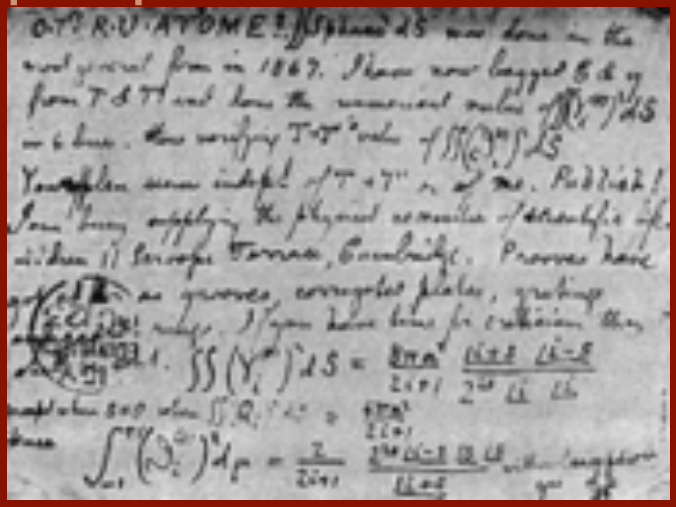
Tesis Doctoral
Ingeniería de Telecomunicación

Sobre las propiedades discriminativas del análisis en componentes principales basado en la norma L1



Autor: José Luis Camargo Olivares
Director: Rubén Martín Clemente

Dpto. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Tesis Doctoral
Ingeniería de Telecomunicación

Sobre las propiedades discriminativas del análisis en componentes principales basado en la norma L1

Autor:

José Luis Camargo Olivares

Director:

Rubén Martín Clemente

Profesor Titular de Universidad

Dpto. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

2021

Tesis Doctoral: Sobre las propiedades discriminativas del análisis en componentes principales basado en la norma L1

Autor: José Luis Camargo Olivares
Director: Rubén Martín Clemente

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

A mis padres y maestros.

Resumen

El análisis de componentes principales (PCA, del inglés *Principal Component Analysis*) basado en la norma L1 es una técnica cada vez más popular para el análisis de datos multivariantes. Un dato multivariante es aquél que se puede asociar a un punto en un espacio de p dimensiones; el conjunto de datos formará, por tanto, una nube de puntos. Tradicionalmente, PCA ha tratado de encontrar las direcciones en las que dicha nube se extiende por el espacio. Se ha utilizado para ello, como idea intuitiva, que en esas direcciones las proyecciones de los puntos han de tener una gran *varianza*: puede haber proyecciones pequeñas; pero, sobre todo, también las habrá de gran tamaño. Este criterio es muy efectivo y se ha mostrado útil en una gran variedad de aplicaciones. Sin embargo, tiene el inconveniente de que la *varianza* es un estadístico poco robusto: si los datos están contaminados con *valores atípicos* o, utilizando un término más habitual, *outliers*, las estimaciones de la varianza tendrán un gran error. Como solución, se ha propuesto sustituir la varianza por el *promedio del valor absoluto de las proyecciones*. La técnica resultante es lo que se ha dado en llamar PCA basado en la norma L1 (la norma L1 es la suma de los valores absolutos de los elementos de un vector) o L1-PCA. Se ha demostrado que, procediendo de esta forma, se consiguen algoritmos muy robustos.

El grupo de investigación en el que se ha realizado esta Tesis lleva varios años dedicado al estudio de L1-PCA. En particular, ha sido capaz de probar que, bajo diferentes hipótesis generales, L1-PCA puede obtener

resultados equivalentes tanto al análisis de componentes independientes (ICA, del inglés *Independent Component Analysis*) como al análisis lineal discriminante de Fisher. Estos resultados han sido publicados en revistas de primer nivel.

En la presente Tesis se continúa con esta línea de investigación. Se demostrará, en concreto, que también existe un vínculo entre L1-PCA y la transformada de Fukunaga-Koontz (FKT, del inglés *Fukunaga-Koontz transform*), a la que también se conoce como técnica de los *common spatial patterns* (CSP) en el tratamiento del electroencefalograma. La FKT es una técnica de pre-procesamiento utilizada en problemas de clasificación binaria donde las clases se *superponen*. Proyecta las clases en direcciones en las que la potencia o varianza de una clase es mucho mayor que la de la otra. De esta forma, la posterior clasificación es mucho más sencilla.

En su formulación original, L1-PCA proyecta los datos de manera que maximiza, en promedio, el valor absoluto de las proyecciones. De esta forma, consigue resultados similares al análisis de componentes principales «clásico». Ahora bien, manteniendo el valor absoluto como función objetivo, pero cambiando «maximizar» por «minimizar», mostraremos a lo largo de esta Tesis que L1-PCA proporciona un resultado equivalente al que se obtiene mediante la transformada de Fukunaga-Koontz. Se proporcionará una prueba rigurosa de este resultado para el caso de tener poblaciones gaussianas multivariantes con media cero y diferentes matrices de covarianza. Para datos no gaussianos, la verificación de este resultado será experimental utilizando bases de datos de imágenes reales. El único requisito que se exige para ello, o hipótesis de partida, es que los datos en bruto (sin procesar) hayan sido «blanqueados», con un pre-procesamiento previo, a fin de eliminar la estructura de su covarianza.

La importancia práctica de este resultado es que la FKT estándar es una técnica *supervisada*, es decir, para estimar los parámetros de la transformación, requiere un conjunto de datos de entrenamiento pertenecientes a cada una de las clases correctamente etiquetados. Por el contrario, minimizar el valor absoluto puede llevarse a cabo de manera totalmente *no supervisada*, haciendo innecesarios por ello los datos de entrenamiento. De esta forma, se ofrece una alternativa completamente novedosa para el cálculo de la

FKT. La importancia teórica del resultado expuesto, por otra parte, reside en que relaciona técnicas muy dispares, la FKT y L1-PCA, lo que nos va a permitir reinterpretar el valor absoluto como un criterio para la extracción de características en problemas de clasificación binaria. Esto abre nuevas líneas de investigación en el área del aprendizaje automático o «machine learning».

Esta Tesis, finalmente, ha dado lugar a la siguiente publicación en revista:

- José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm unsupervised Fukunaga-Koontz transform», *Signal Processing*, vol. 182, mayo 2021, *https://doi.org/10.1016/j.sigpro.2020.107942*.

así como a las siguientes comunicaciones a congresos:

- José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «Unsupervised Classification of Zero-Mean Data Based on L1-Norm Principal Component Analysis», en: H. Sharma, M. Gupta, G. Tomar, W. Lipo (eds) «Communication and Intelligent Systems», *Lecture Notes in Networks and Systems*, vol 204, pp. 967–973, 2021, Springer, Singapur. *https://doi.org/10.1007/978-981-16-1089-9_75*.
- Rubén Martín-Clemente, Vicente Zarzoso, José Luís Camargo, «On the discriminative properties of Principal Component Analysis based on L1-norm» *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dic. 2020, pp. 1673-1676, *https://doi.org/doi:10.1109/CSCI51800.2020.00308*.
- Rubén Martín-Clemente, Jose Luis Camargo, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm based PCA for unsupervised classification», *1st International Electronic Conference on Applied Sciences*, Nov. 2020, *https://doi.org/doi:10.3390/ASEC2020-07639*.

Abstract

Principal Component Analysis (PCA) based on the L1 norm is an increasingly popular technique for multivariate data analysis. A multivariate data is one that can be associated with a point in a space of p dimensions; the data set will therefore form a point cloud. Traditionally, PCA has tried to find the directions in which such a cloud extends through space. Intuitively, in these directions the projections of the points must have a large *variance*: there may be small projections; but, above all, there will also be large ones. This criterion is very effective and has proven useful in a wide variety of applications. However, it has the drawback that the *variance* is not a robust statistic: if the data are contaminated with *outliers*, the variance estimates will have a large error. As a solution, it has been proposed to replace the variance by the *average of the absolute value of the projections*. The resulting technique is what has been called L1-norm based PCA (the L1-norm is the sum of the absolute values of the elements of a vector) or L1-PCA. It has been shown that, as expected, L1-PCA is very robust against outliers.

Our research group has been devoted to the study of L1-PCA for several years. In particular, it has been able to prove that, under different general hypotheses, L1-PCA can obtain results equivalent to both Independent Component Analysis (ICA) and Fisher's linear discriminant analysis. These results have been published in leading journals.

The present Thesis continues with this line of research. In particular, it will be shown that there is also a link between L1-PCA and the Fukunaga-

Koontz transform (FKT), which is also known as the common spatial patterns (CSP) technique in EEG processing. The FKT is a pre-processing technique used in binary classification problems where classes *overlap*. It projects the classes in directions where the power or variance of one class is much greater than that of the other. This makes the subsequent classification much easier.

In its original formulation, L1-PCA projects the data in a way that maximizes, on average, the absolute value of the projections. Henceforth, it achieves results similar to «classical» principal component analysis. Now, keeping the absolute value as the objective function, but changing «maximize» to «minimize», we will show in this Thesis that L1-PCA provides a result equivalent to that obtained by the Fukunaga-Koontz transform. A rigorous proof of this result will be provided for the case of having multivariate Gaussian populations with zero mean and different covariance matrices. For non-Gaussian data, the verification of this result will be experimental using real image databases. The only requirement for this, or starting hypothesis, is that the raw data have been «withened», with prior pre-processing, in order to remove their covariance structure.

The practical significance of this result is that the standard FKT is a *supervised* technique, i.e., to estimate the parameters of the transformation, it requires a set of training data belonging to each of the correctly labeled classes. In contrast, minimizing the absolute value can be carried out in a fully unsupervised manner, thus making training data unnecessary. Thus, a completely novel alternative for the calculation of the FKT is offered. The theoretical importance of the above result, on the other hand, lies in the fact that it relates very different techniques, FKT and L1-PCA, which will allow us to reinterpret the absolute value as a criterion for feature extraction in binary classification problems. This opens new lines of research in the area of machine learning or «machine learning».

This Thesis, finally, has given rise to the following journal publication:

- José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm unsupervised Fukunaga-Koontz

transform», *Signal Processing*, vol. 182, mayo 2021, *https://doi.org/10.1016/j.sigpro.2020.107942*.

as well as to the following conference papers:

- José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «Unsupervised Classification of Zero-Mean Data Based on L1-Norm Principal Component Analysis», en: H. Sharma, M. Gupta, G. Tomar, W. Lipo (eds) «Communication and Intelligent Systems», *Lecture Notes in Networks and Systems*, vol 204, pp. 967–973, 2021, Springer, Singapur. *https://doi.org/10.1007/978-981-16-1089-9_75*.
- Rubén Martín-Clemente, Vicente Zarzoso, José Luís Camargo, «On the discriminative properties of Principal Component Analysis based on L1-norm» *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dic. 2020, pp. 1673-1676, *https://doi.org/doi:10.1109/CSCI51800.2020.00308*.
- Rubén Martín-Clemente, Jose Luis Camargo, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm based PCA for unsupervised classification», *1st International Electronic Conference on Applied Sciences*, Nov. 2020, *https://doi.org/doi:10.3390/ASEC2020-07639*.

Índice Abreviado

Índice

1 Introducción

1.1 El aprendizaje automático

Esta Tesis Doctoral se desarrolla en el campo del aprendizaje automático o «machine learning» [34, 52, 93]. En un sentido amplio, la inteligencia artificial se ocupa de la interacción de las máquinas con el entorno en que se ubican. El objetivo último es obtener ordenadores que piensen como las personas, actúen como ellas y, finalmente, sean capaces de solucionar problemas con mayor rapidez y eficacia que el ser humano. El aprendizaje automático, en concreto, diseña máquinas capaces de aprender de su entorno y tomar decisiones de forma autónoma. Contrasta con la forma de operar tradicional, en la que un experto humano analiza el problema y, después, programa las acciones que ha de llevar a cabo la máquina en cada una de las situaciones.

«Inteligencia artificial» y «aprendizaje automático» se usan a menudo como términos sinónimos. Hace pocos años, además, ha aparecido el concepto de «aprendizaje profundo» o «deep learning» [12, 36, 43, 83]. Podemos pensar en una serie de círculos concéntricos (ver la Figura 1.1), siendo la inteligencia artificial el que contiene a los demás, al ser la idea primitiva. Seguidamente tenemos el aprendizaje automático, que apareció más tarde, pero ha revolucionado el concepto de inteligencia artificial gracias a los resultados espectaculares que permite obtener. Finalmente, el «deep learning»

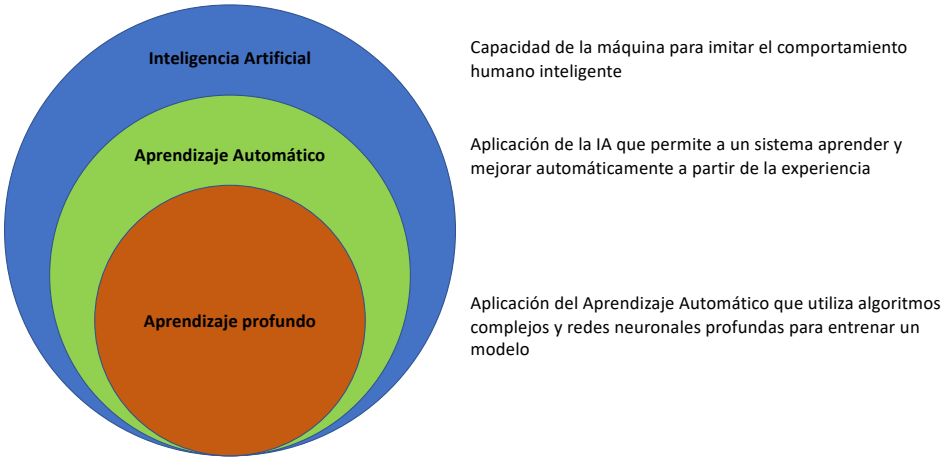


Figura 1.1 Jerarquía de la inteligencia artificial.

aprovecha la creciente capacidad de cálculo de los ordenadores para llevar las técnicas de aprendizaje automático a un nivel de eficacia similar, en la resolución de algunas tareas, a la del ser humano.

El aprendizaje automático sería por tanto el subconjunto de la inteligencia artificial que diseña máquinas capaces de aprender por sí mismas a partir de los datos que reciben. Para ello, los algoritmos se van adaptando a la información que procesan: a partir de un conjunto inicial de datos, las técnicas de aprendizaje automático intentan determinar patrones de ocurrencia, relaciones matemáticas entre las distintas variables, etc. Si este conjunto inicial representa con suficiente precisión las características de todo el espacio muestral, el algoritmo podrá realizar por tanto predicciones sobre otros datos aún no observados a partir de dichas relaciones y patrones [12, 36].

El término «aprendizaje automático» fue adoptado en 1950 por Arthur L. Samuel, investigador en IBM, que escribió un programa capaz de ganar jugando a las damas a los mejores expertos estadounidenses [94]. Desde entonces esta disciplina no ha dejado de crecer hasta la actualidad, donde el término no deja de escucharse en casi cualquier ámbito: así, tenemos el sistema de procesado de lenguaje natural propuesto por Apple para su asistente Siri, el procesamiento inteligente de imágenes de Google Photos, basado en redes neuronales convolucionales, o el algoritmo de recomen-

dación de series de Netflix, además de numerosas técnicas de ayuda al diagnóstico terapéutico o para la predicción de series temporales [12, 36]. La popularización del aprendizaje automático está relacionada, por otra parte, con el progreso tecnológico de las últimas décadas: se estima que se envían alrededor de tres millones de correos electrónicos cada segundo, Google procesa alrededor de 24 Petabytes de datos diariamente, Facebook contabiliza que su red se usa 700 billones de minutos al mes ... Todo ello es muestra de la masiva generación de datos que caracteriza esta cuarta revolución industrial, donde el modelo de negocio se construye en torno a los a sistemas inteligentes, la «Internet of Things» o el «Cloud Computing». Todo ello permite presagiar un futuro dominado por las herramientas del «aprendizaje automático», que serán la base de un sinfín de aplicaciones en los años venideros [97].

Hay una gran variedad de técnicas de aprendizaje automático: redes neuronales, árboles de decisión, regresores o algoritmos de k-vecinos próximos. No obstante, no hay ninguna que sea eficaz en cualquier situación y para cualquier conjunto de datos. Dependiendo de la naturaleza propia de cada problema y de los resultados que se pretenda obtener, se tendrán que escoger unos algoritmos u otros para definir los modelos. Será a través de la posterior validación como nos aseguremos que el rendimiento es el apropiado. En aprendizaje automático, por otra parte, suelen distinguirse dos clases de técnicas, el aprendizaje *supervisado* y el *no supervisado*. A continuación se revisan ambas categorías.

1.1.1 El aprendizaje supervisado

En aprendizaje supervisado [8, 108] la máquina aprende a base de ejemplos o, de forma más precisa, de secuencias de entrenamiento. Éstas son parejas $(\boldsymbol{x}, y)$ formadas por un determinado dato $\boldsymbol{x}$ de entrada y la acción y que ha de realizarse en respuesta. Matemáticamente, podríamos usar el siguiente modelo, utilizando para ello una función $f(\cdot)$ que es, en principio, desconocida:

$$y = f(\boldsymbol{x}).$$

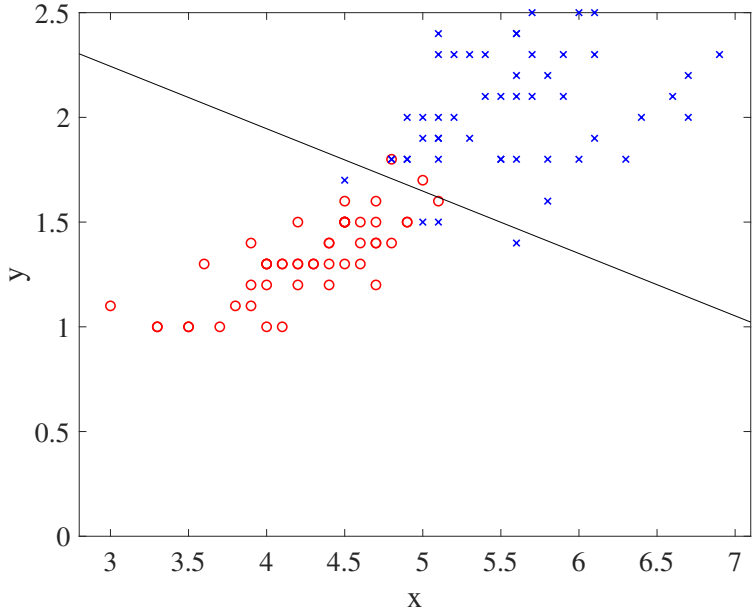


Figura 1.2 Ejemplo de problema de clasificación.

Se desea que, partiendo de la observación de muchas parejas $(\boldsymbol{x}, y)$ de entrenamiento, el algoritmo encuentre una función $\hat{f}(\cdot)$ que aproxime $f(\cdot)$ con suficiente precisión. De esta manera, cuando se introduzca un nuevo conjunto de datos $\boldsymbol{x}$, la máquina podrá obtener de forma autónoma los resultados y que correspondan.

Típicamente, en el aprendizaje supervisado encontramos dos ramas:

1. *La clasificación.* En ella se pretende que cada observación sea asignada a un grupo entre varios los posibles [104]. En este caso, la variable y identifica al grupo al que se adscribe $\boldsymbol{x}$. Un ejemplo de clasificación podría ser la distinción entre dos razas de perros diferentes tales que, a partir de ciertos atributos, fuese posible distinguirlas. Una representación gráfica de ello podría representarse con un modelo como el indicado en la Figura 1.2, donde se pueden apreciar puntos de dos clases (azul y roja). La recta que divide el plano en dos regiones, asociadas de manera aproximada a cada clase, ha sido calculada mediante análisis lineal discriminante de Fisher [98].

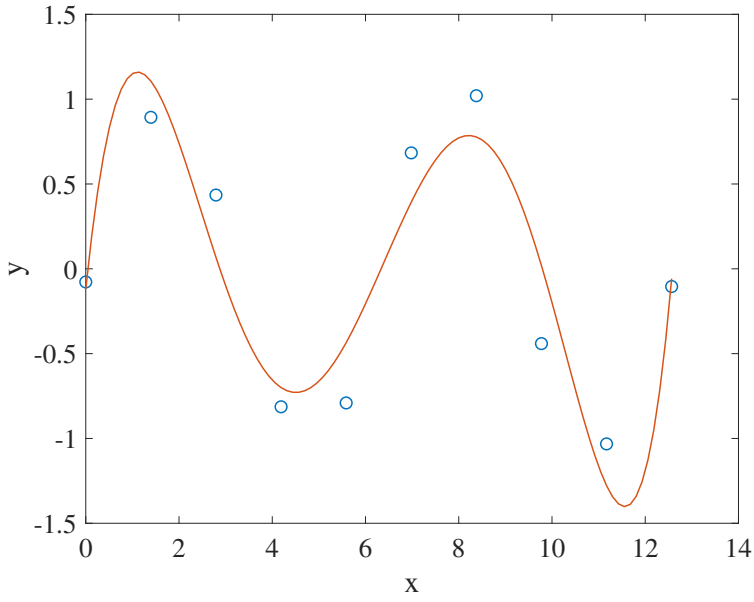


Figura 1.3 Ejemplo de problema de regresión. Aproximación de una serie de puntos con un polinomio de grado 6.

2. *La regresión.* En la regresión la variable resultado y puede tomar cualquier valor. Un ejemplo de regresión podría ser la estimación de la forma de onda de la función $y = \sin(2\pi x)$ a partir de una serie de puntos (x_i, y_i). En este caso, podría aproximarse la función por medio de polinomios, como en la Figura 1.3. Intuitivamente, el resultado obtenido con un polinomio de orden bajo será poco satisfactorio. Sin embargo, se obtendrá igualmente un gran error si el orden es demasiado alto. Este último problema es conocido como «overfitting» o sobreescalado. Éste puede acabar dando mucho peso a los valores atípicos («outliers») del conjunto de datos específico con el que se está trabajando, lo cual desestabiliza el modelo.

El aprendizaje supervisado, en cualquier caso, ha recibido mucha atención en los últimos años. Como consecuencia, disponemos de un gran número de técnicas de enorme eficacia (redes neuronales [42, 44, 98], máquinas de vectores soporte [13, 21, 25], bosques aleatorios o *random*

forests [5, 27, 46], arquitecturas *deep learning* [66, 74, 96], etc.) que permiten resolver una enorme variedad de problemas de clasificación y regresión o predicción.

1.1.2 El aprendizaje no supervisado

En aprendizaje no supervisado [8, 69], la máquina elabora, sin ayuda humana, un modelo de los datos que recibe. En otras palabras, aprende de una base de datos en la que no hay una variable «resultado», y, específica. El objetivo real es explorar y descubrir características o subgrupos dentro de la misma base de datos. Los modelos desarrollados pueden ser después utilizados, por ejemplo, para entender mejor la estructura interna de la información o para predecir nuevas observaciones. En general, el aprendizaje no supervisado tiene una amplia gama de aplicaciones. No obstante, las más comunes son dos:

1. *El análisis de grupos o conglomerados* («clustering» en inglés). Se desea agrupar entre sí los datos que tengan atributos similares. De esta forma, por ejemplo, se pueden buscar cepas de virus con características parecidas [1, 4, 64, 98].
2. *Asociaciones.* El objetivo en este caso es buscar relaciones significativas entre los datos introducidos para, por ejemplo, descubrir relaciones entre los síntomas que permiten diagnosticar la diabetes [1, 4, 64].

El aprendizaje *no* supervisado ha tenido un menor grado de desarrollo, en general, que el supervisado. Cabe destacar, como excepción, el gran número de excelentes algoritmos de análisis de grupos o conglomerados (en inglés, *clustering*) que existen en la literatura [1, 4, 64]. Estos permiten agrupar datos en función del parecido que tienen entre sí o de las propiedades que comparten. En particular, para realizar este tipo de análisis, utilizaremos los siguientes métodos en esta Tesis:

- *Los modelos de mezclas de Gaussianas* [12, 26]. Se trata de modelos probabilísticos que asumen que todas las muestras son generadas aleatoriamente a partir de la combinación de un número finito de

distribuciones Gaussianas. Es un método de agrupamiento «blando», en el a cada punto de datos se asocia un determinado nivel de pertenencia a cada clase. Este nivel coincide con la probabilidad de pertenecer al grupo y, por tanto, oscila entre 0 y 1. El método, que recibe el nombre de algoritmo EM («expectation-maximization» o «esperanza-maximización») podría resumirse de la siguiente manera [12]:

1. Inicializar las distribuciones de las Gaussianas, esto es, sus medias y covarianzas.
2. Agrupamiento «blando» de datos: es la fase de «expectativa» en la que todas las observaciones se asignarán a un grupo de acuerdo a su respectivo nivel de pertenencia. Lo que define a cada grupo es haber sido generado por una distribución Gaussiana diferente.
3. Re-estimación de los parámetros de las Gaussianas: es la fase de «maximización», en la que se usan las «expectativas» para recalcular las medias y varianzas de las distribuciones Gaussianas.
4. Evaluar la probabilidad de haber observado los datos a partir de la mezcla de Gaussianas. Cuanto mayor sea esta probabilidad, más verosimil es que la mezcla de Gaussianas se ajuste a la realidad. Por lo tanto, ésta es la función a maximizar.
5. Repetir desde el paso 2 hasta la convergencia.

- *El algoritmo de las k-medias* [12]. Este método es muy eficiente desde el punto de vista computacional, lo que explica su enorme popularidad. Sin embargo, solo es realmente apropiado para identificar grupos cuya distribución sea esférica. El algoritmo se basa en la hipótesis de que cuanto más cerca estén dos datos entre sí, mayores son las probabilidades de que pertenezcan a un mismo grupo. Con esta idea, los pasos que lleva a cabo el algoritmo se pueden resumir como sigue:
  1. Elegir k, el número de «clusters» que queremos definir.

2. Seleccionar aleatoriamente los centroides de cada grupo o «cluster».
3. Asignar cada punto de datos al centroide más cercano (utilizando la distancia euclídea).
4. Recalcular los nuevos centroides calculando la media de los puntos que pertenecen a cada grupo.
5. Volver al paso 3 hasta conseguir la convergencia.

La elección del número k de «clusters» determina las prestaciones del algoritmo. Para encontrar el valor óptimo se han desarrollado varias técnicas [12]. Cabe destacar, además, que bajo ciertas condiciones el algoritmo de las k-medias produce resultados equivalentes al método de la mezcla de Gaussianas [57].

Otras técnicas de aprendizaje no supervisado incluyen el análisis de componentes principales (ICA, de *independent component analysis*) [9, 18, 20, 51] o los numerosos métodos para la detección de *outliers* (datos atípicos o anómalos) [38, 71, 73].

1.2 Aportaciones de la Tesis

Esta Tesis es una contribución al área del aprendizaje *no supervisado*. En concreto, se profundizará en el estudio de una técnica *no supervisada* novedosa que está recibiendo cada vez más atención en la comunidad científica: el análisis de componentes principales (PCA) basado en la norma L1 o L1-PCA [58, 65, 75]. Se trata de una variante del análisis de componentes principales tradicional [54, 109] cuya característica más destacable es su mayor robustez frente a los valores atípicos (outliers) [65, 75, 77].

El grupo de investigación en el que se ha realizado esta Tesis ha sido capaz de probar en los últimos años que existe una fuerte conexión entre L1-PCA, el análisis de componentes independientes (ICA) y el análisis discriminante lineal (LDA, de *Linear Discriminant Analysis*) [79, 80].

En la presente Tesis se demostrará, además, que existe un vínculo entre la transformada de Fukunaga-Koontz (FKT, del inglés «Fukunaga-Koontz

transform») [33, 101, 111, 112] y L1-PCA. En su formulación tradicional, L1-PCA proyecta linealmente los datos de manera que maximiza, en promedio, el valor absoluto de las proyecciones. De esta forma, consigue resultados similares al análisis de componentes principales «clásico». Ahora bien, conservando el valor absoluto como función objetivo, pero cambiando «maximizar» por «minimizar», mostraremos en las siguientes páginas que L1-PCA proporciona un resultado equivalente al de la transformada de Fukunaga-Koontz de los datos. Este resultado ha sido publicado, asimismo, en [14], esto es,

> José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm unsupervised Fukunaga-Koontz transform», *Signal Processing*, vol. 182, mayo 2021, *https://doi.org/10.1016/j.sigpro.2020.107942*.

Se proporcionará una prueba rigurosa de este resultado para el caso de tener poblaciones gaussianas multivariantes con media cero y diferentes matrices de covarianza. Para datos no gaussianos, la verificación de este resultado será experimental. El único requisito que se exige, o hipótesis de partida, es que los datos en bruto (sin procesar) hayan sido «blanqueados», con un pre-procesamiento previo [61], a fin de eliminar la estructura de su covarianza.

La importancia teórica del resultado expuesto reside en que relaciona técnicas muy dispares, la FKT y L1-PCA, lo que nos va a permitir reinterpretar el valor absoluto como un criterio para la extracción de características en problemas de clasificación binaria. Esto abre nuevas líneas de investigación en el área del aprendizaje automático o «machine learning». Además de su interés teórico, este resultado también tiene relevancia práctica: la FKT estándar es una técnica supervisada, es decir, para estimar los parámetros de la transformación, requiere un conjunto de datos de entrenamiento pertenecientes a cada una de las clases correctamente etiquetados. Por el contrario, minimizar el valor absoluto puede llevarse a cabo de manera totalmente *no supervisada*, haciendo innecesarios por ello los datos de entrenamiento. De esta forma, se ofrece una alternativa completamente novedosa para el cálculo de la FKT.

El presente documento se organiza de la siguiente forma: en el Capítulo 2 se presentan los fundamentos de L1-PCA y su análisis con el análisis principal de componentes clásico. La relación entre L1-PCA y la FKT se desarrollará en el Capítulo 3. Se presentará un algoritmo desarrollado específicamente para la minimización de la norma L1 en el Capítulo 4. El Capítulo 5 se centrará en la validación experimental de los estudios teóricos realizados. Finalmente, las conclusiones del Capítulo 6 pondrán punto y final a esta Tesis.

1.3 Notación

Teniendo en cuenta las recomendaciones que aparecen en [40] y [59], en este documento utilizaremos la siguiente notación:

- Los escalares se denotarán con letra minúscula cursiva (x).
- Se utilizará letra minúscula, cursiva y negrita para los vectores ($\boldsymbol{x}$).
- Las matrices se denotarán con letra mayúscula, negrita y cursiva ($\boldsymbol{A}$).
- La traspuesta de un vector o una matriz se denotará con el símbolo $^\top$.
- Si no se dice otra cosa, los vectores se consideran vectores «columna».
- No obstante, como excepción a lo anterior, para las variables aleatorias no se utilizará la cursiva ($\mathrm{x}, \mathbf{x}$).

Así, por ejemplo, un vector aleatorio de dimensión p se expresará como

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_p \end{bmatrix}.$$

De igual modo, un valor observado de este vector se podrá escribir como

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}.$$

Se explican a continuación otros símbolos utilizados en el documento:

- $\mathbf{0}$ representa un vector de ceros.
- $\boldsymbol{I}$ es la matriz identidad.
- $(\cdot)^\top$ es el operador de trasposición.
- $\det(\cdot)$, $\mathrm{tr}(\cdot)$ denotan, respectivamente, el determinante y la traza de una matriz.
- $P(\mathscr{C}_i)$ es la probabilidad de observar una muestra de la clase $\mathscr{C}_i$
- $\mathbb{E}\{\cdot\}$ operador esperanza matemática.
- $\mathbb{E}\{\cdot \,|\, \mathscr{C}_i\}$ esperanza matemática de las observaciones que pertenecen a la clase $\mathscr{C}_i$.
- $f(\cdot \,|\, \mathscr{C}_i)$ representa la función de densidad de probabilidad de las muestras de la clase $\mathscr{C}_i$.
- $\boldsymbol{\mu}_i = \mathbb{E}\{X \,|\, \mathscr{C}_i\}$ valor medio de las observaciones de la clase i.
- $\boldsymbol{C}_i$ representa la covarianza de la clase i

2 El análisis de componentes principales

El análisis de componentes principales, que se abrevia como PCA, del inglés «principal component analysis», es una técnica de análisis de datos muy utilizada en la práctica [54, 109]. En este capítulo revisaremos los fundamentos de esta técnica y, además, presentaremos la variante de PCA conocida como «análisis de componentes principales basado en la norma L^1» (L1-PCA). Dicha variante será el objeto de estudio en esta Tesis.

2.1 Búsqueda de proyecciones

El análisis de componentes principales se puede considerar un tipo particular de técnica de «búsqueda de proyección» (o «projection pursuit»). Empezaremos, por lo tanto, explicando brevemente en qué consiste la «búsqueda de proyecciones» [32, 55, 84].

Sea

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_p \end{bmatrix}.$$

un vector aleatorio de p dimensiones. Se supondrá en adelante que hemos observado N valores concretos o realizaciones de $\mathbf{x}$, los cuales denotaremos como $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Por conveniencia matemática, y sin ninguna pérdida de generalidad, asumiremos que el promedio de los valores observados es igual a cero,

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n = \mathbf{0}. \tag{2.1}$$

En general, solo podemos representar gráficamente los datos si su dimensión (p) es menor o igual que tres. Cuando esto no ocurre, una posible solución, muy utilizada en la práctica, sería visualizar las variables originales $\mathrm{x}_1, \ldots, \mathrm{x}_p$ *por parejas*. Una segunda alternativa, que será la que adoptaremos en esta Tesis, «condensa» la información contenida en las variables originales, $\mathrm{x}_1, \ldots, \mathrm{x}_p$, en una nueva variable escalar z, combinación lineal de aquéllas. Todo el estudio posterior de los datos se llevará a cabo a partir de esta nueva variable escalar. Por ejemplo, sea

$$\mathrm{z} = \boldsymbol{a}^\top \mathbf{x} = \sum_{i=1}^{p} a_i \mathrm{x}_i. \tag{2.2}$$

Si $a_i = 1/p$, la combinación lineal resultante será precisamente el promedio de las variables aleatorias. Este valor medio, en muchas situaciones prácticas, será representativo del conjunto de datos. Otros valores de los coeficientes a_i, podrían dar lugar a interpretaciones igualmente útiles, como veremos repetidamente a lo largo de este Capítulo.

Las variables escalares obtenidas mediante combinaciones lineales se pueden, además, interpretar geométricamente a partir del concepto de «proyección». En concreto, supongamos que

$$\boldsymbol{a} = [a_1, \ldots, a_p]^\top$$

es un vector de longitud unidad. La proyección del punto $\boldsymbol{x}_n$ sobre la recta en la dirección del vector $\boldsymbol{a}$ se define matemáticamente [86] como

$$\boldsymbol{z}_n = z_n \boldsymbol{a} \tag{2.3}$$

siendo

$$z_n = \sum_{n=1}^{p} a_n x_n = \boldsymbol{a}^\top \boldsymbol{x}_n \tag{2.4}$$

el producto escalar de los dos vectores. Se puede comprobar fácilmente que:

1. El vector $\boldsymbol{z}_n$ está en la dirección de $\boldsymbol{a}$;
2. La combinación lineal z_n, cuya fórmula coincide con (2.2), *representa justamente la componente del vector de datos* $\boldsymbol{x}_n$ *sobre esa dirección*.

Para ilustrar estos conceptos, en la Figura 2.1 mostramos $N = 500$ valores de una variable aleatoria bidimensional ($p = 2$) que tiene distribución Gaussiana.

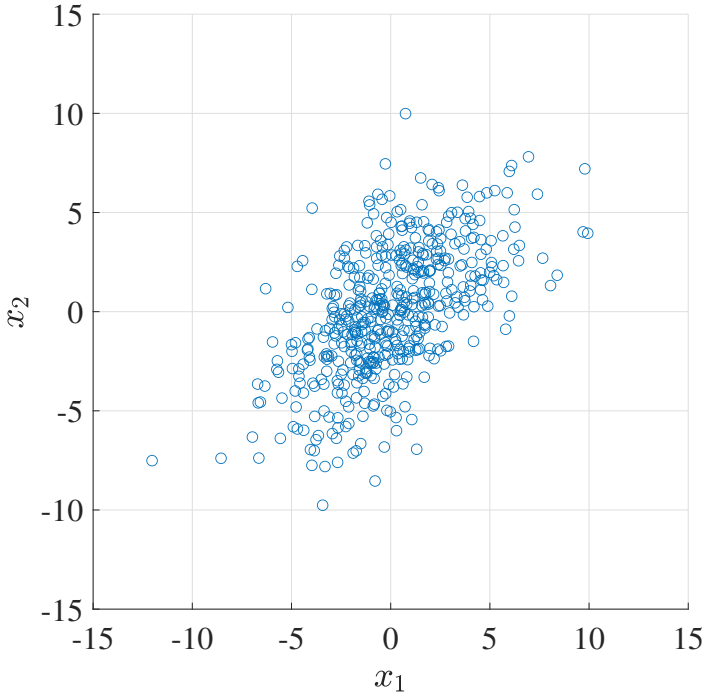


Figura 2.1 Diagrama de dispersión mostrando 500 puntos de una variable aleatoria Gaussiana bidimensional.

La Figura 2.2 muestra ahora la proyección de uno de los puntos de este diagrama de dispersión, al que llamaremos $\boldsymbol{x}_1$, sobre la recta en la dirección del vector

$$\boldsymbol{a} = \begin{bmatrix} \sqrt{2} \\ \sqrt{2} \end{bmatrix}.$$

También se muestra la distancia d_1 que separa al punto de su proyección. Se puede demostrar fácilmente que la línea que une $\boldsymbol{x}_1$ con $\boldsymbol{z}_1 = z_1\boldsymbol{a}$ es perpendicular a $\boldsymbol{a}$, es decir,

$$\boldsymbol{a}^\top(\boldsymbol{x}_1 - z_1\boldsymbol{a}) = 0,$$

y, por lo tanto, la proyección $\boldsymbol{z}_1$ resulta ser también el punto de la recta más cercano a $\boldsymbol{x}_1$.

La línea azul de la Figura 2.3 representa la función de densidad de probabilidad (estimada) de los coeficientes de proyección z, calculada a partir de $z_n = \boldsymbol{a}^\top \boldsymbol{x}_n$, con $n = 1, \ldots, 500$. Se muestra también en la misma Figura, con línea roja, la densidad de probabilidad de los coeficientes de proyección sobre la recta *perpendicular* a $\boldsymbol{a}$. Aunque en ambos casos las proyecciones tienen valor medio cero, observamos que los coeficientes de proyección sobre $\boldsymbol{a}$ (línea azul) alcanzan mayores amplitudes. Ello se puede explicar fácilmente: volviendo a la Figura 2.1 y a la Figura 2.2, vemos que los datos se distribuyen mayormente sobre la diagonal principal del primer cuadrante, es decir, en la dirección de $\boldsymbol{a}$. En cambio, la dispersión sobre la dirección perpendicular a $\boldsymbol{a}$ es relativamente pequeña. Por ello, las proyecciones sobre esta última dirección son menores.

Este ejemplo sugiere que podemos entender cómo se distribuye la nube de puntos en el espacio, incluso cuando no es posible representarla gráficamente, *sin más que proyectar los datos en muchas direcciones y observar el tamaño de las proyecciones*.

Una proyección bien seleccionada proporciona, por lo tanto, información muy valiosa sobre cómo se distribuye la nube de puntos. Esto resulta especialmente útil cuando se trabaja en espacios de más de tres dimensiones y no es posible, por tanto, visualizar la información de una forma directa. La pregunta que surge ahora es cómo escoger las direcciones de proyección

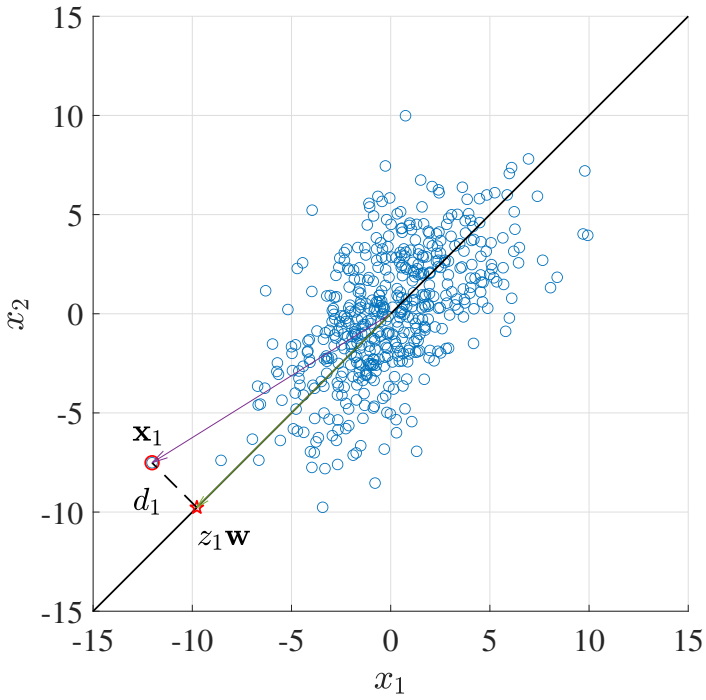


Figura 2.2 Proyección del punto marcado con el círculo rojo sobre la recta en línea negra. Se observa que la proyección viene dada por la intersección de la recta con el plano que contiene al punto y es perpendicular a ella.

más apropiadas. Éste es el objetivo precisamente de las técnicas de «búsqueda de proyección» o «projection pursuit». Típicamente, ejecutar una técnica de «projection pursuit» comprende los siguientes pasos [89]:

1. Escoger el criterio de proyección más adecuado para el tipo de problema que se desee resolver. Por ejemplo, este criterio podría ser maximizar la varianza o la curtosis de los coeficientes de proyección. La selección de este criterio será justamente lo que diferencie una técnica de otra: el análisis de componentes principales (PCA) [54], el análisis de componentes independientes (ICA) [50, 99] o el análisis lineal discriminante de Fisher (LDA) [15, 103], por poner algunos ejemplos, son todas técnicas de «projection pursuit». Sin embargo,

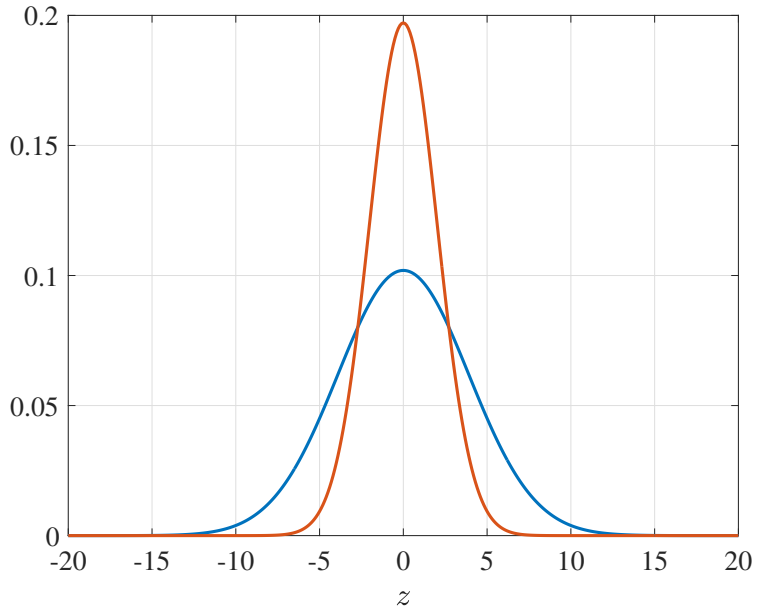


Figura 2.3 Funciones de densidad de probabilidad de las proyecciones de los puntos de la Figura 2.1 sobre la diagonal principal del primer cuadrante (azul) y la línea perpendicular a la misma (rojo).

pese a partir de un mismo principio básico, todas ellas producirán resultados completamente distintos al utilizar criterios diferentes.

2. Encontrar la dirección que maximiza o, en algunos casos, minimiza, el criterio seleccionado. Generalmente habrá al menos un algoritmo (puede que varios) que nos permitan llevar a cabo esta optimización. Cuando esto no sea así, siempre cabe la posibilidad, de generar direcciones al azar y, de entre todas ellas, seleccionar aquélla en la que el criterio toma su valor más alto o bajo.

3. Si el problema lo requiere, se calculan varias direcciones de proyección. Generalmente se añade la restricción de que cada una de estas direcciones sea perpendicular a las demás. Para ello, una posibilidad es optimizar el criterio sucesivas veces, con la condición de que la dirección encontrada debe ser ortogonal a todas las calculadas en los pasos anteriores.

4. Interpretar los resultados obtenidos. Para visualizar la nube de puntos, por ejemplo, podríamos empezar representando gráficamente los datos sobre el plano definido por la primera pareja de direcciones de proyección encontrada.

En definitiva, las técnicas de «projection pursuit» tienen como objetivo encontrar direcciones de proyección «interesantes», a partir de las que sea posible interpretar la información contenida en los datos. Como ya se ha mencionado repetidas veces, hay muchos criterios para escoger una buena dirección de proyección. En la siguiente Sección presentaremos el análisis de componentes principales desde esta perspectiva.

2.2 El análisis de componentes principales

Como mostraba el ejemplo de la Sección anterior, cabe esperar que las direcciones en las que las proyecciones son «más grandes» sean aquéllas a lo largo de las cuales se distribuye la nube de puntos de datos. Dichas direcciones nos revelarán por ello la disposición de los datos en el espacio y se las llama, por tanto, «direcciones principales». El objetivo de PCA es, precisamente, definir criterios y algoritmos apropiados para determinar estas direcciones principales [54]. Discutiremos, en primer lugar, el enfoque más tradicional y, a partir de él, presentaremos después la técnica emergente que se conoce como L1-PCA.

2.2.1 El análisis de componentes principales basado en la norma L^2

Sea

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}_1 \\ \vdots \\ \mathrm{x}_p \end{bmatrix}$$

el vector aleatorio que modela los datos que queremos analizar. Como se vio en la ecuación (2.2),

$$\mathrm{z} = \boldsymbol{a}^\top \mathbf{x}$$

representa la proyección de $\mathbf{x}$ sobre la recta en la dirección del vector unitario $\boldsymbol{a}$.

Supongamos querer encontrar la dirección que «mejor» se alinee con $\mathbf{x}$. Intuitivamente, las proyecciones sobre esta dirección, es decir, los valores observados de z, deberían ser «grandes» *en promedio*. Por ello, como primera idea, podríamos seleccionar la dirección en la que se maximiza

$$\mathbb{E}[\mathrm{z}] = \boldsymbol{a}^\top \mathbb{E}[\mathbf{x}].$$

Sin embargo, es fácil comprobar que $\boldsymbol{a}$ es óptimo cuando tiene la misma dirección que el vector medio de la nube de puntos,

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}].$$

Vemos por tanto que este criterio es poco informativo: nos dice dónde está la nube de puntos (su valor medio); pero no la forma que tiene (las direcciones en las que se expande).

En la práctica, una vez determinado este punto medio, se suele sustraer para que no condicione los análisis posteriores. De esta manera,

$$\boldsymbol{\mu} = \mathbf{0} \rightarrow \mathbb{E}[\mathrm{z}] = 0. \tag{2.5}$$

Supondremos, por lo tanto, que el valor esperado de los datos es cero.

Descartada la idea anterior, el PCA tradicional calcula la dirección deseada como aquélla en la que se maximiza la varianza de las proyecciones [54], esto es,

$$\begin{aligned} \sigma_{\mathrm{z}}^2 &= \mathbb{E}\left[(\mathrm{z} - \mathbb{E}[\mathrm{z}])^2\right] \\ &\underset{\mathbb{E}[\mathrm{z}]=0}{=} \mathbb{E}[\mathrm{z}^2]. \end{aligned}$$

Intuitivamente, este criterio también favorece que las proyecciones sean, en media, grandes. Por ello, nos servirá para alcanzar el objetivo que se desea. Además, dado que la varianza se interpreta como la distancia de los puntos a la media, podemos decir que PCA proyecta en la dirección en la que la

variabilidad o dispersión de los datos es mayor.

En la práctica, dados N valores, $z_1, \ldots, z_N$, de la variable aleatoria z, la varianza se aproxima mediante el promedio de los cuadrados de los coeficientes de proyección z_n, es decir,

$$\frac{1}{N}\sum_{n=1}^{N} z_n^2. \tag{2.6}$$

Finalmente, dado que la norma L^2, o longitud, del vector $[z_1, z_2, \ldots, z_n]$ se define precisamente como [100]

$$\sqrt{\sum_{n=1}^{N} z_n^2},$$

a esta técnica se la conoce también como PCA basado en la norma L^2 o L2-PCA.

2.2.2 Cálculo de las componentes principales en L2-PCA

Sea $\boldsymbol{z} = [z_1, \ldots, z_N]$ el vector fila que contiene los coeficientes de proyección,

$$z_n = \sum_{n=1}^{p} a_n x_n = \boldsymbol{a}^\top \boldsymbol{x}_n,$$

en la dirección de $\boldsymbol{a}$. También se puede escribir

$$\boldsymbol{z} = \boldsymbol{a}^\top \boldsymbol{X},$$

donde $\boldsymbol{X} = [\boldsymbol{x}_1, \ \ldots, \boldsymbol{x}_N]$ es la matriz $p \times N$ cuya enésima columna es $\boldsymbol{x}_n$.

Como estamos suponiendo que la media de los vectores originales $\boldsymbol{x}_n$ es cero, también lo será la del vector $\boldsymbol{z}$. Por ello, podemos aproximar la varianza como:

$$s^2 \stackrel{\text{def}}{=} \frac{1}{N-1}\sum_{n=1}^{N} z_n^2,$$

donde se divide por $N-1$, y no por N, para que el estimador no tenga sesgo [60, 59]. Resulta que

$$\begin{aligned} s^2 &= \frac{\boldsymbol{z}\boldsymbol{z}^\top}{N-1} \\ &= \frac{\boldsymbol{a}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{a}}{N-1} \\ &= \boldsymbol{a}^\top \boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a}, \end{aligned}$$

donde

$$\boldsymbol{C}_{\boldsymbol{x}} = \frac{1}{N-1}\boldsymbol{X}\boldsymbol{X}^\top = \frac{1}{N-1}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top$$

es la matriz de covarianza de los datos.

Se desea maximizar s^2, con la restricción de que $\boldsymbol{a}$ sea un vector de longitud unidad. Es decir, nos enfrentamos al problema:

$$\max_{\|\boldsymbol{a}\|^2=1} s^2 = \max_{\|\boldsymbol{a}\|^2=1} \boldsymbol{a}^\top \boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a}$$

con $\|\boldsymbol{a}\|^2 = \boldsymbol{a}^\top \boldsymbol{a}$. La solución se puede calcular utilizando la técnica de los multiplicadores de Lagrange. Sea

$$\mathscr{L} = \boldsymbol{a}^\top \boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a} - \lambda(\boldsymbol{a}^\top \boldsymbol{a} - 1)$$

la función de Lagrange, donde λ es el multiplicador [105]. Derivando esta expresión con respecto a las componentes de $\boldsymbol{w}$ e igualando a cero, es decir,

$$\frac{\partial \mathscr{L}}{\partial\, \boldsymbol{w}} = 2\boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a} - 2\lambda \boldsymbol{a} = 0,$$

se obtiene la ecuación

$$\boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a} = \lambda \boldsymbol{a}.$$

Esta ecuación es la *clave*: indica que $\boldsymbol{a}$ es un *autovector* de $\boldsymbol{C}_{\boldsymbol{x}}$, mientras que λ es el correspondiente *autovalor* [86].

Para determinar de qué autovector se trata, multiplicamos ambos lados de la ecuación anterior por $\boldsymbol{a}^\top$, obteniendo

$$\boldsymbol{a}^\top \boldsymbol{C_x} \boldsymbol{a} = \lambda \boldsymbol{a}^\top \boldsymbol{a},$$

o, lo que es lo mismo,

$$s^2 = \lambda.$$

Es decir, la varianza coincide numéricamente con el autovalor. Para que la varianza sea máxima, por tanto, el autovalor también tendrá que serlo. Así pues, *la dirección de proyección óptima es la del autovector correspondiente al autovalor más grande de la matriz* $\boldsymbol{C_x}$. En lo que sigue, llamaremos $\boldsymbol{a}_1$ a este autovector. En otras palabras:

$$\boldsymbol{a}_1 = \arg\max_{\boldsymbol{a}} \boldsymbol{a}^\top \boldsymbol{C_x} \boldsymbol{a},$$
$$\text{sujeto a } \|\boldsymbol{a}\|^2 = 1$$

2.2.3 Cálculo de las restantes direcciones principales

Supongamos querer ahora calcular una segunda «dirección principal». De entre las diferentes opciones disponibles para llevar esta tarea a cabo, se utiliza con frecuencia la que describiremos a continuación [54]. En primer lugar, restamos a los datos su componente en la dirección de $\boldsymbol{a}_1$ a fin de eliminarla. Sea:

$$\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n - z_n \boldsymbol{a}_1 \tag{2.7}$$

donde

$$z_n = \boldsymbol{a}_1^\top \boldsymbol{x}_n.$$

Al haber sustraído esta componente, es fácil comprobar que $\boldsymbol{a}_1^\top \hat{\boldsymbol{x}}_n = 0$, lo que quiere decir que los nuevos datos son siempre perpendiculares a $\boldsymbol{a}_1$.

La segunda «dirección principal» se obtiene ahora maximizando las proyecciones de estos nuevos datos $\hat{\boldsymbol{x}}_n$. Repitiendo el procedimiento anterior,

para obtener esta dirección resolveremos el siguiente problema:

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} \boldsymbol{a}^\top \boldsymbol{C}_{\hat{\boldsymbol{x}}} \boldsymbol{a},$$
$$\text{sujeto a } \|\boldsymbol{a}\|^2 = 1$$

donde

$$\boldsymbol{C}_{\hat{\mathbf{x}}} = \frac{1}{N-1} \sum_{n=1}^{N} \hat{\boldsymbol{x}}_n \hat{\boldsymbol{x}}_n^\top.$$

Razonando como antes, la segunda «dirección principal» resultará ser la del autovector de $\boldsymbol{C}_{\hat{\boldsymbol{x}}}$ asociado al autovalor más grande.

Ahora bien, en los siguientes párrafos demostraremos que $\boldsymbol{C}_{\hat{\boldsymbol{x}}}$ y la matriz de covarianzas original $\boldsymbol{C}_{\boldsymbol{x}}$ comparten, en realidad, los mismos autovectores. Por ello, en la práctica no es necesario calcular las variables corregidas. El cálculo de la nueva dirección principal puede hacerse, directamente, a partir de $\boldsymbol{C}_{\boldsymbol{x}}$.

Por conveniencia, vamos a llamar $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_p$ a los autovectores de $\boldsymbol{C}_{\boldsymbol{x}}$, siendo $\lambda_1 > \lambda_2 > \ \ldots > \lambda_p > 0$ los correspondientes autovalores. Empezaremos reescribiendo (2.7) como

$$\hat{\boldsymbol{x}}_n = (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top) \boldsymbol{x}_n,$$

donde $\boldsymbol{I}_p$ es la matriz identidad $p \times p$. Por tanto,

$$\hat{\boldsymbol{x}}_n \hat{\boldsymbol{x}}_n^\top = (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top) \boldsymbol{x}_n \boldsymbol{x}_n^\top (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top).$$

Si ahora sumamos para todo n, obtenemos la siguiente relación entre las matrices de covarianza:

$$\begin{aligned}
\boldsymbol{C}_{\hat{\mathbf{x}}} &= \frac{1}{N-1} \sum_{n=1}^{N} \hat{\boldsymbol{x}}_n \hat{\boldsymbol{x}}_n^\top \\
&= (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top) \left(\frac{1}{N-1} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top \right) (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top) \\
&= (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top) \boldsymbol{C}_{\mathbf{x}} (\boldsymbol{I}_p - \boldsymbol{a}_1 \boldsymbol{a}_1^\top).
\end{aligned}$$

Se observa en primer lugar que

$$\boldsymbol{C}_{\hat{\mathbf{x}}}\boldsymbol{a}_1 = (\boldsymbol{I}_p - \boldsymbol{a}_1\boldsymbol{a}_1^\top)\boldsymbol{C}_{\mathbf{x}}(\boldsymbol{a}_1 - \boldsymbol{a}_1) = \mathbf{0},$$

lo que quiere decir que $\boldsymbol{a}_1$ es también un autovector de $\boldsymbol{C}_{\hat{\mathbf{x}}}$, siendo el autovalor correspondiente igual a cero.

Sea ahora $\boldsymbol{a}_i$ cualquier otro autovector de $\boldsymbol{C}_{\mathbf{x}}$, con autovalor asociado λ_i, esto es,

$$\boldsymbol{C}_{\mathbf{x}}\boldsymbol{a}_i = \lambda_i\boldsymbol{a}_i.$$

Como los autovectores de una matriz de covarianza tienen la propiedad de ser perpendiculares entre sí, es decir, $\boldsymbol{a}_1^\top\boldsymbol{a}_i = 0$, con $i \neq 1$, se tiene que

$$\begin{aligned}\boldsymbol{C}_{\hat{\mathbf{x}}}\boldsymbol{a}_i &= (\boldsymbol{I}_p - \boldsymbol{a}_1\boldsymbol{a}_1^\top)\boldsymbol{C}_{\mathbf{x}}(\boldsymbol{I}_p - \boldsymbol{a}_1\boldsymbol{a}_1^\top)\boldsymbol{a}_i \\ &= \lambda_i\boldsymbol{a}_i.\end{aligned}$$

Por lo tanto, $\boldsymbol{a}_i$ también es un autovector de $\boldsymbol{C}_{\hat{\mathbf{x}}}$. Es más, el autovalor, λ_i, *no cambia su valor numérico* como se quería demostrar. De aquí se deduce para finalizar que, puesto que de todos los autovectores el que tiene el mayor autovalor asociado resulta ser $\boldsymbol{a}_2$, es éste precisamente el que nos da la segunda «dirección principal».

En general, para determinar la emésima «dirección principal», podríamos definir:

$$\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \sum_{i=1}^{m-1}(\boldsymbol{a}_i^\top\boldsymbol{x}_n)\boldsymbol{a}_i.$$

y repetiremos el mismo procedimiento. En definitiva, se obtiene el siguiente resultado fundamental:

Teorema 2.2.1 (Direcciones principales asociadas a L2-PCA) *Las primeras m «direcciones principales» (*$1 \leq m \leq p$*) son las de los autovectores asociados a los m mayores autovalores de la matriz de covarianzas*

$$\boldsymbol{C}_{\mathbf{x}} = \frac{1}{N-1}\boldsymbol{X}\boldsymbol{X}^\top = \frac{1}{N-1}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top.$$

A partir del teorema anterior resulta inmediato calcular en la práctica las direcciones principales. Ahora bien, por razones de eficiencia computacional, el cálculo se suele hacer a partir de la descomposición en valores singulares de la matriz $\boldsymbol{X} \in \mathbb{R}^{p \times N}$ que almacena los datos [91]. Esta descomposición permite escribir:

$$\boldsymbol{X} = \boldsymbol{U}\,\boldsymbol{D}\,\boldsymbol{V}^{\top},$$

donde $\boldsymbol{U} \in \mathbb{R}^{p \times p}$ es la matriz cuyas columnas son los autovectores de $\boldsymbol{X}\,\boldsymbol{X}^{\top}$; $\boldsymbol{D} \in \mathbb{R}^{p \times p}$ es una matriz diagonal que contiene los valores singulares de $\boldsymbol{X}$, los cuales coinciden con $\sqrt{\lambda_i}$, $i = 1, \ldots, p$; finalmente, $\boldsymbol{V} \in \mathbb{R}^{N \times N}$ contiene los autovectores de $\boldsymbol{X}^{\top}\,\boldsymbol{X}$. Por lo tanto, las direcciones principales pueden extraerse directamente de la matriz $\boldsymbol{U}$, dado que los autovectores de $\boldsymbol{X}\,\boldsymbol{X}^{\top}$ son los mismos que los de $\boldsymbol{C}_{\mathbf{x}}$. Este procedimiento es más robusto frente a los errores numéricos que usar un algoritmo que calcule los autovectores de la matriz $\boldsymbol{C}_{\boldsymbol{x}}$ directamente; sin embargo, también puede requerir más tiempo de computación. Para acelerar el método se suelen despreciar los últimos $N - p$ autovectores de $\boldsymbol{V}$ ya que son aleatorios y no es necesario calcularlos. Ello permite llevar a cabo la llamada descomposición en valores singulares reducida (*thin SVD*), consiguiendo con ello que el procedimiento sea viable en cuanto a cálculos [86].

Como observación final, nótese también que, por las propiedades de los autovectores, *todas las direcciones principales resultan ser perpendiculares entre sí.*

2.2.4 Interpretación de L2-PCA

El análisis de componentes principales tradicional admite una interpretación intuitiva, que describiremos a continuación. Aplicando el teorema de Pitágoras al triángulo que se observa en la Figura 2.2, se comprueba que la

hipotenusa $\|\boldsymbol{x}_1\|$ verifica la relación

$$\begin{aligned}\|\boldsymbol{x}_1\|^2 &= d_1^2 + \|\boldsymbol{z}_1\|^2 \\ &= d_1^2 + z_1^2\|\boldsymbol{a}\|^2 \\ &= d_1^2 + z_1^2, \end{aligned} \tag{2.8}$$

donde $\|\boldsymbol{v}\|$ denota la norma L^2 o longitud del vector $\boldsymbol{v}$. Para obtener la igualdad anterior se ha usado, además, que $\|\boldsymbol{a}\|^2 = 1$. Por tanto, se tendrá que

$$d_1^2 = \|\boldsymbol{x}_1\|^2 - z_1^2.$$

En general, la distancia entre un punto cualquiera $\boldsymbol{x}_n$ y su correspondiente proyección $\boldsymbol{z}_n$ tendrá el siguiente valor:

$$d_n^2 = \|\boldsymbol{x}_n\|^2 - z_n^2.$$

Sumando todas las distancias:

$$\sum_{n=1}^{N} d_n^2 = \sum_{n=1}^{N} \|\boldsymbol{x}_n\|^2 - \sum_{n=1}^{N} z_n^2.$$

Como $\sum_{n=1}^{N} \|\boldsymbol{x}_n\|^2$ es una constante que no depende de la dirección de proyección, vemos finalmente que cuando la suma de los cuadrados de las proyecciones,

$$\sum_{n=1}^{N} z_n^2,$$

es máxima, como busca PCA, la suma de los cuadrados de las distancias,

$$\sum_{n=1}^{N} d_n^2,$$

es *mínima*. En otras palabras: la «dirección principal» coincide precisamente con la de la recta que pasa «más cerca» de todos los puntos o, en otras palabras, que mejor los ajusta.

2.2.5 Estandarización o «blanqueado» usando PCA

En estadística es habitual estandarizar o normalizar las variables aleatorias escalares para poder compararlas cuando tienen distintas unidades de medida. El proceso consta de dos pasos: primero se centra la variable (es decir, se le resta su media) y, después, se divide por su desviación típica [89].

Este procedimiento también se puede llevar a cabo cuando se trabaja con vectores aleatorios. En este caso, la estandarización, a la que también se conoce como «blanqueado», resulta estar muy relacionada con el análisis de componentes principales [50, 61]. Para explicar esto, empezamos escribiendo la matriz de covarianza de los datos como:

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{C}_{\mathbf{x}}^{1/2}(\boldsymbol{C}_{\mathbf{x}}^{1/2})^{\top},$$

donde $\boldsymbol{C}_{\boldsymbol{x}}^{1/2}$ es la «raíz cuadrada» de $\boldsymbol{C}_{\boldsymbol{x}}$. La matriz «raíz cuadrada» se construye a partir de la descomposición en autovalores y autovectores de la matriz de covarianza de los datos como sigue: sea

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^{\top},$$

donde $\boldsymbol{D}$ es una matriz diagonal que contiene los autovalores de $\boldsymbol{C}_{\boldsymbol{x}}$, mientras que las columnas de $\boldsymbol{W}$ son los correspondientes autovectores. Recordemos, asimismo, que las columnas de $\boldsymbol{W}$ definen precisamente las direcciones principales asociadas a PCA. Dado que $\boldsymbol{W}$ es una matriz ortogonal, es decir, siempre se cumple que $\boldsymbol{W}^{\top}\boldsymbol{W} = \boldsymbol{I}$, definimos la raíz cuadrada simplemente como:

$$\boldsymbol{C}_{\mathbf{x}}^{1/2} = \boldsymbol{W}\boldsymbol{D}^{1/2}\boldsymbol{W}^{\top}, \tag{2.9}$$

siendo $\boldsymbol{D}^{1/2}$ la matriz diagonal que contiene las raíces cuadradas de los coeficientes de $\boldsymbol{D}$. En realidad, esta definición no es única: si $\boldsymbol{M}$ es cualquier otra matriz ortogonal, es decir, $\boldsymbol{M}\boldsymbol{M}^{\top} = \boldsymbol{I}$, entonces

$$\boldsymbol{C}_{\mathbf{x}}^{1/2}\boldsymbol{M}$$

también sería una raíz cuadrada válida.

El vector aleatorio estandarizado, o «blanqueado», se define ahora como

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \boldsymbol{C}_{\mathbf{x}}^{-1/2}(\boldsymbol{x}-\bar{\boldsymbol{x}}) \\ &= \boldsymbol{C}_{\mathbf{x}}^{-1/2}\boldsymbol{x},\end{aligned} \tag{2.10}$$

siendo $\bar{\boldsymbol{x}}$ el valor medio de $\boldsymbol{x}$, que suponemos nulo. El nuevo vector $\tilde{\boldsymbol{x}}$ tiene también media cero y su matriz de covarianzas es la identidad. En efecto,

$$\boldsymbol{C}_{\tilde{\mathbf{x}}} = \boldsymbol{C}_{\mathbf{x}}^{-1/2}\boldsymbol{C}_{\mathbf{x}}\boldsymbol{C}_{\mathbf{x}}^{-1/2} = \boldsymbol{I}.$$

De esta manera, hemos pasado de tener variables correladas en $\boldsymbol{x}$ a incorreladas en $\tilde{\boldsymbol{x}}$.

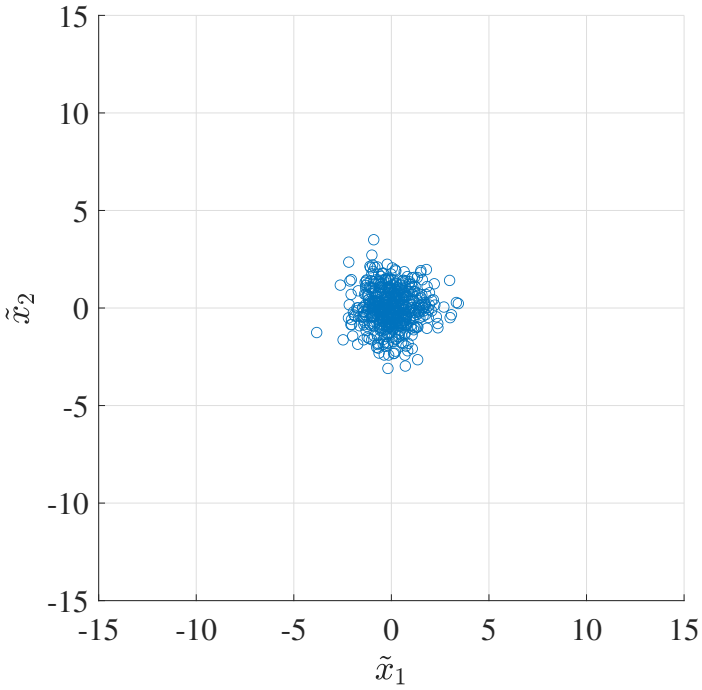


Figura 2.4 Diagrama de dispersión mostrando 500 puntos de una variable aleatoria de Gauss bidimensional blanqueada.

Como ilustración de este método, después de blanquear la nube de puntos que aparece en la Figura 2.1 se obtiene la mostrada en la Figura 2.4. Vemos

que ya no existe una «dirección principal» como antes: los datos parecen extenderse por igual en todas direcciones. Esto puede explicarse de la siguiente forma: combinando las ecuaciones (2.9) y (2.10) se tiene

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \boldsymbol{C}_{\mathbf{x}}^{-1/2}\boldsymbol{x} \\ &= \boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{W}^{\top}\boldsymbol{x} \\ &= \boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{z},\end{aligned}$$

donde $\boldsymbol{z} = \boldsymbol{W}^{\top}\boldsymbol{x}$ es el vector que contiene la proyección de los datos sobre las direcciones principales. La clave está en que al multiplicar por $\boldsymbol{D}^{-1/2}$ se ajusta la varianza de estas proyecciones para que sea siempre igual a la unidad. De esta forma se consigue que la nube de puntos no se distribuya en ninguna dirección apreciablemente más que en las otras.

2.2.6 Sensibilidad de L2-PCA a los datos atípicos

Finalmente, hemos de notar que L2-PCA tiene una severa deficiencia que limita su uso práctico: resulta ser muy sensible a la presencia de datos atípicos («outliers» en inglés). Un valor atípico es aquél que está muy separado del resto de los datos en el espacio, lo que hace sospechar que, probablemente, se trate de un error de medida [2, 39].

Para ilustrar esta deficiencia, en la Figura 2.5 se muestra el efecto de haber añadido un grupo de datos atípicos a la nube de puntos que nos está sirviendo como ejemplo. Resulta que la «dirección principal» calculada se desvía 13.5° respecto de la que habríamos obtenido de no haber añadido los *outliers*.

Esta sensibilidad frente a los datos atípicos de L2-PCA se explica porque el cuadrado en la función objetivo (2.6), es decir,

$$\sum_{n=1}^{N} z_n^2,$$

favorece, amplificando más, a los coeficientes de proyección que tienen mayor magnitud. En cambio, los coeficientes pequeños, menores que uno, ven reducida su influencia. Cabe esperar de todo ello que la varianza tome

valores grandes, y sea máxima, alineando la dirección de proyección con los *outliers*.

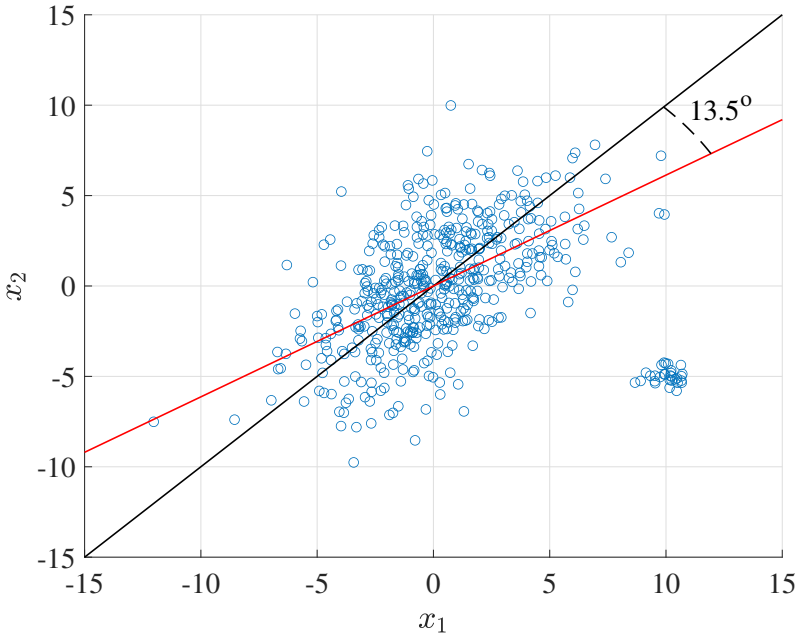


Figura 2.5 Nube de puntos con *outliers* en torno a la coordenada (10, 5). La línea negra muestra la dirección principal de los datos sin *outliers*. La línea roja, la de los datos con *outliers*, calculada con L2-PCA. Vemos que estos últimos han introducido una desviación de 13.5^o en la dirección.

2.3 Análisis de componentes principales basado en la norma L^1

Para paliar la sensibilidad de L2-PCA, se ha propuesto sustituir el problemático cuadrado de la función objetivo por el valor absoluto, dando lugar

al criterio alternativo [65, 75, 77]:

$$M = \sum_{n=1}^{N} |z_n|, \tag{2.11}$$

Como (2.11) es la norma L^1 del vector $\boldsymbol{z}$ cuya enésima componente viene dada por $z_n = \boldsymbol{a}^\top \boldsymbol{x}_n$, al PCA basado en el criterio (2.11) se le conoce como análisis de componentes principales basado en la norma L^1 o, simplemente, L1-PCA.

Para hacer un primer estudio exploratorio de L1-PCA, supongamos que el vector aleatorio $\boldsymbol{x}$ tiene una distribución normal, o gaussiana, en el espacio de p dimensiones. Ésta es una hipótesis habitual en análisis estadístico [95]. Es decir, se supone que la función de densidad de probabilidad de $\boldsymbol{x}$ es

$$f(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{C}_{\mathbf{x}})^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}^{-1}\boldsymbol{x}}, \tag{2.12}$$

donde $\boldsymbol{C}_{\mathbf{x}}$ es la matriz de covarianza de los datos. Sea $z = \boldsymbol{a}^\top \boldsymbol{x}$ la proyección de $\boldsymbol{x}$ sobre la dirección definida por $\boldsymbol{a} \in \mathbb{R}^p$. Utilizando las propiedades de la función gaussiana, se sabe que la función de densidad de probabilidad de z también es gaussiana:

$$f(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{z^2}{2\sigma^2}\right), \tag{2.13}$$

siendo $\sigma^2 = \boldsymbol{a}^\top \boldsymbol{C}_{\mathbf{x}} \boldsymbol{a}$ la varianza de z.

Para N suficientemente grande, (2.11) puede aproximarse a partir de la esperanza matemática de z, es decir,

$$\frac{1}{N} \sum_{n=1}^{N} |z_n| \xrightarrow{N\to\infty} \mathbb{E}\{|z|\}$$

Utilizando ahora la relación [3]

$$\int z e^{-\frac{z^2}{2\sigma^2}} \, \mathrm{d}z = -\sigma^2 e^{-\frac{z^2}{2\sigma^2}} + \text{ constante de integración}$$

se obtiene, tras realizar algunas operaciones sencillas, que

$$\mathbb{E}[|z|] = \int_{-\infty}^{\infty} |z|\, f(z)\, \mathrm{d}z = \sqrt{\frac{2}{\pi}}\,\sigma.$$

En consecuencia, como maximizar la desviación estándar σ es equivalente a maximizar la varianza σ^2, L1-PCA obtendría bajo la hipótesis de gaussianeidad los mismos resultados que L2-PCA.

Cálculo de las componentes principales en L1-PCA

Se han propuesto varios algoritmos (véase por ejemplo [65, 75, 77]) para maximizar la función objetivo (2.11), es decir, para resolver el problema

$$\max_{\|\boldsymbol{a}\|^2=1} M,$$

siendo $M = \sum_{n=1}^{N} |z_n|$ y $z_n = \boldsymbol{a}^\top \boldsymbol{x}_n$.

Aplicando la técnica de los multiplicadores de Lagrange, podemos convertirlo en un problema sin restricciones sobre $\boldsymbol{a}$:

$$\max_{\boldsymbol{a}} \left(M - \lambda \boldsymbol{a}^\top \boldsymbol{a}\right)$$

siendo λ el multiplicador de Lagrange [16]. Calculando la derivada con respecto a $\boldsymbol{a}$ e igualando a cero se obtiene la ecuación:

$$\frac{\partial M}{\partial \boldsymbol{a}} = 2\lambda \boldsymbol{a} \tag{2.14}$$

donde

$$\frac{\partial M(\boldsymbol{a})}{\partial \boldsymbol{a}} = \begin{bmatrix} \frac{\partial M(\boldsymbol{a})}{\partial a_1} \\ \frac{\partial M(\boldsymbol{a})}{\partial a_2} \\ \vdots \\ \frac{\partial M(\boldsymbol{a})}{\partial a_p} \end{bmatrix}.$$

Tomando la función «signo» como derivada del valor absoluto [45], es decir,

$$\frac{\mathrm{d}|z|}{\mathrm{d}z} = \mathrm{sign}(z) = \begin{cases} 1 & z > 0 \\ -1 & z < 0 \end{cases},$$

se obtiene, utilizando la regla de la cadena, que

$$\frac{\partial |z_n|}{\partial \boldsymbol{a}} = \mathrm{sign}(z_n)\boldsymbol{x}_n.$$

Por lo tanto, (2.14) se convierte en:

$$\sum_{n=1}^{N} \mathrm{sign}(z_n)\boldsymbol{x}_n = 2\lambda \boldsymbol{a}. \tag{2.15}$$

Al multiplicar ambos lados de la ecuación por $\boldsymbol{a}^\top$, y recordando que $\boldsymbol{a}^\top \boldsymbol{a} = 1$, se obtiene

$$\lambda = \sum_{n=1}^{N} \mathrm{sign}(z_n) z_n = \sum_{n=1}^{N} |z_n|.$$

Por tanto, sustituyendo en (2.15), las direcciones de proyección óptimas verificarán la ecuación:

$$\boldsymbol{a} = \frac{\sum_{n=1}^{N} \mathrm{sign}(z_n)\boldsymbol{x}_n}{\sum_{n=1}^{N} |z_n|}. \tag{2.16}$$

Es fácil además comprobar que se cumple la restricción $\boldsymbol{a}^\top \boldsymbol{a} = 1$, pues

$$\begin{aligned} \boldsymbol{a}^\top \boldsymbol{a} &= \frac{\sum_{n=1}^{N} \mathrm{sign}(z_n)\boldsymbol{a}^\top \boldsymbol{x}_n}{\sum_{n=1}^{N} |z_n|} \\ &= \frac{\sum_{n=1}^{N} \mathrm{sign}(z_n)\, z_n}{\sum_{n=1}^{N} |z_n|} = 1 \end{aligned}$$

Para resolver (2.16), se ha propuesto la siguiente iteración de punto fijo [65]:

Algoritmo 1: Iteración para determinar la «dirección principal» de los datos según el criterio L1-PCA.

Inicializa $\boldsymbol{a}(0)$ con cualquier vector que tenga longitud unidad ;
para $t = 1, 2, \ldots$ **hacer**
  $z_n(t) = \boldsymbol{a}(t-1)^\top \boldsymbol{x}_n$ para $n = 1, \ldots, N$;
  $\boldsymbol{a}(t) = \sum_{n=1}^{N} \operatorname{sign}(z_n(t)) \boldsymbol{x}_n$;
  $\boldsymbol{a}(t) = \frac{\boldsymbol{a}(t)}{\|\boldsymbol{a}(t)\|}$
fin

Se puede demostrar que, aplicando este algoritmo, la función objetivo crece, o mantiene su valor, iteración a iteración [65]:

$$M(t) \geq M(t-1) \tag{2.17}$$

siendo $M(t) = \sum_{n=1}^{N} |z_n(t)|$. Como M es una función acotada, el algoritmo siempre convergerá a un máximo de la función. No hay garantía, sin embargo, de que éste sea el máximo global: la función M podría tener máximos locales y, por tanto, el algoritmo podría tender a uno de ellos.

Para comprobar (2.17) llevamos a cabo el siguiente razonamiento:

$$\begin{aligned} M(t) &= \sum_{n=1}^{N} |z_n(t)| = \sum_{n=1}^{N} \operatorname{sign}(z_n(t))\, z_n(t) \\ &\geq \sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))\, z_n(t) \\ &= \boldsymbol{a}(t-1)^\top \sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))\, \boldsymbol{x}_n \end{aligned}$$

Ahora bien, los vectores $\boldsymbol{a}(t-1)$ y

$$\sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))\, \boldsymbol{x}_n(t) \tag{2.18}$$

son paralelos por construcción del algoritmo. Por tanto, su producto escalar siempre será mayor que el del vector (2.18) por cualquier otro que tenga la misma longitud que $\boldsymbol{a}(t-1)$. En particular,

$$\begin{aligned}\boldsymbol{a}(t-1)^\top \sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))\boldsymbol{x}_n &\geq \boldsymbol{a}(t-2)^\top \sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))\boldsymbol{x}_n \\ &= \sum_{n=1}^{N} \operatorname{sign}(z_n(t-1))z_n(t-1) \\ &= \sum_{n=1}^{N} |z_n(t-1)| \\ &= M(t-1)\end{aligned}$$

como queríamos demostrar.

Repitiendo ahora el experimento ilustrado en la Figura 2.5, pero donde el cálculo de la dirección principal se lleva a cabo mediante L1-PCA, encontramos que la desviación entre el valor calculado y el ideal se reduce a 5.59^o. El resultado se muestra en la Figura 2.6.

Cálculo de otras direcciones principales

Una vez determinada la primera «dirección principal» mediante L1-PCA, alineada con el vector $\boldsymbol{w}_1$, se plantea el problema de encontrar otras. Para ello, se procede de forma similar a como se hizo en el apartado 2.2.3. El algoritmo sería el siguiente [65]:

Algoritmo 2: Iteración para determinar p «direcciones principales» mediante L1-PCA.

para $j = 2, \ldots, p$ **hacer**
  Sea $\hat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \sum_{i=1}^{j-1} (\boldsymbol{a}_i^\top \boldsymbol{x}_n)\boldsymbol{a}_i$ para $n = 1, \ldots, N$;
  Para calcular $\boldsymbol{a}_j$, aplica el algoritmo L1-PCA a $\{\hat{\boldsymbol{x}}_n\}_{n=1}^{N}$;
fin

De esta forma se determinan p direcciones ortogonales entre sí. No existe

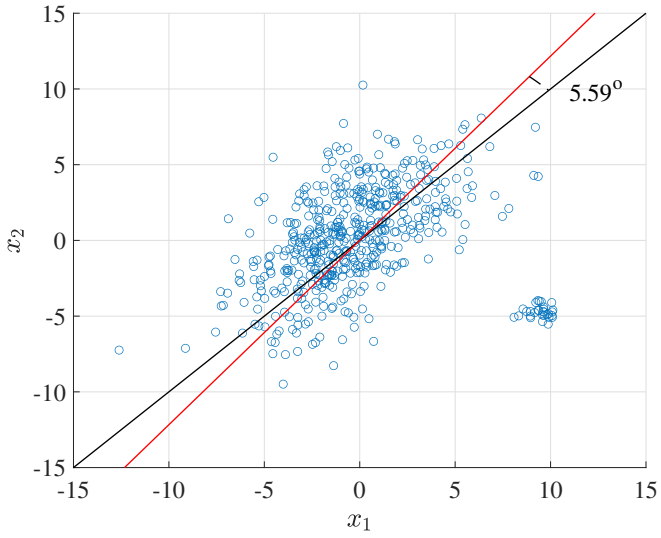


Figura 2.6 Nube de puntos con *outliers* en torno a la coordenada (10, 5). La línea negra muestra la dirección principal de los datos sin *outliers*. La línea roja, correspondiente a los datos con *outliers*, ha sido calculada mediante L1-PCA. Se observa que el error ha disminuido respecto al que se obtenía al utilizar L2-PCA.

garantía teórica de que sean óptimas; pero en los experimentos realizados muestran que el algoritmo produce resultados aceptables en la práctica.

3 Sobre las propiedades discriminativas de L1-PCA

La técnica L1-PCA ha sido presentada, en principio, como la alternativa «robusta» a L2-PCA para datos contaminados con «outliers». En este Capítulo iremos más allá mostrando, en concreto, que L1-PCA también tiene la capacidad de resolver problemas de clasificación en los que los datos provienen de poblaciones que se solapan parcialmente. Este resultado constituye precisamente la principal aportación de esta Tesis.

3.1 Formulación del problema

Sea $\mathbf{x} \in \mathbb{R}^p$ un vector aleatorio en un espacio de p dimensiones. Supondremos que sus muestras, $\boldsymbol{x}_n$, se extraen al azar de dos poblaciones distintas, que llamaremos $\mathscr{C}_1$ y $\mathscr{C}_2$.

Supondremos, además, *que ambas poblaciones tienen igual valor medio,*

$$\boldsymbol{\mu} = \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2,$$

lo que implica que puede haber *un fuerte solapamiento entre las nubes de puntos asociadas a cada clase.* El problema que se plantea en esta Tesis es el siguiente: *dada una muestra aleatoria de* $\mathbf{x}$*, determinar la población (*$\mathscr{C}_1$

o $\mathscr{C}_2$) a la que pertenece. Podemos anticipar ya que nuestra contribución para resolverlo será un algoritmo *no supervisado* basado en L1-PCA.

Como ilustración, la figura Figura 3.1 muestra 1 000 valores observados pertenecientes a dos poblaciones bidimensionales, «roja» y «azul», con respectivas matrices de covarianza

$$\boldsymbol{C}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix},$$
$$\boldsymbol{C}_2 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}.$$

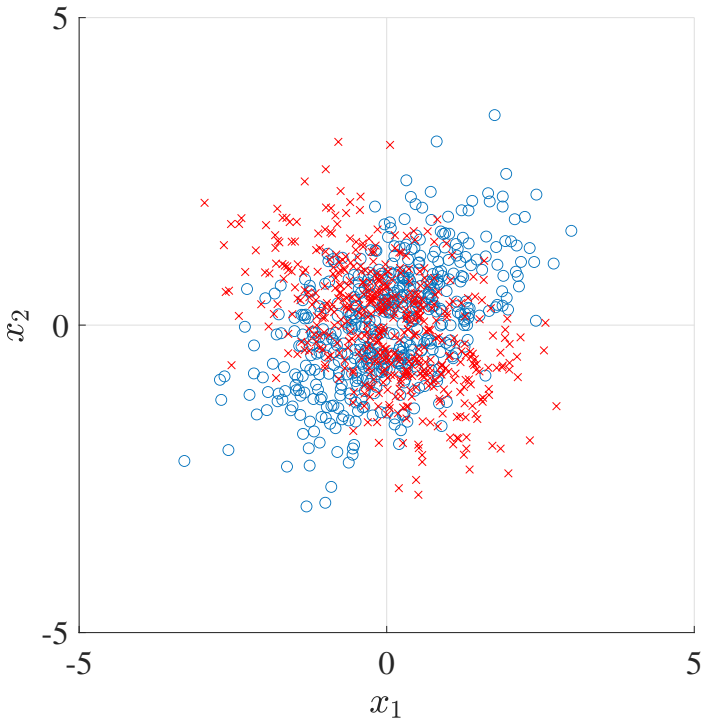


Figura 3.1 Diagrama de dispersión mostrando 1000 puntos de dos variables aleatorias de Gauss bidimensionales (500 puntos por cada variable).

Se aprecia que la población «azul» se extiende principalmente sobre la

recta $x_2 = x_1$, mientras que la roja lo hace sobre $x_2 = -x_1$. No obstante, ambas nubes de puntos se solapan en buena medida. Ello da idea de lo difícil que puede llegar a ser determinar a qué clase pertenece una muestra cualquiera, obtenida al azar, de la población.

3.2 Suposiciones básicas

Se harán tres hipótesis básicas en este texto, sin perjuicio de que también puedan formularse otras hipótesis adicionales (por ejemplo, suponer que la distribución de los datos es Gaussiana) más adelante.

En primer lugar, con el fin de simplificar los cálculos, se asumirá que los datos han sido centrados, es decir, que se ha restado a las observaciones su valor medio, por lo que:

$$\boldsymbol{\mu} = \mathbf{0}. \tag{3.1}$$

En segundo lugar, para poder distinguir las nubes de puntos entre sí, es también necesario que se extiendan sobre direcciones del espacio diferentes o, en el lenguaje de L2-PCA, que tengan distintas «direcciones principales». Matemáticamente, esto implica que las matrices de covarianza de las clases no pueden coincidir:

$$\boldsymbol{C}_1 \neq \boldsymbol{C}_2.$$

Vamos a suponer asimismo que las matrices de covarianza no tienen autovalores repetidos. De aquí se sigue que los correspondientes autovectores formarán un conjunto linealmente independiente.

Finalmente, se supondrá que los datos han sido estandarizados o «blanqueados». Como se ha visto en la Sección 4.2, «blanquear» es un preprocesamiento habitual que consiste en transformar linealmente $\mathbf{x}$ para que tenga como matriz de covarianza la matriz identidad. Matemáticamente:

$$\begin{aligned}\boldsymbol{C}_{\mathbf{x}} &= \mathbb{E}\{\mathbf{x}\mathbf{x}^{\top}\} \\ &= P(\mathscr{C}_1)\mathbb{E}\{\mathbf{x}\mathbf{x}^{\top} \mid \mathscr{C}_1\} + P(\mathscr{C}_2)\mathbb{E}\{\mathbf{x}\mathbf{x}^{\top} \mid \mathscr{C}_2\} \\ &= P(\mathscr{C}_1)\boldsymbol{C}_1 + P(\mathscr{C}_2)\boldsymbol{C}_2 \\ &= \boldsymbol{I}. \end{aligned} \tag{3.2}$$

Nótese el «blanqueamiento» siempre se puede llevar a cabo *sin ninguna pérdida de generalidad*. El procedimiento para realizar el «blanqueado», además, se expuso en la Sección 4.2.

3.2.1 Efectos del «blanqueamiento»

Antes de continuar, nos detendremos un poco más en la hipótesis de «blanqueado». El «blanqueado» es una operación habitual en el análisis de datos. Como también se explicó en la Sección 4.2, generaliza a las variables vectoriales el proceso de tipificación o estandarización (esto es, restar la media y dividir por la desviación típica) que se suele llevar a cabo con las variables escalares.

Además, cuando tenemos una mezcla de poblaciones, el blanqueado va a ser especialmente útil dado que relaciona las covarianzas de las clases de la siguiente manera:

Lema 3.2.1 *Tras el blanqueado,*

$$\boldsymbol{C}_1 = \frac{1}{P(\mathscr{C}_1)}\left[\boldsymbol{I} - P(\mathscr{C}_2)\boldsymbol{C}_2\right]. \tag{3.3}$$

La ecuación (3.3) se puede derivar fácilmente de la ecuación (3.2). De esta forma, cuando los datos han sido «blanqueados», las matrices de covarianza no son independientes una de otra; por el contrario, $\boldsymbol{C}_1$ está completamente determinada por $\boldsymbol{C}_2$ y viceversa. Ello reduce los grados de libertad del problema y lo simplifica: gracias a esta relación, se verá en la siguiente Sección que $\mathscr{C}_1$ y $\mathscr{C}_2$ se encuentran (aproximadamente) en subespacios ortogonales. Esta propiedad será fundamental para poder distinguir una población de otra.

3.3 Revisión del estado de la técnica

En esta Sección presentaremos las técnicas tradicionales para resolver el problema de la clasificación cuando las poblaciones se superponen.

3.3.1 El cociente de verosimilitud

En la estadística clásica, para asignar una observación $\boldsymbol{x}$ a una clase u otra se suele evaluar el llamado cociente de verosimilitud

$$\lambda = \frac{P(\mathscr{C}_1)\, f(\boldsymbol{x}|\mathscr{C}_1)}{P(\mathscr{C}_1)\, f(\boldsymbol{x}|\mathscr{C}_2)},$$

siendo $f(\boldsymbol{x}|\mathscr{C}_i)$ la función de densidad de probabilidad que tendría $\boldsymbol{x}$ si perteneciese a la clase i, mientras que $P(\mathscr{C}_i)$ es la probabilidad *a priori* de pertenecer a dicha clase. Si $\lambda > 1$, se considerará que $\boldsymbol{x}$ pertenece a la clase 1, por ser ésta más probable. Si $\lambda < 2$, la muestra se asigna a la clase 2. Por ejemplo, si $f(\boldsymbol{x}|\mathscr{C}_i)$ corresponde para ambas clases a una distribución Gaussiana, y se supone que las poblaciones tienen valores medios diferentes, con lo que las nubes de puntos están suficientemente separadas, se acaba obteniendo de esta forma el criterio conocido como «análisis lineal discriminante» de Fisher.

Cuando las poblaciones tienen el mismo valor medio, pero diferentes matrices de covarianza, que es el caso que nos ocupa en esta Tesis, es posible obtener con facilidad el cociente de máxima verosimilitud en algunos casos particulares. Por ejemplo, si ambas distribuciones son Gaussianas, es decir,

$$f(\boldsymbol{x}|\mathscr{C}_i) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{C}_i)^{-\frac{1}{2}}\, e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}_i^{-1}\boldsymbol{x}},$$

se obtiene, tras hacer algunos cálculos y suponiendo $P(\mathscr{C}_1) = P(\mathscr{C}_2)$, que

$$\log(\lambda) = -\frac{1}{2}\left(\boldsymbol{x}^\top \boldsymbol{C}_1^{-1}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{C}_2^{-1}\boldsymbol{x}\right),$$

por lo que $\boldsymbol{x}$ se asignará a la clase 1 si

$$\boldsymbol{x}^\top \boldsymbol{C}_2^{-1}\boldsymbol{x} > \boldsymbol{x}^\top \boldsymbol{C}_1^{-1}\boldsymbol{x}$$

y a la clase 2 en caso contrario.

En el ejemplo mostrado en la Figura 3.1, esta regla simplemente se transforma en

$$\boldsymbol{x}^\top \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \boldsymbol{x} > \boldsymbol{x}^\top \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \boldsymbol{x}$$

o, equivalentemente,

$$x_1 x_2 > 0.$$

Es decir, si las coordenadas de la observación tienen el mismo signo (ésta se encuentra en el primer o tercer cuadrantes), se asignará a la clase 1 ('azul'). En caso contrario (la muestra está en el segundo o cuarto cuadrantes), se asignará a la clase 2 ('roja'). Al evaluar esta regla sobre los puntos de la Figura 3.1 se obiene la matriz de confusión mostrada en la Figura 3.2. En promedio, la regla acierta el 67.2 % de las veces.

3.3.2 La transformada de Fukunaga-Koontz

La transformada Fukunaga-Koontz (FKT, del inglés «Fukunaga-Koontz transform») es un popular método de extracción de características utilizado en problemas de clasificación binaria [33]. La técnica consiste en proyectar los datos en direcciones en las que la varianza es mucho mayor para una clase que para la otra. Las reglas de clasificación posteriores se basan en explotar las diferencias entre las varianzas de las dos clases proyectadas.

Aunque la FKT puede usarse en todo tipo de problemas de clasificación, es especialmente útil cuando las dos clases comparten el mismo vector medio, lo que puede llevar a que las correspondientes nubes de puntos estén superpuestas [49, 48]. En este caso, como se muestra en la referencia [90], la FKT es equivalente al criterio óptimo de Chernoff presentado en [28]. Esto significa que la FKT conserva la distancia de Chernoff entre ambas poblaciones después de llevar a cabo la proyección [22]. La FKT también está estrechamente relacionada con el Análisis discriminante lineal (LDA), el Análisis discriminante cuadrático (QDA) y la descomposición en valores singulares generalizada (GSVD) [49, 82, 112]. Por todo ello, la FKT es una herramienta muy utilizada en problemas de clasificación

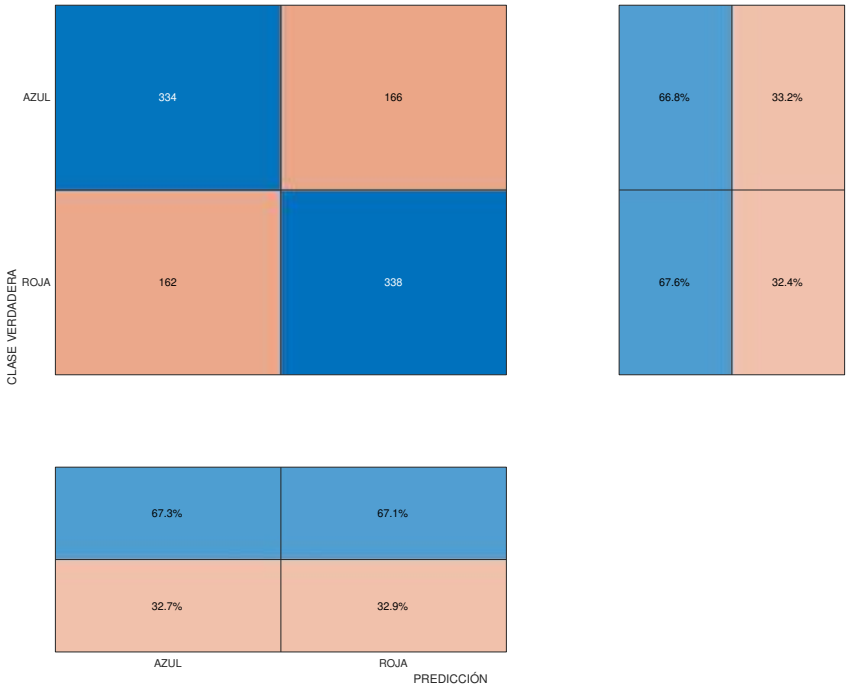


Figura 3.2 Matriz de confusión obtenida mediante la regla tradicional de clasificación (máxima verosimilitud).

de imágenes (donde también se conoce como el método de las «tuned based functions») [6, 10, 56, 70, 87] y en el procesamiento de señales EEG en interfaces cerebro-máquina (donde la FKT recibe el nombre de «common spatial patterns» o CSP) [63, 110], así como en otras áreas de interés [11, 47].

El fundamento de la FKT es el siguiente: sea $(\lambda, \boldsymbol{v})$ cualquier pareja autovalor-autovector de $\boldsymbol{C}_1$, es decir,

$$\boldsymbol{C}_1 \boldsymbol{v} = \lambda \, \boldsymbol{v}. \tag{3.4}$$

Usando (3.3), es decir, las propiedades del «blanqueado», se deduce fácilmente que

$$\frac{1}{P(\mathscr{C}_1)} \left[\boldsymbol{I} - P(\mathscr{C}_2) \boldsymbol{C}_2 \right] \boldsymbol{v} = \lambda \, \boldsymbol{v}$$

y por tanto

$$\boldsymbol{C}_2\boldsymbol{v} = \frac{1 - P(\mathscr{C}_1)\,\lambda}{P(\mathscr{C}_2)}\,\boldsymbol{v}.$$

En otras palabras, si $\boldsymbol{v}$ es un autovector de $\boldsymbol{C}_1$ con autovalor asociado λ, entonces $\boldsymbol{v}$ también es un autovector de $\boldsymbol{C}_2$ con autovalor

$$\mu = \frac{1 - P(\mathscr{C}_1)\,\lambda}{P(\mathscr{C}_2)}.$$

Es más, como

$$\frac{d\mu}{d\lambda} = -\frac{P(\mathscr{C}_1)}{P(\mathscr{C}_2)} < 0,$$

esta transformación es estrictamente decreciente. Ello implica que si los autovalores λ_i de $\boldsymbol{C}_1$ están ordenados de mayor a menor, es decir,

$$\lambda_1 > \lambda_2 > \ldots > \lambda_p > 0,$$

se deduce que los autovalores correspondientes de $\boldsymbol{C}_2$ están ordenados a la inversa, es decir

$$0 < \mu_1 < \mu_2 < \ldots < \mu_p,$$

lo que implica que los autovectores dominantes de $\boldsymbol{C}_1$, es decir, los asociados a los mayores autovalores, son los autovectores menos dominantes de $\boldsymbol{C}_2$ y viceversa. Recuérdese, además, que los autovalores de las matrices de covarianza tienen la propiedad de ser siempre positivos. Recuérdese también que los autovectores son siempre *ortogonales* entre sí y *representan las direcciones principales de los datos*.

Por tanto, gracias al «blanqueado», observamos que las clases se extienden o distribuyen, aproximadamente, en direcciones perpendiculares, lo que va a facilitar la tarea de distinguirlas. En el lenguaje del análisis de componentes principales (L2-PCA) [54], se puede decir que las direcciones donde los datos de la clase 1 tienen mayor variabilidad serán también las direcciones donde la clase 2 varía menos, y las direcciones de mayor variabilidad para clase 2 son las de menor variabilidad para la clase 1 (ver Sección 2.2.1). Este efecto se observa perfectamente, como ilustración, en

la Figura 3.1.

Además, como ya se explicó (en parte) en la Sección 2.2.4, la distancia entre los puntos de datos de una clase dada y el subespacio generado por los autovectores dominantes de la correspondiente matriz de covarianza es *mínima* [54]. En otras palabras: si

$$\boldsymbol{V}_1 = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M]$$

contiene los autovectores asociados a los autovalores $\lambda_1 > \lambda_2 > \ldots > \lambda_M$ de $\boldsymbol{C}_1$, entonces la *mejor aproximación* que se puede obtener de un vector $\boldsymbol{x} \in \mathbb{R}^p$ de la clase 1 como combinación lineal de $M < p$ vectores independientes es, precisamente,

$$\hat{\boldsymbol{x}}_{\mathscr{C}_1} = \boldsymbol{V}_1 \boldsymbol{V}_1^\top \boldsymbol{x}.$$

De igual manera, si

$$\boldsymbol{V}_2 = \left[\boldsymbol{v}_p, \ldots, \boldsymbol{v}_{p-M+1}\right]$$

contiene los autovectores asociados a los autovalores $\mu_p > \mu_2 > \ldots > \mu_M$ de $\boldsymbol{C}_{p-M+1}$ de $\boldsymbol{C}_2$, entonces la mejor aproximación que se puede obtener de un vector $\boldsymbol{x}$ perteneciente a la clase 2 en un espacio de M dimensiones es:

$$\hat{\boldsymbol{x}}_{\mathscr{C}_2} = \boldsymbol{V}_2 \boldsymbol{V}_2^\top \boldsymbol{x}.$$

Nótese además que, suponiendo que $M < p - M + 1$, los autovectores $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M$ y $\boldsymbol{v}_p, \ldots, \boldsymbol{v}_{p-M+1}$ son distintos y, por lo tanto, ortogonales entre sí. Por ello, un vector $\boldsymbol{x}$ perteneciente a la clase i será «parecido» a $\hat{\boldsymbol{x}}_{\mathscr{C}_i}$; pero, al mismo tiempo, $\boldsymbol{x}$ será aproximadamente perpendicular a $\hat{\boldsymbol{x}}_{\mathscr{C}_j}$ con $i \neq j$.

La extracción y clasificación de características puede estar basada en la explotación de todas estas propiedades. La FKT transforma cada observación proyectándola sobre los M autovectores más dominantes de $\boldsymbol{C}_1$, donde M se selecciona en la práctica de forma empírica. Si la distancia entre la observación y su proyección es menor que un umbral predefinido, podemos deducir la presencia de una muestra de $\mathscr{C}_1$. Por el contrario, si la observación puede representarse de forma más precisa en el subespacio generado por los autovectores menos dominantes de $\boldsymbol{C}_1$, la asignaríamos

a la clase $\mathscr{C}_2$. Se han propuesto asimismo diversas variantes de esta idea básica [6, 10, 56, 70, 87]. La más sencilla podría ser, simplemente, asignar $\boldsymbol{x}$ a $\mathscr{C}_1$ si

$$\begin{aligned} &\|\boldsymbol{x}-\hat{\boldsymbol{x}}_{\mathscr{C}_1}\| < \|\boldsymbol{x}-\hat{\boldsymbol{x}}_{\mathscr{C}_2}\| \\ &\Rightarrow \|\boldsymbol{V}_2^\top \boldsymbol{x}\| < \|\boldsymbol{V}_1^\top \boldsymbol{x}\| \end{aligned}$$

y a $\mathscr{C}_2$ en caso contrario.

En la situación mostrada en la Figura 3.1, la aplicación de esta regla lleva a asignar una muestra a la clase 'azul'si

$$|x_1 - x_2| < |x_1 + x_2|,$$

es decir, si la muestra está en el primer o cuarto cuadrante. De esta forma, se obtiene el mismo resultado (en este caso, no tiene por qué ocurrir siempre) que aplicando la técnica desarrollada en la Sección 3.3 a partir del cociente de verosimilitud.

3.4 L1-PCA y la transformada de Fukunaga-Koontz

Para calcular las «direcciones principales» de cada clase, tanto el enfoque tradicional (ver Sección 3.3) como la transformada de Fukunaga-Koontz necesitan conocer *de antemano* las matrices de covarianza $\boldsymbol{C}_i$. En la práctica, éstas se obtienen a partir de un conjunto amplio de observaciones *correctamente etiquetadas*, es decir, cuya clase es conocida. Por esta razón, ambas técnicas son *supervisadas*. Ello supone una evidente limitación: en primer lugar, los algoritmos cometerán errores si las muestras de entrenamiento no han sido etiquetadas de forma correcta; en segundo lugar, lo que es peor, no siempre será posible disponer de conjuntos de entrenamiento.

Como alternativa, en esta Tesis mostraremos que también se puede realizar la clasificación mediante una técnica *no supervisada* que se basa en L1-PCA. En esta Sección, concretamente, se demostrará esta afirmación en el caso Gaussiano. Es decir, cuando la probabilidad de $\mathbf{x}$ condicionada a la

clase i, $f(\boldsymbol{x}|\mathscr{C}_i)$, es Gaussiana:

$$f(\boldsymbol{x}|\mathscr{C}_i) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{C}_i)^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}_i^{-1}\boldsymbol{x}}, \quad i = 1,2. \tag{3.5}$$

Los modelos gaussianos están justificados en la práctica por su sencillez y capacidad para producir resultados precisos incluso cuando los datos no sigan realmente esa distribución.

La distribución global de $\mathbf{x}$ vendrá dada por la mezcla

$$f(\boldsymbol{x}) = P(\mathscr{C}_1)\, f(\boldsymbol{x}|\mathscr{C}_1) + P(\mathscr{C}_2)\, f(\boldsymbol{x}|\mathscr{C}_2).$$

Sea $\mathrm{y} = \boldsymbol{a}^\top \mathbf{x}$ la proyección de $\mathbf{x}$ sobre $\boldsymbol{a} \in \mathbb{R}^p$. Como únicamente la dirección de este vector es importante, supondremos que $\boldsymbol{a}$ tiene longitud unidad. Aplicando las propiedades de la distribución normal, se deduce que la función de densidad de probabilidad de y es una mezcla de Gaussianas, es decir,

$$f(y) = \sum_{k=1,2} \frac{P(\mathscr{C}_k)}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{y^2}{2\sigma_k^2}\right), \tag{3.6}$$

donde

$$\sigma_k^2 = \boldsymbol{a}^\top \boldsymbol{C}_k \boldsymbol{a}.$$

Consideremos ahora minimizar

$$D(\boldsymbol{a}) = \mathbb{E}\{|\mathrm{y}|\} = \mathbb{E}\{|\boldsymbol{a}^\top \mathbf{x}|\} \tag{3.7}$$

en el conjunto de todas las posibles proyecciones. El criterio D, propio de L1-PCA, está también relacionado con la «desviación media absoluta» de la distribución y nos proporciona una medida de la dispersión de los datos en torno a su media. Nótese además que, en L1-PCA tradicional, se *maximiza* D [65, 75], mientras que nuestra propuesta es justo la contraria.

Las soluciones del problema de optimización con restricciones

$$\min_{\boldsymbol{a}} D(\boldsymbol{a}) \text{ con la condición } \|\boldsymbol{a}\|^2 = 1 \tag{3.8}$$

verifican

$$\nabla_{\boldsymbol{a}} D(\boldsymbol{a}) = \ell \nabla_{\boldsymbol{a}} \|\boldsymbol{a}\|^2, \tag{3.9}$$

donde ℓ es un multiplicador de Lagrange y $\nabla_{\boldsymbol{a}}$ representa el gradiente con respecto a $\boldsymbol{a}$. Suponiendo (3.6), y después de algunas operaciones que se detallan en el Apéndice 3.A, obtenemos que las soluciones de (3.9) satisfacen

$$\sum_{k=1}^{2} \frac{P(\mathscr{C}_k)}{\sigma_k} \boldsymbol{C}_k \boldsymbol{a} = \left(\sum_{k=1}^{2} P(\mathscr{C}_k) \sigma_k \right) \boldsymbol{a}. \tag{3.10}$$

Usando ahora la suposición de «blanqueado» (3.3), es decir,

$$\boldsymbol{C}_2 = \frac{1}{P(\mathscr{C}_2)} \left[\boldsymbol{I} - P(\mathscr{C}_1) \boldsymbol{C}_1 \right],$$

obtenemos:

$$\left(\frac{\sigma_2 - \sigma_1}{\sigma_1} \right) P(\mathscr{C}_1) \boldsymbol{C}_1 \boldsymbol{a} = \left[\left(\sum_{k=1}^{2} P(\mathscr{C}_k) \sigma_k \right) \sigma_2 - 1 \right] \boldsymbol{a}. \tag{3.11}$$

Reemplazando el '1'que aparece a la derecha de la expresión por

$$1 = \boldsymbol{w}^\top \boldsymbol{C}_{\mathbf{x}} \boldsymbol{w} = \sum_{k=1}^{2} P(\mathscr{C}_k) \sigma_k^2,$$

que también se deduce de (3.2), y simplificando términos, la ecuación se convierte en:

$$(\sigma_2 - \sigma_1) \boldsymbol{C}_1 \boldsymbol{a} = (\sigma_2 - \sigma_1) \sigma_1^2 \boldsymbol{a}. \tag{3.12}$$

Por lo tanto, aparte de la solución $\sigma_1 = \sigma_2$ (que se corresponde con un máximo de la función objetivo, como se demuestra en 3.B)), encontramos que:

Lema 3.4.1 *Bajo la hipótesis del modelo* (3.6)*, los autovectores de* $\boldsymbol{C}_1$ *(o* $\boldsymbol{C}_2$*) son puntos estacionarios (es decir, en los que se anula la derivada) de la función objetivo* (3.8)*.*

Este resultado se complementa con el siguiente teorema, que permite una mayor precisión:

Lema 3.4.2 *Dado un vector aleatorio* $\mathbf{x}$ *p-dimensional, cuya distribución es una mezcla de dos distribuciones gaussianas, con media cero y matrices de covarianza* $\boldsymbol{C}_1 \neq \boldsymbol{C}_2$ *que verifican la condición de «blanqueamiento»* (3.3)*, las direcciones que* minimizan *D son los autovectores asociados a los autovalores máximo y mínimo de* $\boldsymbol{C}_1$ *(o* $\boldsymbol{C}_2$*). Los restantes autovectores son puntos de silla y pueden ser determinados mediante un procedimiento de deflación.*

En el Apéndice 3.B se puede ver la demostración del teorema anterior. Este resultado sugiere, por tanto, que es posible de llevar a cabo la FKT *de forma no supervisada* minimizando D. El estadístico D, por tanto, cuenta con propiedades discriminativas: este resultado, hasta donde sabemos, no ha sido publicado con anterioridad y es completamente novedoso. Finalmente, aunque el teorema presupone distribuciones Gaussianas, la conclusión se mantiene incluso cuando la distribución de los datos se desvía apreciablemente de esta hipótesis, como veremos en el Capítulo dedicado a los experimentos.

Apéndice

3.A Demostración de la ecuación (3.10)

La función objetivo se define como sigue:

$$\begin{aligned} D(\boldsymbol{a}) &= \mathbb{E}\{|\mathrm{y}|\} \\ &= \int_{-\infty}^{\infty} |y|\, f(y)\, \mathrm{d}y \\ &= \int_{0}^{\infty} y\, f(y)\, \mathrm{d}y - \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y, \end{aligned}$$

y, utilizando que la media es cero, es decir,

$$\begin{aligned} \mathbb{E}\{\mathrm{y}\} &= \int_{0}^{\infty} y\, f(y)\, \mathrm{d}y + \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y \\ &= 0 \Rightarrow \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y \\ &= -\int_{0}^{\infty} y\, f(y)\, \mathrm{d}y, \end{aligned}$$

se obtiene

$$D(\boldsymbol{a}) = \mathbb{E}\{|\mathrm{y}|\} = 2\int_{0}^{\infty} y\, f(y)\, \mathrm{d}y$$

Usando el modelo Gaussiano (3.6), es decir,

$$f(y) = \sum_{k=1,2} \frac{P(\mathscr{C}_k)}{\sqrt{2\pi\sigma_k^2}\exp\left(-\frac{y^2}{2\sigma_k^2}\right)},$$

donde

$$\sigma_k^2 = \boldsymbol{a}^\top \boldsymbol{C}_k \boldsymbol{a}, \tag{3.13}$$

la función objetivo se puede reescribir como:

$$\begin{aligned} D(\boldsymbol{a}) &= 2\sum_{k=1}^{2} P(\mathscr{C}_k) \int_0^\infty \frac{y}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{y^2}{2\sigma_k^2}\right) \mathrm{d}y \\ &\overset{(a)}{=} \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} P(\mathscr{C}_k)\,\sigma_k, \end{aligned} \tag{3.14}$$

donde la igualdad (a) es consecuencia de la siguiente, que aparece en la referencia [37]:

$$\int y\, e^{-\frac{y^2}{2\sigma^2}}\,\mathrm{d}y = -\sigma^2 e^{-\frac{y^2}{2\sigma^2}} + \text{ constante de integración.}$$

Los puntos críticos del problema de optimización (3.8) verifican

$$\nabla_{\boldsymbol{a}} D(\boldsymbol{a}) = \ell \nabla_{\boldsymbol{a}} \|\boldsymbol{a}\|^2. \tag{3.15}$$

donde ℓ es un multiplicador de Lagrange y $\nabla_{\boldsymbol{a}}$ representa el gradiente respecto a $\boldsymbol{a}$.

Vemos que (3.14) es una función de σ_1 y σ_2. Es más sencillo calcular $\nabla_{\boldsymbol{a}}\sigma_k$ si utilizamos la identidad

$$\nabla_{\boldsymbol{a}}\sigma_k^2 = 2\sigma_k \nabla_{\boldsymbol{a}}\sigma_k.$$

y, utilizando (3.13), también

$$\nabla_{\boldsymbol{a}}\sigma_k^2 = \nabla_{\boldsymbol{a}}\left(\boldsymbol{a}^\top \boldsymbol{C}_k \boldsymbol{a}\right) = 2\boldsymbol{C}_k\boldsymbol{a}.$$

Combinando ambas fórmulas, obtenemos

$$\nabla_{\boldsymbol{a}}\sigma_k = \nabla_{\boldsymbol{a}}\sigma_k^2/(2\sigma_k) = \boldsymbol{C}_k\boldsymbol{a}/\sigma_k.$$

Llevando este resultado a la ecuación (3.14), se sigue que

$$\nabla_{\boldsymbol{a}}D(\boldsymbol{a}) = \sqrt{\frac{2}{\pi}}\sum_{k=1}^{2}\frac{P(\mathscr{C}_k)}{\sigma_k}\boldsymbol{C}_k\boldsymbol{a}. \tag{3.16}$$

Procediendo de forma similar, también se obtiene que

$$\nabla_{\boldsymbol{a}}\|\boldsymbol{a}\|^2 = \nabla_{\boldsymbol{a}}\left(\boldsymbol{a}^\top\boldsymbol{a}\right) = 2\boldsymbol{a}. \tag{3.17}$$

Por lo tanto, (3.15) se puede reescribir como

$$\sqrt{\frac{2}{\pi}}\sum_{k=1}^{2}\frac{P(\mathscr{C}_k)}{\sigma_k}\boldsymbol{C}_k\boldsymbol{a} = 2\ell\,\boldsymbol{a}. \tag{3.18}$$

El valor del multiplicador de Lagrange ℓ se puede obtener ahora premultiplicando (3.18) por $\boldsymbol{a}^\top$, tras lo cuál hemos de utilizar (3.13) y $\boldsymbol{a}^\top\boldsymbol{a} = 1$. Tras todo esto, obtenemos:

$$\ell = \frac{1}{\sqrt{2\pi}}\sum_{k=1}^{2}P(\mathscr{C}_k)\sigma_k, \tag{3.19}$$

y, sustituyendo finalmente en (3.18), se obtiene el resultado deseado:

$$\sum_{k=1}^{2}\frac{P(\mathscr{C}_k)}{\sigma_k}\boldsymbol{C}_k\boldsymbol{a} = \left(\sum_{k=1}^{2}P(\mathscr{C}_k)\sigma_k\right)\boldsymbol{a}.$$

3.B Demostración del teorema 3.4.2

Vamos a estudiar ahora si los autovectores de $\boldsymbol{C}_1$ y $\boldsymbol{C}_2$ se corresponden con máximos, mínimos o puntos de silla de la función objetivo basada en la norma L1. Empezaremos calculando la matriz hessiana (o matriz de deriva-

das parciales de segundo orden) de $\mathbb{E}\{|\mathrm{y}|\}$. A partir de la ecuación (3.16), y después de hacer algunos cálculos, esta matriz hessiana resulta ser:

$$\begin{aligned}\Delta_{\boldsymbol{a}}^2\mathbb{E}\{|\mathrm{y}|\} &= \sqrt{\frac{2}{\pi}}\sum_{k=1}^{2}\frac{P(\mathscr{C}_k)}{\sigma_k}\left[\boldsymbol{C}_k-\frac{1}{\sigma_k^2}\boldsymbol{C}_k\boldsymbol{w}(\boldsymbol{C}_k\boldsymbol{w})^\top\right]\\ &\overset{(a)}{=}\sqrt{\frac{2}{\pi}}\sum_{k=1}^{2}\frac{P(\mathscr{C}_k)}{\sigma_k}\left[\boldsymbol{C}_k-\sigma_k^2\boldsymbol{w}\boldsymbol{w}^\top\right],\end{aligned} \tag{3.20}$$

donde (a) viene de $\boldsymbol{C}_k\boldsymbol{w}=\sigma_k^2\boldsymbol{a}$. Operando de forma similar, la matriz hessiana de la restricción $\|\boldsymbol{w}\|^2=1$ es

$$\Delta_{\boldsymbol{a}}^2\|\boldsymbol{w}\|^2 = 2\boldsymbol{I}. \tag{3.21}$$

Finalmente, la matriz hessiana del Lagrangiano es igual a

$$\Delta_{\boldsymbol{a}}^2 L = \Delta_{\boldsymbol{a}}^2\mathbb{E}\{|\mathrm{y}|\} - \ell\,\Delta_{\boldsymbol{a}}^2\|\boldsymbol{w}\|^2, \tag{3.22}$$

siendo ℓ el multiplicador de Lagrange. Ahora, utilizaremos el siguiente resultado del libro [16, Chap. 20], el cuál reescribimos en nuestra propia notación:

Lema 3.B.1 *Sea $\boldsymbol{a}$ un punto crítico (máximo, mínimo o punto de silla) de $\mathbb{E}\{|\mathrm{y}|\}$ sujeto a la restricción $\|\boldsymbol{a}\|^2=1$. Si para todos los vectores unitarios $\boldsymbol{v}$ tales que*

$$\boldsymbol{v}^\top\nabla_{\boldsymbol{a}}\|\boldsymbol{a}\|^2 = 2\boldsymbol{v}^\top\boldsymbol{a}=0, \tag{3.23}$$

se cumple que

$$\boldsymbol{v}^\top(\Delta_{\boldsymbol{a}}^2 L)\boldsymbol{v} > 0, \tag{3.24}$$

entonces $\boldsymbol{a}$ es un mínimo local de la función objetivo. Para los máximos locales, la condición resulta ser $\boldsymbol{v}^\top(\Delta_{\boldsymbol{a}}^2 L)\boldsymbol{v}<0$.

Utilizando (3.19), obtenemos

$$\boldsymbol{v}^\top\Delta_{\boldsymbol{a}}^2 L\,\boldsymbol{v} = \sqrt{\frac{2}{\pi}}\left(\sum_{k=1}^{2}\frac{s_k^2-\sigma_k^2}{\sigma_k}P(\mathscr{C}_k)\right), \tag{3.25}$$

donde $\sigma_k^2 = \boldsymbol{a}^\top \boldsymbol{C}_k \boldsymbol{a}$ y $s_k^2 = \boldsymbol{v}^\top \boldsymbol{C}_k \boldsymbol{v}$. Analicemos el siguiente término con más detalle:

$$\frac{s_2^2 - \sigma_2^2}{\sigma_2} P(\mathscr{C}_2) \tag{3.26}$$

Por una parte, la hipótesis de «blanqueado» (3.2) nos permite escribir:

$$1 = \boldsymbol{v}^\top \boldsymbol{C} \boldsymbol{v} = \sum_{k=1}^{2} P(\mathscr{C}_k) s_k^2 \Rightarrow s_2^2 P(\mathscr{C}_2) = 1 - P(\mathscr{C}_1) s_1^2, \tag{3.27}$$

y, por la misma razón,

$$\sigma_2^2 P(\mathscr{C}_2) = 1 - P(\mathscr{C}_1) \sigma_1^2. \tag{3.28}$$

Utilizando estos resultados, obtenemos

$$\frac{s_2^2 - \sigma_2^2}{\sigma_2} P(\mathscr{C}_2) = -\frac{(s_1^2 - \sigma_1^2)}{\sigma_2} P(\mathscr{C}_1). \tag{3.29}$$

Por lo tanto, sustituyendo en (3.25), se sigue que

$$\boldsymbol{v}^\top \Delta_{\boldsymbol{a}}^2 L \boldsymbol{v} = \sqrt{\frac{2}{\pi}} (s_1^2 - \sigma_1^2) \left(\frac{1}{\sigma_1} - \frac{1}{\sigma_2} \right) P(\mathscr{C}_1). \tag{3.30}$$

Sean $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p$ los autovectores de $\boldsymbol{C}_1$, con $\lambda_1 > \lambda_2 > \ldots > \lambda_p$ los correspondientes autovalores. Consideremos los siguientes casos:

1. Si $\boldsymbol{a} = \boldsymbol{a}_1$ es el autovector dominante de $\boldsymbol{C}_1$, entonces, utilizando las propiedades del cociente de Rayleigh [35],

$$\boldsymbol{a} = \arg\max_{\boldsymbol{v}} \boldsymbol{v}^\top \boldsymbol{C}_1 \boldsymbol{v}, \tag{3.31}$$

y, por lo tanto, $\sigma_1 > s_1$ y $\sigma_1 > \sigma_2$. Se sigue que

$$\boldsymbol{v}^\top \Delta_{\boldsymbol{a}}^2 L \boldsymbol{v} > 0, \tag{3.32}$$

y, por ello, $\boldsymbol{a}$ es un mínimo de la función objetivo basada en la norma L1.

2. De manera similar, si $\boldsymbol{a} = \boldsymbol{a}_p$ es el autovector asociado al menor autovalor de $\boldsymbol{C}_1$, se deduce de nuevo de las propiedades del cociente de Rayleigh [35] que $\sigma_1 < s_1$ y $\sigma_1 < \sigma_2$. Por tanto, $\boldsymbol{v}^\top \Delta_{\boldsymbol{a}}^2 L \boldsymbol{v} > 0$ y $\boldsymbol{a}_p$ sigue siendo un mínimo.

3. Si $\boldsymbol{a} = \boldsymbol{a}_i$, $1 < i < p$, es cualquiera de los restantes autovectores, el signo de (3.30) cuando $\boldsymbol{v} = \boldsymbol{a}_1$ va a ser distinto del signo de $\boldsymbol{v} = \boldsymbol{a}_p$ y, además, tanto $\boldsymbol{a}_1$ como $\boldsymbol{a}_p$ cumplen la restricción (3.23). Por tanto, en la vecindad de $\boldsymbol{a} = \boldsymbol{a}_i$, la función objetivo se incrementa si nos movemos en una cierta dirección y se decrementa si lo hacemos en otra. Por lo tanto, $\boldsymbol{a}$ es un punto de silla. Dicho esto, si resolvemos el problema de optimización de nuevo con la restricción adicional de que $\boldsymbol{a}$ sea ortogonal a $\boldsymbol{a}_1$ y $\boldsymbol{a}_p$, entonces podemos demostrar fácilmente que $\boldsymbol{a}_2$ y $\boldsymbol{a}_{p-1}$ van a ser los nuevos mínimos de la función de coste. En conclusión, todos los autovectores pueden ser obtenidos mediante deflación, es decir, calculados minimizando la función objetivo con la restricción de que han de ser ortogonales a todos los que hemos calculado previamente.

Finalmente, notemos que la ecuación (3.12) también tiene como solución $\sigma_1 = \sigma_2$. Veamos brevemente que esta solución se corresponde con el máximo absoluto de la función objetivo L1. Sea $\boldsymbol{b} = (\sigma_1, \sigma_2)^\top$, $\mathbf{1} = (1,1)^\top$ y $\boldsymbol{D} = \mathrm{diag}(P(\mathscr{C}_1), P(\mathscr{C}_2))$. Definamos también el producto escalar ponderado $(\boldsymbol{b}, \mathbf{1})_{\boldsymbol{D}} = \boldsymbol{b}^\top \boldsymbol{D} \mathbf{1}$. Entonces, usando la desigualdad de Cauchy-Schwarz,

$$(\boldsymbol{b}, \mathbf{1})_{\boldsymbol{D}} \leq \sqrt{(\boldsymbol{b}, \boldsymbol{b})_{\boldsymbol{D}} (\mathbf{1}, \mathbf{1})_{\boldsymbol{D}}} = \sqrt{\sum_{i=1,2} P(\mathscr{C}_i)\, \sigma_i^2} \overset{(a)}{=} 1$$

donde (a) es una consecuencia de la hipótesis de «blanqueado» (3.2). Es ahora sencillo demostrar la siguiente desigualdad, que acota (3.14):

$$\mathbb{E}\{|\mathrm{y}|\} = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} P(\mathscr{C}_k)\, \sigma_k = \sqrt{\frac{2}{\pi}}\, (\boldsymbol{b}, \mathbf{1})_{\boldsymbol{D}} \leq \sqrt{\frac{2}{\pi}},$$

donde la igualdad se da si y solo si $\boldsymbol{b}$ es proporcional a $\mathbf{1}$, lo que implica que $\sigma_1 = \sigma_2$. Esto completa la demostración.

4 Minimización de la norma L1

4.1 Introducción

Se va a proponer un algoritmo completamente novedoso para determinar direcciones perpendiculares que minimizan la norma L1 mínima. El algoritmo es completamente *no supervisado*, es decir, no precisa disponer de datos etiquetados o secuencias de entrenamiento. La estructura de este Capítulo es la siguiente: en la Sección 4.2 se revisarán las hipótesis de partida, que se pueden satisfacer sin más que hacer un simple-preprocesado de los datos; en la Sección 4.3 se presentará la función objetivo propuesta; en la Sección 4.4 repasaremos las herramientas matemáticas para la optimización de una función con restricciones de ortogonalidad; finalmente, en la Sección 4.5 presentaremos el algoritmo propuesto.

4.2 Preprocesamiento

En primer lugar, el algoritmo requiere blanquear o estandarizar los datos. Para cumplir esta condición, se necesitará llevar a cabo el siguiente procesamiento previo: dado un vector aleatorio «coloreado» (es decir, no blanco) $\mathbf{x}_c \in \mathbb{R}^p$, que supondremos que tiene media cero, los datos blanqueados $\mathbf{x}$

se pueden obtener, por ejemplo, como sigue [61]:

$$\mathbf{x} = \boldsymbol{D}^{-1/2}\boldsymbol{V}^{\top}\mathbf{x}_c,$$

donde $\boldsymbol{V}$ es la matriz cuyas columnas son los autovectores de $\mathbb{E}\{\mathbf{x}_c\,\mathbf{x}_c^{\top}\}$, y $\boldsymbol{D}$ es la matriz diagonal de sus autovalores (hay que tener en cuenta que existen otras aproximaciones para el blanqueado que son igualmente válidas [61]). Finalmente, puede comprobarse fácilmente que

$$\boldsymbol{C}_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\,\mathbf{x}^{\top}\} = \boldsymbol{I}.$$

4.3 Función objetivo

Para *maximizar* el criterio $D(\boldsymbol{a})$ basado en la norma L1 que se definió en las ecuaciones (3.7)–(3.8), podemos utilizar cualquiera de los algoritmos que han ya sido propuestos en [65, 75, 76]. Desafortunadamente, por cómo se han definido, ninguno de estos métodos se puede convertir en un algoritmo de *minimización*, que es lo que realmente deseamos llevar a cabo. Por lo tanto, optaremos por un procedimiento completamente nuevo basado en el método del gradiente.

Una posibilidad, que surge directamente del teorema 3.4.2 y su demostración, sería la siguiente: como los autovectores de las matrices de covarianza tienen la propiedad de ser siempre ortogonales, podríamos minimizar $D(\boldsymbol{a})$ sucesivas veces bajo la restricción de que la dirección obtenida en cada minimización fuese ortogonal a todas las calculadas previamente. Sin embargo, esta idea tiene como desventaja que los errores cometidos en el cálculo de una de estas direcciones se acumularían sobre todas las que se calculasen posteriormente. Alternativamente, consideramos la función objetivo

$$J(\boldsymbol{A}) = \sum_{i=1,\ldots,p} D(\boldsymbol{a}_i),$$

donde $\boldsymbol{A}$ es la matriz $[\boldsymbol{a}_1,\ldots,\boldsymbol{a}_p]$ que contiene los vectores de proyección. A continuación nos plantearemos minimizar esta función imponiendo que

las distintas direcciones sean ortogonales entre sí:

$$\begin{aligned} &\min_{\boldsymbol{a}} J(\boldsymbol{A}) \\ &\text{sujeto a } \boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}. \end{aligned} \tag{4.1}$$

De esta manera, todos los vectores $\boldsymbol{a}_i$ se calculan simultáneamente, sin que ninguno sea privilegiado sobre los demás.

4.3.1 Minimización «fallida» de la función objetivo

A fin de encontrar el valor de $\boldsymbol{A}$ para el que (4.1) es mínimo, podemos utilizar el método del gradiente. Este método consiste en desplazarse siempre en la dirección opuesta a la de máxima variación de J, es decir en la dirección opuesta al gradiente [53]:

$$\boldsymbol{A}_{n+1} = \boldsymbol{A}_n - \mu\, \partial J(\boldsymbol{A}_n) \tag{4.2}$$

donde $\boldsymbol{A}_n$ denota el valor de $\boldsymbol{A}$ en la enésima iteración y $\partial J(\boldsymbol{A})$ es la matriz de derivadas $\partial J(\boldsymbol{A})$ de $J(\boldsymbol{A})$ en $\boldsymbol{A}$, es decir,

$$(\partial J(\boldsymbol{A}))_{ij} = \frac{\partial J(\boldsymbol{A})}{\partial \boldsymbol{A}_{ij}}.$$

De esta forma, pretendemos acercarnos iteración a iteración (disminuyendo el valor de la función en cada iteración) a un mínimo.

Sin embargo, el algoritmo (4.2) tiene un grave inconveniente: a la hora de formularlo, no se ha contemplado en ningún momento la restricción de ortogonalidad. En otras palabras, no se puede asegurar que

$$\boldsymbol{A}_{n+1}^\top \boldsymbol{A}_{n+1} = \boldsymbol{I},$$

ni siquiera aunque $\boldsymbol{A}_n^\top \boldsymbol{A}_n = \boldsymbol{I}$. Para incorporar la restricción al método del gradiente, es necesario hacer unas consideraciones geométricas previas, las cuales expondremos en la siguiente Sección.

4.4 El gradiente en la variedad de Stiefel

En el problema de optimización planteado anteriormente, la restricción

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}$$

nos remite a la llamada variedad de Stiefel (*Stiefel manifold* en inglés) [29, 31, 68]. Ésta se define como el conjunto formado por las matrices

$$\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{p \times n}, \tag{4.3}$$

cuyas columnas tienen longitud unidad y son perpendiculares entre sí. Es decir la variedad se puede definir como

$$V_{p,n} = \{\boldsymbol{A} \in \mathbb{R}^{p \times n} : \boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_n\},$$

siendo $\boldsymbol{I}_n$ la matriz identidad $n \times n$. Si las matrices son cuadradas, es decir, $p = n$, lo que supondremos de ahora en adelante, la variedad de Stiefel da paso al llamado «grupo ortogonal especial».

Sea $\boldsymbol{A}(t)$ una función diferenciable, con $\boldsymbol{A}(0) = \boldsymbol{A}$, tal que

$$\boldsymbol{A}(t) \in V_{p,p},$$

es decir, la matriz $\boldsymbol{A}(t)$ es ortogonal para todo t. Podemos considerar, de forma abstracta, que $\boldsymbol{A}(t)$ define un camino dentro de la variedad, por el cuál podemos desplazarnos sin salir de ella dando valores a t. Calculemos ahora la derivada

$$\dot{\boldsymbol{A}}(t) = \frac{d\boldsymbol{A}(t)}{dt} = \lim_{\Delta t \to 0} \frac{\boldsymbol{A}(t + \Delta t) - \boldsymbol{A}(t)}{\Delta t}.$$

Haciendo un desarrollo de Taylor de primer orden se obtiene

$$\boldsymbol{A}(t + \Delta t) \approx \boldsymbol{A}(t) + \dot{\boldsymbol{A}}(t)\Delta t.$$

La interpretación es la siguiente: podemos ir de $\boldsymbol{A}(t)$ al punto inmediatamente posterior, $\boldsymbol{A}(t+\Delta t)$, sin más que movernos una corta distancia en la dirección señalada por $\dot{\boldsymbol{A}}(t)$. Por ello, $\dot{\boldsymbol{A}}(t)$ apunta en la dirección de un camino que atraviesa la variedad. Geométricamente, además, esta dirección debe ser tangente a dicha ruta en el punto $\boldsymbol{A}(t)$. Por esta razón, cuando $t=0$, el conjunto formado por todas las derivadas, para todas las posibles curvas que pasan por $\boldsymbol{A}(0)=\boldsymbol{A}$, recibe el nombre de *espacio tangente* a la variedad en el punto $\boldsymbol{A}$.

Para estudiar este espacio tangente, partimos de la condición de ortogonalidad:

$$\boldsymbol{A}(t)^\top \boldsymbol{A}(t) = \boldsymbol{I}.$$

Derivando esta ecuación, vemos que se cumple que

$$\dot{\boldsymbol{A}}(t)^\top \boldsymbol{A}(t) + \boldsymbol{A}(t)^\top \dot{\boldsymbol{A}}(t) = \boldsymbol{0}.$$

De hecho, como $\boldsymbol{A}(0)=\boldsymbol{A}$ y llamando $\boldsymbol{M}$ a $\dot{\boldsymbol{A}}(0)$ se puede deducir formalmente que el espacio tangente en $\boldsymbol{A}$ está formado por el siguiente conjunto de matrices:

$$\mathscr{T}_{\boldsymbol{A}} V_{p,p} = \{\boldsymbol{M} \in \mathbb{R}^{p\times p} : \boldsymbol{M}^\top \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{M} = \boldsymbol{0}\}.$$

Por otra parte, el producto escalar entre dos matrices $\boldsymbol{Q}_1$ y $\boldsymbol{Q}_2$ se define habitualmente como [29]

$$\langle \boldsymbol{Q}_1 | \boldsymbol{Q}_2 \rangle = \sum_{i,j} \boldsymbol{Q}_1(i,j)\, \boldsymbol{Q}_2(i,j) = \text{traza}(\boldsymbol{Q}_1^\top \boldsymbol{Q}_2),$$

donde la traza de una matriz es la suma de todos los elementos de su diagonal.

Dada una matriz $\boldsymbol{Q}$ cualquiera, no necesariamente ortogonal, siempre habrá una parte de ella que pertenezca al espacio tangente $\mathscr{T}_{\boldsymbol{A}} V_{p,p}$. Para ello, nótese que podemos escribir [29]

$$\boldsymbol{Q} = \pi_{\boldsymbol{A}}(\boldsymbol{Q}) + \pi_{\boldsymbol{A}}^{\perp}(\boldsymbol{Q})$$

siendo

$$\pi_{\boldsymbol{A}}(\boldsymbol{Q}) = \left(\boldsymbol{Q} - \boldsymbol{A}\boldsymbol{Q}^\top \boldsymbol{A}\right)$$
$$\pi_{\boldsymbol{A}}^{\perp}(\boldsymbol{Q}) = \left(\boldsymbol{Q} + \boldsymbol{A}\boldsymbol{Q}^\top \boldsymbol{A}\right).$$

Pues bien, como se decía, es fácil comprobar que $\pi_{\boldsymbol{A}}(\boldsymbol{Q})$ pertenece al espacio tangente de $\boldsymbol{A}$ [29], es decir,

$$\pi_{\boldsymbol{A}}(\boldsymbol{Q})^\top \boldsymbol{A} + \boldsymbol{A}^\top \pi_{\boldsymbol{A}}(\boldsymbol{Q}) = \boldsymbol{0}.$$

Por otra parte, $\pi_{\boldsymbol{A}}^{\perp}(\boldsymbol{Q})$ es siempre perpendicular a dicho espacio tangente, es decir,

$$\langle \pi_{\boldsymbol{A}}^{\perp}(\boldsymbol{Q}) | \boldsymbol{M} \rangle = 0$$

para cualquier matriz $\boldsymbol{M} \in \mathscr{T}_{\boldsymbol{A}} V_{p,p}$. De hecho, de aquí se deduce que $\pi_{\boldsymbol{A}}(\boldsymbol{Q})$ es precisamente la proyección de $\boldsymbol{Q}$ sobre el espacio tangente $\mathscr{T}_{\boldsymbol{A}} V_{p,p}$.

4.4.1 Método «simple» de optimización en la variedad de Stiefel

Ahora ya es posible entender por qué el algoritmo del gradiente (4.2) no funciona y necesita ser mejorado. Dicho algoritmo (4.2) puede ser reescrito como:

$$\begin{aligned}\boldsymbol{A}_{n+1} &= \boldsymbol{A}_n - \mu\, \partial J(\boldsymbol{A}_n) \\ &= \boldsymbol{A}_n - \mu\, \pi_{\boldsymbol{A}}(\partial J(\boldsymbol{A}_n)) - \mu\, \pi_{\boldsymbol{A}}^{\perp}(\partial J(\boldsymbol{A}_n)),\end{aligned}$$

donde hemos descompuesto el gradiente en sus componentes en el espacio tangente y perpendicular a este último. El problema lo da precisamente la componente perpendicular,

$$\pi_{\boldsymbol{A}}^{\perp}(\partial J(\boldsymbol{A}_n))$$

pues tiende a «empujar» a la matriz $\boldsymbol{A}$ fuera de la variedad de Stiefel. Una posible solución al problema sería eliminar este término, con lo que el

algoritmo quedaría:

$$\begin{aligned}\boldsymbol{A}_{n+1} &= \boldsymbol{A}_n - \mu\, \pi_{\boldsymbol{A}}(\partial J(\boldsymbol{A}_n)) \\ &= \boldsymbol{A}_n - \mu\, \left(\partial J(\boldsymbol{A}_n) - \boldsymbol{A}_n \partial J(\boldsymbol{A}_n)^\top \boldsymbol{A}_n\right) \\ &= \left[\boldsymbol{I} - \mu\, \left(\partial J(\boldsymbol{A}_n)\boldsymbol{A}_n^\top - \boldsymbol{A}_n \partial J(\boldsymbol{A}_n)^\top\right)\right] \boldsymbol{A}_n. \end{aligned} \tag{4.4}$$

Este nuevo algoritmo tiene dos propiedades muy deseables:

1. La función objetivo disminuye con cada iteración: consideremos la serie de Taylor de primer orden de J:

$$J(\boldsymbol{A} + \Delta\boldsymbol{A}) = J(\boldsymbol{A}) + \langle \partial J(\boldsymbol{A}) | \Delta\boldsymbol{A} \rangle + \ldots,$$

donde $\langle \partial J(\boldsymbol{A}) | \Delta\boldsymbol{A} \rangle = \text{traza}(\partial J(\boldsymbol{A})^\top \Delta\boldsymbol{A})$. Si tomamos, como en el algoritmo,

$$\Delta\boldsymbol{A} = -\mu\, \pi_{\boldsymbol{A}}(\partial J(\boldsymbol{A})),$$

se obtiene tras hacer algunos cálculos simples, que

$$\langle \partial J(\boldsymbol{A}) | \Delta\boldsymbol{A} \rangle = -\frac{\mu}{2} \langle \nabla J(\boldsymbol{A}) | \nabla J(\boldsymbol{A}) \rangle,$$

que siempre es negativo y, por lo tanto, J disminuye con cada actualización (siempre y cuando, claro está, μ sea suficientemente pequeño o, de lo contrario, la expansión de Taylor de primer orden dejará de ser válida).

2. Preserva la ortogonalidad. Tras hacer cálculos también se obtiene que

$$\boldsymbol{A}_{n+1}^\top \boldsymbol{A}_{n+1} = \boldsymbol{I} + o(\mu^2), \tag{4.5}$$

donde $o(\mu^2)$ es una función que tiene el orden de magnitud de μ^2. Si μ es mucho más pequeño que uno, entonces el término $o(\mu^2)$ es despreciable.

4.5 Algoritmo propuesto

Todavía es posible una mejora más: como muestra la identidad (4.5), la matriz $\boldsymbol{A}$ es aproximadamente ortogonal tras una iteración. Sin embargo, el error se acumula y cabe esperar que, después de muchas iteraciones, deje de cumplirse la restricción.

Para evitar este inconveniente, y basándonos en resultados clásicos de optimización sobre el conjunto de matrices ortogonales [29], proponemos una nueva regla de actualización que, ahora sí, conserva siempre la ortogonalidad tras cada iteración.

Sea

$$\boldsymbol{A}_{n+1} = \boldsymbol{U}_n \boldsymbol{A}_n,$$

donde

$$\boldsymbol{U}_n = \exp(\boldsymbol{M}_n) := \boldsymbol{I} + \boldsymbol{M}_n + \frac{1}{2!}\boldsymbol{M}_n^2 + \ldots$$

y $\boldsymbol{A}_n$ es una matriz ortogonal,

$$\boldsymbol{A}_n^\top \boldsymbol{A}_n = \boldsymbol{I}$$

Si la matriz $\boldsymbol{M}_n$ es antisimétrica (*skew-symmetric* en inglés), es decir,

$$\boldsymbol{M}_n = -\boldsymbol{M}_n^\top,$$

puede demostrarse que la matriz $\boldsymbol{U}_n$ es siempre ortogonal [29] y, por lo tanto, $\boldsymbol{A}_{n+1}$ también lo será, supuesto que $\boldsymbol{A}_n$ lo es.

Solo queda definir la matriz $\boldsymbol{M}_n$ de manera que con cada iteración decrezca la función objetivo. Por las razones expuestas, el siguiente algoritmo está en condiciones de garantizarlo:

Algoritmo 3: Minimización de la función objetivo

Sea $\boldsymbol{A}_0$ una matriz ortogonal cualquiera;
para $n = 0, 1, 2, \ldots$ **hacer**

1. Tomar
$$\boldsymbol{M}_n = \partial J(\boldsymbol{A}_n)\boldsymbol{A}_n^\top - \boldsymbol{A}_n \partial J(\boldsymbol{A}_n)^\top$$
donde $\partial J(\boldsymbol{A})$ es la matriz en derivadas parciales de J con respecto a los elementos de $\boldsymbol{A}$ definida en (4.3),
$$(\partial J(\boldsymbol{A}))_{ij} = \frac{\partial J(\boldsymbol{A})}{\partial \boldsymbol{A}_{ij}}$$
2. Definir $\boldsymbol{U}_n = \exp(-\mu \boldsymbol{M}_n)$ para $\mu \in \mathbb{R}^+$ suficientemente pequeño.
3. Actualizar $\boldsymbol{A}_{n+1} = \boldsymbol{U}_n \boldsymbol{A}_n$.

fin

Finalmente, hemos de proporcionar la fórmula para $\partial J(\boldsymbol{A})$ en (4.3). Como la subderivada[1] del valor absoluto es la función de signo, se encuentra fácilmente que la enésima columna de $\partial J(\boldsymbol{A})$ es igual

$$\mathbb{E}\{\mathbf{x}\,\mathrm{sgn}\,(D(\boldsymbol{A}_i))\},$$

con D definido en (3.7). Por ejemplo, dada la matriz de datos `M_x`, de dimensiones $p \times N$, que contiene N muestras observadas de X, $\partial J(\boldsymbol{A})$ se pueden evaluar mediante el comando de MATLAB® `M_x*sign(W'*M_x)'/q`. Observe que, para realizar este cálculo, no se requiere ningún conocimiento previo de la clase a la que pertenecen los datos, es decir, no se necesitan patrones de entrenamiento. El procedimiento es completamente *no supervisado*.

[1] La subderivada, o subgradiente, generaliza la noción de derivada a funciones convexas no diferenciables [45].

4.5.1 Interpretación del algoritmo

El algoritmo anterior se puede interpretar fácilmente como una técnica de gradiente descendente: para valores pequeños de μ,

$$\boldsymbol{U}_n = \exp(-\mu \boldsymbol{M}_n) \approx \boldsymbol{I} - \mu \boldsymbol{M}_n \tag{4.6}$$

y por lo tanto

$$\boldsymbol{A}_{n+1} = \boldsymbol{U}_n \boldsymbol{A}_n \approx \boldsymbol{A}_n - \mu \boldsymbol{M}_n \boldsymbol{A}_n. \tag{4.7}$$

Curiosamente, el término

$$\begin{aligned} \boldsymbol{M}_n \boldsymbol{A}_n &= \partial J(\boldsymbol{A}_n) \boldsymbol{A}_n^\top \boldsymbol{A}_n - \boldsymbol{A}_n \partial J(\boldsymbol{A}_n)^\top \boldsymbol{A}_n \\ &= \partial J(\boldsymbol{A}_n) - \boldsymbol{A}_n \, \partial J(\boldsymbol{A}_n)^\top \boldsymbol{A}_n \end{aligned}$$

es la proyección del gradiente de J sobre el espacio tangente del conjunto de matrices ortogonales [29], lo que nos permite ver (4.7) como una regla de gradiente aproximada, es decir,

$$\Delta \boldsymbol{A} = \boldsymbol{A}_{n+1} - \boldsymbol{A}_n \approx -\mu \nabla \left(J(\boldsymbol{A}_n) - \boldsymbol{A}_n \, \partial J(\boldsymbol{A}_n)^\top \boldsymbol{A}_n \right),$$

coincidiendo con (4.4). De este modo, el algoritmo decrece la función objetivo con cada iteración (si μ es suficientemente pequeño), a la vez que garantiza que siempre las matrices $\boldsymbol{A}_n$ siempre son ortogonales.

4.5.2 Elección de los parámetros del algoritmo

Finalmente, nos queda por determinar el valor más adecuado para la constante μ del algoritmo. En principio, μ no debe de ser grande; en caso contrario, las aproximaciones de Taylor hechas no serían válidas y la función objetivo no decrecería con cada iteración. No obstante, si μ es demasiado pequeña, la convergencia del método será muy lenta.

Para encontrar el valor óptimo de μ se utilizará el siguiente procedimiento:

Algoritmo 4: Selección del paso μ del algoritmo propuesto

para $n = 0, 1, 2, \ldots$ **hacer**

1. Definir
$$\boldsymbol{M}_n = \partial J(\boldsymbol{A}_n)\boldsymbol{A}_n^\top - \boldsymbol{A}_n \partial J(\boldsymbol{A}_n)^\top$$
donde $\partial J(\boldsymbol{A})$ es la matriz en derivadas parciales de J con respecto a los elementos de $\boldsymbol{A}$ definida en (4.3),
$$(\partial J(\boldsymbol{A}))_{ij} = \frac{\partial J(\boldsymbol{A})}{\partial \boldsymbol{A}_{ij}}$$
donde $\partial J(\boldsymbol{A})$ es la matriz en derivadas parciales de J con respecto a los elementos de $\boldsymbol{A}$ definida en (4.3),
$$(\partial J(\boldsymbol{A}))_{ij} = \frac{\partial J(\boldsymbol{A})}{\partial \boldsymbol{A}_{ij}}$$
2. Llevar a cabo la búsqueda del parámetro μ que minimiza la función
$$J(\exp(-\mu \boldsymbol{M}_n)\boldsymbol{A}_n),$$
con $\mu \in [0, \mu_{\text{máx}}]$, siendo $\mu_{\text{máx}}$ un valor suficientemente grande. Como resultado, se determinará el valor $\mu = \mu_{\text{mín}}$.
3. Definir $\boldsymbol{U}_n = \exp(-\mu_{\text{mín}} \boldsymbol{M}_n)$.
4. Actualizar $\boldsymbol{A}_{n+1} = \boldsymbol{U}_n \boldsymbol{A}_n$.

fin

Para implementar el paso 2 de este método puede usarse cualquiera de las propuestas que aparecen descritas en el Capítulo 8 (pág. 412) de la referencia [81]. Por ejemplo, para la realización de los experimentos de esta Tesis, se ha utilizado la técnica de la aproximación polinómica. Es decir, fijados $\boldsymbol{M}_n$ y $\boldsymbol{A}_n$,
$$f(\mu) = J(\exp(-\mu \boldsymbol{M}_n)\boldsymbol{A}_n),$$
es una función únicamente de μ. Dados tres valores de esta función, $f(\mu_0)$, $f(\mu_1)$ y $f(\mu_2)$, con $f(\mu_0) > f(\mu_1)$ y $f(\mu_1) < f(\mu_2)$, el procedimiento consiste en aproximar $f(\mu)$ mediante una parábola y calcular el valor

$\mu = \mu_{\text{mín}}$ que la minimiza, lo cual se puede llevar a cabo fácilmente [81].

4.5.3 Cota de la aproximación (4.6)

Para discutir la aproximación (4.6), donde $\eta \in \mathbb{R}^+$ es una constante pequeña y positiva, tratemos de acotar el error que se produce. La matriz exponencial se define como [35]:

$$\boldsymbol{U} = \exp(-\eta \boldsymbol{M}) := \boldsymbol{I} + \sum_{k=1}^{\infty} \frac{(-\eta \boldsymbol{M})^k}{k!}$$

donde $\boldsymbol{I}$ es la matriz identidad y hemos omitido el subíndice n para simplificar la notación. Supongamos aproximar la exponencial por $\hat{\boldsymbol{U}} = \boldsymbol{I} - \eta \boldsymbol{M}$. El error en la aproximación se define como

$$R = \boldsymbol{U} - \hat{\boldsymbol{U}} = \sum_{k=2}^{\infty} \frac{(-\eta \boldsymbol{M})^k}{k!}.$$

Utilizando la desigualdad del triángulo, aplicada a las normas de matrices:

$$\|R\| \leq \sum_{k=2}^{\infty} \eta^k \frac{\|\boldsymbol{M}\|^k}{k!} = \eta^2 \|\boldsymbol{M}\|^2 \sum_{k=0}^{\infty} \frac{\varepsilon^k}{(k+2)!} < \eta^2 \|\boldsymbol{M}\|^2 \sum_{k=0}^{\infty} \frac{\varepsilon^k}{k!} = \eta^2 \|\boldsymbol{M}\|^2 \exp(\varepsilon)$$

donde $\varepsilon = \eta \|\boldsymbol{M}\|$. En consecuencia, el error en la aproximación está acotado por

$$\|R\| \leq \eta^2 \|\boldsymbol{M}\|^2 \exp(\eta \|\boldsymbol{M}\|),$$

que, a su vez, está dominado por un término cuya magnitud está en el orden de $\eta^2 \|\boldsymbol{M}\|^2$, siempre que $\eta < \frac{1}{\|\boldsymbol{M}\|}$.

5 Resultados experimentales

5.1 Introducción

A continuacion se realizá una bateria de experimentos para evaluar experimentalmente las prestaciones del algoritmo propuesto. Se trabajará tanto con datos generados artificialmente como con señales reales.

5.2 Simulaciones utilizando datos artificiales

Consideremos primero la mezcla en un espacio bidimensional (esto es, $p = 2$) de datos procedentes de dos clases gaussianas equiprobables, con media cero, y covarianzas respectivas

$$\boldsymbol{C}_1 = \begin{bmatrix} 1 & 0.68 \\ 0.68 & 1 \end{bmatrix} \tag{5.1}$$

$$\boldsymbol{C}_2 = \begin{bmatrix} 1 & -0.68 \\ -0.68 & 1 \end{bmatrix}. \tag{5.2}$$

Obsérvese que $\boldsymbol{C}_1$ y $\boldsymbol{C}_2$ cumplen la condición de blanqueado (3.2) y, por ello, comparten los mismos autovectores,

$$\boldsymbol{v}_1 = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]^\top \approx [\ \ 0.71, 0.71]^\top, \tag{5.3}$$

$$\boldsymbol{v}_2 = \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]^\top \approx [-0.71, 0.71]^\top. \tag{5.4}$$

Tomamos 50 muestras de cada clase (100 en total), cuyo diagrama de dispersion se muestra en la Figura 5.2.1. En esta misma figura, se han trazado con línea discontinua y color «azul» las rectas en la dirección de $\boldsymbol{v}_1$ y $\boldsymbol{v}_2$. Podemos observar que cada una de las clases, $\mathscr{C}_1$ y $\mathscr{C}_2$, se extiende con cierta claridad en la dirección de uno de estos autovectores; pero no en la del otro. En otras palabras, la clase «círculo» se distribuye fundamentalmente entre el primer y tercer cuadrantes (esto es, en la dirección de $\boldsymbol{v}_1$), mientras que la clase «cruz» lo hace en la dirección perpendicular. En el lenguaje de PCA, $\boldsymbol{v}_1$ señala la «dirección principal» de la clase «círculo», mientras que $\boldsymbol{v}_2$ es la de la clase «cruz».

Los vectores que minimizan la norma L1, calculados mediante el algoritmo propuesto en el Capítulo anterior son los siguientes:

$$\boldsymbol{a}_1 \approx [0.73, 0.69]^\top \tag{5.5}$$

$$\boldsymbol{a}_2 \approx [-0.69, 0.73]^\top. \tag{5.6}$$

Sus direcciones están marcadas en «rojo» en la Figura.

Admitiendo que $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$ estiman con suficiente precisión las direcciones principales $\boldsymbol{v}_1$ y $\boldsymbol{v}_2$ de las nubes de puntos, es posible clasificar las observaciones $\boldsymbol{x}$ en dos grupos, A y B, en función de su cercanía a las rectas definidas por dichos vectores $\boldsymbol{a}_i$. La regla sería la siguiente: si

$$\left\|\boldsymbol{x} - \boldsymbol{a}_1(\boldsymbol{a}_1^\top \boldsymbol{x})\right\| < \left\|\boldsymbol{x} - \boldsymbol{a}_2(\boldsymbol{a}_2^\top \boldsymbol{x})\right\|, \tag{5.7}$$

asignamos $\boldsymbol{x}$ a A. En caso contrario, se asignaría $\boldsymbol{x}$ al grupo B.

Al aplicar este criterio, obtenemos la matriz de confusión que se muestra

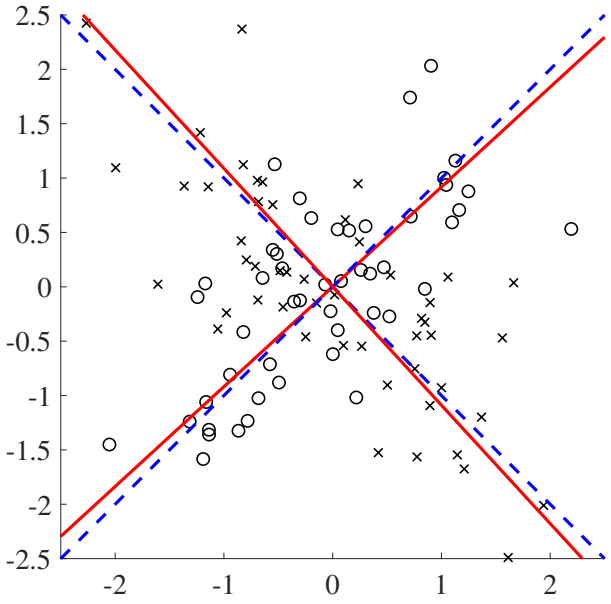


Figura 5.2.1 Diagrama de dispersión de observaciones pertenecientes a dos clases («cruces» y «círculos»). En línea azul discontinua se muestran las direcciones de los autovectores de las matrices de covarianza. Las líneas rojas indican las direcciones de proyección que minimizan el criterio basado en la norma L1, calculadas por el algoritmo *no supervisado* propuesto. Las líneas rojas están rotadas un ángulo de solo 1.61° con respecto a las azules.

en la Tabla 5.2.1. Observamos que se ha formado un grupo que se compone de 35 muestras de la clase $\mathscr{C}_1$ y solo 13 de la clase $\mathscr{C}_2$. El segundo grupo cuenta con 15 instancias de clase $\mathscr{C}_1$ y 37 de clase $\mathscr{C}_2$. Es decir, el algoritmo *no supervisado* separa los datos de manera que la mayoría de los puntos de una misma clase acaban agrupados juntos. Para calcular la precisión obtenida, sumaremos los valores en la diagonal de la matriz de confusión y dividiremos por el número total de muestras: así, dado que de 100 observaciones hemos agrupado correctamente

$$35 + 37 = 72,$$

Tabla 5.2.1 Matriz de confusión obtenida después de aplicar la regla de clasificación (5.7).

		Clase real		
		$\mathscr{C}_1$	$\mathscr{C}_2$	Total
Grupo asignado	A	35	13	48
	B	15	37	52
	Total	50	50	100

podemos afirmar que el método proporciona una precisión del 72 %.

La regla de agrupación (5.7) puede también ser interpretada de la siguiente forma: dada una observación cualquiera $\boldsymbol{x}$, dicha regla es equivalente a

$$|\boldsymbol{w}_1^\top \boldsymbol{x}| > |\boldsymbol{w}_2^\top \boldsymbol{x}|. \tag{5.8}$$

La figura 5.2.2a muestra que la desigualdad (5.8) se cumple para la mayoría (el 70 %) de los elementos de $\mathscr{C}_1$ (la línea naranja se sitúa por encima de la verde). Para los elementos de $\mathscr{C}_2$ ocurre lo contrario, como se ve en la Figura 5.2.2b. Como la magnitud de los puntos proyectados tiene un comportamiento opuesto en una clase y en otra, es decir, si $|\boldsymbol{w}_1^\top \mathbf{x}|$ crece, $|\boldsymbol{w}_2^\top \mathbf{x}|$ decrece y viceversa, el valor de la correlación de Pearson ρ_{12} entre las variables aleatorias $|\boldsymbol{w}_1^\top \mathbf{x}|$ y $|\boldsymbol{w}_2^\top \mathbf{x}|$ ha de ser negativo[1]. De hecho, al calcular esta correlación obtenemos:

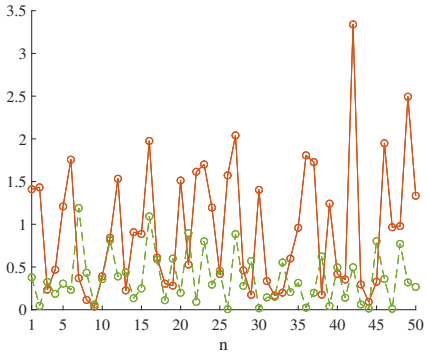
$$\rho_{12} = -0.254,$$

confirmando nuestra intuición. En el siguiente experimento se utilizará esta característica para facilitar la clasificación *no supervisada*.

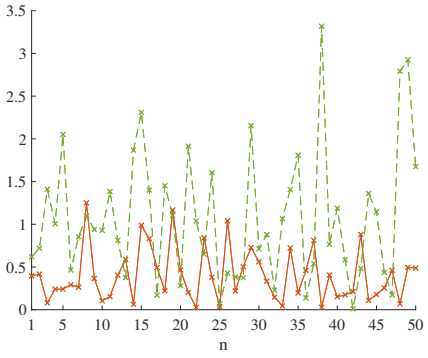
Datos Gaussianos y no Gaussianos

Cuando se aplica a variables p-dimensionales, el algoritmo presentado en el Capítulo anterior determinará p vectores de proyección $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p \in \mathbb{R}^p$. Nos enfrentamos al problema de determinar *de forma no supervisada* cuáles

[1] Dadas las variables $\mathrm{z}_i = |\boldsymbol{w}_i^\top \mathbf{x}|$, $\rho_{ij} = \frac{\mathrm{cov}(\mathrm{z}_i, \mathrm{z}_j)}{\sigma_i \sigma_j}$, donde *cov* representa la covarianza de las variables y σ_i es la desviación estandar de z_i.



(a) Magnitud de las proyecciones de los puntos de la clase 1 sobre $\boldsymbol{w}_1$ (línea naranja) y $\boldsymbol{w}_2$ (línea verde).



(b) Magnitud de las proyecciones de los puntos de la clase 2 sobre $\boldsymbol{w}_1$ (línea naranja) y $\boldsymbol{w}_2$ (línea verde).

Figura 5.2.2 Magnitudes de las proyecciones de las clases en el experimento de la Sección 5.2.

de ellos apuntan en las «direcciones principales» de la clase $\mathscr{C}_1$ y cuáles corresponden a la clase $\mathscr{C}_2$.

Reordenaremos los índices de estos vectores de modo que:

1. $\boldsymbol{a}_1$ sea el mínimo global del criterio de norma L1. Como sugiere el análisis teórico (ver teorema 3.4.2), $\boldsymbol{a}_1$ debe ser por tanto el autovector dominante de una de las clases (y el menos dominante de la otra)

2. $\boldsymbol{a}_p$ sea tal que el coeficiente de correlación de Pearson[2] entre $|\boldsymbol{a}_1^\top \mathbf{x}|$ y $|\boldsymbol{a}_p^\top \mathbf{x}|$ es mínimo:

$$\rho_{1p} < \rho_{1i} \text{ para todo } i \neq p.$$

Cabe por tanto esperar que, como sugiere el experimento previo, $\boldsymbol{a}_1$ y $\boldsymbol{a}_p$ apunten en las «direcciones principales» de clases *distintas*.

[2] Ver nota al pie 1.

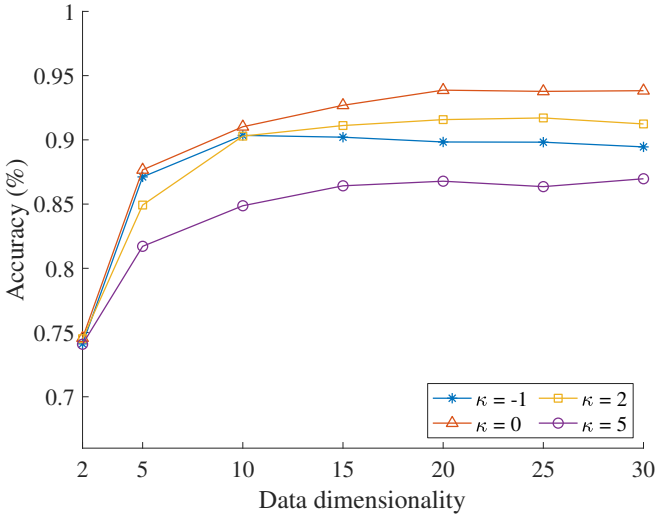


Figura 5.2.3 Precisión del algoritmo desarrollado, en función de la dimensión p del espacio de los datos, para distribuciones marginales con distinto exceso de curtosis κ (por ejemplo, $\kappa = -1.2$ corresponde a una distribución uniforme, $\kappa = 0$ da la distribución Gaussiana o $\kappa = 3$ es propia de la distribución de Laplace). Cada curva ha sido obtenida promediando los resultados de 100 experimentos independientes.

A continuación, inspirándonos en (5.7), adoptamos la siguiente regla:

$$\text{Asignar } \boldsymbol{x} \text{ a } A\,(B) \text{ si: } \left\|\boldsymbol{x} - \boldsymbol{w}_1(\boldsymbol{w}_1^\top \boldsymbol{x})\right\| < (>) \left\|\boldsymbol{x} - \boldsymbol{w}_p(\boldsymbol{w}_p^\top \boldsymbol{x})\right\|.$$

La figura 5.2.3 muestra la precisión obtenida utilizando esta regla de clasificación *no supervisada*, definida como en el experimento previo, ensayando con diferentes distribuciones de datos y valores de p.

En cada simulación se generan las matrices de covarianza al azar y los datos son «blanqueados» en una primera etapa de pre-procesamiento. La función de coste se estima a partir de $N = 50\,p$ muestras de cada clase. Para evaluar la robustez del algoritmo, se ha experimentado también con datos no Gaussianos. Estos se han generado utilizando los algoritmos propuestos en [30, 106], Estos algoritmos, muy conocidos en el ámbito del análisis multivariante, permiten generar datos de distribuciones arbitrarias con

matrices de covarianza predefinidas. Para ser más precisos, transforman variables Gaussianas de forma no lineal mediante un procedimiento que nos permite escoger la media, varianza, coeficiente de asimetría y curtosis de las distribuciones marginales resultantes.

En este experimento, generamos datos de media cero, varianza unitaria y coeficiente de asimetría nulo. No obstante, para explorar diferentes escenarios, el valor del exceso de curtosis[3] κ de los datos marginales irá tomando valores entre -1 (que corresponde a una densidad sub-Gaussiana) y 5 (distribución altamente super-Gaussiana), pasando por cero (variable Gaussiana). De esta manera, podremos testear el funcionamiento del algoritmo cuando la hipótesis de Gaussianeidad de los datos no se cumple. El resultado más notable del experimento es que, como se muestra en la Figura 5.2.3, la precisión del método aumenta con la dimensionalidad p de las variables de entrada.

5.3 Experimento con señales electroencefalográficas (EEG)

En las interfaces cerebro-máquina (BCI, «brain computer interfaces») más usuales, el usuario imagina una de sus extremidades moviéndose y el sistema intenta detectar cuál es [72, 78].

El conjunto de datos 2a de la competición BCI-IV recopila una serie de repeticiones de movimientos simples (mano izquierda, mano derecha, pies o lengua) [24, 67, 102]. En cada sesión, se registraron $p = 22$ canales de EEG a distintos voluntarios, utilizando una frecuencia de muestreo de 250 Hz. Como es habitual en el procesamiento de señales BCI, los datos de EEG ha sido filtrados paso banda entre $8 - 30$ Hz. Este pre-procesamiento asegura que los datos sean de media cero y, apoyándonos en el teorema central del límite, podremos aceptar la hipótesis de Gaussianidad de los datos. Cada movimiento imaginado dura en torno a tres segundos; pero solo utilizaremos los dos últimos en nuestro experimento para evitar los

[3] El exceso de curtosis es el momento central de cuarto orden de los datos estandarizados *menos tres*.

transitorios iniciales. Como ilustración, en la Figura 5.3.1 se muestra uno de estos intervalos.

Para realizar el experimento, concatenamos todos los ensayos de una misma persona y movimiento imaginario en una única matriz de datos $22 \times 30\,000$. Después alimentamos el algoritmo *no supervisado* con pares de matrices correspondientes a movimientos *distintos*. Se obtienen mejores resultados cuando, además, los datos han sido filtrados en la banda entre 12 y 30 Hz (bandas β y γ del EEG) y pre-procesados con el método propuesto en [88] para reducir la no estacionariedad inherente al EEG.

Como ejemplo, la Figura 5.3.2 muestra las densidades de probabilidad de algunas de las proyecciones obtenidas: las diferencias de varianza entre dichas clases son aparentes incluso a simple vista. En cada ensayo, para agrupar las muestras, hemos utilizado la proyección que minimiza la norma L1, las dos proyecciones más correlacionadas con ella y las tres proyecciones que tienen menor correlación con la primera. La Tabla 5.3.1 muestra los resultados obtenidos al clasificar un movimiento imaginado mediante la comparación de la varianza total de cada uno de estos dos grupos de tres proyecciones (la varianza total en el subespacio de la proyecciones se mide con la traza de la matriz de covarianza).

La precisión de la clasificación se calcula para los nueve voluntarios disponibles en la base de datos, considerando todas las combinaciones posibles de tareas imaginadas (ID: mano izquierda-mano derecha, IP: mano izquierda-pies, etc.). Por ejemplo, para el usuario 1 se ha obtenido un alto grado de precisión (93 %) al discriminar entre «mano derecha» y «pies», *de una manera completamente no supervisada*, pero esta precisión se reduce al 52 % para el mismo usuario y el par «pies-lengua». Hay una gran variabilidad entre usuarios y pares de movimientos; no obstante, al promediar para todos los voluntarios, podemos discriminar entre «mano izquierda» y «pies», «mano izquierda» y «lengua» y movimientos de «mano derecha» y «lengua» en más del 70 % de los casos.

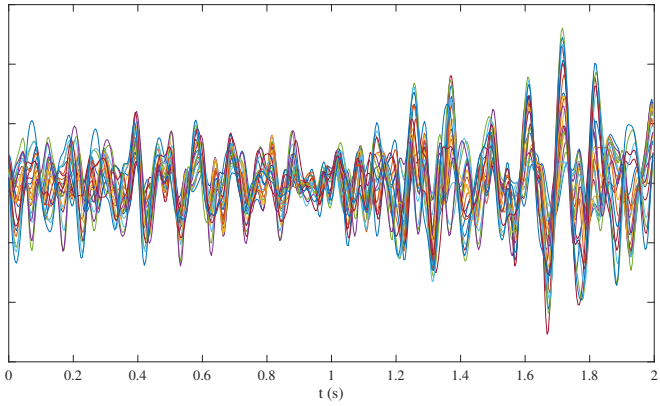


Figura 5.3.1 EEG de $p = 22$ canales registrado mientras el usuario 1 imagina que está moviendo su lengua.

Tabla 5.3.1 Precisión obtenida al discriminar parejas de movimientos imaginados (I = mano izquierda, D = mano derecha, P = pies, L = lengua). Los resultados se muestran para los nueve voluntarios que recoge la base de datos (u1,. . ., u9). La última columna proporciona la precisión por usuario, promediada para todos los tipos de movimiento. La última fila es el promedio de las anteriores.

Usu.	I-D	I-P	I-L	D-P	D-L	P-L	media
u1	0.66	0.89	0.91	0.93	0.92	0.52	0.84
u2	0.52	0.72	0.6	0.68	0.54	0.65	0.64
u3	0.87	0.68	0.69	0.86	0.86	0.54	0.73
u4	0.57	0.7	0.61	0.63	0.65	0.55	0.63
u5	0.53	0.55	0.6	0.55	0.57	0.54	0.56
u6	0.52	0.64	0.56	0.53	0.54	0.53	0.56
u7	0.59	0.73	0.74	0.89	0.89	0.69	0.79
u8	0.77	0.65	0.87	0.59	0.75	0.71	0.72
u9	0.78	0.86	0.88	0.55	0.7	0.76	0.75
media	0.65	0.71	0.72	0.69	0.71	0.61	

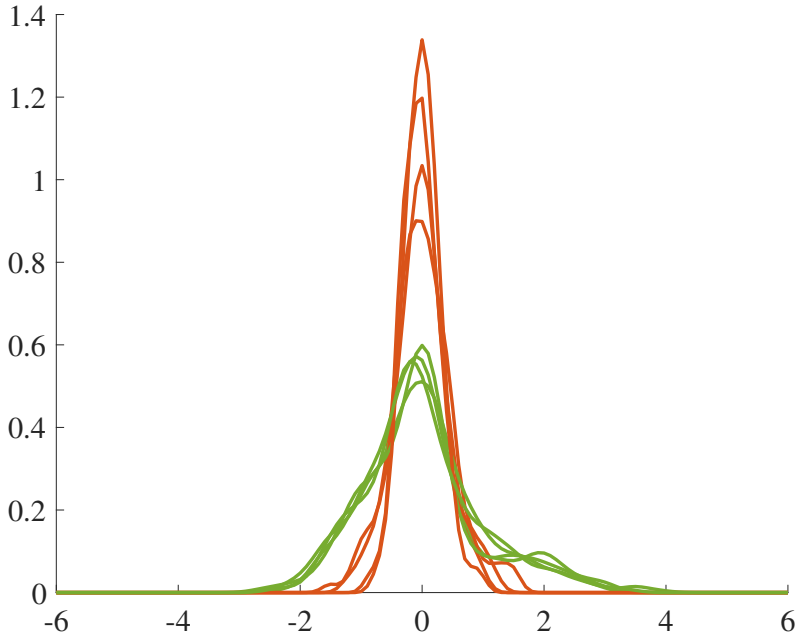


Figura 5.3.2 Funciones de densidad de probabilidad (obtenidas con un método de estimación no paramétrico, utilizando un «kernel» Gaussiano con ancho de banda proporcionado por la regla de Silverman) de varias proyecciones correspondientes algunas señales EEG «mano izquierda» (naranja) y «pie» (verde) del usuario 1. La dirección de proyección en todos los casos es la que minimiza la función objetivo basada en la norma L1, cuando el algoritmo recibe como entrada todas las señales EEG «mano izquierda» y «pie» (verde) del usuario 1. La diferencia entre las varianzas de las distribuciones es claramente visible.

5.4 Procesamiento de imágenes radiográficas digitales

La enfermedad del Coronavirus (COVID-19) ha sido declarada pandemia por la Organización Mundial de la Salud (OMS) y, hasta el 27 de mayo de 2021, alrededor de 3.9 millones de personas han fallecido por su causa en el mundo. En respuesta, se ha llevado a cabo una amplia labor de investigación y desarrollo para encontrar una forma eficaz de diagnóstico o vacunación. La investigación no se limita al ámbito médico y abarca también los campos de

la biotecnología, la ciencia de datos y la inteligencia artificial. El objetivo es la búsqueda de herramientas útiles para el análisis y el diagnóstico asistidos por ordenador (CAD) [19, 41, 23, 107].

Dado que el COVID-19 invade principalmente los pulmones, es importante que los profesionales sanitarios determinen si estos han sido ya infectados para priorizar los niveles de gravedad en los pacientes y la necesidad de hospitalización. El COVID-19 entra a través de las vías respiratorias y provoca una neumonía grave: el pulmón se inflama, llena de fluidos y desarrolla unas manchas denominadas «opacidades de vidrio esmerilado» [19, 23, 107].

Se ha demostrado a partir de numerosas investigaciones que las redes neuronales son eficaces a la hora de hacer diagnósticos a partir de imágenes. Sin embargo, la mayoría de las técnicas de «deep learning» son *supervisadas* y requieren una gran cantidad de datos de entrenamiento. Dado que es una tarea difícil obtener una base de datos de gran tamaño en el campo sanitario, cabe esperar que el rendimiento de los algoritmos se resienta [19, 41]. Esto ha motivado la aplicación de nuestro algoritmo para la detección automática de COVID-19 en imágenes de radiografía de tórax, con la finalidad de conseguir un método de alta precisión que pueda detectarlo en etapas tempranas, definir el tipo exacto de las muestras y mejorar los resultados obtenidos hasta la actualidad, ayudando al reconocimiento de esta enfermedad en la práctica médica.

Para la realización de los experimentos se ha utilizado la base de datos recopilada por las Universidades de Catar, en Doha (Catar), y Daca (Bangladés) [17, 92]. Esta base de datos contiene más de 21 000 imágenes, de las cuales 3 616 corresponden a radiografías de pacientes COVID-19, 6 012 a opacidades pulmonares no-COVID, 1 345 a neumonía no-COVID y 10 200 son radiografías de tórax de pacientes sanos. Algunos ejemplos se muestran en la Figura 5.4.1. La base de datos sirve, además, como patrón para evaluar las prestaciones de los algoritmos que compiten en un reciente reto propuesto por Kaggle[4]. La base de datos distingue entre imágenes de entrenamiento y de *test*. Sin embargo, nótese que el algoritmo propuesto en esta Tesis es no supervisado, por lo que no necesita ninguna secuencia de

[4] *https://www.kaggle.com/tawsifurrahman/covid19-radiography-database*

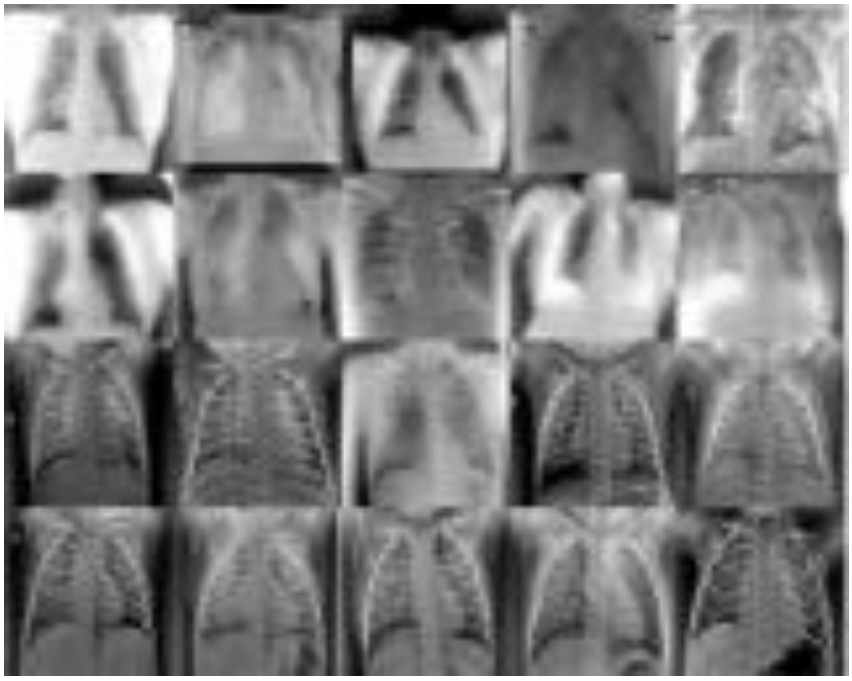


Figura 5.4.1 Imágenes originales de la base de datos. Las 10 primeras radiografías corresponden a pacientes con Covid-19 y las 10 imágenes inferiores pertenecen a pacientes sanos.

entrenamiento. Esto lo hace mucho más atractivo para aplicaciones, como ésta, en las que la cantidad de datos disponible es relativamente escasa.

5.4.1 Pre-procesamiento de las imágenes

En la radiología digital, la imagen se capta en una película fotográfica, que se escanea y procesa. Una vez es enviada al ordenador, se puede trabajar sobre la imagen para realizar ajustes. El pre-procesamiento es realizado automáticamente por el ordenador, mientras que el post-procesamiento lo lleva a cabo el técnico. En nuestro caso, se han ecualizado los histogramas de las imágenes para normalizarlas y evitar que el algoritmo haga una clasificación no supervisada en base al nivel de intensidad de las mismas (esto es, divida las imágenes en «claras» y «oscuras», lo que no sería el resultado pretendido).

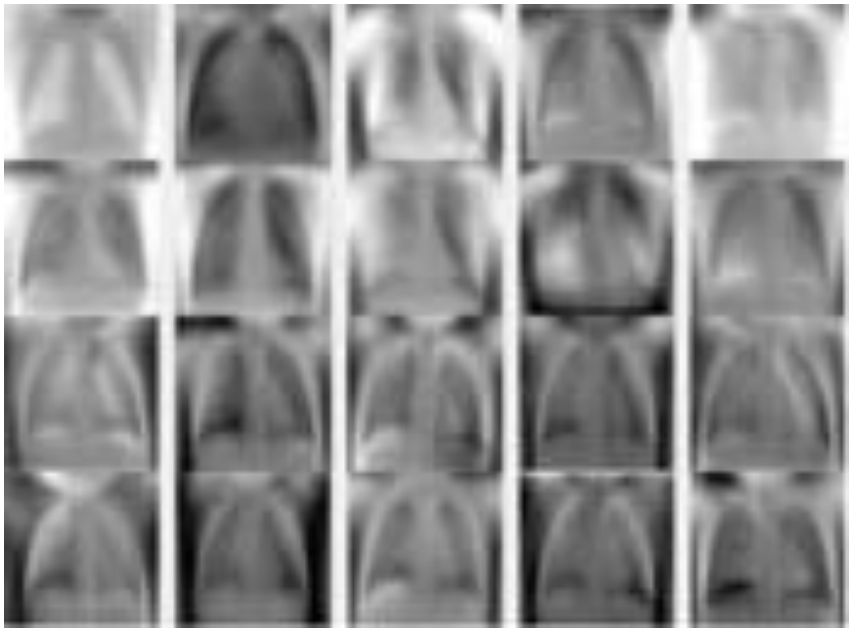


Figura 5.4.2 Radiografías tras la reducción de dimensionalidad.

En segundo lugar, se han seleccionado al azar 840 imágenes de la base de datos (420 COVID y 420 «sanas») para realizar el experimento. Las imágenes se «vectorizan», o convierten en vectores, apilando los píxeles por columnas. Dado que los vectores resultantes tienen un gran tamaño, se realiza seguidamente una reducción de la dimensionalidad, utilizando análisis de componentes principales clásico, para eliminar la redundancia entre dichos píxeles. En la Figura 5.4.2 se muestran algunas imágenes reconstruidas a partir de sus componentes principales más significativas. Nótese que el algoritmo que minimiza la norma L1 trabajará a partir de la información que contienen estas últimas imágenes, no de la que poseen las imágenes originales.

5.4.2 Minimización de la norma L1

Por otra parte, en la Figura 5.4.3 se muestran las proyecciones de las 840 imágenes utilizadas sobre seis de las direcciones obtenidas al minimizar dicha norma L1 (cada «subplot» equivale a una dirección). Las primeras 420

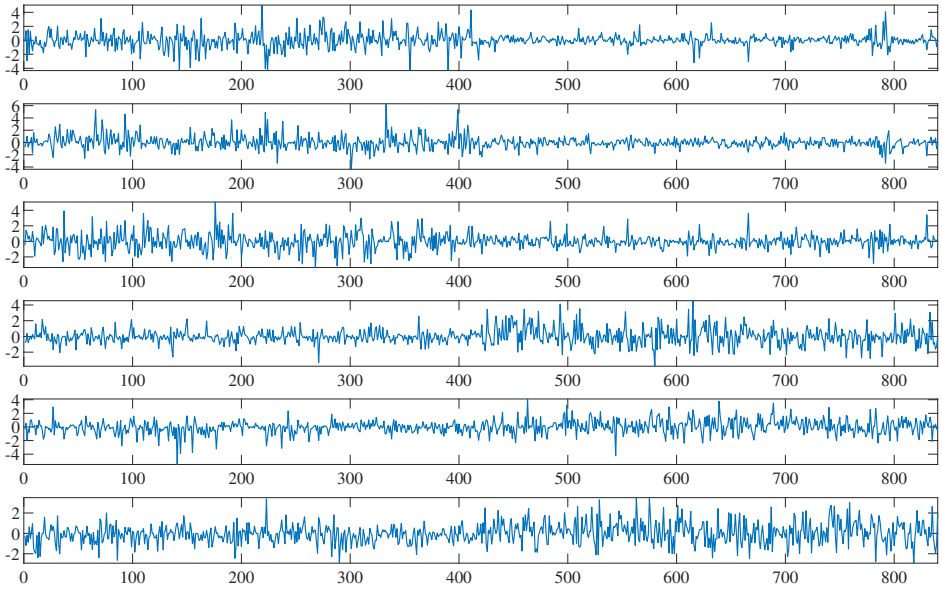


Figura 5.4.3 Proyecciones de radiografías de tórax en seis direcciones que minimizan la norma L1 del conjunto. Las primeras 400 muestras pertenecen a imágenes de pacientes COVID; las últimas se asocian a personas sanas.

muestras de cada gráfica corresponden a la magnitud de las proyecciones de radiografías «COVID». Las últimas 420 a personas sanas. Se observa que, como cabía esperar, la varianza de las proyecciones en cada dirección es claramente distinta para cada grupo.

Resulta interesante convertir las direcciones de proyección $\boldsymbol{w}_i$ calculadas por el algoritmo en «imágenes». Para ello, hemos de llevar a cabo el proceso inverso a la «vectorización» y deshacer la transformada PCA que reduce la dimensionalidad. De esta forma se obtienen imágenes como las de la Fig. 5.4.4. Éstas se pueden interpretar como el «prototipo» de lo que el algoritmo considera una radiografía COVID/no-COVID. Al proyectar las imágenes originales sobre estos prototipos se obtienen los coeficientes de proyección mostrados en la Fig. 5.4.6. Se observa claramente de nuevo una distinción entre la magnitud de las proyecciones de una clase y otra.

Finalmente, las proyecciones de cada imagen en todas las direcciones que minimizan la norma L1 se van a utilizar como entrada a un algoritmo

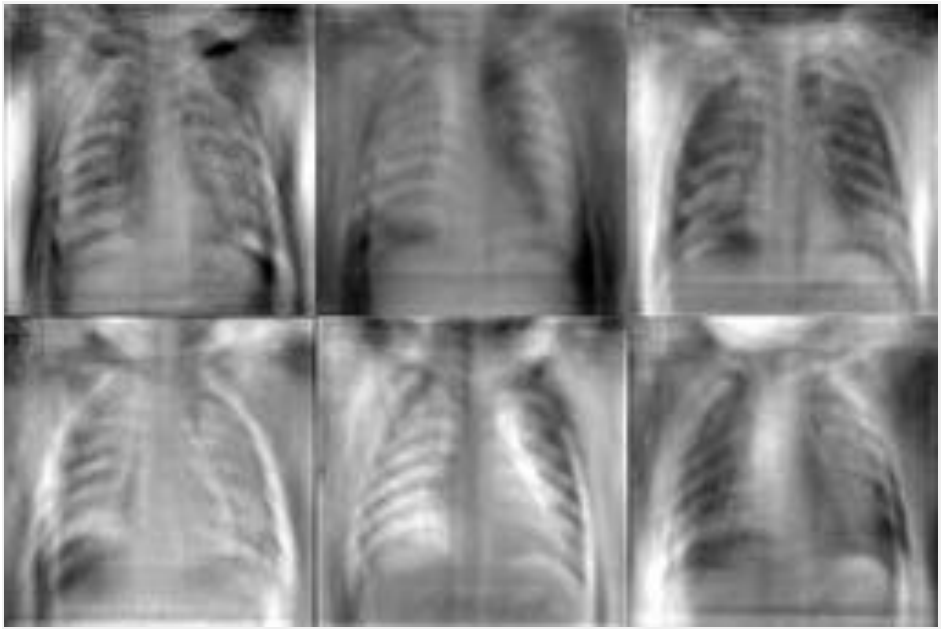


Figura 5.4.4 Imágenes prototipo determinadas por el algoritmo.

de «clustering» basado en mezclas de Gaussianas. Se ajusta el algoritmo para que agrupe las imágenes en dos conjuntos, que se comparan con los grupos originales de radiografías COVID y no-COVID. De esta manera se consigue la matriz de confusión mostrada en la Figura 5.4.5. Se observa que, en promedio, el 91.3 % de las radiografías se han agrupado con las de su misma clase. Por comparación, cuando se clasifican las radiografías directamente con un algoritmo supervisado (basado en el análisis lineal discriminante de Fisher [12]), la precisión obtenida es solo ligeramente mayor (92.6 %).

5.4.3 Proyección de radiografías de tórax no utilizadas anteriormente

Para comprobar la capacidad de generalización de la técnica propuesta, proyectaremos nuevas imágenes de la base de datos sobre las direcciones que minimizan la norma L1 calculadas con anterioridad. Es decir, vamos a proyectar radiografías de tórax *que no han intervenido en la obtención de dichas direcciones*. La magnitud de algunas de las proyecciones obtenidas

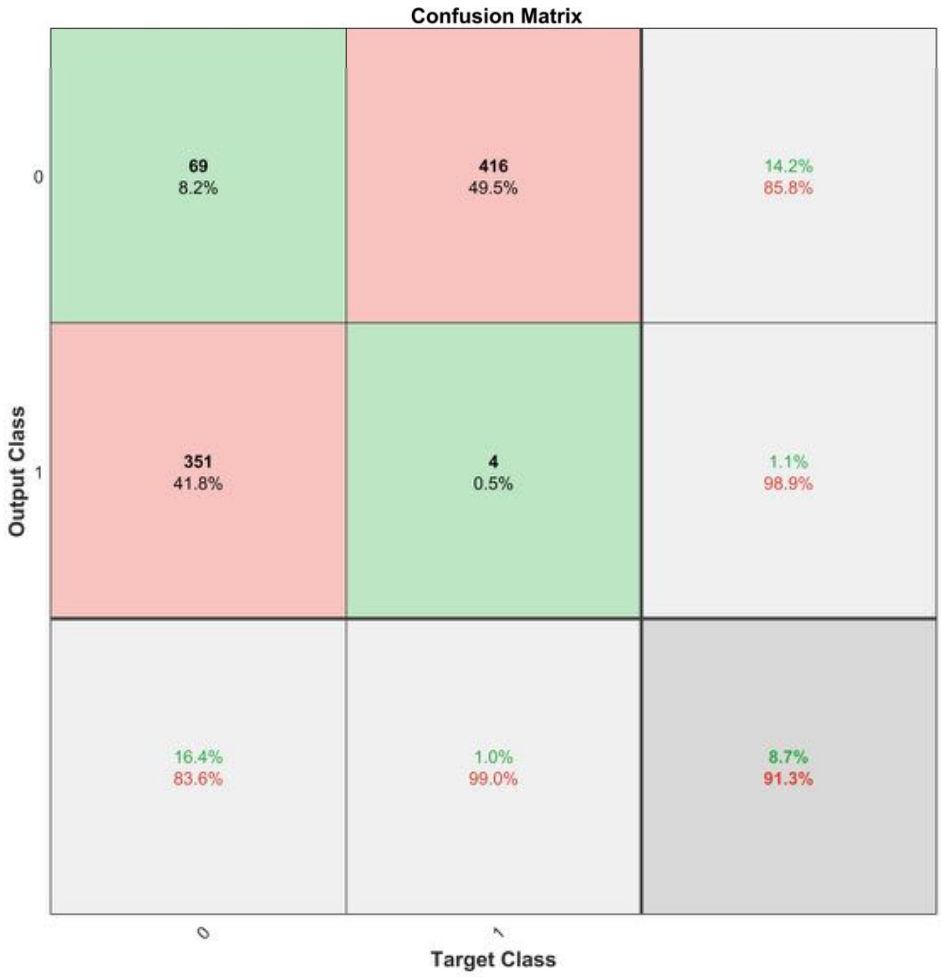


Figura 5.4.5 Matriz de confusión obtenida al hacer una clasificación *no supervisada* de las radiografías de tórax a partir de las proyecciones en las direcciones que minimizan la norma L1.

puede apreciarse en la Fig. 5.4.7. Se aprecia nuevamente clara una distinción entre las proyecciones de radiografías COVID y no-COVID. Al hacer «clustering» con el algoritmo de la mezcla de Gaussianas se obtiene que el 95.2 % de las imágenes acaba agrupadas con las de su misma clase. Curiosamente, este resultado es incluso mejor que el obtenido en el experimento anterior.

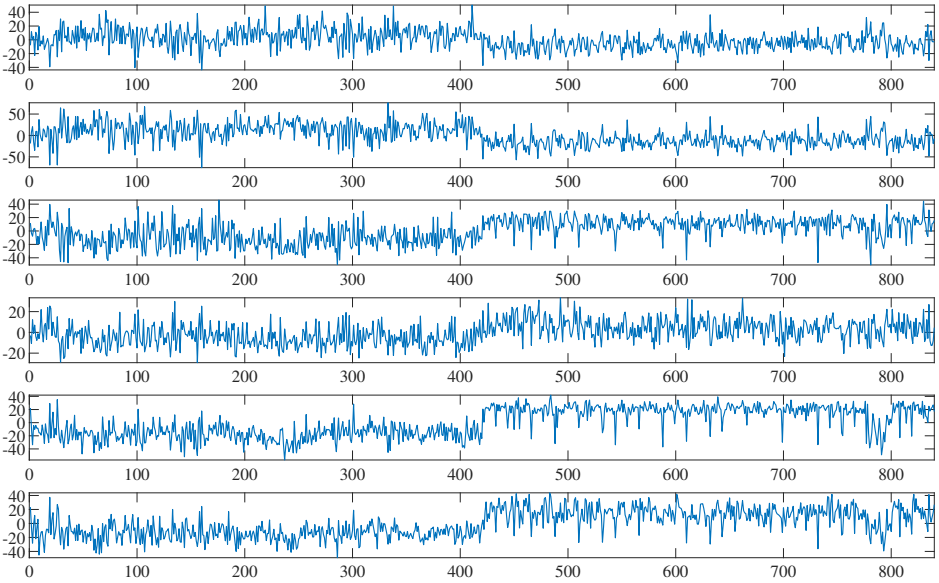


Figura 5.4.6 Proyecciones de radiografías de tórax sobre las imágenes prototipo. Las primeras 420 muestras pertenecen a imágenes de pacientes COVID; las últimas se asocian a personas sanas.

5.5 Procesamiento de caras

El reconocimiento de rostros es una tarea donde las neuronas de la corteza visual responden a características locales específicas, tales como bordes, líneas, ángulos o movimiento, las cuales se combinan en patrones útiles. Dos de los métodos más usados para esta tarea son los conocidos como de «eigenfaces» y el de «fisherfaces» [7, 62, 85]:

- *Eigenfaces:* clasifica basándose en las proyecciones realizadas sobre las direcciones principales, es decir, aquéllas que maximizan la varianza, del conjunto de datos.
- *Fisherfaces:* la proyección se lleva a cabo sobre las direcciones determinadas por el análisis lineal discriminante de Fisher (LDA), las cuales maximizan el cociente inter/intra clase, donde cada clase corresponde a una persona distinta.

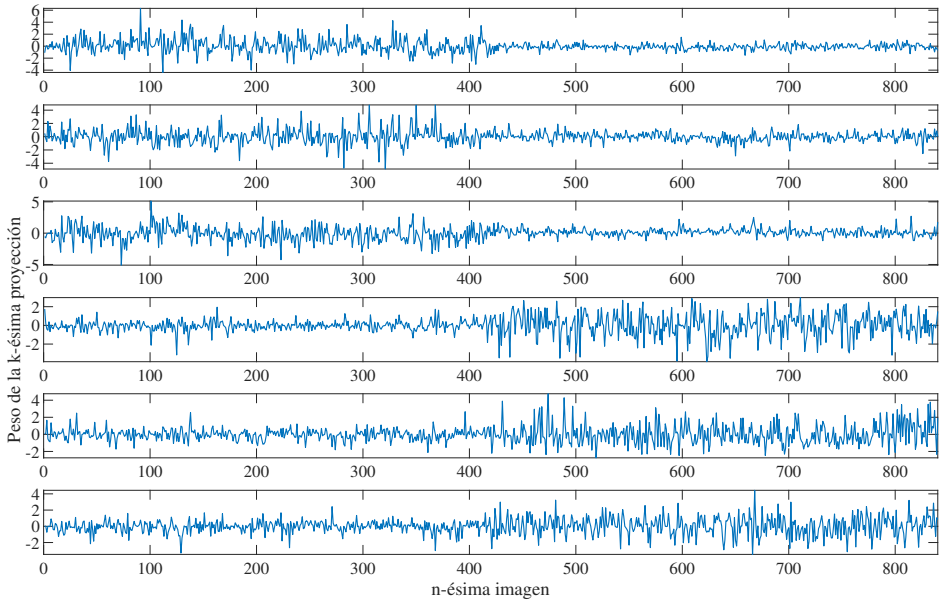


Figura 5.4.7 Proyecciones de radiografías de tórax *no usadas en el entrenamiento* sobre las imágenes prototipo. Las primeras 420 muestras pertenecen a imágenes de pacientes COVID; las últimas se asocian a personas sanas.

Se ha utilizado la base pública de rostros «Yale Face Database B»[5], la cual contiene 5760 imágenes de caras en disposición frontal para un total de 38 personas distintas (9 posiciones x 64 condiciones de iluminación), con un tamaño de 640 × 480 pixeles. En la Fig. 5.5.1 se muestran algunos de los individuos de la base de datos. Como al apilar las columnas de las matrices que almacenan las imágenes se obtiene vectores de una dimensión inmanejable, se ha realizado una reducción de dimensionalidad basado en análisis de componentes principales, reteniendo únicamente las 40 componentes más significativas. Para evaluar visualmente el efecto, la Fig. 5.5.2 muestra la reconstrucción de las imágenes de la Fig. 5.5.1 a partir de estas 40 componentes principales.

Al aplicar el algoritmo de minimización de la norma L1 a los vectores en el espacio de dimensión reducida, no tenemos previamente ninguna

[5] *http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html*

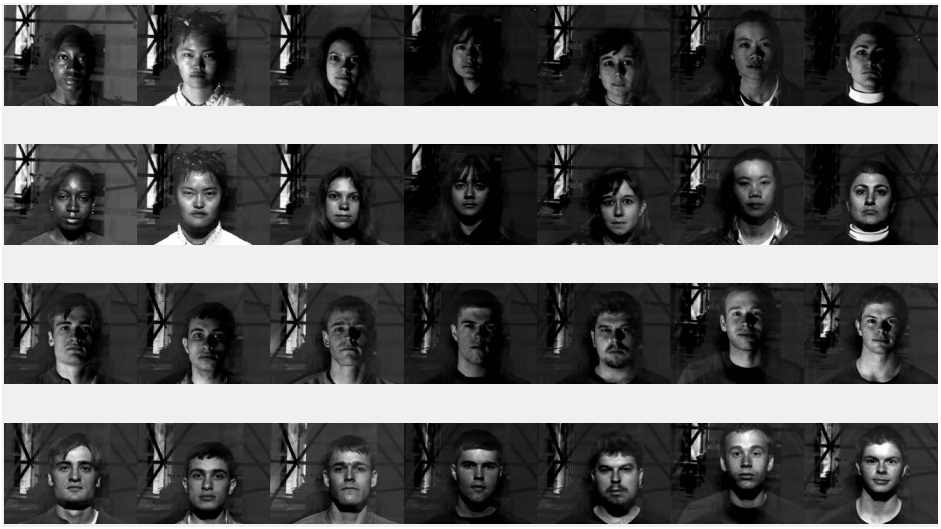


Figura 5.5.1 Base de datos de rostros.

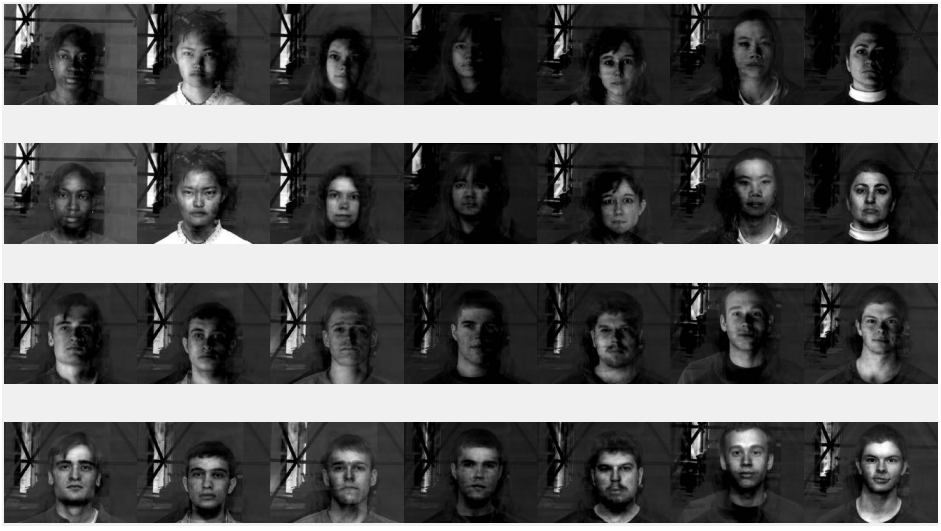


Figura 5.5.2 Base de datos de rostros tras la operación de reducción de dimensionalidad.

intuición sobre qué resultados se van a obtener. Esto se debe a que, como el algoritmo *no es supervisado*, es imposible orientar su respuesta en un sentido o en otro. Para poder interpretar la respuesta del algoritmo, hemos

Figura 5.5.3 Algunas imágenes «prototipo» de rostros generados por el algoritmo.

convertido en imágenes algunos de los vectores en la dirección que minimiza la norma L1. Esto se lleva a cabo invirtiendo el proceso de reducción de dimensionalidad. De esta forma, se consiguen las imágenes «prototipo» que muestra la Figura 5.5.3.

Curiosamente, podemos ver que la primera fila corresponde a rostros masculinos y la segunda fila a rostros femeninos. Seguidamente, proyectamos los vectores de las imágenes originales sobre los vectores de las imágenes «prototipo», obteniendo los coeficientes de proyección que se muestran en la Figura 5.5.4. Estos coeficientes de proyección están ordenados de manera que la primera mitad corresponde a imágenes de hombres y, la segunda mitad, a mujeres. Se aprecia que el algoritmo ha aprendido a distinguir entre unos y otras. Es decir, no ha realizado reconocimiento de rostros, sino una diferenciación según el sexo. De hecho, al realizar «clustering» con las proyecciones y el algoritmo de mezcla de Gaussianas, el 85.7 % de las imágenes queda asignada al grupo en el que su clase es mayoritaria. Hay que destacar que ningún sistema de inteligencia artificial tiene intencionalidad, ya que las decisiones que toma están basadas en los datos con los cuales ha sido entrenado, y si esos datos están sesgados (intencionadamente o no), el algoritmo tendrá un sesgo que puede hacerle confundir rostros masculinos

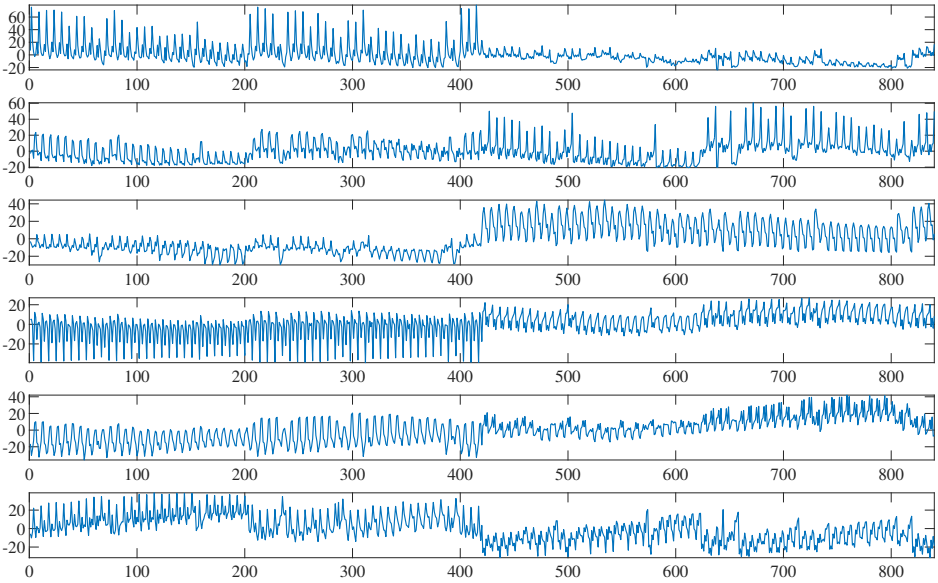


Figura 5.5.4 Magnitud de las proyecciones de las imágenes de la base de datos sobre seis de las imágenes «prototipo» determinadas por el algoritmo. Las imágenes se han ordenado de manera que las primeras 420 corresponden a hombres y, la segunda mitad, a mujeres.

con femeninos o viceversa. En general, nuestros experimentos con la base de datos muestran que el reconocimiento de sexos es fiable siempre que existan unas condiciones del entorno adecuadas (iluminación, ángulo de la cara para diferenciar la forma) para así poder detectar las diferencias, por ejemplo, en el mentón y la barbilla, que tienden a tener una forma más cuadrada en los hombres que en las mujeres.

5.6 Clasificación de ganado ovino

Para este último experimento usaremos un conjunto de imágenes de ovejas de diferentes razas. Específicamente, 420 imágenes en color de la raza «merina» y otras 420 de la raza «suffolk», cada una de tamaño 181×156

píxeles[6]. No obstante, todas las imágenes han sido convertidas de color a escala de grises para simplificar el problema de visión por ordenador.

La oveja merina es una raza «todoterreno» que se encuentra principalmente en España y está orientada hacia la producción de lana y carne. Su fenotipo presenta una buena alzada y desarrollo, cara y pezuñas blancas, mucosas rosadas, sin cuernos, cuerpo cubierto de lana y cara descubierta hasta la altura de los ojos. Por otro lado, la oveja «suffolk» es una raza de complexión fuerte (100 kg-160 kg), originaria de Inglaterra y resultante del cruce entre las razas «southdown» y «norfolk horn». También es criada por su lana, pero se la utiliza sobre todo para la producción de carne. Sus rasgos más característicos son la ausencia de lana en la cabeza y las patas cubiertas de pelaje de color negro opaco. En la Figura 5.6.1 se muestran algunas de las imágenes utilizadas en el experimento.

Las 840 imágenes se han convertido en vectores, a los que se ha restado el valor medio del conjunto. Seguidamente, se procede a reducir la dimensionalidad de los datos, proyectando los vectores en un espacio de 40 dimensiones utilizando análisis de componentes independientes (PCA) tradicional. Si reconstruimos estas proyecciones en el espacio original, invirtiendo PCA, podemos ver el resultado en la Figura 5.6.2. Los vectores en en espacio de 40 dimensiones, finalmente, se «blanquean» para cumplir las condiciones enunciadas en los Capítulos previos.

A simple vista puede parecer una tarea muy difícil clasificarlas, puesto que son todas muy similares. Sin embargo, el algoritmo que minimiza la norma L1 es capaz de encontrar diferencias entre ambas razas, como puede verse en la Fig. 5.6.3, que muestra algunas de las proyecciones sobre las direcciones que minimizan la norma L1.

La Fig. 5.6.4 muestra algunos de los animales «prototipo» obtenidos al convertir los vectores que minimizan la norma L1 en imágenes, tras deshacer la reducción de dimensionalidad. Finalmente, como en los experimentos anteriores, hemos proyectado las imágenes originales sobre estos prototipos. Los coeficientes de proyección obtenidos se han utilizado como entrada a un algoritmo de k-medias para realizar «clustering». La matriz de confusión

[6] Ver *https://www.kaggle.com/divyansh22/sheep-breed-classification*

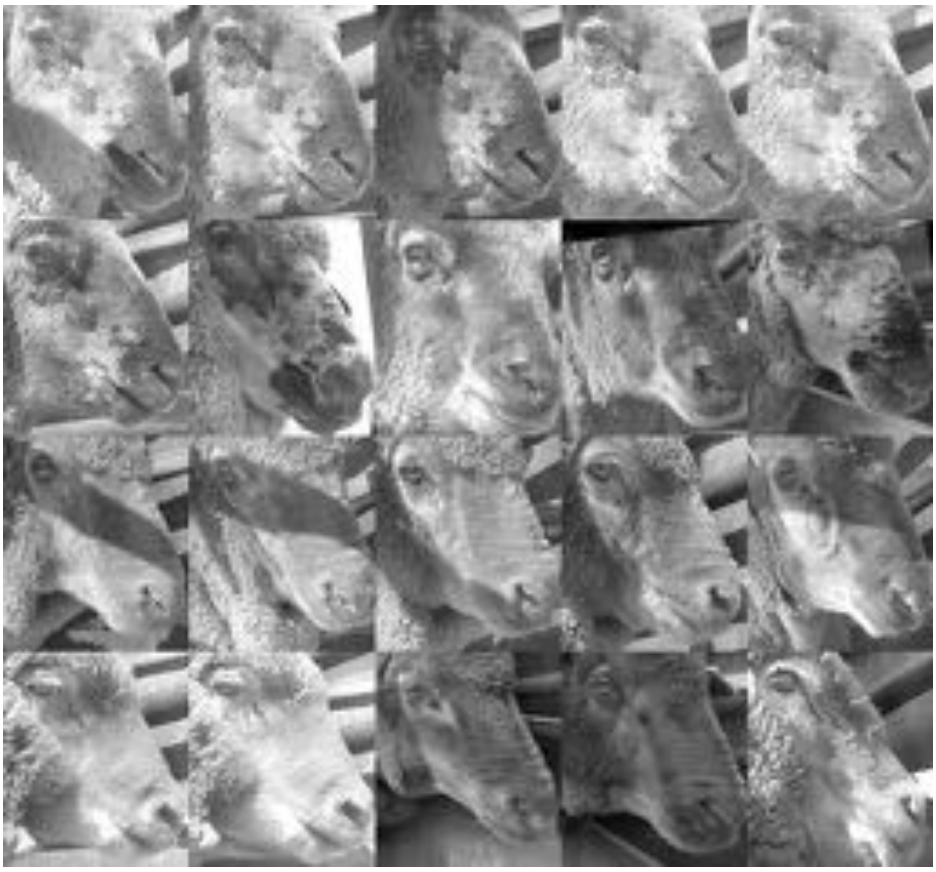


Figura 5.6.1 Primeras dos filas: ovejas «suffolk»; filas inferiores: «merinas».

obtenida tras comparar los grupos obtenidos con las clases originales se muestra en la Fig. 5.6.5. Podemos apreciar que el algoritmo ha sido capaz de distinguir entre razas con una precisión cercana al 80 %.

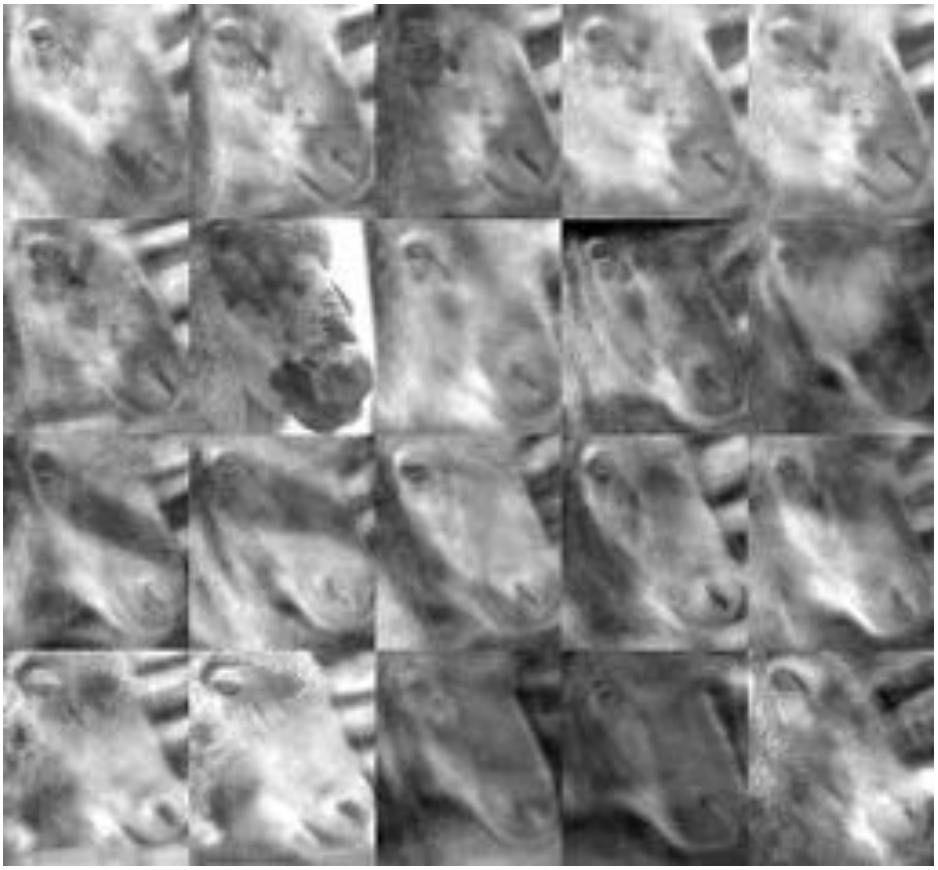


Figura 5.6.2 Imágenes en el espacio de dimensión reducida.

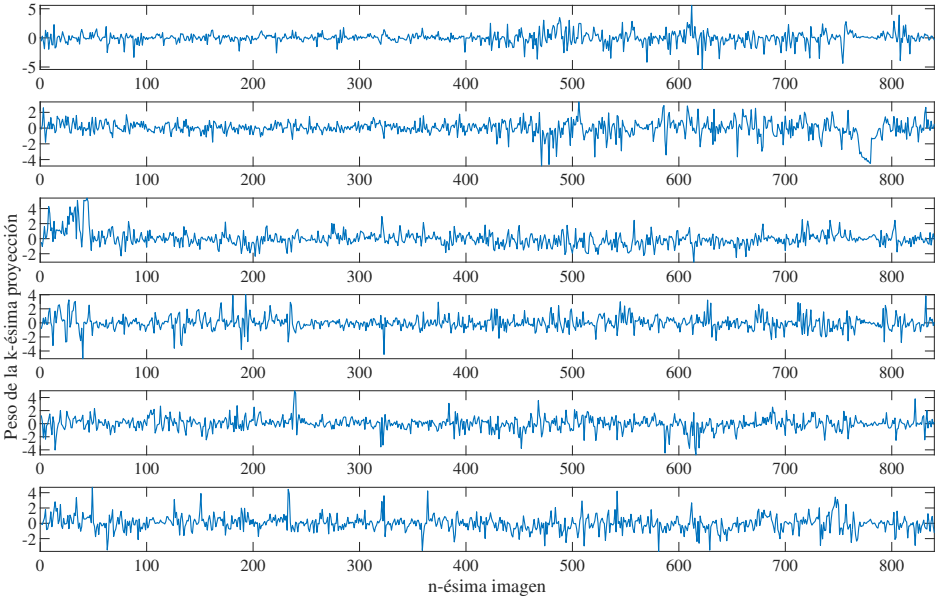


Figura 5.6.3 Coeficientes sobre seis de las direcciones que minimizan la norma L1. En cada caso, los primeros 420 coeficientes corresponden a ovejas «suffolk» y los restantes, a «merina».

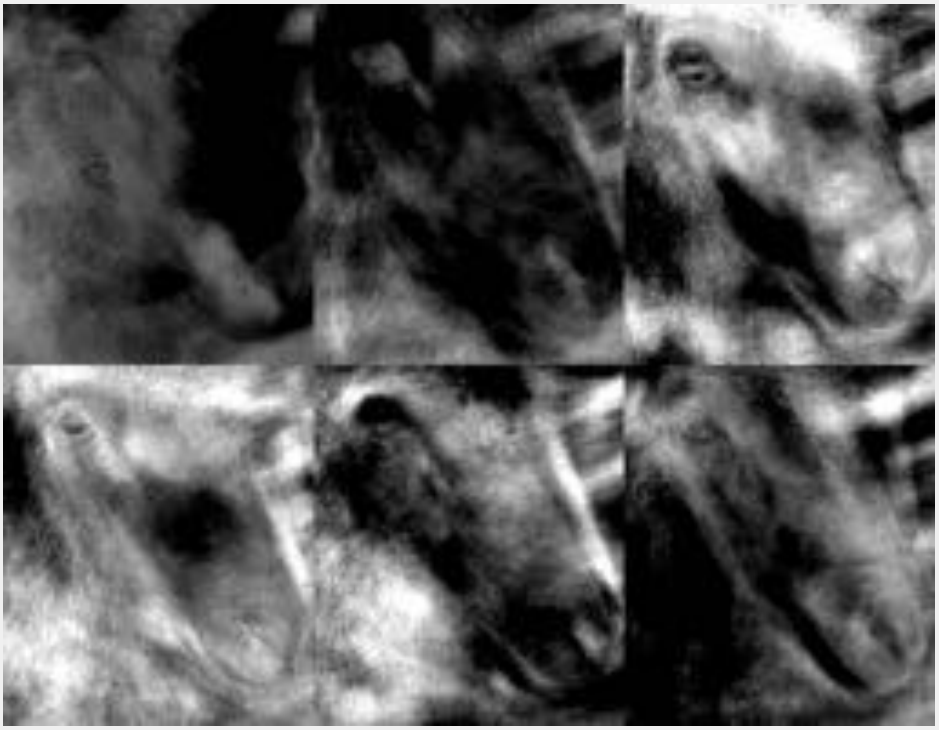


Figura 5.6.4 Ovejas «prototipo» determinadas por el algoritmo.

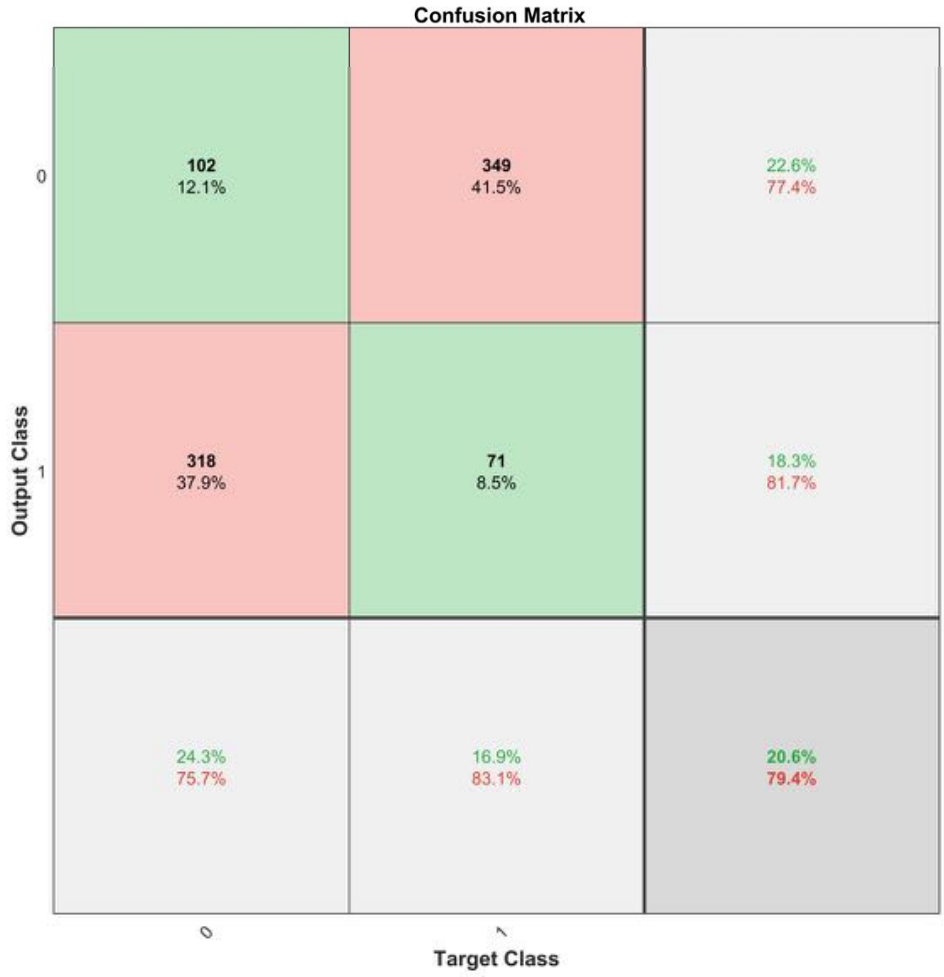


Figura 5.6.5 Matriz de confusión obtenida en la clasificación no supervisada de las razas de ovejas.

6 Conclusiones

Proyectando los datos blanqueados en las direcciones que minimizan el valor absoluto de las proyecciones se lleva a cabo una FKT no supervisada, evitando con ello la necesidad de conjuntos de entrenamiento. Esta conexión entre la norma L1 y la FKT ha pasado desapercibida hasta el momento y dota a los criterios L1 de propiedades discriminativas en escenarios de clasificación binaria, abriendo nuevas líneas de investigación en el área de L1-PCA. La Tesis también demuestra que minimizar la norma L1 se puede llevar a cabo con un algoritmo de descenso del gradiente en el conjunto de la matrices ortogonales. Aunque nuestro análisis teórico asume la Gaussianeidad de los datos, los experimentos numéricos muestran que también se logran buenos resultados cuando no se cumple esta hipótesis.

A Artículo 1

José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm unsupervised Fukunaga-Koontz transform», *Signal Processing*, vol. 182, mayo 2021, *https://doi.org/10.1016/j.sigpro.2020.107942*.

Signal Processing 182 (2021) 107942

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

L1-norm unsupervised Fukunaga-Koontz transform☆

José Luis Camargo[a], Rubén Martín-Clemente[a,*], Susana Hornillo-Mellado[a], Vicente Zarzoso[b]

[a] Signal Theory and Communications Department, Universidad de Sevilla, Spain
[b] Université Côte d'Azur, CNRS, I3S Laboratory, Sophia Antipolis, France

ARTICLE INFO

Article history:
Received 2 May 2020
Revised 21 October 2020
Accepted 11 December 2020
Available online 24 December 2020

MSC:
02.50.Sk
43.60.Cg
07.50.Qx

Keywords:
Fukunaga-Koontz
Common spatial patterns
Tuned-based functions
L1-PCA

ABSTRACT

The Fukunaga-Koontz transform (FKT) is a powerful supervised feature extraction method used in two-class recognition problems, particularly when the classes have equal mean vectors but different covariance matrices. The present work proves that it is also possible to perform the FKT in an unsupervised manner, sparing the need for labeled data, by using a variant of L1-norm Principal Component Analysis (L1-PCA) that minimizes the L1-norm in the feature space. Rigorous proof is given in the case of data drawn from a mixture of Gaussians. A working iterative algorithm based on gradient-descent in the Stiefel manifold is put forward to perform L1-norm minimization with orthogonal constraints. A number of numerical experiments on synthetic and real data confirm the theoretical findings and the good convergence characteristics of the proposed algorithm.

1. Introduction

The Fukunaga-Koontz transform (FKT) is a popular feature-extraction method used in binary classification problems [1]. It projects the data onto directions along which the variance is much larger for one class than for the other. The classification rule thus exploits the difference in variance between the two projected classes.

The true potential of the FKT is revealed when the two classes share the same mean vector, giving overlapping sets of data [2,3]. For equal-mean class distributions, reference [4] shows that the FKT is equivalent to the optimal Chernoff criterion introduced in [5], and thus preserves after the projection as much as possible of the Chernoff distance between both populations [6]. The FKT is also closely related to optimal Linear and Quadratic Discriminant Analysis and the Generalized Singular Value Decomposition, as has been shown in the literature [2,7,8]. Thanks to all its properties, the FKT has been successfully applied to image classification problems (where it is also known as the method of the 'tuned based functions') [9–13] and in EEG signal processing under the name of 'common spatial patterns' [14,15], as well as in other areas of practical interest [16,17].

Our contribution is to show, for the first time, a link between the FKT and a variant of Principal Component Analysis (PCA) called L1-PCA, which is receiving increasing interest due to its resistance to outliers [18–20] and its connections to Independent Component Analysis and Linear Discriminant Analysis [21,22]. L1-PCA linearly projects the data onto a few dimensions that maximize the absolute value of the projected data points. Just changing the word 'maximize' to 'minimize', while retaining the absolute value as objective function, this paper shows that it is also possible to calculate the Fukunaga-Koontz directions of projection. A rigorous proof of this result is given for the case of Gaussian populations with zero mean but different covariance matrices, whereas for non-Gaussian data we provide an experimental demonstration of this result. We only require the raw data points be pre-whitened to remove their covariance structure. The theoretical importance of the above result is that it relates these two apparently disparate techniques, allowing us to re-interpret the absolute value as a feature-extraction criterion in binary classification problems, which opens new lines of research in this area.

Furthermore, apart from its theoretical interest, this result also has practical relevance because the standard FKT is a supervised

☆ This work is funded by the research project ACACIA (refno. US-1264994 US/JUNTA/FEDER, UE) awarded by Fondo Europeo de Desarrollo Regional (FEDER) and Junta de Andalucía (Consejería de Economía, Conocimiento, Empresas y Universidad).

* Corresponding author.
E-mail addresses: jlcamargo@yahoo.es (J.L. Camargo), ruben@us.es (R. Martín-Clemente), susanah@us.es (S. Hornillo-Mellado), vicente.zarzoso@univ-cotedazur.fr (V. Zarzoso).

https://doi.org/10.1016/j.sigpro.2020.107942

Table 1
Notation and symbols used in this paper.

$\mathbf{0}$	vector of zeros
$\boldsymbol{I}$	identity matrix
$(\cdot)^\dagger$	matrix transpose operator
$\det(\cdot)$, $\mathrm{trace}(\cdot)$	determinant and trace of a matrix
X	p-dimensional random variable
$\boldsymbol{x}$	observation of X
$P(\mathcal{C}_i)$	probability of the observed data being drawn from class $\mathcal{C}_i$
$\mathrm{E}\{\cdot\}$	mathematical expectation
$\mathrm{E}\{\cdot \mid \mathcal{C}_i\}$	conditional expectation given the class $\mathcal{C}_i$
$f(\cdot \mid \mathcal{C}_i)$	conditional probability density function (pdf) given $\mathcal{C}_i$
$\boldsymbol{\mu}_i = \mathrm{E}\{X \mid \mathcal{C}_i\}$	mean of class i
$\boldsymbol{\Sigma}_i$	covariance of class i
$\|\boldsymbol{x}\|$	L2-norm of vector $\boldsymbol{x}$
$\|\boldsymbol{X}\|$	Frobenius norm of matrix $\boldsymbol{X}$

technique, which requires a training set of correctly class-labeled data points to estimate the parameters of the transformation; however, by contrast, minimizing the absolute value can be performed in a fully *unsupervised* fashion, making unnecessary the acquisition of training data and opening the door to the computation of the FKT in the same way.

The paper is organized as follows: we first present in Section 2 some general assumptions made in the paper. Section 3 briefly reviews the FKT. In Sections 4 and 5, we state our main results and propose a numerical algorithm for unsupervised FKT based on L1-norm minimization. Section 6 provides a number of numerical experiments that validate our findings in a variety of scenarios. Finally, Section 7 brings the paper to an end. Note that mathematical proofs have been deferred to the Appendices for the reader's convenience.

2. Preliminaries: Notation and basic hypotheses

The following assumptions hold throughout the paper. Let $X \in \mathbb{R}^p$ be a random vector whose samples $\boldsymbol{x}$ are drawn at random from one of two populations, $\mathcal{C}_1$ and $\mathcal{C}_2$. We suppose that $\mathcal{C}_1$ and $\mathcal{C}_2$ have common mean vectors $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ but different covariance matrices $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$. For simplicity, we also suppose that these matrices do not have repeated eigenvalues. Other symbols and notations used in this paper can be found in Table 1.

It is assumed as well that the data have been *centered* (by subtracting the mean across all observations) and *whitened* (or *sphered*). *Centering* implies that

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \mathbf{0}. \tag{1}$$

Whitening consists in transforming the data to have identity covariance matrix:

$$\begin{aligned}\boldsymbol{\Sigma} &= \mathrm{E}\{X X^\dagger\} \\ &= P(\mathcal{C}_1)\mathrm{E}\{X X^\dagger \mid \mathcal{C}_1\} + P(\mathcal{C}_2)\mathrm{E}\{X X^\dagger \mid \mathcal{C}_2\} \\ &= P(\mathcal{C}_1)\boldsymbol{\Sigma}_1 + P(\mathcal{C}_2)\boldsymbol{\Sigma}_2 \\ &= \boldsymbol{I}. \end{aligned} \tag{2}$$

Like centering, whitening can be assumed without any loss of generality: it can be always fulfilled by a simple pre-processing step (see Section 5.1). Whitening is useful, as we will show in the next Section, because it intertwines the class covariances as follows:

Property 1. *After whitening,*

$$\boldsymbol{\Sigma}_1 = \frac{1}{P(\mathcal{C}_1)}\,[\boldsymbol{I} - P(\mathcal{C}_2)\boldsymbol{\Sigma}_2]. \tag{3}$$

Eq. (3) readily follows from Eq. (2). Thanks to this intertwining, we will see in the next Section that $\mathcal{C}_1$ and $\mathcal{C}_2$ lie (approximately) in orthogonal subspaces.

3. The Fukunaga-Koontz transform

Let $(\lambda, \boldsymbol{v})$ be any eigenpair of $\boldsymbol{\Sigma}_1$, i.e.,

$$\boldsymbol{\Sigma}_1 \boldsymbol{v} = \lambda\, \boldsymbol{v}. \tag{4}$$

Using (3), it readily follows that

$$\frac{1}{P(\mathcal{C}_1)}\,[\boldsymbol{I} - P(\mathcal{C}_2)\boldsymbol{\Sigma}_2]\boldsymbol{v} = \lambda\, \boldsymbol{v}$$

and hence

$$\boldsymbol{\Sigma}_2 \boldsymbol{v} = \frac{1 - P(\mathcal{C}_1)\,\lambda}{P(\mathcal{C}_2)}\,\boldsymbol{v}.$$

That is: if $\boldsymbol{v}$ is any eigenvector of $\boldsymbol{\Sigma}_1$ with eigenvalue λ, then $\boldsymbol{v}$ is also an eigenvector of $\boldsymbol{\Sigma}_2$ with eigenvalue

$$\mu = \frac{1 - P(\mathcal{C}_1)\,\lambda}{P(\mathcal{C}_2)}.$$

This transformation is strictly decreasing: if the eigenvalues λ_i of $\boldsymbol{\Sigma}_1$ are ordered from largest to smallest as

$$\lambda_1 > \lambda_2 > \ldots > \lambda_p,$$

it follows that the corresponding eigenvalues of $\boldsymbol{\Sigma}_2$ are reversely ordered as

$$\mu_1 < \mu_2 < \ldots < \mu_p$$

so that the dominant eigenvectors of $\boldsymbol{\Sigma}_1$ are the least dominant eigenvectors of $\boldsymbol{\Sigma}_2$ and *vice versa*. In the language of classical PCA [23], the directions in which the data from class 1 vary the most are also the directions where class 2 varies the least. The opposite is also true: the directions of greatest variance for class 2 are those of lowest variance for class 1. This makes the two classes easier to distinguish.

Furthermore, the averaged squared distance between the data points from one class and the subspace spanned by the dominant eigenvectors of their class covariance matrix is minimal. This is, in this sense, the best-fitting subspace [23].

Feature extraction and classification can be based on exploiting all these properties. The FKT transforms each data point by orthogonally projecting it onto the span of the eigenvectors of $\boldsymbol{\Sigma}_1$ corresponding to the largest and smallest eigenvalues. If the data point is closer to the subspace spanned by the first few dominant eigenvectors than to the subspace spanned by the least dominant eigenvectors, we can assume the presence of a sample of $\mathcal{C}_1$. The opposite suggests allocating it to class $\mathcal{C}_2$. Several variants of this basic approach have been also proposed, see [9–13].

Note finally that, in standard FKT, matrices $\boldsymbol{\Sigma}_1$ or $\boldsymbol{\Sigma}_2$ have to be estimated *a priori* from a set labelled samples. We remark that, for this reason, the standard FKT is a *supervised* technique.

4. Main contribution: Unsupervised FKT via L1-norm minimization

Unsupervised calculation of the FKT is however possible by minimizing the L1-norm of the projection: in this Section we prove this property in the Gaussian case. Gaussian models are justified by their simplicity and ability to produce accurate results in practice, even when violated. In particular, we make the usual assumption that $f(\boldsymbol{x}|\mathcal{C}_i)$ is a p-variate normal density function of the form

$$f(\boldsymbol{x}|\mathcal{C}_i) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{\Sigma}_i)^{-\frac{1}{2}}\, e^{-\frac{1}{2}\boldsymbol{x}^\dagger \boldsymbol{\Sigma}_i^{-1}\boldsymbol{x}}, \quad i = 1, 2. \tag{5}$$

The global distribution of X is given by the mixture

$$f(\boldsymbol{x}) = P(\mathcal{C}_1)\, f(\boldsymbol{x}|\mathcal{C}_1) + P(\mathcal{C}_2)\, f(\boldsymbol{x}|\mathcal{C}_2).$$

Let $Y = \boldsymbol{w}^\dagger X$ be the projection of X into the direction defined by $\boldsymbol{w} \in \mathbb{R}^p$. Only the direction is important, so we can assume $\boldsymbol{w}$ to be

a vector of unit length. From basic statistics, the probability density function of Y is a mixture of Gaussians, i.e.,

$$f(y) = \sum_{k=1,2} \frac{P(\mathcal{C}_k)}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{y^2}{2\sigma_k^2}\right), \tag{6}$$

where $\sigma_k^2 = \boldsymbol{w}^\dagger \boldsymbol{\Sigma}_k \boldsymbol{w}$.

Now, we are interested in minimizing

$$D(\boldsymbol{w}) = \mathrm{E}\{|Y|\} = \mathrm{E}\{|\boldsymbol{w}^\dagger X|\} \tag{7}$$

over all possible projections defined by direction $\boldsymbol{w}$. Criterion $D(\boldsymbol{w})$ is quickly gaining popularity in the field of PCA for the following reason: standard PCA [23] aims to maximize the variance of Y which, for zero-mean data, is given by $\mathrm{E}\{Y^2\}$. Because it is raised to the square power, samples that are far apart from the nominal body of the data completely dominate the value of the variance. Therefore, standard PCA is very sensitive to outliers. An alternative is obtained by replacing Y^2 with $|Y|$, and it is in this way that we arrive at criterion D [18–20]. This variant is called L1-PCA because $D(\boldsymbol{w})$ is estimated in practice by the L1-norm of the vector that contains the samples of Y [18]. Again, we remark that [18,19] focus on *maximizing* $D(\boldsymbol{w})$, while we propose just the opposite.

The directions that solve the constrained optimization problem

$$\min_{\boldsymbol{w}} D(\boldsymbol{w}) \text{ subject to } \|\boldsymbol{w}\|^2 = 1 \tag{8}$$

verify

$$\nabla_{\boldsymbol{w}} D(\boldsymbol{w}) = \ell \, \nabla_{\boldsymbol{w}} \|\boldsymbol{w}\|^2, \tag{9}$$

where ℓ is a Lagrange multiplier and $\nabla_{\boldsymbol{w}}$ stands for the gradient with respect to $\boldsymbol{w}$. Under the Gaussian assumption (6), and after some algebraic derivations detailed in Appendix A, we obtain that

$$\nabla_{\mathbf{w}} D(\mathbf{w}) = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k} \boldsymbol{\Sigma}_k \mathbf{w}, \tag{10}$$

$$\nabla_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 = 2\boldsymbol{w}, \tag{11}$$

$$\ell = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{2} P(\mathcal{C}_k)\sigma_k, \tag{12}$$

and hence the solutions of (9) satisfy

$$\sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k} \boldsymbol{\Sigma}_k \boldsymbol{w} = \left(\sum_{k=1}^{2} P(\mathcal{C}_k)\sigma_k\right) \boldsymbol{w}. \tag{13}$$

Invoking the whitening constraint (3), we get:

$$\left(\frac{\sigma_2 - \sigma_1}{\sigma_1}\right) P(\mathcal{C}_1) \boldsymbol{\Sigma}_1 \boldsymbol{w} = \left[\left(\sum_{k=1}^{2} P(\mathcal{C}_k)\sigma_k\right)\sigma_2 - 1\right] \boldsymbol{w}. \tag{14}$$

Then, by replacing the rightmost '1' with

$$1 = \boldsymbol{w}^\dagger \boldsymbol{\Sigma} \boldsymbol{w} = \sum_{k=1}^{2} P(\mathcal{C}_k)\sigma_k^2,$$

which follows from (2), and simplifying terms, the equation becomes:

$$(\sigma_2 - \sigma_1)\boldsymbol{\Sigma}_1 \boldsymbol{w} = (\sigma_2 - \sigma_1)\,\sigma_1^2\, \boldsymbol{w}. \tag{15}$$

Thus, apart from the solution $\sigma_1 = \sigma_2$ (which defines a maximum, see Appendix B), we find that:

Lemma 1. *Under the working assumption (6), the eigenvectors of* $\boldsymbol{\Sigma}_1$ *(or* $\boldsymbol{\Sigma}_2$*) are stationary points of (8).*

This result is complemented by the following one, which describes the minimizers:

Theorem 1. *For a p-dimensional random vector X distributed as a mixture of two multivariate Gaussian distributions with zero mean and different covariance matrices* $\boldsymbol{\Sigma}_1$ *and* $\boldsymbol{\Sigma}_2$ *verifying the whitening constraint (3), the minimizers of* $D(\boldsymbol{w})$ *are the eigenvectors associated with the maximum and minimum eigenvalues of* $\boldsymbol{\Sigma}_1$ *(or* $\boldsymbol{\Sigma}_2$*). The intermediate eigenvectors are saddle points, but are orthogonal to each other and can be found by a suitable optimization approach with orthogonal constraints.*

Proof and details are given in Appendix B. This result hence suggests an *unsupervised* approach to compute the FKT, based on the above L1-norm criterion. Furthermore, it endows the proposed criterion with discriminative properties in the case of equal-mean populations. Even though the theorem is derived by assuming Gaussian densities, it is still useful even when there are wide deviations from Gaussianity in the data distributions, as we will see in the experiments of Section 6.

5. Algorithm

Let us now propose a working algorithm for finding a set of appropriate projection vectors based on the above criterion. The algorithm is fully unsupervised, i.e., it does not require the labels of the data points.

5.1. Preprocessing

A few words about the whitening constraint (2) may be needed in the first place. To fulfill this condition, we will often require in practice a pre-processing of the data. Specifically, given a 'colored' (i.e., non-white) random vector $X_c \in \mathbb{R}^p$, assumed to have zero mean, whitened data X can be obtained, for example, as follows [24]:

$$X = \boldsymbol{\Gamma}^{-1/2} \boldsymbol{V}^\dagger X_c,$$

where $\boldsymbol{V}$ is the matrix whose columns are the eigenvectors of $\mathrm{E}\{X_c X_c^\dagger\}$, and $\boldsymbol{\Gamma}$ is the diagonal matrix of its eigenvalues (note that there exist other whitening approaches that are equally valid [24]). It is straightforward to check that, as desired, $\boldsymbol{\Sigma} = \mathrm{E}\{X X^\dagger\} = \boldsymbol{I}$.

5.2. Algorithm for joint L1-norm minimization

The maximization of the L1-norm criterion $D(\boldsymbol{w})$ defined in (7)–(8) has already been studied in a number of recent works [18,19,25]. Unfortunately, because of how they have been designed, none of these ad-hoc maximization approaches can be turned into a minimization algorithm. Therefore, we opt here for a gradient-based approach.

As the eigenvectors of the class covariance matrices are always orthogonal, they can be determined by successively minimizing $D(\boldsymbol{w})$ under the constraint that the direction obtained in the current minimization is orthogonal to the previously computed ones (see Appendix B). However, this simple deflation approach has the disadvantage of accumulating estimation errors along successively calculated directions. To avoid these drawbacks, we consider the cost function

$$J(\boldsymbol{W}) = \sum_{i=1}^{p} D(\boldsymbol{w}_i),$$

where $\boldsymbol{W}$ is the matrix $[\boldsymbol{w}_{:,1}\, \boldsymbol{w}_{:,2} \cdots \boldsymbol{w}_{:,p}]$ containing the projection vectors, where $\boldsymbol{w}_{:,n} = \boldsymbol{w}_n$ denotes its n-th column. We are interested in its minimization with orthogonality constrains

$$\min_{\boldsymbol{W}} J(\boldsymbol{W})$$

$$\text{s.t. } \boldsymbol{W}^\dagger \boldsymbol{W} = \boldsymbol{I}. \tag{16}$$

Following classical results of optimization over the set of orthogonal matrices [26], we adopt a minimization approach based on gradient-descent in the Stiefel manifold. As justified in Appendix D, this approach leads to the following multiplicative update scheme, which preserves the orthogonality constraint during iterations:

$$\boldsymbol{W}_{n+1} = \boldsymbol{U}_n \boldsymbol{W}_n, \tag{17}$$

where $\boldsymbol{U}_n = \exp(\boldsymbol{S}_n) = \boldsymbol{I} + \boldsymbol{S}_n + \frac{1}{2!}\boldsymbol{S}_n^2 + \dots$ Choosing a skew-symmetric matrix $\boldsymbol{S}_n$, i.e., $\boldsymbol{S}_n = -\boldsymbol{S}_n^\dagger$, guarantees the orthogonality of matrix $\boldsymbol{U}_n$, and thus that of $\boldsymbol{W}_{n+1}$. Apart from fulfilling this condition, matrix $\boldsymbol{S}_n$ is closely related to the gradient of J, allowing update (17) to perform gradient descent, as detailed next.

The algorithm can be described as follows. Starting from any initial orthogonal matrix $\boldsymbol{W}_0$, repeat the following three steps for $n = 0, 1, 2, \dots$ until convergence [26]:

1: Set

$$\boldsymbol{S}_n = \partial J(\boldsymbol{W}_n)\boldsymbol{W}_n^\dagger - \boldsymbol{W}_n \partial J(\boldsymbol{W}_n)^\dagger$$

where $\partial J(\boldsymbol{W})$ is the matrix of partial derivatives of J with respect to the elements of $\boldsymbol{W}$, i.e.,

$$(\partial J(\boldsymbol{W}))_{ij} = \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}_{ij}} \tag{18}$$

(this equation will be detailed below).

2: Define $\boldsymbol{U}_n = \exp(-\eta\, \boldsymbol{S}_n)$ for $\eta \in \mathbb{R}^+$ small enough.

3: Update $\boldsymbol{W}_{n+1} = \boldsymbol{U}_n \boldsymbol{W}_n$.

It still remains to give a formula for $\partial J(\boldsymbol{W})$ in Eq. (18). Subderivatives, or subgradients, generalize the notion of derivative to non-differentiable convex functions [27]. As the subderivative of the absolute value is the sign function, it is easily found that the nth column of $\partial J(\boldsymbol{W})$ equals

$$\frac{\partial}{\partial \boldsymbol{w}_n} D(\boldsymbol{w}_n) = \mathrm{E}\left\{X \mathrm{sgn}\left(\boldsymbol{w}_n^\dagger X\right)\right\},$$

with D defined in (7). As a simple illustration, given a $p \times q$ data matrix `M_x` containing q observed samples of X, $\partial J(\boldsymbol{W})$ can be evaluated by the MATLAB® command `M_x*sign(W'*M_x)'/q;`. Observe that this calculation does not require any knowledge of the class data labels.

5.2.1. Interpretation of the method

The above algorithm can be easily viewed as a gradient descent approach. To see this, we note that, for small η,

$$\boldsymbol{U}_n = \exp(-\eta\, \boldsymbol{S}_n) \approx \boldsymbol{I} - \eta \boldsymbol{S}_n \tag{18}$$

and therefore

$$\boldsymbol{W}_{n+1} = \boldsymbol{U}_n \boldsymbol{W}_n \approx \boldsymbol{W}_n - \eta\, \boldsymbol{S}_n \boldsymbol{W}_n. \tag{19}$$

Bounds on approximation (18) are given in Appendix C, and show its pertinence for sufficiently small η. Interestingly, the term

$$\begin{aligned} \nabla J(\boldsymbol{W}_n) &= \boldsymbol{S}_n \boldsymbol{W}_n \\ &= \partial J(\boldsymbol{W}_n)\, \boldsymbol{W}_n^\dagger \boldsymbol{W}_n - \boldsymbol{W}_n \partial J(\boldsymbol{W}_n)^\dagger \boldsymbol{W}_n \\ &= \partial J(\boldsymbol{W}_n) - \boldsymbol{W}_n\, \partial J(\boldsymbol{W}_n)^\dagger \boldsymbol{W}_n \end{aligned} \tag{20}$$

is, up to an irrelevant scale factor, the gradient of J in the set of orthogonal matrices, i.e., the projection of the gradient of J on the tangent space of the Stiefel manifold. Details are given in Appendix D. This relation allows us to interpret (19) as an approximate gradient rule, i.e.,

$$\Delta \boldsymbol{W} = \boldsymbol{W}_{n+1} - \boldsymbol{W}_n \approx -\eta \nabla J(\boldsymbol{W}_n).$$

Now, consider the first-order Taylor expansion of J

$$J(\boldsymbol{W} + \Delta \boldsymbol{W}) = J(\boldsymbol{W}) + \langle \partial J(\boldsymbol{W}) | \Delta \boldsymbol{W} \rangle + \dots,$$

where $\langle \partial J(\boldsymbol{W}) | \Delta \boldsymbol{W} \rangle = \mathrm{trace}(\partial J(\boldsymbol{W})^\dagger \Delta \boldsymbol{W})$. By setting, as before,

$$\Delta \boldsymbol{W} = -\eta \nabla J(\boldsymbol{W}),$$

some algebra shows that

$$\langle \partial J(\boldsymbol{W}) | \Delta \boldsymbol{W} \rangle = -\frac{\eta}{2} \langle \nabla J(\boldsymbol{W}) | \nabla J(\boldsymbol{W}) \rangle,$$

which is always negative for sufficiently small adaption step η (otherwise, the first-order Taylor expansion is no longer valid) and, therefore, $J(\boldsymbol{W})$ decreases with every update as desired. Note finally that, if $\boldsymbol{W}$ contains the eigenvectors of $\boldsymbol{\Sigma}_1$ (or $\boldsymbol{\Sigma}_2$) in its columns, it follows from (9)–(12) that $\partial J(\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix containing twice the Lagrange multipliers (12). Therefore, matrix

$$\boldsymbol{S} = \partial J(\boldsymbol{W})\boldsymbol{W}^\dagger - \boldsymbol{W}\partial J(\boldsymbol{W})^\dagger$$

vanishes and the iteration stops.

6. Experimental assessment

Experiments are next performed in a variety of conditions. Tests are applied to both synthetic and real electroencephalographic (EEG) data sets.

6.1. Bivariate Gaussian data

Let us first consider a mixture in a bidimensional space (i.e., $p = 2$) of two equiprobable Gaussian classes with zero-means and respective covariances

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0.68 \\ 0.68 & 1 \end{pmatrix} \text{ and } \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & -0.68 \\ -0.68 & 1 \end{pmatrix}. \tag{21}$$

$\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ fulfill the whitening condition (2) and share the same eigenvectors, i.e.,

$$\begin{aligned} \boldsymbol{v}_1 &= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\dagger \approx (0.71, 0.71)^\dagger, \quad \boldsymbol{v}_2 \\ &= \left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\dagger \approx (-0.71, 0.71)^\dagger. \end{aligned} \tag{22}$$

We draw 50 samples from each class (100 samples in total), whose scatter plot is shown in Fig. 1. The lines through $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are plotted in dashed blue; as recalled in Section 3, classes $\mathcal{C}_1$ and $\mathcal{C}_2$ can be well reconstructed in the line spanned by one of these eigenvectors, and not so well in the line spanned by the other. Here, 'well' means that the average squared distance of the points to the line is minimized. We also draw in red the lines pointing in the direction of

$$\boldsymbol{w}_1 \approx (0.73, 0.69)^\dagger, \quad \boldsymbol{w}_2 \approx (-0.69, 0.73)^\dagger. \tag{23}$$

These are the minimizers of the L1-norm calculated by the unsupervised algorithm presented in Section 5. As $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are estimates of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, we can cluster the observations $\boldsymbol{x}$ into two groups, say A and B, based on their closeness to the subspaces spanned by $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$\begin{aligned} &\text{if } \left\|\boldsymbol{x} - \boldsymbol{w}_1(\boldsymbol{w}_1^\dagger \boldsymbol{x})\right\| < \left\|\boldsymbol{x} - \boldsymbol{w}_2(\boldsymbol{w}_2^\dagger \boldsymbol{x})\right\|, \\ &\quad \text{assign } \boldsymbol{x} \in A \text{ otherwise } \boldsymbol{x} \in B. \end{aligned} \tag{24}$$

By applying this rule, we obtain the confusion matrix shown in Table 2. There is one cluster composed of 35 samples of class 1 and only 13 of class 2, and a second group with 15 instances of class 1 and 37 of class 2. We see that most data points of the same class lie together, which is what one would expect from an unsupervised method. To compute the accuracy, we sum the values on the diagonal of the confusion matrix and divide by the number

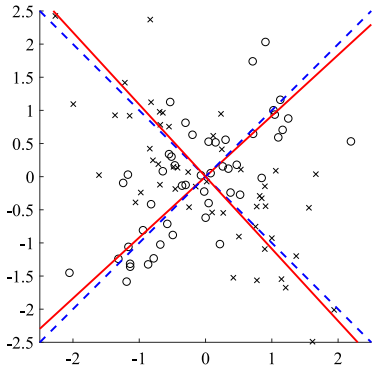


Fig. 1. Scatter plot of the observations of the two classes classes ('crosses' and 'circles'). In dashed blue, we show the lines through the eigenvectors of the class covariance matrices. Red lines point to the projection directions found by the unsupervised algorithm in Section 5. We observe that 'red' axes are rotated through an angle of 1.61° with respect to the 'blue' ones. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Confusion matrix obtained after applying the allocation rule (24).

		Actual class		
		$\mathcal{C}_1$	$\mathcal{C}_2$	Total
Assigned cluster	A	35	13	48
	B	15	37	52
	Total	50	50	100

of samples: overall, we have $35 + 37 = 72$ correctly clustered data points out of 100, implying that the method provides an accuracy of 72%.

Let us see it another way. For any given observed data value $\boldsymbol{x}$, (24) is equivalent to

$$|\boldsymbol{w}_1^\dagger \boldsymbol{x}| > |\boldsymbol{w}_2^\dagger \boldsymbol{x}|. \tag{25}$$

Fig. 2(a) shows that inequality (25) usually holds true for the elements of $\mathcal{C}_1$ as the orange line is usually above the green one. For the elements of $\mathcal{C}_2$, as seen in Fig. 2b, it is just the opposite. To quantify this inverse relationship, we compute Pearson's correlation coefficient of the absolute projections, defined as

$$\rho_{ij} = \frac{\mathrm{cov}(Z_i, Z_j)}{\sigma_i \sigma_j}$$

where $\mathrm{cov}(\cdot,\cdot)$ denotes the covariance of its input variables and σ_i is the standard deviation of $Z_i = |\boldsymbol{w}_i^\dagger X|$. The value of Pearson's correlation coefficient ρ_{12} between $Z_1 = |\boldsymbol{w}_1^\dagger X|$ and $Z_2 = |\boldsymbol{w}_2^\dagger X|$ becomes negative,

$$\rho_{12} = -0.254,$$

indicating that, on average, the magnitudes of the projected points for one class and for the other show opposite behavior.

6.1.1. Multivariate Gaussian and non-Gaussian data

When applied to p-dimensional data points from two equiprobable classes, the algorithm in Section 5 finds p projection vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_p \in \mathbb{R}^p$. To determine which of them correspond to the two most discriminant directions, one can choose the minimizers of $D(\boldsymbol{w})$ as in Theorem 1. We further introduce a slight refinement that experimentally improves the robustness of the classification against errors in the estimation of the eigenspace due to the finite sample size. Let us arrange these vectors so that: (i) $\boldsymbol{w}_1$ is the global minimizer of the L1-norm criterion, and (ii) among all remaining directions $\boldsymbol{w}_i$, $i > 1$, Pearson's correlation coefficient between $|\boldsymbol{w}_1^\dagger X|$ and $|\boldsymbol{w}_p^\dagger X|$ is the most negative, $\rho_{1p} < \rho_{1i}$ for $i \neq p$. As in the previous experiment, $\boldsymbol{w}_1$ and $\boldsymbol{w}_p$ are expected to represent different classes. Then, inspired by (24), we adopt the rule:

$$\text{Assign } \boldsymbol{x} \text{ to } A \text{ if: } \left\|\boldsymbol{x} - \boldsymbol{w}_1(\boldsymbol{w}_1^\dagger \boldsymbol{x})\right\| < \left\|\boldsymbol{x} - \boldsymbol{w}_p(\boldsymbol{w}_p^\dagger \boldsymbol{x})\right\|$$

otherwise, assign $\boldsymbol{x}$ to B. Fig. 3 shows the accuracy of this fully unsupervised classification approach, calculated as in the previous experiment, when tested on different data distributions and values of p. In each simulation, the covariance matrices are generated at random, and the data are whitened as in Section 5.1 to fulfill condition (2). In addition, we draw $N = 50\,p$ samples per each of the two classes, using the algorithms in [28,29] for generating multivariate non-Gaussian data with the specified covariances. These algorithms, widely used in robustness analysis, nonlinearly transform multivariate random Gaussian variables in a way that allows us to fix at will the mean, variance, skewness and kurtosis of the resulting marginal distributions. Specifically, we generate zero-mean, unit variance and zero-skew marginal data. Nevertheless, to explore different scenarios, we consider different values of excess kurtosis κ of the marginal data. Recall that the excess kurtosis is defined as the 4th-order central moment of the standardized (zero-mean, unit-variance) data minus three. In this experiment, κ is varied between -1 (which corresponds to a sub-Gaussian density) to 5 (highly super-Gaussian distribution), passing through 0 (Gaussian variable). Hence, we can test the performance of the algorithm in Section 5 when the assumption (5) for normality of data is not fulfilled. A notable feature is that, as seen in Fig. 3, the performance of the algorithm increases with the dimensionality of the input representation, which could be explained by the fact that it is generally easier to discriminate between classes in a feature space of higher dimensions.

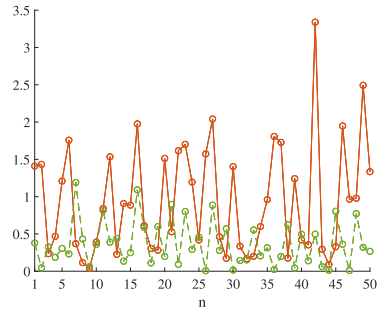
Table 3
Mean number of iterations (MNI) before the convergence of the algorithm as a function of the dimensionality p of the data, averaged over all distributions.

p	2	5	10	15	20	25	30
MNI	6.6	54.5	193.8	349.7	514.2	686.0	813.1
$\frac{\text{MNI}}{p}$	3.3	10.9	19.4	23.3	25.7	27.4	27.1

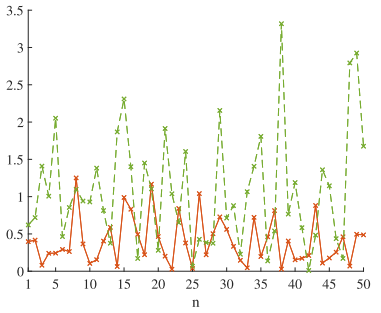
Furthermore, to speed up the algorithm, we have chosen in each iteration the step size η that gives the maximum reduction in the value of the cost function. Simple line-search algorithms, such as the golden-section search method, can be used to solve this problem [30]. Fig. 4 illustrates the convergence of the algorithm for the case of $p = 15$-dimensional data, suggesting that it is roughly independent of the value of the excess kurtosis of the data. In all cases, the algorithm stops when $\|\boldsymbol{W}_{n+1} - \boldsymbol{W}_n\| < 10^{-4}$, where $\boldsymbol{W}_n$ the value of matrix $\boldsymbol{W}$ after the n-th iteration. Table 3 shows the mean number of iterations, averaged over all distributions, before the convergence of the algorithm.

Additionally, L1-norm criteria are also expected to exhibit robustness against large outliers. To test this property, we repeat the experiment with the difference that the data points are now corrupted by replacing 10 per cent of the data samples, at randomly chosen time instants, by Gaussian noise realizations with identity covariance matrix and mean $\boldsymbol{\mu}_{\text{outliers}} = [10, 10, \ldots, 10]^\dagger$, which denotes a p-dimensional vector with all elements equal to 10.

In this new experiment, we have to take into account that the usual covariance estimate is very sensitive to the presence of outliers in the data set and, therefore, the whitening pre-processing, which is ultimately based on the eigendecomposition of that covariance matrix, inherits this sensitivity. To prevent this from affecting the experiment, whitening is performed by using a FastMCD robust estimator of the data covariance [31]. The new results



(a) Values of $|\boldsymbol{w}_1^\dagger \boldsymbol{x}_n|$ (continuous orange line) and and $|\boldsymbol{w}_2^\dagger \boldsymbol{x}_n|$ (dashed green line) for the 50 data points $\boldsymbol{x}_n$ in class $\mathcal{C}_1$.



(b) Values of $|\boldsymbol{w}_1^\dagger \boldsymbol{x}_n|$ (continuous orange line) and and $|\boldsymbol{w}_2^\dagger \boldsymbol{x}_n|$ (dashed green line) for the 50 data points $\boldsymbol{x}_n$ in class $\mathcal{C}_2$.

Fig. 2. Absolute value of the projected data points from (a) class $\mathcal{C}_1$ and (b) class $\mathcal{C}_2$.

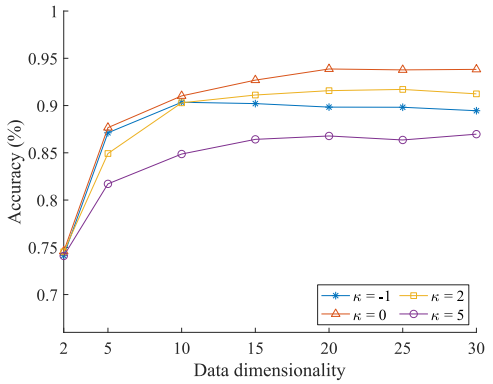


Fig. 3. Accuracy of the proposed method, as the dimensionality p of the data increases, for distributions with different excess kurtoses κ (e.g. $\kappa = -1.2$ corresponds to uniformly distributed marginals, $\kappa = 0$ to the Gaussian distribution or $\kappa = 3$ gives the Laplace distribution). Each curve has been obtained by averaging over 100 independent experiments.

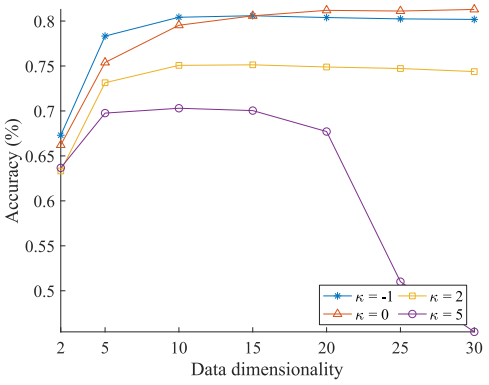


Fig. 5. Accuracy of the proposed method, for distributions with different excess kurtoses κ, when 10 per cent of the data samples are replaced, at randomly chosen time instants, by large outliers. The data covariance matrix, which is necessary for performing the whitening pre-processing, has been estimated by using a robust method.

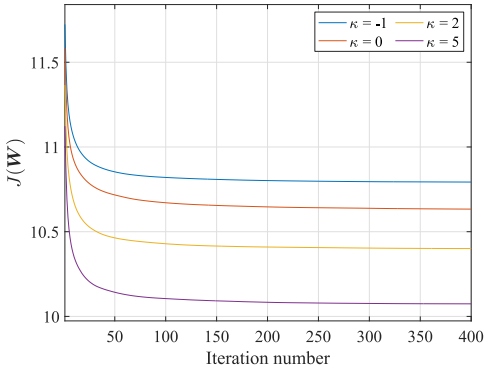


Fig. 4. Convergence of the algorithm as a function of the iteration number for distributions with different excess kurtoses κ and $p = 15$-dimensional data. The curves are obtained by averaging 100 independent experiments.

are represented in Fig. 5. Observe that the most leptokurtic distribution, that with $\kappa = 5$, seems to be severely affected by the presence of outliers. To confirm if this is true, an additional simulation is performed in which the data were whitened before outliers were added, i.e., the covariance matrix was calculated from the outlier-free observations. Fig. 6 shows that the improvement obtained with respect to the previous case is remarkable. We conclude that it is the whitened pre-processing step, which requires estimating a covariance matrix, which actually limits the ability of the proposed technique to fight against outliers. Thus, to fully exploit the capabilities of the L1-norm algorithm, it must be combined with a robust covariance estimator that guarantees that the pre-whitening step is also resistant to outliers. This is not actually surprising, as the traditional FKT also requires a robust estimation of the class covariance matrices.

6.2. Real electroencephalographic (EEG) data

In motor imagery-based brain computer interfaces (BCI's), the user imagines a limb moving and the system tries to identify the imagined movement by analyzing the EEG data recorded dur-

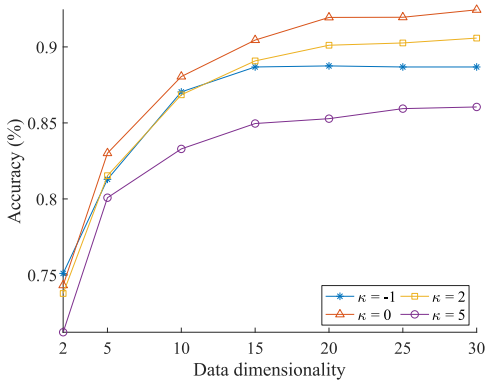


Fig. 6. Accuracy of the proposed method, for distributions with different excess kurtoses κ, when 10% of the data samples are replaced, at randomly chosen time instants, by large outliers. The whitening transformation is performed by the true (outlier-free) covariance matrix of the observations.

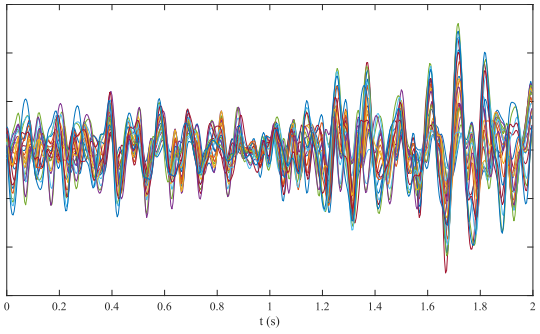


Fig. 7. Butterfly plot of 22-channel EEG recorded while subject number 1 imagines movements of tongue.

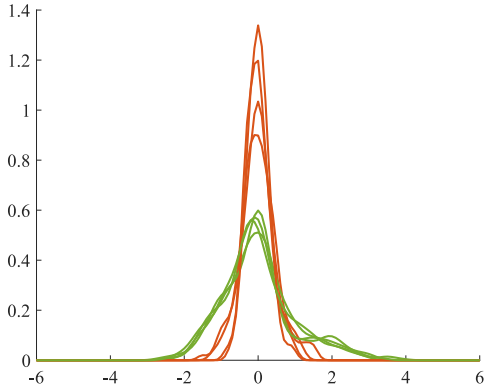


Fig. 8. Density functions (produced with a kernel density estimation method, with Gaussian kernel and Silverman's optimal bandwidth) of the scalar projections of the data from several projected 'left hand' (orange curves) and 'feet' trials (green curves) from user 1. The projection direction is that which minimizes the L1-norm-based objective function, calculated by the algorithm in Section 5 when using as input all 'left-hand' and 'feet' trials of user 1. The difference between the respective variances is apparent. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Accuracy in the discrimination between pairs of imagined movements (L = left hand, R = right hand, F = feet, T = tongue). Results are shown for the nine users in the database (u1,..., u9). Last column gives the accuracy per user averaged over all possible pairs of movements. The last row is the average of all the previous rows.

User	L-R	L-F	L-T	R-F	R-T	F-T	avg
u1	0.66	0.89	0.91	0.93	0.92	0.52	0.84
u2	0.52	0.72	0.6	0.68	0.54	0.65	0.64
u3	0.87	0.68	0.69	0.86	0.86	0.54	0.73
u4	0.57	0.7	0.61	0.63	0.65	0.55	0.63
u5	0.53	0.55	0.6	0.55	0.57	0.54	0.56
u6	0.52	0.64	0.56	0.53	0.54	0.53	0.56
u7	0.59	0.73	0.74	0.89	0.89	0.69	0.79
u8	0.77	0.65	0.87	0.59	0.75	0.71	0.72
u9	0.78	0.86	0.88	0.55	0.7	0.76	0.75
avg	0.65	0.71	0.72	0.69	0.71	0.61	

ing the experiment [32,33]. The dataset 2a from the BCI competition IV comprises a number of trials (repetitions) of some simple limb (left hand, right hand, feet or tongue) motor-imagery movements [34–36]. In each recording session, $p = 22$-channel EEG signals are measured at a sample rate of 250 Hz from volunteers performing the desired imagery tasks. As usual in BCI signal processing, the EEG data are bandpass filtered to $8 - 30$ Hz. This preprocessing ensures that the data are zero-mean and, by central limit arguments, also allows us to support the hypothesis of Gaussianity for long filters.

Each imagined action lasts for about three seconds, but only the final two of them are kept in our experiment to avoid the initial transient effects. For illustration, one of these two-second intervals is shown in Fig. 7. We concatenate all trials of the same imagined movement into a single $22 \times 30\,000$ data matrix, and the algorithm in Section 5 is fed with pairs of matrices of distinct imagined movements. As an example, Fig. 8 depicts the density functions of the scalar projection of some data points, from trials of two distinct imaginary tasks, onto the direction that minimizes the L1 criterion: the differences in variance between the two classes are apparent even to the naked eye. Best results are obtained for data filtered in the band between 12 and 30 Hz (upper α and β bands), as well as pre-processed with the method in [37] to reduce the inherent nonstationarity of the EEG.

Next, for the set of data points of each trial, we retain the scalar projection in the direction of minimum L1-norm and the two projections which are most correlated with the first one, as well as the three projections with the lowest correlation, in a similar way as we have done before in Section 6.1.1. Table 4 shows how well a given imagined movement is classified simply by comparing the total variances of these two groups of three projections. As the trials are actually time-series, and not just a point in a p-dimensional space, comparing variances is easier to do and a feasible criterion. Note that the total variance in each projected subspace is measured by the trace of the covariance matrix of the projected data.

Accuracy is measured for the nine voluntaries in the database and considering all the possible combination of imagined tasks (L-R: left hand-right hand, L-F: left hand-feet, and so on). For example, a high degree (93%) of accuracy in discriminating between 'right hand' and 'feet' imagined movements has been obtained for user 1, achieved in a completely unsupervised fashion, but that accuracy reduces to 52% for the same user and the pair 'feet-tongue'. There is a great variability between users and pairs of movements, nevertheless, averaged over all users, we can discriminate between 'left-hand' and 'feet', 'left-hand' and 'tongue', and 'right-hand' and 'tongue' movements in more than 70% of the cases.

7. Conclusions

Projecting whitened data onto the few dimensions that minimize the absolute value of the projected data points can perform the FKT in a fully unsupervised fashion, sparing the need for training data. This connection between the L1-norm and the FKT had previously gone unnoticed, and endows L1-criteria with discriminative properties in binary classification scenarios, opening new lines of research in the area of L1-PCA. A working iterative algorithm based on gradient-descent in the Stiefel manifold is also put forward to perform L1-norm minimization with orthogonal constraints. Even though our theoretical analysis assumes the normality of the data, numerical experiments show that a good performance can be achieved when this assumption is not fulfilled. Further theoretical research should explore this extension to scenarios with non-Gaussian data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of eqn. (13)

The cost function is defined as follows:

$$D(\boldsymbol{w}) = \mathrm{E}\{|Y|\} = \int_{-\infty}^{\infty} |y|\, f(y)\, \mathrm{d}y = \int_{0}^{\infty} y\, f(y)\, \mathrm{d}y - \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y, \tag{A.1}$$

and invoking the zero mean assumption, i.e.,

$$\mathrm{E}\{Y\} = \int_{0}^{\infty} y\, f(y)\, \mathrm{d}y + \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y = 0 \Rightarrow \int_{-\infty}^{0} y\, f(y)\, \mathrm{d}y = -\int_{0}^{\infty} y\, f(y)\, \mathrm{d}y,$$

we readily get

$$D(\boldsymbol{w}) = \mathrm{E}\{|Y|\} = 2\int_{0}^{\infty} y\, f(y)\, \mathrm{d}y.$$

Under the Gaussian assumption (6), i.e.,

$$f(y) = \sum_{k=1,2} \frac{P(\mathcal{C}_k)}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{y^2}{2\sigma_k^2}\right),$$

where

$$\sigma_k^2 = \boldsymbol{w}^\dagger \boldsymbol{\Sigma}_k \boldsymbol{w}, \tag{A.2}$$

the cost function can be worked out as:

$$D(\boldsymbol{w}) = 2\sum_{k=1}^{2} P(\mathcal{C}_k) \int_{0}^{\infty} \frac{y}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{y^2}{2\sigma_k^2}\right) \mathrm{d}y = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} P(\mathcal{C}_k)\, \sigma_k, \tag{A.3}$$

where the second equality is readily obtained by using the identity [38]:

$$\int y\, e^{-\frac{y^2}{2\sigma^2}} \mathrm{d}y = -\sigma^2\, e^{-\frac{y^2}{2\sigma^2}} + \text{constant of integration}.$$

The stationary points of the constrained optimization problem (8) verify

$$\nabla_{\boldsymbol{w}} D(\boldsymbol{w}) = \ell\, \nabla_{\boldsymbol{w}} \|\boldsymbol{w}\|^2. \tag{A.4}$$

where ℓ is a Lagrange multiplier and $\nabla_{\boldsymbol{w}}$ stands for the gradient with respect to $\boldsymbol{w}$. We see that (A.3) is a function of σ_1 and σ_2. It is easier to calculate $\nabla_{\boldsymbol{w}}\sigma_k$ by noticing that

$$\nabla_{\boldsymbol{w}}\sigma_k^2 = 2\sigma_k \nabla_{\boldsymbol{w}}\sigma_k,$$

and, as follows from (A.2), that

$$\nabla_{\boldsymbol{w}}\sigma_k^2 = \nabla_{\boldsymbol{w}}\left(\boldsymbol{w}^\dagger \boldsymbol{\Sigma}_k \boldsymbol{w}\right) = 2\,\boldsymbol{\Sigma}_k \boldsymbol{w}.$$

Combining both formulas, we readily get $\nabla_{\boldsymbol{w}}\sigma_k = \nabla_{\boldsymbol{w}}\sigma_k^2/(2\sigma_k) = \boldsymbol{\Sigma}_k \boldsymbol{w}/\sigma_k$. Replacing this result in the calculation of the gradient of (A.3), it follows that

$$\nabla_{\boldsymbol{w}} D(\boldsymbol{w}) = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k}\, \boldsymbol{\Sigma}_k \boldsymbol{w}. \tag{A.5}$$

Similarly,

$$\nabla_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 = \nabla_{\boldsymbol{w}}\left(\boldsymbol{w}^\dagger \boldsymbol{w}\right) = 2\boldsymbol{w}. \tag{A.6}$$

Therefore, (A.4) becomes

$$\sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k}\, \boldsymbol{\Sigma}_k \boldsymbol{w} = 2\,\ell\, \boldsymbol{w}. \tag{A.7}$$

The value of ℓ can be obtained by pre-multiplying (A.7) by $\boldsymbol{w}^\dagger$, after which we use (A.2) as well as $\boldsymbol{w}^\dagger \boldsymbol{w} = 1$. By so doing, we finally get:

$$\ell = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{2} P(\mathcal{C}_k)\sigma_k. \tag{A.8}$$

Appendix B. Proof of Theorem 1

Let us study whether the eigenvectors of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ correspond to maxima, minima or saddle points of the L1-norm objective function. We start by calculating the Hessian (matrix of second order partial derivatives) of $\mathrm{E}\{|Y|\}$. From (A.5), and after some algebra, this Hessian is easily found to be:

$$\Delta_{\boldsymbol{w}}^2 \mathrm{E}\{|Y|\} = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k} \left[\boldsymbol{\Sigma}_k - \frac{1}{\sigma_k^2} \boldsymbol{\Sigma}_k \boldsymbol{w}\, (\boldsymbol{\Sigma}_k \boldsymbol{w})^\dagger\right] = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} \frac{P(\mathcal{C}_k)}{\sigma_k} \left[\boldsymbol{\Sigma}_k - \sigma_k^2 \boldsymbol{w}\boldsymbol{w}^\dagger\right]. \tag{B.1}$$

where the second equality follows from $\boldsymbol{\Sigma}_k \boldsymbol{w} = \sigma_k^2 \boldsymbol{w}$. Similarly, the Hessian matrix of the constraint $\|\boldsymbol{w}\|^2 = 1$ reads

$$\Delta_{\boldsymbol{w}}^2 \|\boldsymbol{w}\|^2 = 2\boldsymbol{I}. \tag{B.2}$$

Finally, the Hessian of the Lagrangian equals

$$\Delta_{\boldsymbol{w}}^2 L = \Delta_{\boldsymbol{w}}^2 \mathrm{E}\{|Y|\} - \ell\, \Delta_{\boldsymbol{w}}^2 \|\boldsymbol{w}\|^2, \tag{B.3}$$

where ℓ is the Lagrange multiplier. Then, note the following result in [39, Chap. 20], which we rewrite here in our own notation:

Theorem 2. *Let $\boldsymbol{w}$ be a critical point (maximizer, minimizer or saddle point) of $\mathrm{E}\{|Y|\}$ subject to $\|\boldsymbol{w}\|^2 = 1$. If, for all unit-length vector $\boldsymbol{z}$ such that*

$$\boldsymbol{z}^\dagger \nabla_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 = 2\boldsymbol{z}^\dagger \boldsymbol{w} = 0, \tag{B.4}$$

it holds that

$$\boldsymbol{z}^\dagger (\Delta_{\boldsymbol{w}}^2 L)\boldsymbol{z} > 0, \tag{B.5}$$

then $\boldsymbol{w}$ is a local minimizer. For a local maximizer, the above condition becomes $\boldsymbol{z}^\dagger (\Delta_{\boldsymbol{w}}^2 L)\boldsymbol{z} < 0$.

By using (A.8), we get

$$\boldsymbol{z}^\dagger \Delta_{\boldsymbol{w}}^2 L\, \boldsymbol{z} = \sqrt{\frac{2}{\pi}} \left(\sum_{k=1}^{2} \frac{s_k^2 - \sigma_k^2}{\sigma_k} P(\mathcal{C}_k) \right), \tag{B.6}$$

where $\sigma_k^2 = \boldsymbol{w}^\dagger \boldsymbol{\Sigma}_k \boldsymbol{w}$ and $s_k^2 = \boldsymbol{z}^\dagger \boldsymbol{\Sigma}_k \boldsymbol{z}$. Let us analyze the term:

$$\frac{s_2^2 - \sigma_2^2}{\sigma_2} P(\mathcal{C}_2). \tag{B.7}$$

On the one hand, the whitening condition (2) allows us to write

$$1 = \boldsymbol{z}^\dagger \boldsymbol{\Sigma} \boldsymbol{z} = \sum_{k=1}^{2} P(\mathcal{C}_k)\, s_k^2 \;\Rightarrow\; s_2^2 P(\mathcal{C}_2) = 1 - P(\mathcal{C}_1)\, s_1^2 \tag{B.8}$$

and, by the same token,

$$\sigma_2^2 P(\mathcal{C}_2) = 1 - P(\mathcal{C}_1)\, \sigma_1^2. \tag{B.9}$$

Invoking these results, we get

$$\frac{s_2^2 - \sigma_2^2}{\sigma_2} P(\mathcal{C}_2) = -\frac{(s_1^2 - \sigma_1^2)}{\sigma_2} P(\mathcal{C}_1). \tag{B.10}$$

Hence, substituting in (B.6), it follows that

$$\boldsymbol{z}^\dagger \Delta_{\boldsymbol{w}}^2 L \boldsymbol{z} = \sqrt{\frac{2}{\pi}} (s_1^2 - \sigma_1^2) \Big(\frac{1}{\sigma_1} - \frac{1}{\sigma_2} \Big) P(\mathcal{C}_1). \tag{B.11}$$

Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$ be the eigenvectors of $\boldsymbol{\Sigma}_1$, with $\lambda_1 > \lambda_2 > \ldots > \lambda_p$ the corresponding eigenvalues. Hence, let us consider several cases:

1. If $\boldsymbol{w} = \boldsymbol{v}_1$ is the dominant eigenvector of $\boldsymbol{\Sigma}_1$, then, from the properties of the Rayleigh quotient [40],

   $$\boldsymbol{w} = \text{argmax}_{\boldsymbol{z}} \boldsymbol{z}^\dagger \boldsymbol{\Sigma}_1 \boldsymbol{z}, \tag{B.12}$$

   and, therefore, $\sigma_1 > s_1$ and $\sigma_1 > \sigma_2$. It follows that

   $$\boldsymbol{z}^\dagger \Delta_{\boldsymbol{w}}^2 L \boldsymbol{z} > 0, \tag{B.13}$$

   and therefore $\boldsymbol{w}$ is a minimum of the L1 norm cost function.
2. Similarly, if $\boldsymbol{w} = \boldsymbol{v}_p$ is the least dominant eigenvector of $\boldsymbol{\Sigma}_1$ (the eigenvector associated with the smallest eigenvalue), then, from the Rayleigh quotient again [40], $\sigma_1 < s_1$ and $\sigma_1 < \sigma_2$. It follows that $\boldsymbol{z}^\dagger \Delta_{\boldsymbol{w}}^2 L \boldsymbol{z} > 0$ and $\boldsymbol{w}_p$ is still a minimum.
3. If $\boldsymbol{w} = \boldsymbol{v}_i$, $1 < i < p$, is any of the remaining eigenvectors, the sign of (B.11) when $\boldsymbol{z} = \boldsymbol{v}_1$ is different from that when $\boldsymbol{z} = \boldsymbol{v}_p$, and both $\boldsymbol{v}_1$ and $\boldsymbol{v}_p$ fulfill (B.4) because the eigenvectors are mutually orthogonal. That is, in the vicinity of $\boldsymbol{w} = \boldsymbol{v}_i$ the objective function increases in one direction and decreases in another. Therefore, $\boldsymbol{w}$ is a saddle point. Having said that, **if $\boldsymbol{w}$ is constrained to be orthogonal to $\boldsymbol{v}_1$ and $\boldsymbol{v}_p$, then it can be easily shown that $\boldsymbol{v}_2$ and $\boldsymbol{v}_{p-1}$ are the new minima of the L1-cost and so on**. Hence, all the eigenvectors can be actually calculated by minimization techniques, constrained to be orthogonal to the previously calculated ones.

Finally note that Eq. (15) also has the solution $\sigma_1 = \sigma_2$. Let us briefly show that this solution corresponds to the absolute maximum of the L1-objective function. Let $\mathrm{b} = (\sigma_1, \sigma_2)^\dagger$, $\mathbf{1} = (1,1)^\dagger$ and $\boldsymbol{D} = \text{diag}(P(\mathcal{C}_1), P(\mathcal{C}_2))$. Define the weighted inner product $(\mathrm{b}, \mathbf{1})_{\boldsymbol{D}} = \mathrm{b}^\dagger \boldsymbol{D} \mathbf{1}$. Then, by the Cauchy-Schwarz inequality,

$$(\mathrm{b}, \mathbf{1})_{\boldsymbol{D}} \le \sqrt{(\mathrm{b}, \mathrm{b})_{\boldsymbol{D}}\, (\mathbf{1}, \mathbf{1})_{\boldsymbol{D}}} = \sqrt{\sum_{i=1,2} P(\mathcal{C}_i)\, \sigma_i^2} = 1.$$

where the final equality follows from (2). It is then easy to show the following inequality that restricts (A.3), i.e.,

$$\mathrm{E}\{|Y|\} = \sqrt{\frac{2}{\pi}} \sum_{k=1}^{2} P(\mathcal{C}_k)\, \sigma_k = \sqrt{\frac{2}{\pi}}\, (\mathrm{b}, \mathbf{1})_{\boldsymbol{D}} \le \sqrt{\frac{2}{\pi}},$$

with equality iff b is proportional to $\mathbf{1}$, implying $\sigma_1 = \sigma_2$. This completes the proof.

Appendix C. Upper bounds on approximation (18)

To discuss the approximation (18), where $\eta \in \mathbb{R}^+$ is a small positive constant, let us find an upper bound on the remainder. The exponential matrix is defined in classic textbooks as follows (e.g. see [40])

$$\boldsymbol{U} = \exp(-\eta \boldsymbol{S}) = \boldsymbol{I} + \sum_{k=1}^{\infty} \frac{(-\eta \boldsymbol{S})^k}{k!}$$

where $\boldsymbol{I}$ is the identity matrix and subscript n is omitted for simplicity. Consider approximating the exponential by $\hat{\boldsymbol{U}} = \boldsymbol{I} - \eta \boldsymbol{S}$. The approximation error is computed as

$$\boldsymbol{R} = \boldsymbol{U} - \hat{\boldsymbol{U}} = \sum_{k=2}^{\infty} \frac{(-\eta \boldsymbol{S})^k}{k!}.$$

By the triangle inequality property of matrix norms:

$$\begin{aligned} \|\mathbf{R}\| &\le \textstyle\sum_{k=2}^{\infty} \eta^k \frac{\|\mathbf{S}\|^k}{k!} = \eta^2 \|\mathbf{S}\|^2 \sum_{k=0}^{\infty} \frac{\varepsilon^k}{(k+2)!} \\ &< \eta^2 \|\mathbf{S}\|^2 \textstyle\sum_{k=0}^{\infty} \frac{\varepsilon^k}{k!} = \eta^2 \|\mathbf{S}\|^2 \exp(\varepsilon) \end{aligned}$$

where $\varepsilon = \eta \|\boldsymbol{S}\|$. Consequently, the approximation error norm is upper bounded by

$$\|\mathbf{R}\| \le \eta^2 \|\mathbf{S}\|^2 \exp(\eta \|\mathbf{S}\|).$$

which is dominated by a term of the order of $\eta^2 \|\boldsymbol{S}\|^2$ for small $\eta < \frac{1}{\|\boldsymbol{S}\|}$.

Appendix D. The gradient in the Stiefel manifold

For the reader's interest, let us summarize briefly the most remarkable properties of orthogonality constraints. Consider the set of all n-tuples of orthonormal vectors in $\mathbb{R}^p$. This set is known as the Stiefel manifold and is denoted by $V_{p,n}$ [26]. Alternatively, as an n-tuple $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n)$ of vectors in $\mathbb{R}^p$ can be regarded as a matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n] \in \mathbb{R}^{p \times n}$, the manifold can be also expressed as $V_{p,n} = \{\boldsymbol{W} \in \mathbb{R}^{p \times n} : \boldsymbol{W}^\dagger \boldsymbol{W} = \boldsymbol{I}_n\}$, where $\boldsymbol{I}_n$ is the n-dimensional identity matrix.

Let $\boldsymbol{Q}(t)$ be a differentiable curve on $V_{p,n}$ with $\boldsymbol{Q}(0) = \boldsymbol{W} \in V_{p,n}$. The derivative $\dot{\boldsymbol{Q}}(0)$ can be regarded as the 'tangent vector' at $\boldsymbol{W}$ to the curve. The use of the term 'tangent' is justified because, intuitively, $\dot{\boldsymbol{Q}}(0)$ has the same direction as an infinitesimal displacement $d\boldsymbol{Q}(0)$ along the manifold. The tangent vectors calculated in this way, from each possible curve passing through $\boldsymbol{W}$, form a vector space called the *tangent space* at $\boldsymbol{W}$.

As $\boldsymbol{Q}(t)^\dagger \boldsymbol{Q}(t) = \boldsymbol{I}_p$ for all t, we readily find, after differentiating, that $\dot{\boldsymbol{Q}}(0)^\dagger \boldsymbol{Q}(0) + \boldsymbol{Q}(0)^\dagger \dot{\boldsymbol{Q}}(0) = \boldsymbol{0}$. From here, it follows that the tangent space at $\boldsymbol{W}$ is the set of matrices defined by

$$\mathcal{T}_{\boldsymbol{W}} V_{p,n} = \{\boldsymbol{S} \in \mathbb{R}^{p \times n} : \boldsymbol{S}^\dagger \boldsymbol{W} + \boldsymbol{W}^\dagger \boldsymbol{S} = \boldsymbol{0}\}.$$

Similarly, given any $p \times n$ matrix $\boldsymbol{Z}$, it can be also shown (see e.g. [26]) that

$$\pi_{\mathcal{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = \big(\boldsymbol{I}_p - \boldsymbol{W}\boldsymbol{W}^\dagger\big)\boldsymbol{Z} + \frac{1}{2}\boldsymbol{W}\big(\boldsymbol{W}^\dagger \boldsymbol{Z} - \boldsymbol{Z}^\dagger \boldsymbol{W}\big) \tag{D.1}$$

is the projection of $\boldsymbol{Z}$ onto $\mathcal{T}_{\boldsymbol{W}} V_{p,n}$. Now, consider the problem

$$\min_{\boldsymbol{W} \in \mathbb{R}^{p \times n}} J(\boldsymbol{W}) \text{ s.t. } \boldsymbol{W}^\dagger \boldsymbol{W} = \boldsymbol{I}_n.$$

Given the derivative $\partial J(\boldsymbol{W})$ of $J(\boldsymbol{W})$ at $\boldsymbol{W}$ in the Euclidean space, calculated element-wise, i.e., $(\partial J(\boldsymbol{W}))_{ij} = \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}_{ij}}$, the gradient of $J(\boldsymbol{W})$ on the Stiefel manifold is obtained as the projection $\pi_{\mathcal{T}_{\boldsymbol{W}}}(\partial J(\boldsymbol{W}))$ given by Eq. (D.1). In the particular case of square matrices, $p = n$, $\boldsymbol{W}^\dagger \boldsymbol{W} = \boldsymbol{W}\boldsymbol{W}^\dagger = \boldsymbol{I}_p$ and

$$\pi_{\mathcal{T}_{\boldsymbol{W}}}(\partial J(\boldsymbol{W})) = \frac{1}{2}\big(\partial J(\boldsymbol{W}) - \boldsymbol{W}\partial J(\boldsymbol{W})^\dagger \boldsymbol{W}\big).$$

Observe finally that this formula is the same (up to the 1/2 constant) to the gradient that appears in Eq. (20). Therefore, the algorithm proposed in Section 5.2 computes the gradient-descent minimization of criterion J in the Stiefel manifold.

CRediT authorship contribution statement

José Luis Camargo: Conceptualization, Methodology, Software, Validation, Formal analysis. **Rubén Martín-Clemente:** Conceptualization, Methodology. **Susana Hornillo-Mellado:** Software, Validation, Formal analysis. **Vicente Zarzoso:** Conceptualization, Writing - review & editing.

References

[1] K. Fukunaga, W. Koontz, Application of the Karhunen-Loève expansion to feature selection and ordering, IEEE Trans. Comput. C-19 (4) (1970) 311–318.
[2] X. Huo, M. Elad, A.G. Flesia, R.R. Muise, S.R. Stanfill, J. Friedman, B. Popescu, J. Chen, A. Mahalanobis, D.L. Donoho, Optimal reduced-rank quadratic classifiers using the Fukunaga-Koontz transform with applications to automated target recognition, in: F.A. Sadjadi (Ed.), Automatic Target Recognition XIII, SPIE, 2003, pp. 59–72.
[3] X. Huo, A statistical analysis of Fkunaga–Koontz transform, IEEE Signal Process Lett 11 (2) (2004) 123–126.
[4] J. Peng, G. Seetharaman, W. Fan, S. Robila, A. Varde, Chernoff dimensionality reduction–where Fisher meets FKT, in: Proceedings of the 2011 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2011, pp. 271–282.
[5] R. Duin, M. Loog, Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion, IEEE Trans Pattern Anal Mach Intell 26 (6) (2004) 732–739.
[6] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, 2005.
[7] A. Miranda, P. Whelan, Fukunaga-Koontz transform for small sample size problems, in: Proceedings of the IEE Irish Signals and Systems Conference (ISSC, Dublin, Ireland, 2005), 2005, pp. 1–6.
[8] S. Zhang, T. Sim, Discriminant subspace analysis: a Fukunaga-Koontz approach, IEEE Trans Pattern Anal Mach Intell 29 (10) (2007) 1732–1745.
[9] A. Bal, M.S. Alam, Automatic target tracking in forward-looking infrared video sequences using tuned basis functions, Opt. Eng. 55 (7) (2016) 073102.
[10] H. Binol, Improved Fukunaga–Koontz transform with compositional kernel combination for hyperspectral target detection, J. Indian Soc. Remote Sens. 46 (10) (2018) 1605–1615.
[11] F. Juefei-Xu, M. Savvides, Multi-class Fukunaga Koontz discriminant analysis for enhanced face recognition, Pattern Recognit 52 (2016) 186–205.
[12] R. Liu, E. Liu, J. Yang, Y. Zeng, F. Wang, Y. Cao, Automatically detect and track infrared small targets with kernel Fukunaga-Koontz transform and kalman prediction, Appl Opt 46 (31) (2007) 7780.
[13] S. Ochilov, M.S. Alam, A. Bal, Fukunaga-Koontz transform based dimensionality reduction for hyperspectral imagery, in: S.S. Shen, P.E. Lewis (Eds.), Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII, SPIE, 2006, pp. 1–8.
[14] Z.J. Koles, M.S. Lazar, S.Z. Zhou, Spatial patterns underlying population differences in the background EEG, Brain Topogr 2 (4) (1990) 275–284.
[15] W. Wu, Z. Chen, X. Gao, Y. Li, E.N. Brown, S. Gao, Probabilistic common spatial patterns for multichannel EEG analysis, IEEE Trans Pattern Anal Mach Intell 37 (3) (2015) 639–653.
[16] H. Binol, G. Bilgin, S. Dinc, A. Bal, Kernel Fukunaga–Koontz transform subspaces for classification of hyperspectral images with small sample sizes, IEEE Geosci. Remote Sens. Lett. 12 (6) (2015) 1287–1291.
[17] S. Hoell, P. Omenzetter, Fukunaga-Koontz feature transformation for statistical structural damage detection and hierarchical neuro-fuzzy damage localisation, J Sound Vib 400 (2017) 329–353.
[18] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans Pattern Anal Mach Intell 30 (9) (2008) 1672–1680.
[19] P.P. Markopoulos, G.N. Karystinos, D.A. Pados, Optimal algorithms for L1-subspace signal processing, IEEE Trans. Signal Process. 62 (19) (2014) 5046–5058.
[20] P.P. Markopoulos, S. Kundu, S. Chamadia, N. Tsagkarakis, D.A. Pados, Outlier-resistant data processing with L1-norm principal component analysis, in: Advances in Principal Component Analysis, Springer Singapore, 2017, pp. 121–135.
[21] R. Martín-Clemente, V. Zarzoso, On the link between L1-PCA and ICA, IEEE Trans Pattern Anal Mach Intell 39 (3) (2017) 515–528.
[22] R. Martín-Clemente, V. Zarzoso, LDA via L1-PCA of whitened data, IEEE Trans. Signal Process. 68 (2020) 225–240.
[23] I.T. Jolliffe, Principal component analysis, Springer, New York, NY, 2002.
[24] A. Kessy, A. Lewin, K. Strimmer, Optimal whitening and decorrelation, Am Stat 72 (4) (2018) 309–314.
[25] P.P. Markopoulos, S. Kundu, S. Chamadia, D.A. Pados, Efficient l1-norm principal-component analysis via bit flipping, IEEE Trans. Signal Process. 65 (16) (2017) 4252–4264.
[26] A. Edelman, T.A. Arias, S.T. Smith, The geometry of algorithms with orthogonality constraints, SIAM J. Matrix Anal. Appl. 20 (2) (1998) 303–353.
[27] J.-B. Hiriart-Urruty, C. Lemaréchal, Fundamentals of Convex Analysis, Springer Berlin Heidelberg, 2001.
[28] A.I. Fleishman, A method for simulating non-normal distributions, Psychometrika 43 (4) (1978) 521–532.
[29] C.D. Vale, V.A. Maurelli, Simulating multivariate nonnormal distributions, Psychometrika 48 (3) (1983) 465–471.
[30] J. Mathews, Numerical Methods Using MATLAB, Prentice Hall, Upper Saddle River, N.J, 1999.
[31] P.J. Rousseeuw, K.V. Driessen, A fast algorithm for the minimum covariance determinant estimator, Technometrics 41 (3) (1999) 212–223.
[32] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update, J Neural Eng 15 (3) (2018) 031005.
[33] R. Martín-Clemente, J. Olias, D. Thiyam, A. Cichocki, S. Cruces, Information theoretic approaches for motor-imagery BCI systems: review and experimental comparison, Entropy 20 (1) (2018) 7.
[34] B. Blankertz, C. Vidaurre, M. Tangermann, K.-R. Müller, C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, G. Pfurtscheller, S. Waldert, C. Mehring, A. Aertsen, G.S. Niels Birbaumer K. J. Miller BCI Competition IV dataset, 2008, (http://www.bbci.de/competition/iv/). accessed April 2020.
[35] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, G. Pfurtscheller, Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment, IEEE Trans. Neural Syst. Rehabil. Eng. 15 (4) (2007) 473–482.
[36] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K.J. Miller, G.R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, B. Blankertz, Review of the BCI competition IV, Front Neurosci 6 (2012).
[37] J. Olias, R. Martín-Clemente, M.A. Sarmiento-Vega, S. Cruces, EEG Signal processing in MI-BCI applications with improved covariance matrix estimators, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (5) (2019) 895–904.
[38] I.S. Gradshteyn, I. Ryzhik, Table of Integrals, Series and Products, Academic, Oxford, 2007.
[39] E.K.P. Chong, S.H. Zak, An introduction to optimization, John Wiley & Sons, 2013.
[40] G. Golub, C. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1996.

B Artículo 2

José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, Vicente Zarzoso, «Unsupervised Classification of Zero-Mean Data Based on L1-Norm Principal Component Analysis», en: H. Sharma, M. Gupta, G. Tomar, W. Lipo (eds) «Communication and Intelligent Systems», *Lecture Notes in Networks and Systems*, vol 204, pp. 967–973, 2021, Springer, Singapur. *https://doi.org/10.1007/978-981-16-1089-9_75*.

Unsupervised classification of zero-mean data based on L1-norm Principal Component Analysis

José Luis Camargo[1], Rubén Martín-Clemente[1⋆], Susana Hornillo-Mellado[1], and Vicente Zarzoso[2]

[1] Signal Processing and Communications Department, University of Seville, Spain, jlcamargo@yahoo.es, ruben@us.es, susanah@us.es
[2] Université Cote d'Azur, CNRS, I3S Laboratory, Sophia Antipolis Cedex, France vicente.zarzoso@univ-cotedazur.fr

Abstract. This paper shows that L1-norm PCA, a robust variant of Principal Component Analysis (PCA), can distinguish between zero-mean populations by projecting the data onto directions along which the variance is much larger for one population than for the other. Thus, the variance can be used as a criterion for classification.

Keywords: principal component analysis, machine learning

1 Introduction

Principal Component Analysis (PCA) is one the most widely-used techniques for analyzing multivariate data. It finds application in fields such as image processing, wireless communications, machine learning, biomedical or signal analysis, to name only a few [1]. PCA searches for a *low*-dimensional subspace that minimizes the average squared distance of the data points to it, i.e., the best-fit subspace for the data points. By projecting the data onto this subspace, we also reduce the dimensionality of the data. Therefore, PCA is useful for compression and in pattern recognition and denoising problems. Furthermore, it is well-known that the best-fit subspace maximizes the variance of the scalar projections of the data points on it. Therefore, PCA also captures the directions of maximum variability of the data [1].

After simple algebra, it can be shown that the best-fit subspace is spanned by the first few dominant eigenvectors of the data covariance matrix [1]. From a computational viewpoint, this subspace can be estimated from the singular value decomposition (SVD) of the data matrix. Unfortunately, the SVD is extremely sensitive to the presence of large outliers in the data. This is a serious drawback, since outliers are usually encountered in experiments due to the imperfections in the measuring instruments.

⋆ The research of Drs. Martín and Zarzoso was partially funded by the project ACACIA (ref US-1264994) awarded by the Junta de Andalucía (Consejería de Transformación Económica, Industria, Conocimiento y Universidades) and I+D+i FEDER Andalucía 2014-2020.

To overcome this problem, several authors have proposed robust variants of standard PCA in recent years. The most promising option is to replace the variance with the *median absolute deviation statistic* for measuring the spread of the data. This approach leads to the technique known as L1-norm based PCA (L1-PCA) [2, 3]. Specifically, [2] presented this method with a computationally simple algorithm, whereas [3] identifies the equivalence between L1-norm maximization and 'binary quadratic programming'. The relationship between L1-PCA and Independent Component Analysis (ICA) and Linear Discriminant Analysis have been also discussed in [4, 5].

After reviewing L1-PCA in Section 2, the present contribution investigates the discriminative properties of L1-norm PCA in Section 3. Numerical experiments in Section 4 validate demonstrate the ability of this technique to carry out classification in an unsupervised fashion.

2 L1-norm Principal Component Analysis

Let $\boldsymbol{x} \in \mathbb{R}^p$ be a multivariate random variable measured or observed during the execution of an experiment. For simplicity, we suppose that $\boldsymbol{x}$ is of zero-mean, i.e., $\mathrm{E}[\boldsymbol{x}] = \boldsymbol{0}$, where $\mathrm{E}[\cdot]$ is the expectation operator.

PCA uses the low dimensional subspace defined by the most significant directions of spread of $\boldsymbol{x}$. Let $\boldsymbol{a} \in \mathbb{R}^p$ be the unit-norm vector in the direction of the line that best fits $\boldsymbol{x}$ in the least squares sense. It can be easily shown that $\boldsymbol{a}$ can be obtained as the solution to the variance-maximization problem [1]

$$\max_{\|\boldsymbol{a}\|_2=1} \mathrm{E}\left[\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2\right] = \max_{\|\boldsymbol{a}\|_2=1} \boldsymbol{a}^\top \boldsymbol{C} \boldsymbol{a} \tag{1}$$

where $\boldsymbol{C} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is the data covariance matrix. According to the Rayleigh quotient principle, the maximizer of (1) is the eigenvector associated to the largest eigenvalue of matrix $\boldsymbol{C}$. To determine the whole Q-dimensional best-fitting subspace, (1) is maximized Q times, under the constraint that the direction obtained in the qth optimization is orthogonal to the previously computed directions. As a result, one obtains the subspace spanned by the Q dominant eigenvectors of $\boldsymbol{C}$, which can be efficiently computed by the SVD of the data matrix.

A major drawback of PCA is that the square in (1) overemphasizes the importance of large data, typically outliers, causing $\boldsymbol{a}$ to be aligned with the most significant of them. In order to palliate such a serious weakness, [2] proposed the replacement of the square function by the absolute value, yielding the following alternative criterion:

$$\max_{\|\boldsymbol{a}\|_2=1} \quad \mathrm{E}[|\boldsymbol{a}^\top \boldsymbol{x}|] \tag{2}$$

Given a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the random variable $\boldsymbol{x}$, (2) is approximated in practice by its sample based estimate

$$\max_{\|\boldsymbol{a}\|_2=1} \quad \frac{1}{N} \sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i| \tag{3}$$

Note that $\sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i|$ is the L1-norm of the vector $\boldsymbol{y}$ whose kth entry is given by $y_k = \boldsymbol{a}^\top \boldsymbol{x}_i$. For this reason, PCA based on criterion (2) is usually referred to as 'L1-norm based PCA' or, simply, 'L1-PCA'. To gain some insight into the performance of L1-PCA, let us make the common assumption in data analysis that $f(\boldsymbol{x})$ is a p-variate normal density function of the form

$$f(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{C})^{-\frac{1}{2}} \, e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}^{-1}\boldsymbol{x}}. \tag{4}$$

Let $y = \boldsymbol{a}^\top \boldsymbol{x}$ be the projection of $\boldsymbol{x}$ into the direction defined by $\boldsymbol{a} \in \mathbb{R}^p$. From basic statistics, the probability density function of y is Gaussian as well:

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \tag{5}$$

where $\sigma^2 = \boldsymbol{a}^\top \boldsymbol{C}\boldsymbol{a}$ is the variance of the projected data. Then, by using that

$$\int y \, e^{-\frac{y^2}{2\sigma^2}} \, \mathrm{d}y = -\sigma^2 \, e^{-\frac{y^2}{2\sigma^2}}$$

some algebra leads to the result

$$\mathrm{E}[|y|] = \int_{-\infty}^{\infty} |y| \, f(y) \, \mathrm{d}y = \sqrt{\frac{2}{\pi}} \, \sigma$$

Therefore, because maximizing the standard deviation σ is equivalent to maximizing the variance σ^2, L1-PCA behaves like traditional PCA while offering robustness against the presence of large outliers in the data [2, 3, 6]. Practical algorithms for maximizing (2) have been proposed in [2, 3, 7].

Now, let us show in the next Section that L1-PCA is also endowed with discriminative properties.

3 Main contribution: L1-PCA for classification of zero-mean populations

In binary classification problems, we observe random samples from two distinct zero-mean classes ω_1 and ω_2. The goal is to find a rule to allocate the random samples into one class or the other. By assumption, ω_1 and ω_2 have respective *a priori* probabilities of occurrence π_1 and π_2, as well as covariance matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. It is also supposed that the distribution of the random samples is as a mixture of Gaussians, i.e.,

$$f(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \sum_{i=1}^{2} \pi_i \det(\boldsymbol{C}_i)^{-\frac{1}{2}} \, e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}_i^{-1}\boldsymbol{x}}.$$

By proceeding as in above, we easily get that

$$\mathrm{E}[|\boldsymbol{a}^\top \boldsymbol{x}|] = \mathrm{E}[|y|] = \sqrt{\frac{2}{\pi}} \, (\pi_1 \, \sigma_1 + \pi_2 \, \sigma_2) \tag{6}$$

where $\sigma_i^2 = \boldsymbol{a}^\top \boldsymbol{C}_i \boldsymbol{a}$ is the variance of the ith class in the direction of the unit vector $\boldsymbol{a} \in \mathbb{R}^p$. Unfortunately, no direct conclusions can be drawn from (6) if σ_1 and σ_2 vary independently of each other. To get a more meaningful criterion, let us link the class variances in such a way that when one increases the other decreases, and *vice versa*. This is always possible, without any loss of generality, by a data pre-processing step called *whitening* or *sphering*. *Whitening*, which removes the correlation between the data components, is achieved by linearly transforming the data by any square root of the inverse of the raw data covariance matrix [8]. Thanks to this pre-processing step, the covariance of the data becomes the identity

$$\boldsymbol{C} = E[\boldsymbol{x}\boldsymbol{x}^\top] = \pi_1 \boldsymbol{C}_1 + \pi_2 \boldsymbol{C}_2 = \boldsymbol{I} \tag{7}$$

Then, because $\boldsymbol{a}$ is of unit norm, it follows that

$$E[y^2] = \boldsymbol{a}^\top E[\boldsymbol{x}\boldsymbol{x}^\top]\boldsymbol{a} = \pi_1 \sigma_1^2 + \pi_2 \sigma_2^2 = 1 \tag{8}$$

so σ_1 and σ_2 are now linked as desired. The key result is that, as it can be formally proven:

Lemma 1. *The criterion*

$$D = \sqrt{\frac{2}{\pi}} \left(\pi_1 \, \sigma_1 + \pi_2 \, \sigma_2\right) \tag{9}$$

attains its minimum value, under the constraint $\pi_1 \sigma_1^2 + \pi_2 \sigma_2^2 = 1$*, when the variance of the projected data points is* maximum *for one class and* minimum *for the other.*

We omit the proof due to the lack of space. Nevertheless, some intuition can be gained by observing that the minima of $\alpha + \beta$, subject to $\alpha^2 + \beta^2 = 1$ and $\alpha, \beta \geq 0$, where α and β are any generic variables, are at the limit points of the interval, i.e., $\alpha = 0, \beta = 1$ and $\alpha = 1, \beta = 0$. Similarly, the maximum is attained at $\alpha = \beta$.

The Lemma can be put in connection with the useful technique known as common spatial patterns (CSP), which is widely used in brain-computer interfaces (BCIs). In BCIs, electroencephalogram (EEG) samples are acquired under two different conditions (e.g. imagining tongue and hand movements). CSP projects the data onto directions where the variance of the projected points is higher for one class than for the other [9]. The projected data variances are then used by the BCI as criteria for classification.

Consequently, we see that the L1 criterion clearly possesses discriminative properties, similar to those of CSP. The key here is that CSP is a supervised technique, whose performance relies heavily on the availability of correctly labeled data. On the contrary, minimizing the L1 criterion (6), i.e. E[|y|], *can be performed in a completely unsupervisedly fashion.*

4 Experiments

In brain computer interfaces (BCI's), one imagines a limb moving and the machine tries to detect the imagined movement by analyzing the EEG of the user [10]. The dataset 2a from the BCI competition IV consists of 22-channel EEG signals associated to left-hand, right-hand, feet and tongue motor-imagery movements [11]. As a pre-processing step, we filter the EEG data to $12 - 30$ Hz. By so doing, we ensure that the data are zero-mean and, by central limit arguments, we can also safely make the assumption of Gaussianity.

After applying a *whitening* pre-processing to the data, the gradient descent algorithm in [12] is applied to find orthogonal directions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{22}$ that minimize the criterion

$$\sum_{n=1}^{22} \mathrm{E}[\boldsymbol{a}_i^\top \boldsymbol{x}]$$

To this end, data samples are drawn from EEG signals from two distinct imagined movements. Table 1 shows the accuracy in the detection of the imagined movements of user 1, where classification is performed by comparing the variances the projected data (see Fig. 1). A high degree (93%) of accuracy in discriminating between 'right hand' and 'feet' imagined movements has been obtained, in a completely unsupervised fashion. However, that accuracy reduces to 52% for the same user and the pair 'feet-tongue'.

L-R	L-F	L-T	R-F	R-T	F-T	avg
0.66	0.89	0.91	0.93	0.92	0.52	0.84

Table 1. Accuracy in the discrimination between pairs of imagined movements (L = left hand, R = right hand, F = feet, T = tongue). Results are shown for user 1 of the database. Last column gives the accuracy per user averaged over all movements.

5 Conclusions

Unsupervised classification can be performed by projecting whitened data onto the few dimensions that minimize the absolute value of the projected data points, thus sparing the need for training data. Good performance is shown by numerical experiments. In future, we will explore the application of this technique to real data and carry out a more exhaustive comparison with the CSP approach.

References

1. Jolliffe, I.T.: Principal component analysis. Springer, New York, NY (2002)
2. Kwak, N.: Principal component analysis based on L1-norm maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(9), 1672–1680 (Sep 2008)

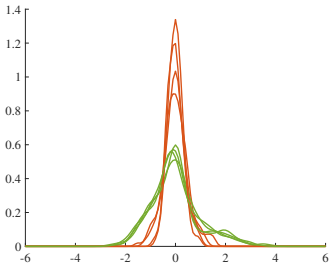


Fig. 1. Density functions of several projected 'left hand' (orange) and 'feet' trials (green) from user 1. The projection direction is that which minimizes the L1-norm-based objective function. The difference between the respective variances is apparent.

3. Markopoulos, P.P., Karystinos, G.N., Pados, D.A.: Optimal algorithms for L1-subspace signal processing. IEEE Transactions on Signal Processing 62(19), 5046–5058 (Oct 2014)
4. Martín-Clemente, R., Zarzoso, V.: On the link between L1-PCA and ICA. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(3), 515–528 (Mar 2017)
5. Martín-Clemente, R., Zarzoso, V.: LDA via L1-PCA of whitened data. IEEE Transactions on Signal Processing 68, 225–240 (Nov 2020)
6. Markopoulos, P.P., Kundu, S., Chamadia, S., Tsagkarakis, N., Pados, D.A.: Outlier-resistant data processing with L1-norm principal component analysis. In: Advances in Principal Component Analysis, pp. 121–135. Springer Singapore (Dec 2017)
7. Markopoulos, P.P., Kundu, S., Chamadia, S., Pados, D.A.: Efficient L1-norm principal-component analysis via bit flipping. IEEE Transactions on Signal Processing 65(16), 4252–4264 (Aug 2017)
8. Kessy, A., Lewin, A., Strimmer, K.: Optimal whitening and decorrelation. The American Statistician 72(4), 309–314 (Jan 2018)
9. Martín-Clemente, R., Olias, J., Thiyam, D., Cichocki, A., Cruces, S.: Information theoretic approaches for motor-imagery BCI systems: Review and experimental comparison. Entropy 20(1), 7 (Jan 2018)
10. Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., Yger, F.: A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. Journal of Neural Engineering 15(3), 031005 (Apr 2018)
11. [dataset] Benjamin Blankertz, Vidaurre, C., Tangermann, M., Müller, K.R., Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A., Pfurtscheller, G., Waldert, S., Mehring, C., Aertsen, A., Niels Birbaumer, Kai J. Miller, G.S.: BCI Competition IV dataset. http://www.bbci.de/competition/iv/ (2008), {http://www.bbci.de/competition/iv/}, last accessed April 2020
12. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications 20(2), 303–353 (Jan 1998)

C Artículo 3

Rubén Martín-Clemente, Vicente Zarzoso, José Luís Camargo, «On the discriminative properties of Principal Component Analysis based on L1-norm» *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dic. 2020, pp. 1673-1676, *https://doi.org/doi:10.1109/CSCI51800.2020.00308*.

On the discriminative properties of Principal Component Analysis based on L1-norm

Rubén Martín-Clemente
Dept. of Signal Processing and Communications
University of Seville)
Seville, Spain
ruben@us.es

Vicente Zarzoso
CNRS, I3S Laboratory
University of Côte d'Azur
Sophia-Antipolis, France
vicente.zarzoso@univ-cotedazur.fr

J. L. Camargo-Olivares
Dept. of Signal Proc. and Comms.
University of Seville)
Seville, Spain
jlcamargo@yahoo.es

***Abstract*—Principal Component Analysis (PCA) is one the most widely-used techniques for the analysis of multivariate data. Unfortunately, PCA is extremely sensitive to the presence of large outliers in the data. To overcome this drawback, a robust variant of standard PCA, based on the L1-norm, has been proposed in recent years. This variant, called L1-PCA behaves like traditional PCA, while offering robustness against the presence of large outliers in the data. This paper shows that, combined with a whitening pre-processing, L1-PCA is also endowed with discriminative properties, allowing it to solve binary classification problems in an unsupervised way, thus sparing the need for training data.**

***Index Terms*—L1-PCA, binary classification, linear discriminant analysis, common spatial patterns**

I. Introduction

The last decade has witnessed an increasing interest in the use of L1-norm based criteria and their applications in the field of machine learning. A particularly remarkable criterion is obtained by replacing the L2-norm of standard Principal Component Analysis (PCA) by the L1-norm, yielding the so-called L1-PCA method [1], [2]. This variant of standard PCA is more robust against outliers, which are erroneous measurements that lie far apart from the main bulk of the data, than the original approach. Additionally, L1-PCA is simple and intuitive compared to other robust PCA techniques, which is why its popularity has experienced a strong growth. To cite a few application examples, L1-PCA has proven to be highly effective for the restoration of faulty data, in robust face recognition, in video surveillance or in dimensionality reduction problems [1]–[4]. Starting from the seminal work in [1], a number of working algorithms for performing L1-PCA have been proposed in the recent literature, including methods with guaranteed convergence to the optimal solutions [3] or approaches that use simplified computation procedures [4].

In the case of Gaussian-distributed data, it can be shown that L1-PCA behaves like traditional PCA, while offering robustness against the presence of large outliers in the dataset [1]–[3]. Nevertheless, the performance of the technique under other working conditions is not sufficiently understood. In fact, only a few attempts have been made to explain the behavior of L1-PCA in alternative scenarios. For example, [5] showed that L1-PCA is also able to perform Independent Component Analysis (ICA) if the data follows the ICA model. Arguably, this lack of studies is due to the difficulties caused by the fact that L1-PCA is (implicitly) a higher-order-statistics-type approach. The present contribution investigates the discrimination ability of L1-PCA in classification problems. It is shown that L1-PCA, under mild assumptions, replicates the operation of some standard supervised classification methods. The most interesting feature is that L1-PCA can perform this task in an *unsupervised manner*, i.e., sparing the need for training data. This is interesting as it opens new research perspectives for L1-PCA in the field of machine learning. Additionally, it paves the way for the use of L1-PCA algorithms in classification problems while taking advantage of the improved robustness to outliers of the L1-norm criterion.

The paper is organized as follows: Section II introduces the L1-norm from standard PCA. Section III shows that the L1-norm is endowed with discriminative properties in binary classification scenarios. Section IV illustrates the performance of the approach through computer simulations. Finally, section V brings the paper to an end.

II. Background

Let $\boldsymbol{x} \in \mathbb{R}^p$ be a multivariate random variable measured or observed during an experiment, where each dimension represents some specific feature of the data. Without loss of generality, we can assume that $\mathrm{E}[\boldsymbol{x}] = \mathbf{0}$, where $\mathrm{E}[\cdot]$ is the expectation operator. The aim of standard PCA is to find linear projections of the variables that have maximal variance [6]. A projection onto the direction of a unit vector $\boldsymbol{a}$ is given by

$$y = \boldsymbol{a}^\top \boldsymbol{x}$$

The variance of the projected data equals

$$\sigma^2(\boldsymbol{a}) = \mathrm{E}\left[y^2\right] \tag{1}$$

The first principal component is the vector that solves the optimization problem [6]

$$\arg\max_{\|\boldsymbol{a}\|_2=1} \sigma^2(\boldsymbol{a}) \tag{2}$$

This work is funded by the research project ACACIA (ref US-1264994) awarded by the Junta de Andalucía (Consejería de Transformación Económica, Industria, Conocimiento y Universidades) and I+D+i FEDER Andalucía 2014-2020.

The $n-$th principal component is the vector that solves the optimization problem (2) subject to the additional constraint of being *orthogonal* to the previous $n-1$ principal components. It can be also shown that the span of the principal components is the best-fit *low*-dimensional subspace for the data points, that is, the subspace that minimizes the average squared distance of the data points to it. Consequently, it is often argued that the principal components are the most meaningful directions that characterize the point cloud of $\boldsymbol{x}$.

Unfortunately, PCA is very sensitive to outliers because these are overemphasized by the square in (1). In order to overcome this drawback, [1] proposed the replacement of the square function by the absolute value, thus giving rise to the following alternative criterion:

$$\arg\max_{\|\boldsymbol{a}\|_2=1} \mathrm{E}[|y|] \tag{3}$$

In practice, given a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the random variable $\boldsymbol{x}$, (3) is approximated by its sample based estimate

$$\max_{\|\boldsymbol{a}\|_2=1} \frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i| \tag{4}$$

Because $\sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i|$ represents the L1-norm of the vector $\boldsymbol{y}$ whose kth entry is given by $y_k = \boldsymbol{a}^\top \boldsymbol{x}_i$, PCA based on criterion (3) is usually referred to as 'L1-norm based PCA' or, simply, 'L1-PCA'. Observe also that standard PCA actually maximizes the L2-norm of vector $\boldsymbol{y}$, and for this reason we also refer to it as 'L2-PCA'.

III. Classification properties of L1-PCA

Gaussian mixture models are useful for modelling data which come from different populations. Consider that we observe random samples drawn from two different classes ω_1 and ω_2 with population means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. It is supposed that the distribution of the random samples can be written as

$$p(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \sum_{i=1}^{2} \pi_i \det(\boldsymbol{C}_i)^{-\frac{1}{2}}\, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}$$

where π_1 and π_2 are the *a priori* probabilities of occurrence of ω_1 and ω_2, with $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ the corresponding class covariance matrices.

It is also supposed, without any loss of generality, that the data are *whitened*. A random variable $\hat{\boldsymbol{x}}$ can be whitened by multiplying it by a matrix $\boldsymbol{Q}$ so that the result $\boldsymbol{x} = \boldsymbol{Q}\hat{\boldsymbol{x}}$ has covariance $\boldsymbol{C} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{Q}\hat{\boldsymbol{C}}\boldsymbol{Q}^\top = \boldsymbol{I}$, where $\hat{\boldsymbol{C}} = \mathrm{E}[\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top]$ and $\boldsymbol{I}$ is the identity matrix. A convenient choice is simply $\boldsymbol{Q} = \hat{\boldsymbol{C}}^{-1/2}$.

Whitening has two important consequences, which are easy to show:

- First of all, it follows that

$$\boldsymbol{C} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{C}_W + \boldsymbol{C}_B = \boldsymbol{I} \tag{5}$$

where

$$\boldsymbol{C}_W = \sum_{i=1}^{2} \pi_i \boldsymbol{C}_i \tag{6}$$

$$\boldsymbol{C}_B = \pi_1\pi_2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \tag{7}$$

are, respectively, the so-called within-class and between-class covariance matrices.

- Secondly, let $y = \boldsymbol{a}^\top \boldsymbol{x}$ be the projection of the whitened variable $\boldsymbol{x}$ into the direction defined by the unit-norm vector $\boldsymbol{a} \in \mathbb{R}^p$. Whitening also implies that

$$\mathrm{E}[y^2] = \boldsymbol{a}^\top \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top]\boldsymbol{a} = \boldsymbol{a}^\top \boldsymbol{a} = 1 \tag{8}$$

As the variance cannot change, observe that the standard PCA criterion (1) becomes useless.

On the downside, whitening requires the calculation of second-order statistics and hence we will loss protection against outliers unless we use robust whitening algorithms. The probability density function of y is given by

$$p(y) = \sum_{i=1}^{2} \frac{\pi_i}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y-m_i)^2}{2\sigma_i^2}\right) \tag{9}$$

where $m_i = \boldsymbol{a}^\top \boldsymbol{\mu}_i$ and $\sigma_i^2 = \boldsymbol{a}^\top \boldsymbol{C}_i \boldsymbol{a}$ are the mean and variance of the projected data from class ω_i. Now, using the identity

$$\int y\, e^{-\frac{y^2}{2\sigma^2}}\, \mathrm{d}y = -\sigma^2\, e^{-\frac{y^2}{2\sigma^2}} + \text{ constant of integration}$$

some algebra shows that, under the zero-mean assumption,

$$\mathrm{E}[|y|] = \int_{-\infty}^{\infty} |y| f(y) \mathrm{d}y = \sqrt{2} \sum_{i=1}^{2} \pi_i \sigma_i g(\alpha_i)$$

where

$$\alpha_i = \frac{m_i}{\sqrt{2}\sigma_i}$$

$$g(\alpha_i) = \frac{1}{\sqrt{\pi}} \exp(-\alpha_i^2) + \alpha_i \,\mathrm{erf}\,(\alpha_i)$$

and $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt$ is the Gauss error function. To simplify the discussion, let us address two different extreme, but complementary, situations:

A. Well-separated classes

Consider first the case where the means of the two classes are large compared to the spread of each class, allowing us to write $|\alpha_i| \to \infty$. Some algebra, and the zero-mean assumption, show that

$$\mathrm{E}[|y|] \underset{|\alpha_i|\to\infty}{\longrightarrow} 2\pi_1\pi_2|\Delta m| \tag{10}$$

where $\Delta m = m_2 - m_1 = \boldsymbol{a}^\top(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$. It readily follows that $\mathrm{E}[|y|]$ is maximal when $\boldsymbol{a}$ is in the direction of the line joining the class centroids, i.e., $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$.

This finding reminds of Fisher's linear discriminant analysis (LDA), which is arguably the oldest classification technique still in use. LDA, in a nutshell, projects the data onto the

direction of vector $\boldsymbol{C}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$, which has the property of maximizing the separation between the classes. Here, $\boldsymbol{C}_W$ is the within-class covariance matrix defined in (6). Using (5) and the Sherman-Woodbury-Morrison formula [7], we get

$$\begin{aligned}\boldsymbol{C}_W^{-1} &= (\boldsymbol{I} - \boldsymbol{C}_B)^{-1} \\ &= \boldsymbol{I} - \frac{\pi_1\pi_2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top}{1 + \pi_1\pi_2\|(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\|^2} \end{aligned} \tag{11}$$

Therefore

$$\boldsymbol{C}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \propto (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

which is the direction that maximizes (10). We conclude that L1-PCA behaves as LDA, even though L1-PCA is an unsupervised approach while LDA is not. A more detailed treatment is provided in [8].

B. Equal-mean classes

After considering well-separated classes, suppose now on the contrary that the populations ω_1 and ω_2 share the same mean, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$, which can be assumed to be zero. This causes the point clouds to partially overlap. It follows that $m_1 = m_2 = 0$ and thus

$$\mathrm{E}[|y|] = (2/\pi)^{1/2}(\pi_1\,\sigma_1 + \pi_2\,\sigma_2) \tag{12}$$

Observe that σ_1 and σ_2 are linked through eqn. (8) and, hence, when one of them increases the other decreases, and *vice versa*. As a consequence of this constraint, it holds true that (proof is omitted):

Theorem 1: Under the whitening assumption, the minimizers of (12) with the constraint $\|\boldsymbol{a}\| = 1$ maximize or minimize the power ratio

$$R(\boldsymbol{a}) = \frac{\sigma_1^2}{\sigma_2^2} \tag{13}$$

This result is better understood in the framework of the technique known as common spatial patterns (CSP), which is widely used in brain-computer interfaces (BCIs). CSP linearly projects zero-mean data onto directions where the criterion (13) is maximal or minimal, i.e., where the variance of the projected data points is significantly higher for one class than for the other. The projected data variances are then used as features for classification [9].

Consequently, the L1 criterion, depending on the scenario, possesses the discriminative capabilities of LDA and CSP: maximizing the L1-norm in the case of well-separated cluster yields LDA; minimizing the criterion when the classes share the same mean produces CSP. Quite interestingly, LDA and CSP are supervised techniques, which require correctly labeled data. On the contrary, the L1 criterion $\mathrm{E}[|y|]$ *can be calculated and optimized in a completely unsupervisedly fashion.*

IV. Computer experiments

As demo to illustrate that L1-PCA possesses discrimination properties, and can be thus used for unsupervised classification, we will use a curious test set consisting of images of sheeps from different breeds [10]. Specifically, we will employ in this experiment 420 colour images of size 181×156 of merino sheeps (see Fig. 1) as well as another 420 images, of the same size, corresponding to suffolk sheeps (see Fig. 2).

Fig. 1. A typical merino sheep

Fig. 2. A typical suffolk sheep

As pre-processing steps, all images are firstly converted from colour to grayscale for simplifying the computer vision problem and, secondly, vectorized, i.e., transformed into a 181×156 column vector. Thirdly, to reduce the dimensionality of the data, we project them onto a space of 30 dimensions using the random projection technique (RP) described in [11]. As its name suggests, RP consists in projecting the data onto random directions to produce a lower dimensional subspace. Thanks to Johnson-Lindenstrauss lemma [12], distances are (almost) preserved after projection. As a result, we get 840 observed vectors of size 30×1, which are collected into a data matrix $\boldsymbol{X}$. Finally, the data matrix is centered, by subtracting the mean vector from each observation, and whitened.

As a preliminary analysis, to get an idea about the separation between classes, t-distributed Stochastic Neighbor Embedding (t-SNE) is used to visualize the 30-dimensional data [13]. This technique generates a two-dimensional embedding of the data points, in such a way that nearby points in the embedding correspond to similar objects while distant points correspond to dissimilar objects. The scatter plot of the points generated by t-SNE is shown in Fig. 3, and suggests that 'merino' and 'suffolk' mainly occupy different regions of the space.

To seek for directions that maximize or minimize the L1-

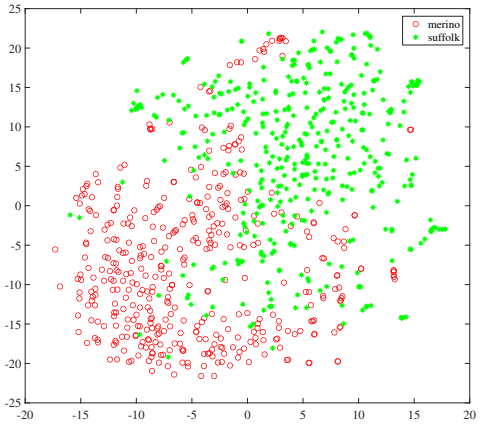


Fig. 3. t-SNE visualizations of 'merino' and 'suffolk' sheeps. The colors of the points indicate the classes of the corresponding individuals.

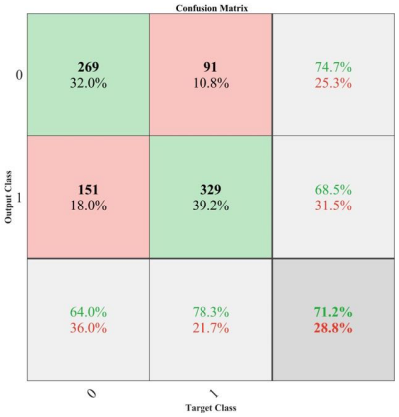


Fig. 4. Confusion matrix of predicted breeds: classes 0 and 1 correspond, respectively, to 'merino' and 'suffolk'.

norm in an natural way, we define the objective function

$$J(\boldsymbol{A}) = \sum_{i=1,\ldots,30} E[|y_i|] \tag{14}$$

where $y_i = \boldsymbol{a}_i^\top \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^{30}$ is the random variable representing the observed sheep images, and $\boldsymbol{A}$ is the matrix whose ith column is $\boldsymbol{a}_i \in \mathbb{R}^{30}$, i.e., $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{30}]$. Next, we use a gradient ascent algorithm to find the matrix $\boldsymbol{A}_M$ that maximizes (14), and a gradient descent method to find the matrix $\boldsymbol{A}_m$ that minimizes the same criterion. In both cases, we restrict matrix $\boldsymbol{A}$ to be orthogonal. The gradient algorithm in [14] is used to carry out these optimizations. Observe that we do not require the data labels to calculate and optimize (14).

Finally, data matrix $\boldsymbol{X}$ is linearly transformed by $\boldsymbol{A}_m$ and $\boldsymbol{A}_M$ and the transformed data are clustered into two groups, in an unsupervised way, by using the well-known Gaussian Mixture Model algorithm. Then, by assigning a point to the Gaussian distribution it most probably belongs to, we partition the space in two classification regions. The performance of the classification can be evaluated by using the true labels as ground truth. Figure 4 illustrates the resulting confusion matrix, showing that the global accuracy equals 71.2%. While this is a modest number, with obvious room for improvement, it serves to illustrate that L1-PCA, after a whitening preprocessing, is endowed with certain discrimination properties.

V. Conclusions

This paper has theoretically shown that projecting whitened data onto the few dimensions that maximize or minimize the absolute value of the projected data points can perform classification in a fully unsupervised fashion, sparing the need for training data and opening new lines of research in the area of L1-PCA. Further research is required to develop a working algorithm capable of exploiting this property in practical settings.

References

[1] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1672–1680, Sep 2008.

[2] P. P. Markopoulos, S. Kundu, S. Chamadia, N. Tsagkarakis, and D. A. Pados, "Outlier-resistant data processing with L1-norm principal component analysis," in *Advances in Principal Component Analysis*. Springer Singapore, Dec 2017, pp. 121–135.

[3] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5046–5058, Oct 2014.

[4] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, Aug 2017.

[5] R. Martín-Clemente and V. Zarzoso, "On the link between L1-PCA and ICA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 515–528, Mar 2017.

[6] I. T. Jolliffe, *Principal component analysis*. New York, NY: Springer, 2002.

[7] G. Golub and C. Van Loan, *Matrix computations*. Baltimore: Johns Hopkins University Press, 1996.

[8] R. Martín-Clemente and V. Zarzoso, "LDA via L1-PCA of whitened data," *IEEE Transactions on Signal Processing*, vol. 68, pp. 225–240, Nov 2020.

[9] R. Martín-Clemente, J. Olias, D. Thiyam, A. Cichocki, and S. Cruces, "Information theoretic approaches for motor-imagery BCI systems: Review and experimental comparison," *Entropy*, vol. 20, no. 1, p. 7, Jan 2018.

[10] [Online]. Available: https://www.kaggle.com/kerneler/starter-sheep-breed-classification-25f1b9f4-3

[11] P. Li, T. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, Eds. ACM, 2006, pp. 287–296.

[12] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," pp. 189–206, 1984.

[13] van der Maaten Laurens and H. Geoffrey, "Visualizing data using t-sne," *Journal of Machine Learning Research*, no. 9, pp. 2579–2605, 2008.

[14] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, Jan 1998.

D Artículo 4

Rubén Martín-Clemente, Jose Luis Camargo, Susana Hornillo-Mellado, Vicente Zarzoso, «L1-norm based PCA for unsupervised classification», *1st International Electronic Conference on Applied Sciences*, Nov. 2020, *https://doi.org/doi:10.3390/ASEC2020-07639*.

proceedings

MDPI

Proceedings

L1 norm based PCA for unsupervised classification †

J.L. Camargo-Olivares [1], R. Martín-Clemente [1,*], S. Hornillo-Mellado[1] and V. Zarzoso [2]

[1] Signal Processing and Communications Department, University of Seville, Spain; jlcamargo@yahoo.es, ruben@us.es, susanah@us.es

[2] I3S Laboratory, University of Cotê d'Azur, France; zarzoso@i3s.unice.fr

* Correspondence: ruben@us.es

† Presented at the 1st International Electronic Conference on Applied Sciences, 10–30 November 2020; Available online: https://sciforum.net/conference/ASEC2020

Published: –

Abstract: Principal component analysis (PCA) is a widespread technique for the analysis of multivariate data, which finds applications in the fields of machine learning and artificial intelligence. Standard PCA seeks to calculate the subspace that minimizes the Euclidean distance (L2-norm) of the data points to it. Unfortunately, PCA is extremely sensitive to the presence of large outliers in the data. Recently, the L1-norm has been proposed as an alternative criterion to classical L2-norm in PCA, drawing considerable research interest on account of its increased robustness to outliers. The proposed contribution shows that, when combined with a whitening preprocessing step, L1-norm based PCA is endowed with discriminative power and can perform data classification in an *unsupervised manner*, i.e., sparing the need for labelled data. By minimizing the L1-norm in the feature space, the technique mimics the action of *common spatial patterns* (CSP), a supervised feature extraction method used in brain computer interfaces. This result is of theoretical interest and opens new interesting research perspectives for L1-PCA. Furthermore, it enables us to perform classification using algorithms for optimizing the L1-norm, which inherit the improved robustness to outliers of the L1-norm criterion. Several numerical experiments will confirm the theoretical findings.

Keywords: principal component analysis; binary classification; machine learning

1. Introduction

L1-norm based criteria are becoming increasingly popular in the fields of machine learning and signal processing. In particular, there is growing interest for the development of L1-norm Principal Component Analysis (L1-PCA) [1,2]. L1-PCA is a variant of traditional PCA which offers enhanced robustness against large outliers. This is an interesting feature because outliers, which are erroneous measurements that lie far apart from the main bulk of the data, are very common in experimental datasets, due to the imperfections in the measuring instruments or the environmental conditions. Specifically, L1-PCA has proven to be highly effective for the restoration of faulty data, in the reconstruction of occluded images or in dimensionality reduction problems [1–4]. However, as negative points, L1-PCA algorithms are either computationally intensive and time consuming [3], despite efforts to simplify their operation [4], or prone to fall into local optima [1]. Furthermore, L1-PCA is a difficult subject to analyze mathematically because, implicitly, it involves the higher-order statistics of the data. For one reason or another, only a few attempts have been made to explain the behavior of L1-PCA in practical situations. Among them, to cite an example, [5] showed that L1-PCA is able to perform Independent Component Analysis (ICA) if the data follows the ICA model. The present contribution continues to investigate the properties of L1-PCA. Here, we report

that L1-PCA, after a minor modification, replicates the operation of the technique known as Common Spatial Patterns (CSP), a *supervised* feature extraction method used in brain computer interfaces [6]. As a result, L1-PCA can be used to separate overlapping populations that are normally distributed and perform data classification in an *unsupervised manner*, i.e., sparing the need for labelled data, which is a remarkable feature. This finding opens new interesting research perspectives for L1-PCA in the field of machine learning. Furthermore, it enables us to develop classification algorithms based on the L1-norm, which inherit the improved robustness to outliers of the L1-norm criterion.

The paper is organized as follows: Section 2 introduces the L1-norm from standard PCA. Section 3 shows that the L1-norm is endowed with discriminative properties in binary classification scenarios. Section 4 illustrates the performance of the approach through computer simulations. Finally, section 5 brings the paper to an end.

2. Background

Let $\boldsymbol{x} \in \mathbb{R}^p$ be a multivariate random variable measured or observed during an experiment. For simplicity, we assume that $\mathrm{E}[\boldsymbol{x}] = \boldsymbol{0}$, where $\mathrm{E}[\cdot]$ is the expectation operator. The aim of standard PCA is to find the best-fit *low*-dimensional subspace for the data points. This is the subspace that minimizes the average squared distance of the data points to it. It can be also shown that this problem is equivalent to finding linear projections of the variables that have maximal variance [7]. A projection onto the direction of a unit vector $\boldsymbol{a}$ is given by

$$y = \boldsymbol{a}^\top \boldsymbol{x}.$$

The variance of the projected data equals

$$\sigma^2(\boldsymbol{a}) = \mathrm{E}\left[y^2\right] \tag{1}$$

The first principal component is the vector that solves the problem

$$\arg\max_{\|\boldsymbol{a}\|_2=1} \sigma^2(\boldsymbol{a}) \tag{2}$$

The $n-$th principal component is the vector that solves the optimization problem (2) subject to the additional constraint of being orthogonal to the previous $n-1$ principal components. The desired best-fitting subspace, finally, is the span of the first few principal components. They are, in other words, the most significant directions characterizing the point cloud of $\boldsymbol{x}$.

However, it is well-known that standard PCA overreacts to large outliers because it takes the square of the projected data in (1). In order to palliate this weakness, [1] proposed the replacement of the square function by the absolute value, yielding the following alternative criterion:

$$\arg\max_{\|\boldsymbol{a}\|_2=1} \mathrm{E}[|y|] \tag{3}$$

In practice, given a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the random variable $\boldsymbol{x}$, (3) is approximated by its sample based estimate

$$\max_{\|\boldsymbol{a}\|_2=1} \quad \frac{1}{N} \sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i| \tag{4}$$

Because $\sum_{i=1}^{N} |\boldsymbol{a}^\top \boldsymbol{x}_i|$ represents the L1-norm of the vector $\boldsymbol{y}$ whose kth entry is given by $y_k = \boldsymbol{a}^\top \boldsymbol{x}_i$, PCA based on criterion (3) is usually referred to as 'L1-norm based PCA' or, simply, 'L1-PCA'. Working algorithms for solving (3) have been proposed in [1,3,4].

2.1. L1-PCA in the case of Gaussian data

To gain some insight into the performance of L1-PCA, let us make the usual assumption that the probability density function of the data is a p-variate normal density function of the form

$$p(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{C})^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}^{-1}\boldsymbol{x}} \tag{5}$$

where $\boldsymbol{C} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is the data covariance matrix. Let $y = \boldsymbol{a}^\top \boldsymbol{x}$ be the projection of $\boldsymbol{x}$ into the direction defined by $\boldsymbol{a} \in \mathbb{R}^p$. The probability density function of y is given by

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \tag{6}$$

where $\sigma^2 = \boldsymbol{a}^\top \boldsymbol{C}\boldsymbol{a}$ is the variance of the projected data. Now, some calculus shows that

$$\mathrm{E}[|y|] = \int_{-\infty}^{\infty} |y|\, p(y)\, \mathrm{d}y = \sqrt{\frac{2}{\pi}}\,\sigma$$

Then, as maximizing the standard deviation σ is equivalent to maximizing the variance σ^2, one sees that L1-PCA behaves in this case like traditional PCA, while offering robustness against the presence of large outliers in the data [1–3].

3. L1-norm based classification

Binary classification problems are ubiquitous in many real-life applications. Consider that we observe random samples drawn from two different populations ω_1 and ω_2 with the same population mean, assumed to be zero. It is supposed that the distribution of the random samples can be modeled as a mixture of Gaussians, i.e.,

$$p(\boldsymbol{x}) = (2\pi)^{-\frac{p}{2}} \sum_{i=1}^{2} \pi_i \det(\boldsymbol{C}_i)^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{C}_i^{-1}\boldsymbol{x}}$$

where π_1 and π_2 are the *a priori* probabilities of occurrence of ω_1 and ω_2, with $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ the corresponding class covariance matrices ($\boldsymbol{C}_1 \neq \boldsymbol{C}_2$). Consider again the L1-norm criterion

$$J(\boldsymbol{a}) = \mathrm{E}[|y|] = \mathrm{E}[|\boldsymbol{a}^\top \boldsymbol{x}|] \tag{7}$$

Similar calculus as above shows that

$$J(\boldsymbol{a}) = \sqrt{\frac{2}{\pi}}\,(\pi_1\,\sigma_1(\boldsymbol{a}) + \pi_2\,\sigma_2(\boldsymbol{a})) \tag{8}$$

where $\sigma_i^2(\boldsymbol{a}) = \boldsymbol{a}^\top \boldsymbol{C}_i \boldsymbol{a}$ is the variance of the ith class in the direction of the unit vector $\boldsymbol{a} \in \mathbb{R}^p$.

Let us assume hereafter, without any loss of generality, that the data are *whitened*. A random variable $\boldsymbol{x}$ is whitened by multiplying it by a matrix $\boldsymbol{Q}$ so that the result $\boldsymbol{Q}\boldsymbol{x}$ has covariance $\boldsymbol{Q}\boldsymbol{C}\boldsymbol{Q}^\top = \boldsymbol{I}$, where $\boldsymbol{C} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ and $\boldsymbol{I}$ is the identity matrix. This goal can be achieved in practice by setting $\boldsymbol{Q} = \boldsymbol{C}^{-1/2}$. To keep the notation simple, the whitened data are also denoted, with some abuse, by $\boldsymbol{x}$. Likewise, the whitened class covariance matrices are still denoted by $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. Whitening implies that

$$\boldsymbol{C} = E[\boldsymbol{x}\boldsymbol{x}^\top] = \pi_1 \boldsymbol{C}_1 + \pi_2 \boldsymbol{C}_2 = \boldsymbol{I} \tag{9}$$

$$E[y^2] = \boldsymbol{a}^\top E[\boldsymbol{x}\boldsymbol{x}^\top]\boldsymbol{a} = \pi_1\sigma_1^2 + \pi_2\sigma_2^2 = 1 \tag{10}$$

Furthermore, eqn. (8) still holds true. The real utility of whitening is that it introduces a constraint, namely, eqn. (10), on the class variances: when one of them increases the other decreases, and *vice versa*. As a consequence, a thorough analysis leads to the following result (proof is omitted):

Theorem 1. *Under the whitening assumption, the minimizers of* (8) *with the constraint* $\|\boldsymbol{a}\| = 1$ *maximize or minimize the power ratio*

$$R(\boldsymbol{a}) = \frac{\sigma_1^2}{\sigma_2^2} \tag{11}$$

This Theorem can be put in relation to the useful technique known as common spatial patterns (CSP), which is widely used in brain-computer interfaces (BCIs). Typically, electroencephalogram (EEG) samples are acquired under two different experimental conditions (e.g. imagining left and right hand movements). CSP linearly projects the data onto directions where the ratio (11) is maximal or minimal or, in simple words, where the variance of the projected data points is significantly higher for one class than for the other. The projected data variances are then be used as features for classification [6]. It follows that the L1 criterion possesses the discriminative capabilities of CSP. Quite interestingly, CSP is a supervised technique, whose performance relies heavily on the availability of correctly labeled data. On the contrary, minimizing the L1 criterion (8) *can be performed in a completely unsupervisedly fashion*.

4. Computer experiments

Some experiments are now conducted to illustrate the potential of the L1-approach

4.1. Experiment 1

To illustrate Theorem 1, let us consider a mixture in a bidimensional space of two equiprobable Gaussian classes, i.e., $\pi_1 = \pi_2 = 1/2$, with zero-means and respective covariances

$$\boldsymbol{C}_1 = \begin{pmatrix} 1 & 0.68 \\ 0.68 & 1 \end{pmatrix} \text{ and } \boldsymbol{C}_2 = \begin{pmatrix} 1 & -0.68 \\ -0.68 & 1 \end{pmatrix}. \tag{12}$$

Observe that matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ fulfill the whitening condition (9). Figure 1 represents the theoretically exact value of the cost function $J(\theta) = \mathrm{E}[|\boldsymbol{a}(\theta)^\top \boldsymbol{x}|]$, with $\boldsymbol{a}(\theta) = [\cos(\theta), \sin(\theta)]^\top$, calculated from eqn. (8). For reference, we also plot the power ratio $R(\theta)$, defined as in (11), in the same Figure. We see that the minima of the L1-cost $J(\theta)$ correspond with either the maximum or the minimum of $R(\theta)$, as predicted by the Theorem. We also see that the maxima of $J(\theta)$ are at 0 and $\pm\pi/2$ rad. At these points, the standard deviations of the projected populations are the same, i.e., $\sigma_1 = \sigma_2$, with $\sigma_i^2 = \boldsymbol{a}^\top \boldsymbol{C}_i \boldsymbol{a}$. It follows that the projected populations are totally mixed, because the different classes cannot be distinguished from each other.

4.2. Experiment 2

To test the L1-norm approach in a multidimensional setting, we perform several experiments with $p \in \{2, 5, 10, 15, 20, 25, 30\}$. In each one, we draw $N = 50\,p$ samples per each of the two Gaussian classes, and the covariance matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are randomly generated. After applying a *whitening* pre-processing to the data, the gradient descent algorithm in [8] is applied to find the orthogonal directions that (globally or locally) minimize the L1-norm criterion (7). The closeness to the subspace spanned by the line in the direction of the global minimum is used as unsupervised criterion to classify the random samples into one cluster or the other. Fig. 2 shows the accuracy of the classification, averaged over 100 independent experiments. Furthermore, L1-norm criteria are also expected to exhibit robustness against large outliers.

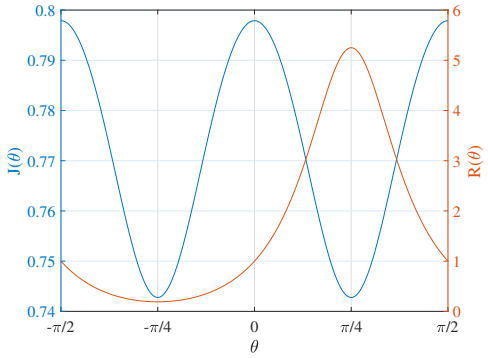


Figure 1. L1-norm function $J(\theta) = \mathrm{E}[|\boldsymbol{a}(\theta)^\top \boldsymbol{x}|]$ and power ratio $R(\theta) = \frac{\sigma_1^2(\theta)}{\sigma_2^2(\theta)}$, illustrating the matching between their extrema.

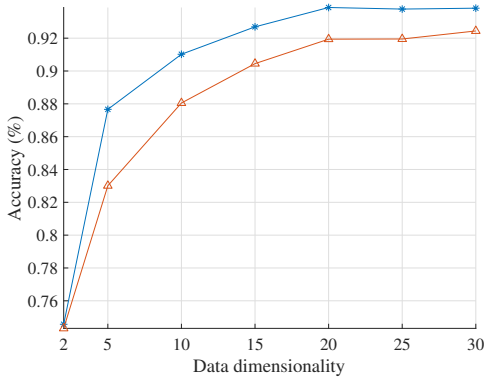


Figure 2. Accuracy in the classification of the data as a function of the data dimensionality. Blue line: accuracy calculated from outlier-free data. Red line: ditto for the outlier-corrupted data.

To test this property, we repeat the experiment with the difference that the whitened data points are now corrupted by replacing 10 per cent of the data samples, at randomly chosen time instants, by Gaussian noise realizations with identity covariance matrix and mean $\boldsymbol{\mu}_{\text{outliers}} = [10, 10, \ldots, 10]^\top$. The new results are also represented in Figure 2, proving the reliability of the L1-norm. In both cases, we see that the performance increases with the dimensionality of the input representation. This finding reflects the well-known fact that it is usually easier to perform classification in high-dimensional spaces.

5. Conclusions

Projecting whitened data onto the few dimensions that minimize the absolute value of the projected data points can perform unsupervised classification in a fully unsupervised fashion, sparing the need for training data and opening new lines of research in the area of L1-PCA. Good performance is shown by numerical experiments.

Appendix

To enable research reproducibility, the following Matlab code can be used to reproduce Fig. 1

```
C1 = [1 0.68; 0.68 1]; C2 = [1 -0.68; -0.68 1]; angles = linspace(-pi/2, pi/2, 100);
for i = 1:numel(angles)
   a = [cos(angles(i)); sin(angles(i))];
   s1(i) = sqrt(a'*C1*a); s2(i) = sqrt(a'*C2*a);
   J(i) = sqrt(0.5/pi)*(s1(i) + s2(i)); R(i) = [s1(i)/s2(i)]^2;
end
yyaxis left, plot(angles, J), ylabel('J(\theta)'),
yyaxis right, plot(angles, R), grid on, ylabel('R(\theta)')
```

In experiment 2, data have been generated for each class by the Matlab command `mvnrnd`. The basic algorithm for finding a direction minimizing the L1-norm is

```
[p,T] = size(X); % X is the data matrix (num features x num samples)
a = randn(p,1); a = a/norm(a); flag=true;
while(flag)
   a_old = a; a = a - 0.1/T*(X*sign(a'*X));  a = a/norm(a);
   if norm(a-a_old) < 0.001, flag = false; end
end
```

Author Contributions: All authors participate in conceptualization, investigation and writing - original draft.

Funding: This work is funded by the research project US-1264994 awarded by the Junta de Andalucía (Consejería de Transformación Económica, Industria, Conocimiento y Universidades).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kwak, N. Principal Component Analysis Based on L1-Norm Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2008**, *30*, 1672–1680.
2. Markopoulos, P.P.; Kundu, S.; Chamadia, S.; Tsagkarakis, N.; Pados, D.A. Outlier-Resistant Data Processing with L1-Norm Principal Component Analysis. In *Advances in Principal Component Analysis*; Springer Singapore, 2017; pp. 121–135.
3. Markopoulos, P.P.; Karystinos, G.N.; Pados, D.A. Optimal Algorithms for L1-subspace Signal Processing. *IEEE Transactions on Signal Processing* **2014**, *62*, 5046–5058.
4. Markopoulos, P.P.; Kundu, S.; Chamadia, S.; Pados, D.A. Efficient L1-Norm Principal-Component Analysis via Bit Flipping. *IEEE Transactions on Signal Processing* **2017**, *65*, 4252–4264.
5. Martín-Clemente, R.; Zarzoso, V. On the Link Between L1-PCA and ICA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 515–528.
6. Martín-Clemente, R.; Olias, J.; Thiyam, D.; Cichocki, A.; Cruces, S. Information Theoretic Approaches for Motor-Imagery BCI Systems: Review and Experimental Comparison. *Entropy* **2018**, *20*, 7.
7. Jolliffe, I.T. *Principal component analysis*; Springer: New York, NY, 2002.
8. Edelman, A.; Arias, T.A.; Smith, S.T. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications* **1998**, *20*, 303–353.

Índice de Figuras

Índice de Tablas

Bibliografía

[1] C. Aggarwal and C. Reddy, *Data clustering: Algorithms and applications*, Chapman and Hall/CRC, 2013.

[2] Charu C. Aggarwal, *Outlier analysis*, Springer-Verlag GmbH, 2016.

[3] Hui Hui Dai Alan Jeffrey, *Handbook of mathematical formulas and integrals [with cdrom]*, Academic Press, 2008.

[4] Giuseppe Amato, Vlastislav Dohnal, Pavel Zezula, and Michal Batko, *Similarity search*, Springer US, 2005.

[5] Yali Amit and Donald Geman, *Shape quantization and recognition with randomized trees*, Neural Computation **9** (1997), no. 7, 1545–1588.

[6] Abdullah Bal and Mohammad S. Alam, *Automatic target tracking in forward-looking infrared video sequences using tuned basis functions*, Optical Engineering **55** (2016), no. 7, 073102.

[7] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, *Eigenfaces vs. fisherfaces: recognition using class specific linear projection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 7, 711–720.

[8] M. Berry, A. Mohamed, and B. Yap, *Supervised and unsupervised learning for data science*, Springer-Verlag GmbH, 2019.

[9] Ella Bingham, *Advances in independent component analysis and learning machines*, Academic Press, London, UK, 2015.

[10] Hamidullah Binol, *Improved Fukunaga–Koontz transform with compositional kernel combination for hyperspectral target detection*, Journal of the Indian Society of Remote Sensing **46** (2018), no. 10, 1605–1615.

[11] Hamidullah Binol, Gokhan Bilgin, Semih Dinc, and Abdullah Bal, *Kernel Fukunaga–Koontz transform subspaces for classification of hyperspectral images with small sample sizes*, IEEE Geoscience and Remote Sensing Letters **12** (2015), no. 6, 1287–1291.

[12] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer-Verlag New York Inc., 2006.

[13] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the fifth annual workshop on Computational learning theory - COLT '92, ACM Press, 1992.

[14] José Luis Camargo, Rubén Martín-Clemente, Susana Hornillo-Mellado, and Vicente Zarzoso, *L1-norm unsupervised fukunaga-koontz transform*, Signal Processing **182** (2021), 107942.

[15] F.Z. Chelali, A. Djeradi, and R. Djeradi, *Linear discriminant analysis for face recognition*, 2009 International Conference on Multimedia Computing and Systems, IEEE, apr 2009.

[16] Edwin K. P. Chong and Stanislaw H. Zak, *An introduction to optimization*, John Wiley & Sons, 2013.

[17] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al

Emadi, Mamun Bin Ibne Reaz, and Mohammad Tariqul Islam, *Can AI help in screening viral and COVID-19 pneumonia?*, IEEE Access **8** (2020), 132665–132676.

[18] Cichocki, *Adaptive blind signal and image processing*, John Wiley & Sons, 2002.

[19] Javier Civit-Masot, Francisco Luna-Perejón, Manuel Domínguez Morales, and Anton Civit, *Deep learning system for COVID-19 diagnosis aid using x-ray pulmonary images*, Applied Sciences **10** (2020), no. 13, 4640.

[20] P. Comon and C. Jutten, *Handbook of blind source separation: Independent component analysis and applications*, Academic Press, 2010.

[21] Corinna Cortes and Vladimir Vapnik, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273–297.

[22] Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, Wiley, Apr 2005.

[23] M. Palacios Cruz, E. Santos, M.A. Velázquez Cervantes, and M. León Juárez, *COVID-19, una emergencia de salud pública mundial*, Revista Clínica Española **221** (2021), no. 1, 55–61.

[24] [dataset] Benjamin Blankertz, Carmen Vidaurre, Michael Tangermann, Klaus-Robert Müller, Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, Gert Pfurtscheller, Stephan Waldert, Carsten Mehring, Ad Aertsen, and Gerwin Schalk Niels Birbaumer, Kai J. Miller, *BCI Competition IV dataset*, http://www.bbci.de/competition/iv/, 2008, Last accessed April 2020.

[25] Dennis Decoste and Bernhard Schölkopf, *Training invariant support vector machines*, Machine Learning **46** (2002), no. 1/3, 161–190.

[26] Dipak Dey, *Essential bayesian models : a derivative of handbook of statistics: Bayesian thinking - modeling and computation, vol. 25*, North-Holland, Amsterdam, 2010.

[27] Thomas G. Dietterich, *An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization*, Machine Learning **40** (2000), no. 2, 139–157.

[28] R.P.W. Duin and M. Loog, *Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 6, 732–739.

[29] Alan Edelman, Tomás A. Arias, and Steven T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Applications **20** (1998), no. 2, 303–353.

[30] Allen I. Fleishman, *A method for simulating non-normal distributions*, Psychometrika **43** (1978), no. 4, 521–532.

[31] C. Fraikin, K. Haper, and P. Van Dooren, *Optimization over the stiefel manifold*, Proc. Applied Mathematics and Mechanics **7** (2007), no. 1, 1062205–1062206.

[32] Jerome H. Friedman, *Exploratory projection pursuit*, Journal of the American Statistical Association **82** (1987), no. 397, 249–266.

[33] K. Fukunaga and W.L.G. Koontz, *Application of the Karhunen-Loève expansion to feature selection and ordering*, IEEE Transactions on Computers **C-19** (1970), no. 4, 311–318.

[34] Joseph Giarratano, *Expert systems : principles and programming*, Thomson Course Technology, Australia Boston, Mass, 2005.

[35] Gene Golub and Charles Van Loan, *Matrix computations*, Johns Hopkins University Press, Baltimore, 1996.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT Press Ltd, 2016.

[37] I. S. Gradshteyn and I.M. Ryzhik, *Table of integrals, series and products*, Academic, Oxford, 2007.

[38] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han, *Outlier detection for temporal data*, Synthesis Lectures on Data Mining and Knowledge Discovery **5** (2014), no. 1, 1–129.

[39] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han, *Outlier detection for temporal data: A survey*, IEEE Transactions on Knowledge and Data Engineering **26** (2014), no. 9, 2250–2267.

[40] Max Halperin, H. O. Hartley, and P. G. Hoel, *Recommended standards for statistical symbols and notation. COPSS committee on symbols and notation*, The American Statistician **19** (1965), no. 3, 12.

[41] Shayan Hassantabar, Mohsen Ahmadi, and Abbas Sharifi, *Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches*, Chaos, Solitons & Fractals **140** (2020), 110170.

[42] Mohamad Hassoun, *Fundamentals of artificial neural networks*, MIT Press, Cambridge, Mass, 1995.

[43] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: Data mining, inference, and prediction, second edition*, Springer Nature, 2009.

[44] Simon Haykin, *Neural networks and learning machines*, Prentice Hall/Pearson, New York, 2009.

[45] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal, *Fundamentals of convex analysis*, Springer Berlin Heidelberg, 2001.

[46] Tin Kam Ho, *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998), no. 8, 832–844.

[47] Simon Hoell and Piotr Omenzetter, *Fukunaga-Koontz feature transformation for statistical structural damage detection and hierarchical neuro-fuzzy damage localisation*, Journal of Sound and Vibration **400** (2017), 329–353.

[48] X. Huo, *A statistical analysis of Fkunaga–Koontz transform*, IEEE Signal Processing Letters **11** (2004), no. 2, 123–126.

[49] Xiaoming Huo, Michael Elad, Ana G. Flesia, Robert R. Muise, S. Robert Stanfill, Jerome Friedman, Bogdan Popescu, Jihong Chen, Abhijit Mahalanobis, and David L. Donoho, *Optimal reduced-rank quadratic classifiers using the Fukunaga-Koontz transform with applications to automated target recognition*, Automatic Target Recognition XIII (Firooz A. Sadjadi, ed.), SPIE, Sep 2003.

[50] A. Hyvarinen and E. Oja, *Independent component analysis: algorithms and applications*, Neural Networks **13** (2000), no. 4-5, 411–430.

[51] Aapo Hyvarinen, *Independent component analysis*, Wiley-Interscience, 2001.

[52] G. Nagpal I. Gupta, *Artificial intelligence and expert systems*, Mercury Learning & Information, 2020.

[53] Daniel N. Wilke Jan A Snyman, *Practical mathematical optimization*, Springer-Verlag GmbH, 2018.

[54] I. T. Jolliffe, *Principal component analysis*, Springer, New York, NY, 2002.

[55] M. C. Jones and Robin Sibson, *What is projection pursuit?*, Journal of the Royal Statistical Society. Series A (General) **150** (1987), no. 1, 1.

[56] Felix Juefei-Xu and Marios Savvides, *Multi-class Fukunaga Koontz discriminant analysis for enhanced face recognition*, Pattern Recognition **52** (2016), 186–205.

[57] Yong Gyu Jung, Min Soo Kang, and Jun Heo, *Clustering performance comparison using k-means and expectation maximization algorithms*, Biotechnology & Biotechnological Equipment **28** (2014), no. sup1, S44–S48.

[58] George N. Karystinos and Athanasios P. Liavas, *Efficient computation of the binary vector that maximizes a rank-deficient quadratic form*, IEEE Transactions on Information Theory **56** (2010), no. 7, 3581–3593.

[59] Steven Kay, *Intuitive probability and random processes using matlab®*, Springer-Verlag GmbH, 2006.

[60] Steven M. Kay, *Fundamentals of statistical processing, volume i*, Prentice Hall, 1993.

[61] Agnan Kessy, Alex Lewin, and Korbinian Strimmer, *Optimal whitening and decorrelation*, The American Statistician **72** (2018), no. 4, 309–314.

[62] M. Kirby and L. Sirovich, *Application of the karhunen-loeve procedure for the characterization of human faces*, IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990), no. 1, 103–108.

[63] Zoltan J. Koles, Michael S. Lazar, and Steven Z. Zhou, *Spatial patterns underlying population differences in the background EEG*, Brain Topography **2** (1990), no. 4, 275–284.

[64] Hans-Peter Kriegel, Peer Kroger, Jorg Sander, and Arthur Zimek, *Density-based clustering*, WIREs Data Mining and Knowledge Discovery **1** (2011), no. 3, 231–240.

[65] N. Kwak, *Principal component analysis based on L1-norm maximization*, IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008), no. 9, 1672–1680.

[66] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, *Deep learning*, Nature **521** (2015), no. 7553, 436–444.

[67] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, *Brain–computer communication: Motivation, aim, and impact of exploring a virtual apartment*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **15** (2007), no. 4, 473–482.

[68] Jun Li, Li Fuxin, and Sinisa Todorovic, *Efficient riemannian optimization on the stiefel manifold via the cayley transform*.

[69] X. Li and K. Wong, *Natural computing for unsupervised learning*, Springer-Verlag GmbH, 2018.

[70] Ruiming Liu, Erqi Liu, Jie Yang, Yong Zeng, Fanglin Wang, and Yuan Cao, *Automatically detect and track infrared small targets with kernel Fukunaga-Koontz transform and Kalman prediction*, Applied Optics **46** (2007), no. 31, 7780.

[71] Tingfeng Liu, Hui Gao, and Jianjun Wu, *Review of outlier detection algorithms based on grain storage temperature data*, 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), IEEE, jun 2020.

[72] F Lotte, L Bougrain, A Cichocki, M Clerc, M Congedo, A Rakotomamonjy, and F Yger, *A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update*, Journal of Neural Engineering **15** (2018), no. 3, 031005.

[73] Harshada C. Mandhare and S. R. Idate, *A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques*, 2017 International Conference

on Intelligent Computing and Control Systems (ICICCS), IEEE, jun 2017.

[74] Adam H. Marblestone, Greg Wayne, and Konrad P. Kording, *Toward an integration of deep learning and neuroscience*, Frontiers in Computational Neuroscience **10** (2016).

[75] Panos P. Markopoulos, George N. Karystinos, and Dimitris A. Pados, *Optimal algorithms for L1-subspace signal processing*, IEEE Transactions on Signal Processing **62** (2014), no. 19, 5046–5058.

[76] Panos P. Markopoulos, Sandipan Kundu, Shubham Chamadia, and Dimitris A. Pados, *Efficient L1-norm principal-component analysis via bit flipping*, IEEE Transactions on Signal Processing **65** (2017), no. 16, 4252–4264.

[77] Panos P. Markopoulos, Sandipan Kundu, Shubham Chamadia, Nicholas Tsagkarakis, and Dimitris A. Pados, *Outlier-resistant data processing with L1-norm principal component analysis*, Advances in Principal Component Analysis, Springer Singapore, Dec 2017, pp. 121–135.

[78] Rubén Martín-Clemente, Javier Olias, Deepa Thiyam, Andrzej Cichocki, and Sergio Cruces, *Information theoretic approaches for motor-imagery BCI systems: Review and experimental comparison*, Entropy **20** (2018), no. 1, 7.

[79] Ruben Martín-Clemente and Vicente Zarzoso, *On the link between L1-PCA and ICA*, IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017), no. 3, 515–528.

[80] _____, *LDA via L1-PCA of whitened data*, IEEE Transactions on Signal Processing **68** (2020), 225–240.

[81] John Mathews, *Numerical methods using matlab*, Prentice Hall, Upper Saddle River, N.J, 1999.

[82] Abhilash Miranda and Paul Whelan, *Fukunaga-Koontz transform for small sample size problems*, Proceedings of the IEE Irish Signals and Systems Conference (ISSC, Dublin, Ireland, 2005), Sep 2005.

[83] Tom Mitchell, *Machine learning*, McGraw-Hill, New York, 1997.

[84] Guy P. Nason, *Robust projection indices*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63** (2001), no. 3, 551–567.

[85] Pablo Navarrete and Javier Ruiz del Solar, *Analysis and comparison of eigenspace-based face recognition approaches*, International Journal of Pattern Recognition and Artificial Intelligence **16** (2002), no. 07, 817–830.

[86] Ben Noble, *Applied linear algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[87] S. Ochilov, M. S. Alam, and A. Bal, *Fukunaga-Koontz transform based dimensionality reduction for hyperspectral imagery*, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII (Sylvia S. Shen and Paul E. Lewis, eds.), SPIE, May 2006.

[88] Javier Olias, Ruben Martín-Clemente, M. Auxiliadora Sarmiento-Vega, and Sergio Cruces, *EEG signal processing in MI-BCI applications with improved covariance matrix estimators*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **27** (2019), no. 5, 895–904.

[89] Daniel Pena, *Anlisis de datos multivariantes*, McGraw-Hill/Interamericana, Madrid, 2002.

[90] Jing Peng, Guna Seetharaman, Wei Fan, Stefan Robila, and Aparna Varde, *Chernoff dimensionality reduction–where Fisher meets FKT*, Proceedings of the 2011 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, Apr 2011.

[91] Alhajj R. and Rokne J., *Singular value decomposition*, Encyclopedia of Social Network Analysis and Mining, Springer New York, 2018, pp. 2538–2538.

[92] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, Serkan Kiranyaz, Saad Bin Abul Kashem, Mohammad Tariqul Islam, Somaya Al Maadeed, Susu M. Zughaier, Muhammad Salman Khan, and Muhammad E.H. Chowdhury, *Exploring the effect of image enhancement techniques on COVID-19 detection using chest x-ray images*, Computers in Biology and Medicine **132** (2021), 104319.

[93] Stuart Russell, *Artificial intelligence : a modern approach*, Prentice Hall, Upper Saddle River, New Jersey, 2010.

[94] A. L. Samuel, *Some studies in machine learning using the game of checkers*, IBM Journal of Research and Development **3** (1959), no. 3, 210–229.

[95] Geza Schay, *Introduction to probability with statistical applications*, Birkhauser, 2018.

[96] Jürgen Schmidhuber, *Deep learning in neural networks: An overview*, Neural Networks **61** (2015), 85–117.

[97] Viktor Schonberger, *Big data : la revolución de los datos masivos*, Turner, Madrid, 2013.

[98] Konstantinos Koutroumbas Sergios Theodoridis, *Pattern recognition*, Syngress Media, 2008.

[99] James V. Stone, *Independent component analysis*, MIT Press Ltd, 2004.

[100] Gilbert Strang, *Introduction to linear algebra*, Cambridge University Press, 2016.

[101] Hao Su, Jie Yang, Lei Sun, and Zhiping Lin, *A solver of fukunaga koontz transformation without matrix decomposition*, 2021 IEEE

International Symposium on Circuits and Systems (ISCAS), IEEE, may 2021.

[102] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J. Miller, Gernot R. Müller-Putz, Guido Nolte, Gert Pfurtscheller, Hubert Preissl, Gerwin Schalk, Alois Schlögl, Carmen Vidaurre, Stephan Waldert, and Benjamin Blankertz, *Review of the BCI competition IV*, Frontiers in Neuroscience **6** (2012).

[103] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien, *Linear discriminant analysis: A detailed tutorial*, AI Communications **30** (2017), no. 2, 169–190.

[104] Sergios Theodoridis, *Introduction to pattern recognition : a matlab approach*, Academic, London, 2010.

[105] George Thomas, *Calculus and analytic geometry*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1996.

[106] C. David Vale and Vincent A. Maurelli, *Simulating multivariate nonnormal distributions*, Psychometrika **48** (1983), no. 3, 465–471.

[107] Thirumalaisamy P. Velavan and Christian G. Meyer, *The COVID-19 epidemic*, Tropical Medicine & International Health **25** (2020), no. 3, 278–280.

[108] S. Arungalai Vendan, Liang Gao, Xiaodong Niu, Abhinav Karan, and Rajeev Kamal, *Welding and cutting case studies with supervised machine learning*, Springer-Verlag GmbH, 2020.

[109] Svante Wold, Kim Esbensen, and Paul Geladi, *Principal component analysis*, Chemometrics and Intelligent Laboratory Systems **2** (1987), no. 1-3, 37–52.

[110] Wei Wu, Zhe Chen, Xiaorong Gao, Yuanqing Li, Emery N. Brown, and Shangkai Gao, *Probabilistic common spatial patterns for multichannel EEG analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **37** (2015), no. 3, 639–653.

[111] Sheng Zhang and T. Sim, *When fisher meets fukunaga-koontz: A new look at linear discriminants*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06), IEEE, 2006.

[112] Sheng Zhang and Terence Sim, *Discriminant subspace analysis: A Fukunaga-Koontz approach*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007), no. 10, 1732–1745.