de aprendizaje máquina se aplican a problemas del mundo real compuestos por observaciones de alta dimensionalidad (cientos o, incluso, miles de dimensiones), aparecen fácilmente problemas numéricos y de sobreajuste. En estos casos, una etapa previa de extracción de características, que permita reducir la dimensionalidad de los datos y eliminar multicolinealidades perjudiciales entre variables, es crucial para poder aplicar de manera adecuada y eficiente estas técnicas de análisis de datos. Por esta razón, las técnicas de extracción de características y, en particular, los métodos de análisis multivariante ("MultiVariate Analysis", MVA) (Mardia et al., 1980; Arenas-García et al., 2013) se han aplicado con éxito en muchas aplicaciones del aprendizaje máquina, tales como en ingeniería biomédica (van Gerven et al., 2012; Hansen, 2007), en teledetección (Arenas-García y Camps-Valls, 2008; Arenas-García y Petersen, 2009) o en quimiometría (Barker y Rayens, 2003), entre muchas otras.

El análisis multivariante (MVA) aglutina una familia de métodos cuyo objetivo es extraer un nuevo conjunto de características representativas del problema mediante la proyección de variables en los datos de entrada y, en ocasiones, de salida. Los algoritmos más conocidos de estos métodos son el Análisis de Componentes Principales ("Principal Component Analysis", PCA) propuesto por Pearson (1901b), las aproximaciones de mínimos cuadrados parciales ("Partial Least Squares", PLS) introducidas por Wold (1966a,b) y el Análisis de Correlaciones Canónicas ("Canonical Correlation Analysis", CCA) presentado por Hotelling (1936). El algoritmo PCA crea un nuevo espacio de representación de datos mediante la búsqueda de las direcciones de mayor varianza de los datos de entrada, proporcionando un conjunto óptimo de características en términos de error cuadrático medio

("Mean Squared Error", MSE) de reconstrucción. A diferencia de otros métodos MVA, PCA trabaja de manera no supervisada, es decir, sólo tiene en cuenta los datos de entrada y no tiene presente las posibles etiquetas disponibles de las observaciones. El enfoque de las aproximaciones PLS, en su forma general, reside en proyectar tanto las variables de entrada como de salida a un nuevo espacio, generando un conjunto de características conocidas como variables latentes. El criterio utilizado para extraer estas variables latentes varía en función del esquema empleado, pero el propósito general consiste en maximizar la covarianza de los dos espacios proyectados. En CCA el objetivo es encontrar las proyecciones lineales de los datos de entrada y salida que maximicen la correlación entre los conjuntos de datos proyectados. Por tanto, en contraste con PLS, CCA explica la correlación en lugar de la covarianza, y esto hace del CCA un caso especial de PLS con sus propias características (véase Wegelin, 2000, para mayor detalle).

En esta tesis doctoral, se prestará especial atención a un cuarto método MVA conocido como PLS ortonormalizado ("Orthonormalized Partial Least Squares", OPLS) designado así por Worsley et al. (1996) y también denominado en la literatura como "semipenalized CCA" (Barker y Rayens, 2003), "multilinear regression" (MLR) (Borga et al., 1997) o "reduced-rank regression" (RRR) (Reinsel y Velu, 1998). El OPLS es conocido por ser óptimo en el sentido de MSE en problemas de regresión multilineal (Roweis y Brody, 1999; Arenas-García et al., 2007); por lo tanto, este método resulta muy competitivo como una etapa de pre-procesamiento en problemas de clasificación y regresión (Arenas-García y Camps-Valls, 2008; Arenas-García et al., 2007; Dhanjal et al., 2009). También existen varios estudios que han tratado de establecer las conexiones entre OPLS y otros métodos discriminatorios o MVA. Así, por ejemplo, destacan los trabajos de Reinsel y Velu (1998) y Sun et al. (2009) donde se demuestra que el OPLS y el CCA obtienen la misma solución en tareas de clasificación balanceadas (es decir, con clases equiprobables) si la matriz de etiquetas está codificada de manera binaria; o, también, el trabajo realizado por De la Torre (2012) donde se propone un marco generalizado para el análisis de componentes, aunque no facilita ni la inclusión de restricciones ni las soluciones eficientes de los algoritmos englobados.

### 1.2.3. Métodos no lineales

A pesar de la variedad de métodos MVA descritos anteriormente, todos ellos tratan con proyecciones lineales, impidiéndoles explotar las posibles relaciones no lineales existentes entre las variables originales. Para abordar esta cuestión, varios autores han propuesto variantes núcleo o *kernel* (Schoelkopf y Smola, 2002; Shawe-Taylor y Cristianini, 2004) donde los datos de entrada y/o salida son mapeados en un espacio de alta dimensionalidad mediante una función no lineal. De este modo, se posibilita la aplicación de los métodos

MVA lineales sobre estos datos transformados. La mayoría de los métodos MVA han sido reformulados en un marco kernel, dando lugar a aproximaciones como el kernel PCA de Scholkopf et al. (1998), el kernel CCA de Lai y Fyfe (2000), el kernel PLS de Rosipal y Trejo (2002) y el kernel OPLS de Arenas-García et al. (2007). La principal ventaja de estas extensiones núcleo se basa en la flexibilidad proporcionada por las expresiones no lineales mientras se sigue resolviendo un problema formulado únicamente con ecuaciones lineales. Debido a esto, los métodos kernel MVA (KMVA) han sido aplicados en una amplia variedad de campos que se caracterizan por sus relaciones no lineales, incluyendo el análisis de datos de teledetección (Arenas-García y Camps-Valls, 2008; Arenas-García y Petersen, 2009), resonancias magnéticas funcionales (fMRI) (Hardoon et al., 2007; Eklund et al., 2012), reconocimiento de expresiones faciales (Zheng et al., 2006) o agrupación de datos genómicos (Yamanishi et al., 2003) entre otros. Sin embargo, como aspecto negativo, la formulación directa de los métodos kernel MVA escala de manera cuadrática con el número de datos de entrenamiento, haciéndolos inviables (o por lo menos poco prácticos) para aquellos conjuntos de datos que contienen unos pocos de miles de patrones. Además, a menos que se regularicen de manera apropiada, estos métodos pueden sobreajustar fácilmente a los datos de entrenamiento (Shawe-Taylor y Cristianini, 2004; Arenas-García et al., 2013).

## 1.3. Problemas abiertos

En este sección, se mencionan algunos de los problemas más importantes que permanecen abiertos en el campo de los métodos MVA y que serían objeto de las aportaciones contenidas en esta tesis doctoral.

### 1.3.1. MVA con dispersión

Aunque las técnicas MVA permiten reducir la dimensionalidad de los datos —facilitando así su manejo en casos de alta dimensionalidad cuando se presentan variables irrelevantes, ruidosas o redundantes—, las proyecciones obtenidas son el resultado de una combinación de todos los elementos originales, incluyendo incluso variables no informativas. Este comportamiento llega a ser, a menudo, bastante nocivo, como se expresa en el principio conocido como *bet-on-sparsity* (Friedman et al., 2004), siendo deseable una solución compuesta únicamente de las características más relevantes o informativas. De esta manera, no sólo se obtendrían, por lo general, soluciones más precisas, sino también más interpretables.

La selección de características se realiza habitualmente como una etapa de procesamiento previo al problema de aprendizaje (Liu y Motoda, 1998; Guyon y Elisseeff, 2003; Guyon et al., 2006). Las técnicas de selección de

características clásicas, tales como los filtros, analizan la utilidad de cada variable mediante algún criterio de relevancia completamente independiente de la tarea posterior a resolver. Los métodos basados en validaciones cruzadas, conocidas como *Wrappers* (Kohavi y John, 1997), también analizan la relevancia de cada característica, pero en este caso se usa como criterio la precisión proporcionada por una herramienta de aprendizaje máquina que resuelve el problema final. Otros métodos de selección de características más recientes, conocidos como integrados ("embedded"), tratan de incrementar su eficiencia combinando el proceso de selección de características con el entrenamiento del predictor final (Weston et al., 2001; Guyon et al., 2002; Weston et al., 2003; Rakotomamonjy, 2003).

En los últimos años, una de las maneras más populares para realizar la selección de características —clasificable dentro de los métodos integrados— es favoreciendo directamente soluciones dispersas que asignan automáticamente coeficientes nulos a las variables que son irrelevantes para la tarea. Por este motivo, desde que Tibshirani (1994) propuso el método *lasso* (LASSO, "Least Absolute Shrinkage and Selection Operator") como una forma de inducir dispersión mediante la inclusión de un término de con la norma $\ell_1$, muchos investigadores han centrado sus trabajos en el uso de esta norma u otras con propiedades equivalentes. De hecho, la facilidad de esta técnica para eliminar características irrelevantes no solo ha provocado su aplicación a problemas de clasificación y regresión (Bi et al., 2003; Xiang y Ramadge, 2012; Dyar et al., 2012), sino que también ha permitido extensiones dispersas de técnicas MVA, tales como los métodos PCA y CCA dispersos de Zou et al. (2006) y Hardoon y Shawe-Taylor (2011), respectivamente. Los autores van Gerven et al. (2012) propusieron también un OPLS disperso, pero desafortunadamente este método no garantiza la ortogonalidad de los datos de entrada proyectados y, como consecuencia, la convergencia a la solución estándar OPLS no está asegurada cuando se eliminan las restricciones de dispersión. En esta tesis doctoral, se demostrará la existencia de estos problemas, se solventarán y se recomendará la elusión de este tipo de aproximaciones que actualmente están siendo usadas por defecto.

Para poder explotar las posibles relaciones no lineales existentes entre las variables y, al mismo tiempo, contrarrestar las propiedades nada deseadas de los métodos KMVA vistas anteriormente, se han propuesto varios métodos KMVA dispersos (véanse, por ejemplo, Hoegaerts et al., 2004; M. Momma, 2003; Arenas-García et al., 2007; Dhanjal et al., 2009). Nótese que cuando se hace referencia a métodos KMVA dispersos, por lo general, se asume selección de muestras en lugar de selección de variables.

En esta tesis doctoral, se aborda el tema de la dispersión en el algoritmo OPLS tanto lineal como kernel. Para llevar a cabo esto, se recurrirá a una formulación OPLS alternativa que simplifica la resolución del problema mediante un problema de autovalores estándar (EVD). Esta formulación que aquí

será denominada como EVD-OPLS, es bien conocida como RRR ("reduced-rank regression") en la comunidad estadística (Reinsel y Velu, 1998), pero no ha sido del mismo modo aplicada en el campo del aprendizaje máquina. La formulación presentada aquí abre la puerta a versiones modificadas de OPLS que imponen restricciones adicionales sobre los vectores de proyección, un hecho que será explotado en los Capítulos 4 y 5 para implementar versiones OPLS dispersas tanto en el marco lineal como no lineal.

### 1.3.2. MVA para selección de variables

En la actualidad, se está requiriendo una dispersión no solo por cada coeficiente individual, sino en la variable completa, de modo que se puedan seleccionar únicamente aquellas variables relevantes presentes en los datos disponibles. Este objetivo está siendo cada vez más perseguido principalmente por el hábito, cada vez más extendido, de capturar y almacenar indiscriminadamente colecciones inmensas de datos para encontrar patrones ocultos que ayuden a tomar decisiones o, incluso, ponerse a la cabeza de algún mercado competitivo. Este paradigma es generalmente conocido como "Big Data" y, aunque el nombre puede llevar a confusiones, el número de observaciones no tiene que ser necesariamente elevado, pudiendo ser alto, por ejemplo, el número de variables de cada observación. Esto podría ocurrir, por ejemplo, en una red de sensores que toma mediciones de una gran variedad de factores. En este último caso, sería deseable detectar únicamente aquellas variables que pueden ser útiles para una determinada tarea. Para tal fin, se podría forzar dispersión para cada variable por separado sobre todos los datos de entrenamiento (conocido como solución parsimoniosa); de este modo, se discriminaría solamente aquellas variables de entrada que son útiles para una tarea en particular. Este tipo de dispersión se podría conseguir incluyendo un término de regularización *group lasso*, propuesto por Yuan y Lin (2006). Aunque esta regularización ha sido también incorporada al OPLS por Chen y Huang (2012), la técnica de group lasso requiere de información a priori para conseguir esta distinción y, además, es muy costosa computacionalmente, siendo inviable en este tipo de soluciones. Otro término de regularización que sí obtiene soluciones parsimoniosas y ha sido eficientemente implementado por Nie et al. (2010) es la norma $\ell_{2,1}$. Shi et al. (2014) proponen incorporar esta solución a métodos MVA; sin embargo, sufren de los mismos problemas que van Gerven y Heskes (2010) y Chen y Huang (2012) para forzar dispersión en la solución. En el Capítulo 6, se explorarán diversas maneras de imponer este tipo de restricciones en los métodos MVA y se confirmarán los problemas ocasionados por la solución propuesta por Shi et al. (2014).

### 1.3.3. MVA con restricciones de no negatividad

Como ya se ha mencionado, en esta tesis doctoral se propondrán extensiones MVA que favorezcan la interpretación de las características extraídas. En particular, cuando se manejan señales espectrales o de energía, se deberían imponer restricciones de no negatividad sobre los vectores de proyección, de manera que las características extraídas puedan ser interpretadas como la energía contenida en una determinada banda de frecuencias ecualizada y los propios vectores de proyección puedan ser vistos como un tipo de banco de filtros. Esta interpretación es útil, por ejemplo, en las aplicaciones que tratan con señales de audio o imágenes, donde el procesamiento de estos datos se lleva a cabo generalmente en el dominio de la frecuencia.

En la literatura reciente sobre aprendizaje máquina, se pueden encontrar otros algoritmos que preservan la no negatividad de la solución. Uno de los algoritmos más populares es la factorización no negativa de matrices ("Non-Negative Matrix Factorization", NMF) introducido por Lee y Seung (1999), que ha sido aplicada, por ejemplo, para separación de fuentes por Virtanen (2007), para transcripción de música por Smaragdis y Brown (2003) o para el análisis espectral de datos por Pauca et al. (2006), entre otros. Otro enfoque quizá menos explorado consiste en incorporar una restricción de no negatividad en la solución de los métodos MVA. Por ejemplo, el algoritmo PCA no negativo ha sido aplicado para la separación ciega de fuente positiva por Oja y Plumbley (2003) o para el análisis de datos metabolómicos por Deng et al. (2012); el PLS no negativo (NPLS) ha sido utilizado para la comprensión de Resonancias Magnéticas Nucleares (RMN) de datos espectroscópicos por Allen et al. (2013); el CCA no negativo (NCCA) ha sido usado para separación de fuentes audiovisuales por Sigg et al. (2007); y el algoritmo OPLS con restricción de positividad (POPLS) ha sido propuesto para clasificación de instrumentos musicales y reconocimiento de género musical por Arenas-García et al. (2006).

A diferencia de las aproximaciones NMF, una ventaja adicional de incorporar restricciones de no negatividad en los métodos MVA es la capacidad de obtener soluciones dispersas e, indirectamente, una selección automática de características. Esta preferencia por la dispersión ha motivado que en los últimos años muchos métodos incorporen términos de regularización $\ell_0$ y $\ell_1$ en sus formulaciones. Sin embargo, a diferencia de los métodos que se consideran en este trabajo, ni la regularización $\ell_0$ ni la $\ell_1$ fuerzan soluciones no negativas por ellas mismas.

Como ya se ha mencionado, un objetivo muy interesante y que se propondrá en este tesis doctoral es el diseño de bancos de filtros que proporcionan características interpretables en problemas supervisados. En la Figura 1.1, se ilustra el proceso completo para la extracción de estas características cuando se trata con imágenes, compuesto principalmente de tres bloques bien diferenciados: 1) una etapa de pre-procesamiento que convierte los datos en

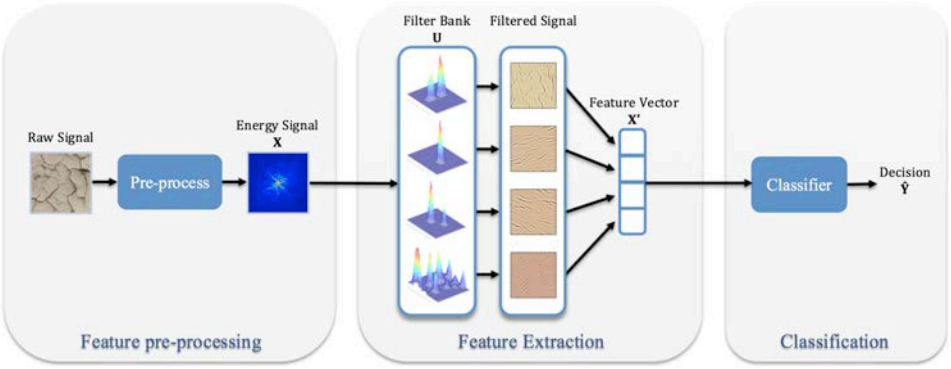



Figura 1.1: Esquema completo de una tarea de reconocimiento de texturas desde de la imagen en bruto hasta la decisión final. En primer lugar, se procesa la imagen para obtener una representación en frecuencia en dos dimensiones (2-D) para pasar posteriormente a través del banco de filtros, de modo que cada característica extraída resume la energía contenida en un cierto rango de frecuencias. Finalmente, la clasificación se realiza en base a las características extraídas.

bruto en una representación de los datos mejor ajustada para el dominio de la frecuencia (véase el apartado 7.1 para mayor detalle); 2) una etapa de extracción de características donde la señal pasa a través de un banco de filtros y, como resultado, se obtiene un vector de características ($\boldsymbol{x}'$ en la Figura 1.1), siendo cada una de sus componentes la energía de la imagen en un cierto rango ecualizado de frecuencias; y 3) una etapa de clasificación donde se usa el vector de características para clasificar la clase asociada a la imagen.

En la mayoría de los trabajos previos, basados en sistemas similares al representado en la Figura 1.1, el único bloque que está diseñado de manera supervisada es el del clasificador, mientras que el banco de filtros está típicamente construido sin ninguna información etiquetada. En lugar de ello, se utiliza una batería suficientemente rica de filtros de propósito general (por ejemplo, los filtros de Gabor propuestos por Turner (1986); Fogel y Sagi (1989)) o se explota el conocimiento experto de la materia en cuestión. Por ello, resulta deseable el uso de etiquetas en esta fase para obtener un diseño supervisado de dichos filtros.

Entre una gran cantidad de tareas visuales, la clasificación de imágenes por texturas es una aplicación interesante que necesita incorporar una etapa de extracción de características. Resulta sorprendente que los métodos con restricciones de no negatividad mencionados anteriormente aún no se hayan aplicado aquí, tal vez debido al amplio y exitoso uso de procedimientos ad hoc de extracción de características. Una de las técnicas más adoptadas es el Filtrado de Gabor ("Gabor Filtering", GF) que fue propuesto para la

clasificación de texturas por Turner (1986) y Fogel y Sagi (1989) y todavía se usa o incluso se ha mejorado su eficiencia (véase Bianconi y Fernández, 2007; Li et al., 2010) para la clasificación invariante a escala y a rotación de texturas (véase Han y Ma, 2007; Bianconi et al., 2008). El Patrón Binario Local ("Local Binary Pattern", LBP) también es una técnica exitosa para la clasificación de texturas (Ojala et al., 2002; Guo et al., 2010), pero no proporciona ningún tipo de interpretación a la solución.

En cuanto a las aplicaciones de clasificación de música basadas en audio (Fu et al., 2011) y, en particular, al campo de recuperación de información musical ("Music Information Retrieval", MIR), la clasificación de género musical ha sido un área de investigación bastante activa en los últimos años. A pesar de la gran variedad de diferentes enfoques para resolver este problema, la extracción de características es un escenario habitual en estas soluciones (Scaringella et al., 2006) y el uso de representaciones dispersas ha sido sugerido en los últimos años como una forma de mejorar las prestaciones (Sturm, 2013; Chen y Ramadge, 2013). Sin embargo, las características dispersas no proporcionan ningún tipo de interpretabilidad a la solución, que es una propiedad deseable para comprender la estructura de la música. Para proporcionar esta capacidad, se pueden imponer restricciones adicionales de no negatividad, como propusieron McKinney y Breebaart (2003) y Arenas-García et al. (2006).

## 1.4. Contribuciones de la tesis doctoral

Las principales contribuciones de esta tesis doctoral, ordenadas por capítulos, son:

- Capítulo 2.— La comparación de la formulación EVD (o RRR) del OPLS (junto con la explicacion de sus ventajas) frente a la formulación basada en un problema de autovalores generalizado (GEV), que es más habitual en el campo del aprendizaje máquina (véanse, por ejemplo, Arenas-García y Camps-Valls, 2008; De la Torre, 2012; Arenas-García et al., 2007; Huang y De la Torre, 2010). Se discutirá la equivalencia entre ambas soluciones y se demostrará que cuando el número de variables de salida es menor que la dimensionalidad de los datos de entrada, la formulación EVD es más eficiente en términos computacionales.

- Capítulo 3.— Un marco generalizado para los métodos MVA con el fin de poder incorporar fácilmente cualquier tipo de restricción sobre la solución obtenida. Aunque en la literatura se pueden encontrar algunos intentos de imponer diversas restricciones a las soluciones MVA (Zou et al., 2006; van Gerven y Heskes, 2010; Chen y Huang, 2012; Shi et al., 2014), todas ellas se basan, por defecto, en la solución ortogonal de Procrustes (Schönemann, 1966). En este trabajo y con el fin de

evitar su uso generalizado, se localizan y se demuestran los problemas ocasionados por el uso de Procrustes en esquemas iterativos: problemas tales como el problema de convergencia —el algoritmo podría no progresar en absoluto si se elimina el término de regularización— o el incumplimiento de la condición de ortogonalidad de las características extraídas.

- Capítulo 4.— Una extensión dispersa del OPLS lineal basada en la formulación EVD y en un término de regularización $\ell_1$. Aunque existen intentos de utilizar EVD para obtener soluciones OPLS dispersas (van Gerven y Heskes, 2010; Chen y Huang, 2012), estos están basados en la solución de Procrustes. El estudio comparativo entre estos esquemas se hará sobre un conjunto de problemas de clasificación de referencia y una tarea de reconocimiento de caras, analizando la precisión y el grado de dispersión obtenido en la solución.

- Capítulo 5.— Una extensión de la solución EVD al marco no lineal o kernel. Esta aproximación propuesta para kernel OPLS disperso es, hasta donde llega nuestro conocimiento, totalmente novedosa, encontrándose propuestas previas con dispersión $\ell_1$ para OPLS únicamente en el espacio de entrada original. Al igual que en el caso lineal, se analiza el poder de discriminación de las características extraídas y el grado de dispersión alcanzado por estos nuevos métodos mediante un conjunto de problemas de clasificación de referencia.

- Capítulo 6.— Una extensión del marco MVA propuesto anteriormente para la selección de variables de entrada, obteniendo así soluciones parsimoniosas. Para ello, se explorarán soluciones MVA parsimoniosas imponiendo dispersión a cada variable por separado, proponiendo así un marco MVA que proporcione la capacidad de seleccionar las variables relevantes y extraer sus características de manera eficiente. Esta es una aplicación muy deseada actualmente para detectar la parte relevante de los datos que están siendo almacenados de manera masiva en el ámbito del "Big Data".

- Capítulo 7.— Un conjunto de métodos que permiten diseñar de manera supervisada y automática bancos de filtros para aplicaciones que tratan con datos espectrales o de energía. Para ello, se incorporará una restricción de no negatividad en la solución OPLS lineal. Las prestaciones de las distintas aproximaciones obtenidas serán probadas sobre dos aplicaciones reales completamente distintas, que son: el reconocimiento de texturas y la clasificación de género musical. Estas propuestas serán comparadas frente a los bancos de filtros ad hoc habitualmente usados en estas aplicaciones.

Con respecto a su estructura, esta tesis doctoral está dividida en tres partes bien diferenciadas:

- Parte I.— Conocimientos Preliminares. En esta primera parte, además de motivar al lector sobre el trabajo realizado, se pretende introducir los conceptos necesarios para el seguimiento de esta tesis doctoral. Además, puesto que esta información se encuentra dispersa o, incluso, perdida en la literatura, se pretende unir todo este conocimiento en un mismo documento, pudiendo ser de este modo una lectura de referencia en el estado del arte de los métodos MVA. Esta primera parte consta de los Capítulos 1 y 2.

- Parte II.— Propuesta Doctoral. Esta segunda parte constituye el grueso de esta tesis doctoral, pues se describen las distintas propuestas de la misma; está compuesta por los Capítulos 3, 4, 5, 6 y 7, cada uno de los cuales presenta una propuesta nueva, y un sexto capítulo adicional, Capítulo 8, donde se exponen las principales conclusiones de este trabajo.

- Apéndices.— En esta tercera y última parte de la tesis doctoral, se incluye material adicional o de apoyo de las partes anteriores, como puede ser la demostración de algún resultado. No obstante, para la comprensión de esta tesis doctoral, no resulta necesaria la lectura de esta última parte complementaria.

# Capítulo 2

# Revisión de conceptos MVA

*Si has construido castillos en el aire, tu trabajo no se pierde; ahora coloca las bases debajo de ellos.*

Henry David Thoreau (1817-1862)

**Resumen:** En el presente capítulo, se pretende ofrecer al lector una visión clara de los métodos MVA. Para ello, y tras revisar algunos conceptos básicos necesarios, se describen los métodos MVA más importantes, así como las diferentes soluciones más comúnmente usadas en la literatura hasta el momento.

## 2.1. Conceptos básicos

Antes de describir los métodos de análisis multivariante más usados en la literatura, se revisarán algunos conceptos básicos del álgebra lineal con el fin de facilitar la exposición del resto de esta tesis doctoral. Los aspectos necesarios para este fin son principalmente tres: 1) el concepto de proyección ortogonal, 2) los métodos de descomposición de matrices en autovalores y valores singulares y 3) la deflacción de matrices. Pero antes de eso, se revisará brevemente la notación que se usará de aquí en adelante.

Dado que los conceptos aquí revisados son básicos, todo lector ya familiarizado con estos términos podría obviar esta parte y saltar a la página 30 para continuar con la lectura de esta tesis doctoral a partir del Apartado 2.2. No obstante lo anterior, cabe decir que este apartado se ha hecho con el propósito de unir en un mismo documento todo el conocimiento necesario para el buen entendimiento de los métodos MVA, pues en la actualidad esta información se encuentra parte dispersa y parte perdida en la literatura, pu-

diendo ser, por lo tanto, una contribución interesante para dicha literatura MVA.

### 2.1.1. Notación

Asumiendo un escenario de aprendizaje supervisado, donde el objetivo es aprender características relevantes de los datos de entrada, se usará un conjunto de $N$ datos de entrenamiendo $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, para $i = 1, \ldots, N$, donde $\boldsymbol{x}_i \in \mathbb{R}^{n\times 1}$ e $\boldsymbol{y}_i \in \mathbb{R}^{m\times 1}$ son considerados como los vectores de entrada y salida, respectivamente. De esta manera, $n$ y $m$ denotan las dimensiones de los espacios de entrada y salida. En problemas de clasificación, $\boldsymbol{y}_i$ será usado para indicar la pertenencia a la clase de la $i$-ésima muestra, por ejemplo, usando una codificación "1-de-$C$" (Bishop, 1995). Por conveniencia notacional, se definen las matrices de entrada y de salida como: $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ e $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]$. Se va a asumir a lo largo de esta tesis doctoral que estas matrices están centradas para eliminar cualquier correlación entre variables producidas por un desplazamiento de sus centros de masas (Shawe-Taylor y Cristianini, 2004). Las estimaciones muestrales de las matrices de covarianza de los datos de entrada y de salida, así como las de sus matrices de covarianza cruzada, pueden ser calculadas como $\mathbf{C_{XX}} = \mathbf{X}\mathbf{X}^\top$, $\mathbf{C_{YY}} = \mathbf{Y}\mathbf{Y}^\top$ y $\mathbf{C_{XY}} = \mathbf{X}\mathbf{Y}^\top$, donde se ha despreciado el factor de escala $\frac{1}{N}$, y el superíndice $^\top$ denota la transpuesta de un vector o de una matriz.

Puesto que en esta tesis doctoral, se trabaja en un escenario multivariante, resulta interesante repasar el concepto de Operador Norma y la notación correspondiente que se usará de aquí en adelante, así como los distintos tipos existentes de este operador que serán usados en las diversas propuestas presentadas en la Parte II de este documento.

#### 2.1.1.1. Operador Norma

A partir de una matriz $\mathbf{A} \in \mathbb{R}^{n\times m}$, se denotará su fila $i$-ésima como $\boldsymbol{a}^i$, su columna $j$-ésima como $\boldsymbol{a}_j$ y el elemento de la fila $i$ y columna $j$ como $A_{ij}$.

Se define la norma $\ell_p$ de un vector $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$, para $p \in (0, \infty)$, como

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}.$$

Los valores más comunes de $p$ son probablemente $p = 1$ y $p = 2$:

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^\top\boldsymbol{x}$$

y los casos extremos que no están incluidos en la norma anterior son:

$$\|\boldsymbol{x}\|_0 = \sum_{i=1}^{n} |x_i|^0 = \#\{i|x_i \neq 0\}, \qquad \|\boldsymbol{x}\|_\infty = \max_i |x_i|,$$

donde $\#\{i|x_i \neq 0\}$ significa que la norma $\ell_0$ devuelve el número de elementos no nulos del vector y $\max_i |x_i|$ denota que la norma $\ell_\infty$ devuelve la magnitud más alta de entre todos los elementos del vector. Por conveniencia, cuando el operador norma de un vector no lleva el subíndice $p$, se hará referencia a la norma $\ell_2$ ($\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2$). Además, el operador norma cumple las siguientes tres condiciones:

1. Condición de no negatividad: $\|\boldsymbol{x}\| \geq 0$, siendo $\|\boldsymbol{x}\| = 0 \Leftrightarrow \boldsymbol{x} = \mathbf{0}$.
2. Condición de escalabilidad: $\|c\boldsymbol{x}\| = c\|\boldsymbol{x}\|, \qquad c \in \mathbb{R}$.
3. Desigualdad Triangular: $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$.

Una vez descrita la notación para la norma de un vector, se va a proceder del mismo modo con la norma de una matriz, $\|\mathbf{A}\|$. En este caso, se pueden describir las normas $\ell_1$ y $\ell_\infty$ como

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^{n} |A_{ij}|, \qquad \|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{m} |A_{ij}|,$$

mientras que la norma $\ell_2$, $\|\mathbf{A}\|_2^2$, corresponde con el máximo autovalor de $\mathbf{A}^\top\mathbf{A}$; pero posiblemente una de las normas más frecuentemente utilizadas cuando se manejan matrices es la norma de Frobenius:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^2} = \left(\sum_{i=1}^{n} \|\boldsymbol{a}^i\|_2^2\right)^{\frac{1}{2}} = \left(\mathrm{Tr}\{\mathbf{A}\mathbf{A}^\top\}\right)^{\frac{1}{2}}.$$

Una norma que también cumple con las tres condiciones mencionadas anteriormente es la norma $\ell_{r,p}$ descrita por Nie et al. (2010) como

$$\|\mathbf{A}\|_{r,p} = \left(\sum_{i=1}^{n} \|\boldsymbol{a}^i\|_r^p\right)^{\frac{1}{p}}.$$

Nótese que la norma de Frobenius es un caso particular de esta norma, siendo $\|\mathbf{A}\|_F = \|\mathbf{A}\|_{2,2}$.

Otro caso particular de esta norma y que se usará en el Capítulo 6 para seleccionar características es la norma $\ell_{2,1}$ que fue introducida por Ding et al. (2006) para solventar la carencia de invarianza rotacional que sufre la norma $\ell_1$,

$$\|\mathbf{A}\|_{2,1} = \sum_{i=1}^{n} \|\boldsymbol{a}^i\|_2. \tag{2.1}$$

De este modo, la norma $\ell_{2,1}$ tiene la propiedad de ser invariante rotacional por filas, es decir, que dada una matriz de rotación[1] cualquiera $\mathbf{R}$, se cumple

[1] Una matriz de rotación es una matriz ortogonal con determinante 1, es decir que cumple las siguientes condiciones: $\mathbf{R}^\top = \mathbf{R}^{-1}$ (es decir, $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$) y $\det(\mathbf{R}) = 1$.

que

$$\|\mathbf{AR}\| = \|\mathbf{A}\|.$$

Nótese que la norma $\ell_0$ no es una norma válida, ya que no cumple la condición de escalabilidad, es decir, $\|c\boldsymbol{x}\|_0 \neq c\|\boldsymbol{x}\|_0$, aplicándose aquí el término "norma" simplemente por conveniencia. Además, aunque el uso de las normas $\ell_0$ y $\ell_{r,0}$ en problemas de optimización son las más deseadas en multitud de ocasiones, ya que devuelve el número de elementos no nulos (o filas completas no nulas en el segundo caso) del vector o matriz, no es una opción viable al tratarse de un problema *NP-hard* (es decir, demasiado complejo para poder ser resuelto matemáticamente). Por lo tanto, esta solución suele ser relajada, o bien a una norma $\ell_p$ con $0 < p < 1$, o bien a la norma $\ell_1$. Esta última opción tiende a ser elegida debido a que es la primera de las normas convexas (es decir, para $p \geq 1$) y, aunque aún no presenta una formulación suave para ser derivable en la solución —como es el caso de la norma $\ell_2$— ahora sí es viable (aunque costosa computacionalmente) gracias a la llegada de la así llamada *optimización convexa*.

### 2.1.2. Proyección ortogonal

Formalmente, se dice que $\mathcal{P}$ es una *proyección ortogonal* de un espacio vectorial $\mathcal{H}$ si es una *transformación lineal* idempotente ($\mathcal{P}^2 = \mathcal{P}$) y autoadjunta ($\langle \boldsymbol{x}, \mathcal{P}\boldsymbol{y}\rangle = \langle \mathcal{P}\boldsymbol{x}, \boldsymbol{y}\rangle \quad \forall \boldsymbol{x},\ \boldsymbol{y} \in \mathcal{H}$, siendo $\langle \cdot, \cdot \rangle$ el producto interno definido en el espacio de Hilbert).

Puesto que esta definición requiere de un conocimiento alto de terminología matemática, a continuación se pretende explicar esta transformación a través de un simple ejemplo.

Supóngase que se quiere obtener la mejor aproximación posible de un espacio $\mathcal{S}(\mathbf{X})$ definido por unos datos disponibles de entrada $\mathbf{X}$ a otros datos de salida $\mathbf{Y}$ también disponibles[2]. Una posible solución sería obtener la combinación lineal de los datos de entrada (transformación lineal) que menos distancia —o menos error de aproximación ($\boldsymbol{e}$)— presente con $\mathbf{Y}$. Este error se puede expresar matemáticamente como

$$\boldsymbol{e} = \boldsymbol{y} - \mathbf{X}\boldsymbol{w} \tag{2.2}$$

siendo $\boldsymbol{w} \in \mathbb{R}^{n\times 1}$ un vector columna con los pesos de la transformación lineal requerida. Sabiendo que la menor distancia entre un punto y un plano es cuando la recta que los separa es ortogonal a dicho plano, entonces se puede definir

$$\mathcal{P}_{\mathbf{X}}(\boldsymbol{y}) = \boldsymbol{z} = \mathbf{X}\boldsymbol{w}$$

como la *proyección ortogonal* de $\boldsymbol{y}$ sobre $\mathcal{S}(\mathbf{X})$. Un ejemplo de dicha proyección ortogonal se puede ver en la Figura 2.1.

[2]El espacio $\mathcal{S}(\mathbf{X})$ está formado por todos los vectores que pueden obtenerse como combinación lineal de las columnas de $\mathbf{X}$.

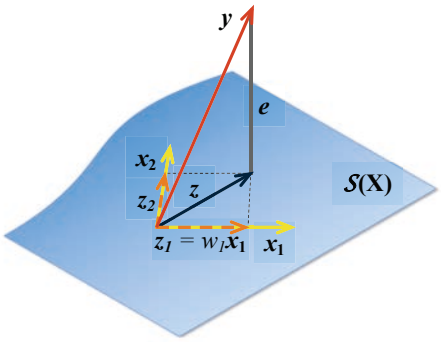



Figura 2.1: Proyección ortogonal de $\boldsymbol{y}$ sobre el espacio definido por $\mathbf{X}$, $\mathcal{S}(\mathbf{X})$.

Como también se sabe que cuando dos vectores son ortogonales su producto interno es cero, entonces se puede conseguir dicha aproximación haciendo que el error sea ortogonal a $\mathcal{S}(\mathbf{X})$, que a su vez se consigue garantizando que el error de aproximación sea ortogonal a todos los vectores del espacio $\mathcal{S}(\mathbf{X})$ y, en particular, a las columnas de la matriz $\mathbf{X}$, es decir, $\mathbf{X}^\top \boldsymbol{e} = \mathbf{0}$. Por lo tanto, si se multiplica por la izquierda a ambos lados de la ecuación (2.2) por $\mathbf{X}^\top$ y se fuerza que $\mathcal{S}(\mathbf{X})$ y $\boldsymbol{e}$ sean ortogonales,

$$\mathbf{X}^\top (\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) = \mathbf{X}^\top \boldsymbol{e} = \mathbf{0},$$

se puede obtener el vector de pesos necesario para dicha transformación lineal —que llamamos proyección ortogonal— despejando como

$$\boldsymbol{w}^* = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{y}.$$

Ahora, sustituyendo esta solución en la ecuación de la proyección ortogonal de $\boldsymbol{y}$ sobre $\mathcal{S}(\mathbf{X})$,

$$\boldsymbol{z} = \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{y},$$

se obtiene que

$$\mathbf{P}_\mathbf{X} = \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \tag{2.3}$$

es la *matriz de proyección ortogonal* sobre $\mathcal{S}(\mathbf{X})$. De este modo, la proyección ortogonal de $\boldsymbol{y}$ sobre $\mathcal{S}(\mathbf{X})$ estaría dada por el producto entre $\boldsymbol{y}$ y $\mathbf{P}_\mathbf{X}$,

$$\mathcal{P}_\mathbf{X}(\boldsymbol{y}) = \boldsymbol{z} = \mathbf{P}_\mathbf{X}\, \boldsymbol{y}.$$

Por lo tanto, el error de aproximación entre los datos de entrada y de salida, cuyo objetivo era encontrar el mínimo error posible, se puede reescribir como:

$$\boldsymbol{e} = \boldsymbol{y} - \mathbf{P}_\mathbf{X}\, \boldsymbol{y}. \tag{2.4}$$

Esta forma de interpretar la proyección ortogonal como una transformación lineal será la usada a lo largo de la presente tesis doctoral.

Además, si la matriz de correlación de $\mathbf{X}$ es una *matriz blanqueada* —es decir, las variables de $\mathbf{X}$ están incorreladas ($\mathbf{X}^\top\mathbf{X} = \mathbf{I}$)— o si es una *matriz ortogonal* —es decir, es una matriz cuadrada cuya inversa es igual a su transpuesta ($\mathbf{X}^\top = \mathbf{X}^{-1}$) formando una base ortonormal donde todos sus vectores son unitarios(con norma unidad) y ortogonales entre sí—, la matriz de proyección ortogonal podría reducirse a $\mathbf{P_X} = \mathbf{X}\mathbf{X}^\top$.

Una propiedad de los operadores de proyección, $\mathcal{P}$, es que son idempotentes, es decir, que si este operador se ejecuta varias veces consecutivas el resultado sería el mismo que si se realizase una única vez:

$$\mathcal{P}_\mathbf{X}\left[\mathcal{P}_\mathbf{X}(\boldsymbol{y})\right] = \mathcal{P}_\mathbf{X}(\boldsymbol{y}),$$

ya que

$$\begin{aligned}
\mathcal{P}_\mathbf{X}\left[\mathcal{P}_\mathbf{X}(\boldsymbol{y})\right] &= \mathbf{P_X}\mathbf{P_X}\ \boldsymbol{y} \\
&= \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\ \boldsymbol{y} \\
&= \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\ \boldsymbol{y} \\
&= \mathbf{P_X}\ \boldsymbol{y} \\
&= \mathcal{P}_\mathbf{X}(\boldsymbol{y}).
\end{aligned}$$

Para terminar, cabe comentar que todo vector $\boldsymbol{y}$ puede descomponerse de forma única como

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{z}^\perp,$$

donde $\boldsymbol{z} \in \mathcal{S}(\mathbf{X})$ y $\boldsymbol{z}^\perp \in \mathcal{S}^\perp(\mathbf{X})$, siendo $\boldsymbol{z}^\perp$ y $\mathcal{S}^\perp(\mathbf{X})$ los complementos ortogonales de $\boldsymbol{z}$ y $\mathcal{S}(\mathbf{X})$ respectivamente (un ejemplo gráfico puede verse en la Figura 2.2). El *complemento ortogonal* de la proyección de $\boldsymbol{y}$ sobre $\mathcal{S}(\mathbf{X})$ puede definirse como

$$\begin{aligned}
\mathcal{P}^\perp_\mathbf{X}(\boldsymbol{y}) = \boldsymbol{z}^\perp &= \boldsymbol{y} - \mathbf{P_X}\ \boldsymbol{y} \\
&= (\mathbf{I} - \mathbf{P_X})\boldsymbol{y}. \qquad (2.5)
\end{aligned}$$

Para verificar esto, se puede demostrar que $\boldsymbol{z}^\perp$ es ortogonal a $\mathbf{X}$ (es decir, a $\mathcal{S}(\mathbf{X})$) de la siguiente manera:

$$\begin{aligned}
\mathbf{X}^\top\boldsymbol{z}^\perp &= \mathbf{X}^\top(\boldsymbol{y} - \mathbf{P_X}\ \boldsymbol{y}) \\
&= \mathbf{X}^\top\boldsymbol{y} - \mathbf{X}^\top\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\boldsymbol{y} \\
&= \mathbf{X}^\top\boldsymbol{y} - \mathbf{X}^\top\boldsymbol{y} \\
&= \mathbf{0}.
\end{aligned}$$

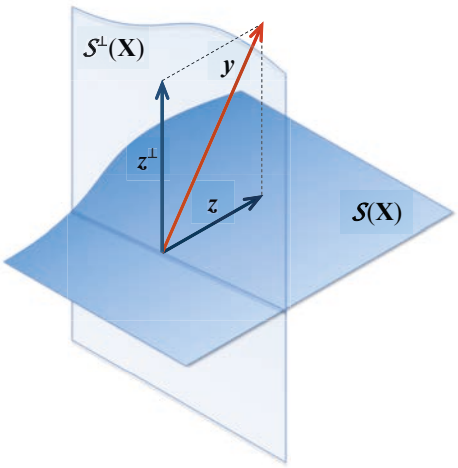



Figura 2.2: Descomposición única del vector $\boldsymbol{y}$ mediante su proyección $\boldsymbol{z}$ y su complemento ortogonal $\boldsymbol{z}^{\perp}$

Cabe destacar la interpretación que se le da merecidamente al complemento ortogonal de la proyección, pues tiene un papel importante tanto en los métodos MVA como en esta tesis doctoral. Para ello, resulta interesante volver al ejemplo anterior sobre la búsqueda de la mínima distancia o error de aproximación entre $\mathbf{X}$ e $\mathbf{Y}$. Si se comparan las ecuaciones (2.4) y (2.5), se ve que dicho error es justo el complemento ortogonal de $\mathbf{Y}$ sobre $\mathcal{S}(\mathbf{X})$ ($\boldsymbol{z}^{\perp}$), como se puede comprobar visualmente en la Figura 2.2.

El proceso de ir eliminando estas proyecciones de una determinada matriz sobre diversos vectores es conocido como *deflacción*, pero antes de profundizar en este tipo de técnicas sería preferible hablar sobre autovectores y autovalores, también conocidos como vectores propios y valores propios o como *eigen*vectores y *eigen*valores.

### 2.1.3. Autovectores y autovalores

La descomposición de una matriz simétrica en autovectores y autovalores tiene muchas aplicaciones en la vida real y, en concreto para esta tesis doctoral, representa la piedra angular de los algoritmos desarrollados. Por lo tanto, este apartado es de gran importancia para la correcta comprensión de las propuestas doctorales aquí expuestas.

A lo largo de toda esta tesis doctoral, el principal objetivo será encontrar aquellas matrices que proyecten los datos a un *espacio de características* de una manera eficiente con el fin de alcanzar el objetivo buscado (como la clasificación automática de los datos disponibles). Una propiedad deseable de dichas características o datos proyectados es que sean ortogonales entre sí

y ahí es donde entra el papel fundamental de este tipo de descomposiciones. En otras palabras, se buscarán matrices, por ejemplo $\mathbf{W}$, que minimicen una determinada función de coste, usualmente el error cuadrático medio ("mean squared error", MSE), y además cumplan la siguiente condición: $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$. La resolución directa suele llevar a difíciles problemas de optimización, pero su lagrangiano revela que el problema puede reformularse como un problema de autovalores fácilmente resoluble por muchas herramientas de computación y, además, otorgan a la solución propiedades de ortogonalidad muy deseables.

Este subapartado está dividido a su vez en tres partes, donde se revisa en cada una de ellas un método distinto para calcular autovalores y autovectores.

#### 2.1.3.1. Problema de autovalores estándar

Como en esta disertación se va a trabajar continuamente con el álgebra matricial y su formulación, así como con matrices de autocovarianza $\mathbf{C}$ que son simétricas, esta explicación va a comenzar a partir del teorema de diagonalización para matrices simétricas, que garantiza que toda matriz simétrica puede diagonalizarse ortogonalmente, es decir que dada una matriz simétrica $\mathbf{C} \in \mathbb{R}^{n\times n}$ existe una matriz diagonal $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ y otra ortogonal $\mathbf{W} \in \mathbb{R}^{n\times n}$ tal que

$$\mathbf{C} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}.$$

A partir de este resultado, se puede definir el siguiente resultado conocido también como el *problema de autovalores estándar* ("EigenValue Decomposition", EVD), escrito en formato matricial:

$$\mathbf{C}\mathbf{W} = \mathbf{W}\mathbf{\Lambda},$$

siendo $\mathbf{W}$ la matriz de autovectores o vectores propios (también conocidos como *eigen*vectores, vectores característicos, vectores latentes o polos) y $\mathbf{\Lambda}$ la matriz de autovalores o valores propios (también conocidos como *eigen*valores, valores característicos o raíces latentes).

La matriz de autovalores $\mathbf{\Lambda}$ solamente tiene valores distintos de cero en su diagonal, los autovalores, que están dispuestos en orden descendente, es decir, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. El conjunto de todos los autovalores es conocido como el *espectro* de la matriz y algunas de sus propiedades dada una matriz simétrica se enumeran en el Apéndice A:

Como ya se ha mencionado anteriormente, la matriz de autovectores $\mathbf{W}$ es ortogonal y, por lo tanto, cumple las siguientes propiedades:

$$\mathbf{W}^\top\mathbf{W} = \mathbf{I}, \qquad \mathbf{W}^\top = \mathbf{W}^{-1}.$$

Además, cada autovector $\boldsymbol{w}_k$ está asociado a su correspondiente autovalor $\lambda_k$ del siguiente modo:

$$\mathbf{C}\boldsymbol{w}_k = \lambda_k\boldsymbol{w}_k \qquad\qquad k = 1, 2, 3, \ldots, n.$$

Como extensión a lo anterior, $\boldsymbol{w}_k$ también estará asociado al autovalor $a\lambda_k$ de la matriz escalada $a\mathbf{C}$, al autovalor $\frac{1}{\lambda_k}$ de $\mathbf{C}^{-1}$ y al autovalor $\lambda_k^p$ de $\mathbf{C}^p$.

Existen diferentes métodos para resolver este problema de autovalores estándar. Uno de los más sencillos de implementar es el *método de las potencias* descrito en la Tabla 2.1. No obstante, el método más usado generalmente por su mayor precisión y fiabilidad, a pesar de su complejidad de implementación, es el método Lanczos (un análisis más detallado de estos métodos se puede encontrar en el libro de Golub y Van Loan, 2012). Como se puede observar en la Tabla 2.1, el método de las potencias permite calcular únicamente el primer autovector y autovalor principal; por lo tanto, si se quiere calcular el resto de autovectores y autovalores, habría que incluir un esquema de deflacción en la solución (para un estudio más detallado de estos métodos de deflacción véase el subapartado 2.1.4).

Tabla 2.1: Pseudocódigo del método de las potencias

1.- Entrada: Matriz a descomponer $\mathbf{C}$.
  2.1.- Inicializar $\boldsymbol{w}^{(0)} = \frac{\boldsymbol{c}_j}{\|\boldsymbol{c}_j\|_2}$ (siendo $\boldsymbol{c}_j$ cualquier columna de $\mathbf{C}$).
  2.2.- Para $i = 1, 2, \ldots$
    2.2.1.- $\boldsymbol{v} = \mathbf{C}\boldsymbol{w}^{(i-1)}$.
    2.2.2.- $\boldsymbol{w}^{(i)} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}$.
    2.2.3.- Si se cumple criterio de convergencia, ir a 3.
3.- Salidas: Autovector principal $\boldsymbol{w}$ y autovalor asociado $\lambda = \boldsymbol{w}^\top \mathbf{C}\boldsymbol{w}$.

#### 2.1.3.2. Problema de autovalores generalizado

En ocasiones, el problema a resolver para encontrar matrices ortogonales es un poco distinto. En concreto, aparece una matriz $\mathbf{B} \in \mathbb{R}^{n\times n}$ adicional en la formulación de la forma:

$$\mathbf{CW} = \mathbf{BW\Lambda},$$

conocido como *problema de autovalores generalizado* ("Generalized EigenValue decomposition", GEV), siendo, en este caso, $\mathbf{W}$ la matriz de autovectores generalizados y $\mathbf{\Lambda}$ la de autovalores generalizados, que deben obedecer el siguiente polinomio característico:

$$\det(\mathbf{C} - \lambda_k \mathbf{B}) = 0.$$

En este caso, si dicha matriz $\mathbf{B}$ fuese invertible —no singular—, este problema quedaría reducido al siguiente problema de autovalores estándar:

$$\mathbf{C}'\mathbf{W}' = \mathbf{W}'\mathbf{\Lambda},$$

siendo $\mathbf{C}' = \mathbf{B}^{-\frac{1}{2}}\mathbf{C}\mathbf{B}^{-\frac{1}{2}}$ una matriz simétrica y $\mathbf{W}' = \mathbf{B}^{\frac{1}{2}}\mathbf{W}$ la nueva matriz de autovectores (véase White, 1958, para una discusión más detallada sobre la obtención de autovectores y autovalores generalizados).

Nótese que, cuando $\mathbf{B}$ es no singular, se requeriría un método de regularización sobre la matriz $\mathbf{B}$, como, por ejemplo, calcular el problema de autovalores estándar de $\mathbf{B}$ y reconstruir $\mathbf{B}$ únicamente con los autovectores y autovalores regulares —es decir, no singulares: generalmente, donde los autovalores son suficientemente altos para considerarse que no tienden a cero y, por lo tanto, que es parte relevante o informativa de los datos)—.

Con respecto al coste computacional, tanto la inversión matricial como el problema de autovalores estándar tienen complejidad de orden cúbico, $\mathcal{O}(n^3)$, siendo $n$ las dimensiones de la matriz cuadrada. Por tanto, sería deseable evitar el problema de autovalores generalizado, pues conlleva la ejecución de dos operaciones de orden cúbico.

#### 2.1.3.3. Descomposición en valores singulares

Una de las factorizaciones de matrices más útiles y usadas en aplicaciones de la vida real es la *descomposición en valores singulares* ("Singular Value Decomposition", SVD):

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

donde las columnas de $\mathbf{U} \in \mathbb{R}^{n\times n}$ se conocen como los vectores singulares derechos de $\mathbf{X} \in \mathbb{R}^{n\times N}$, $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ contiene en su diagonal los valores singulares de $\mathbf{X}$ y $\mathbf{V} \in \mathbb{R}^{N\times n}$ contiene los vectores singulares izquierdos de $\mathbf{X}$.

La ventaja de esta descomposición frente al problema de autovalores estándar radica en la posibilidad de operar sobre matrices no cuadradas. No obstante, resulta importante aclarar que dichos vectores singulares izquierdos y derechos son los autovectores de la matriz de covarianza ($\mathbf{C_{XX}}$) y de la matriz de productos internos ($\mathbf{K}_x$) respectivamente:

$$\begin{aligned}\mathbf{C_{XX}} &= \mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \\ \mathbf{K}_x &= \mathbf{X}^\top\mathbf{X} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top.\end{aligned}$$

De aquí se concluye que $\mathbf{\Lambda} = \mathbf{\Sigma}^2$ y que los autovalores de $\mathbf{C_{XX}}$ y $\mathbf{K}_x$ son los mismos.

Volviendo a la ecuación de la matriz de proyección (2.3) y teniendo en cuenta que $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ y $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, se define como matriz de proyección sobre el espacio definido por las columnas de $\mathbf{X}$ (es decir, sobre las $n$ variables) a

$$\mathbf{P}_{\mathcal{S}_{\text{col}}(\mathbf{X})} = \mathbf{U}\mathbf{U}^\top, \tag{2.6}$$

y la matriz de proyección sobre el espacio definido por las filas de $\mathbf{X}$ (es decir, sobre los $N$ datos) a

$$\mathbf{P}_{\mathcal{S}_{\text{fil}}(\mathbf{X})} = \mathbf{V}\mathbf{V}^\top.$$

Estas dos últimas definiciones serán útiles para la siguiente disertación sobre el proceso de deflacción.

### 2.1.4. Deflacción

Formalmente, la deflacción podría definirse como aquella técnica consistente en anular secuencialmente la influencia del $k$-ésimo autovector de una matriz simétrica dada $\mathbf{C}$, reemplazando el correspondiente autovalor asociado $\lambda_k$ por 0; de este modo, el rango de la nueva matriz decrecería en una unidad.

No obstante, siguiendo el hilo del subapartado anterior, también se puede definir la deflacción como una herramienta para calcular la descomposición de una matriz en autovectores y autovalores de manera secuencial; es decir, una vez calculado el autovector principal de $\mathbf{C}$ —por ejemplo, con el método de las potencias resumido en la Tabla 2.1— se puede obtener el siguiente autovector principal de $\mathbf{C}$ tras aplicar el método de deflacción y, por lo tanto, tras haber eliminado la influencia de ese primer autovector calculado.

El hecho de calcular un problema EVD (o una SVD) de manera secuencial presenta diferentes ventajas, como son: la posibilidad de calcular los $k$ primeros autovectores más importantes de manera ordenada sin tener que calcular la totalidad de los autovectores de la matriz; el cálculo de los autovectores de manera más eficiente en memoria cuando la matriz $\mathbf{C}$ presenta un tamaño muy grande, permitiendo, incluso, su obtención en máquinas donde era inviable su cálculo en bloque por falta de memoria; o la eficiencia computacional en lenguajes de programación donde resulta más eficiente un esquema iterativo que uno que tenga que manejar matrices de tamaño elevado.

Además, con el fin de aportar una interpretación a la técnica de deflacción y, de este modo, poder explicar de una manera más sencilla el motivo de la técnica empleada, se usará la siguiente terminología:

- *Influencia de un autovector* en una matriz: determina el modo en que una matriz depende del espacio definido por ese autovector. La anulación de dicha influencia es posible reemplazando el correspondiente autovalor asociado por 0. Volviendo a la Figura 2.2, se podría decir que cualquier vector del espacio definido por $\mathbf{X}$ ($\mathcal{S}(\mathbf{X})$) no influiría en absoluto sobre cualquier vector proyectado en el espacio ortogonal a $\mathcal{S}(\mathbf{X})$, es decir, proyectado en $\mathcal{S}^{\perp}(\mathbf{X})$.

- *Varianza explicada* considerando que la matriz simétrica a descomponer ($\mathbf{C}$) es una matriz de autocovarianzas, se puede considerar el autovector asociado al mayor autovalor de $\mathbf{C}$ como aquella dirección que presenta —o "explica"— la mayor varianza de los datos, siendo el autovalor precisamente la varianza en esa dirección. De este modo, si se

obtiene un determinado número de direcciones ortogonales, la varianza total es igual a la suma de los correspondientes autovalores; de ahí que se hable de cantidad o porcentaje de varianza explicada o capturada por un determinado autovector con respecto a la varianza total. Por lo tanto, cuando se calcula de manera secuencial cada uno de estos autovectores, la técnica de deflacción se asegura de proyectar, en cada iteración, dichos datos en esa dirección de máxima varianza. Para ello, las direcciones obtenidas en las siguientes iteraciones deben ser ortogonales a las anteriores, evitando así proyectar los datos en direcciones paralelas a aquellas ya obtenidas —o, dicho de otro modo, evitando presentar direcciones donde haya varianza previamente explicada—. Como conclusión, cabe destacar que para calcular los siguientes autovectores de la matriz de manera secuencial hay que sustraer la varianza explicada por los autovectores anteriores —o, dicho con otras palabras, hay que anular la influencia de los autovectores anteriores sobre la matriz deflactada—.

Como aclaración, se dirá que se deflacta una matriz de autocovarianzas $\mathbf{C}$ —simétrica— cuando se esté calculando el problema de autovalores estándar de manera secuencial. En el caso de calcular una SVD de manera secuencial, se dirá entonces que se estará deflactando una matriz $\mathbf{A}$ (no simétrica). Cabe recordar que la matriz de vectores singulares izquierdos de $\mathbf{A}$ es la misma que la matriz de autovectores de la matriz de autocovarianzas de $\mathbf{A}$ (véase el subapartado 2.1.3.3 para más detalle).

Aunque existen distintos métodos de deflacción debido a los diferentes criterios seguidos para sustraer la varianza explicada por los autovectores ya calculados, todos ellos se pueden reducir a uno solo (deflacción de Hotelling) siempre y cuando las direcciones obtenidas sean autovectores propiamente dichos, es decir, que cumplan con todas las propiedades propias de dichas soluciones.

Sin embargo, cuando se incluye algún tipo de restricción sobre la solución (es decir, sobre los autovectores) —como se hará en la segunda parte de esta tesis doctoral—, las soluciones obtenidas dejarían de cumplir las propiedades fundamentales requeridas. De este modo, los distintos métodos de deflacción podrían discutirse en función de estas propiedades. En concreto, se discutirá sobre el cumplimiento de las siguientes tres propiedades fundamentales por parte de los autovectores obtenidos:

- Propiedad 1: $\mathbf{C}\mathbf{w}_j = \lambda_j \mathbf{w}_j$ (definición de autovector).
- Propiedad 2: $\mathbf{w}_j^\top \mathbf{w}_j = 1$ (vector unitario).
- Propiedad 3: $\mathbf{w}_i^\top \mathbf{w}_j = 0 \ \ \forall\, i \neq j$ (vectores ortogonales).

Cuando las direcciones obtenidas no cumplen con alguna de estas propiedades (debido a la inclusión de restricciones en el problema) se las suele

definir como *pseudo*-autovectores. Puesto que la Propiedad 1 es justamente la definición de autovector, como es lógico, todo vector que no la cumpla no podría ser considerado como autovector. En caso de aplicar una técnica de deflacción que requiera unos vectores que no cumplan con alguna de las propiedades arriba mencionadas, provocaría una re-introducción de componentes paralelas de los pseudo-autovectores previamente eliminados y la varianza ya explicada sería de nuevo tenida en cuenta. Para evitar esto, se tendría que tener en cuenta únicamente la varianza adicional explicada por el $k$-ésimo pseudo-autovector, que sería equivalente a la varianza explicada por $\mathcal{P}^{\perp}_{\mathbf{W}_{k-1}}(\boldsymbol{w}_k)$ (es decir, el complemento ortogonal de la proyección del $k$-ésimo pseudo-autovector $\boldsymbol{w}_k$ sobre el espacio definido por los anteriores pseudo-autovectores $\mathcal{S}(\mathbf{W}_{k-1})$):

$$\boldsymbol{z}_k^{\perp} = (\mathbf{I} - \mathbf{P}_{\mathbf{W}_{k-1}})\boldsymbol{w}_k,$$

donde $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}$ forman las columnas de $\mathbf{W}_{k-1}$. Por lo tanto, en cada paso de deflacción solo habría que eliminar la varianza asociada únicamente con $\boldsymbol{z}_k^{\perp}$.

A continuación se hará un repaso sobre los métodos de deflacción más conocidos y sus propiedades para operar sobre pseudo-autovectores (véase el artículo de Mackey, 2009, para un estudio más detallado de los métodos de deflacción con pseudo-autovectores).

#### 2.1.4.1. Deflacción de Hotelling

Este método de deflacción es uno de los más simples y usados. Para resolver de manera secuencial el problema de autovalores de una matriz simétrica dada $\mathbf{C}$ y, suponiendo que la matriz a deflactar inicial es semidefinida positiva (es decir, $\mathbf{C}_0 \succeq 0$ —véase la propiedad (i) del Apéndice A—), entonces, en la $k$-ésima iteración, se eliminaría el autovector principal de $\mathbf{C}_{k-1}$ como:

$$\mathbf{C}_k = \mathbf{C}_{k-1} - \boldsymbol{w}_k \boldsymbol{w}_k^{\top} \mathbf{C}_{k-1} \boldsymbol{w}_k \boldsymbol{w}_k^{\top}.$$

Con esto, se dice que si $\lambda_1 \geq \cdots \geq \lambda_n$ son los autovalores de $\mathbf{C}$ con autovectores asociados $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$, entonces la matriz deflactada en el instante $k$-ésimo, $\mathbf{C}_k$, tiene los siguientes autovalores: $0, 0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n$, manteniendo los mismos $n - k$ autovectores menos significativos de $\mathbf{C}$. Para demostrarlo, valdría simplemente con comprobar que la matriz $\mathbf{C}_k$ resultante de sustraer el $k$-ésimo autovalor de la matriz $\mathbf{C}_{k-1}$ es ortogonal únicamente

a $\boldsymbol{w}_k$:

$$\begin{aligned}
\mathbf{C}_k\boldsymbol{w}_k &= \mathbf{C}_{k-1}\boldsymbol{w}_k - \boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k\boldsymbol{w}_k^\top\boldsymbol{w}_k \overset{\text{Propiedad 2}}{\downarrow} \\
&= \mathbf{C}_{k-1}\boldsymbol{w}_k - \boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k \underset{\text{Propiedad 1}}{\uparrow} = \lambda_k\boldsymbol{w}_k - \lambda_k\boldsymbol{w}_k = 0, \\
\mathbf{C}_k\boldsymbol{w}_j &= \mathbf{C}_{k-1}\boldsymbol{w}_j - \boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k\boldsymbol{w}_k^\top\boldsymbol{w}_j = \\
&= \underset{\text{Propiedad 3}}{\uparrow}\ \mathbf{C}_{k-1}\boldsymbol{w}_j - 0 = \lambda_j\boldsymbol{w}_j, \quad \text{para} \quad j = 1, \ldots, k-1.
\end{aligned}$$

Sin embargo, esto solamente es cierto si se cumplen las tres propiedades de los autovectores comentadas anteriormente y, por consiguiente, no valdría para pseudo-autovectores. Una grave consecuencia de no verificarse la Propiedad 1 es que $\mathbf{C}_k$ podría dejar de ser semidefinida positiva, $\mathbf{C}_k \not\succeq 0$ (véase la propiedad (i) de los autovalores descrita en el Apéndice A), haciendo que en la práctica sea desaconsejable el uso de este esquema de deflacción cuando se manejan pseudo-autovectores.

#### 2.1.4.2. Deflacción por proyección

Este tipo de deflacción se suele usar para calcular secuencialmente los vectores singulares de una determinada matriz dada $\mathbf{A} \in \mathbb{R}^{n\times N}$ (típicamente no simétrica, con $n \neq N$); es decir, para calcular la SVD de $\mathbf{A}$ de manera secuencial.

El modo que tiene esta técnica de conseguir anular la influencia del $k$-ésimo vector singular obtenido, $\boldsymbol{w}_k \in \mathbb{R}^{n\times 1}$, sobre $\mathbf{A}_k$ —o de sustraer de $\mathbf{A}_k$ la varianza explicada por $\boldsymbol{w}_k$— sería proyectando las columnas de $\mathbf{A}_{k-1}$ sobre $\mathcal{S}^\perp(\boldsymbol{w}_k)$ o, en otras palabras, mediante el complemento ortogonal de la proyección de $\mathbf{A}_{k-1}$ sobre $\mathcal{S}(\boldsymbol{w}_k)$ ($\mathbf{A}_k = \mathcal{P}^\perp_{\boldsymbol{w}_k}(\mathbf{A}_{k-1})$, véase (2.5)):

$$\mathbf{A}_k = (\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top)\mathbf{A}_{k-1}.$$

En caso de querer calcular los autovectores (resolver el problema de autovalores estándar secuencialmente) de una matriz simétrica dada $\mathbf{C} \succeq 0$, como puede ser la matriz de covarianza de $\mathbf{A}$ ($\mathbf{C} = \mathbf{A}\mathbf{A}^\top$), el método de deflacción por proyección ("projection deflation") en la $k$-ésima iteración se formularía como:

$$\mathbf{C}_k = (\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top)\mathbf{C}_{k-1}(\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top),$$

donde sería fácil demostrar que si $\boldsymbol{w}_k$ cumple las tres propiedades descritas anteriormente, este método quedaría reducido al de Hotelling.

No obstante, se puede comprobar que la ortogonalidad entre $\mathbf{C}_k$ y $\boldsymbol{w}_k$ únicamente requiere la Propiedad 2,:

$$\mathbf{C}_k\boldsymbol{w}_k = (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^\top)\mathbf{C}_{k-1}(\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^\top)\boldsymbol{w}_k \overset{\text{Propiedad 2}}{\downarrow} (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^\top)\mathbf{C}_{k-1}(\boldsymbol{w}_k-\boldsymbol{w}_k) = 0,$$

y, por lo tanto, esta deflación sí es aplicable al trabajar con pseudo-autovectores de norma unitaria.

Uno de los problemas que tiene este método de deflacción es que no preserva la ortogonalidad requerida en las siguientes rondas del procedimiento secuencial; es decir, dado un vector $\boldsymbol{w}_j$ ortogonal a $\mathbf{C}_{k-1}$ para cualquier $k$ —por ejemplo, $\boldsymbol{w}_{k-1}$— ($\mathbf{C}_{k-1}\boldsymbol{w}_{k-1} = 0$), no se obtiene la ortogonalidad entre $\mathbf{C}_k$ y $\boldsymbol{w}_{k-1}$, ya que

$$\mathbf{C}_k\boldsymbol{w}_{k-1} = (\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top)\mathbf{C}_{k-1}(\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top)\boldsymbol{w}_{k-1} \neq 0.$$

#### 2.1.4.3. Deflacción por complemento de Schur

Al igual que en el caso anterior, esta última técnica de deflacción suele ser usada para calcular secuencialmente la SVD de una matriz dada $\mathbf{A} \in \mathbb{R}^{n\times N}$. Sin embargo, el modo que tiene esta técnica de eliminar de $\mathbf{A}_k$ la influencia del $k$-ésimo vector singular, $\boldsymbol{w}_k \in \mathbb{R}^{n\times 1}$, le hace preferible frente a las demás técnicas de deflacción, como se verá a continuación. En este caso, la varianza explicada por el $k$-ésimo vector singular es sustraída de $\mathbf{A}_k$ mediante la proyección de las filas de $\mathbf{A}_{k-1}$ sobre el complemento ortogonal del espacio definido por la *característica extraída* $\boldsymbol{z}_k = \mathbf{A}_{k-1}^\top\boldsymbol{w}_k$, es decir, sobre $\mathcal{S}^\perp(\boldsymbol{z}_k)$. Por lo tanto, la nueva matriz con el $k$-ésimo autovalor anulado se calcula como $\mathbf{A}_k = \mathcal{P}^\perp_{\boldsymbol{z}_k}(\mathbf{A}_{k-1}^\top) = \mathbf{A}_{k-1}(\mathbf{I} - \mathbf{P}_{\boldsymbol{z}_k})$, siendo $\mathbf{P}_{\boldsymbol{z}_k} = \frac{1}{\boldsymbol{z}_k^\top\boldsymbol{z}_k}\boldsymbol{z}_k\boldsymbol{z}_k^\top$ —como se definió en (2.3)— (véase (2.5) para más detalle); es decir, como:

$$\mathbf{A}_k = \mathbf{A}_{k-1}\left(\mathbf{I} - \frac{\mathbf{A}_{k-1}^\top\boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{A}_{k-1}}{\|\mathbf{A}_{k-1}^\top\boldsymbol{w}_k\|^2}\right). \tag{2.7}$$

En caso de querer calcular secuencialmente los autovectores de una matriz simétrica $\mathbf{C} \succeq 0$, como puede ser la matriz de covarianza de $\mathbf{A}$ ($\mathbf{C} = \mathbf{A}\mathbf{A}^\top$), el método de deflacción por complemento de Schur ("Schur complement deflation") en la iteración $k$-ésima es:

$$\begin{aligned}\mathbf{C}_k &= \mathbf{A}\left(\mathbf{I} - \frac{\mathbf{A}^\top\boldsymbol{w}\boldsymbol{w}^\top\mathbf{A}}{\|\mathbf{A}^\top\boldsymbol{w}\|^2}\right)\left(\mathbf{I} - \frac{\mathbf{A}^\top\boldsymbol{w}\boldsymbol{w}^\top\mathbf{A}}{\|\mathbf{A}^\top\boldsymbol{w}\|^2}\right)\mathbf{A}^\top \\ &= \mathbf{A}\left(\mathbf{I} - \frac{\mathbf{A}^\top\boldsymbol{w}\boldsymbol{w}^\top\mathbf{A}}{\|\mathbf{A}^\top\boldsymbol{w}\|^2}\right)\mathbf{A}^\top \\ &= \mathbf{C}_{k-1} - \frac{\mathbf{C}_{k-1}\boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}}{\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k},\end{aligned} \tag{2.8}$$

donde se han eliminado los subíndices en los pasos intermedios para simplificar la derivación. Se puede ver fácilmente que esta técnica se reduciría a la deflacción de Hotelling si $\boldsymbol{w}_k$ cumpliese todas las propiedades de autovector.

En este caso, se puede comprobar de manera sencilla que $\boldsymbol{w}_k$ es ortogonal a $\mathbf{C}_k$ (por ambos lados) sin que $\boldsymbol{w}_k$ tenga que cumplir las propiedades de autovector, puediendo ser, sin problema alguno, un pseudo-autovector:

$$\mathbf{C}_k\boldsymbol{w}_k = \mathbf{C}_{k-1}\boldsymbol{w}_k - \frac{\mathbf{C}_{k-1}\boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k}{\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k} = \mathbf{C}_{k-1}\boldsymbol{w}_k - \mathbf{C}_{k-1}\boldsymbol{w}_k = 0.$$

Este método de deflacción es único en el sentido en que sí preserva la ortogonalidad en las ulteriores iteraciones del procedimiento secuencial. Dicho de otro modo, dado un vector $\boldsymbol{w}_j$ ortogonal a $\mathbf{C}_{k-1}$ para cualquier $k$ —por ejemplo, $\boldsymbol{w}_{k-1}$— ($\mathbf{C}_{k-1}\boldsymbol{w}_{k-1} = 0$), se preserva la ortogonalidad de $\boldsymbol{w}_{k-1}$ con $\mathbf{C}_k$,

$$\mathbf{C}_k\boldsymbol{w}_{k-1} = \mathbf{C}_{k-1}\boldsymbol{w}_{k-1} - \frac{\mathbf{C}_{k-1}\boldsymbol{w}_k\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_{k-1}}{\boldsymbol{w}_k^\top\mathbf{C}_{k-1}\boldsymbol{w}_k} = 0,$$

pues $\mathbf{C}_{k-1}\boldsymbol{w}_{k-1} = 0$.

## 2.2. Revisión de métodos MVA

En este apartado, se pretende revisar los principales métodos de análisis multivariante: PCA, PLS, CCA y OPLS. El objetivo de esta familia de métodos MVA consiste en proyectar los datos disponibles en los espacios de entrada y/o salida para extraer aquellas características más representativas del problema, permitiendo no solo reducir la dimensionalidad de los datos, sino facilitando el funcionamiento de ulteriores procesos de aprendizaje máquina. Sin embargo, el modo de conseguir dicho fin depende del método usado. Por ello, se hará un repaso tanto de las formulaciones más conocidas de estos métodos como de sus interpretaciones y se ilustrarán sus capacidades mediante un ejemplo comparativo sobre un problema de regresión aplicado a la teledetección (Frank y Asuncion, 2010).

Puesto que, como se verá más adelante, los métodos MVA pueden formularse como problemas de multiregresión lineal, a continuación se repasará sucintamente este aspecto. El objetivo de este problema de regresión múltiple consiste en diseñar un modelo lineal (es decir, obtener una matriz $\mathbf{W}$) con el fin de predecir lo mejor posible la salida a partir de la entrada: $\tilde{\mathbf{Y}} = \mathbf{W}\mathbf{X}$. Para ello, es común ver la minimización del error cuadrático medio (MSE) como formulación del problema: $\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2$, donde $\|\mathbf{A}\|_F$ denota la norma de Frobenius de la matriz $\mathbf{A}$. La solución a este problema es:

$$\mathbf{W} = \mathbf{C}_{\mathbf{XY}}^\top\mathbf{C}_{\mathbf{XX}}^{-1}.$$

No obstante, en caso de existir multicolinealidades o dependencias lineales entre variables de entrada, este problema no estaría bien condicionado

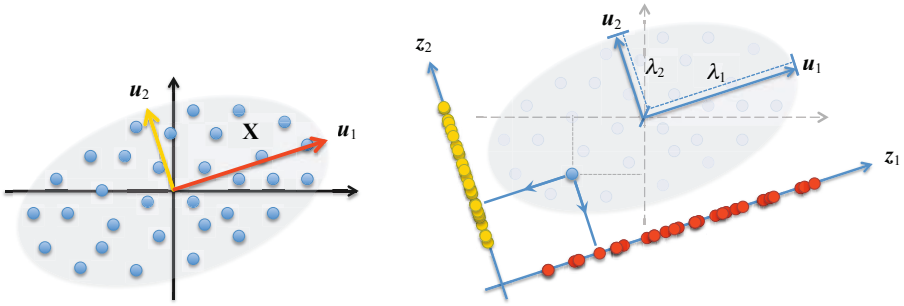
—la solución no sería única—, ya que la matriz $\mathbf{C_{XX}}$ sería singular (no invertible). La solución que los métodos MVA proponen a este problema sería proyectar los datos a un subespacio que preserve la mayor información relevante posible para el problema de regresión. Para conseguir esto, se usa una *matriz de proyección*[3] $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_f}] \in \mathbb{R}^{n \times n_f}$, donde $\boldsymbol{u}_j \in \mathbb{R}^{n \times 1}$ es el $j$-ésimo vector de proyección y $n_f < n$ es el número de características consideradas ("number of features", $n_f$). Los datos de entrada proyectados se denotarán como $\bar{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$ y contendrán las $n_f$ características extraídas de los datos de entrada originales. De este modo, la solución al subsiguiente problema de regresión vendría dada por $\mathbf{W} = \mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}}^\top \mathbf{C}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}$, debiendo invertir únicamente una matriz cuadrada de rango completo que típicamente será de tamaño $n_f \ll n$. En ocasiones, la extracción de características también es aplicada a la matriz de salida $\mathbf{Y}$. Como ya se comentó, el modo de conseguir estas matrices de proyección dependerá de cada método MVA particular.

### 2.2.1. PCA

El análisis de componentes principales ("Principal Component Analysis", PCA), propuesto por Pearson (1901a), es el método de análisis multivariante basado en autovectores más sencillo de todos. Su objetivo es revelar la estructura interna de los datos que mejor explique su varianza o, dicho de otra manera, PCA trata de encontrar las direcciones con máxima varianza en los datos. Para ello, este método realiza una transformación ortogonal, $\mathbf{U}$, de un conjunto de datos, $\mathbf{X}$, con variables, en general, correladas ($\mathbf{C_{XX}} \neq \mathbf{I}$) a otro conjunto cuyas variables, conocidas como componentes principales o *características* ($\mathbf{Z} = \mathbf{U}^\top \mathbf{X}$), están incorreladas linealmente o *blanqueadas*, $\mathbf{C_{ZZ}} = \mathbf{U}^\top \mathbf{C_{XX}} \mathbf{U} = \mathbf{I}$. Cuando el número de características obtenidas es menor que el número de variables originales, se dice que hay reducción de dimensionalidad.

Aunque el cálculo de dicha transformación ortogonal puede formularse de distintas maneras —como se verá a continuación— siempre ha de explicarse la mayor varianza posible en la primera característica y cada una de las siguientes componentes principales deben de recoger, a su vez, la varianza más alta posible siempre y cuando sea ortogonal a las anteriores (es decir, $\mathbf{Z}_{k-1} \boldsymbol{z}_k^\top = 0$, donde $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k-1}$ son las filas de $\mathbf{Z}_{k-1}$). En la Figura 2.3, se muestra un ejemplo para un conjunto de datos de dos dimensiones $\mathbf{X} \in \mathbb{R}^{2 \times 32}$, siendo $\boldsymbol{u}_1$ y $\boldsymbol{u}_2$ los autovectores principales que explican las dos

[3] Nótese que $\mathbf{U}$ no es un operador de proyección (como ha sido descrito en el subapartado 2.1.2) en el sentido rigurosamente matemático, ya que mapea los datos de una espacio $\mathbb{R}^n$ a otro más pequeño $\mathbb{R}^{n_f}$ y, por tanto, no satisface la propiedad de idempotencia de los operadores de proyección. Sin embargo, las columnas de $\mathbf{U}$ definen el espacio $\mathbb{R}^n$ donde los datos son proyectados, siendo en este sentido que nos referimos a $\mathbf{U}$ y $\boldsymbol{u}_i$ como matriz y vectores de proyección respectivamente, y a $\bar{\mathbf{X}}$ como datos proyectados. Esta nomenclatura ha sido ampliamente usada en el campo del aprendizaje automático ("Machine Learning"), particularmente en trabajos que tratan con métodos de extracción de características.




(a) Direcciones de máxima varianza sobre un conjunto de datos de dos dimensiones

(b) Proyección de los datos al espacio de características

Figura 2.3: Interpretación gráfica del PCA

dimensiones de máxima varianza de los datos y $\lambda_1$ y $\lambda_2$ los autovalores asociados a dichos autovectores, que expresan su magnitud; es decir, la cantidad de la varianza total que está explicada por dicho autovector asociado. En la Figura 2.3b, se ve cómo se proyectan los datos sobre las dos características principales $\boldsymbol{z}_1$ y $\boldsymbol{z}_2$.

A menudo, PCA es usado para encontrar una matriz representativa de $\mathbf{X}$ de menor dimensionalidad con el fin de poder reconstruir la matriz original con el menor error cuadrático medio (MSE). Un modo de encontrar el mínimo error de reconstrucción —o de aproximación entre $\mathbf{X}$ y su matriz representativa—, como se comentó en el apartado 2.1.2, sería mediante el uso de las proyecciones ortogonales. Partiendo de la ecuación (2.4) y usando la matriz de proyección ortogonal sobre el espacio definido por las columnas de $\mathbf{X}$ definido en (2.6), se podría formular el problema como:

$$\begin{aligned} \mathbf{U}^* &= \arg\min_{\mathbf{U}} \ \|\mathbf{X} - \mathbf{P}_{\mathcal{S}_{\text{col}}(\mathbf{X})}\mathbf{X}\|_F^2 \\ &= \arg\min_{\mathbf{U}} \ \|\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X}\|_F^2, \\ \text{sujeto a :} \ & \ \mathbf{U}^\top\mathbf{U} = \mathbf{I}, \end{aligned} \tag{PCA.1}$$

donde la condición $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ únicamente tiene como objetivo hacer única la solución de (PCA.1), ya que este problema tiene múltiples soluciones que son óptimas y que presentan el mismo error cuadrático medio.

Un modo de ver esta formulación, que puede ayudar en la explicación de los siguiente métodos MVA, consistiría en traducirla como el siguiente proceso: la matriz $\mathbf{U}^\top$ primeramente mapea o "proyecta" los datos $\mathbf{X}$ al *espacio de características* o *espacio latente*, para que seguidamente la *matriz de proyección* $\mathbf{U}$ vuelva a recuperar $\hat{\mathbf{X}} = \mathbf{U}\mathbf{Z}$ trayéndose dichos *datos proyectados* ($\mathbf{Z} = \bar{\mathbf{X}} = \mathbf{U}^\top\mathbf{X}$) al espacio original.

Si se reescribe esta formulación, sabiendo que $\|\mathbf{A}\|_F^2 = \text{Tr}\{\mathbf{A}\mathbf{A}^\top\}$ y que

las variables (columnas) de $\mathbf{U}$ han de ser ortogonales ($\mathbf{U}^\top\mathbf{U} = \mathbf{I}$),

$$\begin{aligned}\|\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X}\|_F^2 &= \operatorname{Tr}\{(\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X})(\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X})^\top\} \\ &= \operatorname{Tr}\{\mathbf{C}_{\mathbf{XX}}\} - 2\operatorname{Tr}\{\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U}\} + \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U}\} \\ &= \operatorname{Tr}\{\mathbf{C}_{\mathbf{XX}}\} - \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U}\},\end{aligned}$$

se puede ver que la función objetivo del PCA se puede formular como el problema de maximización de varianza con restricciones descrito anteriormente:

$$\begin{aligned}&\mathbf{U}^* = \arg\operatorname*{máx}_{\mathbf{U}} \ \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U}\}, \\ &\text{sujeto a : } \ \mathbf{U}^\top\mathbf{U} = \mathbf{I}.\end{aligned} \tag{PCA.2}$$

Usando multiplicadores de Lagrange, este problema puede reformularse como:

$$\mathbf{U}^* = \arg\operatorname*{máx}_{\mathbf{U}} \ \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U}\} - \operatorname{Tr}\{(\mathbf{U}^\top\mathbf{U} - \mathbf{I})\mathbf{\Lambda}\},$$

siendo $\mathbf{\Lambda}$ la matriz con los multiplicadores de Lagrange. Si ahora se deriva con respecto a $\mathbf{U}$, se iguala a cero y se supone que $\mathbf{\Lambda}$ es diagonal, se ve que este problema se puede resolver con el siguiente problema de autovalores estándar:

$$\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}, \tag{PCA.3}$$

siendo $\mathbf{U} \in \mathbb{R}^{n\times k}$ la matriz de $k \leq n$ autovectores y $\mathbf{\Lambda} \in \mathbb{R}^{k\times k}$ la matriz de autovalores. Además, como se ha visto en el subapartado 2.1.3.3, las columnas de $\mathbf{U}$ también son los vectores singulares izquierdos de $\mathbf{X}$. Por lo tanto, el problema también se puede resolver con la siguiente descomposición en valores singulares (SVD) de $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{PCA.4}$$

siendo $\mathbf{\Sigma} = \mathbf{\Lambda}^{\frac{1}{2}}$ y $\mathbf{V} \in \mathbb{R}^{k\times N}$ los vectores singulares derechos de $\mathbf{X}$.

Puesto que el PCA se puede reducir a un problema de autovalores estándar, la implementación del PCA podría realizarse también mediante cualquiera de los métodos de deflacción descritos en el apartado 2.1.4, obteniendo secuencialmente los autovectores de $\mathbf{C}_{\mathbf{XX}}$ (por ejemplo con la deflacción de Hotelling: $\mathbf{C}_{\mathbf{XX}} \leftarrow \mathbf{C}_{\mathbf{XX}} - \boldsymbol{u}\boldsymbol{u}^\top\mathbf{C}_{\mathbf{XX}}\boldsymbol{u}\boldsymbol{u}^\top$). De este modo, la solución alcanzada en cada iteración es óptima con respecto al criterio del PCA para el número actual de proyecciones.

Es importante recordar que el objetivo del PCA es obtener una matriz de proyección (o de transformación) con el fin de blanquear los datos de entrada, es decir, que las variables de los datos proyectados $\mathbf{Z} = \mathbf{U}^\top\mathbf{X}$ estén incorrelados y tengan varianza unidad:

$$\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{I}.$$

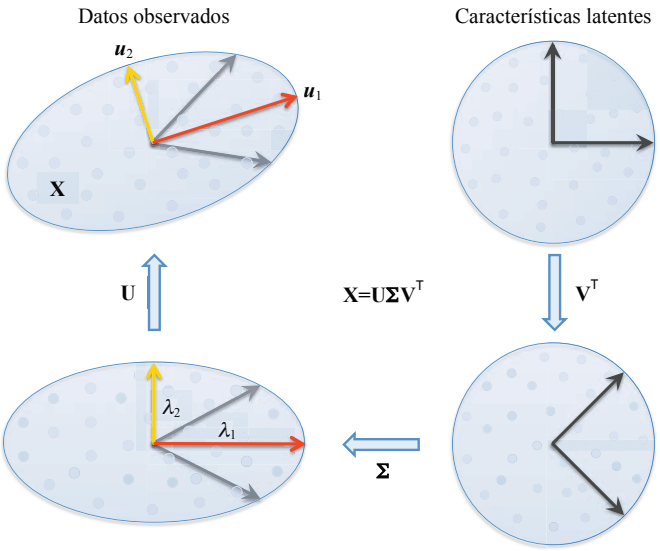



Figura 2.4: Interpretación del PCA con la descomposición SVD

De este modo, el espacio definido por $\mathbf{Z}$ esbozaría una hiperesfera de tamaño unidad. La propiedad de ortogonalidad de $\mathbf{U}$ ($\mathbf{U}^\top\mathbf{U} = \mathbf{I}$) es simplemente una herramienta para conseguir esto, ya que $\mathbf{Z} = \mathbf{U}^\top\mathbf{X}$ se puede ver como la proyección de $\mathbf{X}$ sobre el espacio definido por $\mathbf{U}$ (véanse los Subapartados 3.2.1 y 3.2.4 para su demostración).

En otras palabras, se podría decir que los datos observados son el resultado de la aplicación de una transformación desconocida sobre las verdaderas características subyacentes del problema, conocidas también como *variables latentes*, siendo el objetivo del PCA descubrir esa tranformación para poder recuperar las características latentes deseadas. Este efecto se puede ver bien en la Figura 2.4 que ilustra la solución SVD descrita en la formulación (PCA.4) del problema.

Una de las ventajas de obtener datos blanqueados es la habilidad de facilitar la operación de clasificadores más complejos aplicados sobre estos datos que sobre los originales, acelerando así el proceso de entrenamiento y clasificación.

Sin embargo, el PCA es un método *no supervisado*, es decir, que no tiene en cuenta la información disponible a priori sobre los datos (matriz de etiquetas o datos de salida, $\mathbf{Y}$) de la que se pueda disponer. Por lo tanto, cuando las proyecciones extraídas van a ser usadas en una tarea de aprendizaje supervisado (ya sea clasificación o regresión), el PCA sería subóptimo, pues aquellas proyecciones que contienen la mayor varianza en el espacio de entrada no tienen por qué estar alineadas con la función objetivo. Un ejemplo de esto se puede ver en la Figura 2.5 para un problema de clasificación

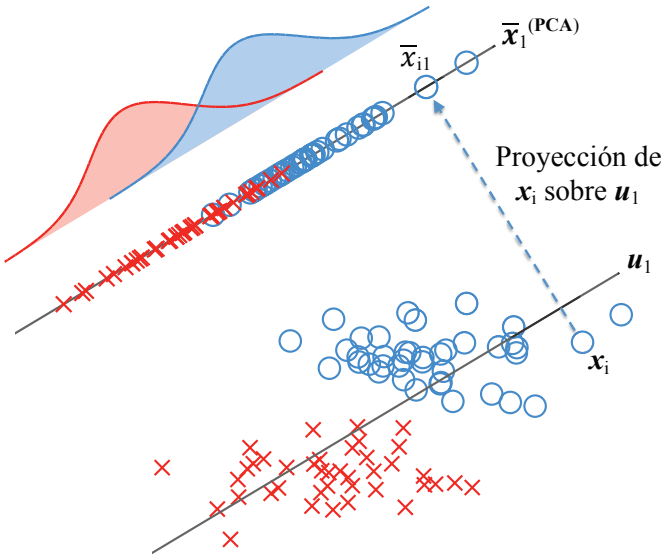



Figura 2.5: Proyección de los datos sobre la primera componente principal del PCA para una tarea de clasificación binaria. Los datos han sido generados con una distribución Gaussiana bidimensional para cada clase, cuyas proyecciones sobre el primer autovector $\bar{\boldsymbol{x}}_1$ se muestran en la parte superior.

binaria donde, a pesar de tratarse de un problema linealmente separable, no se podría discriminar todas las muestras a partir de sus proyecciones sobre el espacio definido por la primera componente principal del PCA. Por esta razón, los métodos MVA supervisados, que hacen uso de los datos de salida para obtener la matriz de proyección, permiten obtener mejores prestaciones que el PCA.

### 2.2.2. PLS

El método de mínimos cuadrados parciales ("Partial Least Squares", PLS) propuesto por Wold (1966a,b) es uno de los métodos MVA supervisados más sencillos. Sin embargo, en función de la implementación usada se obtienen distintos algoritmos con distintas soluciones.

El objetivo de este método es obtener las direcciones que maximicen la covarianza entre los datos de entrada y los de salida, $\mathbf{C_{XY}}$. Para ello, la formulación del problema podría ser la siguiente:

$$\begin{aligned} &\mathbf{U}^*, \mathbf{V}^* = \arg\max_{\mathbf{U},\mathbf{V}} \ \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C_{XY}} \mathbf{V}\}, \\ &\text{sujeto a : } \ \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}. \end{aligned} \tag{PLS.1}$$

Reescribiendo esta formulación del problema en términos de regresión para minimizar el error cuadrático medio entre los datos proyectados de

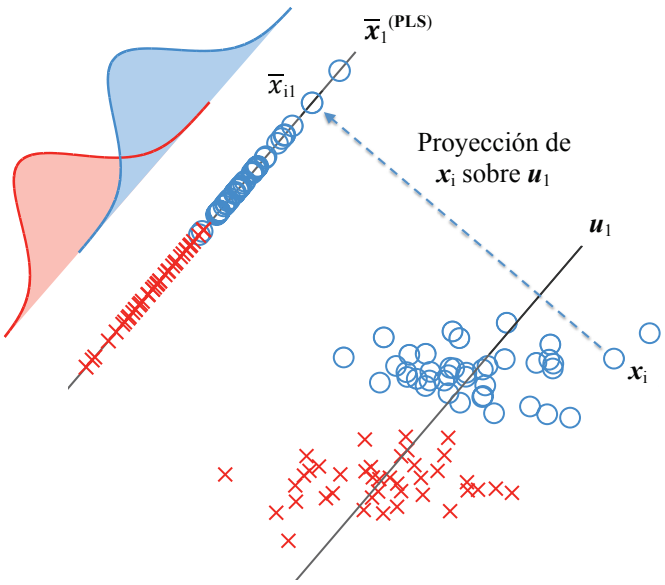



Figura 2.6: Proyección de los datos sobre la primera componente principal del PLS para una tarea de clasificación binaria

entrada $\mathbf{U}^\top\mathbf{X}$ y los de salida $\mathbf{V}^\top\mathbf{Y}$, es fácil ver (del mismo modo que se ha hecho con PCA) que el problema es equivalente a:

$$\begin{aligned} &\mathbf{U}^*, \mathbf{V}^* = \arg\min_{\mathbf{U},\mathbf{V}} \ \|\mathbf{V}^\top\mathbf{Y} - \mathbf{U}^\top\mathbf{X}\|_F^2, \\ &\text{sujeto a : } \ \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I} \ \ (\text{y} \ \ \mathbf{U}\mathbf{U}^\top = \mathbf{V}\mathbf{V}^\top = \mathbf{I}). \end{aligned} \tag{PLS.2}$$

La interpretación bajo esta formulación consiste en encontrar las variables latentes comunes que han generado los dos conjuntos $\mathbf{X}$ e $\mathbf{Y}$ mediante dos transformaciones lineales distintas y desconocidas $\mathbf{U}$ y $\mathbf{V}$ respectivamente. Sin embargo, no busca que las características obtenidas (los datos proyectados) sean ortogonales, propiedad muy importante para los métodos MVA como ya se ha comentado con PCA. La consecuencia de esto puede verse en la Figura 2.6, ya que la división entre clases a partir de los datos proyectados sobre el espacio del primer autovector de PLS no es perfecta cuando debería serlo —pues se trata de un problema de clasificación binario linealmente separable—.

Con estas formulaciones, es fácil ver que la solución del problema se puede obtener mediante la SVD de la matriz $\mathbf{C_{XY}}$:

$$\mathbf{C_{XY}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{PLS.3}$$

ya que $\mathbf{U}$ y $\mathbf{V}$ son matrices ortogonales compuestas por los vectores singulares izquierdos y derechos respectivamente. Debido a esto, el número máximo de vectores de proyección es el rango de $\mathbf{C_{XY}}$.

Sin embargo, este problema puede resolverse utilizando diferentes algoritmos que pueden incluso proporcionar otras soluciones diferentes. En concreto, cuando se procede secuencialmente, el uso de diferentes técnicas de deflacción tiene implicaciones en la solución obtenida. Por ejemplo, el algoritmo PLS-SB propuesto por Sampson et al. (1989) realiza la deflacción por proyección de $\mathbf{C_{XY}}$ para los vectores singulares izquierdos $\boldsymbol{u}$ y de $\mathbf{C_{XY}^\top}$ para los vectores singulares derechos $\boldsymbol{v}$:

$$\mathbf{C_{XY}} \leftarrow (\mathbf{I} - \boldsymbol{u}\boldsymbol{u}^\top)\mathbf{C_{XY}}, \quad \mathbf{C_{XY}^\top} \leftarrow (\mathbf{I} - \boldsymbol{v}\boldsymbol{v}^\top)\mathbf{C_{XY}^\top};$$

en otras palabras, si se despeja la matriz correspondiente no involucrada en la ecuación, se deflactan simplemente $\mathbf{X}$ e $\mathbf{Y}$ como:

$$\mathbf{X} \leftarrow (\mathbf{I} - \boldsymbol{u}\boldsymbol{u}^\top)\mathbf{X}, \quad \mathbf{Y} \leftarrow (\mathbf{I} - \boldsymbol{v}\boldsymbol{v}^\top)\mathbf{Y},$$

respectivamente. Esta deflacción puede ser resumida de una vez eliminando de $\mathbf{C_{XY}}$ la influencia de los $j$-ésimos vectores singulares $\boldsymbol{u}_j$ y $\boldsymbol{v}_j$ (véase el subapartado 2.1.3.3 para ver la relación entre vectores singulares y autovectores):

$$\mathbf{C_{XY}} \leftarrow \mathbf{C_{XY}} - \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j, \tag{PLS-SB}$$

siendo $\sigma_j$ el $j$-ésimo valor singular. Esta solución es la misma que la obtenida al resolver la SVD en bloque.

Por el contrario, si únicamente se necesita la matriz de proyección de los datos de entrada, por ejemplo porque se quiere aplicar un regresor o clasificador sobre los datos proyectados, una de las soluciones más usadas es la propuesta por Wold et al. (1984) conocida como PLS2 de acuerdo con Wegelin (2000). Esta implementación difiere de PLS-SB no solo en que no se deflacta la matriz $\mathbf{Y}$, sino también en el método de deflacción usado, ya que en lugar de aplicar la proyección ortogonal de $\mathbf{X}$ sobre el complemento ortogonal del espacio definido por los autovectores $\boldsymbol{u}$, $\mathcal{P}_{\boldsymbol{u}}^{\perp}(\mathbf{X})$, lo hace sobre el complemento ortogonal del espacio definido por los datos proyectados por dichos autovectores —es decir, por $\bar{\boldsymbol{x}} = \mathbf{X}^\top \boldsymbol{u}$—, $\mathcal{P}_{\bar{\boldsymbol{x}}}^{\perp}(\mathbf{X})$ (para más detalle véase la deflacción por complemento de Schur en el subapartado 2.1.4.3):

$$\mathbf{X} \leftarrow \mathbf{X}\left(\mathbf{I} - \frac{\mathbf{X}^\top \boldsymbol{u}\boldsymbol{u}^\top \mathbf{X}}{\|\mathbf{X}^\top \boldsymbol{u}\|^2}\right). \tag{PLS2}$$

Cabe destacar que esta deflacción no sería la correspondiente a la aplicada en el cálculo secuencial de la SVD de $\mathbf{C_{XY}}$ —como debería ser—, sino en el cálculo secuencial de la SVD de $\mathbf{X}$ —como sería el caso del PCA—. Debido a esto, el número máximo de vectores de proyección que se puede obtener con esta solución es el rango de $\mathbf{C_{XX}}$. Por lo tanto, esta solución ya no coincide con la solución PLS. La ventaja de esta deflacción con respecto a la usada por PLS-SB es que, al usar la deflacción por complemento de Schur, se preserva la ortogonalidad con las subsiguientes rondas del proceso de deflacción; es

decir, además de cumplir $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, también cumple $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}$, siendo $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{n_f}]$ y $\boldsymbol{z}_k = \mathbf{X}_k^\top \boldsymbol{u}_k$ con $k = 1, \ldots, n_f$. Cabe destacar que $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}$ no conlleva a que se cumpla al blanqueamiento de los datos proyectados $\bar{\mathbf{X}}^\top\bar{\mathbf{X}} = \mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{I}$, que, como ya se comentó en la motivación de los métodos MVA, es una propiedad deseada de estos métodos. Como se verá a continuación, los siguientes dos métodos sí cumplen con esta propiedad y, por ello, serán preferidos frente a PLS.

### 2.2.3. CCA

El análisis de componentes canónicas ("Canonical Correlation Analysis", CCA) propuesto por Hotelling (1936) busca las direcciones de máxima correlación entre los datos de entrada y los de salida, a diferencia de PLS que busca las de máxima covarianza. A menudo, este método es usado para estudiar las relaciones entre dos conjuntos de datos distintos.

Sabiendo que el coeficiente de correlación entre los datos de entrada proyectados por un vector $\boldsymbol{u}$ y los de salida proyectados por un vector $\boldsymbol{v}$ es

$$\rho = \frac{\boldsymbol{u}^\top \mathbf{C}_{\mathbf{XY}} \boldsymbol{v}}{\sqrt{\boldsymbol{u}^\top \mathbf{C}_{\mathbf{XX}} \boldsymbol{u}}\sqrt{\boldsymbol{v}^\top \mathbf{C}_{\mathbf{YY}} \boldsymbol{v}}},$$

y teniendo en cuenta que la maximización de esta correlación con respecto a $\boldsymbol{u}$ y $\boldsymbol{v}$ es invariante a cualquier factor de escala, entonces el CCA se puede formular (de forma matricial) como:

$$\begin{aligned} &\mathbf{U}^*, \mathbf{V}^* = \underset{\mathbf{U},\mathbf{V}}{\arg\max}\ \ \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{\mathbf{XY}} \mathbf{V}\}, \\ &\text{sujeto a :}\ \ \mathbf{U}^\top \mathbf{C}_{\mathbf{XX}} \mathbf{U} = \mathbf{V}^\top \mathbf{C}_{\mathbf{YY}} \mathbf{V} = \mathbf{I}. \end{aligned} \tag{CCA.1}$$

Haciendo uso de los multiplicadores de Lagrange de igual modo que con PCA, se llega a que CCA puede solventarse con el siguiente problema de autovalores generalizado:

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{\mathbf{XY}} \\ \mathbf{C}_{\mathbf{XY}}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{\mathbf{XX}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{YY}} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \end{pmatrix} \mathbf{\Lambda}, \tag{CCA.2}$$

que puede reescribirse como el siguiente problema de autovalores estándar:

$$\begin{pmatrix} \mathbf{C}_{\mathbf{XX}}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{C}_{\mathbf{XY}} \\ \mathbf{C}_{\mathbf{XY}}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{C}_{\mathbf{XX}}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \mathbf{U}' \\ \mathbf{V}' \end{pmatrix} = \begin{pmatrix} \mathbf{U}' \\ \mathbf{V}' \end{pmatrix} \mathbf{\Lambda},$$

siendo $\mathbf{U}' = \mathbf{C}_{\mathbf{XX}}^{\frac{1}{2}}\mathbf{U}$, $\mathbf{V}' = \mathbf{C}_{\mathbf{YY}}^{\frac{1}{2}}\mathbf{V}$ y $\mathbf{0}$ una matriz de ceros con las dimensiones adecuadas.

Al igual que PLS, CCA busca las variables latentes comunes que han generado los dos conjuntos $\mathbf{X}$ e $\mathbf{Y}$ mediante dos transformaciones lineales

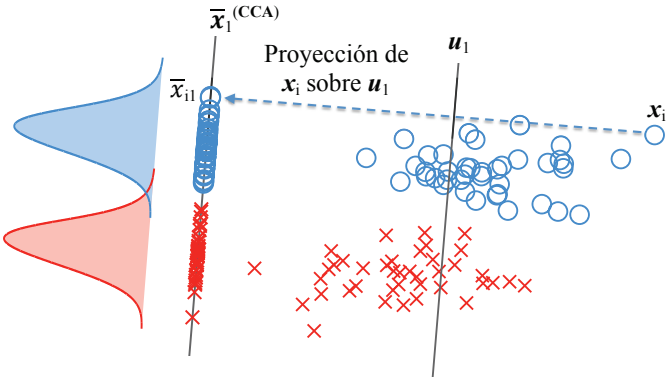



Figura 2.7: Proyección de los datos sobre la primera componente principal del CCA para una tarea de clasificación binaria

distintas y desconocidas $\mathbf{U}$ y $\mathbf{V}$ respectivamente, pero, a diferencia del anterior, CCA sí obtiene unas características (datos proyectados) blanqueadas. La consecuencia de esta diferencia se puede ver en la Figura 2.7, donde en este caso sí que es linealmente separable el problema de clasificación binaria sobre los datos proyectados en el espacio definido por el primer autovector de CCA.

La formulación que describe esta interpretación minimiza el error cuadrático medio entre las dos proyecciones como:

$$\begin{aligned} \mathbf{U}^*, \mathbf{V}^* = \arg\min_{\mathbf{U},\mathbf{V}} \ & \|\mathbf{V}^\top \mathbf{Y} - \mathbf{U}^\top \mathbf{X}\|_F^2, \\ \text{sujeto a : } \ & \mathbf{U}^\top \mathbf{C_{XX}} \mathbf{U} = \mathbf{V}^\top \mathbf{C_{YY}} \mathbf{V} = \mathbf{I}, \end{aligned} \tag{CCA.3}$$

que reescribiéndolo mediante el operador traza se llegaría a la primera formulación (CCA.1).

No obstante, el objetivo que normalmente suele ser de mayor interés para tareas de clasificación o regresión supervisada es aproximar la matriz de salida original $\mathbf{Y}$ y no su proyección. Por este motivo, OPLS suele ser preferido, ya que, como se verá a continuación, es óptimo en este sentido.

### 2.2.4. OPLS

El método de mínimos cuadrados parciales ortonormalizado ("Orthonormalized Partial Least Squares", OPLS[4]) fue propuesto por Worsley et al. (1996) con el fin de combinar las ventajas de PLS y CCA evitando los problemas que obtenía con ellos. En particular, querían hacer que las variables

[4]Es importante no confundir OPLS con el método O-PLS ("Orthogonal Projections to Latent Structures") propuesto por Trygg y Wold (2002) que no supone una mejora en la capacidad predictiva del PLS sino en su interpretabilidad.

de entrada fuesen invariantes a transformaciones lineales o, en otras palabras, que estuviesen blanqueadas (PLS no lo hace), pero no querían blanquear las variables de salida como hace CCA, entre otras razones porque si su número era muy elevado (que en su caso era así), invertir la matriz $\mathbf{C_{YY}}$ era inviable. La formulación que entonces se propuso partía de la solución (PLS.3) $\mathbf{C_{XY}} = \mathbf{U\Sigma V}^\top$, pero ortonormalizando (o blanqueando) las variables de entrada:

$$\mathbf{C_{XX}}^{-\frac{1}{2}}\mathbf{C_{XY}} = \mathbf{U\Sigma V}^\top. \tag{OPLS.1}$$

Con el fin de encontrar un codificador lineal con un cuello de botella de tamaño $r < \text{rango}(\mathbf{C_{XY}})$ que evitase los problemas de sobreajuste y pobre generalización cuando los datos de entrada presentan una alta dimensionalidad, Roweis y Brody (1999) propusieron la misma solución anterior y demostraron que provenía de la minimización del error cuadrático medio $\|\mathbf{Y} - \mathbf{A}_r^\top\mathbf{X}\|_F^2$, siendo $\mathbf{A}_r$ la transformada óptima de rango reducido $r$ de un codificador lineal: $\mathbf{A}_r = \mathbf{C_{XX}}^{-\frac{1}{2}}\mathbf{U\Sigma V}^\top$. Esta formulación fue reescrita por Arenas-García y Camps-Valls (2008) como la solución óptima de mínimo error cuadrático medio:

$$\begin{aligned} &\mathbf{U}^* = \arg\min_{\mathbf{U}} \|\mathbf{Y} - \mathbf{W}\mathbf{U}^\top\mathbf{X}\|_F^2, \\ &\text{sujeto a : } \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{I}. \end{aligned} \tag{OPLS.2}$$

siendo $\mathbf{W}$ la matriz óptima de regresión entre $\mathbf{Y}$ y los datos de entrada proyectados $\bar{\mathbf{X}} = \mathbf{U}^\top\mathbf{X}$: $\|\mathbf{Y} - \mathbf{W}\bar{\mathbf{X}}\|_F^2$. Nótese que, al igual que en el caso del PCA, la condición $\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{I}$ tiene como objetivo hacer única la solución de (OPLS.2), ya que existen múltiples soluciones óptimas con el mismo MSE. Como el objetivo del OPLS es simplemente obtener la matriz de proyección de entrada $\mathbf{U}$, sustituyeron la solución óptima $\mathbf{W}^*$ dentro de la función objetivo dejándola solamente en función de $\mathbf{U}$ (véase el apartado 3.1 para un estudio más detallado de esta solución OPLS). De este modo, se puede reescribir el problema mediante el operador traza como:

$$\begin{aligned} &\mathbf{U}^* = \arg\max_{\mathbf{U}} \ \text{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\mathbf{U}\}, \\ &\text{sujeto a : } \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{I}. \end{aligned} \tag{OPLS.3}$$

La solución de este problema se puede obtener mediante el siguiente problema de autovalores generalizado:

$$\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\mathbf{U} = \mathbf{C_{XX}}\mathbf{U\Lambda}, \tag{OPLS.4}$$

o en forma de problema de autovalores estándar:

$$\mathbf{C_{XX}}^{-\frac{1}{2}}\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\mathbf{C_{XX}}^{-\frac{1}{2}}\mathbf{U}' = \mathbf{U}'\mathbf{\Lambda},$$

siendo $\mathbf{U}' = \mathbf{C_{XX}}^{\frac{1}{2}}\mathbf{U}$.

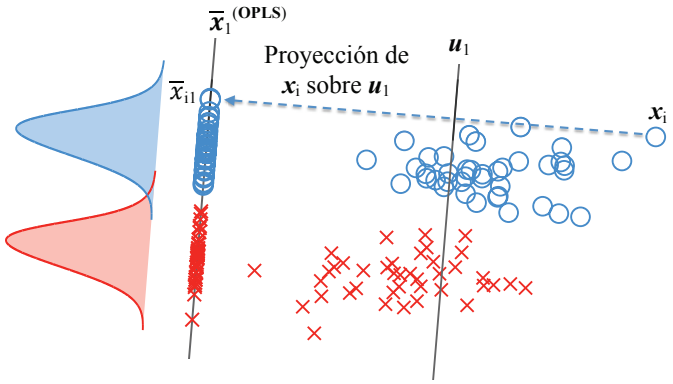



Figura 2.8: Proyección de los datos sobre la primera componente principal del OPLS para una tarea de clasificación binaria

Por tanto, el OPLS es preferible cuando se quiere proyectar los datos de entrada para fines de regresión o clasificación, ya que es óptimo en el sentido de mínimo error cuadrático medio (MSE).

En la Figura 2.8, se puede ver el motivo de esta preferencia, ya que obtiene una clasificación perfecta a partir de los datos proyectados sobre el primer autovector de OPLS en el problema binario. Es interesante comparar esta proyección con la obtenida con CCA en la Figura 2.7. Como se puede ver, es la misma proyección y el mismo autovector. Esto es debido a que, al tratarse de un problema de clasificación donde la matriz de etiquetas es codificada[5] asignando un "1" a la clase a la que pertenece la muestra y "0" en caso contrario, la matriz $\mathbf{C_{YY}}$ sería diagonal y, por lo tanto, CCA se comportaría como el OPLS si y solo si el número de muestras es igual para ambas clases. Esta comparación se estudiará con más detalle en el Subapartado 3.2.2.

### 2.2.5. Ejemplo comparativo de métodos MVA en regresión

En este subapartado, se pretende comparar gráficamente las prestaciones obtenidas de los distintos métodos MVA que se acaban de presentar: PCA, PLS-SB, PLS2, CCA y OPLS. Para ello, se ha reusado el mismo ejemplo ilustrado por Arenas-García y Petersen (2009) debido a la naturaleza redundante del problema utilizado. Este conjunto de datos[6] consta de 4 435/2 000 imágenes de entrenamiento/test tomadas por los escáneres multiespectrales a bordo de los satélites Landsat ("Landsat MSS"). Dichas imágenes constan

[5]Este tipo de codificaciones suele ser usada cuando existen variables categóricas, en contraposición a numéricas, ya que las herramientas de álgebra lineal no podrían trabajar con ellas. En este caso, los datos de salida constituyen un vector de datos categóricos, ya que cada muestra es una asignación a una clase determinada.

[6]El conjunto de datos usado puede descargarse de `https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/satimage/`.

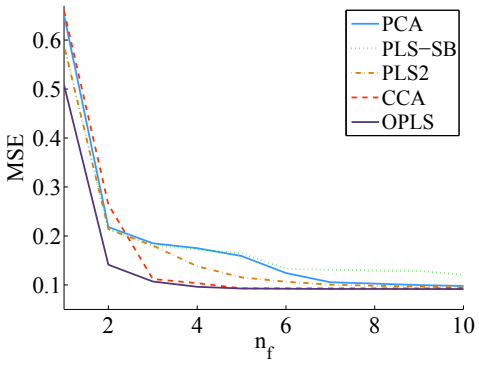



Figura 2.9: Comparación del error cuadrático medio (MSE) obtenido tras proyectar los datos de entrada con los distintos métodos MVA

de $82 \times 100$ píxeles con una resolución espacial de $80 \times 80$ m. Tras el agrupamiento de píxeles vecinos con una ventana de $3 \times 3$ píxeles, cada observación es convertida en un vector de 36 variables de entrada.

Con el fin de evaluar las prestaciones de los distintos métodos de análisis, se ha usado como conjunto de entrada las 26 primeras variables de entradas y como conjunto de salida las otras 10 restantes. Tras el cálculo de las matrices de proyección ($\mathbf{U}$) y de regresión ($\mathbf{W}$), se ha obtenido el error cuadrático medio (MSE) sobre el conjunto de test como: $\text{MSE} = \frac{1}{N}\|\mathbf{Y}_{\text{test}} - \mathbf{W}\mathbf{U}^\top \mathbf{X}_{\text{test}}\|_F^2$, siendo $\mathbf{X}_{\text{test}}$ e $\mathbf{Y}_{\text{test}}$ el conjunto de datos etiquetados usados para evaluar los métodos MVA entrenados.

Los resultados obtenidos por los cinco métodos considerados se muestran en la Figura 2.9 en función del número de características extraídas ($n_f$). La primera conclusión que se puede sacar a la vista de los resultados es la superioridad del OPLS, que alcanza el menor error sin importar el número de características extraídas. Esto es debido a que, como ya se comentó, el OPLS es óptimo en el sentido del MSE. También es interesante destacar la gran diferencia entre las dos implementaciones del PLS, siendo incluso el PCA mejor que el PLS-SB. Esto es debido a que PLS-SB no blanquea los datos de entrada (las características extraídas no son ortogonales), afectando seriamente las prestaciones del ulterior regresor. Por el contrario, PLS2 consigue dicha ortogonalidad en cada paso de deflacción, consiguiendo mejores prestaciones que PCA y PLS-SB. Tanto PLS2 como PCA podrían extraer más características, pero no se han mostrado porque las ventajas obtenidas eran insignificantes.

# Parte II

# Propuesta doctoral

En esta segunda parte de la Tesis, se describen las diferentes propuestas doctorales. Está compuesta de cinco capítulos, uno por propuesta, y aunque estén todas ellas relacionadas entre sí, cada una tiene un fin muy distinto. Además, se incluye un sexto capítulo de conclusiones y algunas ideas para continuar con este trabajo.

Aunque la primera propuesta parezca que carece de originalidad al ser puramente teórica, es quizá la parte que podría aportar más a la comunidad investigadora, ya que en la actualidad se está usando por defecto en la literatura una solución que aquí se demuestra errónea. A través de una serie de demostraciones, se especifica la solución correcta con el fin de crear un marco estándar generalizado para el tipo de problemas aquí tratados.

# Capítulo 3

# Marco general para análisis multivariante

*Obra de mal cimiento, la derriba el viento.*

Anónimo, proverbio español.

**Resumen:** En este capítulo, se expone la base teórica principal de la tesis a partir de la cual en los siguientes capítulos se desarrollan el resto de propuestas. Se comenzará analizando el tratamiento que se ha dado a algunos métodos MVA en la literatura estadística y en la de aprendizaje automático. A raíz de este análisis, en este capítulo, se aunarán los términos y formulaciones de ambos campos y se propondrá el uso de una formulación eficiente y generalizada para todos los métodos MVA, creando un entorno común de trabajo.

Esto, además, permitirá crear un marco general que facilite la inclusión de restricciones, haciendo posible soluciones especializadas en función de las necesidades requeridas. Además, se demostrarán las deficiencias cometidas en las formulaciones MVA con restricciones existentes hasta el momento y se compararán tanto teórica como empíricamente con el marco propuesto en este capítulo.

Si bien este capítulo es puramente teórico, pretendiendo así fijar una base para cualquier tipo de restricción aplicada al problema, lo cierto es que muchas de las restricciones que se podrían aplicar no son derivables, haciéndose necesario la obtención de resultados empíricos. Dichos resultados se irán consiguiendo a medida que avancen los siguientes capítulos.

## 3.1. Formulaciones alternativas en MVA

Revisando la literatura en ML y en estadística, se pueden encontrar formulaciones equivalentes para el mismo problema. Sin embargo, hasta donde llega nuestro conocimiento, esta conexión entre formulaciones no se ha identificado previamente en la literatura, pudiendo ser considerada, por lo tanto, una primera contribución de esta tesis doctoral a la comunidad investigadora. Para facilitar la exposición, se va a concretar para el caso particular del OPLS con el fin de poder proponer un marco común para todos los métodos MVA que obtienen características ortogonales entre sí. Cabe destacar que una de las propiedades más deseadas para los métodos MVA es extraer características incorreladas por las siguientes ventajas:

- Permite analizar las características por orden de relevancia, posibilitando seleccionar el subconjunto óptimo de un determinado número de características dadas.
- Se facilita el entrenamiento de la subsiguiente etapa de clasificación o regresión, disminuyendo de este modo el coste computacional. Puesto que cada uno de los pesos asociados a cada una de las variables incorreladas es independiente de los demás, los métodos de optimización trabajarían sobre curvas de error más suaves y el aprendizaje de la máquina generalmente requeriría menos tiempo.

El objetivo del OPLS es encontrar los vectores de proyección tales que los datos proyectados puedan ajustarse lo mejor posible a los datos de salida en el sentido de mínimo error cuadrático medio (MSE); es decir, OPLS minimiza la siguiente función de coste (Roweis y Brody, 1999),

$$\mathcal{L}(\mathbf{W}, \mathbf{U}) = \|\mathbf{Y} - \mathbf{W}\mathbf{U}^\top \mathbf{X}\|_F^2, \tag{3.1}$$

donde $\mathbf{W} \in \mathbb{R}^{m \times n_f}$ es una matriz de coeficientes de regresión, que pueden ser vistos alternativamente como una matriz de proyección para los datos de salida. Nótese que el problema de arriba es diferente del problema de regresión de mínimos cuadrados estándar ("Least Squares", LS), ya que la matriz $\mathbf{U}$ impone un cuello de botella (Roweis y Brody, 1999). Nótese, también, que la solución a (3.1) no es única puesto que, por ejemplo, $\mathbf{W}$ puede compensar cualquier escalado de la matriz $\mathbf{U}$. Es importante aclarar que cualquier solución de (3.1) no tiene por qué ser OPLS, ya que se debe obtener también ortogonalidad entre los datos de entrada proyectados (o características extraídas): $\mathbf{U}^\top \mathbf{C}_{\mathbf{XX}} \mathbf{U} = \mathbf{I}$. Para que (3.1) proporcione la solución OPLS, se debe incluir alguna condición adicional a este problema que permita obtener dicha incorrelación. En los siguientes apartados, se verá cómo se han formulado estas restricciones en la comunidad estadística y de ML por separado y se estudiará la relación existente entre ellas, incluyendo también la comparación en términos de coste computacional.

### 3.1.1. OPLS como problema de autovalores generalizado

En este subapartado, se revisa la solución al problema OPLS (vista en el subapartado 2.2.4) que es usada asiduamente en la literatura de aprendizaje automático. En este caso, OPLS es visto típicamente como un método de extracción de características, siendo su objetivo encontrar una solución para $\mathbf{U}$ (véanse, por ejemplo, Arenas-García y Petersen (2009), Worsley et al. (1996), Sun et al. (2009) y De la Torre (2012)).

Para presentar esta formulación, se comienza desarrollando la norma de Frobenius de la ecuación (3.1):

$$\mathcal{L}(\mathbf{W},\mathbf{U}) = \mathrm{Tr}\{\mathbf{C_{YY}}\} - 2\,\mathrm{Tr}\{\mathbf{W}^\top\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\} + \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\mathbf{W}^\top\mathbf{W}\}. \quad (3.2)$$

Como ya se ha comentado, los argumentos que minimizan la función de coste no son únicos. Sin embargo, se puede ver que el óptimo $\mathbf{W}$ es unívocamente determinado para un $\mathbf{U}$ fijado como la solución del problema LS definido en (3.1):

$$\mathbf{W} = \mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\left(\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\right)^{-1}. \quad (3.3)$$

Introduciendo esta expresión en (3.2), y tras alguna manipulación algebraica, la función de coste objetivo puede ser expresada como una función solamente de $\mathbf{U}$:

$$\mathcal{L}(\mathbf{U}) = \mathrm{Tr}\{\mathbf{C_{YY}}\} - \mathrm{Tr}\left\{\left(\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\right)^{-1}\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\right\}. \quad (3.4)$$

La minimización de $\mathcal{L}(\mathbf{U})$ es equivalente a la maximización de la segunda traza de la expresión (3.4), es decir, un problema de maximización de cociente de trazas ("ratio-trace") (véanse, por ejemplo, Ngo et al. (2012) y Jia et al. (2009)). Obviamente, el optimizador de (3.4) no es único, puesto que, por ejemplo, al multiplicar $\mathbf{U}$ por una constante el valor de $\mathcal{L}(\mathbf{U})$ no se vería afectado. El minimizador de (3.4) puede ser alternativamente encontrado resolviendo el siguiente problema de optimización:

$$\begin{array}{ll} \max\limits_{\mathbf{U}} & \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\} \\ \text{sujeto a} & \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{I} \end{array} \quad (3.5)$$

Esta solución del OPLS puede ser obtenida resolviendo el siguiente problema de autovalores generalizado:

$$\mathbf{C_{XY}}\mathbf{C}_{\mathbf{XY}}^\top\boldsymbol{u} = \lambda\mathbf{C_{XX}}\boldsymbol{u}. \quad (3.6)$$

El hecho de obtener la solución de (3.5) mediante (3.6) hace que se puedan obtener los vectores de proyección $\boldsymbol{u}$ ordenados en función de su correspondiente autovalor $\lambda$: se va a denotar $\mathbf{\Lambda}_{\mathrm{GEV}}$ como la matriz diagonal que

contiene los $n_f$ autovalores generalizados de mayor valor de (3.6) dispuestos en orden decreciente, mientras que $\mathbf{U}_{\text{GEV}}$ será una matriz cuyas columnas son los correspondientes $n_f$ autovectores principales. Nótese que cualquier matriz $\mathbf{U}_{\mathbf{R}} = \mathbf{U}_{\text{GEV}}\mathbf{R}$, donde $\mathbf{R}$ es una matriz de rotación, es también una solución a (3.5). Sin embargo, $\mathbf{U}_{\text{GEV}}$ tiene la propiedad de que cualquier subconjunto que contenga solamente las primeras $n'_f < n_f$ columnas de la matriz es también una solución OPLS para el número de dimensiones seleccionado. En otras palabras, usando $\mathbf{U}_{\text{GEV}}$, las características extraídas están ordenadas de acuerdo a su relevancia para el problema de regresión (es decir, la primera característica representa la máxima información que se puede resumir con una sola variable, y así sucesivamente), mientras que esto no sería cierto para la matriz rotada $\mathbf{U}_{\mathbf{R}}$.

Una vez se ha obtenido $\mathbf{U}_{\text{GEV}}$, es sencillo calcular los correspondientes coeficientes de regresión mediante la ecuación (3.3)

$$\mathbf{W}_{\text{GEV}} = \mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}_{\text{GEV}}, \tag{3.7}$$

donde también se ha usado el hecho de que las columnas de $\mathbf{U}_{\text{GEV}}$ son ortonormales con respecto a $\mathbf{C}_{\mathbf{XX}}$ (es decir, $\mathbf{U}_{\text{GEV}}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}_{\text{GEV}} = \mathbf{I}$), como se forzó en (3.5).

Una propiedad aún más interesante de la solución OPLS se puede apreciar, en primera instancia, si se observa que para $\mathbf{U}_{\text{GEV}}$ se satisface

$$\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}_{\text{GEV}} = \mathbf{C}_{\mathbf{XX}}\mathbf{U}_{\text{GEV}}\mathbf{\Lambda}_{\text{GEV}}. \tag{3.8}$$

Luego, si se premultiplican ambos términos de (3.8) por $\mathbf{U}_{\text{GEV}}^{\top}$ y sabiendo que $\mathbf{W}_{\text{GEV}} = \mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}_{\text{GEV}}$ y $\mathbf{U}_{\text{GEV}}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}_{\text{GEV}} = \mathbf{I}$, se llega a que

$$\mathbf{W}_{\text{GEV}}^{\top}\mathbf{W}_{\text{GEV}} = \mathbf{\Lambda}_{\text{GEV}},$$

es decir, las columnas de $\mathbf{W}_{\text{GEV}}$ son ortogonales entre sí y sus normas al cuadrado son los correspondientes autovalores.

### 3.1.2. OPLS como problema de autovalores estándar: regresión de rango reducido

En la comunidad estadística, el problema de minimización de (3.1) es visto normalmente como un problema de regresión de rango reducido ("reduced-rank regression", RRR) llegando a un problema de autovalores estándar que proporciona una solución para la matriz de regresión $\mathbf{W}$ (Reinsel y Velu, 1998). Sin embargo, esta formulación no ha sido aplicada con tanta frecuencia en el campo de aprendizaje máquina, donde el objetivo es extraer la mayoría de las características relevantes de los datos de entrada (es decir, encontrar la matriz de proyección $\mathbf{U}$).

Para exponer esta solución, téngase en cuenta que para una matriz de regresión $\mathbf{W}$ dada, se puede obtener una solución cerrada para calcular $\mathbf{U}$. Para ello, primero se deriva (3.2) con respecto a $\mathbf{U}$:

$$\frac{\partial \mathcal{L}(\mathbf{U}, \mathbf{W})}{\partial \mathbf{U}} = -2\mathbf{C}_{\mathbf{XY}}\mathbf{W} + 2\mathbf{C}_{\mathbf{XX}}\mathbf{U}\mathbf{W}^\top\mathbf{W}.$$

Igualando estas derivadas a cero y despejando $\mathbf{U}$, se obtiene la siguiente expresión cerrada para calcular la matriz de proyección óptima asociada a cualquier $\mathbf{W}$ dada:

$$\mathbf{U} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}(\mathbf{W}^\top\mathbf{W})^{-1}. \tag{3.9}$$

Reemplazando esta expresión de nuevo en (3.2), y tras alguna manipulación algebraica, es posible expresar la función de coste OPLS (3.1) en términos de $\mathbf{W}$ solamente como:

$$\mathcal{L}(\mathbf{W}) = \mathrm{Tr}\{\mathbf{C}_{\mathbf{YY}}\} - \mathrm{Tr}\{(\mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top\mathbf{C}_{\mathbf{XY}}^\top\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}\}. \tag{3.10}$$

La minimización de $\mathcal{L}(\mathbf{W})$ puede llevarse a cabo resolviendo el siguiente problema de maximización con restricciones:

$$\begin{aligned} &\max_{\mathbf{W}} && \mathrm{Tr}\{\mathbf{W}^\top\mathbf{C}_{\mathbf{XY}}^\top\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}\} \\ &\text{sujeto a} && \mathbf{W}^\top\mathbf{W} = \mathbf{I}, \end{aligned} \tag{3.11}$$

cuya solución se puede obtener vía el problema de autovalores estándar:

$$\mathbf{C}_{\mathbf{XY}}^\top\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\boldsymbol{w} = \lambda\boldsymbol{w}. \tag{3.12}$$

De manera equivalente a la solución del subapartado anterior, se denota ahora como $\mathbf{\Lambda}_{\mathrm{EVD}}$ a la matriz diagonal que contiene los $n_f$ autovalores más altos de $\mathbf{C}_{\mathbf{XY}}^\top\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}$ dispuestos en orden decreciente, mientras que las columnas de $\mathbf{W}_{\mathrm{EVD}}$ corresponderán a los autovectores asociados. Al igual que antes, se debería notar que cualquier versión rotada de $\mathbf{W}_{\mathrm{EVD}}$ es también un mínimo de (3.10), pero $\mathbf{W}_{\mathrm{EVD}}$ tiene la propiedad de que cualquier subconjunto con las primeras $n_f' < n_f$ columnas sigue siendo la solución OPLS para el número de proyecciones seleccionado.

Usando (3.9), se pueden obtener los vectores de proyección asociados a la matriz de regresión $\mathbf{W}_{\mathrm{EVD}}$ como

$$\mathbf{U}_{\mathrm{EVD}} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}_{\mathrm{EVD}}, \tag{3.13}$$

donde se ha usado la ortogonalidad de $\mathbf{W}_{\mathrm{EVD}}$ ($\mathbf{W}_{\mathrm{EVD}}^\top\mathbf{W}_{\mathrm{EVD}} = \mathbf{I}$) para simplificar. Del mismo modo que con la solución clásica de OPLS, es posible mostrar que la solución derivada en este subapartado satisface también la

ortogonalidad de los datos proyectados. Para ver esto, primero se va a reescribir de forma matricial el problema de autovalores que satisface la matriz de regresión:

$$\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}_{\text{EVD}} = \mathbf{W}_{\text{EVD}}\mathbf{\Lambda}_{\text{EVD}}. \tag{3.14}$$

Ahora, premultiplicando ambos términos de (3.14) por $\mathbf{W}_{\text{EVD}}^{\top}$, se obtiene

$$\mathbf{W}_{\text{EVD}}^{\top}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\mathbf{W}_{\text{EVD}} = \mathbf{\Lambda}_{\text{EVD}}, \tag{3.15}$$

donde se ha usado de nuevo la ortogonalidad de las columnas de $\mathbf{W}_{\text{EVD}}$ para simplificar el término del lado derecho. Si además se observa que, de acuerdo con (3.13), $\mathbf{C}_{\mathbf{XY}}\mathbf{W}_{\text{EVD}} = \mathbf{C}_{\mathbf{XX}}\mathbf{U}_{\text{EVD}}$ y se sustituye en (3.15), se llega a

$$\mathbf{U}_{\text{EVD}}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}_{\text{EVD}} = \mathbf{\Lambda}_{\text{EVD}}, \tag{3.16}$$

demostrando así la condición de ortogonalidad de los datos de entrada proyectados.

Utilizando esta formulación, la solución OPLS se puede obtener en bloque (es decir, todos los vectores de proyección de $\mathbf{U}_{\text{EVD}}$ se calculan de una vez) resolviendo el problema de autovalores (3.14) seguido por (3.13). De hecho, se podría obtener de manera eficiente mediante los siguientes tres pasos:

1. $\mathbf{W}_{\text{LS}} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}$
2. $\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{W}_{\text{LS}}\mathbf{W}_{\text{EVD}} = \mathbf{W}_{\text{EVD}}\mathbf{\Lambda}_{\text{EVD}}$
3. $\mathbf{U}_{\text{EVD}} = \mathbf{W}_{\text{LS}}\mathbf{W}_{\text{EVD}}$,

donde, en el paso 1, $\mathbf{W}_{\text{LS}}$ es la solución al problema de mínimos cuadrados (LS): $\arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}^{\top}\mathbf{X}\|_F^2$.

O bien, se puede calcular secuencialmente los vectores de proyección $\boldsymbol{u}_k$ (es decir, las columnas de $\mathbf{U}_{\text{EVD}}$) iterando (para $k = 1, \ldots, n_f$) sobre los siguientes tres pasos:

P1) Obtener el autovector principal de la matriz $\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}$ simétrica, para conseguir el vector de coeficientes de regresión $\boldsymbol{w}_k$:

$$\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\boldsymbol{w}_k = \lambda_k\boldsymbol{w}_k.$$

El cálculo de $\boldsymbol{w}_k$ se puede implementar fácilmente, por ejemplo, usando el método de las potencias descrito en la Tabla 2.1.

P2) Obtener el correspondiente vector de proyección $\boldsymbol{u}_k$ mediante (3.13) particularizado para $n_f = 1$ y $\mathbf{W} = \boldsymbol{w}_k$, es decir,

$$\boldsymbol{u}_k = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\boldsymbol{w}_k \tag{3.17}$$

P3) Deflactar la matriz $\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}$ para eliminar la influencia del autovector $\boldsymbol{w}_k$ o —sabiendo que $\boldsymbol{w}_k$ es también el vector singular

izquierdo de $\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1/2}$— deflactar esta última matriz, por ejemplo, con el esquema de deflacción por proyección del Apartado 2.1.4.2:

$$\begin{aligned}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1/2} &\leftarrow (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^{\top})\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{C}_{\mathbf{XX}}^{-1/2}\\ \mathbf{Y} &\leftarrow (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^{\top})\mathbf{Y}.\end{aligned} \tag{3.18}$$

No obstante, en la literatura, el OPLS a menudo es deflactado de una manera intuitiva como la sustracción de la mejor predicción —en el sentido LS— que se puede obtener usando las actuales proyecciones de los datos de entrada, es decir,

$$\mathbf{Y} \leftarrow \mathbf{Y} - \boldsymbol{w}_k\boldsymbol{u}_k^{\top}\mathbf{X}, \tag{3.19}$$

que multiplicando por $\mathbf{X}^{\top}$ por la derecha, se obtiene un paso de deflacción que hace más eficiente la implementación del esquema iterativo:

$$\mathbf{C}_{\mathbf{XY}}^{\top} \leftarrow \mathbf{C}_{\mathbf{XY}}^{\top} - \boldsymbol{w}_k\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XX}}. \tag{3.20}$$

En este caso, si se sustituye la solución $\boldsymbol{u}_k$ de (3.17) en el paso de deflacción (3.20), se obtiene que (3.19) es equivalente al proceso de deflacción por proyección correspondiente al OPLS:

$$\begin{aligned}\mathbf{C}_{\mathbf{XY}}^{\top} &\leftarrow (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^{\top})\mathbf{C}_{\mathbf{XY}}^{\top}\\ \mathbf{Y} &\leftarrow (\mathbf{I}-\boldsymbol{w}_k\boldsymbol{w}_k^{\top})\mathbf{Y}.\end{aligned}$$

Esta formulación secuencial será la base para versiones secuenciales del OPLS con restricciones que se presentarán a lo largo de esta Parte II de la tesis doctoral.

### 3.1.3. Equivalencia entre las diferentes formulaciones del OPLS

Es fácil ver que, puesto que las soluciones a (3.5) y (3.11) representan distintos mínimos de la misma función de coste, deberían obtener el mismo valor de $\mathcal{L}(\mathbf{W},\mathbf{U})$. En este subapartado, se derivan las expresiones explícitas que demuestran la equivalencia entre las soluciones OPLS obtenidas, bien mediante la formulación GEV, $\{\mathbf{U}_{\text{GEV}},\mathbf{W}_{\text{GEV}}\}$, bien recurriendo al problema EVD, $\{\mathbf{U}_{\text{EVD}},\mathbf{W}_{\text{EVD}}\}$ (o RRR). Hasta donde llega nuestro conocimiento, esta conexión no ha sido establecida anteriormente, y es por ello que este apartado constituye una primera contribución de esta tesis doctoral.

Con el fin de simplificar esta exposición, se facilitan directamente las relaciones existentes entre las soluciones OPLS descritas en los subapartados

anteriores, dejándose su demostración para el Apéndice B:

$$\begin{aligned} \mathbf{\Lambda}_{\text{EVD}} &= \mathbf{\Lambda}_{\text{GEV}} \quad (= \mathbf{\Lambda}), \\ \mathbf{U}_{\text{EVD}} &= \mathbf{U}_{\text{GEV}}\mathbf{\Lambda}^{1/2}, \\ \mathbf{W}_{\text{EVD}} &= \mathbf{W}_{\text{GEV}}\mathbf{\Lambda}^{-1/2}. \end{aligned} \tag{3.21}$$

De esta manera y puesto que $\mathbf{\Lambda}$ es diagonal, estos resultados implican que las columnas de $\mathbf{U}_{\text{GEV}}$ y $\mathbf{U}_{\text{EVD}}$ tienen la misma dirección y se diferencian únicamente en un factor de escala.

La Tabla 3.1 resume las principales ecuaciones y propiedades de las dos soluciones alternativas del OPLS revisadas, a las cuales se hará referencia en lo sucesivo como GEV-OPLS y EVD-OPLS.

Tabla 3.1: Ecuaciones y propiedades más relevantes de las soluciones GEV y EVD-OPLS

| | **GEV-OPLS (Subsec. 3.1.1)** | **EVD-OPLS (Subsec. 3.1.2)** |
|---|---|---|
| Prob. de autovalores | $\mathbf{C_{XY}C_{XY}^\top U}_{\text{GEV}} = \mathbf{C_{XX}U}_{\text{GEV}}\mathbf{\Lambda}$ (dimensión $n$) | $\mathbf{C_{XY}^\top C_{XX}^{-1}C_{XY}W}_{\text{EVD}} = \mathbf{W}_{\text{EVD}}\mathbf{\Lambda}$ (dimensión $m$) |
| Condición de ortogonalidad | $\mathbf{U}_{\text{GEV}}^\top\mathbf{C_{XX}U}_{\text{GEV}} = \mathbf{I}$<br>$\mathbf{W}_{\text{GEV}}^\top\mathbf{W}_{\text{GEV}} = \mathbf{\Lambda}$ | $\mathbf{U}_{\text{EVD}}^\top\mathbf{C_{XX}U}_{\text{EVD}} = \mathbf{\Lambda}$<br>$\mathbf{W}_{\text{EVD}}^\top\mathbf{W}_{\text{EVD}} = \mathbf{I}$ |
| Relación entre $\mathbf{U}$ y $\mathbf{W}$ | $\mathbf{W}_{\text{GEV}} = \mathbf{C_{XY}^\top U}_{\text{GEV}}$ | $\mathbf{U}_{\text{EVD}} = \mathbf{C_{XX}^{-1}C_{XY}W}_{\text{EVD}}$ |

Aunque la formulación GEV-OPLS ha sido típicamente usada en artículos de aprendizaje automático, es argumentable que la formulación EVD-OPLS ofrece algunas ventajas importantes también en este contexto. En particular, las principales ventajas de EVD-OPLS que serán explotadas en los siguientes apartados son:

- La dimensión de los problemas de autovalores (3.8) y (3.14) son $n$ y $m$ respectivamente. Esto significa que EVD-OPLS es computacionalmente más eficiente para el caso más común $m < n$ (es decir, el número de variables objetivo es menor que la dimensionalidad de los datos de entrada).

- EVD-OPLS facilita la introducción de restricciones sobre la matriz de proyección. Puesto que $\mathbf{U}_{\text{EVD}}$ es la solución de un problema de mínimos cuadrados, se pueden imponer restricciones adicionales fácilmente modificando (3.1). Por ejemplo, se podría favorecer dispersión sobre los vectores de proyección añadiendo un término de penalización "lasso". Sin embargo, incluir restricciones sobre los vectores de proyección de GEV-OPLS no parece tan obvio, ya que $\mathbf{U}_{\text{GEV}}$ es obtenido como la solución del problema de autovalores generalizado (3.8). Nótese que

el hecho de obtener vectores de proyección dispersos facilita la interpretación de la solución y, en casos extremos —donde algunas filas son todo ceros—, lleva a selección de variables de entrada; mientras que no habría una ventaja obvia si se impusiera dispersión sobre la matriz de coeficientes de regresión $\mathbf{W}$ (que podría ser más fácilmente implementado usando la formulación GEV-OPLS).

En el siguiente subapartado, se compara la complejidad computacional de las formulaciones GEV-OPLS y EVD-OPLS. Como se verá en los siguientes capítulos, las propuestas de esta tesis doctoral se basarán en la formulación EVD-OPLS para derivar soluciones dispersas tanto para casos lineales como no lineales, así como soluciones para selección de variables —o parsimoniosas— y soluciones no-negativas.

### 3.1.4. Análisis del coste computacional

Para comparar las necesidades computacionales de GEV-OPLS y EVD-OPLS, en este subapartado se realiza una comparación empírica de complejidad computacional de las dos soluciones. Para hacer una comparación justa, primero se calcula la solución de mínimos cuadrados del problema de regresión, $\mathbf{W}_{\text{LS}} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}$, que tiene una complejidad computacional de $\mathcal{O}(n^3)$. De esta manera, (3.8) y (3.14) se pueden reescribir como los siguientes problemas de autovalores estándar:

$$\text{GEV-OPLS}: \ \mathbf{W}_{\text{LS}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} \tag{3.22}$$

$$\text{EVD-OPLS}: \ \mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{W}_{\text{LS}}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}. \tag{3.23}$$

Como ya se ha discutido, las formulaciones GEV y EVD requieren matrices de tamaño $n \times n$ y $m \times m$ respectivamente, implicando problemas de autovalores con complejidad $\mathcal{O}(n^3)$ y $\mathcal{O}(m^3)$ para GEV-OPLS y EVD-OPLS, respectivamente. Nótese que, una vez el problema de autovalores de EVD-OPLS es resuelto, la matriz de proyección puede ser directamente calculada como $\mathbf{U}_{\text{EVD}} = \mathbf{W}_{\text{LS}}\mathbf{W}_{\text{EVD}}$.

Con el fin de ilustrar cómo las necesidades computacionales escalan para ambos métodos, se ha creado un problema artificial de acuerdo al siguiente modelo de regresión:

$$\mathbf{Y} = \sin(\pi\mathbf{M}\mathbf{X} + 1) + \boldsymbol{\Xi},$$

donde $\mathbf{X} \in \mathbb{R}^{n\times N}$ y $\boldsymbol{\Xi} \in \mathbb{R}^{m\times N}$ son matrices que contienen los datos de entrada y el ruido sobre las observaciones. Los elementos de estas matrices se generan de forma independiente a partir de distribuciones normales con media cero y desviación estándar 0,7 y $5 \cdot 10^{-2}$, respectivamente para $\mathbf{X}$ y $\boldsymbol{\Xi}$. Por último, $\mathbf{M} \in \mathbb{R}^{m\times n}$ es una matriz que contiene los parámetros del modelo, que son tomados independientemente de una distribución uniforme entre 0 y 1.

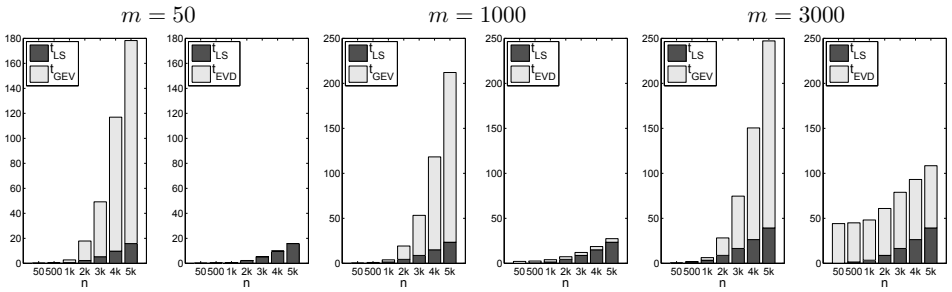



Figura 3.1: Tiempo en segundos requerido por las implementaciones GEV-OPLS (3.22) y EVD-OPLS (3.23). Las subfiguras muestran el tiempo requerido para el cálculo del modelo de regresión de mínimos cuadrados ($t_{\text{LS}}$) y para la solución de los problemas de autovalores estándar y generalizado ($t_{\text{GEV}}$ y $t_{\text{EVD}}$ respectivamente) para $N = 5000$ y diferentes valores de $n$ y $m$.

La Figura 3.1 muestra los tiempos de ejecución de GEV-OPLS y EVD-OPLS para $N = 5000$ y distintos valores de $m$ y $n$. Todos los experimentos han sido ejecutados sobre un ordenador Intel Core i7 CPU 870 con 2.93 GHz y 8 GB de RAM. Como se esperaba, el tiempo computacional de GEV-OPLS crece muy rápidamente con $n$, mientras que el tiempo de ejecución de EVD-OPLS muestra únicamente un ligero incremento, principalmente debido al tiempo adicional requerido para calcular $\mathbf{W}_{\text{LS}}$. Se observa el comportamiento opuesto cuando aumenta la dimensionalidad de salida $m$. Estos resultados respaldan la conclusión de que EVD-OPLS es una implementación más eficiente para el caso común en que la dimensionalidad de entrada excede el número de variables de salida (es decir, $n > m$).

## 3.2. Marco general MVA

En este apartado, se presenta un marco general para aquellos métodos MVA que fuerzan ortogonalidad en las características extraídas. La implementación de estos métodos está basada en el uso del problema de autovalores estándar como hace el modelo RRR, de tal modo que resulta eficiente computacionalmente en el caso común donde el número de variables de entrada es mayor que el de salida.

Para ello, se incluye una matriz definida positiva $\mathbf{\Omega}$ en la función de coste (3.1), permitiendo obtener una formulación MVA generalizada, del siguiente modo (véase el libro de Reinsel y Velu, 1998, para un estudio más detallado

sobre esta formulación):

$$\begin{aligned}\mathcal{L}(\mathbf{W},\mathbf{U}) &= \|\mathbf{\Omega}^{\frac{1}{2}}(\mathbf{Y}-\mathbf{W}\mathbf{U}^\top\mathbf{X})\|_F^2, \\ &= \operatorname{Tr}\{(\mathbf{Y}-\mathbf{W}\mathbf{U}^\top\mathbf{X})^\top\mathbf{\Omega}(\mathbf{Y}-\mathbf{W}\mathbf{U}^\top\mathbf{X})\} \\ &= \operatorname{Tr}\{\mathbf{Y}^\top\mathbf{\Omega}\mathbf{Y}\} - 2\operatorname{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W}\} + \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\mathbf{W}^\top\mathbf{\Omega}\mathbf{W}\},\end{aligned} \tag{3.24}$$

donde se fuerza también la siguiente condición de ortogonalidad sobre los vectores de proyección de salida:

$$\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}.$$

Se puede ver que, cuando $\mathbf{\Omega} = \mathbf{I}$, esta condición es $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ y la solución obtenida es justo el OPLS. Otros valores de $\mathbf{\Omega}$ darán lugar a otras versiones de métodos MVA —como el PCA y el CCA—.

Derivando la función de coste con respecto a $\mathbf{U}$,

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = \mathbf{C_{XX}}\mathbf{U}(\mathbf{W}^\top\mathbf{\Omega}\mathbf{W}) - \mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W},$$

e igualando esta derivada a cero, se obtiene la matriz de proyección de entrada óptima:

$$\mathbf{U} = \mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W}(\mathbf{W}^\top\mathbf{\Omega}\mathbf{W})^{-1} \tag{3.25}$$

Ahora, sustituyendo $\mathbf{U}$ dentro de (3.24), la función de coste quedaría únicamente en función de $\mathbf{W}$,

$$\mathcal{L}(\mathbf{W}) = \operatorname{Tr}\{\mathbf{\Omega}\mathbf{C_{YY}}\} - \operatorname{Tr}\{(\mathbf{W}^\top\mathbf{\Omega}\mathbf{W})^{-1}\mathbf{W}^\top\mathbf{\Omega}\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W}\},$$

que puede ser reescrita como el siguiente problema de maximización con restricciones:

$$\begin{aligned}\max_{\mathbf{W}} \quad & \operatorname{Tr}\{\mathbf{W}^\top\mathbf{\Omega}\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W}\} \\ \text{sujeto a} \quad & \mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I},\end{aligned} \tag{3.26}$$

Si ahora se deriva la función de coste con respecto a $\mathbf{W}$ teniendo en cuenta la restricción de ortogonalidad $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$, se puede obtener el siguiente problema de autovalores generalizado,

$$\mathbf{\Omega}\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W} = \mathbf{\Omega}\mathbf{W}\mathbf{\Lambda} \tag{3.27}$$

que haciendo el cambio de variable $\mathbf{W} = \mathbf{\Omega}^{-\frac{1}{2}}\mathbf{V}$, se convierte en el siguiente problema de autovalores estándar:

$$\mathbf{\Omega}^{\frac{1}{2}}\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}^{\frac{1}{2}}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}. \tag{3.28}$$

De este modo, teniendo en cuenta que se cumple $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$ y partiendo de (3.25), $\mathbf{U}$ se puede expresar como

$$\mathbf{U} = \mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{W} \tag{3.29}$$

o, en función de $\mathbf{V}$, como:

$$\mathbf{U} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{V}, \tag{3.30}$$

pudiéndose obtener la solución de este marco general MVA en bloque de manera eficiente con los siguientes cuatro pasos:

1. $\mathbf{C}_{\mathbf{XY}'} = \mathbf{C}_{\mathbf{XY}}\boldsymbol{\Omega}^{\frac{1}{2}}$
2. $\mathbf{W}_{\text{LS}} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}'}$
3. $\mathbf{C}_{\mathbf{XY}'}^{\top}\mathbf{W}_{\text{LS}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$
4. $\mathbf{U} = \mathbf{W}_{\text{LS}}\mathbf{V}$,

siendo $\mathbf{W}_{\text{LS}}$ la solución al problema LS: $\arg\min_{\mathbf{W}} \|\boldsymbol{\Omega}^{\frac{1}{2}}(\mathbf{Y} - \mathbf{W}^{\top}\mathbf{X})\|_F^2$.

### 3.2.1. Ortogonalidad de las características extraídas

Como se comentó al principio de este apartado, este marco es válido para los métodos MVA que fuerzan ortogonalidad de las características extraídas. Aunque en esta formulación no se ha incluido dicha restricción de manera explícita, sí que se ha hecho de manera implícita; en otras palabras: se puede demostrar que la condición de ortogonalidad para los datos de entrada proyectados ($\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{I}$) se cumple si y solo si se tiene la condición $\mathbf{W}^{\top}\boldsymbol{\Omega}\mathbf{W} = \boldsymbol{\Lambda}_{\text{I}}$, siendo $\boldsymbol{\Lambda}_{\text{I}}$ cualquier matriz diagonal.

Formalmente, en este subapartado, se demostrará que la ortogonalidad de $\mathbf{V} = \boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{W}$ es condición necesaria y suficiente para conseguir ortogonalidad sobre los datos proyectados:

$$\mathbf{W}^{\top}\boldsymbol{\Omega}\mathbf{W} = \boldsymbol{\Lambda}_{\text{I}} \iff \mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \boldsymbol{\Lambda}_{\text{II}}, \tag{3.31}$$

siendo $\boldsymbol{\Lambda}_{\text{I}}$ y $\boldsymbol{\Lambda}_{\text{II}}$, cualquier matriz diagonal y, en particular, la matriz identidad[1] —cuando la condición es impuesta como restricción— o la matriz de autovalores $\boldsymbol{\Lambda}$ —cuando la condición es obtenida como consecuencia de la restricción impuesta—.

[1] Cuando $\boldsymbol{\Lambda}_{\text{II}}$ es la matriz identidad, se dice que las variables proyectadas tienen varianza unidad y que, por lo tanto, están blanqueadas. No obstante, puesto que los elementos de la diagonal de $\boldsymbol{\Lambda}_{\text{I}}$ se pueden calcular fácilmente para poder blanquear las variables mediante un reescalado, resulta suficiente con forzar que $\boldsymbol{\Lambda}_{\text{II}}$ sea diagonal. Debido a esto, se abusará de la terminología y se usarán indistintamente los conceptos de incorrelación, ortogonolidad y blanqueamiento, donde incorrelación y ortogonalidad es lo mismo al considerar que los datos están centrados.

#### 3.2.1.1. Condición suficiente: $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{\Lambda_I} \implies \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda_{II}}$

Para esta demostración, se puede partir de la ecuación (3.27) aplicando la solución (3.29) en ella:

$$\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Omega}\mathbf{W}\mathbf{\Lambda}.$$

Multiplicando por la izquierda por $\mathbf{W}^\top$ a ambos lados de la ecuación, se puede ver que, si se cumple $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{\Lambda}_\mathrm{I}$, se obtiene:

$$\mathbf{W}^\top\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Lambda}_\mathrm{II}(= \mathbf{\Lambda}_\mathrm{I}\mathbf{\Lambda}). \tag{3.32}$$

Por otro lado, si se multiplica por $\mathbf{U}^\top\mathbf{C_{XX}}$ a ambos lados de la ecuación (3.29) por la izquierda y se incluye la solución de (3.32):

$$\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{W}^\top\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Lambda}_\mathrm{II}, \tag{3.33}$$

se concluye que las características extraídas son ortogonales.

Esto resulta útil en la formulación EVD, pues se impone $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$. En este caso, se cumple:

$$\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I} \implies \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda}.$$

#### 3.2.1.2. Condición necesaria: $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{\Lambda_I} \impliedby \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda_{II}}$

Por otro lado, para demostrar la condición necesaria, habría que realizar el procedimiento seguido por la solución GEV del OPLS; es decir, sustituir la solución óptima de $\mathbf{W}$ en la función de coste y derivar con respecto a $\mathbf{U}$, mientras se fuerza $\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda}_\mathrm{II}$. Por lo tanto, si se deriva (3.24) con respecto a $\mathbf{W}$,

$$\frac{\partial\mathcal{L}(\mathbf{W},\mathbf{U})}{\partial\mathbf{W}} = \mathbf{\Omega}\mathbf{W}\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} - \mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U},$$

se iguala a cero y se fuerza $\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda}_\mathrm{II}$, se obtiene:

$$\mathbf{\Omega}\mathbf{W}_\mathrm{GEV} = \mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}_\mathrm{GEV}\mathbf{\Lambda}_\mathrm{II}^{-1}. \tag{3.34}$$

Sustituyendo esta solución en (3.24), la expresión a resolver resultaría en el siguiente problema de maximización:

$$\begin{array}{ll} \underset{\mathbf{U}}{\text{máx}} & \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\mathbf{\Lambda}_\mathrm{II}^{-1}\} \\ \text{sujeto a} & \mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{\Lambda}_\mathrm{II}, \end{array} \tag{3.35}$$

que, si se reescribe introduciendo multiplicadores de Lagrange, la función de coste a maximizar quedaría solamente en función de $\mathbf{U}$ como:

$$\mathcal{L}(\mathbf{U}) = \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\mathbf{\Lambda}_\mathrm{II}^{-1}\} - \mathrm{Tr}\{(\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} - \mathbf{\Lambda}_\mathrm{II})\mathbf{\Lambda}\}$$

Al ser $\mathbf{\Lambda}_{\mathrm{II}}$ una matriz diagonal, el problema (3.35) podría resolverse mediante el siguiente problema de autovalores generalizado:

$$\mathbf{C_{XY}\Omega C_{XY}^{\top}U} = \mathbf{C_{XX}U\Lambda}_{\mathrm{I}},$$

y, por lo tanto, la matriz de coeficientes de Lagrange $\mathbf{\Lambda}$ también sería diagonal, siendo $\mathbf{\Lambda}_{\mathrm{I}} = \mathbf{\Lambda\Lambda}_{\mathrm{II}}$ la versión reescalada de la matriz de autovalores $\mathbf{\Lambda}$. Ahora, multiplicando por $\mathbf{U}^{\top}$ a ambos lados de la ecuación por la izquierda y aplicando (3.34), se obtiene:

$$\mathbf{W}^{\top}\mathbf{\Omega C_{XY}^{\top}U} = \mathbf{\Lambda}_{\mathrm{I}}.$$

Por último, si se multiplica por $\mathbf{W}^{\top}$ a ambos lados de la ecuación (3.34) por la izquierda, se puede ver la relación:

$$\mathbf{W}^{\top}\mathbf{\Omega W} = \mathbf{W}^{\top}\mathbf{\Omega C_{XY}^{\top}U} = \mathbf{\Lambda}_{\mathrm{I}}, \tag{3.36}$$

concluyendo que la condición de incorrelación de $\mathbf{W}^{\top}\mathbf{\Omega W}$ es condición necesaria y suficiente para obtener la deseada ortogonalidad de las características extraídas.

En este caso, esta condición resulta útil en la formulación GEV, donde se impone $\mathbf{U}^{\top}\mathbf{C_{XX}U} = \mathbf{I}$. Por lo tanto, se confirma:

$$\mathbf{U}^{\top}\mathbf{C_{XX}U} = \mathbf{I} \implies \mathbf{W}^{\top}\mathbf{\Omega W} = \mathbf{\Lambda}$$

#### 3.2.1.3. Conclusiones de la condición necesaria y suficiente

Resulta interesante analizar de manera conjunta los resultados (3.33) y (3.36) correspondientes a las condiciones suficiente y necesaria respectivamente:

$$\begin{aligned}\mathbf{W}^{\top}\mathbf{\Omega W} = \mathbf{I} &\implies \mathbf{U}^{\top}\mathbf{C_{XX}U} = \mathbf{W}^{\top}\mathbf{\Omega C_{XY}^{\top}U} = \mathbf{\Lambda}\\ \mathbf{U}^{\top}\mathbf{C_{XX}U} = \mathbf{I} &\implies \mathbf{W}^{\top}\mathbf{\Omega W} = \mathbf{W}^{\top}\mathbf{\Omega C_{XY}^{\top}U} = \mathbf{\Lambda}.\end{aligned}$$

Como se puede observar, la condición necesaria y suficiente (3.31) se cumple en ambos sentidos gracias a la *condición de incorrelación*

$$\mathbf{W}^{\top}\mathbf{\Omega C_{XY}^{\top}U} = \mathbf{\Lambda}. \tag{3.37}$$

Como se verá en el siguiente apartado, cuando se adapte la formulación de este marco general a una versión iterativa, no será suficiente la condición de ortogonalidad $\mathbf{W}^{\top}\mathbf{\Omega W} = \mathbf{I}$ para conseguir el blanqueado de los datos de entrada, debiéndose usar la condición de incorrelación para forzar dicho blanqueado.

### 3.2.2. CCA como caso particular supervisado

Aunque ya se ha derivado la formulación CCA en el subapartado 2.2.3, aquí se realiza una derivación distinta con el fin de demostrar que CCA es un caso particular del marco general MVA que se acaba de revisar. Nótese que para el caso OPLS, esta demostración no es necesaria, pues ya se realizó dicha derivación en el apartado 3.1.2 y, como se comentó anteriormente, su solución se obtiene simplemente sustituyendo $\mathbf{\Omega} = \mathbf{I}$ en el marco general MVA.

Siguiendo el mismo procedimiento de derivación del subapartado anterior, se puede obtener también una solución eficiente para CCA partiendo de la función de coste descrita en (CCA.3) como:

$$\begin{aligned}\mathcal{L}(\mathbf{W},\mathbf{U}) &= \|\mathbf{W}^\top\mathbf{Y} - \mathbf{U}^\top\mathbf{X}\|_F^2, \\ &= \operatorname{Tr}\{\Big(\mathbf{W}^\top\mathbf{Y} - \mathbf{U}^\top\mathbf{X}\Big)^\top \Big(\mathbf{W}^\top\mathbf{Y} - \mathbf{U}^\top\mathbf{X}\Big)\} \\ &= \operatorname{Tr}\{\mathbf{W}^\top\mathbf{C_{YY}}\mathbf{W}\} - 2\operatorname{Tr}\{\mathbf{U}^\top\mathbf{C_{XY}}\mathbf{W}\} + \operatorname{Tr}\{\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\}.\end{aligned} \tag{3.38}$$

Del mismo modo que antes, si se minimiza con respecto a $\mathbf{U}$, es decir, haciendo la derivada

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = \mathbf{C_{XX}}\mathbf{U} - \mathbf{C_{XY}}\mathbf{W},$$

igualando a cero y despejando, se obtiene que

$$\mathbf{U} = \mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{W}.$$

Ahora, sustituyendo $\mathbf{U}$ dentro de (3.38), se obtiene la función de coste solamente en función de $\mathbf{W}$:

$$\mathcal{L}(\mathbf{W}) = \operatorname{Tr}\{\mathbf{W}^\top\mathbf{C_{YY}}\mathbf{W}\} - \operatorname{Tr}\{\mathbf{W}^\top\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{W}\}.$$

Si a continuación se deriva esta función de coste con respecto a $\mathbf{W}$, teniendo en cuenta la restricción de ortogonalidad impuesta por CCA ($\mathbf{W}^\top\mathbf{C_{YY}}\mathbf{W} = \mathbf{I}$), se obtiene el siguiente problema de autovectores generalizado,

$$\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{W}_{\text{CCA}} = \mathbf{C_{YY}}\mathbf{W}_{\text{CCA}}\mathbf{\Lambda}_{\text{CCA}},$$

que haciendo el cambio de variable $\mathbf{W}_{\text{CCA}} = \mathbf{C_{YY}^{-\frac{1}{2}}}\mathbf{V}_{\text{CCA}}$, se convierte en el siguiente problema de autovalores estándar:

$$\mathbf{C_{YY}^{-\frac{1}{2}}}\mathbf{C_{XY}^\top}\mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{C_{YY}^{-\frac{1}{2}}}\mathbf{V}_{\text{CCA}} = \mathbf{V}_{\text{CCA}}\mathbf{\Lambda}_{\text{CCA}}. \tag{3.39}$$

Por último, realizando también el cambio de variable en $\mathbf{U}$, esta solución se puede expresar en función de $\mathbf{V}$ como:

$$\mathbf{U}_{\text{CCA}} = \mathbf{C_{XX}^{-1}}\mathbf{C_{XY}}\mathbf{C_{YY}^{-\frac{1}{2}}}\mathbf{V}. \tag{3.40}$$

Tabla 3.2: Tabla comparativa entre el algoritmo CCA con respecto el marco general MVA

| | CCA | Marco general MVA |
|---|---|---|
| $\mathbf{V}$: | $\mathbf{C_{YY}^{-\frac{1}{2}}C_{XY}^{\top}C_{XX}^{-1}C_{XY}C_{YY}^{-\frac{1}{2}}V} = \mathbf{V\Lambda}$ | $\mathbf{\Omega^{\frac{1}{2}}C_{XY}^{\top}C_{XX}^{-1}C_{XY}\Omega^{\frac{1}{2}}V} = \mathbf{V\Lambda}$ |
| $\mathbf{U}$: | $\mathbf{U = C_{XX}^{-1}C_{XY}C_{YY}^{-\frac{1}{2}}V}$ | $\mathbf{U = C_{XX}^{-1}C_{XY}\Omega^{\frac{1}{2}}V}$ |

Es interesante comparar estas soluciones (3.39) y (3.40) con las soluciones (3.28) y (3.30) (esta comparación se puede ver fácilmente en la Tabla 3.2), donde se puede observar que CCA es un caso particular de este marco general MVA cuando $\mathbf{\Omega = C_{YY}^{-1}}$. No obstante, habría que tener cuidado si se quiere usar $\mathbf{W}$ en lugar de $\mathbf{V}$, ya que el cambio no es el mismo que en el marco general MVA ($\mathbf{W = \Omega^{-\frac{1}{2}}V}$) sino que ahora $\mathbf{W = \Omega^{\frac{1}{2}}V = C_{YY}^{-\frac{1}{2}}V}$.

Por lo tanto, con esta misma formulación, se puede obtener OPLS o CCA en función de $\mathbf{\Omega}$ sustituyendo simplemente $\mathbf{\Omega = I}$ o $\mathbf{\Omega = C_{YY}}$, respectivamente.

### 3.2.3. PCA como caso particular no supervisado

Resulta sencillo mostrar que PCA es un caso particular de este marco MVA generalizado, ya que si se compara la función de coste definida en (PCA.1) con (3.24), simplemente habría que sustituir $\mathbf{Y}$ por $\mathbf{X}$, pues la matriz de salida es también la de entrada, y sustituir $\mathbf{\Omega = I}$ igual que se hace para OPLS.

Nótese también que ya no existiría la matriz de regresión $\mathbf{W}$, ya que es la misma matriz de reconstrucción $\mathbf{U}$ y, por tanto, únicamente sería necesario calcular una de las matrices. Además, como se requiere la condición de ortogonalidad de $\mathbf{U}$, necesariamente habría que calcular la solución correspondiente al problema de autovalores estándar que, haciendo las sustituciones correspondientes, sería:

$$\begin{aligned}\mathbf{C_{XX}C_{XX}^{-1}C_{XX}U} &= \mathbf{U\Lambda}\\ \mathbf{C_{XX}U} &= \mathbf{U\Lambda},\end{aligned}$$

es decir, la solución PCA original (PCA.3).

Aunque, en este caso particular, la solución EVD no aporta ventaja computacional alguna, su uso resultará interesante en formulaciones donde se desee imponer restricciones sobre la matriz de proyección $\mathbf{U}$.

### 3.2.4. Conclusiones del marco general MVA

En este apartado, se ha generalizado mediante una matriz genérica $\mathbf{\Omega}$ la formulación eficiente del OPLS para el resto de métodos MVA que extraen

características incorreladas. Además, se ha demostrado que al imponer la condición de ortogonalidad $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$, se obtiene el blanqueamiento de los datos de entrada requerido gracias a la condición de incorrelación (3.37), donde se obtiene que los datos de entrada proyectados por $\mathbf{U}$ son ortogonales a los datos de salida proyectados por $\mathbf{W}$.

Nótese que para el caso del PCA, donde la salida es la entrada ($\mathbf{Y} = \mathbf{X}$) y, por lo tanto, los vectores de proyección de salida son los mismos que los de entrada ($\mathbf{W} = \mathbf{U}$), se demuestra que la incorrelación de los vectores de proyección —ya que $\mathbf{\Omega} = \mathbf{I}$ para el PCA (véase el subapartado 3.2.3)— es condición necesaria y suficiente para obtener la ortogonalidad de los datos proyectados.

Por lo tanto, una conclusión de este marco general es que, al igual que la condición $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$ es la herramienta usada por el PCA para conseguir la incorrelación de las características extraídas, la restricción $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$ es la herramienta empleada por este marco MVA para obtener dicha propiedad. Además, este procedimiento de obtener la ortogonalidad de los datos de entrada proyectados resulta ser el modo eficiente cuando el número de variables de salida es menor que el de entrada ($m < n$) —como se discutió en el Subapartado 3.1.4—.

## 3.3. Solución iterativa MVA con restricciones

En el primer apartado de este capítulo, se analizaron dos formulaciones distintas para resolver el problema OPLS. Una de ellas incluye de manera explícita en su formulación los objetivos deseados: la función de coste y la incorrelación de los datos proyectados ($\mathbf{U}^\top \mathbf{C}_{\mathbf{XX}} \mathbf{U} = \mathbf{I}$); y la otra alcanza dichos objetivos de manera más eficiente, pero indirecta. Es decir, formula un problema aparentemente distinto, pues está sujeto a $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, pero ciertamente equivalente, ya que, como se ha demostrado, es condición necesaria y suficiente para obtener incorrelación de los datos proyectados ($\mathbf{U}^\top \mathbf{C}_{\mathbf{XX}} \mathbf{U} = \mathbf{\Lambda}$).

En el segundo apartado de este capítulo, se tomó esta formulación eficiente y se creó un marco general válido para los métodos MVA que blanquean los datos de entrada. En este caso, la restricción impuesta por todos ellos es $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$ y la diferencia entre los distintos métodos viene dada por la selección de la matriz $\mathbf{\Omega}$.

No obstante, la aplicación de los métodos MVA en problemas de la vida real requiere, a menudo, la incorporación de restricciones adicionales sobre los vectores de proyección. Por ello, en este apartado, se extiende la generalización del apartado anterior a un marco general MVA que permite la incorporación de restricciones. Para ello, se modifica la función objetivo in-

cluyendo dichas restricciones:

$$\mathcal{L}(\mathbf{W},\mathbf{U}) = \left\| \mathbf{\Omega}^{\frac{1}{2}} \left( \mathbf{Y} - \mathbf{W}\mathbf{U}^\top \mathbf{X} \right) \right\|_F^2 + \gamma R\left(\mathbf{U}\right), \tag{3.41}$$

donde $R\left(\mathbf{U}\right)$ es la restricción añadida o término de regularización y $\gamma$ es el factor de penalización que permite controlar la importancia de la regularización frente a la función de coste original.

Habitualmente, estas restricciones no son derivables y, por consiguiente, la solución al problema (3.41) no tiene una forma cerrada. La solución usada por defecto en la literatura consiste en una formulación iterativa, donde se divide la función objetivo (3.41) en dos problemas acoplados (véase este proceso iterativo resumido en la Tabla 3.3):

1) Paso$-\mathbf{U}$: Fijando $\mathbf{W}$ y, considerando $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$, se llega al siguiente problema de mínimos cuadrados regularizado:

$$\underset{\mathbf{U}}{\arg\min} \quad \|\bar{\mathbf{Y}} - \mathbf{U}^\top \mathbf{X}\|_F^2 + \gamma R\left(\mathbf{U}\right), \tag{3.42}$$

   donde, al ser $\mathbf{W}$ constante, se ha multiplicado por $\mathbf{W}^\top \mathbf{\Omega}$ a la función de coste (3.41) por la izquierda y se ha definido $\bar{\mathbf{Y}} = \mathbf{W}^\top \mathbf{\Omega} \mathbf{Y}$ como la matriz de los datos de salida proyectados. Resulta interesante definir este paso, puesto que se puede aprovechar una gran variedad de soluciones eficientes ya existentes, y muy bien estudiadas, para distintos problemas de mínimos cuadrados regularizados.

2) Paso$-\mathbf{W}$: Fijando $\mathbf{U}$, se minimiza la función de coste (3.24) sujeto a $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$, es decir,

$$\begin{array}{ll} \underset{\mathbf{W}}{\arg\min} & \|\mathbf{\Omega}^{\frac{1}{2}} \left( \mathbf{Y} - \mathbf{W}\bar{\mathbf{X}} \right) \|_F^2 \\ \text{sujeto a} & \mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I} \end{array} \tag{3.43}$$

   donde $\bar{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$ sería la matriz de los datos de entrada proyectados.

Resulta importante destacar que esta formulación iterativa proviene de la división en dos pasos de la formulación MVA general que permitía imponer $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$ para obtener proyecciones blanqueadas. Como se verá, será un punto clave de este apartado analizar si $\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} = \mathbf{I}$ sigue siendo condición necesaria y suficiente para el blanqueado y, en función de ello, definir la solución de los Pasos$-\mathbf{U}$ y $-\mathbf{W}$. Esto marcará la diferencia entre los soluciones existentes hasta el momento en la literatura y la propuesta que se presentará en este capítulo.

Inicialmente, este proceso iterativo fue propuesto por Zou et al. (2006) para el PCA disperso ("sparse PCA") y en ella se resuelve el Paso$-\mathbf{W}$ mediante la aproximación ortogonal de Procrustes ("orthogonal Procrustes problem", estudiada por Schönemann, 1966), que, si bien es capaz de obtener

Tabla 3.3: Pseudocódigo del proceso iterativo para el marco general MVA con restricciones

1.- Entradas: matrices positivas $\mathbf{X}$ y $\mathbf{Y}$.
  2.1.- Inicializar $\mathbf{W}^{(0)}$.
  2.2.- Para $i = 1, 2, \ldots$
    2.2.1.- Paso$-\mathbf{U}$: Obtener $\mathbf{U}^{(i)}$ resolviendo el problema (3.42).
    2.2.2.- Paso$-\mathbf{W}$: Obtener $\mathbf{W}^{(i)}$ resolviendo el problema (3.43).
    2.2.3.- Si se cumple el criterio de convergencia, ir a 3.
3.- Salidas: $\mathbf{U}$, $\mathbf{W}$.

el mínimo de (3.43), descuida la incorrelación de las características extraídas —como se demostrará más adelante—. A pesar de ello, otros autores han seguido por defecto esta aproximación y han extendido erróneamente otros métodos MVA a aproximaciones supervisadas dispersas tales como OPLS disperso (van Gerven et al., 2012), group-lasso OPLS (propuesto como SRRR por Chen y Huang, 2012) o CCA con regularización $\ell_{2,1}$ (propuesto como L21SDA por Shi et al., 2014).

Formalmente, si se quiere obtener la matriz deseada $\mathbf{M} \in \mathbb{R}^{n\times m}$ dadas las matrices $\mathbf{B} \in \mathbb{R}^{m\times N}$ y $\mathbf{A} \in \mathbb{R}^{n\times N}$, el problema ortogonal de Procrustes se define como:

$$\begin{array}{ll} \underset{\mathbf{M}}{\arg\min} & \|\mathbf{B} - \mathbf{M}^\top \mathbf{A}\|_F^2, \\ \text{sujeto a} & \mathbf{M}^\top \mathbf{M} = \mathbf{I} \end{array} \tag{3.44}$$

que, a partir de la descomposición de valores singulares $\mathbf{A}\mathbf{B}^\top = \mathbf{Q}\mathbf{\Sigma}\mathbf{P}^\top$, tiene como solución $\hat{\mathbf{M}} = \mathbf{Q}\mathbf{P}^\top$.

Tal y como se demuestra a continuación, esta aproximación de Procrustes tiene dos problemas clave que la hacen carecer de las propiedades y, por lo tanto, de las habilidades de los métodos MVA.

Una vez presentadas estas limitaciones, en el Apartado 3.3.2, se propondrá una solución alternativa al Paso$-\mathbf{W}$ para solventar los problemas producidos por el uso de la aproximación de Procrustes y, una vez demostrada su validez, se usará como base para proponer el marco general MVA con restricciones, que será aplicado por el resto de propuestas de esta tesis doctoral.

### 3.3.1. Problemas de la aproximación de Procrustes

En este subapartado, se muestran los dos problemas clave que presenta el problema ortogonal de Procrustes cuando se aplica dentro del proceso iterativo usado para resolver métodos MVA regularizados. El objetivo de

este apartado, por lo tanto, es justificar que el uso que se está haciendo actualmente por defecto de esta solución es incorrecto y, así, poder demostrar posteriormente cuáles son las soluciones válidas.

Para ello, se pretende trabajar en base a una generalización de la propiedad definida por Zou et al. (2006), que declaraba que un buen método MVA regularizado debería reducirse al método MVA original si se anula el término de regularización. Por lo tanto, las siguientes demostraciones partirán del caso en que $\gamma = 0$ y se comprobará la convergencia a la solución MVA original. Pero antes de pasar a ello, a continuación se describirá en detalle la solución de Procrustes que se debería usar en este marco MVA iterativo.

El Paso−$\mathbf{W}$ (3.43) del algoritmo iterativo puede reescribirse haciendo uso de los multiplicadores de Lagrange ($\mathbf{\Xi}$) como el problema de maximización de la siguiente función de coste:

$$\mathcal{L}_{\mathbf{\Xi}}(\mathbf{W}) = 2\,\mathrm{Tr}\{\mathbf{W}^\top \mathbf{\Omega} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{U}\} - \mathrm{Tr}\{(\mathbf{W}^\top \mathbf{\Omega} \mathbf{W} - \mathbf{I})\mathbf{\Xi}\}, \tag{3.45}$$

que, derivando con respecto a $\mathbf{W}$ e igualando a cero, da lugar a:

$$\mathbf{\Omega} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{U} = \mathbf{\Omega} \mathbf{W} \mathbf{\Xi}. \tag{3.46}$$

Si se reescribe en función de $\mathbf{V} = \mathbf{\Omega}^{\frac{1}{2}} \mathbf{W}$ —como se definió en al Apartado 3.2—,

$$\mathbf{\Omega}^{\frac{1}{2}} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{U} = \mathbf{V} \mathbf{\Xi},$$

se podría describir el problema ortogonal de Procrustes en función de la siguiente descomposición de valores singulares:

$$\mathbf{\Omega}^{\frac{1}{2}} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{U} = \mathbf{Q} \mathbf{D} \mathbf{P}^\top, \tag{3.47}$$

o, reescribiéndolo en función de la matriz $\mathbf{V}$ calculada en la iteración anterior ($\mathbf{V}^{(i-1)}$) y sustituyendo $\mathbf{U}$ por la ecuación (3.30) —que es la solución obtenida cuando se anula la restricción—, como:

$$\begin{aligned} \mathbf{\Omega}^{\frac{1}{2}} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{C}_{\mathbf{XX}}^{-1} \mathbf{C}_{\mathbf{XY}} \mathbf{\Omega}^{\frac{1}{2}} \mathbf{V}^{(i-1)} &= \mathbf{Q} \mathbf{D} \mathbf{P}^\top \\ \mathbf{C} \mathbf{V}^{(i-1)} &= \mathbf{Q} \mathbf{D} \mathbf{P}^\top, \end{aligned} \tag{3.48}$$

siendo $\mathbf{C} = \mathbf{\Omega}^{\frac{1}{2}} \mathbf{C}_{\mathbf{XY}}^\top \mathbf{C}_{\mathbf{XX}}^{-1} \mathbf{C}_{\mathbf{XY}} \mathbf{\Omega}^{\frac{1}{2}}$. A partir de esta descomposición, la solución de la aproximación de Procrustes propuesta por Zou et al. (2006) puede ser definida como $\mathbf{V}_P = \mathbf{Q} \mathbf{P}^\top$, donde el subíndice $_P$ denota la solución de Procrustes.

Una vez la solución ha sido definida, ya se puede proceder a demostrar los dos problemas clave presentes en la aplicación de la aproximación de Procrustes, que son:

- Las variables de los datos de entrada proyectados ya no están incorreladas, impidiendo discriminar cuáles son las características más importantes. Poniendo como ejemplo al PCA, las componentes principales —o características extraídas—, dado este caso, contendrían una gran parte de la varianza descrita por cualquier otra componente principal, dejando de comportarse, por lo tanto, como un PCA. Nótese que este problema desmonta por sí solo la naturaleza de todo método MVA.

- La dependencia de la inicialización del proceso iterativo hasta el punto en que puede causar que el algoritmo no progrese en absoluto.

#### 3.3.1.1. Correlación de las variables proyectadas usando Procrustes

Para analizar la correlación de las variables proyectadas, se va a analizar la matriz de autocovarianza de los datos de entrada proyectados. Para ello, sustituyendo $\mathbf{U}$ por la solución de la ecuación (3.30), se puede reescribir dicha autocovarianza en términos de $\mathbf{V}$ como

$$\mathbf{U}^\top \mathbf{C_{XX}} \mathbf{U} = \mathbf{V}^\top \mathbf{C} \mathbf{V}.$$

Además, de la solución de Procrustes se sabe que $\mathbf{CV} = \mathbf{QDP}^\top$ y, puesto que $\mathbf{V}_P = \mathbf{QP}^\top$, se obtiene que la matriz de autocovarianza de los datos de entrada proyectados es:

$$\mathbf{U}^\top \mathbf{C_{XX}} \mathbf{U} = \mathbf{V}_P^\top \mathbf{C} \mathbf{V}_P = \mathbf{PDP}^\top,$$

que, en un caso general, no es diagonal y, por lo tanto, no se garantiza que los datos estén blanqueados.

Como aclaración, nótese que $\mathbf{P} = \mathbf{I}$ es la única solución posible para que haya incorrelación —en cuyo caso $\mathbf{V} = \mathbf{QP}^\top = \mathbf{Q}$—, puesto que $\mathbf{\Lambda} = \mathbf{D}$ es una matriz diagonal y $\mathbf{P}$ es la matriz de vectores singulares derechos —es decir, es una matriz ortogonal ($\mathbf{P}^\top = \mathbf{P}^{-1}$)—. Es decir, que no se podría dar el caso en el que una matriz ortogonal escalada por filas sea igual a ella misma escalada por columnas ($\mathbf{P\Lambda} \neq \mathbf{\Lambda P}$).

En otras palabras: el único caso válido de la aproximación de Procrustes sería inicializar el algoritmo con la solución óptima del método MVA original sin regularización, como hace Zou et al. (2006). De este modo, al hacer la descomposición (3.47), se obtendría directamente que $\mathbf{P} = \mathbf{I}$ y $\mathbf{D} = \mathbf{\Lambda}$. Sin embargo, a medida que el parámetro de regularización crece ($\gamma > 0$), esta solución no estaría forzando la incorrelación entre variables de entrada proyectadas —como se demostrará en el subapartado 3.3.2— y, como consecuencia, a medida que pasan las iteraciones, la solución alcanzada se aleja de manera incontrolada de la ortogonalidad de las características extraídas.

#### 3.3.1.2. Dependencia de la inicialización usando Procrustes

En este apartado, se va a analizar cómo la solución obtenida por la aproximación de Procrustes depende de la inicialización elegida. En particular, se considerará que el algoritmo se inicializa con una matriz ortogonal $\mathbf{V}^{(0)}$ (caso bastante habitual) y se analizará la solución a la que se llega en los pasos del proceso iterativo (desde $\mathbf{V}^{(0)}$ hasta $\mathbf{V}^{(1)}$, donde se indica con el superíndice $^{(i)}$ la $i$-ésima iteración):

1. Inicializar $\mathbf{V}^{(0)}$.
2. $\mathbf{U}^{(1)} = \mathbf{C}_{\mathbf{XX}}^{-1}\mathbf{C}_{\mathbf{XY}}\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{V}^{(0)}$.
3. $\boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}^{(1)} = \mathbf{QDP}^{\top}$.
4. $\mathbf{V}^{(1)} = \mathbf{QP}^{\top}$.

Con el fin de poder expresar $\mathbf{V}^{(1)}$ en función de $\mathbf{V}^{(0)}$, el paso 3 puede reescribirse en función de $\mathbf{V}^{(0)}$ —del mismo modo que se pasa de (3.47) a (3.48)— como:

$$\mathbf{CV}^{(0)} = \mathbf{QDP}^{\top}. \tag{3.49}$$

Para facilitar las derivaciones, a continuación se realizan unos pasos de álgebra lineal: multiplicando por la derecha a ambos lados de la ecuación (3.49) por sus transpuestas, se obtiene

$$\mathbf{QD}^2\mathbf{Q}^{\top} = \mathbf{CV}^{(0)}\mathbf{V}^{\top(0)}\mathbf{C},$$

y si se hace lo mismo, pero por la izquierda, se consigue

$$\mathbf{PD}^2\mathbf{P}^{\top} = \mathbf{V}^{\top(0)}\mathbf{CCV}^{(0)}.$$

Con esto se definen las siguiente igualdades que serán útiles para la presente demostración:

$$\mathbf{Q} = \mathbf{CV}^{(0)}\mathbf{V}^{\top(0)}\mathbf{CQD}^{-2}, \tag{3.50}$$

$$\mathbf{P} = \mathbf{V}^{\top(0)}\mathbf{CCV}^{(0)}\mathbf{PD}^{-2}. \tag{3.51}$$

Ahora, introduciendo (3.50) y (3.51) en la expresión del Paso 4 para $\mathbf{V}^{(1)}$ y suponiendo que $\mathbf{V}^{(0)}$ se inicializa como una matriz ortogonal (es decir, $\mathbf{V}^{\top(0)} = \mathbf{V}^{-1(0)}$), se obtiene

$$\begin{aligned}
\mathbf{V}^{(1)} &= \mathbf{QP}^{\top} \\
&= \mathbf{CV}^{(0)}\mathbf{V}^{\top(0)}\mathbf{C}(\mathbf{QD}^{-4}\mathbf{P}^{\top})\mathbf{V}^{\top(0)}\mathbf{CCV}^{(0)} \\
&= \mathbf{CV}^{(0)}\mathbf{V}^{\top(0)}\mathbf{C}(\mathbf{CV}^{(0)})^{-4}\mathbf{V}^{\top(0)}\mathbf{CCV}^{(0)} \\
&= \mathbf{CCC}^{-4}\mathbf{CCV}^{(0)} \\
&= \mathbf{V}^{(0)}.
\end{aligned}$$

Por lo tanto, se demuestra que la aproximación de Procrustes que se está usando en el proceso iterativo no progresa en absoluto cuando se anula el término de regularización y la matriz $\mathbf{V}$ es inicializada como una matriz ortogonal (es decir, $\mathbf{V}^{\top(0)} = \mathbf{V}^{-1(0)}$) con $n_f = m$. Este es el caso de van Gerven et al. (2012), que inicializa el algoritmo con los autovectores de $\mathbf{C_{YY}}$. Nótese también que, puesto que se impone $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$ (o $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$), la matriz ortogonal es una elección razonable para su inicialización, siendo la matriz identidad una elección clásica en estos casos.

### 3.3.2. Solución propuesta

En este subapartado, se presenta una solución alternativa al Paso$-\mathbf{W}$ que solventa los problemas ocasionados por la aproximación de Procrustes. La solución que se propone aquí se centrará en conseguir que se fuerce ortogonalidad de las características extraídas durante el procedimiento iterativo —obteniendo la misma solución que los métodos MVA cuando no entra en juego la restricción—.

Como punto de partida, es interesante volver a recordar por qué la formulación EVD (3.24) —la usada para generar la formulación iterativa— es válida para obtener soluciones MVA si no se impone explícitamente el blanqueamiento de los datos de entrada requerido. La respuesta a esto se analizó en el subapartado 3.2.1, donde se obtuvo que, imponiendo la restricción $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$, se cumplía la condición (3.33):

$$\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{W}^\top\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Lambda},$$

donde la primera igualdad indicaba que se conseguía la ortogonalidad de los datos proyectados y la segunda igualdad (3.37),

$$\mathbf{W}^\top\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Lambda},$$

es decir, la condición de incorrelación, forzaba la ortogonalidad entre los datos de entrada y salida proyectados por $\mathbf{U}$ y $\mathbf{W}$, respectivamente.

Sin embargo, dividir dicha formulación EVD en dos pasos acoplados dentro de un procedimiento iterativo ocasiona que el hecho de forzar $\mathbf{W}^\top\mathbf{\Omega}\mathbf{W} = \mathbf{I}$ en el Paso$-\mathbf{W}$ no sea suficiente para que en el Paso$-\mathbf{U}$ se obtenga una solución que consiga el blanqueamiento de los datos —como se ha demostrado si se aplica la solución por defecto que hace uso de la aproximación de Procrustes—; por consiguiente, la formulación iterativa usada en la literatura ya no sería válida para obtener soluciones MVA.

Para encontrar una solución que cumpla las igualdades de (3.33) y, por lo tanto, que haga de este procedimiento iterativo una formulación MVA válida, se van a analizar los dos problemas acoplados (3.42) y (3.43) descritos en los Pasos$-\mathbf{U}$ y $-\mathbf{W}$.

- Paso$-\mathbf{U}$: si se deriva (3.42) con respecto a $\mathbf{U}$ y se iguala a cero, se obtiene:

$$\mathbf{C_{XX}U} = \mathbf{C_{XY}\Omega W} - \gamma\frac{\partial R(\mathbf{U})}{\partial \mathbf{U}} \tag{3.52}$$

y, anulando el término de regularización $R(\mathbf{U})$ para el análisis, se obtiene: $\mathbf{C_{XX}U} = \mathbf{C_{XY}\Omega W}$ que, multiplicando por $\mathbf{U}^\top$ a ambos lados por la izquierda, permite verificar que se cumple la primera igualdad de (3.33) sin necesidad de ser forzada:

$$\mathbf{U}^\top\mathbf{C_{XX}U} = \mathbf{W}^\top\mathbf{\Omega C}_{\mathbf{XY}}^\top\mathbf{U}.$$

Esta igualdad indica que la relación existente entre los datos de entrada proyectados es la misma que la relación de los datos de entrada proyectados con los de salida proyectados.

- Paso$-\mathbf{W}$: para empezar, se parte de la igualdad de partida (3.46) obtenida tras derivar (3.43) con respecto a $\mathbf{W}$ e igualar a cero:

$$\mathbf{\Omega C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Omega W\Xi},$$

donde $\mathbf{\Xi}$ era la matriz de multiplicadores de Lagrange de la formulación equivalente (3.45). Ahora, si se multiplica por la izquierda a ambos lados de la ecuación por $\mathbf{W}^\top$, sabiendo que en este paso se fuerza $\mathbf{W}^\top\mathbf{\Omega W} = \mathbf{I}$, se obtiene

$$\mathbf{W}^\top\mathbf{\Omega C}_{\mathbf{XY}}^\top\mathbf{U} = \mathbf{\Xi}.$$

Por lo tanto, como en este Paso$-\mathbf{W}$ se fuerza que la condición de incorrelación sea igual a la matriz de multiplicadores de Lagrange, para que se dé la condición de blanqueado se hace completamente necesario que

$$\mathbf{\Xi} = \mathbf{\Lambda}.$$

Un modo de conseguir que la matriz de multiplicadores de Lagrange sea diagonal —pues $\mathbf{\Lambda}$ es diagonal— sería resolviendo la ecuación de partida del Paso$-\mathbf{W}$ (3.46) mediante un problema de autovalores. Nótese que, a partir de este punto, la solución del Paso$-\mathbf{W}$ de esta propuesta difiere de la aproximación de Procrustes usada por defecto –donde esta aproximación no fuerza que $\mathbf{\Xi}$ sea diagonal—.

Para conseguir que $\mathbf{\Xi}$ sea diagonal, se puede multiplicar por la derecha a ambos lados de la ecuación (3.46) por su transpuesta y luego por $\mathbf{W}$. Teniendo en cuenta la restricción $\mathbf{W}^\top\mathbf{\Omega W} = \mathbf{I}$, la solución óptima $\mathbf{W}$ vendría dada por el siguiente problema de autovalores generalizado:

$$\mathbf{\Omega C}_{\mathbf{XY}}^\top\mathbf{UU}^\top\mathbf{C_{XY}\Omega W} = \mathbf{\Omega W\Xi}^2,$$

que puede ser reescrito como el siguiente problema de autovalores estándar:

$$\mathbf{\Omega}^{\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{\Omega}^{\frac{1}{2}}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^{2}, \tag{3.53}$$

siendo $\mathbf{W} = \mathbf{\Omega}^{-\frac{1}{2}}\mathbf{V}$. De este modo, sí se verifica la segunda igualdad de (3.33): $\mathbf{W}^{\top}\mathbf{\Omega}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U} = \mathbf{\Lambda}$ y, por consiguiente, al resolver el Paso$-\mathbf{U}$, se obtiene $\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{\Lambda}$. Por lo tanto, la formulación iterativa resumida en la Tabla 3.3 sí sería válida para obtener soluciones MVA siempre y cuando se obtenga la solución del Paso$-\mathbf{W}$ (3.43) mediante (3.53).

Nótese que la solución propuesta en (3.53) también puede ser obtenida como los vectores singulares izquierdos $\mathbf{Q}$ de la descomposición SVD aplicada en la aproximación de Procrustes, puesto que, si se hace uso de (3.47), se obtiene:

$$\mathbf{\Omega}^{\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{\Omega}^{\frac{1}{2}} = \mathbf{Q}\mathbf{D}^{2}\mathbf{Q}^{\top}, \tag{3.54}$$

pudiéndose reescribir el problema de autovalores estándar de (3.53) en función de $\mathbf{Q}$ como:

$$\mathbf{\Omega}^{\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{\Omega}^{\frac{1}{2}}\mathbf{Q} = \mathbf{Q}\mathbf{D}^{2}.$$

Por lo tanto, la solución aquí propuesta puede ser también definida como $\mathbf{V} = \mathbf{Q}$ (o $\mathbf{W} = \mathbf{\Omega}^{-\frac{1}{2}}\mathbf{Q}$) con $\mathbf{\Lambda} = \mathbf{D}$.

Como se comentó en el Subapartado 3.3.1.1, cuando la solución que hace uso de la aproximación de Procrustes es inicializada con la solución original del método MVA en cuestión, la matriz de vectores singulares derechos era $\mathbf{P} = \mathbf{I}$, coincidiendo con la solución aquí propuesta cuando el término de regularización se anula. Sin embargo, a medida que el parámetro de regularización crece ($\gamma > 0$), la solución de Procrustes —al no forzar la condición de incorrelación (3.33) en el Paso$-\mathbf{W}$ del proceso iterativo— no estaría forzando en cada iteración la ortogonalidad de las características extraídas en el Paso$-\mathbf{U}$ y, a medida que pasan las iteraciones, la desviación con las igualdades de (3.37) aumentaría de manera descontrolada. Por el contrario, la solución aquí propuesta sí cumple la condición de incorrelación (3.33) en el Paso$-\mathbf{W}$ y, por lo tanto, en el Paso$-\mathbf{U}$ de cada iteración, se estaría forzando la incorrelación de las características extraídas mediante la relación (3.52) si $\gamma = 0$ o una aproximación a dicha ortogonalidad en función del término de regularización $R(\mathbf{U})$ y de su término de penalización $\gamma > 0$.

En la Tabla 3.4, se muestra un resumen del Paso$-\mathbf{U}$ y $-\mathbf{W}$ siguiendo el procedimiento iterativo propuesto para los métodos MVA más conocidos aplicando un término de regularización.

### 3.3.3. Experimentos

Aunque en el apartado anterior se ha demostrado teóricamente los problemas presentes en el empleo de la aproximación de Procrustes en la implementación iterativa de los métodos MVA, así como la validez y unicidad

Tabla 3.4: Resumen de los pasos necesarios del procedimiento iterativo propuesto para los métodos MVA más conocidos con un término de regularización incluido. Nótese que la salida proyectada para CCA es $\bar{\mathbf{Y}} = \mathbf{W}^\top \mathbf{C}_{\mathbf{YY}}^{-1}\mathbf{Y}$, para OPLS es $\bar{\mathbf{Y}} = \mathbf{W}^\top\mathbf{Y}$ y para PCA es $\bar{\mathbf{X}} = \mathbf{W}^\top\mathbf{X}$.

| | Cálculo de $\mathbf{U}$ (Paso$-\mathbf{U}$) | Cálculo de $\mathbf{V}$ (Paso$-\mathbf{W}$) | Cálculo de $\mathbf{W}$ |
|---|---|---|---|
| **Marco general** | $\arg\min_{\mathbf{U}} \|\bar{\mathbf{Y}} - \mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma R(\mathbf{U})$ | $\boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\mathbf{U}^\top\mathbf{C}_{\mathbf{XY}}\boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$ | $\mathbf{W} = \boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{V}$ |
| **CCA** ($\boldsymbol{\Omega} = \mathbf{C}_{\mathbf{YY}}^{-1}$) | $\arg\min_{\mathbf{U}} \|\bar{\mathbf{Y}} - \mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma R(\mathbf{U})$ | $\mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}}\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\mathbf{U}^\top\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$ | $\mathbf{W} = \mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}}\mathbf{V}$ |
| **OPLS** ($\boldsymbol{\Omega} = \mathbf{I}$) | $\arg\min_{\mathbf{U}} \|\bar{\mathbf{Y}} - \mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma R(\mathbf{U})$ | $\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\mathbf{U}^\top\mathbf{C}_{\mathbf{XY}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$ | $\mathbf{W} = \mathbf{V}$ |
| **PCA** ($\boldsymbol{\Omega} = \mathbf{I}$) | $\arg\min_{\mathbf{U}} \|\bar{\mathbf{X}} - \mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma R(\mathbf{U})$ | $\mathbf{C}_{\mathbf{XX}}^\top\mathbf{U}\mathbf{U}^\top\mathbf{C}_{\mathbf{XX}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$ | $\mathbf{W} = \mathbf{V}$ |

de nuestra solución, aquí se pretende mostrar empíricamente los efectos que se obtendrían al aplicarse en un problema real. Por lo tanto, las implementaciones que se comparan aquí son: la solución propuesta (referida como "Propuesta" en las figuras) y aquella que usa la aproximación de Procrustes (citada como "Procrustes" en las figuras). Como punto de referencia para dichas comparaciones, se van a usar las implementaciones propuestas en el marco general MVA descritas en el apartado 3.2 de los algoritmos originales ("Original" en las figuras).

En este caso, el problema usado (*segment*) se ha obtenido de Frank y Asuncion (2010) y se han usado 1617 muestras en cada subconjunto de datos de entrenamiento seleccionado —donde las 693 muestras restantes del conjunto se han empleado para evaluar los algoritmos— con 18 dimensiones o variables de entrada y 7 dimensiones de salida.

En estos experimentos, el objetivo es mostrar tres aspectos importantes que se han de cumplir en los métodos MVA (PCA en subfiguras (b) y (a), CCA en subfiguras (d) y (c) y OPLS en subfiguras (f) y (e) de las Figuras 3.2, 3.3, 3.4):

- Minimización (o maximización en el caso de CCA) de la función objetivo (véase Figura 3.2).— El objetivo de este experimento es mostrar, en función del número de características extraídas, si las soluciones comparadas obtienen las mismas prestaciones que la solución MVA original, pues todas las soluciones minimizan la misma función objetivo. Estas curvas han de converger al mismo valor cuando se usan todas las características, ya que en este caso el cuello de botella aplicado en estos algoritmos no influiría.

- Incorrelación de las variables de entrada proyectadas (véase Figura 3.3).— Los resultados que se muestran en esta figura muestran la diferencia entre la matriz de autocovarianza de los datos de entrada pro-

yectados entre el método original y los algoritmos comparados aquí. El resultado deseado es que dicha diferencia sea cero.

- Varianza acumulada explicada (véase Figura 3.4).— El objetivo de este experimento es ver si las soluciones obtenidas se pueden considerar soluciones MVA, debiendo coincidir con la solución del método MVA original. Estas curvas son interesantes, ya que cuando las variables proyectadas no están incorreladas, cada una de ellas podría contener varianza explicada de las otras. Para mostrar la varianza exclusivamente explicada por cada variable de entrada proyectada, se calcula la descomposición QR de la autocovarianza de los datos de entrada proyectados,

$$\mathbf{U}^\top \mathbf{C}_{\mathbf{XX}} \mathbf{U} = \mathbf{QR}.$$

  La varianza explicada en exclusiva por la $j$-ésima variable proyectada sería el valor absoluto del $j$-ésimo elemento de la diagonal de R, $|\mathbf{R}_{jj}|$ (para mayor detalle véase Zou et al., 2006). De este modo, la varianza explicada total acumulada por las $k$ variables proyectadas se calcularía como

$$\sum_{j=1}^{k} |\mathbf{R}_{jj}|.$$

En las correspondientes subfiguras (a), (c) y (e), los resultados mostrados se han obtenido como un promedio de 50 inicializaciones aleatorias distintas. En este caso, se ha mantenido la misma partición en los conjuntos de entrenamiento y test con el fin de poder mostrar la dependencia que presenta la solución de Procrustes a la inicialización del algoritmo. Como se puede ver en todas estas subfiguras, la solución de Procrustes presenta una gran desviación típica como consecuencia de una grave dependencia de la inicialización, mientras que en la solución propuesta esta desviación típica es nula.

Con respecto a las subfiguras (b), (d) y (f), se muestran las respectivas curvas obtenidas como el promedio de 50 realizaciones —o ejecuciones— independientes, seleccionando aleatoriamente un conjunto de entrenamiento y test distinto cada vez. Para todos los algoritmos y todas las ejecuciones, se ha usado la misma inicialización seleccionada de manera aleatoria. Con esto se pretende mostrar que la solución de Procrustes no es robusta ante distintas realizaciones del mismo problema; como se puede ver en las Figuras 3.3d, 3.3f, 3.4d y 3.4f, donde existe una cierta desviación típica en su solución (esto también es visible en la medida de incorrelación o en la medida de la varianza explicada acumulada).

Por último —pero no por eso menos importante—, hay que destacar que los algoritmos propuestos convergen a la misma solución que la de los métodos MVA, mientras que la aproximación de Procrustes —como se puede ver en la Figura 3.2— no minimiza (o maximiza) la función objetivo para

$n_f < k$, siendo $k$ el número total de características posibles. Esto se debe a que, con el uso de Procrustes, no se consigue incorrelación entre las variables proyectadas —como puede verse en la Figura 3.3—, causando asimismo que la varianza explicada en cada proyección sea mucho menor que en las soluciones propuestas —como queda reflejado en la Figura 3.4—.

## 3.4. Conclusiones

En este capítulo, con el fin de crear un marco general MVA que permita incluir restricciones sobre los vectores de proyección, se ha demostrado en primera instancia la equivalencia de dos soluciones distintas al mismo problema y la eficiencia de cada uno de ellos en función del tamaño del problema. Con este resultado, se ha propuesto un marco general MVA eficiente para los métodos MVA asegurando la ortogonalidad de las características extraídas.

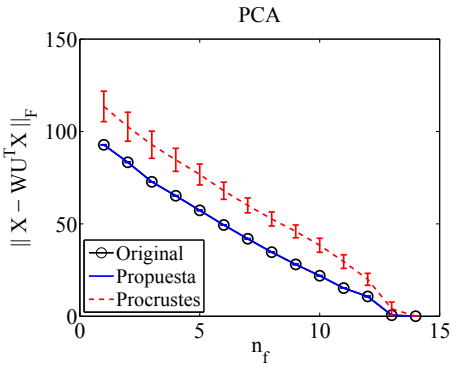
Finalmente y usando como base este marco general MVA, se ha propuesto un algoritmo iterativo que permite resolver métodos MVA con términos de regularización adicionales. Además, se ha demostrado teóricamente que la solución existente actualmente y usada por defecto presenta dos graves problemas: no fuerza incorrelación de las variables de entrada proyectadas y es dependiente de la inicialización del algoritmo. En esta demostración, también se ha concluido que la solución propuesta es única para obtener soluciones MVA, pues fuerza la incorrelación de las características extraídas, que es una propiedad deseada en los métodos MVA. Además, aunque la solución de Procrustes converge al método propuesto cuando se anula el término de regularización si y solo si se inicializa con la solución original MVA —pues convergería en el primer paso a la misma—, no fuerza la ortogonalidad deseada cuando entra en juego dicha regularización, perdiendo la capacidad de devolver características dispuestas en orden de relevancia. Con la solución aquí propuesta, esto sí se sigue cumpliendo.

Estos resultados también han sido demostrados empíricamente para tres de los algoritmos MVA más populares (PCA, CCA y OPLS) cuando se anula el término de regularización. El objetivo del resto de esta tesis doctoral será demostrar la validez de esta propuesta también cuando se introducen distintos términos de regularización —presentando uno distinto por capítulo—, así como la utilidad de añadir cada uno de ellos.

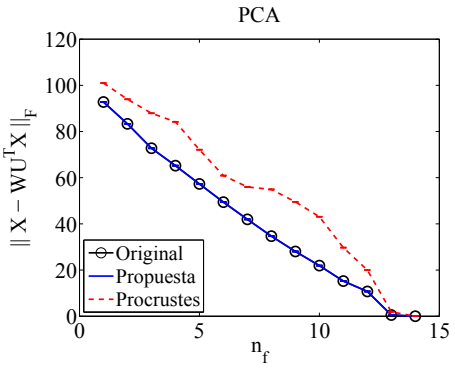
## En los próximos capítulos...

Hasta el momento se ha demostrado la validez del marco general aquí propuesto para distintos métodos MVA: PCA, CCA y OPLS. En los próximos capítulos, se estudiarán distintas particularizaciones del término de regularización y, con el fin de evitar redundancia, únicamente se hará para OPLS, ya que es el método que obtiene la solución óptima en el sentido de

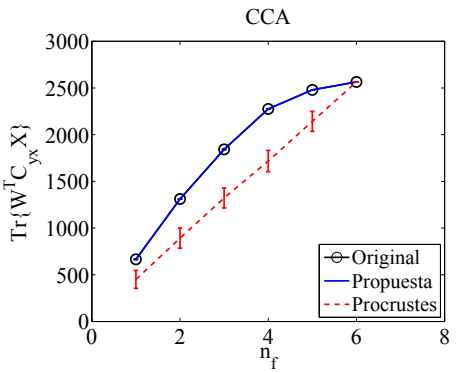
mínimo error cuadrático medio y el PCA es un caso particular del OPLS para el caso no supervisado.



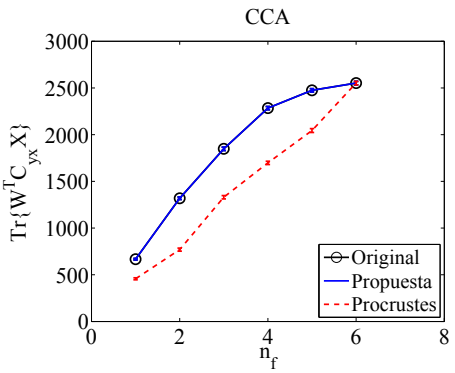

(a) Promedio de 50 inicializaciones distintas



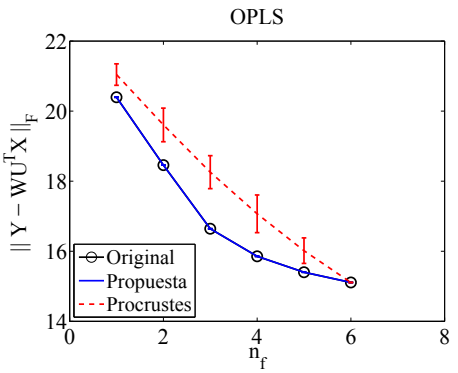

(b) Promedio de 50 ejecuciones aleatorias



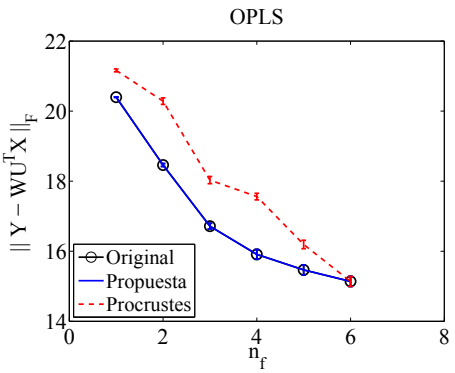

(c) Promedio de 50 inicializaciones distintas




(d) Promedio de 50 ejecuciones aleatorias

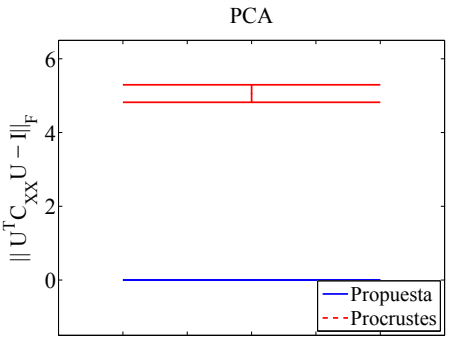



(e) Promedio de 50 inicializaciones distintas

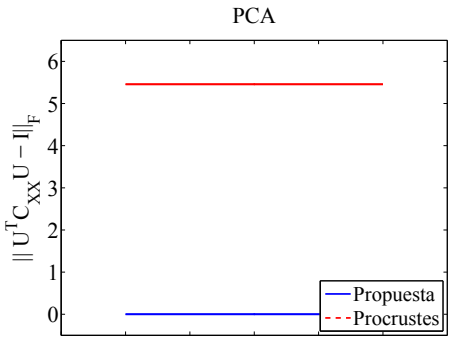



(f) Promedio de 50 ejecuciones aleatorias

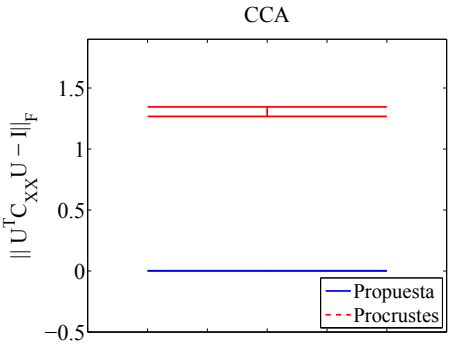
Figura 3.2: Comparativa en la consecución de la función objetivo para los métodos PCA, CCA y OPLS y sus versiones iterativas



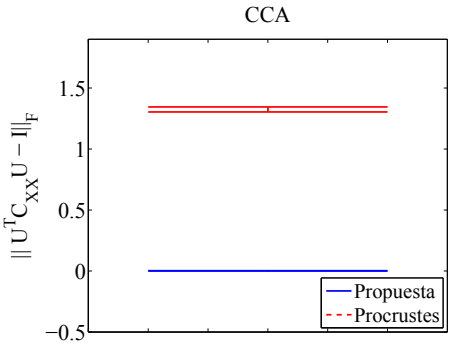

(a) Promedio de 50 inicializaciones distintas



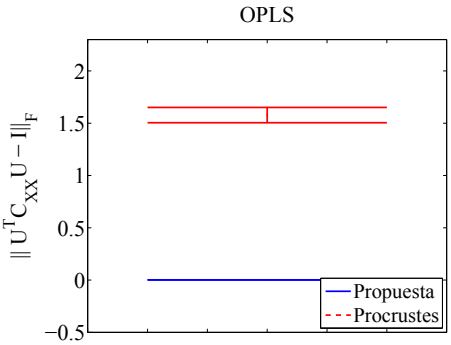

(b) Promedio de 50 ejecuciones aleatorias



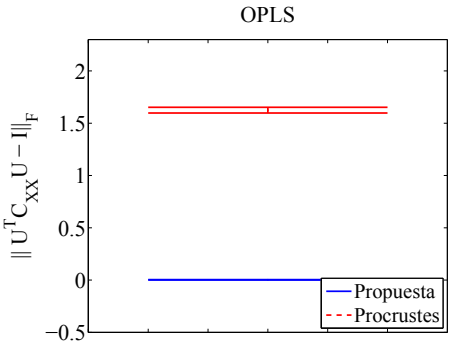

(c) Promedio de 50 inicializaciones distintas




(d) Promedio de 50 ejecuciones aleatorias

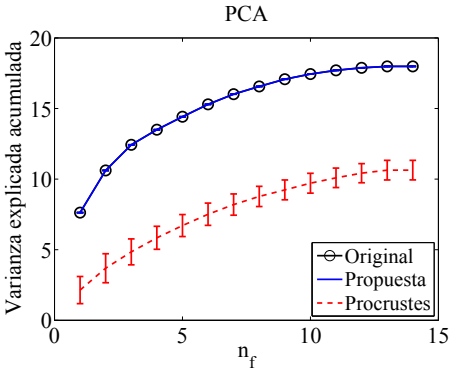



(e) Promedio de 50 inicializaciones distintas

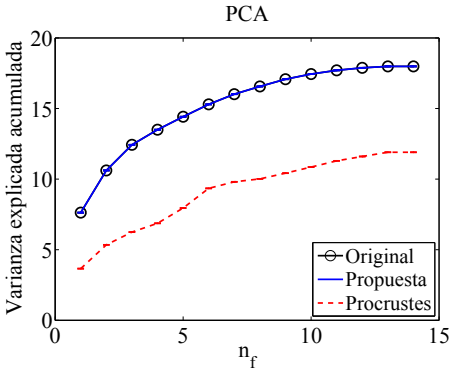



(f) Promedio de 50 ejecuciones aleatorias

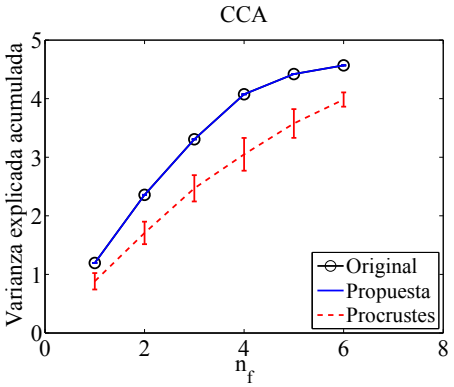
Figura 3.3: Comparativa en la consecución del blanquemiento de los datos de entrada para las versiones iterativas de los métodos PCA, CCA y OPLS



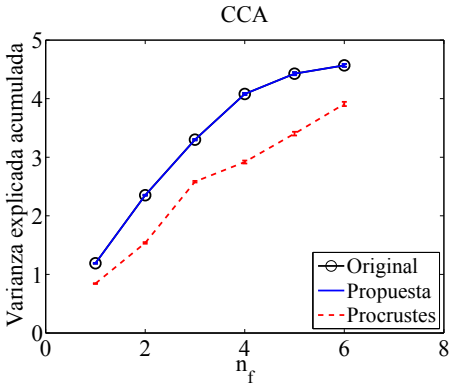

(a) Promedio de 50 inicializaciones distintas



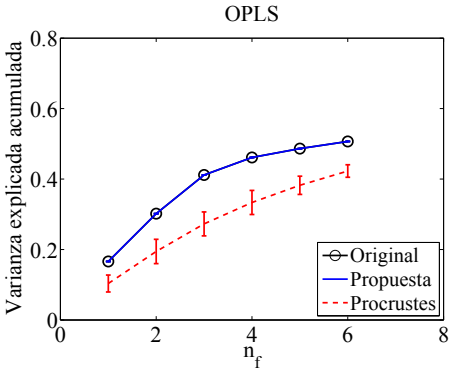

(b) Promedio de 50 ejecuciones aleatorias



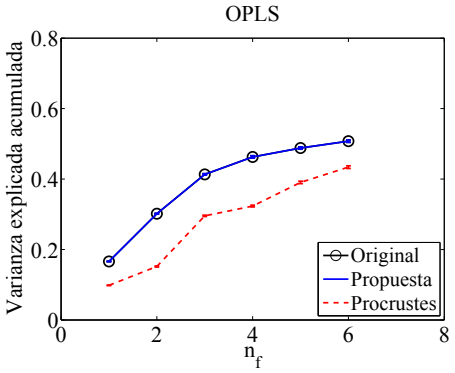

(c) Promedio de 50 inicializaciones distintas




(d) Promedio de 50 ejecuciones aleatorias




(e) Promedio de 50 inicializaciones distintas




(f) Promedio de 50 ejecuciones aleatorias

Figura 3.4: Comparativa de la varianza explicada acumulada obtenida por las versiones iterativas de los métodos PCA, CCA y OPLS

# Capítulo 4

# MVA con restricciones de dispersión

*Todo debe simplificarse lo máximo posible, pero no más.*

Albert Einstein (1879-1955)

**Resumen:** En el capítulo anterior, se propuso una formulación general para incluir restricciones en los métodos MVA que facilitaría, por ejemplo, la obtención de extensiones dispersas de estos algoritmos basadas en la norma $\ell_1$. En este capítulo, se explota esta propiedad para obtener una versión dispersa del algoritmo OPLS y se analiza el poder de discriminación de este nuevo método sobre problemas de clasificación. Además, se compara el grado de dispersión obtenido por esta solución con los métodos del estado del arte para extracción de características dispersas.

## 4.1. OPLS disperso

Es este capítulo, se propone una nueva solución OPLS que impone dispersión sobre los vectores de proyección. De esta manera, el método no solo llevará a cabo una extracción de características, sino también una selección de las variables más relevantes para generar cada vector de proyección. Esto permite soluciones más interpretables que involucran solamente a unas pocas variables originales, siendo una propiedad deseable de los algoritmos de aprendizaje automático en muchos contextos. Para obtener esta solución OPLS dispersa, se hará uso de la formulación EVD (es decir, se usará la restricción $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ durante todas las derivaciones); en otras palabras, se usará el marco general MVA con restricciones propuesto en el Capítulo 3.

Es bien conocido que añadir un término de regularización $\ell_1$ (bautizado como *lasso*) produce soluciones dispersas, ya que facilita o, incluso, fuerza que los coeficientes asociados a las variables irrelevantes de la solución se anulen. Esta aproximación se basará en la implementación del método *lasso* ("least absolute shrinkage and selection operator"), que resuelve el problema de mínimos cuadrados sujeto a la regularización $\ell_1$. De este modo, se modifica el problema OPLS (3.1) como la minimización de

$$\mathcal{L}_{\text{reg}}(\mathbf{W}, \mathbf{U}) = \|\mathbf{Y} - \mathbf{W}\mathbf{U}^\top \mathbf{X}\|_F^2 + \gamma_1 \|\mathbf{U}\|_1 \tag{4.1}$$

sujeto a $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$. Aquí, $\gamma_1$ es el parámetro que controla la cantidad de regularización y $\|\mathbf{U}\|_1$ es la norma $\ell_1$ de la matriz $\mathbf{U}$, es decir, la suma de los valores absolutos de todas las componentes de la matriz.

### 4.1.1. Algoritmo de resolución en modo bloque

Para resolver (4.1) se hará uso del algoritmo propuesto en el Capítulo 3 basado en la aplicación iterativa de los siguientes dos pasos:

1) Paso$-\mathbf{W}$: Fijando $\mathbf{U}$, minimizar (4.1) sujeto a $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$.

   Cuando (4.1) es minimizado solamente con respecto a $\mathbf{W}$, ambos términos de regularización pueden ser ignorados. De esta manera, este paso se reduce a la minimización de la función de coste LS sujeto a la restricción $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, llegando a ser, de este modo, similar a EVD-OPLS, pero con la diferencia que $\mathbf{W}$ es optimizado para un $\mathbf{U}$ genérico, es decir, sin asumir (3.9). Como se ha demostrado en el Capítulo 3, la solución de este problema está dado por el problema de autovalores estándar:

   $$\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}}^\top \mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} \mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \tag{4.2}$$

   donde $\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} = \mathbf{U}^\top \mathbf{C}_{\mathbf{X}\mathbf{Y}}$. Nótese que la dimensión de la matriz que necesita ser analizada es $m$, al igual que ocurría con el problema EVD-OPLS estándar.

2) Paso$-\mathbf{U}$: Fijando $\mathbf{W}$, minimizar (4.1) con respecto a $\mathbf{U}$ solamente.

   Existen varios métodos eficientes para resolver este problema *lasso*. Léase, por ejemplo, Bach et al. (2011) y Yuan et al. (2010) como buenos resúmenes de métodos de optimización con regularización $\ell_1$. En el apartado de experimentos, se hará uso de la implementación facilitada por MOSEK 6.0[1], aunque cualquier otra implementación *lasso* podría también ser considerada aquí.

Se ha observado mediante experimentos preliminares que la inicialización del algoritmo no es crítica, pudiendo inicializar $\mathbf{U}$ en la primera iteración

[1] `http://www.mosek.com`.

como la matriz identidad. Como mecanismo de parada del algoritmo, se va a usar $\mathrm{Tr}\{\mathbf{\Lambda}^{(i)} - \mathbf{\Lambda}^{(i-1)}\} \leq \delta$, donde el superíndice denota el índice de la iteración y $\delta$ es una constante muy pequeña. En pocas palabras: el algoritmo se detiene cuando la diferencia entre los autovalores del Paso$-\mathbf{W}$ entre dos iteraciones consecutivas es menor que una constante arbitraria.

También merece la pena mencionar que el Paso$-\mathbf{U}$ puede ser modificado para imponer restricciones de dispersión sobre filas enteras de $\mathbf{U}$ —en lugar de hacerlo en cada componente aislada— de manera similar a lo realizado con el algoritmo conocido como *group-lasso* (Friedman et al., 2010). Sin embargo, esta última aproximación implica un incremento de memoria y de coste computacional requerido. Aún así, ofrece la ventaja adicional de que todos los vectores de proyección están limitados a usar las mismas variables de los datos de entrada, favoreciendo así una selección de características real, pues fuerza a que la misma característica original sea, o bien eliminada, o bien conservada, para todas las proyecciones.

Como ya se ha demostrado en el Capítulo 3, existe una diferencia muy importante entre la aproximación propuesta y el algoritmo introducido por van Gerven y Heskes (2010). Partiendo de la descomposición en autovalores singulares, $\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} = \mathbf{P}\mathbf{D}\mathbf{Q}^\top$ —donde $\mathbf{D}$ es una matriz diagonal que contiene los valores singulares y $\mathbf{P}$ y $\mathbf{Q}$ contienen los vectores singulares izquierdos y derechos respectivamente—, el resultado del Paso$-\mathbf{W}$ del algoritmo propuesto sería $\mathbf{W} = \mathbf{Q}$, mientras que el problema ortogonal de Procrustes usado en van Gerven y Heskes (2010) produciría una versión rotada de $\mathbf{W} = \mathbf{Q}\mathbf{P}^\top$. Esta rotación, además, implica que sin los términos de regularización (es decir, $\gamma_1 = \gamma_2 = 0$), el algoritmo de van Gerven y Heskes (2010) no converge en general a la solución OPLS, sino a una versión rotada de la matriz de proyección OPLS. Como ya se ha discutido, esto no es una cuestión irrelevante, ya que la solución OPLS real garantiza que las proyecciones extraídas están ordenadas de acuerdo a su relevancia —es decir, las primeras $n'_f < n_f$ características contienen tanta información como es posible para ese número de variables en el sentido de minimizar (4.1)—, además de ser ortogonales entre sí. Como se ha demostrado en el subapartado 3.3.1.1, esta propiedad no se cumple para soluciones rotadas. Por otro lado, como se demuestra en el subapartado 3.3.1.2, a diferencia de la solución propuesta, el algoritmo de van Gerven y Heskes (2010) depende de la inicialización.

### 4.1.2. Implementación secuencial usando deflacción

De igual manera que la implementación secuencial de EVD-OPLS, se puede derivar el algoritmo secuencial que implementa el esquema de extracción de características OPLS disperso que se acaba de describir. El algoritmo secuencial extrae primeramente el par de vectores $\{\boldsymbol{u}_k, \boldsymbol{w}_k\}$ que minimiza (4.1) para $n_f = 1$ y, seguidamente, deflacta la matriz de covarianza cruzada $\mathbf{C}_{\mathbf{X}\mathbf{Y}}$. Estos dos pasos se repiten hasta que se alcanza el número deseado de caracte-

rísticas. La extracción de los pares de vectores $\{\boldsymbol{u}_k, \boldsymbol{w}_k\}$, para $k = 1, \ldots, n_f$, se lleva a cabo iterando los Pasos$-\mathbf{U}$ y $-\mathbf{W}$ descritos anteriormente. Nótese que, puesto que en cada paso se está resolviendo un problema unidimensional, la solución del Paso$-\mathbf{W}$ se puede obtener simplemente como:

$$\boldsymbol{w}_k = \frac{\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}^{\top}}{\|\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}\|}, \tag{4.3}$$

donde $\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}} = \boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}$.

Es importante conceder el espacio necesario para aclarar la técnica de deflacción usada, ya que aunque los vectores $\boldsymbol{w}_k$ (para $k = 1, \ldots, n_f$) son autovectores reales, los vectores $\boldsymbol{u}_k$ son soluciones dispersas obtenidas mediante un término de regularización, conocidas, conforme a Mackey (2009), como pseudo-autovectores. La influencia de estas soluciones, al no satisfacer las propiedades necesarias para la mayoría de los métodos de deflacción, no se eliminaría por completo cuando dichos métodos son usados, pudiendo aparecer componentes paralelas a estos pseudo-autovectores en las subsiguientes iteraciones del proceso (véase el subapartado 2.1.4 de esta tesis doctoral o el artículo de Mackey, 2009, para mayor detalle). En este caso, debido a que las soluciones $\boldsymbol{w}_k$ son autovectores reales —o vectores singulares derechos de $\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}$—, se podría usar la deflacción por proyección usada en la ecuación (3.18), ya que se despejaría $\boldsymbol{u}_k^{\top}$,

$$\mathbf{C}_{\mathbf{XY}} \leftarrow \mathbf{C}_{\mathbf{XY}}(\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^{\top}),$$

que sería equivalente a deflactar las columnas de $\mathbf{Y}$:

$$\mathbf{Y} \leftarrow (\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^{\top})\mathbf{Y}. \tag{4.4}$$

Ahora, si se sustituye $\boldsymbol{w}_k$ por la solución (4.3), esta deflacción se puede reescribir únicamente en función de $\boldsymbol{u}_k$ como:

$$\mathbf{C}_{\mathbf{XY}} \leftarrow \mathbf{C}_{\mathbf{XY}}\left(\mathbf{I} - \frac{\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}}{\|\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k\|^2}\right), \tag{4.5}$$

donde se ha traspuesto todo con el fin de mostrar la equivalencia con (2.7), es decir, con la proyección sobre el complemento ortogonal del espacio transformado por el pseudo-autovector obtenido —como propone la deflacción por complemento de Schur—. Premultiplicando su transpuesta por el lado izquierdo, se puede comprobar que coincide con la deflacción por complemento de Schur descrita en (2.8):

$$\begin{aligned}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top} &\leftarrow \mathbf{C}_{\mathbf{XY}}\left(\mathbf{I} - \frac{\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}}{\|\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k\|^2}\right)\mathbf{C}_{\mathbf{XY}}^{\top} \\ &= \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top} - \frac{\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}}{\boldsymbol{u}_k^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_k}.\end{aligned}$$

Nótese que esta técnica de deflacción con respecto a $\boldsymbol{u}_k$ correspondería a deflactar la matriz cuadrada $\mathbf{C_{XY}}\mathbf{C_{XY}}^\top$ sujeto a la condición de ortogonalidad de los datos proyectados y, por tanto, correspondería a una solución deflactada de una versión escalada del problema GEV-OPLS (3.8).

Para demostrar que se elimina la influencia de las soluciones obtenidas para los siguientes pasos, se puede confirmar, del mismo modo que se hizo en el subapartado 2.1.4, que se cumple tanto para $\boldsymbol{w}_k$,

$$\mathbf{C_{XY}}\boldsymbol{w}_k \leftarrow \mathbf{C_{XY}}(\mathbf{I} - \boldsymbol{w}_k\boldsymbol{w}_k^\top)\boldsymbol{w}_k = \mathbf{C_{XY}}(\boldsymbol{w}_k - \boldsymbol{w}_k) = 0,$$

(ya que $\boldsymbol{w}_j^\top\boldsymbol{w}_k = 1$ solamente para $j = k$ y 0 en caso contrario) como para el pseudo-autovector $\boldsymbol{u}_k$,

$$\begin{aligned}\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k \leftarrow\ & \mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k - \frac{\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k\boldsymbol{u}_k^\top\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k}{\boldsymbol{u}_k^\top\mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k} \\ =\ & \mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k - \mathbf{C_{XY}}\mathbf{C_{XY}}^\top\boldsymbol{u}_k = 0,\end{aligned}$$

sin la necesidad de satisfacer las propiedades de autovector.

No obstante, si se quiere aportar una interpretación del método de deflacción (4.5) para el OPLS y sus versiones con restricciones, resulta interesante reescribirla como una proyección ortogonal de $\mathbf{Y}$ sobre el complemento ortogonal del espacio definido por $\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}^\top$ (es decir, $\mathcal{P}^\perp_{\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}^\top}(\mathbf{Y})$),

$$\mathbf{Y} \leftarrow \left(\mathbf{I} - \frac{\mathbf{Y}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^\top\mathbf{Y}^\top}{\|\mathbf{Y}\bar{\boldsymbol{x}}_k\|^2}\right)\mathbf{Y},$$

o deflactando la matriz de autocovarianzas de $\mathbf{Y}$:

$$\mathbf{C_{YY}} \leftarrow\ \mathbf{C_{YY}} - \frac{\mathbf{C_{YY}}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^\top\mathbf{C_{YY}}}{\bar{\boldsymbol{x}}_k^\top\mathbf{C_{YY}}\bar{\boldsymbol{x}}_k}.$$

De este modo, se puede ver que en cada iteración del procedimiento de deflacción se elimina la influencia de cada característica extraída de la matriz de salida $\mathbf{Y}$. Si se compara con las ecuaciones (2.7) y (2.8), se puede ver que las características extraídas $\bar{\boldsymbol{x}}$ son los autovectores —o pseudo-autovectores— de la deflacción por complemento de Schur usada.

La Tabla 4.1 incluye el pseudocódigo para el algoritmo secuencial disperso que se acaba de describir. Nótese que, en esta tabla, el subíndice $k$ se usa para indicar el $k$-ésimo vector de proyección (es decir, $k = 1, \ldots, n_f$), mientras que el superíndice $i$ indica el número de ejecuciones de los Pasos$-\mathbf{U}$ y $-\mathbf{W}$ que son necesarias para converger por cada vector de proyección. Se pueden usar diferentes criterios de convergencia para el paso 2.2.3 del algoritmo. En el apartado de experimentos, se usará la distancia coseno,

$$d_{\cos}\left(\boldsymbol{u}_k^{(i)}, \boldsymbol{u}_k^{(i-1)}\right) = \frac{\boldsymbol{u}_k^{(i)\top}\boldsymbol{u}_k^{(i-1)}}{\|\boldsymbol{u}_k^{(i)}\|\|\boldsymbol{u}_k^{(i-1)}\|}, \tag{4.6}$$

y se utilizará como criterio de parada $d_{\cos}\left(\boldsymbol{u}_k^{(i)}, \boldsymbol{u}_k^{(i-1)}\right) > 1-\delta$, donde $\delta$ es un parámetro de tolerancia. Otras opciones consistirían en controlar la distancia coseno entre los vectores de coeficientes de regresión o los autovalores del Paso$-\mathbf{W}$.

Tabla 4.1: Pseudocódigo del algoritmo secuencial con deflacción

1.- Entradas: matrices centradas $\mathbf{X}$ e $\mathbf{Y}$, $n_f$, $\gamma_1$, $\gamma_2$.
2.- Para $k = 1, \ldots, n_f$
  2.1.- Inicializar $\boldsymbol{u}_k^{(0)} = \mathbf{1} * \delta_k$ ‡.
  2.2.- Para $i = 1, 2, \ldots$
    2.2.1.- Actualizar $\boldsymbol{w}_k^{(i)}$ usando (4.3).
    2.2.2.- Actualizar $\boldsymbol{u}_k^{(i)}$ resolviendo el problema *lasso* (4.1) para $n_f = 1$.
    2.2.3.- Si se cumple el criterio de convergencia, los valores actuales de salida serían $\{\boldsymbol{u}_k, \boldsymbol{w}_k\}$, en caso contrario volver a 2.2.
  2.3.- Deflactar la matriz de covarianza cruzada: $\mathbf{C_{XY}} \leftarrow \mathbf{C_{XY}}\left(\mathbf{I} - \frac{\mathbf{C_{XY}^\top} \boldsymbol{u}_k \boldsymbol{u}_k^\top \mathbf{C_{XY}}}{\|\mathbf{C_{XY}^\top} \boldsymbol{u}_k\|^2}\right)$.
3.- Salidas: $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_f}]$, $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_f}]$.

‡ El vector de proyección $\boldsymbol{u}_k$ se inicializa como un vector con su $k$-ésima componente igual a 1 y todas las demás componentes igual a 0.

## 4.2. Experimentos

En este apartado, se analizará el poder discriminatorio de la solución OPLS dispersa ("Sparse OPLS", SOPLS). Con este propósito, se van a evaluar las prestaciones de esta aproximación sobre nueve problemas multi-clase tomados de Frank y Asuncion (2010). La Tabla 4.2 resume sus principales características, siendo $N_{train}$ y $N_{test}$ el número de muestras en los conjuntos de entrenamiento y test respectivamente. Para completar este estudio, también se analizará la convergencia de la solución SOPLS propuesta con respecto a aquella del OPLS cuando la restricción de dispersión es eliminada. Por último, se mostrarán también las ventajas de las soluciones dispersas en una tarea de reconocimiento de caras.

### 4.2.1. Extracción lineal de características dispersas

Este subapartado analiza las capacidades de la aproximación SOPLS propuesta contra el método OPLS estándar y el algoritmo OPLS disperso propuesto por van Gerven et al. (2012) que hace uso de la solución del problema de Procrustes; por esta razón, esta última solución se denotará como P-SOPLS ("Procrustes Sparse OPLS").

Para calcular las soluciones de las diferentes aproximaciones bajo estudio, el método OPLS sigue los pasos descritos en las ecuaciones (3.13) y (3.14), P-SOPLS sigue el procedimiento descrito por van Gerven y Heskes

Tabla 4.2: Principales propiedades de los problemas de referencia seleccionados

| | $N_{train}/N_{test}$ | $n$ | $m$ (núm. de clases) |
|---|---|---|---|
| *arrhythmia* | 315 / 135 | 276 | 16 |
| *letter* | 10000 / 10000 | 16 | 26 |
| *mfeatures* | 1400 / 600 | 649 | 10 |
| *optdigits* | 3823 / 1797 | 64 | 10 |
| *pendigits* | 7494 / 3498 | 16 | 10 |
| *satellite* | 4435 / 2000 | 36 | 6 |
| *segment* | 1310 / 1000 | 18 | 7 |
| *vehicle* | 500 / 346 | 18 | 4 |
| *yeast* | 1038 / 446 | 8 | 10 |

(2010), y la aproximación SOPLS propuesta usa la formulación detallada en la Tabla 4.1, parando su proceso iterativo, bien cuando la distancia coseno (4.6) alcanza el nivel de tolerancia $\delta = 10^{-12}$, bien cuando se completa un número máximo de 500 iteraciones. El parámetro de regularización $\gamma_1$ de las aproximaciones SOPLS y P-SOPLS se ha ajustado mediante un proceso de *validación cruzada* ("Cross-Validation", CV) seleccionando dicho valor de un conjunto de 40 valores logarítmicamente equiespaciados entre $10^{-4}$ y $10^{-1}$.

Para probar la capacidad de discriminación del conjunto de características proporcionadas para cada método, se ha entrenado una Máquina de Vectores Soporte lineal para clasificación ("Support Vector Machine", C-SVM) usando como entradas el número máximo de proyecciones ($r = \text{rango}(\mathbf{C_{XY}})$) y seleccionando el parámetro de coste $C$ entre un conjunto de valores $\{1, 10, 100, 1000\}$ con una CV de 10 particiones ("10-fold CV"). Es importante señalar que el problema *segment* está mal condicionado (es decir, $\text{rango}(\mathbf{C_{XX}}) < n$) imposibilitando la aplicación de OPLS; por esta razón, se ha aplicado el PCA como un paso de preprocesamiento para reducir la dimensión de los datos de entrada a $\text{rango}(\mathbf{C_{XX}})$, pudiéndose así aplicar el OPLS[2]. Esto no fue necesario para las aproximaciones dispersas (P-SOPLS y SOPLS), ya que la regularización $\ell_1$ incluida hace posible la resolución de problemas mal condicionados sin ningún tipo de preprocesamiento.

La Tabla 4.3 muestra la precisión total ("Overall Accuracy", OA) proporcionada por estas tres técnicas de selección de características, usando todos los vectores de proyección, y la tasa de dispersión ("Sparsity rates" SR) de los vectores de proyección, definida como el cociente entre el número de coeficientes iguales a cero y el número total de coeficientes.

[2]Una mejor opción, de acuerdo a Arenas-García et al. (2013), sería incluir un término de regularización $\ell_2$ en la función de coste para estimar la varianza de ruido de los datos de entrada y así compensar su efecto, pero requeriría validar dicho parámetro.

Tabla 4.3: Precisión total ("Overall Accuracy", OA) alcanzada por los algoritmos OPLS, P-SOPLS y SOPLS. También se incluyen las tasas de dispersión ("Sparsity rates" SR) de P-SOPLS y SOPLS.

| | OPLS | P-SOPLS | | SOPLS | |
|---|---|---|---|---|---|
| | OA(%) | OA(%) | SR(%) | OA(%) | SR(%) |
| *arrhythmia* | 50,37 | **69,63** | 77.63 | **69,63** | 76.06 |
| *letter* | 84,89 | 84,85 | 11,33 | **85,05** | 10,94 |
| *mfeatures* | 97,83 | **98,33** | 38.64 | **98,33** | 31.55 |
| *optdigits* | 94,21 | 94,27 | 42,47 | **95,05** | 29,93 |
| *pendigits* | 92,08 | 91,68 | 39,58 | **92,22** | 43,06 |
| *satellite* | 85,7 | 85,90 | 17,22 | **86,10** | 27,22 |
| *segment* | 92,8 | **95,60** | 90,74 | 94,90 | 93,52 |
| *vehicle* | 78,32 | 77,17 | 25,93 | **78,03** | 1,85 |
| *yeast* | 58,52 | **58,74** | 35,94 | 58,27 | 23,44 |

Cuando las características SOPLS son usadas para entrenar la C-SVM, se supera a OPLS en todas las bases de datos, mientras que mejora o empata con el método P-SOPLS en términos de OA.

Aparte de su mayor capacidad de discriminación, la principal ventaja del método SOPLS propuesto recae en su formulación dispersa que hace que sea más fácil analizar qué variables no contribuyen para obtener las nuevas proyectadas. Para llevar a cabo este análisis, la Figura 4.1 representa las matrices de proyección $\mathbf{U}$ obtenidas por los métodos OPLS, SOPLS y P-SOPLS en tres problemas representativos. Mirando estas figuras, se puede ver que en los problemas que presentan una alta SR, como *segment*, la extracción de características se convierte prácticamente en selección de variables, puesto que la mayoría de estas características están asociados solamente con una de las variables originales. En *satellite*, las características 8, 31, 32 y 36 son eliminadas de los primeros vectores de proyección (los más importantes) del algoritmo SOPLS.

### 4.2.2. Convergencia a la solución OPLS de los métodos SOPLS con $\gamma_1 = 0$

En este subapartado, se compara la convergencia de las soluciones SOPLS y P-SOPLS al OPLS estándar si la restricción de dispersión tiende a cero ($\gamma_1 \to 0$). Para llevar a cabo este análisis, se va a analizar la ortogonalidad de los datos proyectados para las implementaciones en bloque de los algoritmos SOPLS y P-SOPLS.

La Figura 4.2 refleja la distancia de Frobenius entre la matriz de covarianza de los datos proyectados cuando se usan los algoritmos SOPLS o

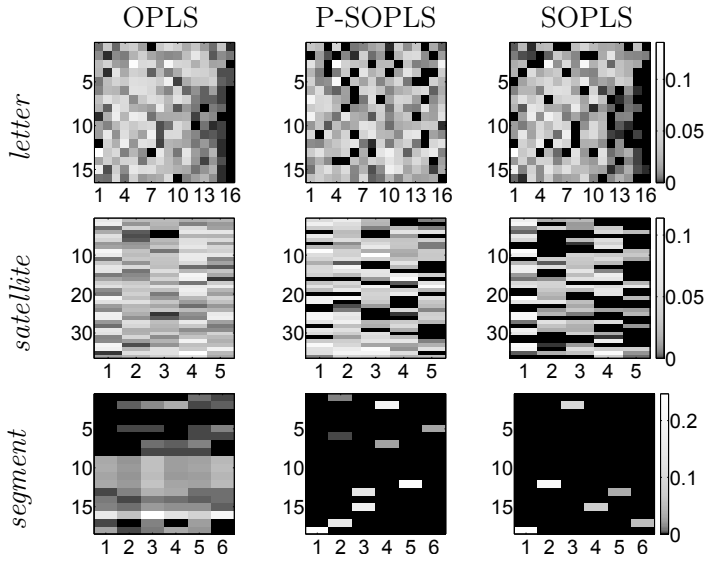



Figura 4.1: Representación de la matriz de proyección $U$ $(n \times n_f)$ en OPLS, P-SOPLS, y SOPLS para tres problemas representativos.

P-SOPLS y la matriz $\mathbf{\Lambda}$ (la covarianza de los datos proyectados cuando se usa el algoritmo OPLS).

Como se esperaba, cuando $\gamma_1$ está próximo a cero, la matriz de datos proyectados obtenida con el método SOPLS es ortogonal, tendiendo su solución a la del OPLS; cuando $\gamma_1$ incrementa, la solución SOPLS pone la mayoría de sus coeficientes a cero, haciendo diferentes la soluciones SOPLS y OPLS. Sin embargo, el algoritmo P-SOPLS no presenta este comportamiento deseado de ortogonalidad (como se demuestra en el Subapartado 3.3.1.1). A pesar de la reducción de ortogonalidad de los datos proyectados cuando se añade la penalización $\ell_1$, si se presta atención al valor de $\gamma_1$ seleccionado por el proceso de CV (marcado con un círculo o un cuadrado en las curvas de la Figura 4.2), se puede observar que el algoritmo SOPLS propuesto tiende a seleccionar puntos de trabajo con soluciones que producen características más ortogonales que aquellas del P-SOPLS.

La ventaja de estas características ortogonales se puede ver claramente en la Figura 4.3, donde se muestra la precisión total frente al número de proyecciones usado $(1 \leq n_f \leq r)$ para los tres métodos bajo estudio: OPLS, SOPLS, y P-SOPLS. La aproximación propuesta mejora los resultados de P-SOPLS cuando se aplica un cuello de botella $(n_f < r)$, mostrando ventajas significativas en ocho de los nueve problemas. Este incremento de las prestaciones se debe al hecho de que las proyecciones obtenidas por el SOPLS son más ortogonales que aquellas del P-SOPLS, como se discutió con la Figura 4.2.

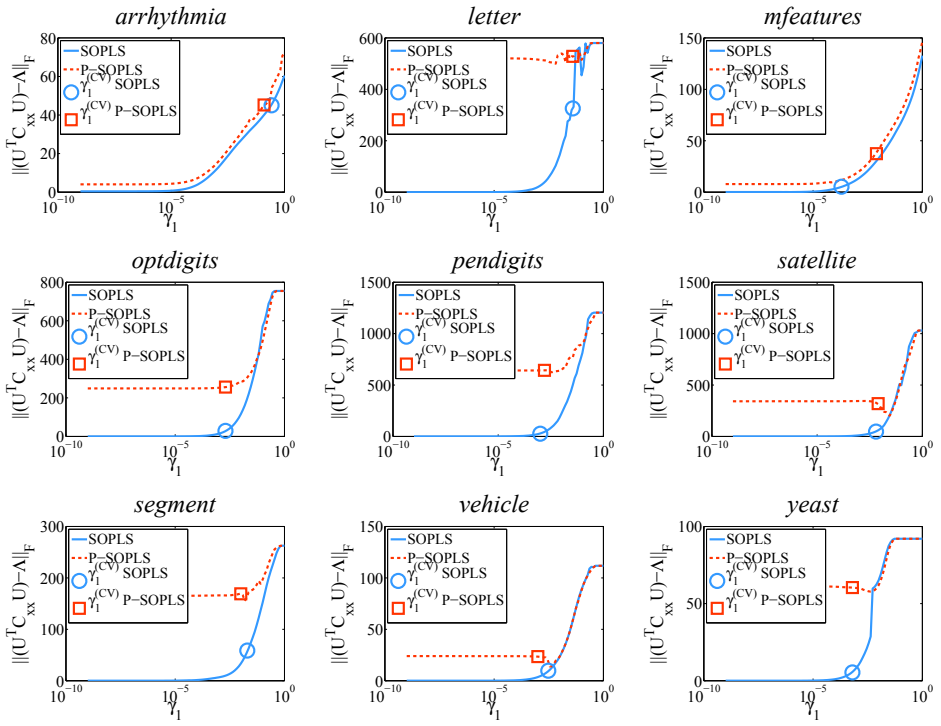



Figura 4.2: Distancia de Frobenius entre la matriz de covarianza de los datos proyectados cuando se usa el algoritmo SOPLS o P-OPLS y la matriz $\mathbf{\Lambda}$ (la covarianza de los datos proyectados cuando se usa el algoritmo OPLS). Los marcadores muestran el parámetro de penalización por la norma $\ell_1$ seleccionado por CV para ambos algoritmos.

### 4.2.3. Extracción de características dispersas para reconocimiento de caras

Con el objetivo de mostrar las ventajas de SOPLS sobre OPLS en un problema real, en este subapartado se analizarán las prestaciones de estos algoritmos sobre una base de datos de imágenes de caras. En particular, esta base de datos es un fragmento de "Labeled Faces in the Wild" (LFW)[3]. La base de datos completa contiene más de 13 000 imágenes de caras de 1 680 personas. Sin embargo, para poder trabajar con un conjunto de datos bien definido, se ha seleccionado únicamente a aquellas personas con al menos 20 imágenes disponibles. Esto da como resultado un conjunto reducido de 62 personas, compuesto por 2 276 imágenes de entrenamiento y 756 de test. El tamaño de las imágenes es de $50 \times 37$ píxeles, reordenados como un vector columna de 1 850 variables.

Para estudiar las ventajas de la dispersión inducida por la aproximación

[3] `http://vis-www.cs.umass.edu/lfw/lfw-funneled.tgz` (233MB)

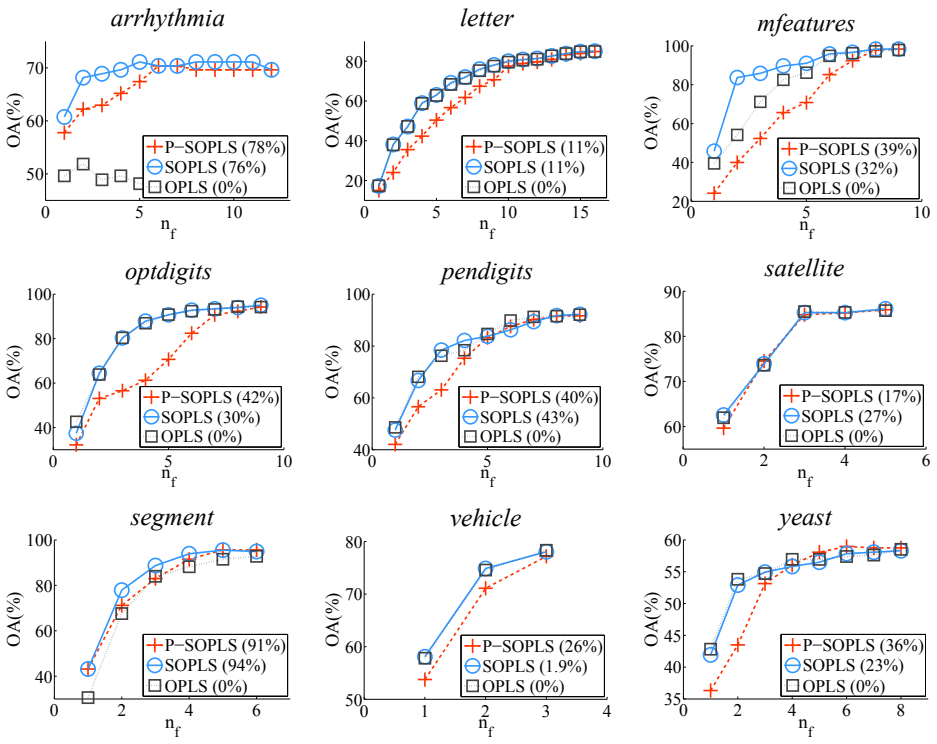
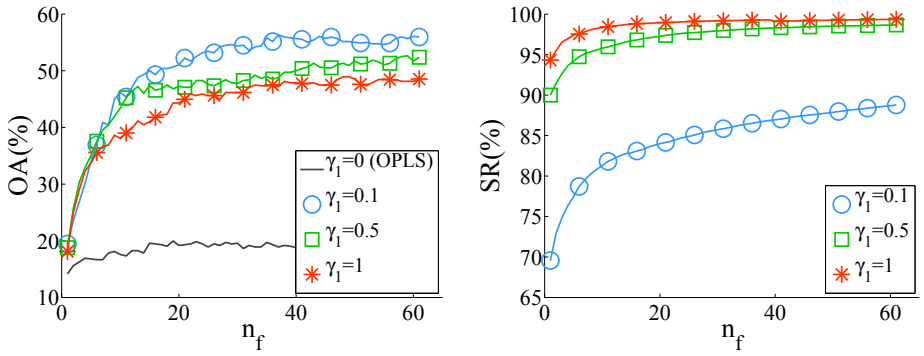



Figura 4.3: Precisión total (OA) (%) producida por los algoritmos OPLS, SOPLS y P-SOPLS para distintos números de características $n_f$. En la leyenda se muestran las tasas de dispersión (SR) alcanzadas cuando se usan todas las proyecciones ($n_f = r$).

SOPLS, se entrena el algoritmo con tres valores diferentes del parámetro de penalización, $\gamma_1 \in \{0{,}1, 0{,}5, 1\}$, de forma que se obtengan soluciones con diferentes grados de dispersión. Como criterio de parada, se ha fijado el número máximo de iteraciones a 50 y el parámetro de tolerancia $\delta$ a $10^{-5}$. Como en los subapartados anteriores, se entrena la C-SVM con las características extraídas para evaluar la precisión de los algoritmos OPLS y SOPLS.

En la Figura 4.4, se representa la precisión total (OA, izquierda) y la tasa de dispersión (SR, derecha) de las soluciones OPLS y SOPLS en función del número de características extraídas. Como se esperaba, la tasa de dispersión crece cuando se incrementa $\gamma_1$. Además, se puede ver que la introducción del término de regularización $\ell_1$ conduce a precisiones significativamente más altas. Esta ventaja se debe al hecho de que en esta aplicación la representación de los datos originales tiene un elevado número de características redundantes e irrelevantes, causando sobreajuste en la solución OPLS estándar, un problema que no sufren las versiones dispersas.

Para analizar la ventaja de las solución SOPLS desde el punto de vista




a) Overall Accuracy b) Sparsity Rate

Figura 4.4: Evolución de OA y SR conforme al número de proyecciones ($n_f$) obtenido por OPLS ($\gamma_1 = 0$) y SOPLS. Se analiza el comportamiento de SOPLS para distintos valores de $\gamma_1$. Como referencia, si se clasificase al azar, se obtendría una OA = 1,61 %.

de su interpretabilidad, la Figura 4.5 muestra los 6 primeros vectores de proyección obtenidos por las aproximaciones OPLS y SOPLS para diferentes valores de $\gamma_1$. Se puede apreciar que la solución OPLS no produce información útil alguna sobre la mayoría de las regiones relevantes usadas para clasificar las diferentes caras; sin embargo, si se observan los vectores de proyección producidos por la aproximación SOPLS, especialmente cuando se usa un valor alto de $\gamma_1$ ($\gamma_1 = 1$), se puede ver cómo los coeficientes no nulos se asocian a píxeles de regiones de los ojos y boca. Para valores pequeños de $\gamma_1$ ($\gamma_1 = 0{,}1$), la localización de los coeficientes no nulos no es muy informativa; sin embargo, incluso en este caso, SOPLS evita el problema de sobreajuste y funciona mucho mejor que el OPLS estándar.

## 4.3. Conclusiones

La implementación del algoritmo OPLS que se está usando con más frecuencia en el campo del aprendizaje máquina está basado en la solución a un problema de autovectores generalizado. En el capítulo anterior, se defendió una formulación general para los métodos MVA que admitía restricciones sobre los coeficientes de regresión, dando lugar a problemas de autovalores estándar y, por consiguiente, disfrutando de las siguientes dos ventajas: 1) los algoritmos resultantes requerían menos memoria y menos recursos de la CPU y 2) permitían implementar algoritmos MVA con nuevas restricciones como, por ejemplo, la dispersión, añadiendo un término de regularización $\ell_1$.

Explotando esta segunda ventaja, se han propuesto las implementaciones bloque y secuencial dispersas para el OPLS lineal (algoritmo SOPLS). Los resultados numéricos sobre unas bases de datos de referencia y sobre una

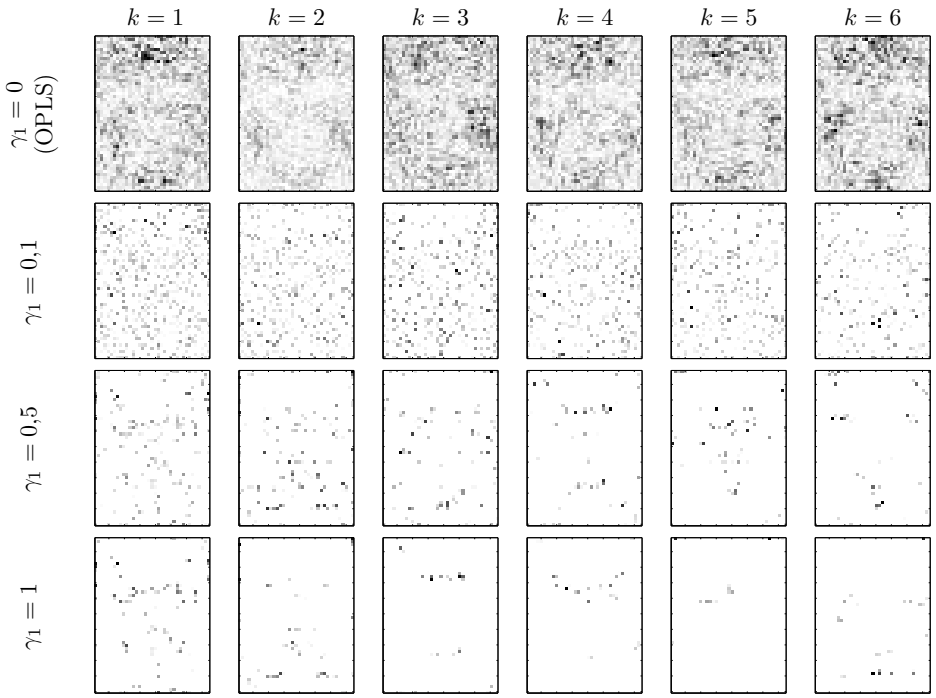



Figura 4.5: Seis primeros vectores de proyección para distintos valores de $\gamma_1$, correspondiendo $\gamma_1 = 0$ al algoritmo OPLS y $\gamma_1 > 0$ al algoritmo SOPLS

tarea de reconocimiento de caras confirman la eficiencia del algoritmo aquí propuesto. Además, con estos resultados, se confirma de forma empírica que las soluciones basadas en el problema ortogonal de Procrustes presentan los problemas discutidos en el capítulo anterior, aún cuando se incluye el término de regularización $\ell_1$.

En el Capítulo 5, se extenderá la aplicación de este término de regularización $\ell_1$ para soluciones no lineales. Además, en el Capítulo 6, se explorará la idea de incorporar términos de regularización que impongan dispersión sobre filas enteras de la matriz $\mathbf{U}$, de modo que todos los vectores de proyección dependan del mismo grupo de variables originales. Con estas soluciones parsimoniosas, se puede seleccionar a aquellas variables más relevantes para el problema en cuestión, problema abierto y muy interesante actualmente en el mundo real conocido como "Big Data".

Indicar nuevamente que aunque se ha considerado el caso OPLS, los resultados podrían extenderse directamente a cualquier otro método MVA que pueda inscribirse en el marco general del Capítulo 3.

# Capítulo 5

# MVA no lineal

*La corrupción de una cosa corresponde necesariamente a la generación de otra.*

Aristóteles (384 a. C.-322 a. C.)

**Resumen:** En este capítulo, se proponen dos métodos que extienden la idea de dispersión sobre la solución OPLS lineal, propuesta en el capítulo anterior, al ámbito no lineal o *kernel*. Estos dos esquemas propuestos obtienen soluciones dispersas en el espacio de las muestras, en lugar del de las variables de entrada; el segundo de ellos además permite incluir dispersión a priori sobre el número de muestras usadas, posibilitando así un mayor ahorro computacional. De este modo, se consigue una doble dispersión de la solución que, como se verá más adelante, permite mejorar el rendimiento de los algoritmos existentes hasta el momento.

## 5.1. Extensiones kernel de métodos MVA

Dado que las relaciones entre las variables son, a menudo, no lineales, en este Capítulo se atenderá esta necesidad proponiendo formulaciones no lineales capaces de capturar estas relaciones. Para llevar a cabo esta tarea, se va a hacer uso de los métodos núcleo o kernels, ya que son una herramienta muy útil para tal fin. En concreto, se prestará atención a la extensión kernel del algoritmo OPLS (KOPLS) (véanse Arenas-García y Petersen, 2009; Arenas-García et al., 2007), cuya formulación se presenta a continuación a modo de introducción de este capítulo.

A lo largo de este apartado, se considerará que los datos de entrada $\mathbf{X}$ son mapeados dentro de un *espacio de Hilbert generado por funciones*

*kernel* ("Reproducing Kernel Hilbert Space", RKHS) a través de una función de mapeo, $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathcal{F}$, donde la dimensión del espacio objetivo $\mathcal{F}$ es normalmente muy alta o incluso infinita. Los datos de entrenamiento son apilados juntos en la matriz $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)]$, de modo que ahora las $n_f$ proyecciones de los datos de entrada están dadas por $\bar{\boldsymbol{\Phi}} = \mathbf{U}^\top \tilde{\boldsymbol{\Phi}}$, donde $\tilde{\boldsymbol{\Phi}}$ es la versión centrada de $\boldsymbol{\Phi}$ y $\mathbf{U}$ es la matriz de proyección de tamaño $\dim(\mathcal{F}) \times n_f$. De este modo, la función de coste del OPLS (3.1) se puede reescribir en el espacio de características como,

$$\mathcal{L}_{\mathcal{F}}(\mathbf{W}, \mathbf{U}) = ||\mathbf{Y} - \mathbf{W}\mathbf{U}^\top \tilde{\boldsymbol{\Phi}}||_F^2. \tag{5.1}$$

Con el fin de resolver el problema anterior para el caso habitual en el que la dimensión de $\mathcal{F}$ es infinita, se hará uso del *Teorema de Representación* ("Representer's Theorem") (Shawe-Taylor y Cristianini, 2004), que especifica que los vectores de proyección se pueden expresar como una combinación lineal de los datos de entrada mapeados, $\mathbf{U} = \tilde{\boldsymbol{\Phi}}\mathbf{A}$, $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{n_f}]$ y $\boldsymbol{\alpha}_k$ parametriza el $k$-ésimo vector de proyección. Introduciendo esta expresión en (5.1), se obtiene

$$\mathcal{L}_{\mathcal{F}}(\mathbf{W}, \mathbf{A}) = ||\mathbf{Y} - \mathbf{W}\mathbf{A}^\top \mathbf{K}_{\boldsymbol{x}}||_F^2, \tag{5.2}$$

donde $\mathbf{K}_{\boldsymbol{x}} = \tilde{\boldsymbol{\Phi}}^\top \tilde{\boldsymbol{\Phi}}$ es la matriz kernel centrada, que conlleva únicamente productos internos en $\mathcal{F}$. Las diferentes funciones kernel para construir dichas matrices kernel y el proceso de centrado de estas matrices son explicados en detalle por Schoelkopf y Smola (2002) y por Shawe-Taylor y Cristianini (2004).

No obstante, KOPLS requiere la inversión de la matriz $\mathbf{K}_{\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}}$ que, por lo general, está mal condicionada, de modo que se hace necesario algún tipo de regularización. Más aún, cuando se trata de grandes conjuntos de datos, los requisitos computacionales y de memoria para manejar e invertir matrices kernel (de tamaño $N \times N$) hace que sea generalmente inviable trabajar con este método. Por estas razones, en el siguiente subapartado se dirige la atención al método KOPLS de complejidad reducida ("reduced KOPLS", rKOPLS) propuesta por Arenas-García et al. (2007), que fuerza soluciones parsimoniosas a priori y solventa algunos problemas prácticos inherentes al método KOPLS estándar.

Se mostrará asimismo cómo este método se puede beneficiar de la formulación basada en el problema de autovalores estándar, similar a aquella propuesta para EVD-OPLS. Se puede ver que (5.2) es formalmente equivalente a (3.1). Por lo tanto, las formulaciones KOPLS basadas en problemas de autovalores generalizado y estándar (GEV- y EVD-KOPLS respectivamente) se pueden obtener fácilmente reemplazando $\mathbf{U}$ por $\mathbf{A}$ y $\mathbf{X}$ por $\mathbf{K}_{\boldsymbol{x}}$ en las formulaciones lineales. Resulta interesante destacar que el ahorro computacional de la formulación EVD puede ser incluso aún más importante en este caso, ya que el tamaño del problema de la descomposición de la matriz

en GEV-KOPLS incrementa con $N$, mientras que EVD-KOPLS sigue suponiendo la descomposición de una matriz de tamaño $m \times m$. Es por esto que resulta sorprendente que Huang y De la Torre (2010), partiendo del método RRR consistente en la formulación EVD, lleguen a la extensión kernel GEV-KOPLS.

Al igual que en el caso lineal, la formulación EVD-rKOPLS presentada en la siguiente subapartado goza de dos ventajas principales: una mayor eficiencia con respecto al coste de la CPU y la posibilidad de imponer restricciones adicionales en los vectores de proyección. Esta segunda propiedad será explotada para obtener una formulación dispersa de rKOPLS al final de este apartado.

### 5.1.1. KOPLS reducido como un problema de autovalores estándar

La formulación rKOPLS, presentada por Arenas-García et al. (2007), está dada por $\mathbf{U} = \tilde{\mathbf{\Phi}}_R \mathbf{B}$, donde $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_f}]$ es la matriz de coeficientes del modelo reducido y $\tilde{\mathbf{\Phi}}_R$ es una matriz que contiene un subconjunto de $R$ datos de entrenamiento ($R < N$), seleccionados aleatoriamente[1]. Introduciendo la nueva expresión de $\mathbf{U}$ en (5.2), se obtiene la siguiente función de coste objetivo:

$$\mathcal{L}_{\mathcal{F}}(\mathbf{W}, \mathbf{B}) = ||\mathbf{Y} - \mathbf{W}\mathbf{B}^\top \mathbf{K}_R||_F^2, \tag{5.3}$$

donde $\mathbf{K}_R = \tilde{\mathbf{\Phi}}_R^\top \tilde{\mathbf{\Phi}}$ es una matriz de kernels de tamaño $R \times N$. En otras palabras: mientras que los vectores de proyección KOPLS se obtienen como una combinación lineal de todos los datos de entrenamiento ($\mathbf{U} = \tilde{\mathbf{\Phi}}\mathbf{A}$), rKOPLS fuerza dispersión a priori expresando los vectores de proyección como combinaciones lineales de un conjunto reducido de los datos de entrada. Cabe destacar las diferencias entre el concepto de *dispersión* en los algoritmos lineal y kernel: mientras que para el caso lineal, la dispersión es inducida sobre las variables originales de los datos, en KOPLS esta dispersión se refiere a la capacidad de estos métodos para expresar la solución en términos de un conjunto reducido de datos de entrenamiento que conlleva, principalmente, un ahorro computacional (tanto durante la fase de entrenamiento como en la fase de test). También es importante tener en cuenta que, dado que la matriz kernel $\mathbf{K}_R$ aún involucra a todos los datos de entrenamiento disponibles, rKOPLS resulta en una aproximación más potente que el mero submuestreo.

[1] Aquí se recurre a la estrategia de selección aleatoria que se usó en Arenas-García et al. (2007), aunque también se podrían haber usado otras estrategias más sofisticadas, como el submuestreo de Nyström (Williams y Seeger, 2001) o las características de Fourier aleatorias ("Random Fourier Features") (Yang et al., 2012), tanto para rKOPLS como para la versión dispersa que se presentará en el siguiente subapartado. Nótese que una selección más cuidadosa del subconjunto $\tilde{\mathbf{\Phi}}_R$ da lugar generalmente a una mayor precisión para un valor de $R$ fijado a expensas de una fase de entrenamiento más costosa.

Arenas-García et al. (2007) proponen una solución a (5.3) basada en un problema de autovalores generalizado (GEV-rKOPLS). Como alternativa, en este trabajo se propone reformular este problema como un problema de autovalores estándar. De nuevo, la derivación de la solución EVD-rKOPLS es directa dada la similitud entre (3.1) y (5.3): únicamente sería necesario reemplazar $\mathbf{U}$, $\boldsymbol{u}_k$, $\mathbf{C_{XX}}$ y $\mathbf{C_{XY}}$, respectivamente, por $\mathbf{B}$, $\boldsymbol{\beta}_k$, $\mathbf{K}_R\mathbf{K}_R^\top$ y $\mathbf{K}_R\mathbf{Y}^\top$. Entonces, se podría obtener un algoritmo EVD-rKOPLS bloque realizando los siguientes tres pasos:

1. $\mathbf{W}_{\text{LS}} = \left(\mathbf{K}_R\mathbf{K}_R^\top\right)^{-1}\mathbf{K}_R\mathbf{Y}$
2. $\mathbf{Y}\mathbf{K}_R^\top\mathbf{W}_{\text{LS}}\mathbf{W}_{\text{EVD}} = \mathbf{W}_{\text{EVD}}\mathbf{\Lambda}_{\text{EVD}}$
3. $\mathbf{B}_{\text{EVD}} = \mathbf{W}_{\text{LS}}\mathbf{W}_{\text{EVD}}$

Nótese que el parámetro $R$ actúa como un tipo de regularizador, haciendo que $\mathbf{K}_R\mathbf{K}_R^\top$ sea de rango completo. Esto también dicta los requisitos de cálculo y memoria del algoritmo, recuperándose la solución KOPLS cuando $R = N$. La Tabla 5.1 resume las principales características de KOPLS (véase Arenas-García y Petersen, 2009), GEV-rKOPLS y EVD-rKOPLS en términos de necesidades computacionales y de memoria. Nótese que el esquema EVD-rKOPLS propuesto es generalmente más eficiente en términos temporales y de almacenamiento que las otras dos soluciones.

Tabla 5.1: Tabla comparativa de los requisitos de memoria y coste computacional

| | GEV-KOPLS | GEV-rKOPLS | EVD-rKOPLS |
|---|---|---|---|
| Dimensiones de la matriz kernel | $N \times N$ | $R \times N$ | $R \times N$ |
| Requisitos de memoria | $\mathcal{O}(N^2)$ | $\mathcal{O}(R^2)$ | $\mathcal{O}(R^2)$ |
| Complejidad de GEV/EVD | $\mathcal{O}(N^3)$ | $\mathcal{O}(R^3)$ | $\mathcal{O}(m^3)$ |

### 5.1.2. rKOPLS disperso

La solución KOPLS estándar viene dada normalmente por una matriz de proyección densa $\mathbf{A}$. Por lo tanto, para extraer características de los nuevos datos, sería necesario calcular los kernels entre estos nuevos datos y todas las muestras de entrenamiento. El algoritmo rKOPLS alivia este problema imponiendo dispersión a priori sobre el número de kernels a calcular, hecho que conlleva ahorros computacionales y de memoria; aunque los vectores en $\mathcal{F}$, que definen la solución, se seleccionan de manera aleatoria. Debido a esto, rKOPLS no garantiza ni la selección de los datos de entrenamiento más representativos para la expansión ni la representación más dispersa.

Tratando de dar una solución a este problema, en este subapartado se añade un término de regularización $\ell_1$ en la función de coste objetivo

rKOPLS para inducir mayor dispersión en la solución en función de los vectores $\boldsymbol{\beta}_k$. De esta manera, el método selecciona automáticamente las muestras más representativas de $\tilde{\boldsymbol{\Phi}}_R$ y reduce el número de kernels que necesitan ser calculados para la proyección de nuevos datos. Nótese que, por brevedad, se presenta directamente la versión dispersa sobre el algoritmo rKOPLS en lugar de hacerlo primero sobre KOPLS, ya que cuando $R = N$ se obtendría la solución KOPLS dispersa (SKOPLS).

El nuevo esquema rKOPLS disperso, al que nos referiremos a partir de ahora como SrKOPLS, viene dado por la minimización de

$$\mathcal{L}_{\mathcal{F}} = ||\mathbf{Y} - \mathbf{W}\mathbf{B}^\top \mathbf{K}_R||_F^2 + \gamma_1 \|\mathbf{B}\|_1. \tag{5.4}$$

Imponer dispersión en la matriz $\mathbf{B}$ tiene efectos beneficiosos con respecto a la generalización y al coste computacional para los datos de test —se calculan menos kernels—, como se verá en la sección de experimentos. Además, se puede esperar que las soluciones sean más compactas, es decir, que la solución SrKOPLS reducirá el número de kernels necesarios para la extracción de características.

Para minimizar (5.4), es necesario recurrir a una formulación EVD que impone la restricción habitual $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$, para así poder llevar a cabo la minimización sin restricciones con respecto a $\mathbf{B}$. De este modo, se pueden usar nuevamente los algoritmos del Apartado 4.1, simplemente reemplazando $\mathbf{U}$, $\boldsymbol{u}_k$, $\mathbf{C_{XX}}$ y $\mathbf{C_{XY}}$ por $\mathbf{B}$, $\boldsymbol{\beta}_k$, $\mathbf{K}_R\mathbf{K}_R^\top$ y $\mathbf{K}_R\mathbf{Y}^\top$ respectivamente.

Una formulación bloque del algoritmo SrKOPLS consistiría en aplicar iterativamente los dos pasos siguientes:

1) Paso$-\mathbf{W}$: Fijado $\mathbf{B}$, encontrar $\mathbf{W}$ como la solución del siguiente problema de autovalores estándar $\mathbf{Y}\bar{\mathbf{K}}_R^\top\bar{\mathbf{K}}_R\mathbf{Y}^\top\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$, donde $\bar{\mathbf{K}}_R = \mathbf{B}^\top\mathbf{K}_R$.

2) Paso$-\mathbf{B}$: Fijado $\mathbf{W}$, resolver el problema *lasso* para minimizar (5.4) con respecto a $\mathbf{B}$ solamente.

Para esta formulación bloque, se ha usado la misma inicialización y el mismo criterio de parada que en la solución lineal.

Si se prefiere una implementación secuencial de SrKOPLS, en cada paso se resolvería un problema unidimensional seguido por la deflacción de la matriz $\mathbf{K}_R\mathbf{Y}^\top$. En este caso, se puede calcular la solución del Paso$-\mathbf{W}$ como

$$\boldsymbol{w}_k = \frac{\mathbf{Y}\bar{\boldsymbol{k}}_R^\top}{\|\mathbf{Y}\bar{\boldsymbol{k}}_R^\top\|}, \tag{5.5}$$

donde $\bar{\boldsymbol{k}}_R = \boldsymbol{\beta}_k^\top\mathbf{K}_R$. Con respecto al criterio de parada, se utiliza también el mismo criterio que se aplicó para el algoritmo SOPLS lineal:

$$d_{\cos}\left(\boldsymbol{u}_k^{(i)}, \boldsymbol{u}_k^{(i-1)}\right) = \frac{\boldsymbol{\beta}_k^{(k)\top}\mathbf{K}_{RR}\boldsymbol{\beta}_k^{(i-1)}}{\left(\boldsymbol{\beta}_k^{(k)\top}\mathbf{K}_{RR}\boldsymbol{\beta}_k^{(i)}\right)\left(\boldsymbol{\beta}_k^{(k-1)\top}\mathbf{K}_{RR}\boldsymbol{\beta}_k^{(i-1)}\right)}, \tag{5.6}$$

Tabla 5.2: Pseudocódigo del algoritmo SrKOPLS secuencial con deflacción

1.- Entradas: matrices centradas $\mathbf{K}_R$ y $\mathbf{Y}$, $n_f$, $\gamma_1$.
2.- Para $k = 1, \ldots, n_f$
  2.1.- Inicializar $\boldsymbol{\beta}_k^{(0)} = \mathbf{1}. * \delta_k$ ‡.
  2.2.- Para $i = 1, 2, \ldots$
    2.2.1.- Actualizar $\boldsymbol{w}_k^{(i)}$ usando (5.5).
    2.2.2.- Actualizar $\boldsymbol{\beta}_k^{(i)}$ resolviendo el problema *lasso* (5.4).
    2.2.3.- Si se cumple el criterio de convergencia, los valores actuales de salida. serían $\{\boldsymbol{\beta}_k, \boldsymbol{w}_k\}$, en caso contrario volver a 2.2.
  2.3.- Deflactar la matriz de covarianza cruzada: $\mathbf{Y}\mathbf{K}_R^\top \leftarrow \mathbf{Y}\mathbf{K}_R^\top - \boldsymbol{w}_k\boldsymbol{\beta}_k^\top \mathbf{K}_R\mathbf{K}_R^\top$.
3.- Salidas: $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_f}]$, $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_f}]$.

‡ El vector de proyección $\boldsymbol{\beta}_k$ se inicializa como un vector con su $k$-ésima componente igual a 1 y todas las demás componentes igual a 0.

requiriendo $d_{\cos}\left(\boldsymbol{u}_k^{(i)}, \boldsymbol{u}_k^{(i-1)}\right) > 1-\delta$, donde $\delta$ es un parámetro de tolerancia y $\mathbf{K}_{RR} = \mathbf{K}_R\mathbf{K}_R^\top$. En la Tabla 5.2, se proporciona el pseudocódigo para la implementación secuencial que se acaba de describir.

## 5.2. Experimentos

En este apartado, se analiza el poder discriminatorio de las soluciones no lineales SKOPLS y SrKOPLS. Con este propósito, se van a evaluar las prestaciones de estos métodos sobre los problemas multi-clase usados en el capítulo anterior y obtenidos de Frank y Asuncion (2010), cuyas propiedades fundamentales se recogen en la Tabla 4.2.

### 5.2.1. Extracción de características no lineales

En este subapartado se estudian las prestaciones obtenidas por las extensiones OPLS no lineales utilizando métodos kernel y la formulación EVD. Para evitar problemas computacionales de estas formulaciones, también se incluirán en este análisis sus versiones reducidas (rKOPLS y SrKOPLS) para poder estudiar el rendimiento de estos métodos cuando manejan grandes conjuntos de datos. Por esta razón, se evaluarán las prestaciones de SKOPLS y KOPLS en siete problemas de tamaño medio y bajo: *arrhythmia*, *mfeatures*, *optdigits*, *satellite*, *segment*, *vehicle*, y *yeast*; sus formulaciones reducidas (rKOPLS y SrKOPLS) serán analizadas sobre los mismos problemas que las versiones lineales, excepto para el problema de *arrhythmia* donde su reducido número de muestras de entrenamiento impide la aplicación del proceso de submuestreo.

Para todos los métodos bajo estudio, se ha usado un kernel Gaussiano

Tabla 5.3: Tabla comparativa entre los algoritmos KOPLS y SKOPLS en términos de la precisión total (OA). En el algoritmo SKOPLS, también se muestra la tasa de dispersión y el cociente entre el número de muestras útiles ($N_u$) y el total de muestras de entrenamiento ($N$).

| | KOPLS | SKOPLS | | |
|---|---|---|---|---|
| | OA(%) | OA(%) | SR(%) | $N_u/N$ (tasa %) |
| *arrhythmia* | 71.85 | **73.33** | 27.88 | 315/315 (100 %) |
| *mfeatures* | 96.33 | **96.67** | 86.03 | 878/1400 (62.71 %) |
| *optdigits* | **98.33** | 98.16 | 42.52 | 3809/3823 (99.63 %) |
| *satellite* | **91.45** | **91.45** | 44.86 | 4114/4435 (92.76 %) |
| *segment* | **95.5** | **95.5** | 75.78 | 847/1310 (64.65 %) |
| *vehicle* | 82.08 | **83.53** | 65 | 362/500 (72.4 %) |
| *yeast* | 58.3 | **60.54** | 94.31 | 244/1038 (23.51 %) |

con parámetro de dispersión $\sigma$,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right),$$

Una vez han sido extraídas las nuevas características de cada método, se entrena una C-SVM lineal para medir la capacidad de discriminación de cada subconjunto de características proyectadas.

Del mismo modo que en el Subapartado 4.2.1, se han ajustado los parámetro libres mediante un proceso de validación cruzada con 10 particiones ("10-fold CV"), seleccionando el parámetro $C$ de la SVM a partir del conjunto de valores $\{1, 10, 100, 1000\}$ y barriendo $\sigma$ en el conjunto $\{0{,}5, 1, 1{,}5, 2\} \times \sigma_0$, siendo $\sigma_0$ la mediana de la distancia entre todos los datos de entrada. En los métodos dispersos (SKOPLS y SrKOPLS), se ha validado el parámetro de regularización $\gamma_1$ en el conjunto de valores $\{10^{-7}, 10^{-6}, 10^{-5}\}$ y para el proceso iterativo se ha fijado un criterio de parada de $\delta = 10^{-12}$ con un número máximo de 500 iteraciones.

Debido a que las matrices kernel están, por lo general, mal condicionadas (rango($\mathbf{K}_{\boldsymbol{x}}$) $< N$), KOPLS necesita incluir un término de regularización $\ell_2$ para poder calcular su solución. Por este motivo, se ha incluido una penalización $\ell_2$ en los métodos KOPLS y SKOPLS, donde el parámetro de regularización se ha seleccionado mediante CV entre un conjunto de valores $\{10^{-9}, 10^{-8}, \ldots, 10^{-1}\}$ y $\{10^{-12}, 10^{-9}, 10^{-7}, 10^{-5}\}$ para los métodos KOPLS y SKOPLS respectivamente.

En la Tabla 5.3, se comparan los resultados obtenidos del algoritmo SKOPLS propuesto con los de KOPLS. Se puede ver que SKOPLS presenta prestaciones similares o mejores que el método KOPLS en todos los problemas excepto en *optdigits*. Además, debido a que la formulación dispersa de

Tabla 5.4: Precisión total (OA) y tasa de dispersión (SR) de los algoritmos rKOPLS y SrKOPLS para diferentes tamaños de subconjuntos de datos de entrenamiento ($R = 250$, 500 and 1000)

| | | $R = 250$ | | $R = 500$ | | $R = 1000$ | |
|---|---|---|---|---|---|---|---|
| | | rKOPLS | SrKOPLS | rKOPLS | SrKOPLS | rKOPLS | SrKOPLS |
| *letter* | OA | 90.9 | **91.44** | 93.14 | **93.38** | 94.52 | **94.55** |
| | SR | – | 6,79 % | – | 9,35 % | – | 3,27 % |
| *mfeatures* | OA | **98.31** | 98.05 | 97.97 | **98.53** | – | – |
| | SR | – | 18,89 % | – | 13,92 % | – | – |
| *optdigits* | OA | **97.45** | 97.40 | 97.77 | **98.01** | 98.15 | **98.17** |
| | SR | – | 6,74 % | – | 18,13 % | – | 37,82 % |
| *pendigits* | OA | 97.76 | **97.81** | 98.17 | **98.22** | 98.14 | **98.16** |
| | SR | – | 10,90 % | – | 19,74 % | – | 10,73 % |
| *satellite* | OA | **89.91** | 89.78 | **90.59** | 90.42 | 91 | **91.22** |
| | SR | – | 18,24 % | – | 10,64 % | – | 24,91 % |
| *segment* | OA | 95.98 | **96.11** | 95.58 | **95.75** | – | – |
| | SR | – | 29,75 % | – | 50,77 % | – | – |
| *vehicle* | OA | 80.58 | **81.96** | 80.26 | **81.56** | – | – |
| | SR | – | 57,41 % | – | 76,39 % | – | – |
| *yeast* | OA | 56.93 | **60.04** | 56.77 | **60.11** | – | – |
| | SR | – | 44,20 % | – | 44,93 % | – | – |

SKOPLS está destinada a eliminar muestras de los vectores de proyección, este método posee la ventaja adicional de reducir la complejidad computacional de la solución, produciendo tasas de dispersión (SRs) de alrededor del 40 % en *optdigits* y *satellite* o, incluso, del 80 % en *mfeatures*, *segment* y *yeast*. Estas altas tasas de dispersión implican importantes reducciones de la carga computacional, ya que en problemas tales como *yeast* únicamente sería necesario calcular el 20 % de los kernels para obtener los datos proyectados.

En la Tabla 5.4, se comparan las soluciones eficientes de KOPLS (rKOPLS) y de SKOPLS (SrKOPLS); debido a que la solución de estas aproximaciones depende de un proceso de submuestreo, la Tabla 5.4 incluye la precisión total (OA) como resultado de un promedio de 10 ejecuciones independientes. Esta eficiente técnica permite fijar *a priori* el grado de parsimonia (o tasa de dispersión *a priori*) que es, dicho de otro modo, el tamaño del subconjunto inicial de datos de entrenamiento seleccionado aleatoriamente ($R$). Para este experimento, se muestran las soluciones obtenidas con $R = 250$, $R = 500$ y $R = 1000$; por ejemplo, en el problema *vehicle*, la tasa de dispersión fijada a *a priori* para $R = 250$ es del 50 %.

Los resultados muestran, para cualquier valor de $R$, que el método SrKOPLS propuesto tiende a superar a rKOPLS en casi todos los problemas, permitiendo concluir que las proyecciones de SrKOPLS son más discriminatorias que aquellas de rKOPLS. Incluso en el caso donde se aplica un submuestreo más agresivo ($R = 250$), SrKOPLS mejora la precisión de rKOPLS

en cinco de los ocho problemas y es capaz de reducir, aún más, la complejidad de la solución; nótese que se obtienen tasas de dispersión (SR) de alrededor del 30 % en *segment* y cercanas al 60 % en *vehicle*.

## 5.3. Conclusiones

La implementación del algoritmo KOPLS que se está usando con más frecuencia en el campo del aprendizaje máquina está basada en la resolución de un problema de autovalores generalizado. En este capítulo, se ha revisado una formulación KOPLS que impone restricciones sobre los coeficientes de regresión, dando lugar a problemas de autovalores estándar. Al igual que en el capítulo anterior, se ha defendido este tipo de implementaciones por dos motivos principales: 1) los algoritmos resultantes requieren menos memoria y menos recursos de la CPU, y 2) permiten implementar el algoritmo KOPLS disperso añadiendo un término de regularización $\ell_1$.

Explotando esta segunda ventaja, se han propuesto las implementaciones bloque y secuencial de las extensiones no lineales mediante métodos kernel (algoritmo SrKOPLS). Los resultados numéricos sobre bases de datos de referencia confirman la eficiencia de los algoritmos aquí propuestos. El algoritmo SrKOPLS, supera las prestaciones del rKOPLS estándar en la mayoría de los problemas, con la ventaja adicional de obtener soluciones aún más dispersas.

En la actualidad, se está trabajando en la idea de incorporar términos de regularización que impongan dispersión sobre filas enteras de la matriz $\mathbf{B}$, de modo que todos los vectores de proyección dependan del mismo grupo de datos de entrenamiento. Con estas soluciones parsimoniosas, se podría seleccionar aquellas muestras más relevantes para el problema en cuestión, problema abierto y muy interesante actualmente en aplicaciones “Big Data”.

# Capítulo 6

# MVA para selección de variables

*Una palabra bien elegida puede economizar no solo cien palabras, sino cien pensamientos.*

Henri Poincaré (1854-1912)

**Resumen:** En la actualidad, existe una tendencia creciente en capturar indiscriminadamente una ingente cantidad de datos para poder sacar el mayor provecho de esa información. Sin embargo, una gran parte de esos datos, a menudo, carece de información relevante para la tarea a resolver o es redundante, ocasionando problemas de multicolinealidad. Además, tanto por cuestiones computacionales como de capacidad de almacenamiento, en ocasiones es deseable e, incluso, necesario descartar las variables nada informativas o redundantes. Tratando de cubrir esta necesidad dentro de los métodos MVA, en este capítulo, se proponen soluciones que permiten seleccionar aquellas variables relevantes para el fin deseado y, al mismo tiempo, lidiar con la información redundante tal que se anulan los efectos perniciosos de las multicolinealidades.

Las ventajas de esta propuesta son analizadas en un problema de regresión generado artificialmente y en dos problemas reales muy distintos: la clasificación de distintos carcinomas humanos y un sistema de reconocimiento facial.

## 6.1. Selección de variables relevantes en MVA

Actualmente está creciendo el uso de dispositivos personales —como los teléfonos móviles inteligentes, dispositivos *vestibles* ("weareables"), redes de

sensores, etc.— que constantemente capturan o, incluso, almacenan indiscriminadamente información potencialmente útil. Debido a esto, está surgiendo una explosión de datos disponibles a la espera de ser exprimidos con el fin de extraer conocimiento y, de este modo, sacar algún tipo de provecho. El problema ante este crecimiento exponencial de inmensas colecciones de datos es conocido como "Big Data" y las posibles soluciones a este problema, en cualquiera de los frentes abiertos —como la necesidad de almacenamiento, el tratamiento en tiempo real de los mismos o la algoritmia necesaria tanto para la extracción de información como para la visualización de esta para la ayuda a la toma de decisiones—, están en una fase aún demasiado inmadura.

Uno de los problemas abiertos dentro de este contexto consiste en extraer únicamente la parte relevante y útil de la ingente e intratable cantidad de datos disponible. El objetivo de esto, principalmente, sería detectar de manera eficiente aquellos patrones ocultos que pueden ayudar a tomar las mejores decisiones posibles.

Sin embargo, estas colecciones de datos, debido al modo indiscriminado de capturar este tipo de información, pueden ser difíciles de tratar o, incluso, ir en detrimento de los objetivos a conseguir, pues mucha de esta información puede ser redundante o, incluso, irrelevante a estos fines.

Una solución deseable sería seleccionar únicamente aquellas variables relevantes y descartar el resto. Este problema es conocido como *selección de variables*. En caso de enfrentarse con problemas con una única variable de salida, una de las soluciones más conocidas para este fin sería incluir una regularización $\ell_1$ en la formulación del problema o, si se dispusiese de información a priori sobre la estructura de los datos, utilizar el *Group Lasso* (Yuan y Lin, 2006). Con estas soluciones, se obtendrían valores próximos a cero en los coeficientes correspondientes a las variables irrelevantes o redundantes, permitiendo de este modo seleccionar las variables necesarias.

Por el contrario, si se trabaja con problemas con más de una dimensión de salida, como es el caso de esta tesis doctoral, cada variable no estaría representada simplemente por un coeficiente, sino por un vector de coeficientes. Este tipo de soluciones, donde se fuerza que todos los elementos de dicho vector sean cero, son conocidas como *soluciones parsimoniosas*. Para ello, Nie et al. (2010) propuso una implementación eficiente para obtener soluciones parsimoniosas consistente en imponer un término de regularización con la norma $\ell_{2,1}$ sobre la función de coste objetivo.

No obstante, este tipo de métodos de selección de variables asignan pesos —o vectores de pesos, en el caso que haya más de una variable de salida— no nulos a cada una de las variables y, aunque permite ordenar dichas variables por orden de relevancia (ranking de variables), dicho ranking puede no ser perfecto, pudiéndose preferir una variable redundante o ruidosa antes que una relevante. Además, independientemente de la calidad del ranking de variables, generalmente resulta difícil determinar el punto de corte para

discriminar las relevantes de las demás. Debido a un ranking de variables no perfecto devuelto por este tipo de métodos o a una mala selección de este punto de corte, esta selección de variables relevantes no es perfecta, seleccionándose también variables redundantes o, incluso ruidosas. A causa de esta detección no perfecta, las variables seleccionadas presentan multicolinealidades entre sí y, por lo tanto, las prestaciones pueden verse seriamente dañadas, causando, incluso, el sobreajuste sobre los datos de entrenamiento. Además, este efecto suele agravarse cuando se presentan datos de alta dimensionalidad, es decir, cuando hay el número de variables de entrada es mayor que el de muestras.

Una solución que hace frente a este problema de multicolinealidades —en el caso de más de una variable de salida—, como ya se ha visto, es la aplicación de los métodos MVA, que proyectan los datos de entrada a un espacio de menor dimensionalidad tal que las variables proyectadas de los datos de entrada estén incorreladas entre sí; de este modo, se eliminaría esa multicolinealidad presente en el espacio original.

Por lo tanto, dada la propiedad deseable de seleccionar variables informativas y dado el problema de la aparición inevitable de variables redundantes entre aquellas seleccionadas —causando así problemas de multicolinealidades en detrimento de las prestaciones—, el objetivo de este capítulo consiste en proponer métodos MVA que permitan seleccionar las variables relevantes para el problema a resolver y que, al mismo tiempo, puedan lidiar con las variables redundantes escogidas erróneamente, paliando así su efecto pernicioso sobre el resultado final.

### 6.1.1. Group Lasso y la norma $\ell_{2,1}$

El término *Group Lasso* es referido al término de regularización que, dada una estructura conocida del problema donde las variables están agrupadas en bloques disjuntos: $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_G\}$, permite detectar aquellos grupos de variables que tienen más relevancia para resolver el problema tratado y eliminar aquellos que son irrelevantes para tal fin.

Por lo general, la técnica Group Lasso es aplicada en problemas de regresión univariante o de clasificación binaria, es decir, en problemas donde la dimensión de salida es $m = 1$. En estos casos, el término de regularización aplicado se puede escribir como:

$$R(\boldsymbol{u}) = \sum_{g=1}^{G} \sqrt{\rho_g} ||\boldsymbol{u}^{\mathcal{G}_g}||_2,$$

donde $\rho_g$ es el número de variables de cada grupo y $\boldsymbol{u}^{\mathcal{G}_g}$ es el vector resultante de coger los coeficientes de $\boldsymbol{u}$ correspondientes a las variables pertenecientes al grupo $\mathcal{G}_g$, siendo $\boldsymbol{u}$ la solución del problema objetivo a optimizar. Nótese que si cada variable es un grupo distinto —es decir, no habría grupos—, la

regularización Group Lasso se convertiría en el término de regularización $\ell_1$ (véase el Apartado 2.1.1.1 para más detalle):

$$R(\boldsymbol{u}) = \sum_{g=1}^{G} |u_g| = ||\boldsymbol{u}||_1,$$

donde $u_g$ es el $g$-ésimo elemento del vector $\boldsymbol{u}$; nótese que la norma $\ell_2$ de un escalar ($||\cdot||_2$) es su valor absoluto $|\cdot|$ y $\rho_g = 1$ para todos los grupos, ya que cada grupo tendría un único elemento.

No obstante, en esta tesis doctoral, se está trabajando con problemas de regresión multivariante o de clasificación multiclase, es decir, cuando la dimensión de salida es $m \geq 2$. En estos casos, los coeficientes que antes estaban dispuestos en un vector columna $\boldsymbol{u} \in \mathbb{R}^{n\times 1}$ ahora constituyen una matriz $\mathbf{U} \in \mathbb{R}^{n\times m}$ y, por lo tanto, cada variable ya no está representada únicamente por un solo coeficiente, sino por un vector fila de $\mathbf{U}$ ($\boldsymbol{u}^k \quad \forall k = 1, \ldots, m$) de coeficientes.

En este tipo de problemas multivariante, si se dispone de información a priori sobre los grupos que forman las distintas variables de entrada, se podría reescribir el término de regularización Group Lasso como:

$$R(\mathbf{U}) = \sum_{g=1}^{G} \sqrt{\rho_g} ||\mathbf{U}^{\mathcal{G}_g}||_F,$$

donde, en este caso, $\mathbf{U}^{\mathcal{G}_g}$ es la matriz resultante de coger las filas de $\mathbf{U}$ correspondientes a las variables pertenecientes al grupo $\mathcal{G}_g$.

Puesto que en un problema de selección de variables no tiene por qué disponerse de conocimiento a priori, cada una de las $n$ variables de entrada sería un grupo distinto con $\rho_g = 1$, pues el tamaño de cada grupo es de una única variable. Por lo tanto, ahora el término de regularización resultante de este problema de selección de variables sin conocimiento de grupos a priori sería:

$$R(\mathbf{U}) = \sum_{g=1}^{n} ||\boldsymbol{u}^g||_2 = ||\mathbf{U}||_{2,1},$$

que, como se puede comprobar, es exactamente la norma $\ell_{2,1}$ (2.1) (véase el Subapartado 2.1.1.1 donde se revisa esta norma), siendo $\boldsymbol{u}^g$ la $g$-ésima fila de $\mathbf{U}$ correspondiente a la $g$-ésima variable.

Puesto que en este capítulo no se tiene en cuenta el conocimiento de grupos de variables a priori, se considerará simplemente la norma $\ell_{2,1}$ en lugar del término de regularización Group Lasso genérico.

### 6.1.2. Soluciones MVA para selección de variables

En este subapartado, se pretende explotar la propiedad de invarianza rotacional por filas que disfruta la norma $\ell_{2,1}$ con el propósito de incorporarla

a la formulación general MVA con restricciones.

Esta regularización permite obtener soluciones parsimoniosas, pudiendo así discriminar aquellas variables menos relevantes del problema. De este modo, el marco general de los métodos MVA para selección de variables podría describirse como la minimización de la función de coste

$$\begin{aligned}&\mathcal{L}(\mathbf{W},\mathbf{U}) = \|\mathbf{\Omega}^{\frac{1}{2}}\left(\mathbf{Y}-\mathbf{W}\mathbf{U}^\top\mathbf{X}\right)\|_F^2 + \gamma\|\mathbf{U}\|_{2,1},\\ &\text{sujeto a}: \ \mathbf{W}^\top\mathbf{\Omega}\mathbf{W}=\mathbf{I}.\end{aligned}$$

Con el fin de facilitar la exposición de la propuesta que se presenta en este apartado, se va a hacer el cambio de variable que se hizo para el marco general MVA en los Apartados 3.2 y 3.3: $\mathbf{W} = \mathbf{\Omega}^{-\frac{1}{2}}\mathbf{V}$. De este modo, se puede reescribir la función de coste objetivo como:

$$\begin{aligned}&\mathcal{L}(\mathbf{V},\mathbf{U}) = \|\mathbf{Y}'-\mathbf{V}\mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma\|\mathbf{U}\|_{2,1},\\ &\text{sujeto a}: \ \mathbf{V}^\top\mathbf{V}=\mathbf{I},\end{aligned}$$

donde $\mathbf{Y}' = \mathbf{\Omega}^{\frac{1}{2}}\mathbf{Y}$ sería la nueva matriz de salida. Cabe recordar que para el caso del OPLS y el PCA, donde $\mathbf{\Omega} = \mathbf{I}$, la nueva matriz de salida sería $\mathbf{Y}' = \mathbf{Y}$ e $\mathbf{Y}' = \mathbf{X}$, respectivamente; mientras que para el CCA, donde $\mathbf{\Omega} = \mathbf{C}_{\mathbf{YY}}^{-1}$, sería $\mathbf{Y}' = \mathbf{C}_{\mathbf{YY}}^{-\frac{1}{2}}\mathbf{Y}$.

Teniendo en cuenta la solución iterativa descrita en la Tabla 3.4 para los métodos MVA con restricciones, esta solución podría resolverse mediante un procedimiento iterativo consistente en la minimización de (3.42) y la resolución de (3.53), es decir, iterando sobre los dos siguientes pasos:

1. Paso$-\mathbf{U}$: Fijado $\mathbf{V}$, actualizar $\mathbf{U}$ resolviendo

$$\arg\min_{\mathbf{U}} \|\bar{\mathbf{Y}} - \mathbf{U}^\top\mathbf{X}\|_F^2 + \gamma\|\mathbf{U}\|_{2,1}, \tag{6.1}$$

   donde $\bar{\mathbf{Y}} = \mathbf{V}^\top\mathbf{Y}'$ es la proyección de la nueva matriz de salida.

2. Paso$-\mathbf{V}$: Fijado $\mathbf{U}$, actualizar $\mathbf{W}$ resolviendo

$$\mathbf{C}_{\mathbf{XY}'}^\top\mathbf{U}\mathbf{U}^\top\mathbf{C}_{\mathbf{XY}'}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^2, \tag{6.2}$$

   donde $\mathbf{C}_{\mathbf{XY}'} = \mathbf{C}_{\mathbf{XY}}\mathbf{\Omega}^{\frac{1}{2}}$.

Para resolver (6.1), se va a hacer uso de la solución eficiente e iterativa propuesta por Nie et al. (2010). Dicha solución a (6.1) se puede escribir como

$$\mathbf{U} = (\mathbf{C}_{\mathbf{XX}} + \gamma\mathbf{G})^{-1}\mathbf{C}_{\mathbf{X}\bar{\mathbf{Y}}} \quad \text{si} \quad N > n \tag{6.3}$$

o como

$$\mathbf{U} = \mathbf{G}^{-1}\mathbf{X}(\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X} + \gamma\mathbf{I})^{-1}\bar{\mathbf{Y}}^\top \quad \text{si} \quad n > N \tag{6.4}$$

donde (6.3) se usaría cuando el número de muestras es mayor que el de variables de entrada ($N > n$) y, en caso de tratarse de un problema de alta dimensionalidad ($n > N$), se usaría (6.4), donde se ha aplicado una de las identidades del conjunto de Searle (Searle, 1982).

La matriz $\mathbf{G}$ que aparece en (6.3) y (6.4) es una matriz diagonal donde el $i$-ésimo elemento de su diagonal viene dado por

$$G_{ii} = \frac{1}{2\|\boldsymbol{u}^i\|_2}, \tag{6.5}$$

siendo $\boldsymbol{u}^i$ la $i$-ésima fila de $\mathbf{U}$. Para evitar problemas de inestabilidad numérica cuando $\|\boldsymbol{u}^i\|_2 = 0$, se podría usar una constante pequeña $\epsilon$ que tienda a cero (por ejemplo, $\epsilon = 10^{-16}$) para calcular cada elemento de la diagonal de $\mathbf{G}$ como $G_{ii} = \frac{1}{2\sqrt{\|\boldsymbol{u}^i\|_2^2+\epsilon^2}}$.

Respecto a cuestiones de implementación, se observa que existen dos procesos iterativos a distinto nivel: uno entre los Pasos–$\mathbf{U}$ y –$\mathbf{V}$, y otro entre los cálculos de $\mathbf{U}$ y $\mathbf{G}$. Sin embargo, como los dos están en función de la misma matriz $\mathbf{U}$, se podrían solapar en un único procedimiento iterativo como se resume en la Tabla 6.1.

Tabla 6.1: Pseudocódigo del algoritmo MVA iterativo con norma $\ell_{2,1}$

1.- Entradas: matrices centradas $\mathbf{X}$ e $\mathbf{Y}$, $\boldsymbol{\Omega}$ y $\gamma$.
- 2.1.- Inicializar $\mathbf{V}^{(0)} = \mathbf{I}$, $\mathbf{G}^{(0)} = \mathbf{I}$.
- 2.2.- Para $k = 1, 2, \ldots$
  - 2.2.1.- Actualizar $\mathbf{U}^{(k)}$ en función de $\mathbf{V}^{(k-1)}$ y $\mathbf{G}^{(k-1)}$ usando (6.3) o (6.4).
  - 2.2.2.- Actualizar $\mathbf{V}^{(k)}$ en función de $\mathbf{U}^{(k)}$ usando (6.2).
  - 2.2.3.- Actualizar $\mathbf{G}^{(k)}$ en función de $\mathbf{U}^{(k)}$ usando (6.5).
  - 2.2.4.- Si se cumple el criterio de convergencia, ir a 3.

3.- Salidas: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{G}$.

A partir de este punto, resulta muy interesante explotar la invarianza rotacional por filas de la norma $\ell_{2,1}$ (véase el Apartado 2.1.1.1 para más detalle). Para ello, se puede ver que la actualización de $\mathbf{U}$ depende únicamente de la norma vectorial $\ell_2$ de cada una de sus filas. Por lo tanto, si se reescribiese $\mathbf{U} = \mathbf{U}'\mathbf{V}$, siendo

$$\mathbf{U}' = \begin{cases} (\mathbf{C}_{\mathbf{XX}} + \gamma\mathbf{G}^{(k-1)})^{-1}\mathbf{C}_{\mathbf{XY}'} & \text{si } n < N \\ \mathbf{G}^{(k-1)^{-1}}\mathbf{X}(\mathbf{X}^\top\mathbf{G}^{(k-1)^{-1}}\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{Y}'^\top & \text{si } n > N, \end{cases}$$

se puede reescribir cada elemento de la diagonal de $\mathbf{G}$ únicamente en función

de $\mathbf{U}'$, puesto que

$$G_{ii} = \frac{1}{2\|\boldsymbol{u}^i\|_2} = \frac{1}{2\|\boldsymbol{u}'^i\mathbf{V}\|_2} = \frac{1}{2\sqrt{\boldsymbol{u}'\mathbf{V}\mathbf{V}^\top\boldsymbol{u}'^\top}} = \frac{1}{2\|\boldsymbol{u}'^i\|_2}.$$

Esto es debido a que $\mathbf{V}$ es la solución a un problema de autovalores estándar, es decir, es ortogonal, cumpliéndose $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$. Por lo tanto, el procedimiento iterativo entre los Pasos–$\mathbf{U}$ y –$\mathbf{V}$ (para el cálculo de $\mathbf{V}$) sería innecesario, pudiéndose calcular $\mathbf{V}$ fuera del bucle una vez se ha calculado $\mathbf{U}'$ —es decir, tras haber finalizado el proceso iterativo entre $\mathbf{U}'$ y $\mathbf{G}$—. Finalmente, los vectores de proyección de entrada se obtendrían como $\mathbf{U} = \mathbf{U}'\mathbf{V}$.

Por lo tanto, debido a esta propiedad de invariancia rotacional que disfruta la norma $\ell_{2,1}$, el coste computacional de esta nueva solución quedaría reducida tantas veces el coste del cálculo de $\mathbf{V}$ como el número de iteraciones que tardaría en converger la implementación descrita en la Tabla 6.1.

Cabe destacar que esta última solución sería similar al marco general MVA introducido en el Apartado 3.2 y se resumiría en los siguientes tres pasos:

1. $\mathbf{U}'^* = \arg\min_{\mathbf{U}'} \|\mathbf{Y}' - \mathbf{U}'^\top\mathbf{X}\|_F^2 + \gamma\|\mathbf{U}'\|_{2,1}$,
2. $\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'\mathbf{U}'^\top\mathbf{C}_{\mathbf{X}\mathbf{Y}'}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}^2$,
3. $\mathbf{U} = \mathbf{U}'\mathbf{V}$.

Nótese que, aunque la derivación seguida en el Apartado 3.2 para el cálculo de la solución del marco general MVA sería igualmente válida para obtener esta solución, el cálculo de $\mathbf{V}$ se realizaría con el problema de autovalores estándar

$$\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}.$$

Esto sería válido, siempre y cuando $\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'$ sea una matriz simétrica, ya que el problema de autovalores estándar descompone únicamente este tipo de matrices (véase el subapartado 2.1.3). Para evitar este problema, habría únicamente que premultiplicar su transpuesta por la izquierda:

$$\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'\mathbf{V}\mathbf{V}^\top\mathbf{U}'\top\mathbf{C}_{\mathbf{X}\mathbf{Y}'} = \mathbf{V}\boldsymbol{\Lambda}^2\mathbf{V}^\top$$
$$\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'\mathbf{U}'^\top\mathbf{C}_{\mathbf{X}\mathbf{Y}'}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}^2,$$

llegando a la misma solución que en el Paso 2 del algoritmo que se acaba de describir[1].

En la Tabla 6.2, se resume la generalización del algoritmo no iterativo (versión 2) que se acaba de describir para los métodos MVA (L21MVA) con la capacidad de seleccionar las variables de entrada más relevantes.

[1]Se puede extender el marco MVA presentado en el Apartado 3.2 para casos donde la matriz $\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}$ no es simétrica y el término de regularización es invariante rotacional; para estos casos, modificando el segundo paso, el marco MVA debería reescribirse como: 1) $\mathbf{U}' = \arg\min_{\mathbf{U}'} \|\mathbf{Y}' - \mathbf{U}'^\top\mathbf{X}\|_F^2 + \gamma R(\mathbf{U}')$; 2) $\mathbf{C}_{\mathbf{X}\mathbf{Y}'}^\top\mathbf{U}'\mathbf{U}'^\top\mathbf{C}_{\mathbf{X}\mathbf{Y}'}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}^2$; y 3) $\mathbf{U} = \mathbf{U}'\mathbf{V}$.

Tabla 6.2: Pseudocódigo del algoritmo MVA alternativo con norma $\ell_{2,1}$

1.- Entradas: matrices centradas $\mathbf{X}$ e $\mathbf{Y}$, $\mathbf{\Omega}$, $\gamma$.

2.1.- Inicializar $\mathbf{G}^{(0)} = \mathbf{I}$ e $\mathbf{Y}' = \mathbf{\Omega}^{\frac{1}{2}}\mathbf{Y}$.

2.2.- Para $k = 1, 2, \ldots$

2.2.1.- $$\mathbf{U}' = \begin{cases} (\mathbf{C_{XX}} + \gamma\mathbf{G}^{(k-1)})^{-1}\mathbf{C_{XY'}} & \text{si } \; n < N \\ \mathbf{G}^{(k-1)^{-1}}\mathbf{X}(\mathbf{X}^\top\mathbf{G}^{(k-1)^{-1}}\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{Y}'^\top & \text{si } \; n > N \end{cases}$$

2.2.2.- $G_{ii} = \dfrac{1}{2\|\boldsymbol{u}'^i\|_2}$, para $i = 1, \ldots, n$.

2.2.3.- Si se cumple el criterio de convergencia, ir a 2.3.

2.3.- $\mathbf{C}^\top_{\mathbf{XY'}}\mathbf{U}'\mathbf{U}'^\top\mathbf{C_{XY'}}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^2$.

2.4.- $\mathbf{U} = \mathbf{U}'\mathbf{\Omega}^{\frac{1}{2}}\mathbf{V}$.

3.- Salidas: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{G}$.

Puesto que, en los algoritmos descritos en la Tabla 6.2, el problema de autovalores estándar se calcula una única vez, es preferible proceder siempre con esta implementación, ya que, como se comentó anteriormente, se obtiene un considerable ahorro del coste computacional. Para diferenciar los distintos algoritmos entre las versiones 1 —solución de la Tabla 6.1— y 2 —solución de la Tabla 6.2—, se denotará con los sufijos (v1) y (v2), respectivamente, tras el nombre de cada método.

En el Paso 2.2.4 del algoritmo de la Tabla 6.1 y en el Paso 2.2.3 del correspondiente a la Tabla 6.2, se pueden utilizar distintos criterios de convergencia. En el apartado de experimentos, se ha usado el mismo mecanismo de parada para todos los algoritmos: $\|\operatorname{diag}(\mathbf{G}^{(k)}) - \operatorname{diag}(\mathbf{G}^{(k-1)})\|_2 \leq \delta$, donde los superíndices indexan la iteración, el operador "diag" extrae un vector con los elementos de la diagonal de la matriz correspondiente y $\delta$ es una pequeña constante. De esta manera, el algoritmo se detiene cuando las soluciones obtenidas en dos iteraciones consecutivas difieren menos de un pequeño umbral.

Es interesante comentar que el algoritmo propuesto por Shi et al. (2014) para tareas de clasificación denominado L21SDA —consistente en imponer la norma $\ell_{2,1}$ a una versión dispersa del LDA ("Linear Discriminant Analysis") denominado SDA ("Sparse Discriminant Analysis")— equivaldría al algoritmo descrito en la Tabla 6.1 para $\mathbf{\Omega} = \mathbf{C}^{-1}_{\mathbf{YY}}$ (es decir, al CCA con norma $\ell_{2,1}$ iterativo propuesto aquí), con una única, pero importante diferencia: el L21SDA usa la aproximación de Procrustes para resolver el paso 2.2.2 del algoritmo de la Tabla 6.1, es decir, para calcular $\mathbf{V}$. Como se comentó en el Apartado 3.3.1, las graves consecuencias de usar este paso son tanto la incapacidad de obtener características incorreladas entre sí como la falta de

garantía en la convergencia del algoritmo. La consecuencia de esto se puede ver en las propias curvas del artículo de Shi et al. (2014) tanto para SDA como para L21SDA, donde las prestaciones para un subconjunto de características son mucho más bajas que las demás, debido a la imposibilidad de obtener por orden de relevancia las características extraídas. Esto también se discutirá en el apartado de experimentos.

Se puede encontrar una propuesta similar para el caso del OPLS, donde Chen y Huang (2012) presentan una solución OPLS con norma $\ell_{2,1}$ denominada SRRR. Dicha propuesta es formulada como un método RRR con un término de regularización "Group Lasso", donde se considera cada variable como un grupo distinto (es decir, la norma $\ell_{2,1}$) y lo resuelve con un método que denomina "Variational Group Lasso" que consiste en la misma implementación propuesta por Nie et al. (2010) —usada también aquí—. Al igual que en el caso del L21SDA, la diferencia con los algoritmos aquí propuestos es que emplean la aproximación de Procrustes, conllevando los problemas comentados anteriormente. Además, el método SRRR es mucho más costoso computacionalmente que los algoritmos tanto de la Tabla 6.1 (v1) como de la Tabla 6.2 (v2), ya que ejecuta los dos procedimientos iterativos anidados: iteran entre el Paso–$\mathbf{U}$ y el Paso–$\mathbf{V}$ y dentro del Paso-$\mathbf{U}$ iteran también entre $\mathbf{U}$ y $\mathbf{G}$.

A diferencia del L21SDA o del SRRR, la inicialización para $\mathbf{V}$ del algoritmo de la Tabla 6.1 no es crítica, convergiendo en cualquier caso a la misma solución que el algoritmo de la Tabla 6.2 que no requiere de esa inicialización. En el apartado de experimentos se inicializa $\mathbf{V}$ con la matriz identidad.

## 6.2. Experimentos

En este apartado de experimentos, se pretende mostrar la capacidad que tienen los métodos MVA para tratar con datos que presentan alta multicolinealidad entre las variables de entrada en tareas de selección de variables.

Para ello, se van a llevar a cabo tres experimentos distintos: 1) se compararán los algoritmos propuestos L21OPLS y L21CCA con la implementación eficiente y robusta ante muestras defectuosas ("outliers") del problema LS con norma $\ell_{2,1}$ ("Robust Feature Selection", RFS) propuesta por Nie et al. (2010) en un problema de regresión de alta dimensionalidad generado artificialmente para introducir un alto grado de multicolinealidad entre las variables de entrada; 2) se compararán todos los métodos descritos anteriormente con RFS en dos problemas reales de alta dimensionalidad que presentan multicolinealidad para una tarea de reconocimiento de caras y otra de clasificación de distintos carcinomas humanos a partir de chips de ADN (o microarrays), que analizan las expresiones génicas; y 3) se compara el L21CCA iterativo sin y con el uso de la aproximación de Procrustes (L21SDA) en función del número de características extraídas con el fin de ilustrar los problemas

derivados del uso de Procrustes. A diferencia de este último experimento, los dos primeros tienen como objetivo mostrar la capacidad que tienen los métodos propuestos de lidiar con los problemas ocasionados por la selección de variables que presentan multicolinealidades entre sí.

### 6.2.1. Problema de regresión con alta multicolinealidad

En este subapartado, se ha generado un problema de regresión artificial sencillo que introduce multicolinealidad entre las variables de entrada para poder analizar las prestaciones de los algoritmos propuestos con respecto al estado del arte en selección de variables. De este modo, se controla la cantidad de variables relevantes, redundantes y ruidosas introducidas en el problema.

El espacio de entrada $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ estará compuesto por $n = 4000$ variables aleatorias divididas en tres grupos: $n_{relev} = 500$ variables relevantes, generadas siguiendo una distribución Gaussiana de media 0 y varianza seleccionada aleatoriamente entre 0 y 4; $n_{redund} = \frac{n}{2} = 2000$ variables redundantes, obtenidas como combinación lineal de variables relevantes; y las $n_{ruid} = 1500$ restantes variables ruidosas, es decir, variables Gaussianas independientes con media 0 y varianza unidad. Por lo tanto, sin pérdida de generalidad, se pueden agrupar todas estas variables en una única observación como $\boldsymbol{x} = (\boldsymbol{x}_{relev}^\top, \boldsymbol{x}_{redund}^\top, \boldsymbol{x}_{ruid}^\top)^\top$.

Entonces, el modelo de regresión construido para estimar el vector de salida $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$, siendo $m = 10$ el número de variables de salida, es el siguiente:

$$\boldsymbol{y} = \begin{pmatrix} W_{relev} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{x} + \epsilon,$$

donde $\epsilon$ es un vector de ruido Gaussiano con media 0 y varianza $10^{-6}$, $W_{relev} \in \mathbb{R}^{m\times n_{relev}}$ es una matriz fijada con sus elementos seleccionados aleatoriamente con una distribución uniforme entre $-1$ y $+1$ y $\mathbf{0}$ es una matriz con todo ceros del correspondiente tamaño. De este modo, la matriz de pesos total es construida tal que $\boldsymbol{y}$ dependa únicamente de las variables relevantes de entrada.

Se va a usar un conjunto de $N = 500$ muestras de entrenamiento ($\mathbf{X} \in \mathbb{R}^{n\times N}$ e $\mathbf{Y} \in \mathbb{R}^{m\times N}$) y se evaluará con 210 observaciones de test, siendo un conjunto total de 710 datos correspondiendo a una partición 70/30(%) para los conjunto de entrenamiento y test, respectivamente. Ambos conjuntos de datos están centrados y normalizados por la desviación típica de cada variable.

Se ha usado el mismo criterio de parada para todos los algoritmos comparados, deteniendo la ejecución al llegar a un máximo de 50 iteraciones o cuando la norma de Frobenius de la diferencia entre las soluciones obtenidas en dos iteraciones consecutivas es menor que un valor de tolerancia $\delta = 10^{-6}$.

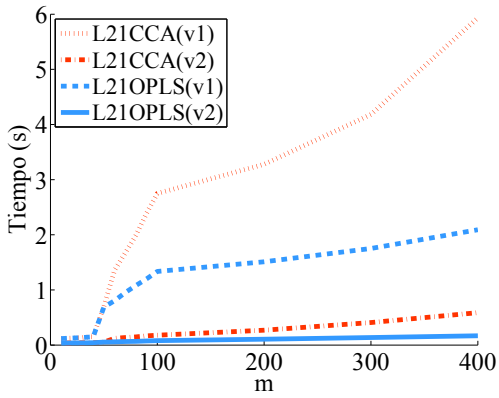


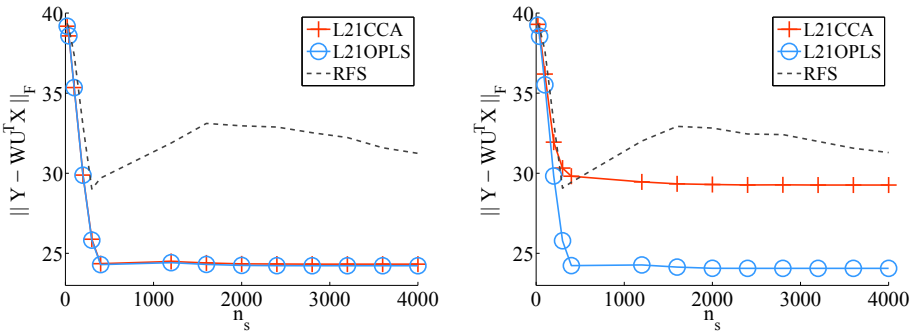

Figura 6.1: Tiempo (en segundos) que requieren las dos versiones (v1) y (v2) de los algoritmos L21MVA propuestos en función del número de variables de salida ($m$) —obtenido como promedio de 10 realizaciones independientes—. A modo representativo, se ha reducido el tamaño del problema una decima parte, siendo el número de variables de entrada $n = 400$ y el número de muestras usadas $N = 50$.

Los resultados obtenidos son el promedio de 10 ejecuciones aleatorias independientes sobre distintos conjuntos de datos.

La selección de variables se lleva a cabo cogiendo las $n_s < n$ mejores variables tras ser ordenadas por orden de relevancia según el correspondiente valor de $\|\boldsymbol{u}'_i\|$ o $\|\boldsymbol{u}_i\|$ (con $i = 1, \ldots, n$) para RFS o para los métodos MVA, respectivamente. Una vez se han obtenido las $n_s$ variables, se calcula el regresor óptimo en sentido MSE, bien usando como entrada las $n_s$ variables originales seleccionadas en el caso de RFS, bien las $n_f$ características extraídas a partir de las $n_s$ variables por los algoritmos L21MVA.

Para el estudio comparativo, se ha usado la versión 2 (v2) de los algoritmos MVA propuestos, ya que, además de obtener la misma solución que la versión 1 (v1), conllevan un coste computacional considerablemente menor. En la Figura 6.1, se muestra una comparación de tiempos de ejecución (en segundos) en función del número de variables de salida ($m$) entre las dos versiones del CCA y del OPLS. Como se puede observar, la versión 2 (v2) de ambos métodos aumenta la eficiencia con respecto a la versión 1 (v1) a medida que crece $m$. Además, L21CCA(v1) escala bastante mal con $m$, pues CCA requiere de una operación adicional de coste $\mathcal{O}(m^3)$ debido al cálculo de $\boldsymbol{\Omega} = \mathbf{C}_{\mathbf{YY}}^{-1/2}$.

En la Figura 6.2, se muestra el error cuadrático medio (MSE) obtenido por los algoritmos propuestos L21OPLS y L21CCA y por el algoritmo de referencia RFS en función del número de variables seleccionadas. En el caso de los métodos MVA, se han usado todas las características extraídas. Nótese que las prestaciones del algoritmo L21SDA usando todas las características




(a) $\gamma = 0{,}5$ (b) $\gamma = 100$

Figura 6.2: Curvas comparativas en términos de MSE según el número de variables seleccionadas ($n_s$)

serían las mismas que con L21CCA, así que se referenciará en la leyenda como este último. En la subfigura (a), se seleccionó como parámetro de penalización $\gamma = 0{,}5$, mientras que en la subfigura (b), se usó $\gamma = 100$. Como se puede ver, para este problema, la influencia del parámetro de penalización en las prestaciones de los métodos que minimizan el MSE es bastante débil —mejorando incluso un poco el L21OPLS con $\gamma = 100$—, mientras que para las extensiones del CCA —incluyendo L21SDA— es muy notable, empeorando significativamente las prestaciones cuanto mayor es $\gamma$. Con esto se podría pensar que los métodos que minimizan el MSE son más robustos frente a cambios del parámetro de penalización $\gamma$.

A la vista de los resultados de la Figura 6.2, se puede concluir que ante un conjunto de datos difícil de tratar por su alta dimensionalidad y multicolinealidad, los métodos MVA pueden lidiar con este tipo de problemas, incluso, en tareas de selección de variables. La multicolinealidad, como ya se ha comentado, produce problemas serios de sobreajuste. Este es el caso del método RFS que, aunque es un método robusto ante "outliers", sufre de un grave sobreajuste provocado por la información redundante del problema. También es interesante comentar que los métodos MVA, debido a la proyección que hacen de las variables seleccionadas, obtienen unas características ortogonales entre sí, eliminando en gran medida las multicolinealidades de las posibles variables redundantes seleccionadas. Como se puede observar, cuando se seleccionan las 500 primeras variables, en su mayoría relevantes debido al término de regularización $\ell_{2,1}$, se obtienen las mejores prestaciones. Estos buenos resultados se deben a la combinación de las selección de variables junto con la extracción de características en el espacio proyectado, donde estas variables seleccionadas han sido blanqueadas. Por lo tanto, una vez elegidas todas las variables relevantes, las prestaciones se mantienen estables gracias a ese blanqueamiento de las variables seleccionadas.

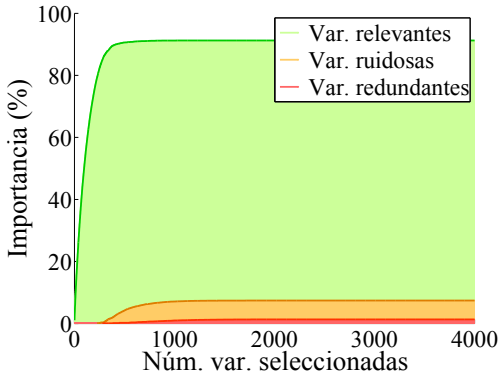



Figura 6.3: Relación de importancia acumulada aportada por las variables seleccionadas del problema

Con el fin de evaluar la capacidad de selección de variables relevantes de los algoritmos propuestos, se muestra en la Figura 6.3 el porcentaje de importancia acumulada según el ranking de variables generado por la regularización $\ell_{2,1}$ (donde la importancia de cada variable es estimada como $\|\boldsymbol{u}_i\|$ para $i = 1, \ldots, n$). En este caso, no se ha observado diferencia alguna en el ranking devuelto por los algoritmos L21CCA, L21OPLS y RFS y, por lo tanto, la Figura 6.3 es igualmente válida para los tres métodos. Como se puede observar, las primeras variables que se seleccionarían serían las relevantes, llegando a su máximo del 90 % de importancia en torno a las 500 primeras variables seleccionadas que es justamente el número de variables relevantes del problema ($n_{relev} = 500$). Las variables ruidosas y redundantes se seleccionarían después; sin embargo, entre las primeras 500 variables seleccionadas, aparecen tanto ruidosas como relevantes. Esta es la razón de las malas prestaciones de RFS cuando se seleccionan solamente 500 variables en comparación con los algoritmos MVA, ya que estos últimos, además de reducir la dimensionalidad, blanquean las variables seleccionadas, cancelando la multicolinealidad originada por esa redundancia.

### 6.2.2. Problemas de clasificación reales de alta dimensionalidad y multicolinealidad

En este subapartado, se pretende mostrar la utilidad de las propuestas hechas en este capítulo en problemas del mundo real. Se han seleccionado dos problemas de clasificación de alta dimensionalidad y con multicolinealidades entre sus variables, cuyas características principales se resumen en la Tabla 6.3:

- "Human Carcinomas Data Set" (*Carcinomas*) (Su et al., 2001; Yang et al., 2006): El conjunto de datos de carcinomas humanos está compuesto por un total de 174 muestras correspondientes a 11 clases distintas:

Tabla 6.3: Principales propiedades de los problemas de referencia seleccionados: número de muestras de entrenamiento ($N_{train}$) y test ($N_{test}$), variables de entrada ($n$), variables de salida ($m$) y número de imágenes de entrenamiento por persona ($p$)

| | $N_{train}/N_{test}$ | $n$ | $m$ |
|---|---|---|---|
| *Carcinomas* | 139 / 35 | 9182 | 11 |
| *Yale* ($p = 8$) | 120 / 45 | 1024 | 15 |

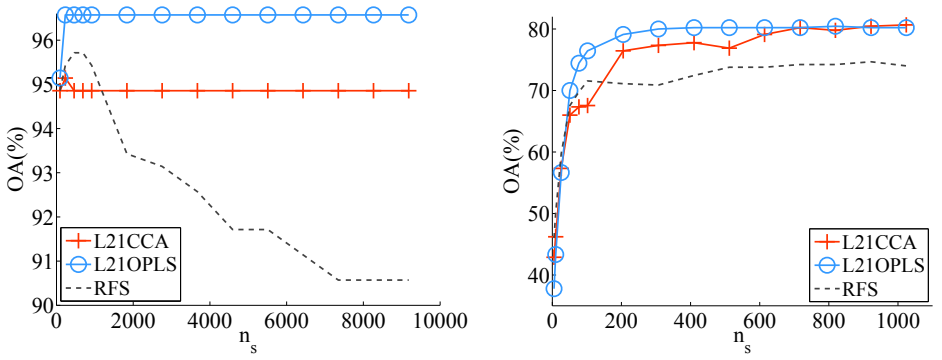
próstata, vejiga/uretra, mama, colorrectal, gastroesofágico, riñón, hígado, ovario, páncreas, adenocarcinomas de pulmón y carcinoma escamocelular de pulmón; y tienen 26, 8, 26, 23, 12, 11, 7, 27, 6, 14, 14 muestras respectivamente. En los datos originales de Su et al. (2001), cada muestra contiene 12 533 genes. En el conjunto de datos preprocesado de Yang et al. (2006), hay 174 muestras y 9182 genes[2] .

- "Yale Face Database" (*Yale*) (Cai et al., 2006)[3]: La base de datos de caras de Yale contiene 165 imágenes en escala de grises en formato GIF de 15 individuos. Hay 11 imágenes por sujeto, uno por cada expresión facial diferente o configuración: con luz centrada, con gafas, feliz, con luz izquierda, sin gafas, normal, con luz derecha, triste, con sueño, sorprendido y guiñando un ojo.

Para realizar una comparación justa entre los métodos propuestos y el algoritmo de referencia RFS, se ha seleccionado el único parámetro libre del modelo ($\gamma$) mediante un proceso de validación cruzada con 10 particiones ("10-fold CV"). El presente estudio comparativo tiene en cuenta todas las características extraídas por los métodos propuestos, pues resultaría el modo más justo de compararlos frente a RFS, que no realiza dicha extracción o proyección de variables. Por lo tanto, esta comparación consiste en evaluar las prestaciones obtenidas por el clasificador $C$-SVM a partir de, bien las variables seleccionadas por RFS, bien las características extraídas (es decir, el resultado de proyectar las variables seleccionadas con $\mathbf{U}$) con los métodos propuestos. El parámetro $C$ del clasificador no ha sido validado, pues se observó previamente que la elección de su valor no influía significativamente en las prestaciones obtenidas, excepto en el caso del L21OPLS para el problema de *Yale*, donde se seleccionó $C = 1$ por validación. Por lo tanto, para todos los casos, se ha fijado $C = 1$. Además, todos los conjuntos de datos empleados han sido centrados. Por último, es importante comentar que, debido al uso

[2]La base de datos aquí usada está disponible en `https://sites.google.com/site/feipingnie/file/NIPS2010_data.zip` (18.3 MB)

[3]La base de datos aquí usada está disponible en `http://www.cad.zju.edu.cn/home/dengcai/Data/Yale/8Train.zip` (23 KB)



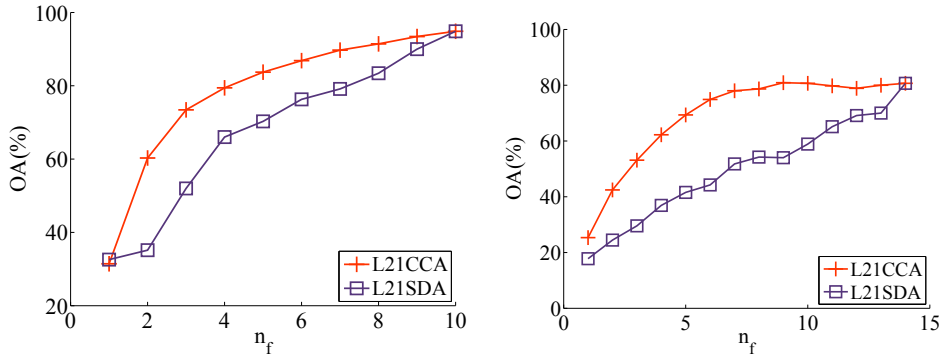

(a) *Carcinomas* (b) *Yale*

Figura 6.4: Curvas comparativas en términos de OA según el número de variables seleccionadas ($n_s$)

de todas las características extraídas, los resultados obtenidos por el método L21CCA(v1) y por L21SDA son exactamente los mismos, pues la inicialización ha sido la misma para ambos (se ha usado la inicialización de L21SDA propuesta por Shi et al. (2014), $\mathbf{W}_0 = \mathbf{C}_{\mathbf{YY}}^{\frac{1}{2}}$, con fines comparativos).

En la Figura 6.4, se muestra la evolución de la precisión total (OA) obtenida por las dos versiones de L21OPLS y L21CCA y por RFS tanto en (a) *Carcinomas* como en (b) *Yale*. Como se puede observar, la solución L21OPLS supera al resto de métodos. Las curvas de L21CCA quedan por encima de RFS en todos los casos.

Resulta interesante estudiar más detenidamente el problema *Carcinomas*. En función de las curvas L21OPLS y L21CCA, se podría concluir que toda la información relevante del problema se encuentra dentro del 2 % de las variables, que es el punto donde estos algoritmos alcanzan su máximo y quedan estancados. Por el contrario, RFS sufre de un problema grave de sobreajuste debido a la alta multicolinealidad presente es este problema. Además, necesita más variables para alcanzar su valor máximo de precisión. Una posible causa sería que entre el 2 % de las variables seleccionadas haya también variables redundantes que menguan las capacidades del clasificador. En el caso de los métodos propuestos, esto no ocurre, pues blanquean estas variables, anulando así el efecto pernicioso de la información redundante. Para corroborar esta conclusión, se han ejecutado 10 particiones distintas del conjunto de entrenamiento y se han seleccionado aquellas variables comunes dentro del 2 % de las primeras variables seleccionadas entre todas las ejecuciones, resultando solamente en un 0,5 % de las variables; el resultado obtenido por el clasificador $C$-SVM entrenado con ese 0,5 % de las variables seleccionadas es $96{,}86 \pm 3{,}42$ de precisión total, que es aproximadamente el mismo que el conseguido por el L21OPLS usando el 2 % de ellas ($96{,}29 \pm 2{,}71$).




(a) *Carcinomas* (b) *Yale*

Figura 6.5: Curvas comparativas en términos de OA según el número características extraídas entre el algoritmo L21CCA iterativo y su versión usando la solución de Procrustes (L21SDA)

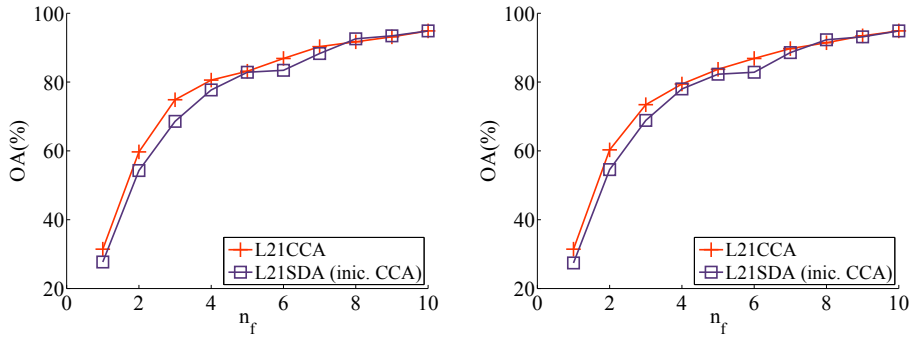
### 6.2.3. Evaluación de la solución basada en Procrustes

En este subapartado, se compara el algoritmo L21SDA propuesto por Shi et al. (2014) con el método L21CCA(v1) aquí presentado, siendo la única diferencia entre ambos algoritmos el uso del problema ortogonal de Procrustes. En el Capítulo 3, se demostró teóricamente los problemas que tiene el empleo de la aproximación de Procrustes en este tipo de esquemas iterativos y en el Capítulo 4 se confirmó empíricamente para soluciones dispersas. En este subapartado, se pretende hacer lo mismo para soluciones parsimoniosas, denunciando así el uso que se está haciendo por defecto de esta aproximación de Procrustes en esquemas MVA iterativos.

El procedimiento experimental es el mismo que en el subapartado anterior, pero las curvas ilustradas a continuación se harán en función del número de características extraídas en lugar del número de variables seleccionadas.

En la Figura 6.5, se muestra la OA obtenida usando todas las variables de los problemas (a) *Carcinomas* y (b) *Yale*. Se puede ver claramente la superioridad de L21CCA(v1) cuando se seleccionan $n'_f < n_f$ características extraídas, como se justifica en el apartado 3.3.1. Cuando se usan todas las características extraídas, los resultados son los mismos, pues el clasificador final usa toda la información proyectada.

En el Apartado 3.3.1, se demostró que, cuando se cancela el término de regularización, la única inicialización que podría ser válida para el uso de la aproximación de Procrustes en estos esquemas sería la solución del propio método MVA original a resolver. Para observar los efectos de esta inicialización en el algoritmo L21SDA, se ilustra la comparación con L21CCA(v1) para *Cacinomas* y *Yale* en las Figuras 6.6 y 6.7 respectivamente. Las subfiguras 6.6a y 6.7a muestran las prestaciones obtenidas usando el subconjunto




(a) OA cuando se usa únicamente el 5 % de las variables

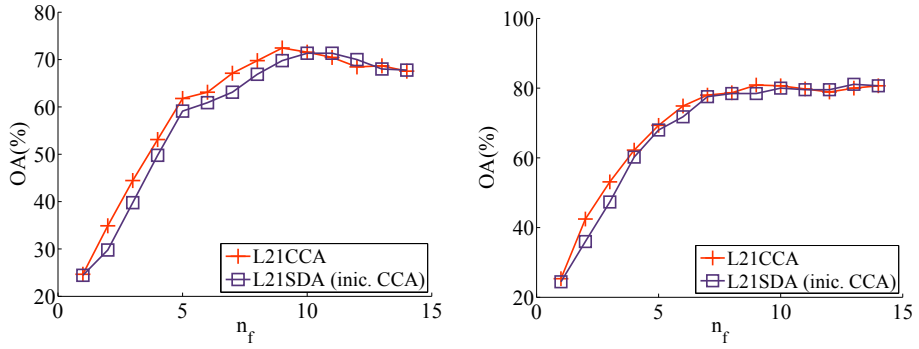
(b) OA cuando se usan todas las variables

Figura 6.6: OA para el problema *Carcinomas* cuando L21SDA ha sido inicializado con la solución del CCA (es decir, $\mathbf{W}_0 = \mathbf{W}_{\text{CCA}}$), ya que sería la única opción válida para el uso del problema ortogonal de Procrustes. Se ha observado que la inicialización del L21CCA es irrelevante.

de variables seleccionadas que producían el comienzo del estancamiento de la precisión en las curvas de la Figura 6.4. En las subfiguras 6.6b y 6.7b se usan todas las variables del problema. Aunque la diferencia es menor, aún se puede apreciar la superioridad del marco MVA con restricciones propuesto en el Capítulo 3 y reflejado en la solución L21CCA(v1) en contra de la aproximación de Procrustes.

Con el fin de confirmar estos problemas que tiene la aproximación de Procrustes para el caso, no solo del CCA, sino también del OPLS, se muestra una comparación entre el método SRRR propuesto por (Chen y Huang, 2012), que usa la aproximación de Procrustes, y el método L21OPLS aquí propuesto. En la Figura 6.8, se aprecia el mismo efecto observado en la comparación anterior entre L21SDA y L21CCA, tanto para el problema *Carcinomas* (subfigura (a)) como para *Yale* (subfigura (b)).

Por último, para completar el estudio entre los métodos propuestos aquí y los métodos existentes en la literatura, se muestra también un estudio comparativo del coste computacional de los mismos. Como se puede observar en la Figura 6.9, se puede ver rápidamente que los métodos propuestos, además de presentar mejores prestaciones cuando se selecciona un subconjunto de características, son computacionalmente más eficientes, pues requieren de menos tiempo para obtener la solución. Se puede observar que cuanto mayor sea el número de variables de entrada —como es el caso de *Carcinomas* (en la subfigura (a))—, mayor será la diferencia entre la versión eficiente del método L21OPLS y el SRRR. Además, cuanto menor es la diferencia entre el número de variables de entrada y de salida —como es el caso de *Yale* (en la subfigura (b))— la diferencia de tiempos de ejecución entre la versión




(a) OA cuando se usa únicamente el 10 % de las variables (b) OA cuando se usan todas las variables

Figura 6.7: OA para el problema *Yale* cuando L21SDA ha sido inicializado con la solución del CCA (es decir, $\mathbf{W}_0 = \mathbf{W}_{\text{CCA}}$), ya que sería la única opción válida para el uso del problema ortogonal de Procrustes. Se ha observado, de nuevo, que la inicialización del L21CCA es irrelevante.

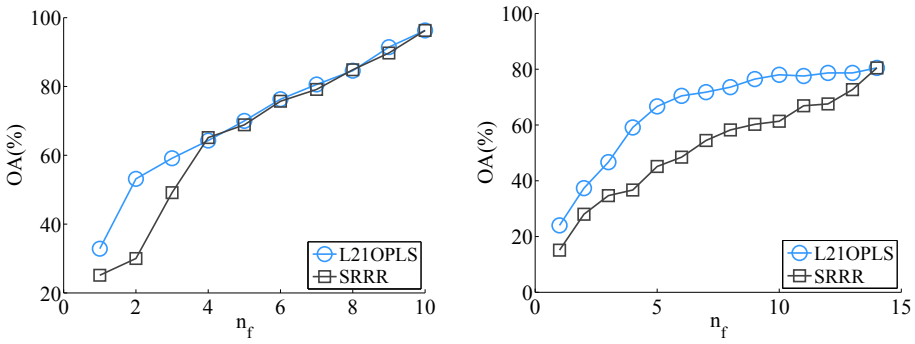
eficiente del método L21CCA y el L21SDA es mayor.

Una última conclusión adicional que se podría sacar a la vista de las Figuras 6.2, 6.4 y 6.9 sería la preferencia de usar el método OPLS en lugar del CCA, pues el OPLS no solo obtiene mejores prestaciones sino que también es computacionalmente más eficiente.

## 6.3. Conclusiones

En este capítulo, se han propuesto soluciones parsimoniosas que permiten seleccionar aquellas variables con información relevante del problema y, al mismo tiempo, lidiar con los problemas propios de las multicolinealidades mediante la extracción de características, blanqueando las variables de entrada seleccionadas. Para ello, se ha particularizado el marco MVA con restricciones propuesto en el Capítulo 3 para soluciones parsimoniosas, usando el término de regularización $\ell_{2,1}$. Además, se ha explotado la propiedad de invarianza rotacional de esta norma para converger al marco MVA generalizado introducido en el apartado 3.2, formulando así una segunda versión más eficiente de esta solución.

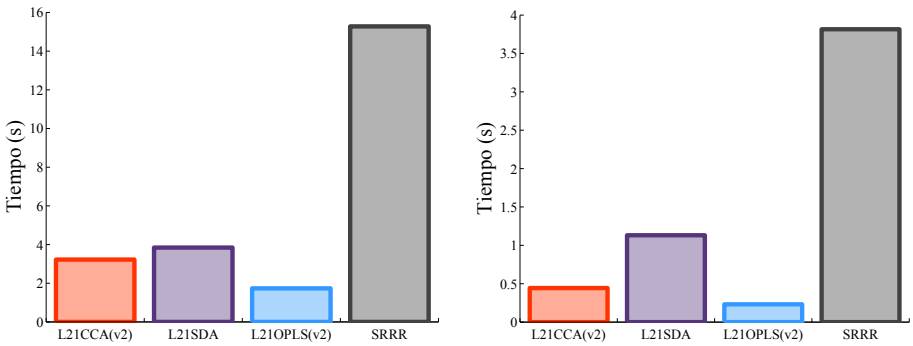
En los experimentos, se ha ilustrado la habilidad de estos métodos de tratar con conjuntos de datos que presentan problemas de multicolinealidad tanto en tareas de regresión como de clasificación, mejorando la capacidad discriminatoria del estado del arte en selección de variables. Además, se han confirmado los problemas de usar el problema ortogonal de Procrustes en esquemas MVA iterativos también para soluciones parsimoniosas. Como conclusión general de los resultados obtenidos en este capítulo, cabría decir




(a) *Carcinomas* (b) *Yale*

Figura 6.8: Curvas comparativas en términos de OA según el número características extraídas entre el algoritmo L21CCA iterativo y su versión usando la solución de Procrustes (L21SDA)

que los métodos MVA aquí propuestos, no solo son computacionalmente más eficientes, sino que pueden mejorar prestaciones cuando existen multicolinealidades entre las variables seleccionadas.




(a) *Carcinomas* (b) *Yale*

Figura 6.9: Estudio comparativo del tiempo (en segundos) que requieren los métodos propuestos (L21CCA y L21OPLS) y los existentes en la literatura (L21SDA y SRRR) para los problema (a) *Carcinomas* y (b) *Yale*.

# Capítulo 7

# MVA con restricciones de no negatividad

*Aquella teoría que no encuentre aplicación práctica en la vida, es una acrobacia del pensamiento.*

Swami Vivekananda (1863-1902)

**Resumen:** Las tareas de análisis de datos visuales o de audio tienen que tratar, por lo general, con señales no negativas y de alta dimensionalidad. Sin embargo, la mayoría de los métodos de análisis de datos sufren de sobreajuste y problemas numéricos cuando los datos tienen más de unas pocas dimensiones, necesitando un procesamiento previo de reducción de dimensionalidad. Además, la interpretabilidad en aplicaciones de audio o vídeo es una propiedad deseable, especialmente cuando se trabaja con señales espectrales o de energía, debiéndose cumplir la no negatividad en las soluciones. Debido a estas dos necesidades, en este capítulo se proponen diferentes métodos para reducir la dimensionalidad de los datos mientras se asegura la no negatividad y la interpretabilidad de la solución. En particular, se propone una metodología para diseñar bancos de filtros de una manera supervisada para aplicaciones que tratan con datos cuyos valores son no negativos. Se analiza el poder discriminatorio de las características extraídas con los métodos propuestos para dos aplicaciones diferentes y ampliamente estudiadas: la clasificación de texturas y de género musical. Además, se comparan los bancos de filtros obtenidos por los métodos propuestos con otros métodos de referencia en el estado del arte para la extracción ad hoc de características en estos ámbitos.

## 7.1. Revisión de aplicaciones con bancos de filtros

Con el objetivo de ilustrar las ventajas que poseen los métodos MVA supervisados (en particular, el OPLS), se va a considerar en este capítulo la aplicación de estos métodos sobre las dos tareas siguientes: el reconocimiento de texturas en imágenes y la clasificación de género musical. Por este motivo, el presente apartado ofrece un breve resumen de los conocimientos necesarios tanto del procesamiento de imágenes requerido para el uso del OPLS en el reconocimiento de texturas (véase el Subapartado 7.1.1) como del procesamiento del audio de las canciones para su posterior clasificación del género musical correspondiente (véase el Subapartado 7.1.2).

### 7.1.1. Clasificación de texturas

En este subapartado, se pretende revisar la aplicabilidad de los métodos basados en el algoritmo OPLS con restricciones de no negatividad en la tarea de clasificar texturas presentes en una determinada imagen.

Como punto de partida, resulta de utilidad volver a examinar la Figura 1.1 introducida al comienzo de esta tesis doctoral (en el Apartado 1.2), donde se ilustran todas las etapas encontradas habitualmente en una aplicación de reconocimiento de texturas. Siguiendo estas fases, se puede ver que la imagen en cuestión es primeramente preprocesada (fase 1) y, posteriormente, transformada al dominio de la frecuencia (fase 2) con el fin de facilitar la extracción de características relevantes en el proceso de filtrado (fase 3). Por último, se emplea un clasificador para discriminar entre todas las posibles texturas diferentes (fase 4).

Una sencilla y, a la vez, habitual etapa de pre-procesamiento en este área consiste en aplicar una transformada de Fourier rápida bi-dimensional ("2D-Fast Fourier Transform") a cada imagen por separado ($\boldsymbol{x}$, una vez vectorizado) que suele ser transformada posteriormente a escala de grises si la imagen original está en color. Esto permite a la siguiente etapa extraer características ($\bar{\boldsymbol{x}}$) en el dominio de la frecuencia mediante un banco de filtros ($\mathbf{U}$) como

$$\bar{\boldsymbol{x}} = \mathbf{U}\boldsymbol{x}.$$

Una de las técnicas de extracción de características más conocida para la clasificación de texturas es el *filtrado de Gabor*. Sin embargo, los filtros de Gabor ("Gabor Filters", GF) muestran una fuerte dependencia sobre varios parámetros cuyos valores podrían afectar significativamente a las prestaciones discriminatorias del subsiguiente clasificador. Debido a este hecho, el diseño de un banco de filtros de Gabor, consistente en la selección de un conjunto apropiado de valores para los parámetros de los filtros, es un trabajo crítico y bastante complejo (véase el Apéndice C para un mayor detalle sobre los filtros de Gabor). La manera en que el banco de filtros analiza

el dominio espacial y frecuencial depende de las posibles combinaciones de los diferentes parámetros. Debido a esto, se puede encontrar en la literatura diferentes filtros de Gabor propuestos, ajustado cada uno de ellos a una aplicación en particular. Los efectos de los parámetros de los filtros de Gabor sobre la tarea de clasificación de texturas han sido evaluados exhaustivamente por Bianconi y Fernández (2007). En la Tabla 7.1, se facilita un resumen de estos resultados, donde se muestra aquellos parámetros de los filtros que parecen ser más críticos para esta tarea en concreto, así como los conjuntos de valores que fueron evaluados. El número total de filtros en el banco viene dado por $n_f$ y $F_r$ es la relación entre frecuencias adyacentes.

Tabla 7.1: Parámetros de los filtros de Gabor y su relevancia para la tarea de clasificación de texturas según Bianconi y Fernández (2007)

| Parámetro | Valor | ¿Relevante? |
|---|---|---|
| Relación entre frecuencias ($F_r$) | $\sqrt{2}$, 2 | Sí |
| Número de frecuencias ($n_F$) | 4, 5, 6 | No |
| Número de orientaciones ($n_O$) | 4, 6, 8 | No |
| Parámetro de suavizado 1 ($\eta$) | 0,5, 1,0, 1,5 | Sí |
| Parámetro de suavizado 2 ($\gamma$) | 0,5, 1,0, 1,5 | Sí |

Una conclusión a destacar del trabajo de Bianconi y Fernández (2007) es que los parámetros de suavizado $\gamma$ y $\eta$ son parámetros importantes, mientras que el número de frecuencias y el número de orientaciones tienen, en general, poco efecto sobre la clasificación de texturas. Este resultado contradice la creencia, ampliamente aceptada, de que los parámetros que influyen en mayor medida a las prestaciones de la clasificación de texturas están relacionados con el número de orientaciones ($n_O$), el número de frecuencias ($n_F$) y la frecuencia más alta de todos los filtros.

Como se muestra en el estudio de Bianconi y Fernández (2007), el diseño de los bancos GF puede ser muy costoso computacionalmente debido al proceso de validación que se necesita para ajustar los parámetros libres. Además, la forma general de los GF se predefine a priori y, al margen de su uso generalizado en la clasificación de texturas, no existen garantías de que los GF sean la opción más adecuada para una tarea en particular. En contraste a esto, los métodos aquí propuestos usan las etiquetas disponibles para construir el banco de filtros y no asumen ninguna forma predefinida para la respuesta en frecuencia de los filtros. Por esta razón, se espera que sean capaces de extraer las características más discriminatorias para cada tarea supervisada particular.

Para terminar este subapartado, hay algunas consideraciones prácticas

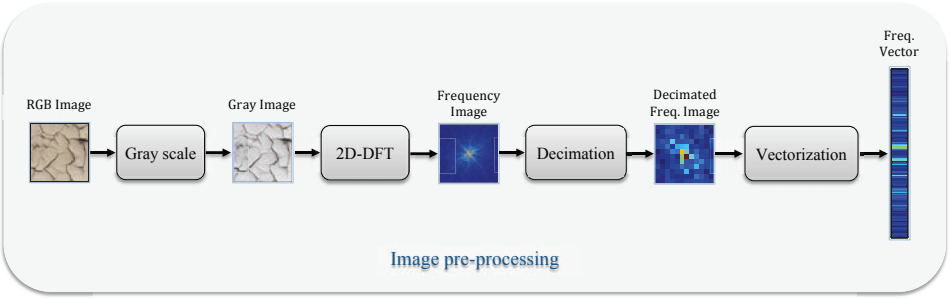



Figura 7.1: Ejemplo del esquema de pre-procesamiento aplicado a una imagen perteneciente a la clase "tierra" de la base de datos CGTextures. Los dos últimos bloques se incluyen solamente para los métodos propuestos.

que se deberían tener en cuenta para poder aclarar las diferencias entre los esquemas aquí propuestos y la aplicación directa de los GF:

- El filtrado de Gabor produce dos características por imagen filtrada: la media ($\mu$) y la desviación estándar ($\sigma$) de la imagen filtrada. Por el contrario, los métodos propuestos generan únicamente una característica por imagen filtrada (es decir, la mitad), que conceptualmente es equivalente a la media. Una posible mejora sería incluir una característica adicional en función de la desviación típica, aunque habría que modificar la formulación OPLS.
- Para facilitar la ejecución de los algoritmos propuestos, se diezma cada frecuencia de la imagen usando la energía media de cada píxel vecino de la imagen. Esto da lugar a una resolución más baja ($\rho \times \rho$) y, por tanto, a una reducción de la dimensionalidad del vector de frecuencia vectorizado a $n$ variables, siendo $n = \rho^2$. Este paso de pre-procesamiento se representa en la Figura 7.1. La primera mitad de este esquema también representa el pre-procesamiento requerido por los GF.

### 7.1.2. Clasificación de género musical

En este subapartado, se va a revisar la aplicabilidad de los esquemas basados en OPLS presentados en este capítulo para aplicaciones de reconocimiento musical. Aunque aquí se considera el caso particular de la clasificación de género musical, esta aproximación se podría extender directamente a otras tareas de recuperación de información musical ("Music Information Retrieval", MIR). Como antes, el objetivo del diseño automático del banco de filtros es el de obtener buenas tasas de reconocimiento, mientras que, al mismo tiempo, se extraen características interpretables.

La aplicación completa de reconocimiento musical se puede resumir en tres bloques bien diferenciados representados en la Figura 7.2: 1) la etapa de

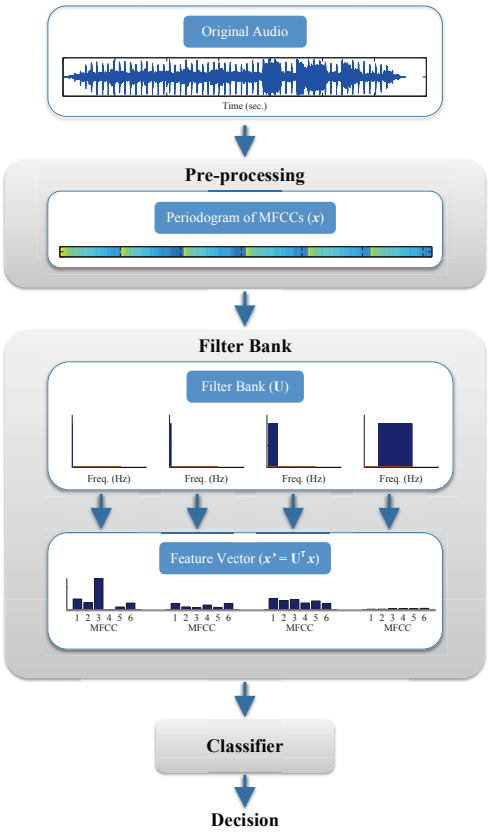



Figura 7.2: Esquema completo del proceso de clasificación de género musical a partir de una canción de audio en bruto a la decisión final. El clip de audio se procesa principalmente para obtener una representación en frecuencia que, en este caso, es un periodograma de los primeros 6 MFCC. Los periodogramas se pasan entonces a través del banco de filtros, de modo que cada característica extraída resume la energía contenida en un cierto rango de frecuencias. Por último, se realiza la clasificación en base a las características extraídas.

pre-procesamiento de audio que transforma los datos en bruto en información útil para el siguiente paso; 2) un banco de filtros que tiene como objetivo reducir la dimensionalidad de los datos y facilitar así el trabajo de la etapa ulterior; 3) y el clasificador, que toma la decisión final de reconocimiento.

La etapa de pre-procesamiento del audio, que transforma las señales de audio sin procesar en información relevante para el siguiente paso, se suele subdividir en dos etapas (véase la Figura 7.3): la extracción de características de corta duración (o "short-time feature extraction"), que consta de características extraídas en períodos que van desde 5 hasta 100 ms, donde las señales de música pueden considerarse aproximadamente estacionarias (véase, por ejemplo Aucouturier et al., 2005); y la integración de caracte-

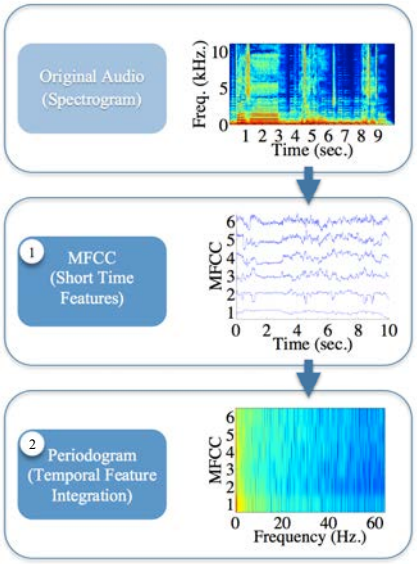



Figura 7.3: Esquema del pre-procesamiento de un fragmento de diez segundos de la canción "Follow The Sun" de "Xavier Rudd"

rísticas temporales (o "temporal feature integration"), que es el proceso de combinar todos los vectores de características pertenecientes a un mismo rango temporal en un único vector de características, con el fin de capturar la información temporal relevante de la trama de audio.

A continuación, se detallan estas dos etapas:

1. **Características de corta duración**: Como representación de las características de corta duración, se ha seleccionado los coeficientes "Mel Frequency Cepstral Coefficients" (MFCC) (Meng et al., 2006; Pampalk, 2006) debido a su uso generalizado y gran éxito en varios campos de la MIR (McKinney y Breebaart, 2003; Mandel et al., 2006). Los MFCC se clasifican por orden decreciente de riqueza de representación de la envolvente espectral. Por lo tanto, los MFCC inferiores contienen información sobre las variaciones lentas en la envolvente espectral. El primer coeficiente, por ejemplo, está correlado con la dimensión perceptual de la intensidad. En los experimentos, se utilizan únicamente los 6 primeros MFCC y, a fin de minimizar el "aliasing" en el MFCC, se aplica un tamaño de trama de 30 ms y un tamaño de salto de 7,5 ms. Cada fragmento de música se normaliza en energía antes de la etapa de extracción MFCC.

2. **Integración de características temporales**: Con el fin de capturar la información temporal relevante de la trama, primero se estima el espectro de potencia de cada MFCC utilizando un periodograma, como

sugieren McKinney y Breebaart (2003). Posteriormente, se concatenan estas seis características de energía en un único vector de características. Existen muchos otros métodos de integración de características temporales (véase Meng et al., 2007, para una buena revisión de estas técnicas).

Una vez los datos en bruto se han convertido en una representación no negativa (es decir, los periodogramas de los MFCC, $\mathbf{X}$), el siguiente paso se basa en aplicar un banco de filtros, $\mathbf{U}$, con el fin de extraer las características no negativas deseadas,

$$\bar{\mathbf{X}} = \mathbf{U}^\top \mathbf{X},$$

que se puede ver como la energía contenida en ciertas bandas de frecuencias de cada periodograma de los MFCC. Téngase también en cuenta que, para conservar una interpretación de energía en las proyecciones, este banco de filtros $\mathbf{U}$ debe contener necesariamente coeficientes no-negativos, ya que se aplica directamente sobre el espectro de potencia estimado (periodograma), $\bar{\boldsymbol{x}}_i = \mathbf{U}^\top \boldsymbol{x}_i$, donde $\boldsymbol{x}_i$ es el periodograma del $i$-ésimo MFCC y $\bar{\boldsymbol{x}}_i$ es el correspondiente $i$-ésimo vector de características que tiene tantas componentes como el número de filtros en el banco. Estos vectores de características se introducirán, finalmente, en el subsiguiente clasificador.

Con el fin de diseñar el banco de filtros ($\mathbf{U}$), existen dos alternativas diferentes: utilizar conocimiento experto, siendo esta aproximación la más usada habitualmente a pesar de no estar adaptada a la tarea de reconocimiento; y los esquemas supervisados que se proponen en este capítulo y que usan la información de las etiquetas, permitiendo así el diseño ad hoc de los bancos de filtros para cada tarea de reconocimiento (véase el Apartado 7.2 donde se proponen diferentes soluciones supervisadas). Un ejemplo de la primera alternativa es el banco predefinido de filtros "Philips" usado por McKinney y Breebaart (2003), donde los autores sugieren resumir las componentes de potencia en cuatro bandas de frecuencia: 1) 0 Hz (componente DC); 2) 0–2 Hz (ritmo); 3) 3–15 Hz (modulación de la energía, por ejemplo, el vibrato); y 4) $>$ 20 Hz (asociado a la "rugosidad" percibida). Por lo tanto, para este banco de filtros particular, $\mathbf{U}$ es una matriz de tamaño $D \times 4$, donde $D = \frac{f_s}{2} + 1$ es el número de puntos del periodograma y $f_s$ es la longitud de las series MFCC usadas para calcular el periodograma (medida en número de muestras). En este capítulo, se utilizará $f_s = 256$. En el subapartado de experimentos 7.3.2, se compararán las soluciones propuestas con este banco de filtros fijado a fin de evaluar el poder discriminatorio de las soluciones supervisadas.

## 7.2. Diseño supervisado de filtros con técnicas MVA

En este apartado, se formulan distintos métodos con el fin de diseñar bancos de filtros en un escenario de aprendizaje supervisado, donde el objetivo consiste en aprender características relevantes de los datos de entrada usando un conjunto de $N$ datos de entrenamiento $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, para $i = 1, \ldots, N$, siendo $\boldsymbol{x}_i \in \mathbb{R}^{n\times 1}$ y $\boldsymbol{y}_i \in \mathbb{R}^{m\times 1}$ los vectores de los datos de entrada y de etiquetas respectivamente. En este capítulo, se supone que todas las entradas de $\boldsymbol{x}_i$ son no negativas, siendo el caso de aquellas aplicaciones donde el espacio de entrada consta de características espectrales.

Cuando los datos de entrada son características espectrales (es decir, no negativas), se puede considerar $\mathbf{U}$ como un banco de filtros de frecuencia, siempre que las entradas de $\mathbf{U}$ sean forzadas para que sean no negativos y $\bar{\boldsymbol{x}}_i$ se puede interpretar como la salida no negativa de cada uno de los filtros del banco. Sin embargo, cuando se centra la matriz $\mathbf{X}$, también se puede ver a $\bar{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$ como las proyecciones de los datos de entrada centrados, aunque ya no se garantice que sus entradas sean no negativas. No obstante, el centrado de los datos no afecta al diseño del banco de filtros y es recomendable para fines de aprendizaje automático si algún proceso de regresión se ve implicado en el esquema global (Shawe-Taylor y Cristianini, 2004). De hecho, resulta bastante sencillo mostrar la irrelevancia de la operación de centrado con respecto a la interpretabilidad de las características extraídas, ya que

$$\mathbf{U}^\top \tilde{\boldsymbol{x}}_i = \mathbf{U}^\top \boldsymbol{x}_i - \mathbf{U}^\top \boldsymbol{\mu}_x = \bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{\bar{x}}, \tag{7.1}$$

donde $\boldsymbol{\mu}_{\bar{x}}$ es la media de los datos filtrados. Por lo tanto, la interpretación del banco de filtros sigue siendo válida cuando se trabaja con datos centrados y, además, los problemas de optimización resultan numéricamente más estables.

Se va a utilizar el OPLS como punto de partida para diseñar el banco de filtros. OPLS, como ya se ha visto, es óptimo en el sentido de MSE. Para forzar que los coeficientes del filtro sean no negativos, se va a añadir una restricción de no negatividad a la función de coste del OPLS. Así pues, el problema de minimización que se propone para el diseño del banco de filtros es el siguiente:

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{W}}{\text{mín}} & \|\mathbf{Y} - \mathbf{W}\mathbf{U}^\top \mathbf{X}\|_F^2 \\ \text{sujeto a} & \mathbf{U} \geq \mathbf{0} \end{array} \tag{7.2}$$

donde $\mathbf{U} \geq \mathbf{0}$ indica que todos los elementos de la matriz $\mathbf{U}$ han de ser no negativos.

Para resolver este problema (7.2), en este apartado se van a proponer cuatro algoritmos distintos:

1. OPLS no negativo ("Non-negative OPLS", NOPLS): Basado en alternar dos problemas convexos acoplados (es decir, calculando $\mathbf{U}$ y $\mathbf{W}$ iterativamente).

2. NOPLS con la aproximación de Procrustes (P-NOPLS): Parecido al NOPLS, pero calculando $\mathbf{W}$ mediante el problema ortogonal de Procrustes ("Orthogonal Procrustes problem" estudiado por Schönemann, 1966).

3. NOPLS deflactado (defNOPLS): Implementación secuencial del NOPLS usando deflacción.

4. OPLS al estilo NMF o "NMF-like OPLS" (NMF-OPLS): Se puede considerar como una versión supervisada del problema NMF.

Nótese que, aunque todos los algoritmos intentan resolver el mismo problema de optimización, en general convergerán a distintas soluciones. En los siguientes subapartados, se derivarán estos algoritmos y sus resultados serán comparados en el apartado experimental. Para completar este estudio, también se va a considerar un quinto método conocido como "Positive constrained OPLS" (POPLS) propuesto por Arenas-García et al. (2006) y que permite resolver (7.2) con programación cuadrática ("Quadratic Programming", QP).

### 7.2.1. OPLS no negativo

El algoritmo propuesto en este subapartado parte del marco general para métodos MVA con restricciones propuesto en el Capítulo 3. Del mismo modo en que el Capítulo 4 incluye el término de regularización $\ell_1$ para forzar dispersión en la solución sobre este marco MVA generalizado, el presente algoritmo reemplaza esta restricción de dispersión por aquella que fuerza soluciones no negativas.

Por lo tanto, siguiendo los mismo argumentos que en el Capítulo 3, aquí se propone el siguiente procedimiento iterativo para resolver la función objetivo (7.2):

1) Paso$-\mathbf{W}$: Fijado $\mathbf{U}$, minimizar (7.2) con respecto a $\mathbf{W}$, sujeto a $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$.

   La solución de este problema viene dada por el problema de autovalores estándar

   $$\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}}^\top \mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} \mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \tag{7.3}$$

   donde $\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} = \mathbf{U}^\top \mathbf{C}_{\mathbf{X}\mathbf{Y}}$. Nótese que la dimensión de la matriz que necesita ser analizada es $m$, convirtiéndolo en un paso más eficiente en el caso común en que $m < n$.

2) Paso$-\mathbf{U}$: Fijado $\mathbf{W}$, minimizar (7.2) con respecto a $\mathbf{U}$ solamente.

   Remítase el lector a (Van Benthem y Keenan, 2004; Kim y Park, 2008) como buenos resúmenes sobre métodos de optimización que resuelven el problema de mínimos cuadrados no negativos ("Non-Negative

Least Squares", NNLS). En el apartado de experimentos, se usará la implementación en MATLAB de un algoritmo conocido como "block principal pivoting algorithm" facilitado por Kim y Park (2008)[1]. En caso de querer añadir también un término de regularización $\ell_1$ a (7.2) (es decir, $\lambda||\mathbf{U}||_1$), se podría usar el algoritmo "Monotone Incremental Forward Stagewise Regression" (MIFSR) propuesto por Hastie et al. (2007) con las modificaciones introducidas por Sigg et al. (2007).

El método NOPLS, por tanto, consiste en aplicar estos dos Pasos, $-\mathbf{W}$ y $-\mathbf{U}$, de manera iterativa hasta que se cumpla algún criterio de convergencia. Se ha visto en experimentos preliminares que la inicialización del algoritmo no es crítica, inicializándose $\mathbf{U}$ en la primera iteración simplemente con la matriz identidad. Como mecanismo de parada, se usa $\mathrm{Tr}\{\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k-1)}\} \leq \delta$, donde el superíndice indica la $k$-ésima iteración y $\delta$ es una pequeña constante. En pocas palabras: el algoritmo se detiene cuando la diferencia entre los autovalores del Paso$-\mathbf{W}$ entre dos iteraciones consecutivas es menor que una pequeña constante prefijada $\delta$.

### 7.2.2. NOPLS con la aproximación de Procrustes

Este segundo método propuesto consiste en la modificación del Paso$-\mathbf{W}$ de NOPLS aplicando la solución del problema ortogonal de Procrustes diseccionada cuidadosamente en el Apartado 3.3.1. Esta aproximación ha sido usada, por ejemplo, por Zou et al. (2006) y van Gerven et al. (2012) para obtener soluciones dispersas de PCA y OPLS respectivamente. Como se comentó anteriormente, fijando la matriz de proyección $\mathbf{U}$ obtenida en el Paso$-\mathbf{U}$ del subapartado anterior, el Paso$-\mathbf{W}$ del algoritmo es:

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \|\mathbf{Y} - \mathbf{W}\bar{\mathbf{X}}\|_F^2 \\ \text{sujeto a} & \mathbf{W}^\top\mathbf{W} = \mathbf{I}. \end{array} \tag{7.4}$$

Schönemann (1966), denominó a este problema como "Orthogonal Procrustes problem" y definió su solución como

$$\mathbf{W}_{\text{PROCRUSTES}} = \mathbf{QP}^\top, \tag{7.5}$$

dada la descomposición en valores singulares $\mathbf{C}_{\bar{\mathbf{X}}\mathbf{Y}} = \mathbf{PDQ}^\top$. Puesto que la solución de (7.3) es $\mathbf{W}_{\text{NOPLS}} = \mathbf{Q}$, se puede ver que P-NOPLS consiste simplemente en una versión rotada de esta matriz durante el Paso$-\mathbf{W}$. Sin embargo, téngase en cuenta que:

- El proceso de rotación afecta a la relevancia y el ordenamiento de las características extraídas. Para la formulación NOPLS propuesta en el

[1] El código está disponible en `http://www.cc.gatech.edu/~hpark/software/nmf_bpas.zip`

subapartado anterior, se puede afirmar que las características (o bancos de filtros) se clasifican en función de su relevancia. Esto es: el primer filtro captura el máximo de información posible con un único filtro con respecto al criterio (7.2) y así sucesivamente. El proceso de rotación impide afirmar esto mismo para P-NOPLS.

- En el Subapartado 3.3.1.2, se demostró que la aproximación basada en Procrustes es muy sensible a la inicialización y que, para algunas inicializaciones, el algoritmo podría no progresar en absoluto.

Los dos argumentos anteriores justifican la preferencia por NOPLS sobre la solución P-NOPLS. Sin embargo, P-NOPLS también ha sido incluido en los experimentos en aras del ulterior estudio comparativo.

### 7.2.3. Implementación secuencial de NOPLS usando deflacción

En este subapartado, se describe un algoritmo secuencial que implementa el esquema OPLS no negativo introducido en el Subapartado 7.2.1. Este algoritmo secuencial consta de los dos siguientes pasos: 1) la extracción del vector de proyección $\boldsymbol{u}_j$, que representa la respuesta en frecuencia del siguiente filtro a incluir en el banco; y 2) la aplicación de un proceso de deflacción para eliminar la influencia del $j$-ésimo autovector mediante la cancelación del autovalor asociado. Estos pasos se repiten para $j = 1, \ldots, n_f$ hasta que se alcanza el número de filtros o características deseado.

El diseño del siguiente filtro consiste en la extracción de un par de vectores $\{\boldsymbol{u}_j, \boldsymbol{w}_j\}$ que son óptimos con respecto a (7.2). Esto se puede hacer mediante la iteración de los Pasos$-\mathbf{W}$ y $-\mathbf{U}$ descritos para el algoritmo NOPLS. Dado que en este caso se está resolviendo un problema unidimensional en cada paso, la solución del Paso$-\mathbf{W}$ se simplifica a

$$\boldsymbol{w}_j = \frac{\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}^{\top}}{\|\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}}\|}, \tag{7.6}$$

donde $\mathbf{C}_{\bar{\boldsymbol{x}}\mathbf{Y}} = \boldsymbol{u}_j^{\top}\mathbf{C}_{\mathbf{XY}}$.

Puesto que el vector $\boldsymbol{u}_j$ es una solución a un problema con la restricción de no negatividad impuesta, ya no se puede admitir que $\boldsymbol{u}_j$ disfruta de las cualidades propias de un autovector y, por ello, se considera que es un pseudo-autovector (véase el Subapartado 2.1.4). Debido a esto, ha de aplicarse una técnica de deflacción con la habilidad de eliminar la influencia de este tipo de soluciones. Por este motivo, se recurre nuevamente a la deflacción por complemento de Schur:

$$\mathbf{C}_{\mathbf{XY}} \leftarrow \mathbf{C}_{\mathbf{XY}}\left(\mathbf{I} - \frac{\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_j\boldsymbol{u}_j^{\top}\mathbf{C}_{\mathbf{XY}}}{\boldsymbol{u}_j^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\boldsymbol{u}_j}\right), \tag{7.7}$$

descrita también en (4.5) (o (4.4), si se reescribe únicamente en términos del autovector $\boldsymbol{w}_i$).

La Tabla 7.2 facilita el pseudocódigo del algoritmo secuencial que se acaba de describir. Téngase en cuenta que, en esta tabla, el subíndice $j$ se utiliza para indexar los vectores de proyección (es decir, $j = 1, \ldots, n_f$), mientras que el superíndice $k$ indexa la aplicación iterativa de los Pasos$-\mathbf{W}$ y $-\mathbf{U}$ necesarios para converger a cada vector de proyección. En el Paso 2.2.3 del algoritmo, se pueden utilizar distintos criterios de convergencia. Uno de ellos, usado en el apartado de experimentos, consiste en controlar la distancia coseno

$$d_{\cos}\left(\boldsymbol{u}_j^{(k)}, \boldsymbol{u}_j^{(k-1)}\right) = \frac{\boldsymbol{u}_j^{(k)\top}\boldsymbol{u}_j^{(k-1)}}{\|\boldsymbol{u}_j^{(k)}\|\|\boldsymbol{u}_j^{(k-1)}\|}, \tag{7.8}$$

siendo el criterio de parada $d_{\cos}\left(\boldsymbol{u}_j^{(k)}, \boldsymbol{u}_j^{(k-1)}\right) > 1 - \delta$, donde $\delta$ es un parámetro de tolerancia. Otras posibilidades consistirían en observar la distancia coseno entre los coeficientes de regresión o entre los autovalores del Paso$-\mathbf{W}$.

Tabla 7.2: Pseudocódigo del algoritmo NOPLS secuencial usando deflacción

1.- Entradas: matrices centradas $\mathbf{X}$ e $\mathbf{Y}$, $n_f$.

2.- Para $j = 1, \ldots, n_f$

  2.1.- Inicializar $\boldsymbol{u}_j^{(0)} = \mathbf{1}. * \delta_j$ ‡.

  2.2.- Para $k = 1, 2, \ldots$

    2.2.1.- Actualizar $\boldsymbol{w}_j^{(k)}$ usando (7.6).

    2.2.2.- Actualizar $\boldsymbol{u}_j^{(k)}$ resolviendo la version unidimensional del problema NNLS (7.2) sujeto a $\boldsymbol{u}_j^{(k)} \geq 0$.

    2.2.3.- Si se cumple el criterio de convergencia, los valores actuales de salida serían $\{\boldsymbol{u}_j, \boldsymbol{w}_j\}$, en caso contrario volver a 2.2.

  2.3.- Deflactar la matriz de covarianza cruzada usando (7.7).

3.- Salida: $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_f}]$.

‡ El vector de proyección $\boldsymbol{u}_j$ se inicializa como un vector con su $j$-ésima componente igual a 1 y todas las demás componentes igual a 0.

### 7.2.4. OPLS con una formulación tipo NMF

En este subapartado, se resuelve el problema (7.2) utilizando un enfoque NMF ("Non-negative Matrix Factorization"), en particular, se recurre a la regla de actualización multiplicativa ("Multiplicative Updating rule", MU) propuesta por Seung y Lee (2001) que es, quizá, el algoritmo NMF más

conocido debido a su sencillez. Además, la función de coste del algoritmo "Projected-NMF" propuesto por Yuan y Oja (2005) y algunas relaciones entre varias versiones expuestas por Choi (2008) pueden ser de utilidad para apreciar las similitudes entre NMF y la versión supervisada que se propone aquí.

A diferencia de los algoritmos anteriores, los métodos NMF requieren valores no negativos tanto para $\mathbf{X}$ como para $\mathbf{Y}$ (es decir, $\mathbf{X} \geq \mathbf{0}$ e $\mathbf{Y} \geq \mathbf{0}$) y, por consiguiente, se debería considerar la restricción adicional $\mathbf{W} \geq \mathbf{0}$. Puesto que ciertos datos tendrán valores negativos tras la operación de centrado, se puede añadir un valor constante (por ejemplo, el valor mínimo de los datos de entrada) a todo el conjunto de datos con el fin de forzar esta no negatividad necesaria.

La función de coste a minimizar también está dada por (7.2), aunque en este caso se añade también la restricción $\mathbf{W} \geq \mathbf{0}$:

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{W}}{\text{mín}} & \|\mathbf{Y} - \mathbf{W}\mathbf{U}^\top\mathbf{X}\|_F^2 \\ \text{sujeto a} & \mathbf{U} \geq \mathbf{0}, \\ & \mathbf{W} \geq \mathbf{0}. \end{array} \tag{7.9}$$

Con el fin de facilitar la derivación de la actual propuesta, se puede reescribir la función de coste de (7.2) en términos del operador traza ($||\mathbf{A}||_F^2 = \mathrm{Tr}\{\mathbf{A}\mathbf{A}^\top\}$):

$$\mathcal{L}(\mathbf{W},\mathbf{U}) = \mathrm{Tr}\{\mathbf{C_{YY}}\} - 2\,\mathrm{Tr}\{\mathbf{W}^\top\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}\} + \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U}\mathbf{W}^\top\mathbf{W}\}. \tag{7.10}$$

Como resumen de la regla de MU, supóngase que el gradiente de (7.10) con respecto a $\mathbf{U}$ o $\mathbf{W}$ se puede descomponer como

$$\partial\mathcal{L} = \partial\mathcal{L}^+ - \partial\mathcal{L}^-,$$

donde $\partial\mathcal{L}^+ \geq 0$ y $\partial\mathcal{L}^- \geq 0$. Entonces, la regla de actualización elemento a elemento sigue como (Choi, 2008):

$$\Psi \leftarrow \Psi \circ \frac{\partial\mathcal{L}^-}{\partial\mathcal{L}^+}, \tag{7.11}$$

donde $\circ$ indica el producto de Hadamard (es decir, elemento a elemento), $\frac{\mathbf{A}}{\mathbf{B}}$ representa la división elemento a elemento, es decir, $\left[\frac{\mathbf{A}}{\mathbf{B}}\right]_{ij} = \frac{A_{ij}}{B_{ij}}$ (para la $i$-ésima fila y la $j$-ésima columna) y $\Psi$ es la matriz que necesita ser actualizada. Nótese que esta actualización mantiene la no negatividad de la solución $\Psi$ en cada paso.

Para aplicar la regla MU en este caso, hay que obtener las primeras derivadas de (7.10) con respecto a $\mathbf{U}$

$$\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{W})}{\partial\mathbf{U}} = -2\mathbf{C_{XY}}\mathbf{W} + 2\mathbf{C_{XX}}\mathbf{U}\mathbf{W}^\top\mathbf{W},$$

que, considerando que todas las matrices implicadas son no negativas, permiten identificar

$$\partial\mathcal{L}^{+}_{\mathbf{U}} = \mathbf{C_{XX}UW}^{\top}\mathbf{W}, \qquad \partial\mathcal{L}^{-}_{\mathbf{U}} = \mathbf{C_{XY}W}.$$

De manera similar, las primeras derivadas de (7.10) con respecto a $\mathbf{W}$ son

$$\frac{\partial\mathcal{L}(\mathbf{U},\mathbf{W})}{\partial\mathbf{W}} = -2\mathbf{C}^{\top}_{\mathbf{XY}}\mathbf{U} + 2\mathbf{WU}^{\top}\mathbf{C_{XX}U},$$

de modo que se puede reconocer

$$\partial\mathcal{L}^{+}_{\mathbf{W}} = \mathbf{WU}^{\top}\mathbf{C_{XX}U}, \qquad \partial\mathcal{L}^{-}_{\mathbf{W}} = \mathbf{C}^{\top}_{\mathbf{XY}}\mathbf{U}.$$

Por lo tanto, a partir de la ecuación (7.11), las actualizaciones MU de $\mathbf{U}$ y $\mathbf{W}$ que constituyen el grueso del método NMF-OPLS están dadas por

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{\mathbf{C}^{\top}_{\mathbf{XY}}\mathbf{U}}{\mathbf{WU}^{\top}\mathbf{C_{XX}U}}, \qquad \mathbf{U} \leftarrow \mathbf{U} \circ \frac{\mathbf{C_{XY}W}}{\mathbf{C_{XX}UW}^{\top}\mathbf{W}}.$$

Como es de esperar en los algoritmos NMF, se ha visto en experimentos preliminares que la inicialización del algoritmo es crítica. El método NNDSVD ("Non-Negative Double Singular Value Decomposition")[2] propuesto por Boutsidis y Gallopoulos (2008) ofrece un buen punto de partida para los algoritmos NMF, así que se ha aplicado sobre la matriz $\mathbf{C_{XY}}$, ya que se está trabajando con un esquema supervisado. De este modo, se pueden inicializar las matrices $\mathbf{U}$ y $\mathbf{W}$ como las matrices izquierda y derecha respectivamente de la descomposición aproximada por NNDSVD: $\mathbf{C_{XY}} \sim \mathbf{UW}^{\top}$.

Las restricciones de no negatividad generalmente producen un gran número de ceros en la matriz solución, causando a menudo problemas numéricos que hacen que la actualización MU se estanque antes de lo deseado. Gillis y Glineur (2012) demostraron que se puede conseguir una pequeña mejora sustituyendo los ceros por una constante pequeña tendiendo a cero (por ejemplo, $\epsilon = 10^{-16}$). De este modo, las actualizaciones MU levemente mejoradas estarían dadas por

$$\mathbf{W} \leftarrow \text{máx}\left(\epsilon, \mathbf{W} \circ \frac{\mathbf{C}^{\top}_{\mathbf{XY}}\mathbf{U}}{\mathbf{WU}^{\top}\mathbf{C_{XX}U}}\right), \tag{7.12}$$

$$\mathbf{U} \leftarrow \text{máx}\left(\epsilon, \mathbf{U} \circ \frac{\mathbf{C_{XY}W}}{\mathbf{C_{XX}UW}^{\top}\mathbf{W}}\right). \tag{7.13}$$

Además, en los algoritmos NMF, es habitual incluir un paso de normalización en cada iteración de actualización MU con el fin de facilitar la

[2]Se ha usado la versión NNDSVDa del algoritmo de Boutsidis y Gallopoulos (2008), así como la implementación proporcionada por sus autores.

convergencia. En este caso, también se va a aplicar este paso, normalizándose las matrices $\mathbf{U}$ y $\mathbf{W}$ con su respectiva norma de Frobenius.

En la Tabla 7.3 se facilita el pseudocódigo del algoritmo NMF-OPLS que se acaba de describir. En el Paso 2.2.4 del algoritmo, se pueden utilizar distintos criterios de convergencia. En este caso, se ha usado en el apartado de experimentos $||\mathbf{U}^{(k)} - \mathbf{U}^{(k-1)}||_F \leq \delta$ como mecanismo de parada, donde los superíndices indexan la iteración y $\delta$ es una pequeña constante. Entonces, el algoritmo se detiene cuando las soluciones obtenidas en dos iteraciones consecutivas difieren menos de un pequeño umbral.

Tabla 7.3: Pseudocódigo del algoritmo NMF-OPLS

1.- Entradas: matrices positivas $\mathbf{X}$ y $\mathbf{Y}$.
- 2.1.- Inicializar $\mathbf{W}^{(0)}$ y $\mathbf{U}^{(0)}$ con el algoritmo NNDSVD.
- 2.2.- Para $k = 1, 2, \ldots$
  - 2.2.1.- Actualizar $\mathbf{W}^{(k)}$ usando (7.12).
  - 2.2.2.- Actualizar $\mathbf{U}^{(k)}$ usando (7.13).
  - 2.2.3.- Normalizar $\mathbf{W}^{(k)}$ y $\mathbf{U}^{(k)}$.
  - 2.2.4.- Si se cumple el criterio de convergencia, ir a 3.

3.- Salidas: $\mathbf{U}$, $\mathbf{W}$.

Para terminar, ténganse en cuenta algunas consideraciones con respecto al algoritmo que se acaba de describir: 1) la principal ventaja de la actualización MU es su simplicidad y facilidad de implementación; sin embargo, suele conllevar una convergencia lenta, como observaron Kim y Park (2008); 2) la aplicación de NMF-OPLS requiere que se seleccione *a priori* el número de filtros del banco ($n_f$) y no sería factible una implementación secuencial del mismo, ya que la operación de sustracción requerido por la deflación violaría la restricción de no negatividad; y 3) la implementación basada en NMF, a diferencia de NOPLS, no garantiza ni que los filtros del banco (es decir, las columnas de $\mathbf{U}$) estén ordenados por relevancia ni la ortogonalidad de las características extraídas.

### 7.2.5. OPLS con restricciones de positividad

Por completitud, en este subapartado se describe el algoritmo propuesto por Arenas-García et al. (2006) para resolver (7.2). En este caso, la matriz $\mathbf{W}$ no se calcula de forma explícita, ya que el truco aquí es expresar $\mathbf{W}$ en función de $\mathbf{U}$ e introducirla en (7.2) para obtener un problema de optimización únicamente en función de $\mathbf{U}$.

La matriz de regresión óptima se puede calcular minimizando (7.2) con

respecto a $\mathbf{W}$ solamente, siendo la solución: $\mathbf{W} = \mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\left(\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}\right)^{-1}$. Introduciendo este resultado en (7.2), se puede reescribir la función de coste objetivo en función de $\mathbf{U}$ solamente como

$$\begin{aligned}\mathcal{L}(\mathbf{U}) &= \|\mathbf{Y} - \mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\left(\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}\right)^{-1}\mathbf{U}^{\top}\mathbf{X}\|_F^2 \\ &= \mathrm{Tr}\{\mathbf{C}_{\mathbf{YY}}\} - \mathrm{Tr}\{\left(\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}\right)^{-1}\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\}.\end{aligned}$$

De este modo, se llega al siguiente problema de optimización

$$\begin{array}{ll}\underset{\mathbf{U}}{\text{máx}} & \mathrm{Tr}\{\left(\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U}\right)^{-1}\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{XY}}^{\top}\mathbf{U}\} \\ \text{sujeto a} & \mathbf{U} \geq 0, \\ & \mathbf{U}^{\top}\mathbf{U} = \mathbf{I},\end{array} \tag{7.14}$$

donde se ha incluido esta última restricción para obtener una de las infinitas posibles soluciones de la función de coste (7.14). Nótese que esta restricción es diferente de la usada por el problema OPLS. Sin embargo, Arenas-García et al. (2006) prefieren esta restricción, ya que puede ser incorporada directamente en la representación hiperesférica de los vectores de proyección, donde cada $\boldsymbol{u}_j$ está representado por un radio $r_j$ y $n-1$ ángulos $\theta_j^{(s)}$, $s = 1, \ldots, n-1$. De esta manera, la optimización se puede resolver con respecto a $\theta_j^{(s)}$ para $r_j = 1$ y las restricciones $0 \leq \theta_j^{(s)} \leq \frac{\pi}{2}$ garantizan la no negatividad de la solución. Esta aproximación fue llevada acabo por Arenas-García et al. (2006) para resolver los problemas de convergencia de la función *fmincon* de Matlab con la implementación directa de (7.14).

Un inconveniente de este método es que la propiedad deseada del OPLS, $\mathbf{U}^{\top}\mathbf{C}_{\mathbf{XX}}\mathbf{U} = \mathbf{I}$, no se cumple, provocando que los filtros no estén ordenados de acuerdo a su poder discriminatorio. Para corregir esto, se ha aplicado una implementación secuencial usando la deflacción por complemento de Schur del mismo modo que se ha hecho en los subapartados anteriores. El algoritmo POPLS secuencial resultante se resume en la Tabla 7.4.

## 7.3. Experimentos

En este apartado, se analizan las prestaciones de todos los bancos de filtros supervisados propuestos en dos tareas de clasificación: el reconocimiento de texturas en imágenes y la clasificación de género musical. Con el fin de evaluar las propuestas, se analiza su poder discriminatorio y su interpretabilidad en comparación con los bancos de filtros, ampliamente estudiados, de Gabor y Philips que están diseñados ad hoc para las aplicaciones aquí consideradas.

Tabla 7.4: Pseudocódigo del algoritmo POPLS con deflacción

1.- Entradas: matrices centradas $\mathbf{X}$ e $\mathbf{Y}$.

2.- Para $j = 1, \ldots, n_f$

2.1.- Actualizar $\boldsymbol{u}_j$ resolviendo la versión unidimensional de (7.14), es decir,

$$\max_{\boldsymbol{u}_j} \frac{\boldsymbol{u}_j^\top \mathbf{C}_{\mathbf{XY}} \mathbf{C}_{\mathbf{XY}}^\top \boldsymbol{u}_j}{\boldsymbol{u}_j^\top \mathbf{C}_{\mathbf{XX}} \boldsymbol{u}_j},$$

sujeto a $\boldsymbol{u}_j \geq 0$ y $||\boldsymbol{u}_j|| = 1$.

2.2.- Deflactar la matriz de covarianza cruzada usando (7.7).

3.- Salida: $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_f}]$.

### 7.3.1. Experimento 1: Clasificación de texturas

En este subapartado, se consideran dos tareas de clasificación de texturas diferentes: una clasificación basada en un conjunto predefinido de categorías, que es un escenario más realista para la clasificación de texturas; y la tarea de detección de la imagen original, que es una tarea utilizada habitualmente en la literatura.

La primera tarea considera un escenario real para la clasificación de texturas, donde cada imagen pertenece a una clase específica de texturas[3] entre 10 categorías diferentes: corteza ("bark"), tierra ("earth"), grava ("gravel"), madera contrachapada ("plywood"), nieve ("snow"), ladrillo ("brick"), hierba ("grass"), hiedra ("ivy"), cielo ("sky") y agua ("water"). A fin de proporcionar más muestras a la base de datos, cada imagen se divide en un conjunto de 16 sub-imágenes. La segunda tarea considera el conjunto de datos Brodatz (Brodatz, 1966), que ha sido ampliamente utilizado en la literatura de clasificación de texturas. En este experimento, cada imagen también se divide en un conjunto de 16 sub-imágenes y el objetivo de la tarea de clasificación consiste en asignar a cada sub-imagen la imagen original. En la Tabla 7.5, se resumen las principales características de estos conjuntos de datos y en la Figura 7.4, se muestra un extracto de 5 imágenes por clase del conjunto de datos CGTextures, donde cada clase se compone de diferentes imágenes, haciendo de esta una tarea difícil en la clasificación de texturas.

Para los siguientes experimentos, se ha dividido cada imagen —de lado $L = 480$ píxeles— en 16 sub-imágenes y, para los métodos aquí propuestos, se ha convertido cada sub-imagen en una imagen frecuencial de $12 \times 12$ píxeles

[3]Las texturas se descargaron de `http://www.cgtextures.com/` en 2009 y el conjunto de datos creado y utilizado en este subapartado se puede descargar de `http://www.tsc.uc3m.es/~smunoz/CGTextures.zip`. Debido al origen de las texturas, se hará referencia a este conjunto de datos como CGTextures.

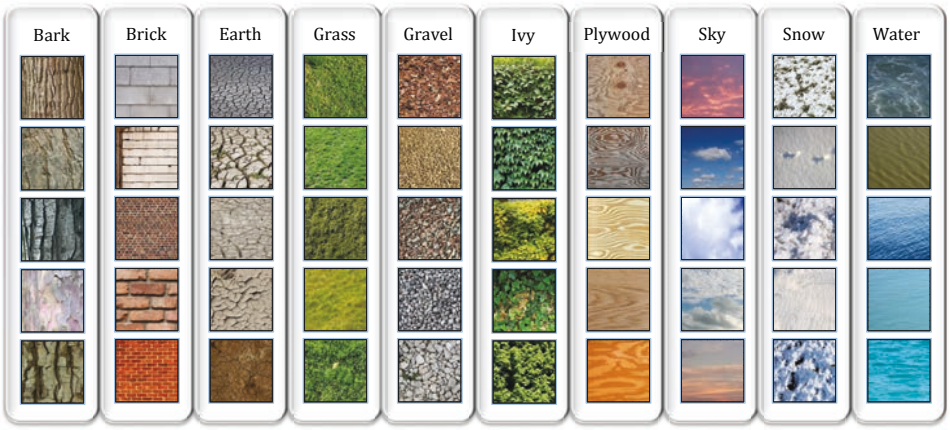



Figura 7.4: Extracto de cinco imágenes por clase del problema CGTextures. En el paso de pre-procesamiento, cada una de estas imágenes de tamaño 480×480 píxeles es dividida en 16 sub-imágenes de tamaño 120×120, que son las imágenes usadas para la tarea de clasificación de texturas.

Tabla 7.5: Descripción de las principales características de los conjuntos de datos de imágenes usados para la clasificación de texturas

| | Núm. imágenes (entrenamiento/test) | Tamaño | Núm. clases |
|---|---|---|---|
| CGTextures | 3840/1568 | 120×120 | 10 |
| Brodatz (Brodatz, 1966) | 1332/444 | 120×120 | 111 |

(es decir, $\rho = 12$), diezmando la imagen frecuencial original por un factor de 10.

En el caso del filtrado de Gabor, se ha hecho también validación cruzada para los parámetros $\eta$ y $\gamma$ (véase la Tabla 7.1), fijando sus valores a $\eta = 0{,}5$ y $\gamma = 0{,}5$ para ambos conjuntos de datos. El resto de los parámetros han sido fijados de acuerdo con Bianconi y Fernández (2007): $n_F = 4$, $n_O = 6$, y $F_r = \sqrt{2}$. Asimismo, se ha validado (CV) el número de filtros en el banco para cada método bajo estudio.

Así pues, se va a estudiar el poder discriminatorio y la interpretabilidad de los diseños de filtros supervisados propuestos en comparación con los, bien diseñados y de modo ad hoc, bancos de filtros de Gabor (Bianconi y Fernández, 2007). Tras diseñar cada banco de filtros, se va a entrenar una C-SVM utilizando los datos de entrada proyectados ($\bar{\mathbf{X}} = \mathbf{UX}$) con el fin de evaluar la precisión total (OA) de cada método; el valor óptimo del parámetro $C$ de la SVM ha sido validado (CV) para cada método bajo estudio. Dado que el objetivo aquí es la obtención de un subconjunto de características interpretables útiles para estas tareas de clasificación, se va a hacer hincapié

Tabla 7.6: Tabla comparativa de las prestaciones entre los métodos propuestos y los Filtros de Gabor ordenados para el conjunto de datos CGTextures

| Algoritmo | OA( %) | $n_f$ | #caract. | NZ - SR( %) |
|---|---|---|---|---|
| NOPLS | **79,91** | 9 | 9 | 66/1440 - (95.42) |
| P-NOPLS | 77.74 | 10 | 10 | 42/1440 - (97.08) |
| defNOPLS | 77.81 | 9 | 9 | 59/1440 - (95.90) |
| NMF-OPLS | 75.96 | 10 | 10 | 65/1440 - (95.49) |
| POPLS | 74.49 | 10 | 10 | 45/1440 - (96.88) |
| OPLS | 79.21 | 8 | 8 | 1152/1440 - (20.00) |
| sorted GF | 73.47 | 24 | 48 | 181140/345600 - (52.41) |

en extraer la cantidad óptima de energía para cada una de las bandas de frecuencia que componen la imagen. La interpretabilidad de los métodos se analizará midiendo el número de frecuencias utilizadas por cada banco de filtros y visualizando los datos proyectados resultantes.

#### 7.3.1.1. Clasificación de texturas en la base de datos CGTextures

En la Tabla 7.6 y en la Figura 7.5, se comparan las prestaciones obtenidas por los métodos propuestos y por el banco de Filtros de Gabor (GF) sobre la base de datos CGTextures. En particular, la Figura 7.5 muestra la evolución de la precisión total (OA) con respecto al número de filtros en el banco y en la Tabla 7.6, se muestra la OA de cada método cuando $n_f$ ha sido seleccionado mediante CV. Para llevar a cabo un análisis justo, se han incluido los resultados de GF ordenando los filtros de acuerdo al MSE en el conjunto de entrenamiento, es decir, por cada $n_f$ seleccionado, dicho subconjunto de filtros alcanza las mejores tasas de reconocimiento.

Como era de esperar, los diseños de los filtros supervisados propuestos y, en especial, los algoritmos NOPLS presentan una mayor precisión con respecto a los esquemas GF —nótese que NOPLS mejora las prestaciones del resto de algoritmos incluyendo OPLS—. Además, el número de filtros utilizados por los bancos de filtros supervisados es menos de la mitad que el número de filtros seleccionados para el banco GF. Asimismo, es importante señalar que, aunque todos los métodos utilizan un número parecido de filtros ($n_f$), el número de bandas de frecuencia seleccionadas por los métodos propuestos es significativamente menor que para GF —como se puede ver en la Tabla 7.6 con la tasa de coeficientes no nulos de los filtros ("Non-Zero coefficients", NZ) y la tasa de dispersión (SR $= 1 -$ NZ)—.

Además, como se ha explicado en el Subapartado 7.1.1, los métodos propuestos extraen únicamente una característica por cada filtro (véase #caract. en la Tabla 7.6), mientras que GF utiliza dos características extraídas por cada filtro: la media de la imagen filtrada ($\mu$) y su desviación estándar ($\sigma$).

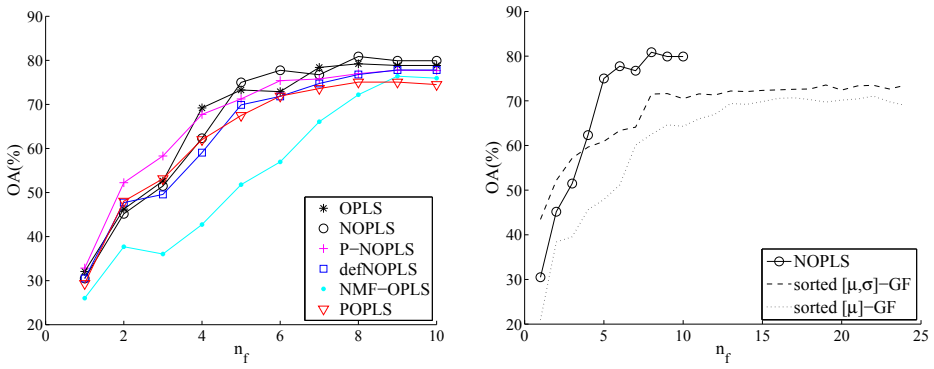



Figura 7.5: Curvas comparativas de las prestaciones entre (a) los métodos propuestos y (b) el mejor de los métodos NOPLS y el banco con los Filtros de Gabor ordenados usando, bien la media y la desviación estándar (sorted $[\mu, \sigma]$-GF), bien solamente la media (sorted $[\mu]$-GF) de cada imagen filtrada.

Con el fin de comparar el rendimiento entre GF con uno o dos características por filtro y el mejor de los métodos propuestos, se muestra una comparación de la evolución de la OA en función del número de filtros considerados en la Figura 7.5b.

En resumen: se puede afirmar que los métodos propuestos son más discriminatorios, más selectivos y más dispersos que GF. Con el fin de analizar la interpretabilidad de cada método bajo estudio de una manera cualitativa, en la Figura 7.6, se muestran los primeros 10 filtros ($\boldsymbol{u}$) del banco de filtros que proporciona cada método, así como un ejemplo de las imágenes filtradas ($\boldsymbol{x}_F = \boldsymbol{x} * \boldsymbol{u}$, siendo $*$ la operación de convolución) de una imagen de la clase *hierba* (o "grass"). Como se puede observar, los filtros supervisados son más precisos y selectivos que los del banco GF, siendo una mezcla de filtros paso-banda orientados horizontalmente, verticalmente y de manera oblicua. Es interesante destacar la similitud entre los filtros en los bancos de NOPLS, defNOPLS e incluso los primeros filtros de POPLS, lo cual es indicativo de que NOPLS funciona mejor que P-NOPLS. Con respecto al banco GF, se puede observar que la peor precisión del sistema de clasificación recae sobre las características obtenidas por GF, indicando que este conjunto de filtros no pudo extraer características suficientemente discriminatorias para la tarea en cuestión. Con todo esto, se confirma la conveniencia de diseñar bancos de filtros de manera supervisada.

#### 7.3.1.2. Clasificación de texturas en la base de datos Brodatz

En este subapartado, se evalúan los diferentes métodos bajo estudio sobre el escenario de clasificación de texturas Brodatz. En este caso, cada sub-

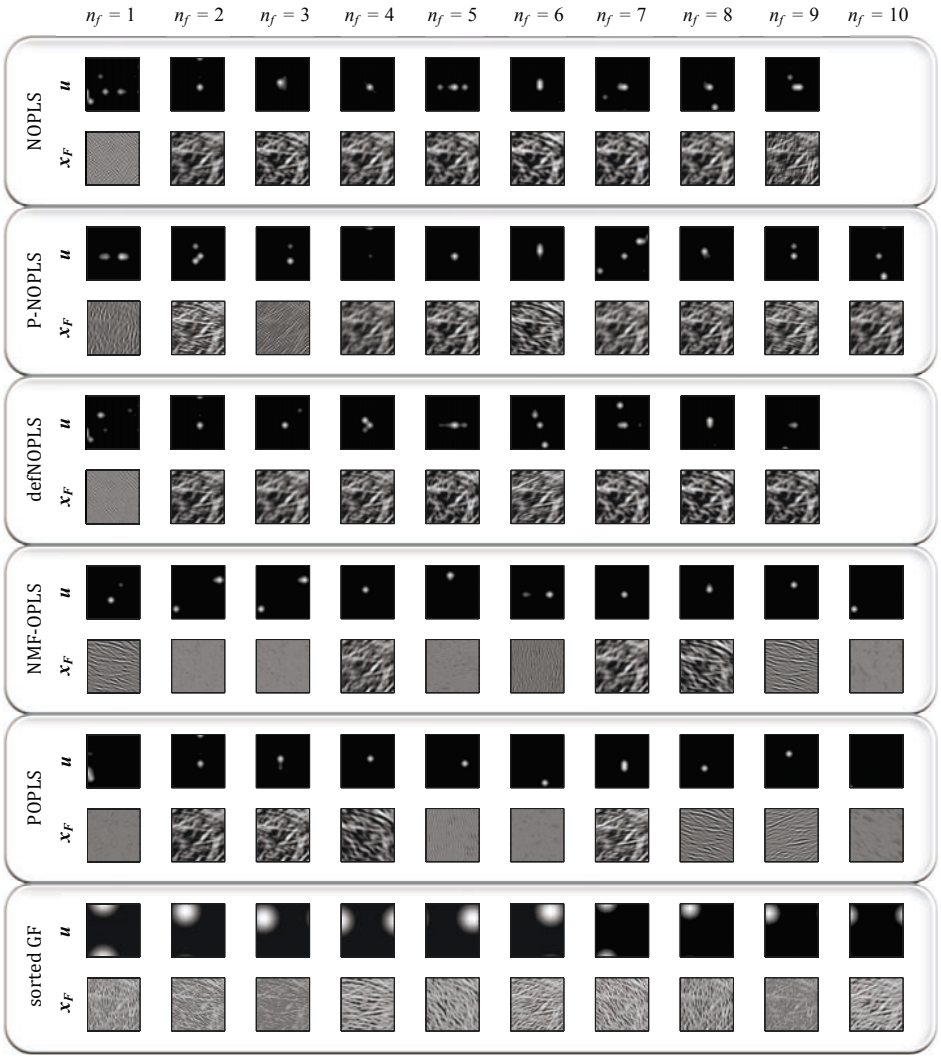



Figura 7.6: Representación de la respuesta en frecuencia ($\boldsymbol{u}$) de los 10 primeros filtros utilizados por cada método en la tarea de clasificación de texturas. Las correspondientes imágenes filtradas ($\boldsymbol{x}_F$) para un ejemplo de la clase *hierba* también se han representado para los diferentes métodos y filtros.

imagen ha de ser asignada a su imagen original y, por consiguiente, el número de clases a ser etiquetadas es el mismo que el número original de imágenes disponibles en la base de datos. Al igual que en el subapartado anterior, se comparan los métodos propuestos con el banco GF, aunque en este caso se utiliza el banco GF propuesto por Bianconi y Fernández (2007), donde el banco GF está diseñado ad hoc para esta tarea en particular.

Siguiendo el mismo procedimiento experimental que en el experimento anterior, en la Tabla 7.7 y en la Figura 7.7 se incluye una comparación de los

diferentes métodos bajo estudio. Nuevamente, para obtener una comparación más justa, los filtros en el banco GF han sido ordenados según el criterio (7.2) medido sobre el conjunto de entrenamiento. Para examinar más a fondo las diferencias entre los métodos propuestos, en este subapartado se va a analizar los tiempos de entrenamiento requeridos para obtener los bancos de filtros. Todas las simulaciones se ejecutaron utilizando Matlab 8 en un MacBook Pro con 8 GB de memoria RAM y un procesador 2,9 GHz dual-core Intel Core i7 CPU.

Como se puede observar, todos los métodos supervisados son más discriminatorios que el banco GF, incluso cuando hay pocos filtros en los bancos. Aunque P-NOPLS es ligeramente más discriminatorio que NOPLS, su entrenamiento es mucho más costoso y el número de filtros es también más alto. Es importante remarcar que NOPLS es el algoritmo más rápido (2,34 s) y requiere la mitad de características que GF, mientras que, en este caso, la solución defNOPLS es la más discriminatoria y el segundo más rápido (14,84 s). P-NOPLS y NMF-OPLS necesitan alrededor de 20 s y POPLS es considerablemente más lento con 12 h y 12 min. A diferencia de la base de datos anterior, aquí GF utiliza menos filtros que los esquemas supervisados; sin embargo, el número de coeficientes es parecido (excepto para P-NOPLS) y el número de bandas de frecuencia de las imágenes necesarias para los algoritmos propuestos es considerablemente más pequeño que para GF (véase NZ y SR en la Tabla 7.7). Comparando con los resultados de OPLS, se puede observar que la solución OPLS estándar obtiene las peores prestaciones usando cualquier subconjunto de filtros. Este hecho señala que las restricciones de no negatividad no solo proporciona soluciones interpretables, sino también (en algunos casos) mejora las prestaciones.

Nótese que, como se explicó en el apartado 7.2, P-NOPLS y NMF-OPLS no ordenan los filtros del banco en función de importancia. Una de las consecuencias de esto es que requieren más filtros que los otros métodos supervisados; por ejemplo, se puede ver que P-NOPLS necesita el doble de filtros que el resto de métodos.

### 7.3.2. Experimento 2: Clasificación de género musical

Este segundo bloque de experimentos tiene como objetivo clasificar el género musical de una canción a partir del periodograma de los 6 primeros MFCC extraídos de cada canción. El conjunto de datos utilizado aquí ha sido investigado previamente por Arenas-García et al. (2006), Meng et al. (2007) y Meng y Shawe-Taylor (2005), y sus resultados han revelado una gran dificultad para clasificar con éxito cada canción de acuerdo a su género musical (véanse Arenas-García et al., 2006; Meng y Shawe-Taylor, 2005). Además, el estudio de evaluación humana de Meng y Shawe-Taylor (2005) ha encontrado que la definición humana del género musical para los audios en este conjunto de datos presenta baja consistencia, dando como resultado un

Tabla 7.7: Tabla comparativa de las prestaciones entre los métodos propuestos y el ordenado GF en la base de datos de Brodatz

| Algoritmo | OA(%) | $n_f$ | #caract. | NZ - SR(%) |
|---|---|---|---|---|
| NOPLS | 90.32 | 24 | 24 | 238/15984 - (98.51) |
| P-NOPLS | 91.22 | 105 | 105 | 256/15984 - (98.40) |
| defNOPLS | **92**,**12** | 53 | 53 | 291/15984 - (98.18) |
| NMF-OPLS | 90.99 | 63 | 63 | 144/15984 - (99.10) |
| POPLS | 91.67 | 55 | 55 | 95/15984 - (99.41) |
| OPLS | 85.81 | 20 | 20 | 2880/15984 - (81.98) |
| sorted GF | 90.09 | 23 | 46 | 179771/345600 - (47.98) |

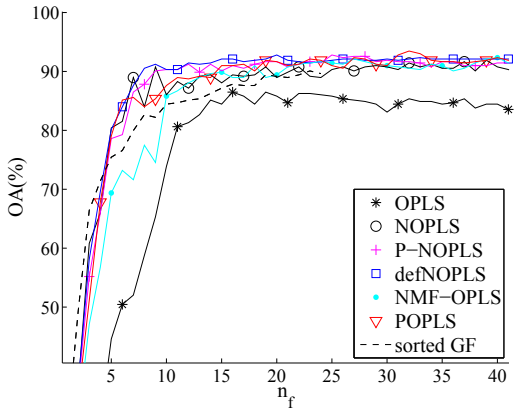



Figura 7.7: Figura comparativa de las prestaciones entre los métodos propuestos y GF para la base de datos Brodatz. Estas curvas representan la OA en función del número de filtros usado en el banco de filtros ($n_f$).

conjunto de datos difíciles de aplicar para la tarea de clasificación de género musical. No obstante lo anterior, es interesante estudiar cómo el diseño de bancos de filtros supervisados funciona en esta configuración.

El conjunto de datos consta de 1 317 fragmentos de música de 30 s cada uno, distribuidos en partes iguales entre los siguientes 11 géneros musicales: *Alternative*, *Country*, *Easy Listening*, *Electronica*, *Jazz*, *Latin*, *Pop&Dance*, *Rap&Hiphop*, *R&B and Soul*, *Reggae* y *Rock*. En caso de la categoría *Latin*, solamente hay 117 muestras musicales. Los fragmentos de música están codificados en MP3 con una tasa de bit de 128 kbps o un mayor submuestreo con factor dos a 22 050 Hz. Nótese que este conjunto de datos tiene un promedio de 1,83 canciones por artista, que es otra de las razones que lo hace tan difícil para la clasificación de género.

Con fines comparativos, se va a considerar el banco de filtros Philips ("Philips Filters") propuesto por McKinney y Breebaart (2003) para una

tarea de clasificación de género musical. Como se explicó en la parte final del subapartado 7.1.2, se usarán periodogramas de longitud $D = 129$, de manera que el tamaño de las matrices $\mathbf{U}$, que caracterizan tanto el banco de filtros Philips como los bancos supervisados diseñados con cualquiera de los métodos propuestos, será $129 \times n_f$, siendo $n_f = 4$ para el banco de filtros Philips.

Debido a la falta de un subconjunto específico de test, se aplica un procedimiento de validación cruzada con 10 particiones con el fin de medir la precisión de la clasificación de cada método. En cada partición, se obtienen los filtros óptimos con nueve particiones de los datos —tal como se describe en el subapartado 7.1.2— y, posteriormente, se evalúa las prestaciones sobre la partición restante. Téngase en cuenta que muestras de la misma canción no pueden ser divididas en particiones diferentes. Dicho de otro modo: las particiones son definidas en función de las canciones y no en función de las muestras del conjunto de datos.

En la Tabla 7.8, se comparan las prestaciones entre los esquemas con filtros supervisados y el banco de filtros Philips. En concreto, esta tabla muestra la OA (promedio de las 10 particiones) cuando se usan los 4 y los 10 primeros filtros del banco ($n_f = 4$ y $n_f = 10$ respectivamente). En el caso del banco de filtros Philips, los resultados se analizan solamente con 4 filtros, ya que este es su número máximo de filtros disponibles. La Tabla 7.8 también incluye la tasa de coeficientes no nulos de los filtros (NZ), así como el tiempo requerido para diseñar los diferentes bancos de filtros. Para completar este análisis, la Figura 7.8 muestra la OA promedio en función del número de filtros de todos los métodos bajo estudio.

Como se explicó en el Apartado 7.2, dos de los métodos propuestos (P-NOPLS y NMF-OPLS) carecen de la capacidad de clasificar los filtros del banco con respecto a la importancia de cada filtro. Como consecuencia de esta carencia, cuando se usan solamente unos pocos filtros, las prestaciones pueden verse afectadas negativamente, como es el caso aquí, donde estos métodos son incluso superados por el banco de filtros Philips cuando $n_f = 4$. En cuanto al resto de los filtros supervisados, no resulta tan claro cuál de ellos presenta las mejores prestaciones: aunque POPLS tiene la mejor precisión con $n_f = 10$, NOPLS obtiene prestaciones parecidas, pero con un menor porcentaje de coeficientes no nulos. Más aún, las precisiones obtenidas por los métodos defNOPLS y NOPLS son las mejores cuando se utilizan pocos filtros (véase la Figura 7.8a), mejorando significativamente las prestaciones del banco de filtros Philips. Con respecto a OPLS, se puede observar que, como era de esperar, es el algoritmo más rápido, ya que no incluye restricciones en su formulación; sin embargo, se puede ver que no solo no obtiene soluciones interpretables —todos sus coeficientes son no nulos—, sino que también obtiene las peores prestaciones cuando usa los primeros filtros.

Para concluir este apartado, en la Figura 7.9, se muestran los 4 primeros

Tabla 7.8: OA (%) de los distintos métodos bajo estudio en la tarea de clasificación de género. Los resultados están dados para bancos con $n_f = 4$ y $n_f = 10$ filtros. También se muestra el número de coeficientes distintos de cero (NZ) como un porcentaje del número total de coeficientes, junto con el tiempo de entrenamiento requerido por cada método.

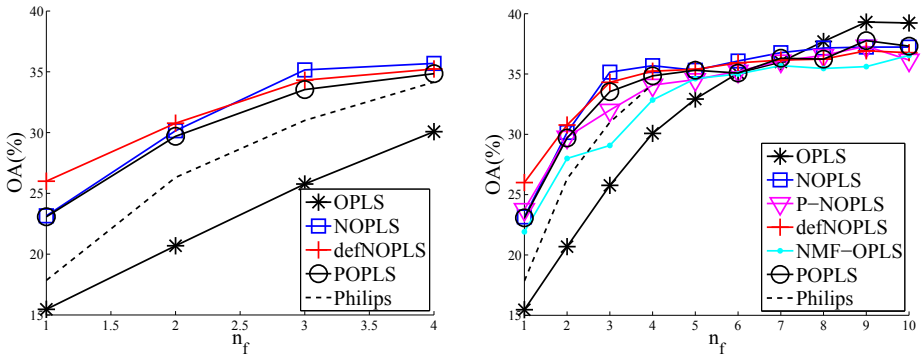
| Algoritmo | OA ($n_f = 4$) | OA ($n_f = 10$) | NZ(%) | Tiempo (s) |
|---|---|---|---|---|
| NOPLS | **35.69** | 37.23 | **2.9** | 6.56 |
| P-NOPLS | 34.07 | 36.15 | 16.67 | 7.65 |
| defNOPLS | 35.23 | 36.77 | 3.9 | 15.40 |
| NMF-OPLS | 32.85 | 36.54 | 6.27 | 32.13 |
| POPLS | 34.85 | 37.31 | 13.76 | 2667.59 |
| OPLS | 30.08 | **39.23** | 100.0 | **2.7** |
| Filtros Philips | 34.15 | - | 3.84 | - |

filtros obtenidos en una única partición[4] para el primer MFCC, de manera que se pueda analizar la información proporcionada por cada banco de filtros. Es interesante destacar que —de manera similar al banco de filtros Philips— NOPLS, defNOPLS y POPLS prestan atención a tres regiones bien diferenciadas de los espectros (a pesar de no presentarse en el mismo orden): las frecuencias de modulación más bajas, que incluyen componentes en la escala del ritmo; las frecuencias de modulación más altas, que están relacionados con la *rugosidad* en la percepción; y las frecuencias de modulación de los instrumentos, que son las frecuencias más importantes de los periodogramas de los MFCC. Además, los esquemas supervisados son más flexibles en la definición de los filtros y pueden ajustar las frecuencias de corte e, incluso, moldear la forma de onda del filtro para obtener las mejores prestaciones posibles en la tarea de clasificación de género musical. Esta superioridad en las prestaciones obtenidas por parte de las técnicas supervisadas permite concluir la conveniencia de usar las etiquetas disponibles no solo para el entrenamiento del clasificador final, sino también en el diseño de los filtros utilizados en la etapa de extracción de características.

## 7.4. Conclusiones

En este capítulo, se han presentado diferentes métodos versátiles con el fin de diseñar bancos de filtros interpretables para unas determinadas tareas de clasificación de imágenes o de audio. Todos los métodos propuestos se basan en un diseño supervisado con una función de coste objetivo común, y difieren

[4]Se ha comprobado que las diferencias entre los filtros obtenidos en cada partición no son muy significativas, por lo que las conclusiones presentadas pueden ser fácilmente generalizadas al resto de particiones.




a) b)

Figura 7.8: Precisión total (OA) respecto a: (a) un estudio comparativo detallado entre los mejores bancos de filtros supervisados y el banco de filtros Philips (solamente los primeros 4 filtros); y (b) una comparación completa entre todos los métodos con el banco de filtros completo

en el modo de resolver este problema no convexo. Como una alternativa al algoritmo POPLS propuesto en Arenas-García et al. (2006), se han propuesto en este capítulo diversos métodos que requieren mucho menos tiempo de entrenamiento y obtienen unas prestaciones parecidas o, incluso, mejores que POPLS. Además, estas propuestas mejoran a aquellos bancos de filtros que están siendo utilizados en el estado del arte de las aplicaciones visuales y de audio y que han sido muy bien estudiados y diseñados ad hoc para cada tarea en cuestión.

En el apartado de experimentos, se ha mostrado la versatilidad de los métodos propuestos, donde se han abordado dos tareas de clasificación muy diferentes: la clasificación de texturas y de género musical. Las ventajas de estos esquemas sobre otros métodos de extracción de características son: 1) que proporcionan interpretaciones físicas elegantes de las características extraídas; 2) que son más discriminatorios a la vez que requieren un menor número de filtros; 3) que proporcionan soluciones más interpretables y dispersas; y 4) que ajustan sus bancos de filtros para cada tarea en particular, a diferencia de los bancos de filtros genéricos. En base a estos resultados, se puede concluir que los algoritmos NOPLS tanto bloque como deflactados parecen obtener los mejores resultados en términos de precisión, dispersión y requisitos de computación y que, por lo tanto, deberían ser una opción preferible frente a los otros métodos, incluyendo los diseños que ya existen de filtros basados en conocimiento experto.

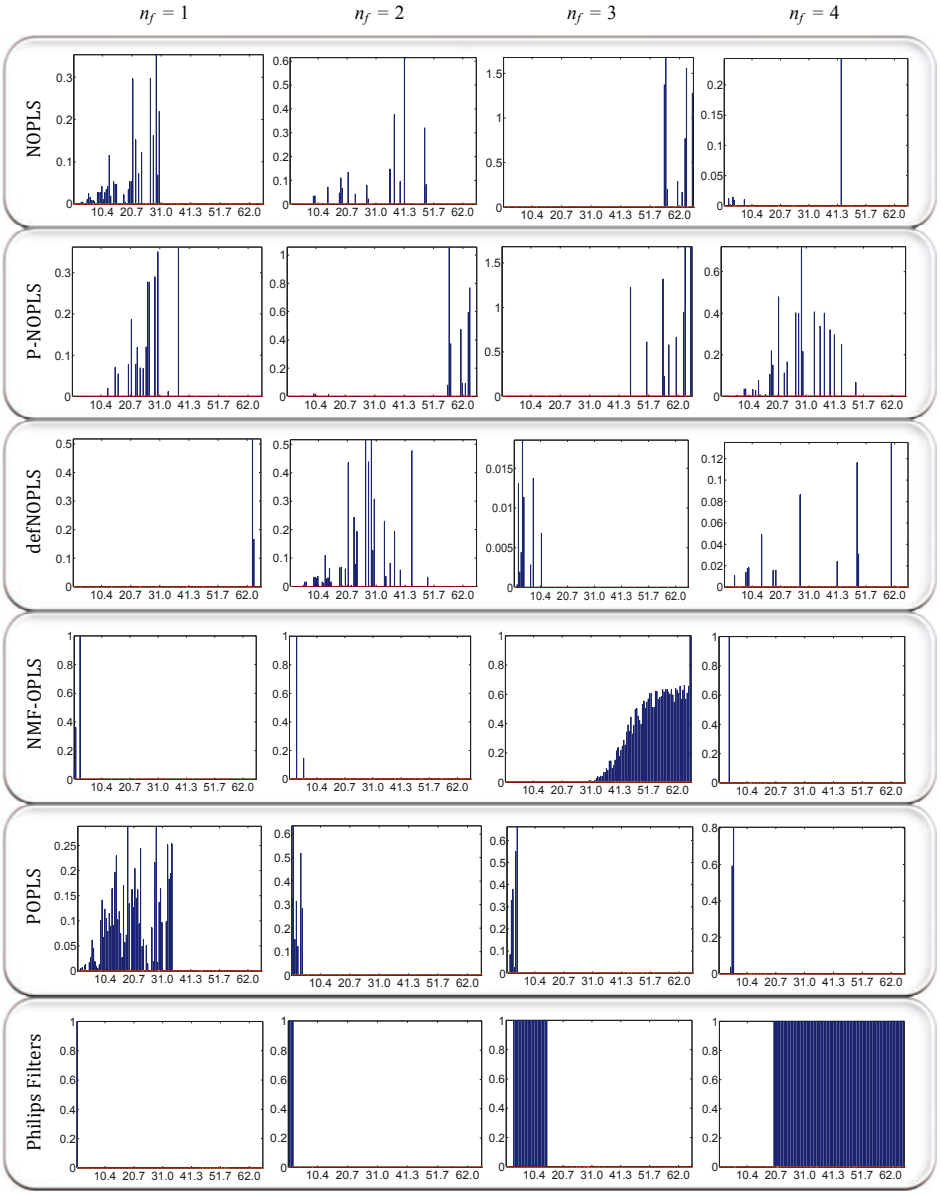



Figura 7.9: Respuesta en frecuencia de los cuatro primeros filtros diseñados por cada algoritmo

# Capítulo 8

# Conclusiones y líneas futuras

*Cuando llegamos a la meta, creemos que*
*el camino ha sido el bueno.*
Paul Valéry (1871-1945)

**Resumen:** En este capítulo, se reflexiona sobre el trabajo realizado a lo largo de esta tesis doctoral, revisando las soluciones propuestas en cada capítulo. Además, como fruto de este análisis, se identifican diversas líneas de investigación que quedan abiertas.

## 8.1. Conclusiones

En esta tesis doctoral, se ha propuesto un marco general MVA que engloba algunos de los métodos de análisis multivariante más conocidos en la literatura debido a su utilidad y buenas prestaciones —como PCA, CCA y OPLS—, excluyendo aquellos métodos MVA que no blanquean los datos de entrada (características ortogonales), como el PLS. La ventaja de obtener estas características ortogonales son principalmente dos: la posibilidad de ordenar las características por orden de importancia, de forma tal que se pueda reducir la dimensionalidad de entrada con el subconjunto de vectores más representativos del problema; y la capacidad de facilitar el entrenamiento de ulteriores máquinas de aprendizaje que usan como entrada dichos datos blanqueados.

Las ventajas que presenta este marco general MVA son esencialmente las siguientes:

- Eficiencia.— Permite obtener soluciones eficientes en función del tamaño de los conjuntos de entrada y salida, reduciendo considerablemente el coste computacional cuando la diferencia entre sus dimensiones es alta.

- Flexibilidad o versatilidad.— Permite incluir restricciones adicionales en función de las necesidades del problema, de modo que aporta soluciones especializadas para una tarea concreta.
- Base teórica firme.— Aquí se ha demostrado teóricamente que las soluciones MVA con restricciones propuestas hasta el momento en la literatura presentan graves problemas en su formulación debido al uso de la aproximación de Procrustes. Para solventarlo, se ha demostrado que la única solución posible para evitar tales problemas se obtiene mediante el uso de este marco general MVA con restricciones.

En particular, la mencionada característica de flexibilidad ha sido profusamente explotada en la tesis doctoral, desarrollando nuevos métodos MVA que incorporan diversas características deseables, principalmente con el objetivo de obtener métodos más precisos y, sobre todo, más interpretables. De esta manera, se han presentado métodos MVA que favorecen:

- Soluciones dispersas, de forma que cada característica extraída se obtenga como combinación lineal de algunas de las variables de entrada originales.
- Soluciones no lineales dispersas que permiten capturar las relaciones no lineales entre variables al mismo tiempo que seleccionan las funciones kernel relevantes para tal fin.
- Soluciones parsimoniosas que permiten seleccionar las variables relevantes del problema. Nótese que las soluciones dispersas no realizan una extracción de características tan útil, pues la extracción de las diferentes variables suele hacerse a partir de subconjuntos de características originales diferentes para cada variable extraída.
- Soluciones con restricciones de no negatividad para diseñar bancos de filtros supervisados que definan los rangos frecuenciales donde se concentra la energía de interés para la tarea a resolver.

La parte experimental de cada una de las propuestas presentadas ha confirmado en todos los casos que no solo se obtienen iguales o mejores prestaciones que las soluciones existentes en el estado del arte, sino que también se aporta un valor añadido que está siendo cada vez más demandado: la interpretabilidad de las soluciones.

Los ejemplos más claros provienen de las soluciones parsimoniosas —que permitirían conocer, por ejemplo, qué parte del genoma es relevante para determinar los distintos tipos de carcinomas, además de obtener una mayor tasa de acierto— o de las soluciones con restricciones de no negatividad —que distinguirían los rangos de frecuencia más importantes para clasificar, por ejemplo, texturas o géneros musicales, además de mejorar las prestaciones obtenidas en el estado del arte—.

## 8.2. Líneas futuras de investigación

Las líneas futuras de investigación podrían dividirse en nuevas formulaciones, nuevas aplicaciones y nuevas implementaciones:

- Como nuevo tipo de solución, se podría aplicar la norma $\ell_{2,1}$ presentada en el Capítulo 6 sobre las soluciones no lineales propuestas en el Capítulo 5. De este modo, al forzar dispersión sobre filas enteras de la matriz solución, se conseguiría seleccionar las muestras relevantes del problema, al mismo tiempo que se capturan las relaciones no lineales entre variables. Este tipo de soluciones resultan de suma importancia en problemas donde se ha capturado de manera indiscriminada una gran cantidad de datos y se quiere saber cuáles de ellos son relevantes para una determinada tarea. Este tipo de problemas se suelen encontrar en escenarios de "Big Data".

- Como nueva aplicación, se podrían evaluar las soluciones con restricciones de no negatividad sobre otras tareas que permitan organizar la música por su naturaleza, como, por ejemplo, la clasificación o detección de instrumentos musicales. De este modo, permitiría incluir nuevas funcionalidades en programas de reproducción musical; por ejemplo, si a alguien le gusta mucho el saxofón, se le podría ofrecer un listado de canciones donde se toca dicho instrumento.

- Y como mejora en la implementación, se podría adaptar las soluciones aquí propuestas o, incluso, crear algoritmos de aprendizaje máquina implementables de manera completamente distribuida o *embarazosamente paralelizables* ("embarrassingly parallel"), sin perder la cualidad aquí conseguida de interpretabilidad de las soluciones. La principal motivación de esta línea de investigación se debe al imparable crecimiento en el número de instituciones que están invirtiendo en infraestructuras para el procesamiento de datos y extracción de conocimiento, como son los *clústeres* de ordenadores para acelerar el tratamiento de los datos disponibles mediante la división de la tarea a ejecutar en otras más pequeñas y distribuidas entre las distintas máquinas. Por lo tanto, extendiendo las soluciones aquí propuestas a su implementación distribuida, se conseguiría no solo obtener soluciones interpretables que ayudan a la comprensión del problema y a la toma final de decisiones, sino también acelerar la obtención de dichas soluciones. Esto es de suma importancia actualmente en el mercado, ya que debido a la ingente cantidad de datos disponibles, resulta inviable recurrir a la mayoría de los algoritmos de aprendizaje máquina existentes. Esto está provocando, en la actualidad, el abandono de estas soluciones —y, como consecuencia, de sus excelentes prestaciones— por unos resultados poco precisos pero factibles de obtener.

Mirando las tendencias y necesidades existentes en la actualidad, sería deseable, además, analizar la viabilidad de los métodos aquí propuestos para la creación de herramientas *"Invisible Analytics"*: herramientas que, dada una gran cantidad de datos y variables ("Big Data"), permiten obtener la parte importante de ellos, obtener patrones ocultos en ellos y devolver respuestas a las cuestiones realizadas en tiempo real, de manera transparente para el usuario. En particular, las soluciones parsimoniosas propuestas, al permitir seleccionar únicamente las variables necesarias para una pregunta realizada ("Big Question"), permiten devolver una respuesta adecuada de manera eficiente ("Big Answer"). Debido a esto, el desarrollo de un método de selección de variables preciso y eficiente —como el propuesto en el Capítulo 6— resulta especialmente atractivo para ser incluido como parte de dichas herramientas de "Invisible Analytics".

# Parte III

# Apéndices

# Apéndice A

# Material complementario para la revisión de conceptos MVA

## Propiedades de los autovalores

Algunas de las propiedades más destacables de los autovalores, dada una matriz simétrica $\mathbf{C}$, son:

(a) Como $\mathbf{C}$ es simétrica, sus autovalores son siempre reales y sus correspondientes autovectores son todos distintos y ortogonales.

(b) El producto de todos los autovalores corresponde al determinante de la matriz $\mathbf{C}$:

$$\det(\mathbf{C}) = \prod_{k=1}^{n} \lambda_k.$$

(c) La suma de todos los autovalores determina la traza de $\mathbf{C}$:

$$\mathrm{Tr}\{\mathbf{C}\} = \sum_{k=1}^{n} \lambda_k.$$

(d) La matriz de autovalores de $\mathbf{C}^{-1}$ es $\mathbf{\Lambda}^{-1}$.

(e) La matriz de autovalores de $\mathbf{C}^p$, siendo $p$ un número natural no nulo, es $\mathbf{\Lambda}^p$.

(f) La matriz de autovalores de $a\mathbf{C}$, siendo $a$ un escalar, es $a\mathbf{\Lambda}$.

(g) Si $\mathbf{C}$ es singular (es decir, de rango deficiente, rango($\mathbf{C}$) $= r < n$), entonces los últimos $n - r$ autovalores serán iguales a cero, siendo el orden de la diagonal de $\Lambda$: $\lambda_1 \geq \cdots \geq \lambda_r \geq \lambda_{r+1} = \cdots = \lambda_n = 0$.

(h) Si todos los autovalores de $\mathbf{C}$ son mayores que cero, $\lambda_k > 0,\ k = 1, \ldots, n$, se dice que $\mathbf{C}$ es *definida positiva* ($\mathbf{C} \succ 0$) y se cumple que $\boldsymbol{v}^\top \mathbf{C} \boldsymbol{v} > 0,\ \forall \boldsymbol{v} \in \mathbb{R}^n$.

(i) Si todos los autovalores de $\mathbf{C}$ son mayores o iguales a cero, $\lambda_k \geq 0,\ k = 1, \ldots, n$, se dice que $\mathbf{C}$ es *semidefinida positiva* ($\mathbf{C} \succeq 0$) y se cumple que $\boldsymbol{v}^\top \mathbf{C} \boldsymbol{v} \geq 0,\ \forall \boldsymbol{v} \in \mathbb{R}^n$. Toda matriz de covarianzas $\mathbf{C}$ ha de cumplir esta propiedad.

(j) Todos los autovalores de $\mathbf{C}$ deben satisfacer el polinomio característico: $\det(\mathbf{C} - \lambda_k \mathbf{I}) = 0$.

# Apéndice B

# Material complementario para el marco general MVA

*La práctica debe siempre ser edificada sobre la buena teoría.*

Leonardo Da Vinci

## Demostración de equivalencia entre OPLS y RRR

Se comenzará señalando que, puesto que las columnas de $\mathbf{U}_{\text{EVD}}$ y $\mathbf{U}_{\text{GEV}}$ definen el mismo espacio de $\mathbb{R}^{n\times n_f}$, estas deberían verificar que $\mathbf{U}_{\text{EVD}} = \mathbf{U}_{\text{GEV}}\mathbf{A}$ para alguna matriz cuadrada e invertible $\mathbf{A} \in \mathbb{R}^{n_f}$. Sustituyendo esta expresión en (3.14), y tomando en consideración que las columnas de $\mathbf{U}_{\text{GEV}}$ son $\mathbf{C_{XX}}$-ortonormales (es decir, $\mathbf{U}^\top\mathbf{C_{XX}}\mathbf{U} = \mathbf{I}$), se llega a

$$\mathbf{A}^\top\mathbf{U}_{\text{GEV}}^\top\mathbf{C_{XX}}\mathbf{U}_{\text{GEV}}\mathbf{A} = \mathbf{A}^\top\mathbf{A} = \mathbf{\Lambda}_{\text{EVD}}. \tag{B.1}$$

Puesto que $\mathbf{\Lambda}_{\text{EVD}}$ admite la factorización de Cholesky y es única, necesariamente se obtiene que $\mathbf{A} = \mathbf{A}^\top = \mathbf{\Lambda}_{\text{EVD}}^{1/2}$ y

$$\mathbf{U}_{\text{EVD}} = \mathbf{U}_{\text{GEV}}\mathbf{\Lambda}_{\text{EVD}}^{1/2}. \tag{B.2}$$

A continuación, se mostrará la relación entre las matrices de coeficientes de regresión. Para tal fin, se puede insertar (3.13) en (3.14), obteniendo $\mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}_{\text{EVD}} = \mathbf{W}_{\text{EVD}}\mathbf{\Lambda}_{\text{EVD}}$. Además, si se usa (B.2) con (3.7), se puede mostrar fácilmente que $\mathbf{W}_{\text{GEV}}\mathbf{\Lambda}_{\text{EVD}}^{1/2} = \mathbf{C}_{\mathbf{XY}}^\top\mathbf{U}_{\text{EVD}}$. Usando de manera conjunta estas dos últimas ecuaciones, resulta sencillo llegar a

$$\mathbf{W}_{\text{EVD}} = \mathbf{W}_{\text{GEV}}\mathbf{\Lambda}_{\text{EVD}}^{-1/2}. \tag{B.3}$$

Para concluir esta demostración, se necesitaría mostrar que $\mathbf{\Lambda}_{\text{EVD}} = \mathbf{\Lambda}_{\text{GEV}} = \mathbf{\Lambda}$, en cuyo caso sería suficiente usar (B.3) junto con la condición

$\mathbf{W}_{\text{GEV}}^{\top}\mathbf{W}_{\text{GEV}} = \mathbf{\Lambda}_{\text{GEV}}$ para la solución OPLS clásica. Y si se recurre también a la condición de ortonormalidad de las columnas de $\mathbf{W}_{\text{EVD}}$, finalmente se obtiene

$$\mathbf{W}_{\text{GEV}}^{\top}\mathbf{W}_{\text{GEV}} = \mathbf{\Lambda}_{\text{EVD}}^{1/2}\mathbf{W}_{\text{EVD}}^{\top}\mathbf{W}_{\text{EVD}}\mathbf{\Lambda}_{\text{EVD}}^{1/2} = \mathbf{\Lambda}_{\text{EVD}} = \mathbf{\Lambda}_{\text{GEV}}. \tag{B.4}$$

# Apéndice C

# Material complementario para las soluciones MVA no negativas

## Un breve resumen de los bancos de filtros de Gabor

En tareas de clasificación de texturas, es frecuente realizar el análisis en frecuencia de una señal bi-dimensional por medio de un filtro de Gabor también bi-dimensional que consiste en una onda sinusoidal modulada por una envolvente Gaussiana. Las desviaciones estándar de esta envolvente Gaussiana tanto en la dirección de la onda como en la ortogonal a esta están determinadas por los parámetros de suavizado $\gamma$ y $\eta$ respectivamente. Estos parámetros determinan la selectividad del filtro en el dominio espacial.

El filtro de Gabor en este dominio se define como sigue (véase Kamarainen, 2003):

$$\psi(x,y) = \frac{F}{\pi\gamma\eta}\ \mathrm{e}^{i2\pi F x'}\ \mathrm{e}^{F^2\left[\left(\frac{x'}{\gamma}\right)^2+\left(\frac{y'}{\eta}\right)^2\right]},$$

donde $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, $\theta$ es el ángulo entre el eje $x$ del dominio espacial y la dirección de la onda sinusoidal y $F$ es la frecuencia central del filtro. En esta formulación, el eje de la envolvente Gaussiana y la dirección de la onda están alineados.

El filtro de Gabor se puede formular también en el dominio de la frecuencia como

$$\Psi(u,v) = \mathrm{e}^{\left(\frac{\pi}{F}\right)^2\left[\gamma^2(u'-F)+\eta^2 v'^2\right]},$$

siendo $u' = u\cos\theta + v\sin\theta$ y $v' = -u\sin\theta + v\cos\theta$.

# Bibliografía

*El ver mucho y el leer mucho aviva los ingenios de los hombres.*

Miguel de Cervantes Saavedra
(1547-1616)


Allen, G. I., Peterson, C., Vannucci, M. y Maletić-Savatić, M. Regularized partial least squares with an application to NMR spectroscopy. *Statistical Analysis and Data Mining*, vol. 6(4), páginas 302–314, 2013.

Arenas-García, J. y Camps-Valls, G. Efficient kernel orthonormalized PLS for remote sensing applications. *IEEE Trans. Geosci. Remote Sens.*, vol. 46(10), páginas 2872–2881, 2008.

Arenas-García, J., Larsen, J., Hansen, L. K. y Meng, A. Optimal filtering of dynamics in short-time features for music organization. En *Proc. 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, páginas 290–295. Victoria, Canada, 2006.

Arenas-García, J., Petersen, K., Camps-Valls, G. y Hansen, L. K. Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods. *IEEE Signal Process. Mag.*, vol. 30(4), páginas 16–29, 2013.

Arenas-García, J. y Petersen, K. B. Kernel multivariate analysis in remote sensing feature extraction. En *Kernel Methods for Remote Sensing Data Analysis* (editado por G. Camps-Valls y L. Bruzzone). Wiley, 2009.

Arenas-García, J., Petersen, K. B. y Hansen, L. K. Sparse kernel orthonormalized PLS for feature extraction in large data sets. En *Advances in Neural Information Processing Systems 19*, páginas 33–40. The MIT Press, 2007.

Aucouturier, J.-J., Pachet, F. y Sandler, M. The way it sounds": timbre models for analysis and retrieval of music signals. *IEEE Trans. Multimedia*, vol. 7(6), páginas 1028–1035, 2005.

BACH, F., JENATTON, R., MAIRAL, J. y OBOZINSKI, G. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, páginas 19–53, 2011.

BARKER, M. y RAYENS, W. Partial least squares for discrimination. *Journal of Chemometrics*, vol. 17(3), páginas 166–173, 2003.

BI, J., BENNETT, K., EMBRECHTS, M., BRENEMAN, C. y SONG, M. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research 3*, páginas 1229–1243, 2003.

BIANCONI, F. y FERNÁNDEZ, A. Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, vol. 40(12), páginas 3325–3335, 2007.

BIANCONI, F., FERNÁNDEZ, A. y MANCINI, A. Assessment of rotation-invariant texture classification through Gabor filters and discrete Fourier transform. En *Proc. 20th Intl. Congress on Graphical Engineering*. Valencia, Spain, 2008.

BISHOP, C. *Neural Networks for Pattern Recognition*. Oxford University Press, New York (NY), 1995.

BORGA, M., LANDELIUS, T. y KNUTSSON, H. A unified approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, Linköping University, SE-581 83 Linköping, Sweden, 1997.

BOUTSIDIS, C. y GALLOPOULOS, E. SVD based initialization: A head start for nonnegative matrix factorization. *Journal of Pattern Recognition*, vol. 41(4), páginas 1350–1362, 2008.

BRODATZ, P. *Textures: a photographic album for artists and designers*, vol. 66. Dover New York, 1966.

CAI, D., HE, X., HAN, J. y ZHANG, H.-J. Orthogonal laplacianfaces for face recognition. *IEEE Trans. Image Process.*, vol. 15(11), páginas 3608–3614, 2006.

CHEN, L. y HUANG, J. Z. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, vol. 107(500), páginas 1533–1545, 2012.

CHEN, X. y RAMADGE, P. J. Music genre classification using multiscale scattering and sparse representations. En *Proc. 47th Annual Conf. on Information Sciences and Systems (CISS)*, páginas 1–6. Baltimore, Maryland, USA, 2013.

CHOI, S. Algorithms for orthogonal nonnegative matrix factorization. En *Proc. IEEE Intl. Joint Conf. on Neural Networks, IJCNN 2008*, páginas 1828–1832. Hong Kong, China, 2008.

DENG, L., CHENG, K.-K., DONG, J., GRIFFIN, J. L. y CHEN, Z. Non-negative principal component analysis for NMR-based metabolomic data analysis. *Chemometrics and Intelligent Laboratory Systems*, vol. 118(0), páginas 51–61, 2012.

DHANJAL, C., GUNN, S. R. y SHAWE-TAYLOR, J. Efficient sparse kernel feature extraction based on partial least squares. *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 31(8), páginas 1347–1361, 2009.

DING, C., ZHOU, D., HE, X. y ZHA, H. $R_1$-PCA: Rotational invariant $L_1$-norm principal component analysis for robust subspace factorization. En *Proc. 23th Intl. Conf. on Machine Learning (ICML-06)*, páginas 281–288. 2006.

DYAR, M., CARMOSINO, M., SPEICHER, E., OZANNE, M., CLEGG, S. y WIENS, R. Comparison of partial least squares and lasso regression techniques as applied to laser-induced breakdown spectroscopy of geological samples. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 2012.

EKLUND, A., ANDERSSON, M. y KNUTSSON, H. fMRI analysis on the GPU - possibilities and challenges. *Computer Methods and Programs in Biomedicine*, vol. 105(2), páginas 145–161, 2012.

FOGEL, I. y SAGI, D. Gabor filters as texture discriminator. *Biological Cybernetics*, vol. 61(2), páginas 103–113, 1989.

FRANK, A. y ASUNCION, A. UCI machine learning repository. 2010.

FRIEDMAN, J., HASTIE, T., ROSSET, S., TIBSHIRANI, R. y ZHU, J. [consistency in boosting]: Discussion. *The Annals of Statistics*, vol. 32(1), páginas 102–107, 2004.

FRIEDMAN, J., HASTIE, T. y TIBSHIRANI, R. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

FU, Z., LU, G., TING, K. M. y ZHANG, D. A survey of audio-based music classification and annotation. *IEEE Trans. Multimedia*, vol. 13(2), páginas 303–319, 2011.

VAN GERVEN, M. A. J., CHAO, Z. C. y HESKES, T. On the decoding of intracranial data using sparse orthonormalized partial least squares. *Journal of Neural Engineering*, vol. 9(2), páginas 26017–26027, 2012.

VAN GERVEN, M. A. J. y HESKES, T. Sparse orthonormalized partial least squares. En *Proc. 22nd Benelux Conf. on Artificial Intelligence (BNAIC 2010)*. Luxembourg, 2010.

GILLIS, N. y GLINEUR, F. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *Neural computation*, vol. 24(4), páginas 1085–1105, 2012.

GOLUB, G. H. y VAN LOAN, C. F. *Matrix computations*, vol. 3. JHU Press, 2012.

GUO, Z., ZHANG, L. y ZHANG, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.*, vol. 19(6), páginas 1657–1663, 2010.

GUYON, I. y ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, páginas 1157–1182, 2003.

GUYON, I., GUNN, S., NIKRAVESH, M. y ZADEH, L., editores. *Feature Extraction, Foundations and Applications*. Studies in Fuzziness and Soft Computing. Springer, 2006.

GUYON, I., WESTON, J., BARNHILL, S. y VAPNIK, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, vol. 46(1-3), páginas 389–422, 2002.

HAN, J. y MA, K.-K. Rotation-invariant and scale-invariant Gabor features for texture image retrieval. *Image and Vision Computing*, vol. 25(9), páginas 1474–1481, 2007.

HANSEN, L. K. Multivariate strategies in functional magnetic resonance imaging. *Brain and Language*, vol. 102(2), páginas 186–191, 2007.

HARDOON, D., MOURAO-MIRANDA, J., BRAMMER, M. y SHAWE-TAYLOR, J. Unsupervised analysis of fMRI data using kernel canonical correlation. *NeuroImage*, vol. 37(4), páginas 1250–1259, 2007.

HARDOON, D. y SHAWE-TAYLOR, J. Sparse canonical correlation analysis. *Machine Learning*, vol. 83(3), páginas 331–353, 2011.

HASTIE, T., TAYLOR, J., TIBSHIRANI, R. y WALTHER, G. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics 1*, páginas 1–29, 2007.

HOEGAERTS, L., SUYKENS, J. A. K., VANDEWALLE, J. y DE MOOR, B. Primal space sparse kernel partial least squares regression for large scale problems. En *Proc. IEEE Intl. Joint Conf. on Neural Networks (IJCNN)*, páginas 561–566. IEEE, Budapest, Hungary, 2004.

HOTELLING, H. Relations between two sets of variates. *Biometrika*, vol. 28, páginas 321–377, 1936.

HUANG, D. y DE LA TORRE, F. Bilinear kernel reduced rank regression for facial expression synthesis. En *Proc. European Conf. Computer Vision (ECCV)*, páginas 364–377. Springer, 2010.

JIA, Y., NIE, F. y ZHANG, C. Trace ratio problem revisited. *IEEE Trans. Neural Networks*, vol. 20(4), páginas 729–735, 2009.

KAMARAINEN, J.-K. *Feature extraction using Gabor filters*. Tesis Doctoral, Lappeenranta University of Technology, 2003.

KIM, J. y PARK, H. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. En *Proc. 8th IEEE Intl. Conf. on Data Mining (ICDM'08)*, páginas 353–362. IEEE, Pisa, Italy, 2008.

KOHAVI, R. y JOHN, G. Wrappers for feature selection. *Artificial Intelligence*, vol. 97(1-2), páginas 273–324, 1997.

LAI, P. L. y FYFE, C. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, vol. 10(5), páginas 365–377, 2000.

LEE, D. D. y SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, vol. 401(6755), páginas 788–791, 1999.

LI, W., MAO, K., ZHANG, H. y CHAI, T. Designing compact Gabor filter banks for efficient texture feature extraction. En *Proc. 11th Intl. Conf. on Control Automation Robotics & Vision (ICARCV)*, páginas 1193–1197. Singapore, 2010.

LIU, H. y MOTODA, H. *Feature Selection for Knowledge Discover and data Mining*. Kluwer Academic Publishers, Norwell, MA, 1998.

M. MOMMA, K. B. Sparse kernel partial least squares regression. En *Proc. Conf. on Learning Theory (COLT 2003)*, páginas 216–230. Washington, DC, USA, 2003.

MACKEY, L. W. Deflation methods for sparse PCA. En *Advances in Neural Information Processing Systems 21*, páginas 1017–1024. Curran Associates, Inc., 2009.

MANDEL, M. I., POLINER, G. E. y ELLIS, D. P. Support vector machine active learning for music retrieval. *Multimedia systems*, vol. 12(1), páginas 3–13, 2006.

MARDIA, K. V., KENT, J. T. y BIBBY, J. M. *Multivariate analysis*. Academic press, 1980.

McKinney, M. F. y Breebaart, J. Features for audio and music classification. En *Proc. Intl. Symposium on Music Information Retrieval (ISMIR)*, vol. 3, páginas 151–158. Baltimore, Maryland, USA, 2003.

Meng, A., Ahrendt, P., Larsen, J. y Hansen, L. K. Temporal feature integration for music genre classification. *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15(5), páginas 1654–1664, 2007.

Meng, A., Larsen, J. y Hansen, L. K. *Temporal feature integration for music organisation*. Tesis Doctoral, Technical University of Denmark, Danmarks Tekniske Universitet, Department of Informatics and Mathematical Modeling, Institut for Informatik og Matematisk Modellering, Lyngby, Denmark, 2006.

Meng, A. y Shawe-Taylor, J. An investigation of feature models for music genre classification using the support vector classifier. En *Proc. 6th Intl. Conf. on Music Information Retrieval (ISMIR)*, páginas 604–609. London, UK, 2005.

Ngo, T. T., Bellalij, M. y Saad, Y. The trace ratio optimization problem. *SIAM Rev.*, vol. 54(3), páginas 545–569, 2012.

Nie, F., Huang, H., Cai, X. y Ding, C. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. En *Advances in Neural Information Processing Systems 23*, páginas 1813–1821. The MIT Press, 2010.

Oja, E. y Plumbley, M. Blind separation of positive sources using nonnegative PCA. En *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation*. Nara, Japan, 2003.

Ojala, T., Pietikainen, M. y Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 24(7), páginas 971–987, 2002.

Pampalk, E. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. Tesis Doctoral, Vienna University of Technology, Vienna, Austria, 2006.

Pauca, V. P., Piper, J. y Plemmons, R. J. Nonnegative matrix factorization for spectral data analysis. *Linear algebra and its applications*, vol. 416(1), páginas 29–47, 2006.

Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2(11), páginas 559–572, 1901a.

PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, vol. 2(6), páginas 559–572, 1901b.

RAKOTOMAMONJY, A. Variable selection using SVM-based criteria. *Journal of Machine Learning Research 3*, páginas 1357–1370, 2003.

REINSEL, G. C. y VELU, R. P. *Multivariate reduced-rank regression: theory and applications*. Springer New York, 1998.

ROSIPAL, R. y TREJO, L. J. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research 2*, páginas 97–123, 2002.

ROWEIS, S. y BRODY, C. Linear heteroencoders. Informe Técnico 1999-002, Gatsby Computational Neuroscience Unit, 1999.

SAMPSON, P. D., STREISSGUTH, A. P., BARR, H. M. y BOOKSTEIN, F. L. Neurobehavioral effects of prenatal alcohol: Part II. partial least squares analysis. *Neurotoxicology and teratology*, vol. 11(5), páginas 477–491, 1989.

SCARINGELLA, N., ZOIA, G. y MLYNEK, D. Automatic genre classification of music content: a survey. *IEEE Signal Process. Mag.*, vol. 23(2), páginas 133–141, 2006.

SCHOELKOPF, B. y SMOLA, A. *Learning with kernels*. MIT Press, 2002.

SCHOLKOPF, B., SMOLA, A. y MULLER, K.-R. Non linear component analysis as kernel eigenvalue problem. *Neural Computation*, vol. 10(5), páginas 1299–1319, 1998.

SCHÖNEMANN, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, vol. 31(1), páginas 1–10, 1966.

SEARLE, S. R. *Matrix algebra useful for statistics*. John Wiley and Sons, 1982.

SEUNG, D. y LEE, L. Algorithms for non-negative matrix factorization. En *Advances in neural information processing systems 13*, páginas 556–562. The MIT Press, 2001.

SHAWE-TAYLOR, J. y CRISTIANINI, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

SHI, X., YANG, Y., GUO, Z. y LAI, Z. Face recognition by sparse discriminant analysis via joint $L_{2,1}$-norm minimization. *Pattern Recognition*, vol. 47(7), páginas 2447–2453, 2014.

SIGG, C., FISCHER, B., OMMER, B., ROTH, V. y BUHMANN, J. Nonnegative CCA for audiovisual source separation. En *Proc. IEEE Intl. Workshop on Machine Learning for Signal Processing*, páginas 253–258. Thessaloniki, Greece, 2007.

SMARAGDIS, P. y BROWN, J. C. Non-negative matrix factorization for polyphonic music transcription. En *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, páginas 177–180. IEEE, New Paltz, NY, 2003.

STURM, B. L. On music genre classification via compressive sampling. En *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME 2013)*. San Jose, USA, 2013.

SU, A. I., WELSH, J. B., SAPINOSO, L. M., KERN, S. G., DIMITROV, P., LAPP, H., SCHULTZ, P. G., POWELL, S. M., MOSKALUK, C. A., FRIERSON, H. F. ET AL. Molecular classification of human carcinomas by use of gene expression signatures. *Cancer research*, vol. 61(20), páginas 7388–7393, 2001.

SUN, L., JI, S., YU, S. y YE, J. On the equivalence between canonical correlation analysis and orthonormalized partial least squares. En *Proc. 21st Intl. Joint Conf. on Artificial Intelligence (IJCAI-09)*, páginas 1230–1235. Pasadena, California, USA, 2009.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, vol. 58(1), páginas 267–288, 1994.

DE LA TORRE, F. A least-squares framework for component analysis. *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 34(6), páginas 1041–1055, 2012.

TRYGG, J. y WOLD, S. Orthogonal projections to latent structures (O-PLS). *Journal of chemometrics*, vol. 16(3), páginas 119–128, 2002.

TURNER, M. R. Texture discrimination by Gabor functions. *Biological Cybernetics*, vol. 55(2-3), páginas 71–82, 1986.

VAN BENTHEM, M. H. y KEENAN, M. R. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of chemometrics*, vol. 18(10), páginas 441–450, 2004.

VIRTANEN, T. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15(3), páginas 1066–1074, 2007.

WEGELIN, J. A. A survey of partial least squares (PLS) methods, with emphasis on the two-block case. Informe Técnico 371, Department of Statistics, University of Washington, Seattle, 2000.

WESTON, J., MUKHERJEE, S., CHAPELLE, O., PONTIL, M., POGGIO, T. y VAPNIK, V. Feature selection for SVMs. En *Advances in Neural Information Processing Systems 13*, páginas 668–674. MIT Press, 2001.

WESTON, J., PEREZ-CRUZ, F., BOUSQUET, O., CHAPELLE, O., ELISSEEFF, A. y SCHOLKOPF, B. Feature selection and transduction for prediction of molecular bioactivity for drug design. *Bioinformatics*, vol. 19(6), páginas 764–771, 2003.

WHITE, P. A. The computation of eigenvalues and eigenvectors of a matrix. *Journal of the Society for Industrial and Applied Mathematics*, vol. 6(4), páginas 393–437, 1958.

WILLIAMS, C. y SEEGER, M. Using the nyström method to speed up kernel machines. En *Advances in Neural Information Processing Systems 13*, páginas 682–688. MIT press, Cambridge, MA, 2001.

WOLD, H. Estimation of principal components and related models by iterative least squares. En *Multivariate Analysis*, páginas 391–420. Academic Press, 1966a.

WOLD, H. Non-linear estimation by iterative least squares procedures. En *Research Papers in Statistics*, páginas 411–444. Wiley, 1966b.

WOLD, S., ALBANO, C., DUNN, W. J., EDLUND, U., ESBENSEN, K., GELADI, P., HELLBERG, S., JOHANSSON, E., LINDBERG, W. y SJOSTROM, M. Multivariate data analysis in chemistry. En *Chemometrics, Mathematics and Statistics in Chemistry*, página 17. Reidel Publishing Company, 1984.

WORSLEY, K. J., POLINE, J. B., FRISTON, K. J. y EVANS, A. C. Characterizing the response of PET and fMRI data using multivariate linear models. *NeuroImage*, vol. 6(4), páginas 305–319, 1996.

XIANG, Z. J. y RAMADGE, P. J. Fast lasso screening tests based on correlations. En *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, páginas 2137–2140. IEEE, Kyoto, Japan, 2012.

YAMANISHI, Y., VERT, J., NAKAYA, A. y KANEHISA, M. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, vol. 19(suppl 1), páginas i323–i330, 2003.

YANG, K., CAI, Z., LI, J. y LIN, G. A stable gene selection in microarray data analysis. *BMC Bioinformatics*, vol. 7(1), página 228, 2006.

YANG, T., LI, Y.-F., MAHDAVI, M., JIN, R. y ZHOU, Z.-H. Nyström method vs random fourier features: A theoretical and empirical comparison.

En *Advances in Neural Information Processing Systems 25*, páginas 476–484. Curran Associates, Inc., 2012.

YUAN, G. X., CHANG, K. W., HSIEH, C. J. y LIN, C. J. A comparison of optimization methods and software for large-scale L1-regularized linear classification. *Journal of Machine Learning Research 11*, páginas 3183–3234, 2010.

YUAN, M. y LIN, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 68(1), páginas 49–67, 2006.

YUAN, Z. y OJA, E. Projective nonnegative matrix factorization for image compression and feature extraction. En *Proc. 14th Scandinavian Conf. Image Analysis (SCIA 2005)*, páginas 333–342. Joensuu, Finland, 2005.

ZHENG, W., ZHOU, X., ZOU, C. y ZHAO, L. Facial expression recognition using kernel canonical correlation analysis (KCCA). *IEEE Trans. Neural Networks*, vol. 17(1), páginas 233–238, 2006.

ZOU, H., HASTIE, T. y TIBSHIRANI, R. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, vol. 15(2), páginas 265–286, 2006.

# Índice alfabético

# Lista de acrónimos

| | |
|---|---|
| **CCA** | "Canonical Correlation Analysis"<br>(Análisis de Correlaciones Canónicas) |
| **CV** | "Cross-Validation"<br>(Validación Cruzada) |
| **EVD** | "EigenValue Decomposition"<br>(Problema de Autovalores Estándar) |
| **fMRI** | "functional Magnetic Resonance Imaging"<br>(Resonancia Magnética funcional) |
| **GEV** | "Generalized EigenValue decomposition"<br>(Problema de Autovalores Generalizado) |
| **GF** | "Gabor Filtering"<br>(Filtrado de Gabor) |
| **KCCA** | "Kernel Canonical Correlation Analysis"<br>(Análisis de Correlaciones Canónicas Kernel) |
| **KMVA** | "Kernel MultiVariate Analysis"<br>(Análisis Multivariante Kernel) |
| **KOPLS** | "Kernel Orthonormalized Partial Least Squares"<br>(Mínimos Cuadrados Parciales Ortonormalizado Kernel) |
| **KPCA** | "Kernel Principal Component Analysis"<br>(Análisis de Componentes Principales Kernel) |
| **KPLS** | "Kernel Partial Least Squares"<br>(Mínimos Cuadrados Parciales Kernel) |
| **LBP** | "Local Binary Pattern"<br>(Patrón Binario Local) |
| **LASSO** | "Least Absolute Shrinkage and Selection Operator"<br>(Reducción Mínima Absoluto y Operador de Selección) |

**LS** "Least Squares"
(Mínimos Cuadrados)

**MFCC** "Mel Frequency Cepstral Coefficients"
(Coeficientes Cepstrales en las Frecuencias de Mel)

**MIR** "Music Information Retrieval"
(Recuperación de Información Musical)

**ML** "Machine Learning"
(Aprendizaje Máquina)

**MLR** "MultiLinear Regression"
(Regresión MultiLineal)

**MSE** "Mean Squared Error"
(Error Cuadrático Medio)

**MU** "Multiplicative Updating rule"
(regla de Actualización Multiplicativa)

**MVA** "MultiVariate Analysis"
(Análisis Multivariante)

**NMF** "Non-Negative Matrix Factorization"
(Factorización No Negativa de Matrices)

**NNDSVD** "Non-Negative Double Singular Value Decomposition"
(Doble Descomposición de Valores Singulares No Negativa)

**NOPLS** "Non-Negative Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado No Negativo)

**NZ** "Non-Zero coefficients"
(coeficientes No Nulos)

**OA** "Overall Accuracy"
(Precisión Total)

**OPLS** "Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado)

**O-PLS** "Orthogonal Projections to Latent Structures"
(Proyecciones Ortogonales sobre Estructuras Latentes)

**PCA** "Principal Component Analysis"
(Análisis de Componentes Principales)

**PLS** "Partial Least Squares"
(Mínimos Cuadrados Parciales)

**POPLS** "Positive Constrained Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado con restricciones de Positividad)

**P-NOPLS** "Procrustes Non-Negative Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado No Negativo usando Procrustes)

**P-SOPLS** "Procrustes Sparse Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado Disperso usando Procrustes)

**RFS** "Robust Feature Selection"
(Selección Robusta de Características)

**RKHS** "Reproducing Kernel Hilbert Space"
(Espacio de Hilbert Generado por Funciones Kernel)

**rKOPLS** "reduced Kernel Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado Kernel reducido)

**RRR** "Reduced-Rank Regression"
(Regresión de Rango Reducido)

**SOPLS** "Sparse Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado Disperso)

**SrKOPLS** "Sparse reduced Kernel Orthonormalized Partial Least Squares"
(Mínimos Cuadrados Parciales Ortonormalizado Kernel reducido y Disperso)

**SR** "Sparsity Rate"
(Tasa de Dispersión)

**SRRR** "Sparse Reduced-Rank Regression"
(Regresión de Rango Reducido Disperso)

**SVD** "Singular Value Decomposition"
(Descomposición en Valores Singulares)

**SVM** "Support Vector Machine"
(Máquina de Vectores Soporte)