Una Técnica Robusta Para Kernel PCA

por

Luis Ernesto Mora Forsbach

Lic., Universidad Veracruzana (2002)

Sometida a revisión al Departamento de Ciencias de la Computación como cumplimiento parcial de los requisitos para la obtención del grado de

Maestría en Ciencias

en el

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS

Octubre 2005

Firma del Autor .....
Departamento de Ciencias de la Computación
Octubre 2005

Aprobada por .....
Johan Jozef Lode van Horebeek
Director de tesis

Aprobada por .....
Mariano José Juan Rivera Meraz

Aprobada por .....
Rogelio Hasimoto Beltrán

Una Técnica Robusta Para Kernel PCA

por
Luis Ernesto Mora Forsbach

Sometida a revisión al Departamento de Ciencias de la Computación
en Octubre 2005, en cumplimiento parcial de los requisitos
para la obtención del grado de
Maestría en Ciencias

Resumen

Kernel PCA generaliza el Análisis de Componentes Principales (PCA) a dominios no-lineales. A pesar de ser una técnica ampliamente utilizada en aplicaciones de diversa índole, poco se ha hecho por estudiar la influencia de observaciones atípicas (outliers) que pudieran presentarse en el conjunto de datos que se analiza. Ésto aún cuando es conocido que PCA es una técnica muy sensible a este tipo de observaciones y que Kernel PCA hereda esta sensibilidad. Existen muchos métodos para realizar PCA de forma robusta, por ello, el objetivo principal de este trabajo es presentar una versión de Kernel PCA robusto que se basa en uno de esos métodos. El método propuesto corresponde a una generalización de un método robusto para PCA del mismo modo como Kernel PCA generaliza a PCA clásico. Para ello se modifican los estimadores de la matriz de covarianza y media usados en Kernel PCA y se hace uso de la distancia de Mahalanobis en el espacio implícito mediante el uso del kernel. Se evalua y discute el método propuesto en esta tesis con varias aplicaciones.

Director de Tesis: Johan Jozef Lode van Horebeek

Agradecimientos

Agradezco de forma muy especial al Dr. Johan van Horebeek por toda su paciencia a lo largo del desarrollo de esta tesis, su ayuda y motivación constante.

A todo el personal del Centro de Investigación en Matemáticas, en particular al departamento de Ciencias de la Computación, por haberme permitido ser parte de su comunidad y brindarme todo el apoyo que recibí.

Al Consejo Nacional de Ciencia y Tecnología, por el financiamiento para realizar mis estudios de maestría.

Notación utilizada

Matriz de datos/datos transformados (observaciones en filas): X

Número de datos: m

Espacio original: $\mathcal{X}$

Dimensión del espacio original: N

Espacio transformado (de características): $\mathcal{H}$

Dimensión del espacio transformado: h

Función de transformación: Φ

Matriz de covarianza (de datos originales/transformados): Σ

Media (de datos originales/transformados): μ

Estimador de matriz de covarianza (de datos originales/transformados): $C = \hat{\Sigma}$

Estimador de media (de datos originales/transformados): $\hat{\mu}$

Producto punto entre los vectores x, y: $\langle x, y \rangle$ ó $(x \cdot y)$

Función de decisión mayor-que: $\overset{?}{>}$

Índice general

Índice de figuras

Capítulo 1

Introducción

El Análisis de Componentes Principales mediante Kernels (Kernel PCA, por sus siglas en inglés) es una técnica propuesta por B. Schölkopf, A. Smola y K.-R. Müller en [10]. Esta técnica generaliza el Análisis de Componentes Principales (PCA) a dominios no-lineales. A pesar de ser ampliamente utilizada muchas aplicaciones de diversa naturaleza, poco se ha hecho para estudiar la influencia de observaciones atípicas (outliers) que pudieran presentarse al realizar Kernel PCA. Es un aspecto relevante que PCA es una técnica muy sensible a este tipo de observaciones.

Al ser Kernel PCA una generalización directa de PCA - y en cuyo desarrollo no se considera la posible presencia de outliers -, cabe esperarse que su desempeño mejore si inicialmente se formula considerando la posibilidad de la presencia de tales observaciones en los datos con los que se trabaja. El objetivo principal de este trabajo es presentar un método robusto para Kernel PCA basado en uno de los métodos robustos existentes para PCA clásico. En un sentido más amplio, este trabajo pretende ser un acercamiento a la aplicación de conceptos estadísticos robustos en los métodos de kernel.

Kernel PCA es PCA utilizando el *truco del kernel*. Tal truco se puede aplicar en algoritmos donde los datos aparezcan solamente en forma de productos punto. En Kernel PCA los estimadores de media y covarianza se formulan de forma que se tenga esta característica en el algoritmo. Es por ello que al considerar los trabajos existentes para PCA Robusto se tiene en cuenta la cuestión de que debe siempre mantenerse el poder usar el truco del kernel (asegurando que los datos aparezcan únicamente como productos punto).

La organización de este trabajo es la siguiente: en el capítulo 2 se presenta someramente una exposición sobre PCA, ésta no pretende ser exhaustiva, para un tratamiento completo se puede referir a ([7]) y ([6]) por ejemplo. El Capítulo 3 presenta la teoría sobre Kernel PCA tal como fue desarrollado. El Capítulo 4 trata sobre aquellos métodos existentes en la literatura cuyo objetivo es realizar PCA de manera robusta; la idea es enfocarse en aquellos métodos más ampliamente conocidos dando preferencia a aquellos que puedan ser aplicables al dominio de Kernel PCA, para así presentar la generalización robusta de Kernel PCA que se propone en el Capítulo 5. El Capítulo 6 presenta diversos experimentos en los que se trata de analizar la utilidad y desempeño del método introducido en el Capítulo 5, junto con una pequeña discusión

de lo obtenido. Finalmente, el Capítulo 7 presenta algunas conclusiones y perspectivas sobre este trabajo. Cabe señalar que con el fin de presentar el material aquí expuesto con la mayor claridad posible, se decidió mantener las abreviaciones del inglés del Análisis de Componentes Principales (PCA), Análisis de Componentes Principales mediante Kernels (Kernel PCA o KPCA) e incluso de términos como *outlier* para denominar observaciones atípicas.

Capítulo 2

Análisis de Componentes Principales (PCA)

Este capítulo constituye un repaso de la idea general del Análisis de Componentes Principales (PCA). PCA es probablemente la más antigua y más conocida de las técnicas de análisis multivariado, fue introducido por Pearson en 1901 y desarrollado de manera independiente por Hotelling alrededor de 1933. La idea central de PCA es reducir la dimensionalidad (a través de una proyección) de un conjunto de datos en el cual existen un gran número de variables interrelacionadas, manteniendo lo más que sea posible de la información presente en los datos. Por definición en PCA, se mide el grado de interés de una dirección a través de la variabilidad de los datos al ser proyectados sobre esta dirección. Se puede demostrar que calcular los componentes principales se reduce al problema de encontrar eigenvalores-eigenvectores de una matriz semi-definida positiva. Se pueden distinguir algunos de los objetivos primordiales de realizar PCA en el área de computación tales como son reducción de dimensionalidad, interpretación y reconocimiento de patrones en datos de alta dimensión.

Realizar PCA tiene que ver con describir la estructura de varianza-covarianza de N variables aleatorias $X_1, X_2, \ldots X_N$. Algebráicamente los Componentes Principales (PC) son combinaciones lineales decorrelacionadas de las N variables aleatorias mientras que geométricamente los PC representan la selección de un nuevo sistema de coordenadas obtenido al rotar el sistema original tomando a $X_1, X_2, \ldots X_N$ como los ejes originales. Al realizar PCA, los nuevos ejes representarán las direcciones ordenadas en variabilidad decreciente.

Los Componentes Principales derivados de poblaciones multivariadas normales tienen interpretaciones útiles, aunque su obtención en general no requiere de asumir la distribución multivariada normal.

Considérese el vector aleatorio $X = (X_1, X_2, \ldots, X_N)^T$ con matriz de covarianza Σ. Si consideramos las combinaciones lineales definidas como

$$Y_1 = l_1^T X = l_{11}X_1 + l_{21}X_2 + \ldots + l_{N1}X_N$$

$$\vdots$$

$$Y_N = l_N^T X = l_{1N}X_1 + l_{2N}X_2 + \ldots + l_{NN}X_N.$$

Se puede demostrar que $Var(Y_i) = l_i^T \Sigma l_i$ y que $Cov(Y_i, Y_k) = l_i^T \Sigma l_k$. Recordemos que la matriz de covarianza Σ está definida como:

$$\Sigma = \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) & \ldots & Cov(X_1, X_N) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & \ldots & Cov(X_2, X_N) \\ \vdots & & \vdots & \\ Cov(X_N, X_1) & Cov(X_N, X_2) & \ldots & Cov(X_N, X_N) \end{bmatrix}.$$

Se define entonces el i-ésimo componente principal como la combinación lineal $l_i^T X$ que maximiza $Var(l_i^T X)$ sujeto a $l_i^T l_i = 1$ y $Cov(l_i^T X, l_k^T X) = 0$ para $k < i$.

A continuación se muestra un resultado importante que relaciona los Componentes Principales a la matriz de covarianza asociada a un vector aleatorio. La demostración constituye en si la derivación del método de PCA y se presenta a continuación.

Observación 2.0.1 *Sea Σ la matriz de covarianza asociada con el vector aleatorio $X = (X_1, X_2, \ldots, X_N)^T$. Tenga Σ los pares de eigenvalor-eigenvector (λ_1, v_1), (λ_2, v_2), ... ,(λ_N, v_N) donde $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$. El i-ésimo componente principal está dado como*

$$Y_i = v_i^T X.$$

Con estas elecciones: $Var(Y_i) = v_i^T \Sigma v_i = \lambda_i$ y $Cov(Y_i, Y_k) = v_i^T \Sigma v_k = 0$ para $i \neq k$.

Para derivar esta forma de los componentes principales, considérese primero $v_1^T X$; v_1 maximiza $Var(v_1^T X) = v_1^T \Sigma v_1$. Es claro que el máximo no será logrado para v_1 finito, por lo que una condición de normalización debe imponerse, en este caso será que v_1 sea de norma unitaria $v_1^T v_1 = 1$. Ahora, se tiene definido un problema de optimización con restricciones. El Lagrangiano resultante es de la forma

$$v_1^T \Sigma v_1 - \lambda(v_1^T v_1 - 1).$$

Considerando las condiciones de optimalidad, observamos que la diferenciación del Lagrangiano con respecto a v_1 e igualando a cero resulta en

$$\Sigma v_1 - \lambda v_1 = 0$$

o bien

$$\Sigma v_1 = \lambda v_1.$$

Luego entonces, λ es un eigenvalor de Σ y v_1 el correspondiente eigenvector. Nótese que la cantidad a maximizar es

$$v_1^T \Sigma v_1 = v_1^T \lambda v_1 = \lambda v_1^T v_1 = \lambda$$

por lo cual es necesario buscar λ tan grande como sea posible para obtener λ_1. Para encontrar el segundo componente principal v_2 nótese también que

$$Cov(v_1^T X, v_2^T X) = v_1^T \Sigma v_2 = v_2^T \Sigma v_1 = v_2^T \lambda_1 v_1 = \lambda_1 v_2^T v_1 = \lambda_1 v_1^T v_2.$$

Por lo que $v_2^T v_1 = 0$ es una de las restricciones posibles para indicar decorrelación entre $v_1^T X$ y $v_2^T X$. Entonces, el Lagrangiano para encontrar el segundo componente principal será

$$v_2^T \Sigma v_2 - \lambda_2 (v_2^T v_2 - 1) - \phi v_2^T v_1.$$

Diferenciando con respecto a v_2 e igualando a cero resulta en

$$\Sigma v_2 - \lambda_2 v_2 - \phi v_1 = 0. \tag{2.1}$$

Multiplicando la expresión anterior por v_1^T resulta en

$$v_1^T \Sigma v_2 - \lambda_2 v_1^T v_2 - \phi v_1^T v_1 = 0.$$

Es fácil notar que los dos primeros términos de la expresión anterior son iguales a cero y que $v_1^T v_1 = 1$. Por lo tanto, $\phi = 0$. Luego entonces, por (2.1)

$$\Sigma v_2 = \lambda_2 v_2,$$

dado que λ_2 debe ser lo más grande posible (con v_2 ortogonal a v_1) implica que es el segundo eigenvalor más grande de Σ. Este procedimiento puede continuarse para encontrar todos los componentes principales correspondientes a Σ.

Ahora bien, consideremos ya no el caso de una población, sino de m observaciones de N variables $\{x_i\}_{i=1}^m \in \Re^N$, mismas que representan observaciones independientes de una población N-dimensional con vector de media μ y matriz de Covarianza Σ. Estas observaciones tienen media muestral $\hat{\mu}$, con matriz de covarianza muestral C.

Si consideramos que las observaciones ya están centradas, esto es $\bar{x} = \sum_{j=1}^m x_j = 0$, podemos estimar Σ mediante $C = \hat{\Sigma}$ como

$$C = \frac{1}{m} \sum_{j=1}^{m} x_j x_j^T.$$

Así, encontramos que si C es la matriz de covarianza muestral con pares eigenvalor-eigenvector $(\hat{\lambda_i}, \hat{v_i})$, la i-ésima proyección sobre el Componente Principal i está dada por

$$\hat{y_i} = \hat{v_i}^T x$$

con x una observación de X y donde la varianza muestral de $\hat{y_i}$ es $\hat{\lambda_i}$ y la covarianza entre $\hat{y_i}$ y $\hat{y_k}$ es cero para $i \neq k$.

2.1. Selección del número de componentes

Es importante no ignorar la cuestión de cómo decidir cuántos componentes principales retener para mantener una porción aceptable de variabilidad en X. Generalmente, se adopta la idea de tratar de reducir la dimensionalidad del conjunto de datos reemplazando las N variables usando los primeros p componentes principales ($p < N$). Es posible que los últimos componentes principales posean virtudes que por esta elección sean ignoradas.

Uno de los criterios para obtener el número de componentes principales retenidos, p, es el porcentaje acumulativo de variación total. En éste se selecciona un porcentaje de la variación que los componentes deben retener, este porcentaje está dado por

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{N} Var(X_i)} \times 100 = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \times 100.$$

Otro criterio posible es la llamada regla de Kaiser, éste consiste en retener los componentes principales cuyas varianzas sean $\lambda_i \geq \bar{\lambda}$ donde $\bar{\lambda}$ es el promedio de las varianzas.

2.2. Aplicaciones de PCA

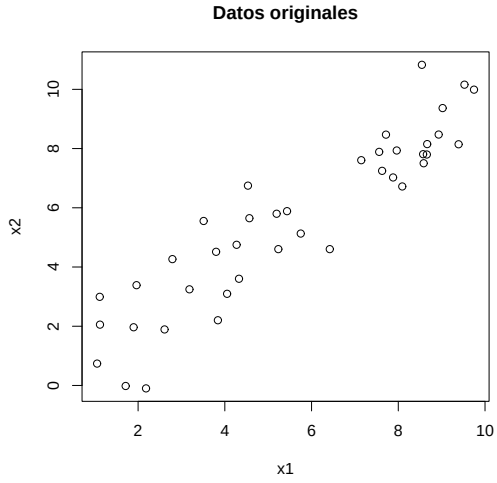

Figura 2-1: Datos generados de manera artificial para ilustrar PCA.

Un ejemplo ilustrativo sencillo es un conjunto de datos en dos dimensiones como el que se aprecia en la Figura (2-1), donde se puede ver la variabilidad de los datos en dos direcciones (con una dirección predominante).

Después de construir el estimador de la matriz de covarianza y de centrar apropiadamente los datos, encontramos sus eigenvalores-eigenvectores. Éstos nos arrojan

-como se vió en la sección anterior- las direcciones de máxima varianza si se toman las restricciones correspondientes. En la Figura (2-2) se graficaron las direcciones de los eigenvalores obtenidos de la matriz de covarianza. Se observa claramente que las direcciones en las que se encuentran son donde existe más variabilidad en los datos, siendo la dirección correspondiente al eigenvector con el eigenvalor más grande la que representa la mayor variabilidad en general.

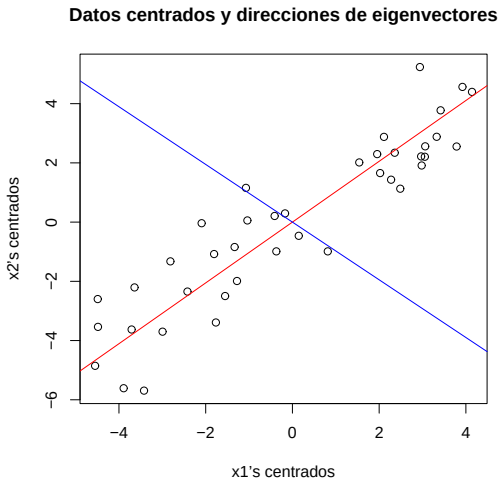

Figura 2-2: Dirección de los eigenvectores del estimador de la matriz de covarianza. En rojo se muestra el eigenvector con el mayor eigenvalor y en azul el siguiente. Obsérvese que el primer eigenvalor (en rojo) cruza sobre la mayor variabilidad en los datos.

En este punto, se puede hacer una reducción de dimensionalidad de los datos que orginalmente se encuentran en N dimensiones (aunque trabajando en $\mathbb{R}^2$ es claro que no tenemos problemas de dimensionalidad) y escoger solamente los $p \leq N$ Componentes Principales que correspondan a la mayor varianza (en este ejemplo el primer componente contiene el 70 % de variabilidad). Para este ejemplo tomamos todos, lo que implica hacer una rotación del sistema de coordenadas hacia las direcciones de mayor varianza. Los colocamos en lo que se conoce como un "vector de características"donde se encuentran los eigenvectores en cada una de las entradas de este vector. Para obtener los Componentes Principales, solo basta con proyectar nuestros datos originales en la dirección de los eigenvectores obtenidos. Como se mencionó, estamos conservando todos los eigenvectores, lo que se obtiene para este ejemplo es una rotación de los ejes de coordenadas hacia la dirección de más variabilidad indicada por los eigenvectores. Esto se ilustra en la Figura (2-3).

En problemas de compresión con pérdida, se escogen generalmente los Componentes que representarán la mayor variabilidad de los datos, si se quieren reconstruir los datos originales se observará que no corresponden exactamente a los originales debido a la eliminación de información (que se espera sea relativamente insignificante) producida por la eliminación de componentes. Sin embargo, cuando se toman

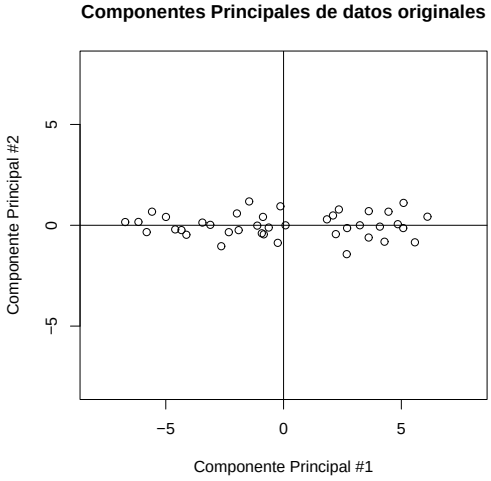

Figura 2-3: Obtención de Componentes Principales. Como se conservan todos los eigenvectores, los Componentes Principales representan una rotación de los ejes de coordenadas en las direcciones de más variabilidad.

todos los componentes como en este ejemplo, podemos obtener íntegramente los datos originales proyectando nuevamente los datos sobre el vector de características (considerando que debemos sumar la media de los datos que restamos originalmente). Lo que resulta entonces son datos que graficados son exactamente iguales a la Figura (2-1).

Aunque quizá este ejemplo no muestra el verdadero potencial de PCA, intenta ilustrar la idea general de lo que busca y como lo realiza. Otro ejemplo ilustrativo en el que efectivamente se realiza una disminución de dimensionalidad se puede apreciar en la Figura (2-4). En la Figura de la izquierda se aprecia un conjunto artificial en tres dimensiones, buscando una proyección interesante se puede notar que existe una combinación lineal en la que los datos prácticamente son constantes. Por lo tanto, realizar PCA sobre los datos y eliminar la dirección de menor variabilidad representaría eliminar la información que en cierto sentido es redundante, con ello, se gana en reducir la dimensión de los datos tal como se muestra en la Figura (2-5).

Un ejemplo usando datos reales es el que representa la Figura (2-6). En este ejemplo, a un conjunto de datos correspondiente a imágenes de dígitos escaneados y normalizados de dimensión 16×16 se le realiza PCA. Se consideraron únicamente las imágenes que representan al dígito 6 y el dígito 1. La Figura de la derecha muestra una gráfica del segundo componente principal contra el primero. Estos dos componentes retienen el 35.2 % por ciento de variabilidad en los datos. Si se quisiera clasificar entre las dos clases de dígitos usando, por ejemplo, una red neuronal, puede observarse que retener *solamente dos componentes* parece determinar muy bien la diferencia entre las dos clases. Si en determinada aplicación se tuvieran restricciones de memoria de almacenaje, una reducción de dimensionalidad de 256 a 2 sería aceptable.

¿Qué sucede cuando tenemos datos cuya dirección de máxima variabilidad es

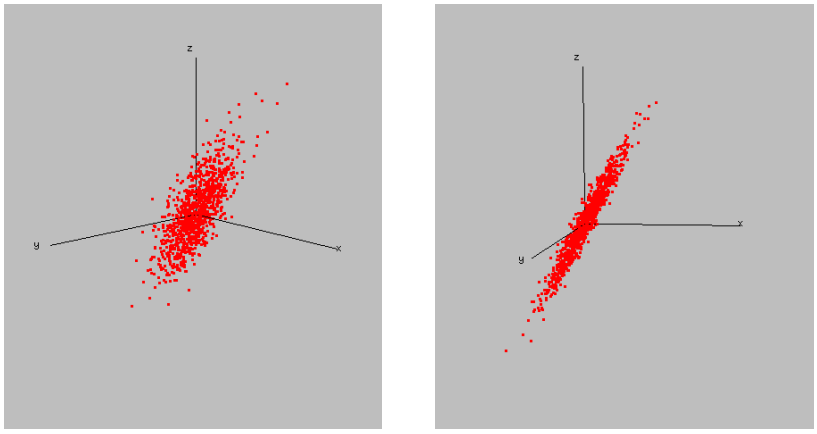

Figura 2-4: Un conjunto de datos artificial en 3 dimensiones. Buscando una proyección interesante se puede notar que existe una combinación lineal en la que los datos prácticamente son constantes.

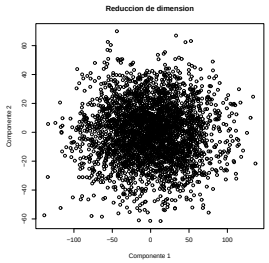

Figura 2-5: La reducción de dimensionalidad de los datos de tres a dos dimensiones correspondiente a los datos de la Figura (2-4).

nolineal, como en la Figura (2-7)? En este caso se puede observar que los datos se encuentran esparcidos en una forma que semeja quizá una curva cuadrática (como fueron creados artificialmente, este es efectivamente el caso).

Si se realiza un Análisis de Componentes Principales como en los ejemplos anteriores, se observará que las direcciones de máxima variabilidad no podrán captar la mayor variabilidad que se desearía obtener (que sería siguiendo una forma aproximadamente cuadrática), tal como en la Figura (2-8). Al hacer el mapeo a espacios no lineales con el Análisis de Componentes Principales mediante Kernels (Kernel PCA), se puede lograr lo que se busca.

Otra cuestión importante a considerar se ilustra en la Figura (2-9). Aquí, un conjunto de datos proveniente de una distribución normal con vector de media 0 y matriz de covarianza $diag(1{,}5, 0{,}5)$ se contamina por una pequeña proporción de observaciones provenientes de otra distribución. La Figura de la izquierda muestra las direcciones de los componentes principales *sin* los datos contaminantes, mientras que la Figura de la derecha muestra los efectos sobre los componentes principales que las

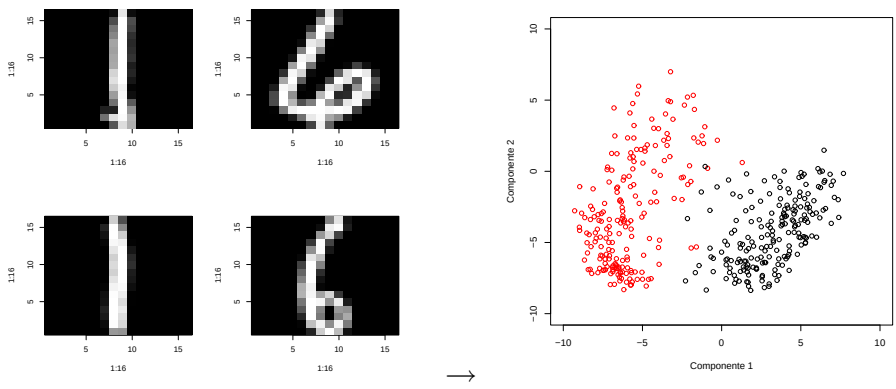

Figura 2-6: Para un conjunto de imágenes de dígitos de dimensión 16×16 se toman los primeros dos componentes principales.

observaciones contaminantes introducen. Obviamente, éste es un efecto que se desearía evitar. Para ello existe un importante número de trabajos cuyo fin es precisamente obtener soluciones como la Figura izquierda aún en la presencia de contaminación como la presente en la Figura derecha. Un ejemplo extremo es la Figura (2-10). En ésta, un solo dato que se introdujo en la muestra y que difiere drásticamente de la distribución causa una disrupción total en los componentes principales. Obviamente, no todas las ubicaciones geométricas tienen la misma influencia en los componentes principales, esta cuestión será tratada en el capítulo correspondiente a PCA Robusto.

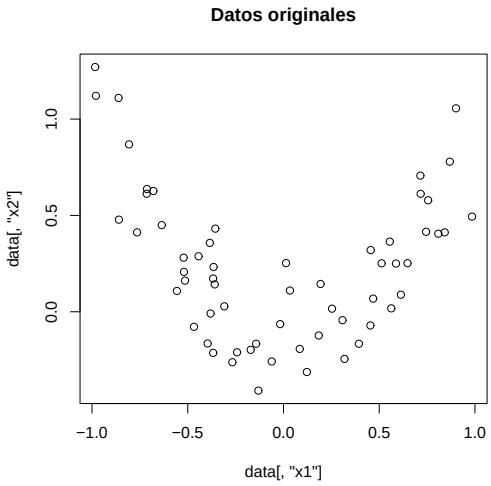

Figura 2-7: Datos generados artificialmente cuya dirección de máxima variabilidad pareciera ser nolineal.

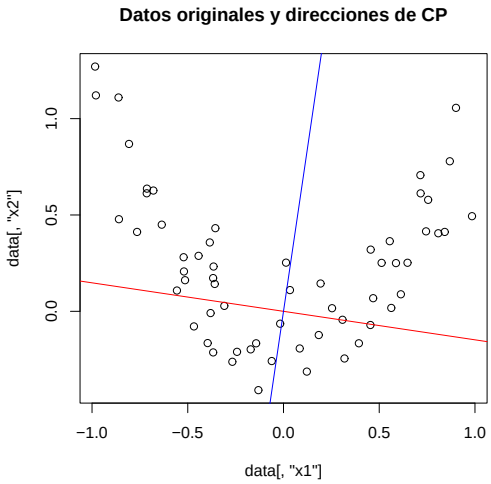

Figura 2-8: Dirección de los Componentes Principales obtenidos como en el ejemplo anterior. Sería deseable poder encontrar la dirección nolineal de mayor variabilidad.

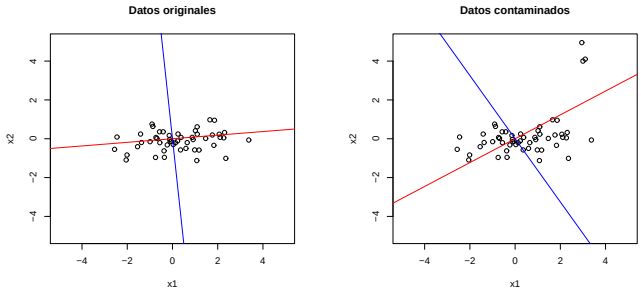

Figura 2-9: El efecto sobre los componentes principales al contaminar un conjunto de datos con outliers.

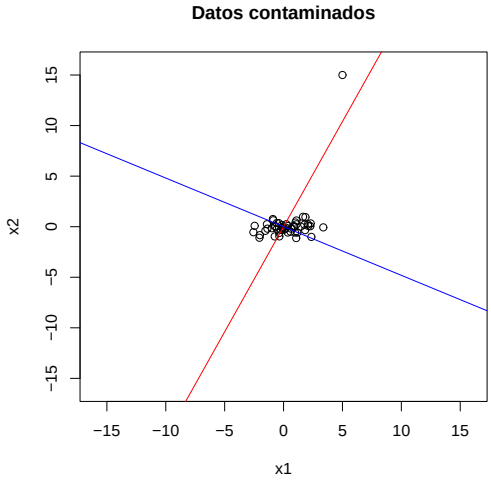

Figura 2-10: Un ejemplo extremo de la influencia de outliers en PCA. Una sola observación drásticamente atípica cambia drásticamente los componentes principales.

Capítulo 3

Kernel PCA

En estadística, la transformación de datos es algo natural cuando se piensa que un fenómeno puede ser explicado considerando nuevas relaciones. Ya sea utilizando conocimiento a priori o de manera empírica, es común pensar en transformar observaciones a espacios de diferente dimensionalidad que en la que se encuentran originalmente. Por ejemplo, en regresión suelen añadirse términos de interacción como

$$y \sim x \rightarrow y \sim x + x^2$$

en este caso a través de los residuales podríamos interpretar que la transformación se pudiera aproximar mejor al modelo que se estudia. Este caso se representa en la Figura (3-1). Esta transformación equivale a trabajar con $\{(y_i, x_i, x_i^2)\}$ en lugar de $\{(y_i, x_i)\}$.

En el Análisis de Componentes Principales mediante Kernels (Kernel PCA), se hace una transformación implícita de los datos y sobre estos datos transformados se realiza PCA. Puede ser que al transformar los datos se capture de mejor manera la variabilidad en el conjunto de datos que se está analizando.

En este capítulo se presenta el método basado en kernels para realizar Análisis de Componentes Principales. El método basado en kernels puede verse como una generalización del Análisis de Componentes Principales, donde se practica una técnica que ayuda a la simplificación del problema resultante de la generalización. Esta técnica consiste en la utilización del kernel. Puede verse esta generalización como una extensión del PCA a dominios no-lineales, y de donde la utilización del kernel ayuda a eliminar los inconvenientes de trabajar en el contexto nolineal.

3.1. Un ejemplo sencillo de métodos de kernel

Para ilustrar cómo funcionan los métodos de kernel en general, se presenta aquí un ejemplo de clasificación sencillo (tomado de [9]). Supóngase que se tiene un conjunto de datos de entrenamiento $\{(x_i, y_i)\}$ con $y_i \in \{-1, 1\}$, y se quiere clasificar un punto nuevo a la clase a la cual tenga menor distancia con la media de la clase. Esta

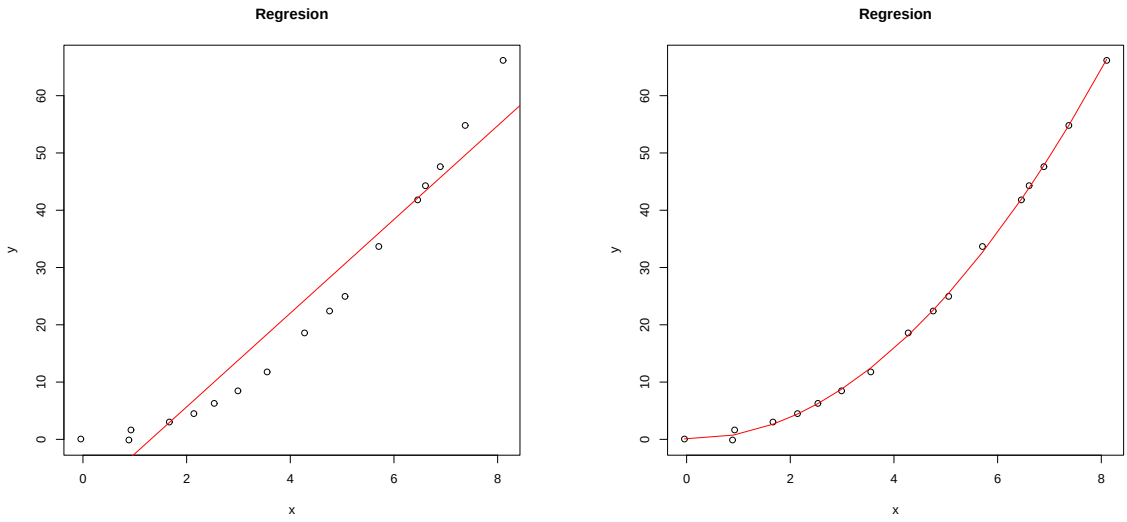

Figura 3-1: Una transformación de los datos en regresión agregando un término de interacción.

construcción geométrica puede ser formulada exclusivamente en términos del producto punto entre los datos como se verá a continuación.

Sean m_+ y m_- el número de datos en cada una de las clases, la media de cada clase es entonces

$$c_+ = \frac{1}{m_+} \sum_{i|y_i=+1} x_i,$$

$$c_- = \frac{1}{m_-} \sum_{i|y_i=-1} x_i.$$

Se asigna a un nuevo punto x la clase cuya media está más cercana como en la Figura (3-2). Esto lleva a considerar

$$\|c_- - x\|^2 \overset{?}{>} \|c_+ - x\|^2,$$

$$\|c_-\|^2 + \cancel{\|x\|^2} - 2\langle x, c_-\rangle \overset{?}{>} \|c_+\|^2 + \cancel{\|x\|^2} - 2\langle x, c_+\rangle,$$

$$\|c_-\|^2 - \|c_+\|^2 + 2\langle x, c_+\rangle - 2\langle x, c_-\rangle \overset{?}{>} 0,$$

$$y = signo(\langle x, c_+\rangle - \langle x, c_-\rangle + b), \tag{3.1}$$

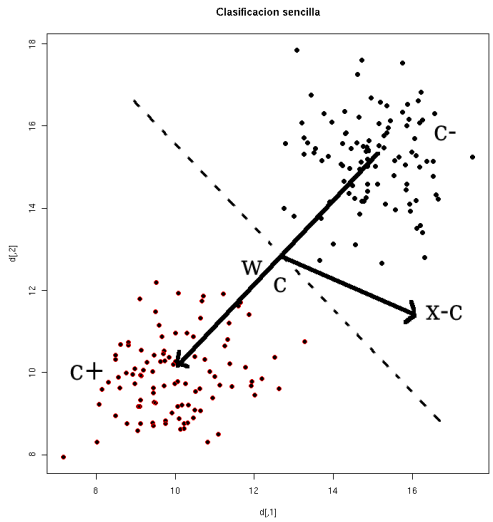

Figura 3-2: Ejemplo simple de clasificación. En la mitad del camino entre c_- y c_+ se encuentra el punto $c = (c_- + c_+)/2$. El vector $w = c_+ - c_-$ conecta las medias de las clases.

donde se ha definido

$$b = \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2)$$

con la norma $\|x\| = \sqrt{\langle x, x\rangle}$. Es instructivo reescribir la ecuación (3.1) en términos de los datos de entrenamiento x_i. Para ello, se susituyen en (3.1) las definiciones previamente mencionadas de c_- y c_+ con lo que se obtiene la función de decisión:

$$y = signo(\frac{1}{m_+}\sum_{i|y_i=+1}\langle x, x_i\rangle - \frac{1}{m_-}\sum_{i|y_i=-1}\langle x, x_i\rangle + b), \tag{3.2}$$

donde b es

$$b = \frac{1}{2}(\frac{1}{m_+^2}\sum_{(i,j)|y_i=y_j=+1}\langle x_i, x_j\rangle - \frac{1}{m_-^2}\sum_{(i,j)|y_i=y_j=-1}\langle x_i, x_j\rangle). \tag{3.3}$$

Cuando se tienen situaciones como la de la Figura (3-2) este algoritmo funciona razonablemente bien. Sin embargo, si se presenta una situación como la de la Figura (3-3) el algoritmo fracasa, puesto que en ese caso las medias prácticamente coinciden. Igual al ejemplo de regresión en el inicio de este capítulo, se puede llevar los datos de la Figura (3-3) a un espacio de dimensión mayor donde las medias de las clases no coincidan y el algoritmo pueda funcionar. Utilizar el *truco del kernel* en esta situación implica poder hacer tal transformación de forma indirecta. El requisito escencial para poder usar el truco del kernel es que un algoritmo pueda ser formulado de manera tal que los datos solamente aparezcan en forma de productos punto (como es el caso de

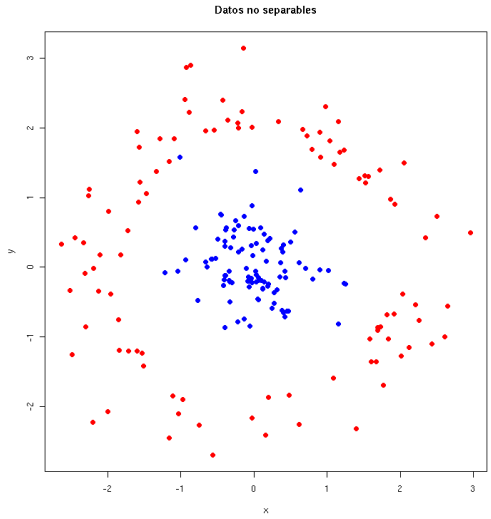

Figura 3-3: Datos no separables mediante el algoritmo de clasifiación que considera las medias de las clases.

este ejemplo). Entonces, todas las ocurrencias de los productos punto se sustituyen por un *kernel*. De manera informal, un kernel es una función que regresa el valor del producto punto entre las imágenes de los dos argumentos:

$$k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle .$$

Dada una funcion k, es posible verificar si es un kernel. Como ilustración considérese el *kernel polinomial*:

$$k(x, y) = (\gamma \langle x, y \rangle + c)^d, \gamma > 0. \tag{3.4}$$

Se puede demostrar que este kernel corresponde a un mapeo Φ a un espacio de características que es generado por todos los productos de d entidades de un patrón de entrada, por ejemplo, para $N = 2,\ d = 2,\ c = 0$ y $\gamma = 1$:

$$(x \cdot y)^2 = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2,$$

$$(x \cdot y)^2 = (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \cdot (y_1^2, \sqrt{2} y_1 y_2, y_2^2) = \Phi(x) \cdot \Phi(y)$$

el espacio implícito es $\Phi :\ (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$. Así, sustituir las ocurrencias de productos punto por un kernel corresponde a realizar *una transformación implícita de los datos* al espacio Φ. Para el ejemplo de clasificación con medias, se puede sustituir en (3.2) y (3.3)

$$\langle x, y \rangle \rightarrow k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle ,$$

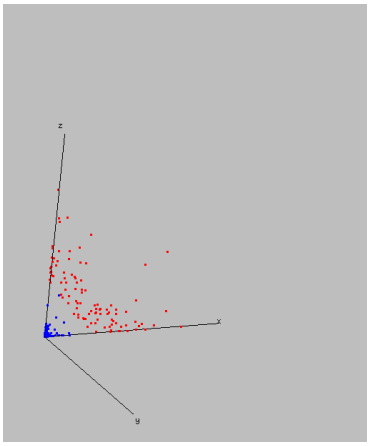

Figura 3-4: Los datos de la Figura (3-2) de pueden transformar a un nuevo espacio donde el algoritmo de clasificación mediante medias pueda funcionar. Usando el truco del kernel se puede hacer esta transformación de manera implícita.

lo cual implica hacer la transformación implícita determinada por el kernel. Utilizando el kernel polinomial de grado 2 (3.4) con $c = 0$ los datos de la Figura (3-3) se mapean a un espacio representado por la Figura (3-4). En esta nueva situación, el simple algoritmo de clasificación aquí presentado recupera su utilidad.

3.2. Formulación del método Kernel PCA

Kernel PCA es el resultado de aplicar el truco del kernel al Análisis de Componentes Principales (PCA). Sabemos por lo presentado en capítulos anteriores que PCA es una transformación de base para diagonalizar (encontrar eigenvalores-eigenvectores) del estimador de la matriz de covarianza de los datos x_k, $k = 1, \ldots, m$, $x_k \in \mathbb{R}^N$. La matriz de covarianza se puede estimar como

$$C = \frac{1}{m} \sum_{j=1}^{m} (x_j - \hat{\mu})(x_j - \hat{\mu})^T$$

y la media como

$$\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x_i.$$

Las nuevas coordenadas en la base de eigenvectores (las proyecciones ortogonales) son los Componentes Principales. Con Kernel PCA, se pretende generalizar este contexto a uno nolineal de la forma siguiente: supóngase que se hace el mapeo de los datos de forma nolineal a un *espacio de características* $\mathcal{H}$ mediante

$$\Phi : \mathbb{R}^N \to \mathcal{H}, \; x \mapsto \Phi(x).$$

El método se basa en que para ciertas elecciones de Φ, aún si $\mathcal{H}$ tiene una dimensionalidad arbitrariamente grande, aún se puede realizar PCA en $\mathcal{H}$.

Supóngase que los datos mapeados al espacio de características $\Phi(x_1), \ldots, \Phi(x_m)$ están centrados, es decir, $\sum_{k=1}^{m} \Phi(x_k) = 0$ (suposición que más tarde será eliminada). Para hacer PCA cuando la estimación de la matriz de covarianza es

$$C = \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j)\Phi(x_j)^T \tag{3.5}$$

se debe encontrar eigenvalores $\lambda \geq 0$ y eigenvectores $v \in \mathcal{H}$ que satisfagan

$$\lambda v = Cv. \tag{3.6}$$

Se puede observar que debido a (3.5) y (3.6) todas las soluciones v están en el espacio generado por $\Phi(x_1), \ldots, \Phi(x_m)$:

$$v = \frac{1}{\lambda m} \sum_{j=1}^{m} \Phi(x_j) \ \left\langle \Phi(x_j), v \right\rangle .$$

Esto es equivalente a decir que existen coeficientes $\alpha_1, \ldots, \alpha_m$ tales que

$$v = \sum_{i=1}^{m} \alpha_i \Phi(x_i). \tag{3.7}$$

Ahora bien, gracias a lo anterior se puede considerar el siguiente sistema equivalente a (3.6) (ver el Apéndice):

$$\lambda(\Phi(x_k) \cdot v) = (\Phi(x_k) \cdot Cv). \tag{3.8}$$

para todo $k = 1, \ldots, m$.

Sustituyendo en la expresión (3.8) las expresiones (3.5) y (3.7):

$$\lambda(\Phi(x_k) \cdot \sum_{i=1}^{m} \alpha_i \Phi(x_i)) = (\Phi(x_k) \cdot \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j)\Phi(x_j)^T \sum_{i=1}^{m} \alpha_i \Phi(x_i))$$

$$\lambda \sum_{i=1}^{m} \alpha_i \left\langle \Phi(x_k), \Phi(x_i) \right\rangle = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \left\langle \Phi(x_k), \sum_{j=1}^{m} \Phi(x_j) \left\langle \Phi(x_j), \Phi(x_i) \right\rangle \right\rangle$$

para todo $k = 1, \ldots, m$.

Definiendo la matriz K de Gram, de $m \times m$

$$K_{ij} = (\Phi(x_i) \cdot \Phi(x_j)) \tag{3.9}$$

y acomodando términos se llega a:

$$m\lambda K\alpha = K^2\alpha \tag{3.10}$$

donde α denota el vector columna con entradas $\alpha_1, \ldots, \alpha_m$. Para encontrar las soluciones de (3.10) se resuelve el problema de eigenvalores dado por

$$m\lambda\alpha = K\alpha$$

porque todas las soluciones de interés para la expresión anterior satisfacen (3.10) (ver [9]).

Con esto, se puede encontrar los eigenvectores $\{v^k\}$ en $\mathcal{H}$ a partir de la combinación lineal dada por (3.7). Para obtener eigenvectores normalizados en $\mathcal{H}$ se requiere que $v^k \cdot v^k = 1$. Esto se traduce que α debe cumplir:

$$1 = \sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k (\Phi(x_i) \cdot \Phi(x_j)) = (\alpha^k \cdot K\alpha^k) = \lambda_k(\alpha^k \cdot \alpha^k). \tag{3.11}$$

Para obtener los componentes principales en $\mathcal{H}$, se calcula la proyección de un punto $(\Phi(x)$ en los eigenvectores v^k mediante

$$(v^k \cdot \Phi(x)) = \sum_{i=1}^{m} \alpha_i^k (\Phi(x) \cdot \Phi(x_i)). \tag{3.12}$$

Ahora bien, nótese que ni en (3.9) ni en (3.12) se requiere $\Phi(x)$ en forma explícita, sino solo en forma de productos punto. Entonces, la propuesta es utilizar funciones de kernel para calcular estos productos punto sin realizar el mapeo Φ.

Así, sustituyendo con kernels todas las ocurrencias de $\Phi(x) \cdot \Phi(y)$ se llega al siguiente algoritmo:

Algoritmo Kernel PCA:

- Calcular la matriz de Gram K (3.9)
- Diagonalizar K (encontrar $K = V\Lambda V^T$, donde V es la matriz que tiene como columnas los eigenvectores de K y Λ es una matriz diagonal con los correspondientes eigenvalores)
- Normalizar los coeficientes α de expansión de eigenvectores v (usando (3.11))
- Extraer l componentes principales, calculando las proyecciones de los datos sobre los primeros l eigenvectores (usando (3.12))

El supuesto bajo el que trabaja este algoritmo es el uso del estimador (3.5).

Hasta este punto, se ha supuesto que los $\Phi(x_i)$ están centrados en el espacio $\mathcal{H}$. Como no se puede en general centrar los datos (no se puede calcular la media de un

conjunto de datos que no se tiene en forma explícita) se siguen los pasos anteriores usando $\tilde{\Phi}(x) = \Phi(x) - (1/m)\sum_{i=1}^{m}\Phi(x_i)$. Los elementos de la matriz de Gram son ahora

$$\widetilde{K_{ij}} = \tilde{\Phi}(x_i)\cdot\tilde{\Phi}(x_j) = (\Phi(x_i) - \frac{1}{m}\sum_{r=1}^{m}\Phi(x_r))\cdot(\Phi(x_j) - \frac{1}{m}\sum_{s=1}^{m}\Phi(x_s))$$

$$\widetilde{K_{ij}} = (\Phi(x_i)\cdot\Phi(x_j) - \frac{1}{m}\sum_{r=1}^{m}\Phi(x_j)\cdot\Phi(x_r) - \frac{1}{m}\sum_{s=1}^{m}\Phi(x_i)\cdot\Phi(x_s) +$$

$$+\frac{1}{m^2}\sum_{r=1}^{m}\sum_{s=1}^{m}\Phi(x_r)\cdot\Phi(x_s)).$$

Definiendo la matriz 1_m de $m \times m$ cuyas entradas son todas $1/m$, se llega a que la matriz que se debe diagonalizar (en lugar de K) para el algoritmo de Kernel PCA se puede expresar en términos de K misma como

$$\widetilde{K_{ij}} = K - 1_m K - K 1_m + 1_m K 1_m.$$

Cabe señalar que el uso de esta matriz de Gram centrada (y el respectivo kernel) se encuentra justificado por la Proposición (A.2.1) que se puede encontrar en el Apéndice A.

Ejemplos de kernels

Para los fines de esta exposición, la definición informal de kernel dada en secciones anteriores es suficiente. En el Apéndice A se puede encontrar la definición formal junto con algunas propiedades.

Como ejemplo de kernels usados en la literatura, podemos encontrar

- El kernel polinomial mencionado anteriormente: $k(x,y) = \gamma(x\cdot y + c)^d$
- Kernel de base radial: $k(x,y) = exp(-\|x-y\|^2/(2\sigma^2))$
- Kernel sigmoide: $k(x,y) = tanh(\kappa(x\cdot y) + \Theta)$

Un ejemplo de aplicación de Kernel PCA

Para entender el siguiente ejemplo, es necesario notar que en general no se puede graficar directamente la dirección de los componentes principales v^k (tal como en los ejemplos de capítulos anteriores) cuando se trabaja con datos en el plano (x,y). Ésto se debe a que en general se encuentran en un espacio de diferente dimensionalidad a los datos. El procedimiento gráfico involucra entonces graficar las proyecciones de una región del plano (x,y) y resaltar los puntos en los que las proyecciones sobre v^k

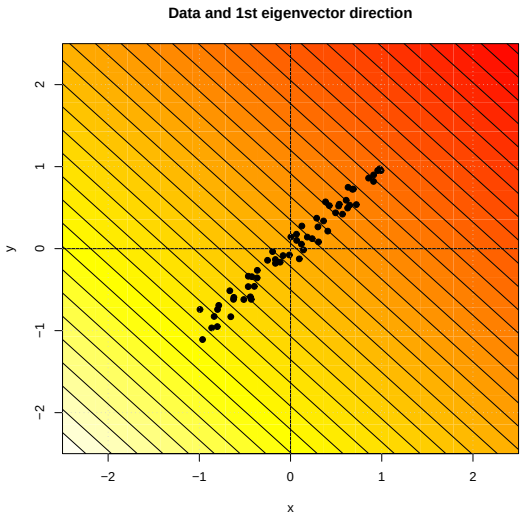

Figura 3-5: Kernel PCA para la transformación idéntica. Las líneas que muestran valores constantes de proyección sobre los PC indican la misma variación lineal que en el caso clásico.

tiene un valor constante. Es decir, se toma un conjunto de puntos equiespaciados en el plano y se proyectan sobre el k-ésimo componente principal, ésto corresponde a una función que va de $\Re^2$ a $\Re$; las curvas de nivel de esta función ayudan a visualizar los componentes principales. Lo anterior se puede aclarar si se considera la Figura (3-5), donde se utiliza un kernel polinomial de grado $d = 1$, mismo que corresponde a la transformación identidad y a realizar PCA clásico. Las líneas que muestran valores constantes de proyección sobre el primer PC (curvas de nivel) indican la misma variación lineal que en el caso clásico. Se podría realizar PCA a estos datos y el primer CP estaría en dirección ortogonal a las líneas que indican proyección constante.

Así, la utilidad de Kernel PCA se ejemplifica en los datos de la Figura (3-6). En ésta se grafican, usando el procedimiento descrito, los primeros dos componentes principales del conjunto de datos obtenidos al utilizar un kernel polinomial de grado $d = 2$. El es idéntico a hacer la transformación

$$(x_1, x_2) \rightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

y realizar PCA clásico sobre los datos transformados. Como puede imaginarse, el resultado que se obtiene depende fuertemente de la elección del kernel (y por ende de la transformación implícita que se hace sobre los datos).

Ahora bien, el tema de esta tesis tiene que ver directamente con lo que se manifiesta en la Figura (3-7). Aunque la interpretación del efecto de outliers sobre los componentes principales nolineales es más complicada, se puede apreciar también el efecto indeseable que una sola observación puede tener. La Figura de la izquierda muestra un conjunto de datos en donde todas las observaciones siguen la misma es-

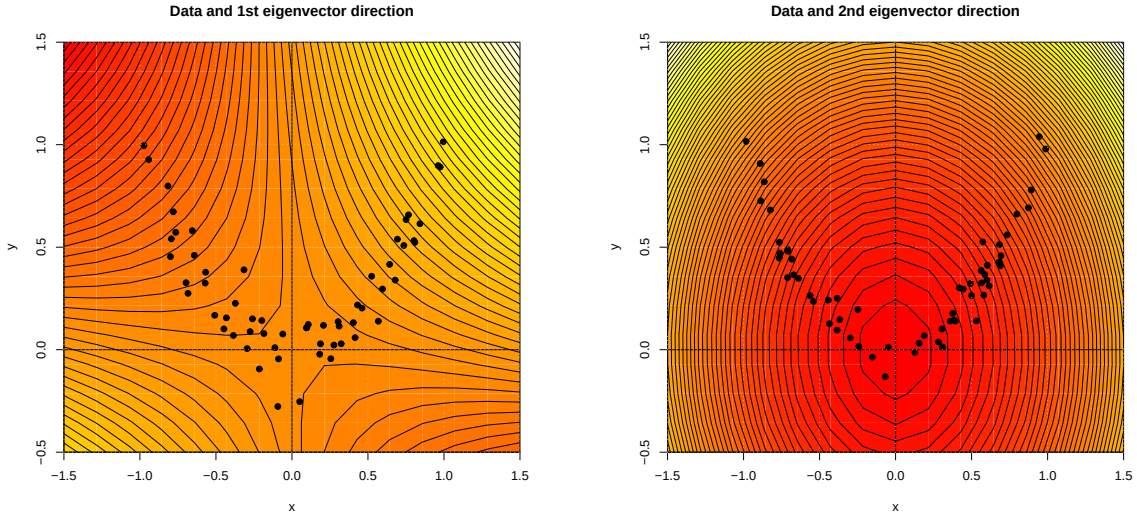

Figura 3-6: Para un conjunto de datos artificial, se realiza Kernel PCA con un kernel polinomial de grado $d = 2$. La Figura de la izquierda es la proyección del primer componente principal, y la Figura de la derecha la proyección sobre el segundo componente.

tructura, mientras que la Figura de la derecha muesta una observación en el extremo derecho que afecta drásticamente las proyecciones sobre el primer componente principal. En el caso de la transformación de kernel idéntica, se obtiene *exactamente* el mismo comportamiento frente a outliers en PCA y Kernel PCA como era de esperarse. No es de extrañarse que Kernel PCA sea sensible a datos atípicos, ya que en su formulación se considera un estimador de la matriz de covarianza que sufre de esa sensibilidad. La reformulación de Kernel PCA incluyendo elementos de PCA Robusto es el tema del capítulo central de esta tesis y se presentará más adelante. Mientras tanto, conviene mencionar algunos otros aspectos de Kernel PCA.

3.2.1. El problema de la preimagen

Usar un kernel k en lugar de un producto punto en el espacio de entrada corresponde a mapear implícitamente los datos a un espacio con producto punto $\mathcal{H}$ con el mapa $\Phi : \mathcal{X} \to \mathcal{H}$ y tomar el producto punto ahí. El precio pagado por esta elegancia es que todas las soluciones son obtenidas como expansiones en términos de los datos de entrada (de entrenamiento). Recuérdese que en Kernel PCA todas las soluciones cumplen que

$$v = \sum_{j=1}^{m} c_i \Phi(x_j).$$

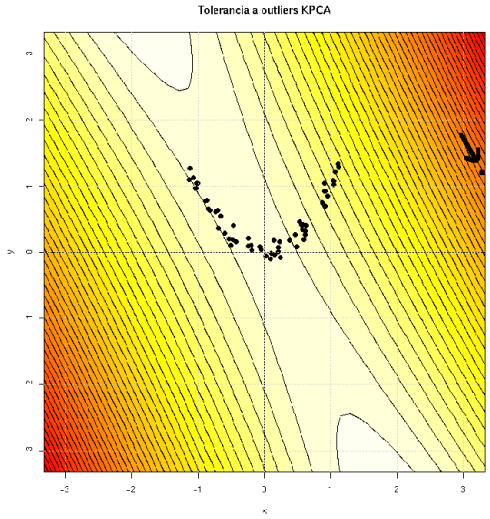

Figura 3-7: Aunque la interpretación del efecto de outliers sobre sobre los componentes principales es más complicada, se puede apreciar también el efecto indeseable que una sola observación puede tener.

Como se quiere evaluar las proyecciones sobre estas soluciones, se obtiene la expansión $\sum_i \alpha_i k(x_i, x)$ que puede ser evaluada sin importar la dimensión de v.

Un problema que se presenta en los métodos de kernel - y específicamente en Kernel PCA - es el de encontrar *preimágenes*. Por encontrar una preimagen, nos referimos al problema de encontrar $z \in \Re^N$ dado un Ψ tal que $\Phi(z) = \Psi$. Debido a que el mapa Φ puede ser arbitrario y nunca se realiza de manera explícita, en general debemos conformarnos con encontrar z tal que

$$\rho(z) = \|\Psi - \Phi(z)\|^2$$

sea pequeño[1]. Así, de muchas aplicaciones que en PCA son fácilmente aplicables como eliminación de ruido y compresión se tiene que en Kernel PCA involucran un problema de optimización. El problema de la preimagen es entonces una de las desventajas que involucra utilizar Kernel PCA.

3.2.2. Kernel PCA en el contexto del aprendizaje máquina

En el *aprendizaje supervisado*, se busca predecir los valores de una o más variables de respuesta $Y = (Y_1, \ldots, Y_m)$ para un conjunto dado de variables predictoras $X = (X_1, \ldots, X_p)$. El conjunto de entrenamiento son las parejas $(x_1, y_1), \ldots, (x_n, y_n)$ donde y_i es la respuesta i-ésima al predictor $x_i = (x_{i1}, \ldots, x_{ip})$. Un método correspondiente al área de aprendizaje supervisado tal como clasificación presenta una respuesta $\hat{y}_i$ para cada x_i; se puede entonces juzgar el desempeño observando el error asociado con las predicciones del método supervisado. Tal error se puede cuantificar mediante una

[1]Pequeño en este contexto depende del problema en cuestión que se esté resolviendo.

función de error $L(y, \hat{y})$. Es entonces fácilmente cuantificable el grado de éxito que se tiene al desarrollar un nuevo método en esta área.

Según Hastie ([3]), el aprendizaje supervisado se puede caracterizar formalmente como un problema de estimación de densidades. Para entender este enfoque, supóngase a (X, Y) como variables aleatorias con una correspondiente distribución conjunta $P(X, Y)$, el problema de aprendizaje es determinar propiedades de interés de la densidad condicional $P(X|Y)$. Estas propiedades pueden ser por ejemplo parámetros μ que minimicen el error esperado en cada x

$$\mu(x) = \min_{\theta} E_{Y|X} L(Y, \theta)$$

donde L puede ser por ejemplo $L(y, \hat{y}) = (y - \hat{y})^2$.

Para el caso de *aprendizaje no-supervisado* - al que corresponde Kernel PCA -, se cuenta con N observaciones $(x_1, x_2, \ldots, x_N)$ de un vector aleatorio X, mismo que tiene una densidad conjunta $P(X)$. Lo que se quiere es encontrar propiedades de su densidad, pero en este caso sin contar con variables de respuesta como y_i en el caso supervisado. Surge aquí una distinción de suma importancia entre el aprendizaje supervisado y no-supervisado: en el caso supervisado - como se mencionó anteriormente - existe la posibilidad de cuantificar el éxito o fracaso de un método y existe por ende una manera objetiva de juzgarlo; para el caso no-supervisado no existe tal medida de éxito o fracaso, por lo cual determinar la utilidad de los diversos métodos es un asunto complicado.

Capítulo 4

PCA Robusto

PCA es una técnica altamente sensible a observaciones atípicas (outliers). A sabiendas de este hecho, se han realizado trabajos cuyo objetivo es precisamente superar la sensibilidad del PCA hacia estas observaciones atípicas. En este capítulo se pretende presentar un recuento conciso de los métodos más conocidos para el caso específico cuando se quiere realizar PCA robusto. Existen excelentes trabajos sobre estadística robusta en general, al lector interesado en métodos robustos en general se le refiere a Huber ([4]) y Hampel ([2]).

4.1. Estadística Robusta

Continuamente se presentan procedimientos estadísticos en los cuales los supuestos establecidos no se cumplen (como por ejemplo suponer independencia o cierto modelo de distribución). Huber ([4]) define 'robustez' como *'insensibilidad a pequeñas desviaciones de los supuestos'*. Por su parte Hampel ([2]) afirma que *'en un sentido informal, la estadística robusta es un cuerpo de conocimiento, parcialmente formalizado en* teorías de robustez*, que tienen que ver con desviaciones de supuestos idealizados en estadística'*.

En este sentido, se busca fundamentalmente salvaguardarse de las situaciones donde los supuestos resultan no cumplirse. Como es el caso, por ejemplo, cuando existen errores drásticos producto de situaciones tales como transmitir o capturar incorrectamente el valor de una observación de un conjunto de datos. Es decir, se busca evitar que una fracción pequeña de las observaciones afecten seriamente la calidad de los estimadores. En un sentido más amplio, la robustez distribucional tiene que ver desviaciones leves de una distribución supuesta (usualmente la distribución multivariada normal); lo que se busca entonces es que la inferencia estadística de los parámetros de interés no se afecte por las observaciones que no provengan de la distribución idealizada. Se puede pensar que el grupo de datos con el que se trabaja proviene de una distribución $Q(x;\theta_1,\theta_2)$ conformado por la mezcla de una distribución idealizada $R(x;\theta_1)$ y una distribución contaminante $S(x;\theta_2)$:

$$Q(x;\theta_1,\theta_2) = (1-\epsilon)R(x;\theta_1) + \epsilon S(x;\theta_2)$$

donde $0 \leq \epsilon \leq 1$ y ϵ representa el nivel de contaminación (si el tamaño de la muestra es m, $m(1-\epsilon)$ de los datos provienen de $R(x;\theta_1)$ y ϵm datos provienen de $S(x;\theta_2)$).

En ambos casos, las observaciones que divergen de los supuestos se conocen como *atípicas*, *contaminantes*, o como se les denomina en inglés *outliers* (dicha denominación constituye el grueso en la literatura y por ende perdominará en este trabajo).

Basándonse en ([2]) y ([4]), se espera de un procedimiento robusto:

1. Debe describir la estructura que mejor ajusta al grueso de los datos.
2. Desviaciones pequeñas de los supuestos estadísticos deben producir solamente desviaciones pequeñas en el desempeño del método.
3. Desviaciones grandes no deben causar una catástrofe.
4. Se debe poder identificar outliers para un tratamiento posterior si éste se desea realizar.

De las características anteriores, el punto número 4 define el enfoque que se busca en este trabajo. Se puede ejemplificar diciendo que si se presenta la distribución $Q(x;\theta_1,\theta_2)$, se pretende modelar el grueso de los datos considerando que $Q(x;\theta_1,\theta_2) \approx R(x;\theta_1)$ y que el resto de los ϵ por ciento datos generados por $S(x;\theta_2)$ deben tener un peso menor. Esto en contraste con una postura definida por modelar solamente $R(x;\theta_1)$ desechando todos los ϵ por ciento de las observaciones provenientes de $S(x;\theta_2)$. Se considera que las observaciones atípicas consituyen fuente de información, a través de la historia se han presentado casos en los cuales un análisis de outliers ha llevado a descubrimientos trascendentes. Inclusive, es práctica común querer determinar cuando se presenta un outlier en aplicaciones tecnológicas, tal como se menciona en ([11]).

4.1.1. Punto de quiebre de un estimador

Un concepto importante en estadística robusta es el punto de quiebre de un estimador. Se define como la proporción de los datos necesario para hacer que un estimador tenga un valor infinito. Es decir, es un número que indica que porcentaje de datos en una muestra se necesitan para cambiar arbitrariamente la estimación. Por ejemplo, la media tiene un punto de quiebre de $1/m$, puesto que solamente es necesaria una observación para hacer que el valor de la media se vuelva arbitrariamente cercano a infinito. Por otro lado, la mediana tiene un punto de quiebre de $1/2$, puesto que es necesario que la mitad de los datos cambien su valor para hacer que la mediana se mueva arbitrariamente cerca de $+/-\infty$.

4.1.2. Conceptos de M-estimadores y función de influencia

Un M-estimador es un estimador T_m definido mediante

$$min_{T_m} \sum_{i=1}^{m} \rho(x_i; T_m) \tag{4.1}$$

o mediante la ecuación implícita

$$\sum_{i=1}^{m} \psi(x_i; T_m) = 0, \tag{4.2}$$

donde $\psi(x_i; T_m) = \frac{\partial \rho(x; T_m)}{\partial T_m}$ y ρ es una función derivable arbitraria. El nombre de M-estimador proviene de *máxima verosimilitud generalizada*, ya que es fácil notar que si $\rho(x; T_m) = log\ f(x; T_m)$, T_m es entonces el estimador de máxima verosimilitud. El estimador no se modifica si $\psi(x_i; T_m)$ es multiplicado por una constante positiva.

Para los M-estimadores existe la propiedad de que la función ψ que los origina es proporcional a otra llamada *función de influencia*. Dicha función indica la sensibilidad de un estimador T ante la presencia de observaciones atípicas y también sirve para calcular la varianza asintótica de T. Si se conoce la función de influencia de un estimador dado se puede analizar la robustez frente a outliers. Así, mide cómo cambia el estimador bajo contaminaciones de la verdadera distribución. Por la proporcionalidad entre función de influencia y ψ, se puede proceder primero proponiendo una función de influencia y después encontrando el M-estimador respectivo, o proponiendo directamente ψ (o inclusive proponiendo directamente ρ).

Formalmente, Huber ([4]) define la función de influencia como:

$$FI(x; T, F) = lim_{\epsilon \to 0} \frac{T[(1-\epsilon)F + \epsilon\delta_x] - T(F)}{\epsilon},$$

donde $0 \geq \epsilon \geq 1$ si es que el límite existe. F es considerada como la verdadera distribución de los datos y δ_x como la distribución contaminante (que a su vez es degenerada - asigna todo la masa de probabilidad en x). T es un funcional del espacio de funciones de distribución al espacio de parámetros, donde se tiene que $T_m \to T(F)$.

4.2. Métodos Robustos para PCA

Tipos de outliers en PCA

Antes de proceder con el recuento de técnicas para PCA Robusto, cabe señalar que en PCA la influencia de las observaciones depende de su colocación geométrica con respecto al centroide de datos. La Figura (4-1) muestra los componentes principales robustos para un conjunto de datos junto con un *plot* diagnóstico ([5]) donde en el eje horizontal se indica el *score distance* de cada observación, definido como

$$SD_j = \sqrt{\sum_{i=1}^{m} \frac{t_{ji}^2}{\lambda_i}}$$

donde t_{ji} es la proyección del dato j-ésimo en el i-ésimo componente y λ_i es la variación explicada también por el i-ésimo componente. En el eje vertical se grafica la distancia ortogonal de cada observación al subespacio generado por PCA (en la Figura (4-1) se tomó este espacio generado como de dimensión 1). Las observaciones más dañinas

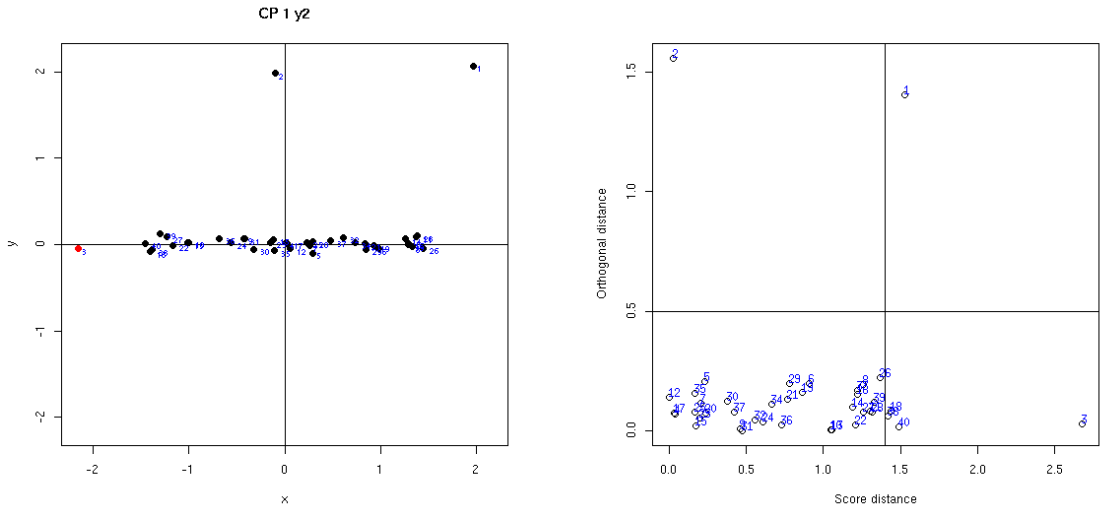

Figura 4-1: Clasificación de outliers en el caso de PCA. En la figura izquierda, se realizó PCA de manera robusta; la figura derecha es un *plot* diagnóstico para clasificar los outliers según su posición.

son las que en este tipo de gráficos diagnóstico se encuentran en la región superior derecha.

La distancia de Mahalanobis

En las técnicas robustas para PCA, muchas usan una forma u otra de la *distancia de Mahalanobis* entre una observación x_i y la media μ:

$$d_i = \sqrt{(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)} \tag{4.3}$$

donde Σ es la matriz de covarianza.

La distancia de Mahalanobis tiene como caso particular a la distancia Euclidiana. A diferencia con la distancia Euclideana, para la distancia de Mahalanobis, se toman en cuenta las correlaciones entre las variables, cuando todas las variables tienen varianza igual a uno y no existe correlación entre ellas, la distancia de Mahalanobis coincide con la Euclideana.

La distancia de Mahalanobis de una observación a la media o centroide de la nube de puntos se utiliza en la detección de observaciones atípicas, puesto que por lo general este tipo de observaciones suelen estar lejos del centroide (y sobre direcciones de baja correlación) cuando se tiene (o supone) simetría multivariada elipsoidal.

La Figura (4-2) muestra, para un conjunto de observaciones provenientes de una normal multivariada, los valores constantes tanto de la distancia Euclidiana como de la de Mahalanobis. Nótese como la distancia de Mahalanobis toma en consideración la correlación entre las variables, en este ejemplo los valores constantes son elipses a diferencia de círculos concéntricos en el caso de la distancia Euclideana.

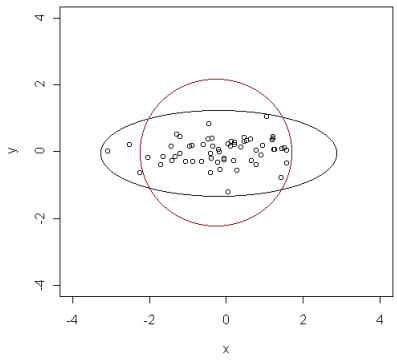

Figura 4-2: Valores constantes de la distancia Euclideana (rojo) y Mahalanobis (negro) para una distribución multivariada normal. La distancia de Mahalanobis toma en cuenta la correlación entre las variables.

4.2.1. Submuestreo de los datos

El primer método robusto para PCA que se considera es el submuestreo de los datos, mismo que es una opción directa, rápida y sencilla. Sin embargo, no hay garantía alguna de que las observaciones atípicas (de exisitir) sean eliminadas del conjunto submuestreado. Además, no son raras las ocasiones en que el tamaño de la muestra hace que un submuestreo resulte prohibitivo. El punto de quiebre será proporcional a el tamaño de la muestra removido al submuestrear.

4.2.2. Estimación robusta de la matriz de covarianza

Los métodos para obtener estimaciones robustas de la matriz de covarianza tienen dos principales vertientes: la obtención de los elementos individuales de la matriz y la estimación de la misma usando un subconjunto de las observaciones disponibles. Al usar estos métodos, los componentes principales se pueden obtener de la eigendescomposición de la matriz de covarianza obtenida de forma robusta.

Métodos de componentes individuales

Un grupo de técnicas son las conocidas como métodos individuales. En éstos, se estima de manera independiente cada elemento de la matriz de covarianza o correlación. Por ejemplo, se puede tratar de obtener una matriz de correlación estimando de manera robusta cada coeficiente de correlación de manera separada, utilizando cualquier información disponible. En el caso de matrices de covarianza, se pueden utilizar estimadores truncados para cada desviación estándar y con ayuda de las correlaciones robustas se puede obtener una estimación de la matriz de covarianza.

Una de las ventajas de este tipo de método es su poco tiempo de cómputo necesario, así como tener un punto de quiebre alto (que por lo general se determina por el

número de muestras truncadas en cada iteración). Como desventaja, se puede mencionar que la matriz de covarianza resultante puede no ser (semi) positiva definida, y en ese caso suele aplicarse trucos como multiplicar por algún factor que asegure consistencia.

Truncamiento Multivariado

De todos los procedimientos multivariados, quizá el más sencillo sea el Truncamiento Multivariado. Este método usa la distancia de Mahalanobis y es sensible a los valores iniciales.

Para este método se evalua para cada observación la distancia de Mahalanobis d_i (4.3) . Una proporción fija de éstas observaciones son truncadas de la muestra con base en los valores ordenados de d_i y el resto se usan para obtener estimaciones robustas de la media y la matriz de covarianza. Si es posible, se sustituyen los valores que se retiraron de la muestra al hacer el truncamiento. El proceso se repite hasta que se cumpla algún criterio, tal como que la norma de Frobenius de la matriz de correlación varíe poco entre iteraciones. El Truncamiento Multivariado es el más rápido de los métodos multivariados y tiene un punto de quiebre es proporcional al tamaño de la muestra truncada.

Truncamiento Multivariado Iterativo

Este método consiste en encontrar la observación con distancia de Mahalanobis más alta, eliminarla, recalcular la media y la matriz de covarianza usando los $m - 1$ puntos restantes y de éstos volver a encontrar la observación con distancia de Mahalanobis más alta para eliminarla. Se puede proponer varios criterios para saber cuántas observaciones eliminar ([8]). El punto de quiebre, sin embargo, no puede ser mejor a $1/(N + 1)$.

4.2.3. M-estimadores

En el contexto de PCA, los M-estimadores involucran pesar cada observación al obtener estimadores tanto de la media (4.4) como la matriz de covariana (4.5). A las observaciones atípicas se les disminuye su peso en lugar de ser truncadas tal como ocurre en el método de Truncamiento Multivariado. Estas técnicas son iterativas donde las estimaciones $\hat{\mu}$ y C pueden cambiar con cada iteración.

Para entender cómo surgen en torno a PCA, considérese el caso de obtener una estimación de ubicación (de la media) en el caso unidimensional. En términos de M-estimadores, las ecuaciones (4.1) y (4.2) indican encontrar T_m tal que

$$min_{T_m} \sum_{i=1}^{m} \rho(x_i - T_m)$$

o bien

$$\sum_{i=1}^{m} \psi(x_i - T_m) = 0.$$

Esta última ecuación puede ser reescrita de forma equivalente como

$$\sum_{i=1}^{m} \frac{\psi(x_i - T_m)}{x_i - T_m}(x_i - T_m) = \sum_{i=1}^{m} \omega_i (x_i - T_m) = 0,$$

con

$$\omega_i = \frac{\psi(x_i - T_m)}{x_i - T_m}.$$

Lo cual lleva a la representación de T_m como una media pesada, con los respectivos pesos dependientes de la muestra y asignados de forma proporcional a la función de influencia:

$$T_m = \frac{\sum_{i=1}^{m} \omega_i x_i}{\sum_{i=1}^{m} \omega_i}.$$

Ahora bien, para el caso multivariado y haciendo suposiciones de distribuciones elípticas se considera ya no $x_i - T_m$ sino la distancia de Mahalanobis entre x_i y T_m.

Para el caso general que involucra PCA, se debe considerar que se estiman al mismo tiempo tanto la ubicación ($\hat{\mu}$) como la escala (C). Esto se traduce (ver [8, 6, 4]) en los estimadores de media y covarianza definidos por

$$\hat{\mu} = \sum_{i=1}^{m} \frac{\omega_\mu(d_i) x_i}{\sum_{i=1}^{m} \omega_\mu(d_i)}, \tag{4.4}$$

$$C = \frac{\sum_{i=1}^{m} \omega_C(d_i^2)(x_i - \mu)(x_i - \mu)^T}{f\{\omega_C(d_i^2)\}}. \tag{4.5}$$

La determinación de $\omega_\mu(d)$, $\omega_C(d^2)$, y $f\{\omega_C(d_i^2)\}$ es específico de la técnica particular que se aplique.

Como se puede ver, también aquí la distancia de Mahalanobis (4.3) juega un papel central, ya que en (4.4) y (4.5) los pesos a las observaciones están en función de ésta. Existen diversas formas de asignar esos pesos, entre las cuales se encuentran las siguientes (en todas ellas N es la dimensión de las observaciones):

1. Maronna

$$\omega_\mu(d_i) = \frac{N + \nu}{\nu + d_i^2},$$

$$\omega_C(d_i^2) = \frac{N + \nu}{\nu + d_i^2},$$

donde ν es por lo general igual a 1. Se define $f\{(\omega_C(d_i)\} = 1/m$.

2. Huber

$$\omega_\mu(d_i) = \begin{cases} 1 & d_i \leq c_1 \\ c_1/d & d_i > c_1 \end{cases},$$

$$\omega_C(d_i^2) = \{\omega_C(d_i)\}^2/c_2.$$

El valor de c_1 se fija usualmente en $\chi^2_{N\cdot 10}$. El valor de c_2 es una corrección para hacer el estimador de C insesgado. $f\{(\omega_C(d_i)\}$ se fija en $1/m$.

3. Cambell

Siendo

$$\omega^*(d_i) = \begin{cases} d_i & d_i \leq c_3 \\ c_3 e^{-\frac{1}{2}(d_i - c_3)^2/c_4^2)} & d_i > c_3 \end{cases} \tag{4.6}$$

donde $c_3 = \sqrt{p} + c_5/\sqrt{2}$. Entonces

$$\omega_\mu(d_i) = \{\omega^*(d_i)\}/d_i,$$

$$\omega_C(d_i^2) = \{\omega_\mu(d_i)\}^2,$$

$$f\{(\omega_C(d_i)\} = \sum_{j=1}^{m} \{\omega_C(d_j)\}^2 - 1.$$

Campbell sugiere usar $c_4 = 1{,}25$ y $c_5 = 2$. Esto produce pesos que decrecen más rápidamente que en el esquema de Huber. El valor de c_5 se iguala a algún porcentaje de la distribución normal. Campbell utiliza el valor de $c_5 = 0{,}05$.

Los M-estimadores generalmente tienen un punto de quiebra de alrededor de $1/N$, lo cual puede ser un problema para una aplicación con un gran número de variables.

4.2.4. Estimación directa de los Componentes Principales

Los métodos discutidos en la Sección (4.2.3) producen estimadores robustos de Σ y μ. Los eigenvectores y eigenvalores se obtienen de estos estimadores robustos de Σ, por lo que los componentes principales resultantes al realizar PCA se llaman robustos porque comenzaron con una matriz robusta.

A diferencia de los métodos anteriores, aquí se discutirá un método representativo de otro paradigma para PCA Robusto presentado por Campbell ([1]). En este tipo de métodos se obtienen estimadores robustos de los eigenvalores Λ y eigenvectores V de Σ directamente. De éstos, se pueden encontrar estimadores robustos de Σ al hacer la multiplicación $\Sigma = V\Lambda V^T$.

El procedimiento de Campbell consiste en:

1. Tomar un estimador inicial de v_1, el primer eigenvector del estimador de Σ, C.
2. Formar los componentes principales $y_p = v_1^T(x_p - \hat{\mu})$.
3. Determinar los M-estimadores de media y varianza de y_p, y los pesos asociados ω_p de cada observación. (Aquí Campbell propone comenzar con la mediana y

$(0{,}74(rango\ intercuartil))^2$ de y_p como varianza para tener estimadores robustos iniciales; $0{,}74 = (2 \times 0{,}675)^{-1}$ y 0,675 es el 75-percentil de la distribución normal estándar; está asignación es para asegurar que la proporción de observaciones a las que se les disminuye el peso sea pequeña.)

4. Recalcular la media y varianza de y_p junto con los respectivos pesos ω_p, tomar los pesos ω_p como el mínimo de los pesos para la iteración actual y la previa. Ésto con el fin de evitar oscilaciones en la solución.

5. Calcular $\hat{\mu}$ y C como en (4.4) y (4.5), usando los últimos pesos ω_m del paso anterior.

6. Determinar el primer eigenvalor y eigenvector v_1 de C.

7. Repetir los pasos del 2 al 5 hasta que los estimadores sucesivos del eigenvalor estén suficientemente cerca. Para determinar direcciones sucesivas v_i, $2 \leq i \leq N$, proyectar los datos al espacio ortogonal al generado por los eigenvectores previos $v_1, \cdots, v_{i-1}$, y repetir los pasos del 2 al 5; tomar como el estimador inicial el segundo eigenvector de la última iteración del eigenvector pasado. El procedimiento para las direcciones sucesivas es hacer $x_{ip} = (I - V_{i-1}V_{i-1}^T)x_p$, donde $V_{i-1} = (v_1, \cdots, v_{i-1})$.

8. Repetir los pasos del 2 al 5 con x_{ip} reemplazando a x_p, y determinar el primer eigenvector v.

9. Las proyecciones sobre el componente principal están dadas por $v^T x_{ip} = v^T(I - V_{i-1}V_{i-1}^T)x_p$, y por ende $v_i = (I - V_{i-1}V_{i-1}^T)x_p$. Para determinar los subsecuentes eigenvalores, por ejemplo el k-ésimo, se substituye a x_p por su proyección en v_{k-1} y se repite el algoritmo.

El algoritmo se detiene hasta que todos los eigenvalores y eigenvectores, junto con los pesos asociados, se determinen. También, puede detenerse cuando cierta proporción de la varianza se encuentre explicada por los eigenvalores encontrados hasta la k-ésima iteración.

Como se mencionó anteriormente, se puede encontrar el estimador robusto de la matriz de covarianza multiplicando ($\Sigma = V\Lambda V^T$). También, cabe hacer notar que Campbell menciona que este procedimiento también garantiza un estimador de la matriz de covarianza que es semi-positivo definido.

4.2.5. Projection Pursuit

El método de Projection Pursuit (Búsqueda de Proyecciones) fue utilizado por Li y Chen en 1985 para encontrar componentes principales. En este método un estimador robusto de escala es maximizado en lugar de la varianza. En proyection pursuit se buscan también las direcciones con máxima dispersión, pero en lugarde usar la varianza como medida de dispersión, se usan estimadores robustos de escala S_n, conocidos como *índices de Projection-Pursuit*.

Para una secuencia de observaciones x_j, el primer 'eigenvector' se define como

$$v_{S_n,1} = max\ S_n(a^T x1, \ldots, a^T x_p),\ \|a\| = 1.$$

El eigenvalor asociado es, por definición,

$$\lambda_{S_n,k} = S_n^2(v_{S_n,k}x1, \ldots, v_{S_n,k}x_p).$$

Los componentes principales se obtienen al proyectar sobre estos ."eigenvectores".

Un ejemplo de medidas de dispersión robusta es la *Median Absolute Deviation* (MAD), se define para $(y_1, \ldots, y_n)$ como

$$MAD(y_1, \ldots, y_n) = 1{,}486\ med_i\|y_i - med_j y_j\|$$

donde *med* es el operador de mediana.

4.2.6. Elipsoide de Volumen Mínimo

Este popular método que puede encontrarse en ([8]) consiste en buscar el elipsoide que contiene $h\,\%$ de observaciones y tiene volumen mínimo. El estimador robusto del vector de medias es el centroide del elipsoide y la matriz de covarianza robusta es la correspondiente al subconjunto de datos que son cubiertos por el elipsoide (multiplicada por algún valor para obtener consistencia). Se tiene que el volumen del elipsoide es $\approx \sqrt{det(C)}$.

El algoritmo del método del elipsoide de volumen mínimo se presenta a continuación, en su forma más cruda:

- Obtener una submuestra de l observaciones, determinar μ_l y C_l
- Calcular $m_l^2 = med_l(x_i - \mu_l)C_l^{-1}(x_i - \mu_l)^T$
- El volumen del elipsoide es proporcional a $(det(m_l^2 C_l))^{\frac{1}{2}} = (det(C_l))^{\frac{1}{2}}(m_l)^{l-1}$
- Repetir para muchas l y quedarse con los h que minimicen $(det(m_l^2 C_l))^{\frac{1}{2}}$

Capítulo 5

Propuesta de una Técnica para Kernel PCA Robusto

En muchas aplicaciones de hoy en día no es extraño encontrarse con situaciones donde el número de observaciones disponibles m es solamente ligeramente mayor a la dimensión de los datos N (e inclusive es menor). Tal es el caso con Kernel PCA. Aunque atacar este nuevo tipo de problemas tal vez suponga una nueva maquinaria teórica, se pretende aquí generalizar un método diseñado en estadística tradicional para incorporarlo a Kernel PCA. Con ello, se pretende hacer de él un método más robusto.

De la exposición de los capítulos anteriores se llega ahora a tratar de redefinir el método de Kernel PCA para integrarle conceptos de PCA Robusto. De los métodos vistos en PCA Robusto, hay algunos que son directamente aplicables a Kernel PCA, tal como el submuestreo de los datos. Con respecto a este método, hay que recordar que Kernel PCA busca las soluciones en el espacio generado (en $\mathcal{H}$) por los datos. Es decir, con los datos se crea un conjunto de vectores ortonormal sobre el cual se busca los componentes principales. Reducir la cantidad de datos implica potencialmente eliminar dimensiones al espacio de búsqueda (a menos que los datos eliminados sean combinación lineal de los no-eliminados en $\mathcal{H}$, lo cual, debido a la dimensionalidad es poco factible).

De otros métodos de PCA Robusto, hay algunos que sencillamente no se pueden aplicar a Kernel PCA, ejemplo es la estimación individual de los elementos de la matriz de covarianza.

El método que aquí se propone corresponde a uno de los más sencillos de PCA Robusto. Consiste en redefinir los estimadores de media y covarianza y sustituirlos por

$$\hat{\mu} = \frac{\sum_{i=1}^{m} \omega_i \Phi(x_i)}{\sum_{i=1}^{m} \omega_i}, \tag{5.1}$$

$$C = \frac{1}{\sum_{i=1}^{m} \omega_i^2 - 1} \sum_{i=1}^{m} \omega_i^2 (\Phi(x_i) - \hat{\mu})(\Phi(x_i) - \hat{\mu})^T. \tag{5.2}$$

ya que estas elecciones corresponden a pesar individualmente cada dato con un peso ω_i. Esto hace posible tomar variantes como trucamiento multivariado (asignando un peso cero) o los correspondientes a M-estimadores.

Ya sea que se piense en utilizar este tipo de estimadores o de algún otro tipo, es prácticamente indispensable considerar la distancia de Mahalanobis de alguna u otra forma como hacen casi todos los métodos robustos de PCA. En este capítulo se presentará cómo encontrarla cuando se trabaja en el espacio de características $\mathcal{H}$ *conservando el truco del kernel*. También se presentará la forma en que los nuevos estimadores de media y covarianza pueden introducirse en Kernel PCA a través de la ponderación individual de los datos transformados implícitamente en $\mathcal{H}$. A la cuestión de la sensibilidad a valores iniciales de los estimadores, se presenta también algunas sugerencias. En la parte final de este capítulo se esboza el método propuesto.

5.1. Distancia de Mahalanobis en el espacio $\mathcal{H}$

Para la derivación de la distancia de Mahalanobis en el espacio de características $\mathcal{H}$ se supondrá que la dimensión de éste es finita (cuando es el caso de usar, por ejemplo, un kernel polinomial) y también que los datos están centrados (más adelante se regresará a este punto para derivar el caso general). Considérese el estimador de la matriz de covarianza usado comúnmente en Kernel PCA, dado como

$$C = \frac{1}{m}\sum_{i=1}^{m} \Phi(x_i)\Phi^T(x_i) = \frac{1}{m}X^T X$$

donde

$$X = \begin{pmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_m) \end{pmatrix}. \tag{5.3}$$

Usando esta notación, la matriz de Gram puede escribirse simplemente como $K = XX^T$.

Como primer paso, es necesario observar que el rango de la matriz de Gram, K, y del estimador de la matriz de covarianza en el espacio de características, C, es el mismo ([11]). Para verificarlo, considérese un par eigenvalor - eigenvector de K, λ^K y α con $\alpha^T\alpha = 1$, entonces

$$K\alpha = \lambda^K\alpha,$$

$$XX^T\alpha = \lambda^K\alpha,$$

que multiplicando por $\frac{1}{m}X^T$ resulta

$$\frac{1}{m}X^T X X^T \alpha = \frac{\lambda^K}{m} X^T \alpha,$$

$$C(X^T\alpha) = \frac{\lambda^K}{m}(X^T\alpha).$$

Esto es, $\frac{\lambda^K}{m}$ es un eigenvalor de C y $X^T\alpha$ el eigenvector correspondiente con norma

$$\|X^T\alpha\|^2 = \alpha^T X X^T \alpha = \alpha^T \cancel{K\alpha} = \cancel{\lambda^K \alpha} \cancel{= 1} = \lambda^K (\alpha^T \cancel{\alpha}) = \lambda^K.$$

Así, un eigenvector normalizado de C es entonces $v = \frac{1}{\sqrt{\lambda^K}} X^T \alpha$.

Al ser C una matriz simétrica de N por N se tiene que $rango(C) \leq N$, asimismo, para la matriz de Gram se cumple que $rango(K) \leq m$. Como un eigenvalor λ^K de K se puede encontrar apartir de un eigenvalor λ_C de C (mediante $\lambda^K = m\lambda^C$) y viceversa, el estimador de la matriz de covarianza en el espacio de características tiene a lo mucho un rango determinado por $min(rango(C), rango(K))$.

Surge aquí entonces una cuestión ineludible (y que básicamente no se menciona en la literatura): si $\mathcal{H}$ posee una dimensión mayor que el número de observaciones disponibles m entonces el estimador de la matriz de covarianza en $\mathcal{H}$ es automáticamente singular. Por ello, se va a considerar momentáneamente dos casos: cuando el estimador es no-singular (y la matriz de Gram lo es) y cuando el estimador de covarianza es singular.

Estimador de covarianza de rango completo

Cuando C es de rango completo con $rango(C) = h$ donde h es la dimensión del espacio $\mathcal{H}$ (y por lo tanto K será singular), la derivación de la distancia de Mahalanobis en $\mathcal{H}$ se basa en considerar la descomposición espectral de C

$$C = V\Lambda V^T.$$

Se tiene que la distancia de Mahalanobis (cuadrada) es

$$d^2 = \Phi(x)^T (V\Lambda V^T)^{-1} \Phi(x).$$

Utilizando la descomposición espectral de C

$$d^2 = \Phi(x)^T (V\Lambda V^T)^{-1} \Phi(x) = \Phi(x)^T (V\Lambda^{-1} V^T) \Phi(x),$$

o bien

$$d^2 = \begin{pmatrix} \Phi(x)\cdot v_1, & \Phi(x)\cdot v_2, & \cdots, & \Phi(x)\cdot v_h \end{pmatrix} \begin{pmatrix} 1/\lambda_1^C & 0 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2^C & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & 1/\lambda_h^C \end{pmatrix} \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_h \end{pmatrix}$$

donde h es la dimensión del espacio al cual transforma implícitamente el kernel y λ^C es un eigenvalor del estimador de la matriz covarianza; entonces se llega a

$$d^2 = \sum_{k=1}^{h} \frac{(\Phi(x)\cdot v_k))^2}{\lambda_k^C}.$$

$\Phi(x)\cdot v_k$ no es otra cosa que la proyección de $\Phi(x)$ en el componente principal k-ésimo, mismo que puede encontrarse fácilmente recordando que

$$\Phi(x)\cdot v_k = \sum_{i=1}^{m} \alpha_i^k \Phi(x)\cdot(\Phi(x_i)).$$

Donde α^k es el k-ésimo eigenvector de la matriz de Gram. Aquí, se puede notar por el argumento previo que λ_k^C también se puede sustituir por los valores propios de la matriz de Gram ya que $\lambda^K = m\lambda^C$. Así, la distancia de Mahalanobis en $\mathcal{H}$ puede calcularse utilizando únicamente los eigenvalores y eigenvectores de la matriz de Gram *para valores de λ diferentes de cero*:

$$d^2 = m\sum_{k=1}^{m} \frac{(\sum_{i=1}^{m} \alpha_i^k \Phi(x)\cdot\Phi(x_i))^2}{\lambda_k^K}. \tag{5.4}$$

La expresión (5.4) se reduce a la distancia de Mahalanobis cuando el rango de la matriz de Gram es mucho menor que el número de datos m (situación que corresponde a $m \gg N$. Al aumentar el rango de K, la situación es número de observaciones m aproximadamente (pero mayor) a h.

Estimador de covarianza singular

Cuando se deja a un lado la suposición de que C es de rango completo, surge el problema de cómo invertir C para encontrar la distancia de Mahalanobis. Como se quiere utilizarla en el contexto de componentes principales, y se sabe que en Kernel PCA todas las soluciones de componentes principales están en el espacio generado por los datos, se puede buscar la distancia de Mahalanobis tomándola en el espacio generado.

Sea X la matriz de datos como en la ecuación (5.3), con los mismos centrados en $\mathcal{H}$. Nuevamente, considérese la descomposición $C = V\Lambda V^T$ (conocida como factorización Takagi). En este caso, Λ contendrá elementos en la diagonal $\lambda_i^C = 0$, $i > r$ para algún r. Para proyectar los datos a este subespacio (que es el que generan) tomamos la

nueva matriz de datos $X_r = XV_r$ donde V_r contiene las primeras r columnas de V correspondientes a $\lambda_i \neq 0$. La nueva estimación de la matriz de covarianza en el subespacio generado es

$$C_r = X_r^T X_r = (XV_r)^T (XV_r)$$

y la distancia (cuadrada) de Mahalanobis en el subespacio es

$$d_r^2 = \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_r \end{pmatrix}^T (C_r)^{-1} \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_r \end{pmatrix} = \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_r \end{pmatrix}^T ((XV_r)^T (XV_r))^{-1} \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_r \end{pmatrix}$$

o bien definiendo

$$\Phi_r(x) = \begin{pmatrix} \Phi(x)\cdot v_1 \\ \Phi(x)\cdot v_2 \\ \vdots \\ \Phi(x)\cdot v_r \end{pmatrix},$$

$$d_r^2 = \Phi_r(x)^T ((XV_r)^T (XV_r))^{-1} \Phi_r(x) = \Phi_r(x)^T (V_r^T X^T X V_r)^{-1} \Phi_r(x),$$

$$d_r^2 = \Phi_r(x)^T (V_r^T C V_r)^{-1} \Phi_r(x) = \Phi_r(x)^T (V_r^T V \Lambda V^T V_r)^{-1} \Phi_r(x),$$

$$d_r^2 = \Phi(x)_r^T \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & \vdots & \cdots & 0 \\ 0 & \cdots & & 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} \lambda_1^C & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2^C & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \lambda_r^C & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & 1 & 0 \\ \vdots & \cdots & & & 0 \\ 0 & \cdots & & & 0 \end{pmatrix} \Phi_r(x)$$

$$d_r^2 = \Phi(x)_r^T \begin{pmatrix} \lambda_1^C & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2^C & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \lambda_i^C & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & \lambda_r^C \end{pmatrix} \Phi_r(x).$$

Sustituyendo $\Phi_r(x)$ y multiplicando se obtiene

$$d_r^2 = \sum_{k=1}^{r} \frac{(\Phi(x) \cdot v_k))^2}{\lambda_k^C}.$$

Nuevamente $\Phi(x) \cdot v_k$ no es otra cosa que la proyección de $\Phi(x)$ en el componente principal k-ésimo. En términos de los eigenvalores/eigenvectores de la matriz de Gram

$$d_r^2 = m \sum_{k=1}^{r} \frac{(\sum_{i=1}^{m} \alpha_i^k \Phi(x) \cdot \Phi(x_i))^2}{\lambda_k^K}$$

para λ^K diferente de cero. Se puede sustituir el límite r por m puesto que siempre se trabaja en el subespacio generado por los datos (para λ_k diferente de cero) con lo cual $d_r^2 = d^2$ y se obtiene nuevamente (5.4).

El término ortogonal

Si K es de rango completo (y C es singular) la fórmula (5.4) carece de un término ortogonal al subespacio generado por los datos. Para un $\Phi(x)$ dado, la distancia completa de Mahalanobis debe considerar también, mediante un razonamiento puramente geométrico, la diferencia entre la norma de $\Phi(x)$ y su proyección en el espacio generado:

$$d^2 = m \sum_{k=1}^{m} \frac{(\sum_{i=1}^{m} \alpha_i^k \Phi(x) \cdot \Phi(x_i))^2}{\lambda_k} + \gamma(\|\Phi(x)\|^2 - \sum_{k}^{m} (v_k \cdot \Phi(x))^2),$$

$$d^2 = m \sum_{k=1}^{m} \frac{(\sum_{i=1}^{m} \alpha_i^k k(x, x_i))^2}{\lambda_k} + \gamma(\|k(x,x)\|^2 - \sum_{k}^{m} (\sum_{i=1}^{m} \alpha_i^k k(x, x_i))^2). \tag{5.5}$$

El término γ (que no es calculable directamente) indica la contribución de la incertidumbre que se tiene sobre el complemento ortogonal. Sin embargo, como se pretende usar (5.5) sobre los mismos datos que generan el subespacio, el término ortogonal es siempre nulo. *Luego entonces, para Kernel PCA, (5.4) corresponde a la distancia de Mahalanobis para los datos con los que se estiman los componentes principales.*

Para incluir explícitamente la contribución ortogonal en la distancia de Mahalanobis, considérese (para C singular) una derivación alternativa donde se considera una versión *regularizada* de C:

$$C_{reg} = C + \gamma I$$

donde I es la matriz identidad. Desarrollando la distancia de Mahalanobis para esta versión regularizada se llega a que

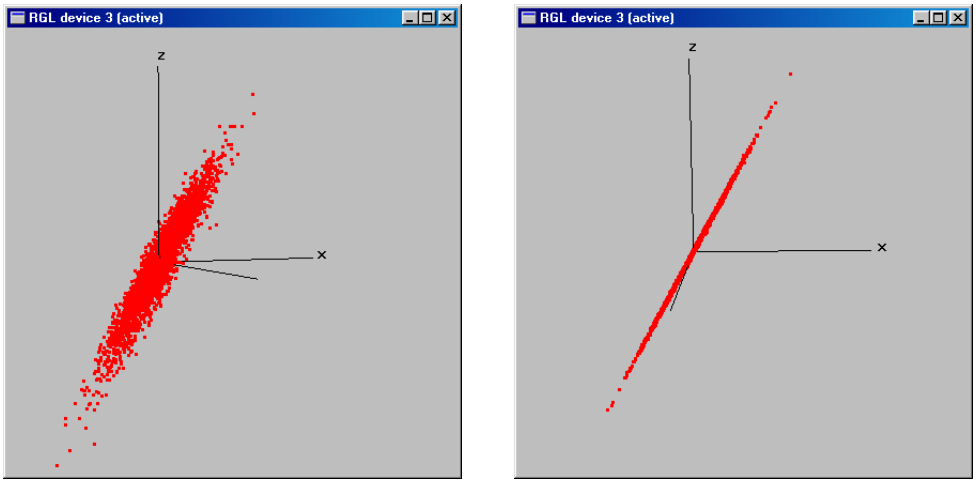

Figura 5-1: Un conjunto de datos con un estimador de matriz de covarianza C singular. No es posible crear una base ortogonal de $\mathbb{R}^3$ con las observaciones disponibles.

$$d^2 = m \sum_{k=1}^{m} \frac{(\sum_{i=1}^{m} \alpha_i^k \Phi(x) \cdot \Phi(x_i))^2}{\lambda_k^K + \gamma} \tag{5.6}$$

y γ indica la importancia relativa que tiene el complemento al subespacio que generan los datos. El efecto que tiene γ es disminuir la influencia de los eigenvalores más pequeños obtenidos a través del estimador singular de la matriz de covarianza. Asignarle un valor numérico en la práctica no es tarea sencilla. Como su función es disminuir el efecto de los eigenvalores más pequeños puede considerarse como alternativa truncar la sumatoria en (5.4) para algún valor j cercano a m, lo que produce un efecto similar a contemplar el complemento ortogonal al subespacio generado en la distancia de Mahalanobis.

Si K es singular, puede ser que el estimador de la matriz de covarianza también lo sea, como se muestra un caso en la Figura (5-1) donde se presenta un conjunto de datos en $\mathbb{R}^3$ con un estimador de matriz de covarianza C singular. No es posible crear una base ortogonal de $\mathbb{R}^3$ con las observaciones disponibles. Aún así, cuando se tiene que el rango de K es mucho menor que el número de datos (equivalente a trabajar con más datos que dimensiones) el término ortogonal al subespacio en la distancia de Mahalanobis puede ser despreciado (evitando así la regularización o el truncamiento de la sumatoria).

Para dejar atrás la suposición de que los datos están centrados en $\mathcal{H}$, se consideran ahora las proyecciones de datos centrados sobre los eigenvectores obtenidos de los datos centrados, es decir

$$d = m \sum_{k=1}^{m} \frac{(\tilde{\phi}(x) \cdot \tilde{u_k})^2}{\lambda_k^K}$$

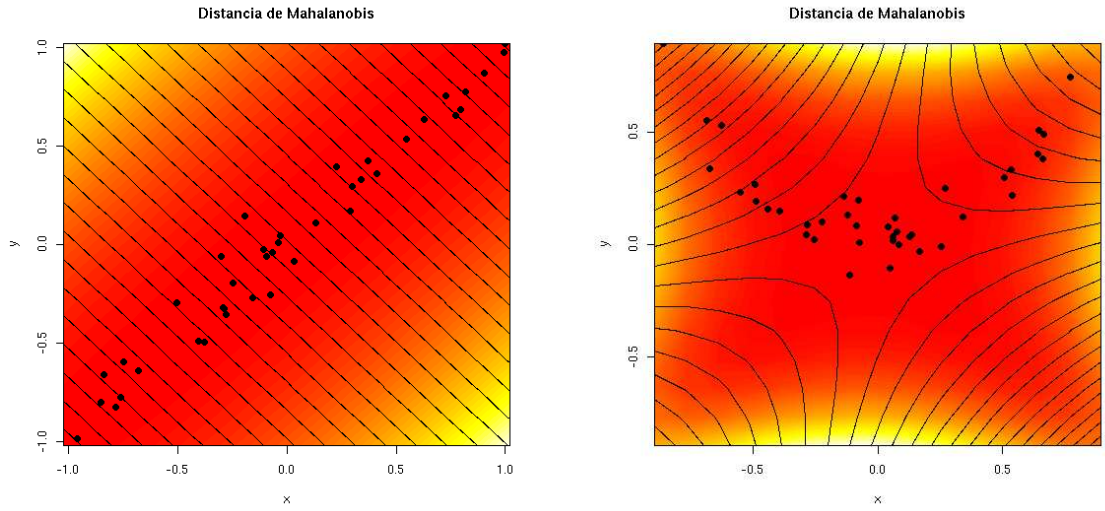

Figura 5-2: Distancias de Mahalanobis para puntos en el plano usando falso color de cuerpo caliente para graficar la magnitud. La Figura izquierda es la transformación idéntica mientras que la derecha es usando un kernel polinomial de grado 2.

y

$$\tilde{\phi}(x) \cdot \tilde{u_k} = \sum_{i=1}^{m} \alpha_i^k \tilde{\phi}(x) \cdot (\tilde{\phi}(x_i)).$$

Se puede demostrar que las proyecciones sobre los componentes principales se pueden encontrar en términos del kernel centrado

$$\tilde{\phi}(x) \cdot \tilde{u_k} = \sum_{i=1}^{m} \alpha_i^k \tilde{k}(x, x_i).$$

Una consideración similar se hace en el Apéndice sobre las proyecciones cuando se utilizan estimadores ponderados como los que se presentan en la siguiente sección.

La Figura (5-2) muestra las distancias de Mahalanobis para puntos en el plano usando falso color de cuerpo caliente para graficar su magnitud. La Figura izquierda es la transformación idéntica (cuyo resultado era el que se esperaba) mientras que la derecha es usando un kernel polinomial de grado 2. En ambos casos se tiene que el rango de K es mucho menor al número de datos.

5.2. Ponderación en el espacio de características

Los nuevos estimadores de la media y covarianza para los datos transformados en $\mathcal{H}$ para Kernel PCA se definen aquí de la siguiente forma respectivamente

$$\mu = \frac{\sum_{i=1}^{m} \omega_i \Phi(x_i)}{\sum_{i=1}^{m} \omega_i}, \tag{5.7}$$

$$C = \frac{1}{\sum_{i=1}^{m} \omega_i^2 - 1} \sum_{i=1}^{m} \omega_i^2 (\Phi(x_i) - \mu)(\Phi(x_i) - \mu)^T. \tag{5.8}$$

La solución para μ y C proviene de un proceso iterativo. Estos estimadores se pueden comparar con los del trabajo de Campbell ([1]) y se diferencían de sus estimadores únicamente en que aquí se usan los datos transformados $\Phi(x)$. Como se mencionó anteriormente, utilizar estos nuevos estimadores para Kernel PCA permiten la introducción de varios métodos robustos.

Es fácil demostrar que los estimadores anteriores de media y covarianza implican una asignación de peso a cada una de las observaciones (mismos que serán determinados en el proceso iterativo). Así:

$$X = \begin{pmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_m) \end{pmatrix} \mapsto \begin{pmatrix} \omega_1 \phi^T(x_1) \\ \omega_2 \phi^T(x_2) \\ \vdots \\ \omega_m \phi^T(x_m) \end{pmatrix}.$$

Cabe aquí preguntarse si esta asignación de pesos puede mantenerse sin tener que hacer la transformación explícita de $\Phi : x \mapsto \Phi(x)$ (manteniendo el truco del kernel). Efectivamente, es posible como se mostrará a continuación.

Recuérdese que para los eigenvectores de la matriz de Covarianza existen coeficientes α_i $(i = 1, \cdots, m)$ tales que

$$v = \sum_{i=1}^{m} \alpha_i \Phi(x_i).$$

Una ponderación individual de los datos *en el espacio de características* implica para los eigenvectores de C que

$$v = \sum_{i=1}^{m} \alpha_i \omega_i \Phi(x_i).$$

Definiendo la matriz diagonal de m por m W, cuyos elementos de la diagonal son los pesos (ordenados) de cada uno de los datos, hace que la matriz de X_w con las observaciones ponderadas puede escribirse como

$$X_w = \begin{pmatrix} \omega_1 \phi^T(x_1) \\ \omega_2 \phi^T(x_2) \\ \vdots \\ \omega_m \phi^T(x_m) \end{pmatrix} = \begin{pmatrix} \omega_1 & 0 & 0 & \cdots & 0 \\ 0 & \omega_2 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & \omega_m \end{pmatrix} \begin{pmatrix} \phi^T(x_1) \\ \phi^T(x_2) \\ \vdots \\ \phi^T(x_m) \end{pmatrix}.$$

$$X_w = WX.$$

Con la ponderación, la matriz de Gram y de Covarianza (en $\mathcal{H}$) pueden escribirse respectivamente como

$$K_w = X_w X_w^T,$$

$$C_w = \frac{1}{\sum_{i=1}^m \omega_i^2 - 1} X_w^T X_w,$$

donde nuevamente se supone que los datos están centrados en $\mathcal{H}$ (más sobre esto a continuación).

Sean α_w y v_w los eigenvectores correspondientes a la matriz de Gram y Covarianza al hacer la ponderación, $K_w\alpha_w = \lambda_w^K \alpha_w$, $C_w v_w = \lambda_w^C v_w$. Entonces, se puede observar que

$$K_w\alpha_w = \lambda_w^K \alpha_w,$$

$$X_w X_w^T \alpha_w = WXX^T W\alpha_w = \lambda_w^K \alpha_w,$$

Multiplicando por $\frac{1}{\sum_{i=1}^m \omega_i - 1} X^T W$,

$$\frac{1}{\sum_{i=1}^m \omega_i - 1} X^T WWXX^T W\alpha_w = \frac{\lambda_w^K}{\sum_{i=1}^m \omega_i - 1} X^T W\alpha_w,$$

$$C_w X^T W\alpha_w = \frac{\lambda_w^K}{\sum_{i=1}^m \omega_i - 1} X^T W\alpha_w.$$

Lo que indica que un eigenvector normalizado de C_w es

$$v = \frac{1}{\sqrt{\lambda_w^K}} X^T W\alpha_w = \frac{1}{\sqrt{\lambda_w^K}} \sum_{i=1}^m \alpha_{w_i}\omega_i \Phi(x_i) = \frac{1}{\sqrt{\lambda_w^K}} \sum_{i=1}^m \beta_i \Phi(x_i),$$

donde $\beta_i = \alpha_{w_i}\omega_i$. Así, se tiene que incluir el nuevo estimador de la Covarianza para Kernel PCA (5.8) implica solamente cambiar los coeficientes en la expansión. Es evidente que utilizar (5.8) se traduce para la matriz de Gram que originalmente se calcula como

$$K_{ij} = \Phi(x_i) \cdot \Phi(x_j) = k(x_i, x_j)$$

en calcularla ahora de la forma

$$K_{w_{ij}} = (\omega_i \Phi(x_i)) \cdot (\omega_j \Phi(x_j)) = \omega_i \omega_j k(x_i, x_j).$$

Para obtener los Componentes Principales en $\mathcal{H}$ basta encontrar los coeficientes de expansión β_i $(i = 1, \cdots, m)$, problema que se traduce en encontrar los eigenvectores/valores de la matriz de Gram ponderada K_w y los respectivos pesos de cada observación ω_i $(i = 1, \cdots, m)$. Se puede proyectar entonces con

$$v \cdot \Phi(x) = \sum_{i=1}^{m} \beta_i \Phi(x) \cdot \Phi(x_i) = \sum_{i=1}^{m} \alpha_{w_i} \omega_i \Phi(x) \cdot \Phi(x_i) = \sum_{i=1}^{m} \alpha_{w_i} \omega_i k(x, x_i).$$

Ahora, se dejará la suposición de que los datos se encuentran centrados en $\mathcal{H}$ $(\mu = \frac{\sum_{i=1}^{m} \omega_i \Phi(x_i)}{\sum_{i=1}^{m} \omega_i} \neq 0)$. Entonces, los elementos de la matriz de Gram ponderada y centrada son

$$\widetilde{K_{w_{ij}}} = \widetilde{\Phi}_w(x_i) \cdot \widetilde{\Phi}_w(x_j) = \omega_i \omega_j (\Phi(x_i) - \frac{\sum_{r=1}^{m} \omega_r \Phi(x_r)}{\sum_{p=1}^{m} \omega_p}) \cdot (\Phi(x_j) - \frac{\sum_{s=1}^{m} \omega_s \Phi(x_s)}{\sum_{p=1}^{m} \omega_p})$$

$$= \omega_i \omega_j (\Phi(x_i) \cdot \Phi(x_j) - \frac{\sum_{r=1}^{m} \omega_r \Phi(x_j) \cdot \Phi(x_r)}{\sum_{p=1}^{m} \omega_p} - \frac{\sum_{s=1}^{m} \omega_s \Phi(x_i) \cdot \Phi(x_s)}{\sum_{p=1}^{m} \omega_p} +$$

$$+ \frac{\sum_{r=1}^{m} \sum_{s=1}^{m} \omega_r \omega_s \Phi(x_r) \cdot \Phi(x_s)}{(\sum_{p=1}^{m} \omega_p)^2}),$$

$$= \omega_i \omega_j (k(x_i, x_j) - \frac{\sum_{r=1}^{m} \omega_r k(x_j, x_r)}{\sum_{p=1}^{m} \omega_p} - \frac{\sum_{s=1}^{m} \omega_s k(x_i, x_s)}{\sum_{p=1}^{m} \omega_p} +$$

$$+ \frac{\sum_{r=1}^{m} \sum_{s=1}^{m} \omega_r \omega_s k(x_r, x_s)}{(\sum_{p=1}^{m} \omega_p)^2}) = \omega_i \omega_j \tilde{k}(x_i, x_j). \tag{5.9}$$

Donde $\tilde{k}(x_i, x_j)$ es la versión centrada del kernel correspondiente a la matriz de Gram centrada tal como en Kernel PCA clásico, solamente que usando el estimador de la media definido en (5.7). Usando una notación matricial, se obtiene la expresión más compacta para la matriz de Gram centrada y ponderada que hay de diagonalizar

$$\widetilde{K_w} = W(K - 1_\omega WK - KW1_\omega + 1_\omega WKW1_\omega)W \tag{5.10}$$

donde $W = diag(\omega_1, \omega_2, \cdots, \omega_m)$ y $(1_\omega)_{ij}$ es la matriz de m por m cuyos elementos son todos $\frac{1}{\sum_{p=1}^{m} \omega_p}$.

Para encontrar el k-ésimo componente principal en $\mathcal{H}$, se calculan las proyecciones de las imágenes Φ centradas de datos de prueba t sobre el k-ésimo eigenvector de la matriz de covarianza de los datos centrados y ponderados

$$\tilde{v_w}^k \cdot \Phi(t) = \sum_{i=1}^{m} \tilde{\alpha_w}_i^k \omega_i (\tilde{\Phi}_w(x_i)\tilde{\Phi}_w(t)).$$

Considérese un conjunto de datos de prueba $t_1, \cdots, t_l$, y defínase dos matrices de $l \times m$ mediante

$$K_{ij}^{prueba} = (\Phi(t_i) \cdot \Phi(x_j)) = k(t_i, x_j)$$

y

$$\widetilde{K_{w_{ij}}^{prueba}} = ((\Phi(t_i) - \frac{\sum_{r=1}^{m} \omega_r \Phi(x_r)}{\sum_{r=1}^{m} \omega_r}) \cdot (\Phi(x_j) - \frac{\sum_{s=1}^{m} \omega_s \Phi(x_s)}{\sum_{i=1}^{m} \omega_i})).$$

De forma similar a (5.9), podemos expresar $\widetilde{K_{w_{ij}}^{prueba}}$ en términos de K_{ij}^{prueba} y llegar a

$$\widetilde{K_{w_{ij}}^{prueba}} = K_{ij}^{prueba} - L_\omega W K - K^{prueba} W M_\omega + L_\omega W K W M_\omega.$$

Aquí, L_ω es una matriz de $l \times m$ con todas sus entradas iguales a $(1/\sum_{i=1}^{m} \omega_i)$; M_ω tiene las mismas entradas pero sus dimensiones son $m \times m$, mientras que K tiene entradas $K_{ij} = k(x_i, x_j)$.

Luego entonces, si se quiere encontrar el k-ésimo componente principal para los l datos de prueba se puede hacer con

$$P_k = \widetilde{K_{ij}^{prueba}} W \tilde{\alpha_w}^k. \tag{5.11}$$

Conviene aquí hacer un comentario sobre el kernel centrado de la expresión (5.9), mismo que se obtuvo utilizando el estimador de la media definido como (5.7). Se podría preguntar si existe alguna restricción sobre los pesos para mantener la validez del kernel. Para ello se presentan las siguientes Proposiciones cuya demostración puede encontrarse en [9] (ver también el Apéndice para definiciones).

Proposición 5.2.1 (Matrices Semi-Positiva y Condicionalmente Definidas) *Sea K una matriz simétrica, $\mathbf{e} \in \mathbb{R}^m$ el vector de todas las entradas igual a 1, $\mathbf{I}$ la matriz identidad de $m \times m$, y sea $\mathbf{c} \in \mathbb{C}^m$ tal que satisfaga $\mathbf{e}^*\mathbf{c} = 1$. Entonces*

$$\tilde{K} = (\mathbf{I} - \mathbf{e}\mathbf{c}^*)K(\mathbf{I} - \mathbf{c}\mathbf{e}^*)$$

es semi-positivo definida si y sólo si K es condicionalmente semi-positiva definida.[1]

Este resultado implica una generalización de la Proposición sobre la construcción de kernels semi positivos definidos a partir de kernels condicionalmente semi positivos tal y como se definen en el Apéndice:

[1] $\mathbf{c}^*$ es el vector obtenido de transponer y tomar el complejo conjugado de $\mathbf{c}$.

Proposición 5.2.2 (Añadiendo un origen general) *Sea k un kernel simétrico, $x_1, \cdots, x_m \in \mathcal{X}$, y sea $c_i \in \mathbb{C}$ tal que satisfaga $\sum_{i=1}^m c_i = 1$. Entonces*

$$\tilde{k}(x_r, x\prime) = \frac{1}{2}(k(x, x\prime) - \sum_{i=1}^{m} c_i k(x, x_i) - \sum_{i=1}^{m} c_i k(x_i, x\prime) + \sum_{i,j=1}^{m} c_i c_j k(x_i, x_j))$$

es semi-positivo definido si y sólo si k es condicionalmente semi-positivo definido.

Para el caso del kernel (5.9), centrado mediante el estimador de la media (5.7), se tiene que $c_i = \omega_i / \sum_{k=1}^m \omega_k$. Como se supone que en general el kernel que se utilice para obtener componentes principales será (condicionalmente) semi-positivo definido, únicamente resta que la condición $\sum_{i=1}^m c_i = 1$ sea satisfecha, lo cual se cumple trivialmente para *cualquier* elección de ω_i $(i = 1, \cdots, m)$.

Sin embargo, como se quiere cambiar el estimador de la covarianza C por

$$C = \frac{1}{\sum_{i=1}^m \omega_i^2 - 1} \sum_{i=1}^{m} \omega_i^2 (\Phi(x_i) - \hat{\mu})(\Phi(x_i) - \hat{\mu})^T$$

debe tenerse presente que elecciones de pesos tales que $\sum_{i=1}^m \omega_i^2 \leq 1$ harán que C quede indefinida (al ser multiplicada por un escalar negativo). Por ello, cualquier asignación de pesos reales positivos que cumpla $\sum_{i=1}^m \omega_i^2 > 1$ será suficiente para garantizar el buen comportamiento de los estimadores (5.8) y (5.7) en el sentido de mantener que se trabaje siempre con matrices semi-positvas definidas.

5.3. Método propuesto

Resumiendo los resultados de las secciones anteriores: Se proponen nuevos estimadores para Kernel PCA de la forma (5.1) y (5.2). Estos estimadores permiten generalizar los métodos robustos de PCA correspondientes a truncamiento multivariado (asignando pesos iguales a cero) y M-estimadores (pesos en función de la distancia de Mahalanobis igual a los de Campbell como se menciona en ese capítulo sobre PCA Robusto). Tanto los métodos basados en truncamiento multivariado como los basados en M-estimadores hacen uso de la distancia de Mahalanobis por lo cual se calculó mediante el truco del kernel para ser utilizada en $\mathcal{H}$, resultando en (5.4). La ponderación necesaria para poder usar (5.1) y (5.2) llevó a las ecuaciones (5.10) y (5.11).

La técnica propuesta es la generalización de los métodos robustos para PCA clásico arriba mencionados, por lo que es de naturaleza iterativa. El algoritmo consiste en

Algoritmo para Kernel PCA Robusto:

1. Calcular la matriz de Gram $\widetilde{K_w}$ centrada y ponderada mediante (5.10) con pesos iniciales igual a 1

2. Repetir hasta convergencia: Calcular la distancia de Mahalanobis de las observaciones mediante (5.4) (considerar que las proyecciones sobre los componentes son centradas y ponderadas, ver el Apéndice). Asignar pesos mediante una función de influencia ó asignar pesos igual a cero a las observaciones con mayor distancia de Mahalanobis. Esta elección define en gran parte el tipo de método que se generaliza de PCA Robusto a Kernel PCA como se mencionó anteriormente

3. Diagonalizar $\widetilde{K_w}$ con los pesos obtenidos en el paso anterior

4. Los componentes principales robustos de los datos de prueba se extraen usando (5.11)

La convergencia se puede definir de varias maneras, pudiendo ser tan simple como realizar una sola iteración o tener poca variación en los pesos (si es que se utiliza un función de influencia). Nótese también que calcular la distancia de Mahalanobis implica encontrar los eigenvectores y eigenvalores de $\widetilde{K_w}$ con los pesos actuales como se observa en (5.4), por lo que este es el paso más costoso del método. Si el rango de la matriz de Gram es mucho menor al número de datos se debe calcular la distancia de Mahalnobis como en (5.4), si K es de rango completo, contemplar la posibilidad de añadir un término de regularización γ como en (5.6) o de forma equivalente truncar la sumatoria en la expansión (5.4) para algún valor r cercano a m.

Algunos autores sugieren dar un inicio robusto al vector de medias y la matriz de covarianza al aplicar el método de M-estimadores en el caso clásico. Ésto con el fin de evitar el efecto de enmascaramiento que pueden producir las observaciones atípicas (que se mencionó en el capítulo 4) y promover una convergencia más rápida para reducir el número de iteraciones en la estimación de los componentes principales robustos. Entre estas sugerencias se incluye utilizar el vector de medianas muestrales, entre otros. El problema que surge al querer utilizar estas sugerencias para Kernel PCA es que, por ejemplo, no es posible calcular directamente un vector de medianas muestrales en $\mathcal{H}$ (ni siquiera uno de medias muestrales). A este respecto, para el método que se propone se conforma con comenzar con los estimadores clásicos obtenidos a asignar pesos iniciales igual a 1.

Cabe mencionar que la función de influencia describe el efecto (aproximado y estandarizado) de una observación adicional en cualquier punto x sobre un estadístico T,dada una muestra *grande* con distribución F. El hecho de que en general Kernel PCA basa la búsqueda de los Componentes Principales en el subespacio generado por los datos (dentro de un espacio de - potencialmente - muy alta dimensionalidad), podría implicar que la condición de poseer una muestra *grande* en ese espacio no esté satisfecha (que es lo que generalmente sucederá). A este comentario pueden

añadirse muchas críticas extra al tratar de robustificar Kernel PCA, pero en particular éste hace que seleccionar los parámetros para la función de influencia (y demás parámetros existentes) dependa de la aplicación en cuestión y del conocimiento que se tenga a la mano. Después de todo, la simple elección de un kernel y sus parámetros lleva consigo una carga muy similar.

Capítulo 6

Análisis del Método y Discusión

En este capítulo de la tesis se pretende evaluar el desempeño del método propuesto. Dado el sustento teórico en que se basa el método propuesto (que es una extensión directa de PCA Robusto a Kernel PCA) se podría pensar que los resultados para PCA Robusto se extienden de manera directa a Kernel PCA, sin embargo, hay que considerar cuestiones importantes como son el hecho de trabajar con pocos datos en espacios de muy alta dimensión.

En la literatura generalmente las formas de evaluar los métodos no-supervisados se realizan de forma heurística, por lo cual los experimentos de este tipo que aquí se presentan apelan a la intuición y tienen un carácter cuantitativo muy limitado. Estos experimentos se presentan en las primeras secciones de este capítulo. También se presenta al final del capítulo un ejemplo donde el método propuesto es parte de un método supervisado (clasificación) lo que facilita tener resultados cuantitativos para su evaluación. Este método se presenta en la sección de Extracción de Características para Clasificación.

Entre los conjuntos de datos utilizados se encuentra la base de datos de dígitos del *US Postal Service (USPS)*. Esta base de datos contiene 9298 dígitos escritos a mano. Este conjunto viene separado en 7291 datos de entrenamiento y 2007 de prueba. Cada dígito es una imagen de 16 por 16, representada como un vector de 256 dimensiones en el rango -1 a 1. El preprocesamiento utilizado es el mismo que se menciona en ([9]) y consiste en realizar un suavizado utilizando un kernel gaussiano de $\sigma = 0{,}75$. Los demás conjuntos de datos utilizados se refieren en su momento.

A continuación se presentan algunos ejemplos donde se trata de estudiar el comportamiento de lo propuesto en la tesis.

6.1. Experimentos con datos artificiales

En la literatura sobre Kernel PCA los datos artificiales se utilizan de forma regular para ilustrar el método, por lo cual se también incluyen aquí para ilustar el método robusto propuesto. Las líneas en las figuras que se mostrarán representan valores constantes de las proyecciones sobre el primer componente principal tal como se explicó en el capítulo sobre Kernel PCA.

La Figura (6-1) muestra una comparación visual entre Kernel PCA y Kernel PCA Robusto para el kernel polinomial de grado 1 (kernel identidad). Esto equivale a realizar PCA clásico robusto. Las líneas negras representan el primer PC robusto y las rojas el PC no-robusto. Se observa que la versión robusta describe la variabilidad del grueso de los datos como se espera, descartando la influencia de la observación atípica. En cambio, la versión no robusta se ve influenciada fuertemente por la observación atípica y ésto se refleja sobre el primer PC.

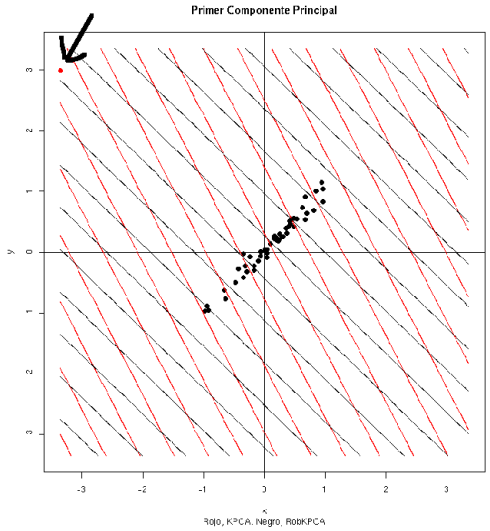

Figura 6-1: Comparación entre Kernel PCA y Kernel PCA Robusto. Las líneas negras representan el primer componente principal robusto y los rojas el no-robusto para el kernel identidad. La versión robusta describe la variabilidad del grueso de los datos como se espera, descartando la influencia de la observación atípica.

La Figura (6-2) muestra una comparación visual entre Kernel PCA y Kernel PCA Robusto para el kernel polinomial de grado 2. En este caso el rango de la matriz de Gram centrada es mucho menor al número de datos (rango 3 para los datos Figura (6-2)), en el método propuesto se puede utilizar entonces, por ejemplo, la función de influencia de Campbell con parámetro de dimensión el número de eigenvalores de la matriz de Gram diferentes de cero, calculando la distancia de Mahalanobis como en (5.4) para λ_k distintas de cero. Se observa nuevamente que la versión robusta, a diferencia de la no robusta, describe la variabilidad del grueso de los datos como se espera descartando la influencia de la observación atípica.

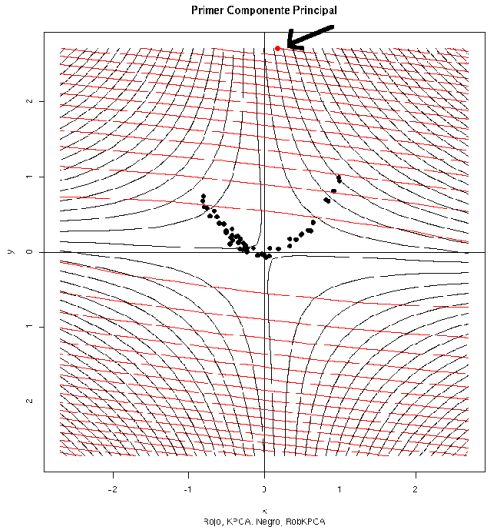

Figura 6-2: Comparación entre Kernel PCA y Kernel PCA Robusto. Las líneas negras representan el primer componente principal robusto y los rojas el no-robusto para el kernel polinomial de grado 2. Nuevamente, la versión robusta describe la variabilidad del grueso de los datos como se espera, descartando la influencia de la observación atípica.

6.2. Elipticidad en las proyecciones sobre los componentes principales

Explicación del experimento

Una de las aplicaciones de Kernel PCA es poder presentar visualmente datos en espacios de alta dimensión mediante una reducción a variables que se pueden visualilzar (ya sea una, dos o tres dimensiones). Aunque la varianza no se concentre exclusivamente en éstos primeros componentes, graficarlos usando dos dimensiones ayuda a tener una idea general de cómo se ven los datos. Sobre la cuestión si solamente dos componentes principales son adecuados para representar la mayor varianza de los datos se refiere a ([7]).

Para entender estos experimentos, considérese primero la Figura (6-3), donde se muestra un conjunto de datos generados de una distribución normal con $\mu^T = (10, 10)$ y matriz de covarianza $\Sigma = diag(1, 0{,}4)$ que fueron contaminados por unos pocos datos provenientes de otra distribución. Al obtener los componentes principales (clásicos) se obtiene las proyecciones sobe los primeros dos componentes como se muestra en la Figura (6-4). Obsérvese cómo los datos contaminantes atraen hacia ellos la estructura de variabilidad, hecho que se desea evitar puesto que se quiere estudiar la variabilidad del grueso de los datos (disminuyendo el efecto de los datos contaminantes).

Al re-estimar los componentes principales, esta vez usando el método propuesto y asignando peso $\omega_i = 0$ a los datos contaminantes se obtienen proyecciones sobre los primeros dos componentes como se muestra en la Figura (6-4). En este caso,

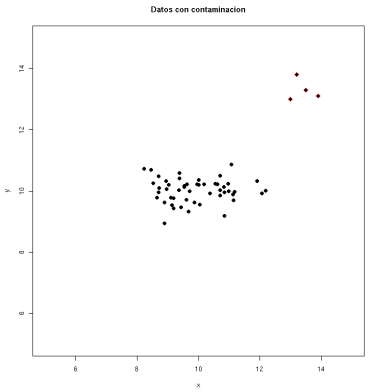

Figura 6-3: Datos normales contaminados.

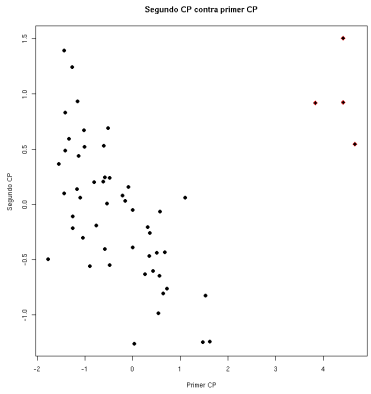

Figura 6-4: Primeros dos componentes principales de datos contaminados.

se obtiene una estimación más fidedigna de los componentes principales. Los datos contaminantes aparecen a una distancia ortogonal grande del primer componente principal y su calidad de outliers se ve reflejada en las proyecciones.

Un caso cualitativamente parecido sucede para datos en alta dimensión. Usando la base de datos de dígitos USPS, se tomó en cuenta las imágenes que representan el dígito cero. La Figura (6-6) muestra las primeras 6 imágenes de dichos datos.

La Figura (6-7) por su parte muestra una imagen que fue introducida para contaminar los dígitos correspondientes al cero. A esta imágen se le alteraron drasticamente unos cuantos pixeles aumentando su valor de intensidad (los datos USPS están normalizados al intervalo $(-1, 1)$). Esta alteración al conjunto de datos provoca proyecciones como la mostrada en la Figura (6-7). Ahora bien, al asignarle un peso $\omega_i = 0$ al dato contaminante con el método que se propone en esta tesis, se obtienen proyecciones como las de la Figura (6-9)). En estas proyecciones se puede observar que el dato contaminante ya no presenta una distancia ortogonal grande al primer componente (el primer componente posee una variación del 32 % del total, un screeplot es la Figura (6-10)). También se puede notar que el dato contaminante es atraído hacia

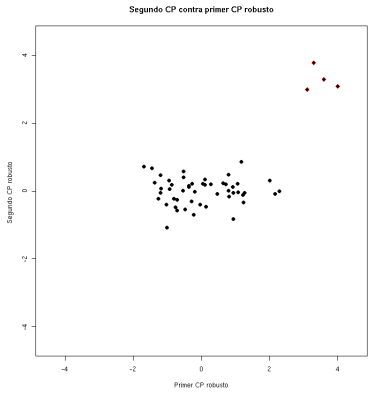

Figura 6-5: Primeros dos componentes principales de datos contaminados, versión robusta.

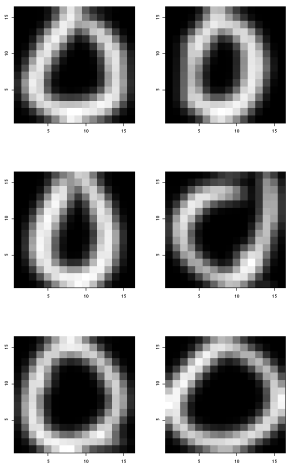

Figura 6-6: Primeras seis imágenes de dígitos 0, datos USPS.

el centro de la nube de datos. Lo que se ha logrado al eliminar la influencia del dato contaminante es obtener datos que proyectados sobre los primeros dos componentes principales *tienen un aspecto más elíptico*.

Este efecto ayuda a describir la variabilidad en el conjunto de datos USPS como se muestra en la Figura (6-11)). Así, una forma de evaluar la calidad de un método robusto para obtener componentes principales es observar las proyecciones sobre los primeros dos componentes obtenidos.

Experimentos datos USPS

Para los datos USPS, se puede ver en la Figura (6-12)) los primeros dos componentes principales de los dígitos que representan al número 1 en los datos USPS. Se puede notar que aquellos datos que atraen hacia ellos la variabilidad son funda-

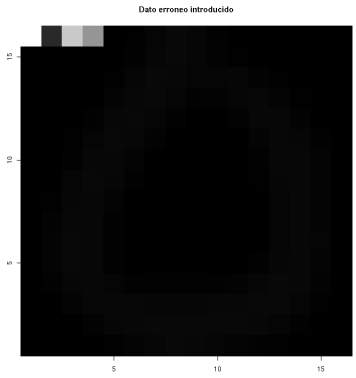

Figura 6-7: Una imagen alterada drasticamente en unos cuantos pixeles es introducida en el conjunto de dígitos correspondientes al cero del conjunto USPS.

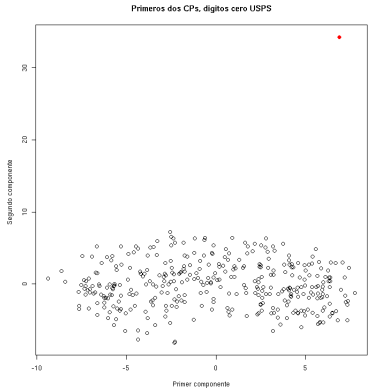

Figura 6-8: Las proyecciones sobre los primeros dos componentes de los dígitos cero contaminados con la imagen de la Figura (6-7). El punto rojo representa el dato contaminante.

mentalmente distintos a la estructura general de los números. Inclusive, se podría pensar que los más extremos podrían estar mal etiquetados y representar al número 7. Usando el método propuesto, se puede disminuir la influencia de aquellos datos con mayor distancia de Mahalanobis (usando la función de influencia de Campbell cuyos parámetros se ajustan para cada experimento particular), con lo cual se obtiene las proyecciones presentadas en la Figura (6-13)). Los puntos en rojo representan los tres datos extremos cuyas imágenes se presentan en la Figura (6-12)); en este caso también algunos de los datos extremos tienden al centro de las proyecciones.

Como es de esperarse, las proyecciones sobre los CP dependen fuertemente de la elección del kernel. Este hecho se refleja en las Figura (6-14)), donde para el mismo dígito anterior (el uno), al usar un kernel polinomial de grado 7 los datos extremos de las Figura (6-12)) cambian su relación con respecto a los demás (indicados en rojo). tanto para la versión normal como la robusta.

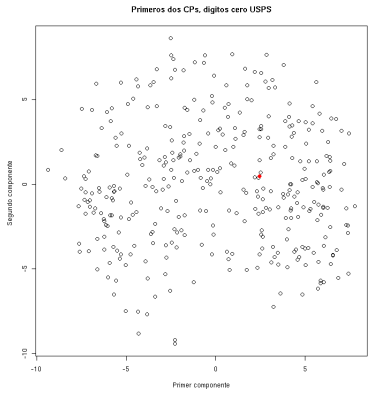

Figura 6-9: Las proyecciones sobre los primeros dos componentes principales robustos de los dígitos cero contaminados con la imagen de la Figura (6-7). El punto rojo representa el dato contaminante.

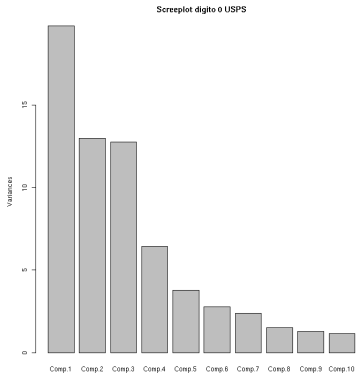

Figura 6-10: Variabilidad descrita por los primeros componentes principales de los dígitos cero, conjunto USPS.

Otro ejemplo lo encontramos en la Figura (6-15)). En esta se tomó para el mismo conjunto de datos USPS las imágenes correspondientes al dígito 5. La gráfica superior muestra los primeros dos componentes usando el kernel polinomial de grado 1 (que equivale a realizar PCA clásico). Se puede observar que las proyecciones se notan bastante elípticas. Al usar una transformación correspondiente al kernel polinomial de grado 5 se obtiene la gráfica inferior izquierda. En ésta también se puede notar que las proyecciones son bastante elípticas, por lo que aplicar el método robusto propuesto (gráfica inferior derecha) no representa un beneficio considerable. A esto hay que mencionar que este fenómeno implica que los resultados aceptables de Kernel PCA no son alterados por la versión robusta, tal como se esperaría que fuera.

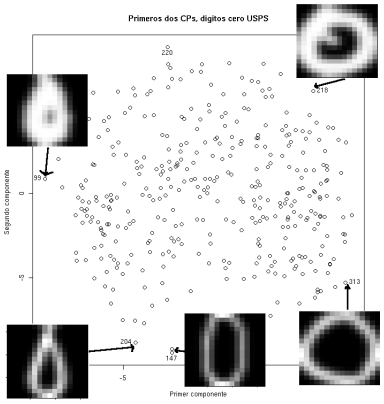

Figura 6-11: Primeros 2 componentes dígitos cero, conjunto USPS.

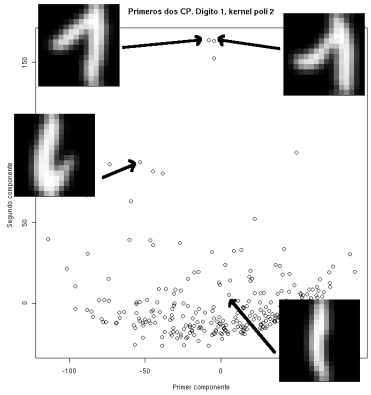

Figura 6-12: Primeros 2 componentes dígitos uno, conjunto USPS. Se usó un kernel polinomial de grado 2.

Experimentos con ORL face database (ATT)

Un ejemplo más consiste considerar las imágenes de rostros del Olivetti Research Laboratory (esta base de datos se conoce como ORL face database (ATT)). En esta base de datos se encuentran imágenes de 112 por 92 en escala de grises con intensidad en el rango $[0, 255]$. Contiene 10 diferentes imágenes de 40 sujetos distintos. La idea en este ejemplo es considerar las imágenes correspondientes a un sujeto y añadir una imágen de un sujeto distinto para realizar Kernel PCA al conjunto. Las 10 imágenes del sujeto seleccionado se presentan en la gráfica de la izquierda en la Figura (6-16), mientras que la gráfica derecha presenta la imagen adicional que se añadió.

Al realizar Kernel PCA sobre estos datos con un kernel polinomial de grado 3, se nota (como era de esperarse) que en las proyecciones sobre los dos primeros CP la imagen del sujeto al cual no pertence la mayoría de las imágenes se encuentre cambiando la estructura de variabilidad. La gráfica izquierda de la Figura (6-17) muestra las proyecciones sobre los dos primeros CP usando la versión normal de Kernel

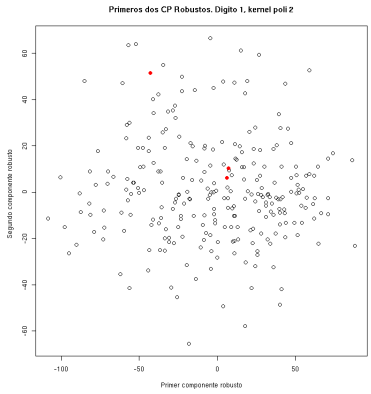

Figura 6-13: Primeros 2 componentes principales robustos dígitos uno, conjunto USPS. Se usa el mismo kernel polinomial de grado 2 que en el caso normal.

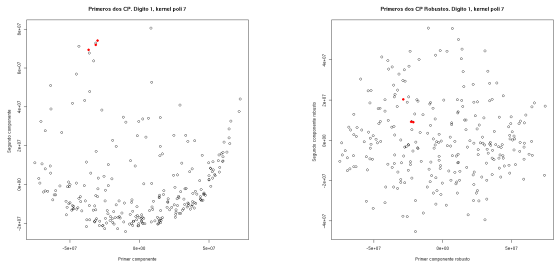

Figura 6-14: Primeros 2 componentes principales versión normal (izquierda) y robusta (derecha) del dígitos uno, conjunto USPS. Se usa el mismo kernel polinomial de grado 7. Las proyecciones dependen del kernel utilizado.

PCA. Para enfatizar la ganancia que se obtiene mediante Kernel PCA Robusto, la gráfica de enmedio es un acercamiento para incluir solo los puntos que representan imágenes del mismo sujeto en la gráfica izquierda. Al comparar esta gráfica con la gráfica de la derecha, se puede observar que la escala indica que la variación entre las imágenes del sujeto de interés ha aumentado.

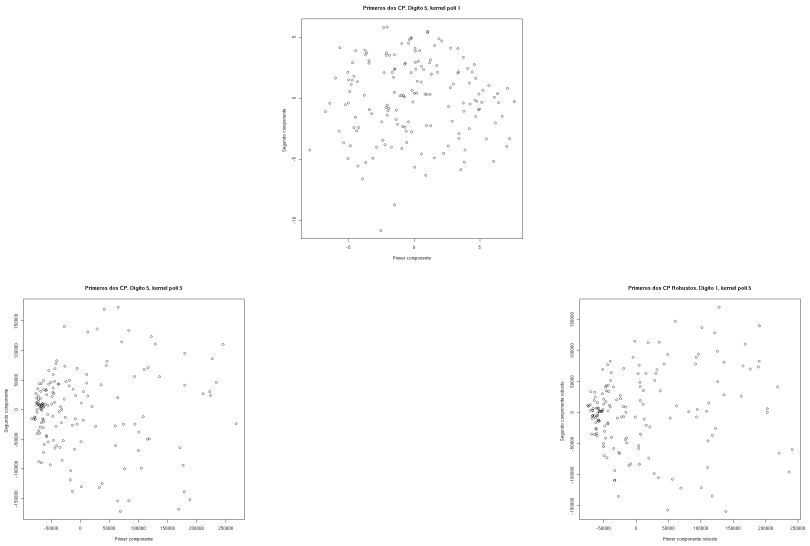

Figura 6-15: Primeros 2 componentes principales versión normal (izquierda) y robusta (derecha) de los dígitos 5, conjunto USPS. Se usa el mismo kernel polinomial de grado 5. Se puede observar claramente que las proyecciones dependen del kernel utilizado.

Figura 6-16: Izquierda: Imágenes del sujeto seleccionado. Derecha: imagen añadida para contaminar los datos del sujeto seleccionado. Se realiza Kernel PCA/Kernel PCA Robusto sobre el conjunto de imágenes.

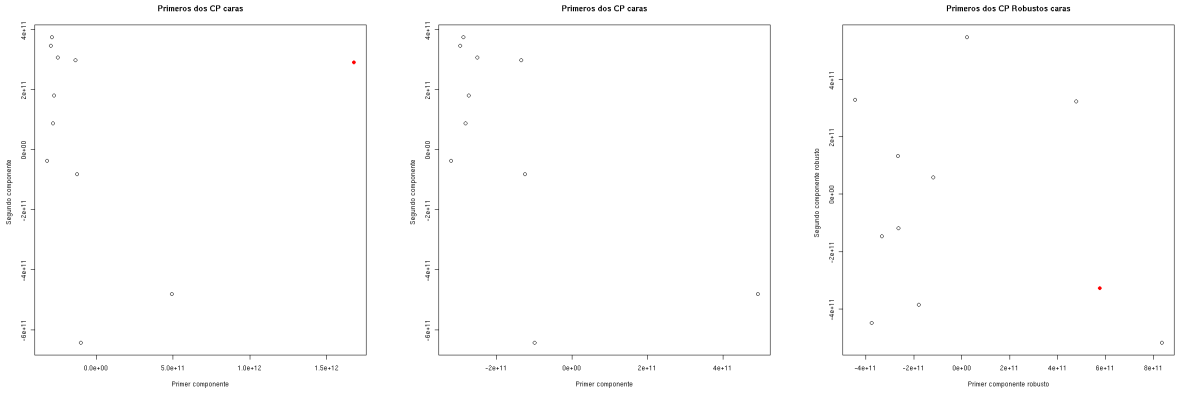

Figura 6-17: Proyecciones sobre los dos primeros CP de las imágenes ORL (ATT). Izquierda: Versión Kernel PCA. Enmedio: Acercamiento sobre los datos que pertencen al mismo sujeto. Derecha: Versión Kernel PCA Robusto. En las gráficas, el punto correspondiente a la imagen contaminante se ilustra en color rojo.

6.3. Aplicaciones diversas

Esta aplicación se tomó de ([12]). En esa publicación se propone un método robusto para PCA (clásico) y esta aplicación se usa para ilustrarlo. La aplicación toma en cuenta 506 imágenes (de 120 por 160) obtenidas de una cámara estática a lo largo de dos días. La Figura (6-18) muestra algunos ejemplos de las imágenes. Lo que se busca es aplicar PCA (y un método robusto) para construir un modelo del fondo que capture la variación de iluminación. Un 45 % de las imágenes capturada por la cámara contiene personas en varias ubicaciones. Este tipo de imágenes donde se encuentran personas (que pueden permanecer durante varias capturas de la cámara) se consideran indeseables ya que la escena sobre la cual se quiere construir el modelo es el fondo.

Figura 6-18: Ejemplos de imágenes tomadas de una cámara estática. Se busca construir un modelo de la iluminación de fondo ignorando la influencia de las personas.

En ([12]) se toman 20 vectores base obtenidos tanto por PCA como por el método que ahí se propone, y la evaluación consiste en observar la reconstrucción de algunas imágenes con presencia de personas. Cuando estas imágenes con personas se reconstruyen dejando fantasmas (donde estaba ubicada la persona), quiere decir que la presencia de personas tuvo un efecto negativo en la recuperación de la iluminación de fondo. En cambio (usando el método que proponen) se puede observar que usando la base de vectores obtenidos de manera robusta hace que se ignore completamente a las personas cuando se reconstruyen imágenes donde éstas aparecen.

El método propuesto en esta tesis y el publicado en ([12]) son comparables solo de forma cualitativa, entre otras razones porque en ese método se puede considerar outliers a los pixeles individuales - imposible de lograrse usando el método del kernel -. También, en Kernel PCA/Kernel PCA Robusto el problema de encontrar la reconstrucción (problema de la preimagen) no es trivial como se mencionó en capítulos anteriores.

Sin embargo, se decidió resolver el mismo problema de iluminación de fondo usando el conjunto de datos, usando un kernel polinomial de grado 1 (kernel identidad), mismo que equivale a realizar PCA clásico. La ventaja de realizar Kernel PCA usando el kernel identidad es que la complejidad del problema está en función del número de datos (que para este ejercicio son 506) y no de su dimensionalidad (19200). Como se mostrará más adelante, los resultados obtenidos aquí y en ([12]) son cualitativamente similares.

Para superar el problema de la preimagen, basta recordar que los vectores base se pueden obtener de

$$v_k = X^T \alpha_k$$

donde α_k es el k-ésimo eigenvector de la matriz de Gram y X es la matriz que tiene como filas a los datos, siempre y cuando se utilice el kernel identidad (recuérdese que para el caso robusto es $v = X^T W \alpha$). Entonces, puede fácilmente encontrarse la reconstrucción de un dato x como

$$x^* = VV^T(x - \hat{\mu}) + \hat{\mu}$$

donde W es la matriz con los eigenvectores base y $\hat{\mu}$ es el estimador de la media.

Como se sabe de antemano que existe un 45 % de imágenes indeseables (que contienen personas) se ajustaron los parámetros de la función de influencia de forma tal que aproximadamente el 45 % de los datos con mayor distancia de Mahalanobis tuvieran disminuído su peso.

Al igual que en ([12]) se tomaron 20 vectores base para hacer la reconstrucción. La Figura (6-19) muestra los resultados para algunas de las imágenes. La columna de la izquierda es la imagen original, la siguiente columna es la reconstrucción usando PCA clásico (implementado como Kernel PCA con kernel identidad) donde se puede notar la presencia de fantasmas en la reconstrucción, hecho que indica que las imágenes con personas tuvieron una influencia en la generación de los vectores base. La tercera columna es la versión robusta (implementada como Kernel PCA Robusto con kernel identidad), en ella se puede ver que las personas son prácticamente ignoradas en la reconstrucción. Estos resultados son cualitativamente similares a los obtenidos en ([12]).

La Figura (6-20) muestra gráficas de diagnóstico (como las explicadas en capítulos anteriores) para la versión clásica y la versión robusta. En el eje horizontal se grafica la distancia sobre la base de los componentes principales, mientras que verticalemente se grafica la distancia ortogonal a la base de los componentes. Se etiquetaron individualmente las imágenes para saber dónde existían personas. La idea de éstas gráficas diagnóstico es observar que los datos típicos se concentran en la zona inferior izquierda, mientras que los outliers se deben encontrar en las zonas extrema superior (outliers ortogonales pero con proyección al centro de la nube de datos), extrema derecha (outliers con distancia grande sobre los componentes) y esquina superior derecha (outliers ortogonales y sobre los componentes, los más peligrosos). El número de componentes que se usó para proyectar y obtener estas gráficas fue aquel que proporcionara el 95 %

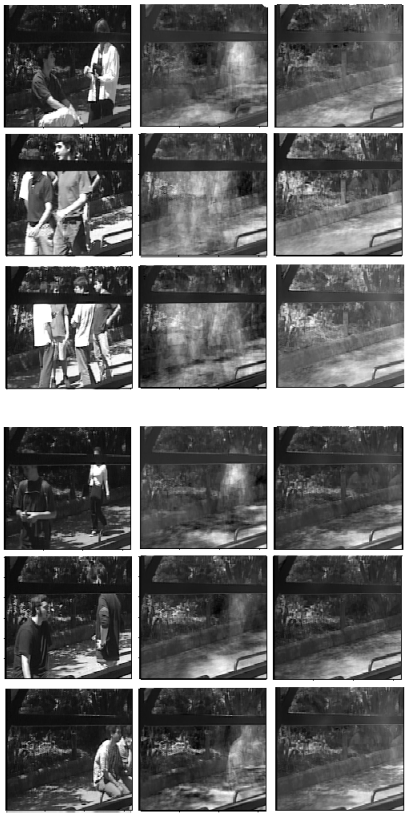

Figura 6-19: Reconstrucción de imágenes Kernel PCA y Robusto, ambos con el kernel identidad (que corresponde a PCA clásico). Columna izquierda: imagen original, siguiente columna: reconstrucción PCA, columna derecha: reconstrucción Kernel PCA Robusto.

de la varianza en ambos casos (debido a la ponderación, la variabilidad descrita para un número fijo de componentes robustos puede cambiar). Obsérvese que outliers que aparecen en Kernel PCA como outliers sobre los componentes (hacia la derecha de la gráfica) en la versión robusta también se separan ortogonalmente.

Aunque sería deseable observar una separación clara entre los puntos rojos y los negros (outliers y datos típicos), si se observa la Figura (6-21) se puede notar que aquellos datos marcados como outliers que se encuentran sobre la nube de observaciones típicas corresponden a personas que debido a la iluminación y contraste de la imagen apenas son perceptibles. Básicamente, su presencia pasa desapercibida, mientras que personas que son fácilmente visibles por la iluminación y el contraste con el fondo se detectan de manera bastante aceptable (y que son aquellas observaciones más dañinas).

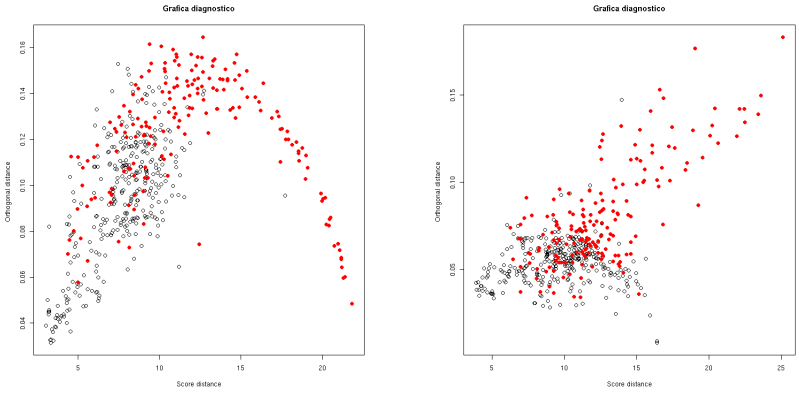

Figura 6-20: Gráficas diagnósticas para los componentes principales obtenidos mediante Kernel PCA (izquierda) y Kernel PCA Robusto (derecha). Los puntos marcados con rojo representan imágenes que fueron previamente etiquetadas como aquellas donde existen personas en la escena.

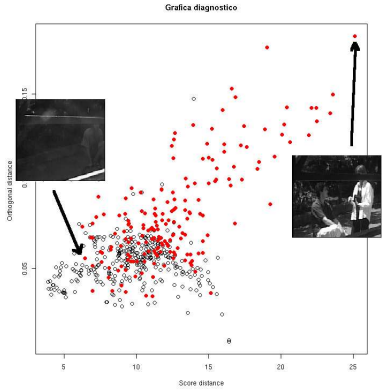

Figura 6-21: Gráfico diagnóstico para los componentes principales robustos. Los puntos marcados con rojo representan imágenes que fueron previamente etiquetadas como aquellas donde existen personas en la escena. Aquellos sobre la nube general de puntos son personas que casi no pueden distinguirse en las imágenes debido a la iluminación y el contraste.

6.4. Extracción de características para clasificación

En ([9]) y ([10]) se aplica Kernel PCA como *un extractor de características no-lineales*. Esta aplicación consiste en hacer de Kernel PCA una parte de un método supervisado (clasificación). Básicamente, esta aplicación consiste en realizar Kernel PCA utilizando un kernel predefinido (distinto al kernel polinomial de grado 1, que sería equivalente a PCA) a un conjunto de datos de forma tal que las proyecciones sobre los k primeros componentes principales sean las nuevas variables que representen a los datos. La Figura (6-22) esquematiza cómo la extracción nolineal de características funciona.

Para probar la utilidad de las características extraídas (las proyecciones sobre los componentes principales), éstas se utilizan para alimentar un clasificador lineal. En este caso, una máquina de soporte vectorial lineal con margen suave al igual que en ([9]) como en ([10]). En esas mismas publicaciones se utiliza el conjunto de datos USPS (descrito en la introducción de este capítulo) para la obtención de características y clasificación. Debido a la complejidad computacional de utilizar todo el conjunto de entrenamiento (7291 imágenes) los autores utilizan una muestra de estos datos de tamaño 3000 (lo que hace sumamente difícil reproducir sus resultados, al desconocerse exactamente los datos utilizados). Sin embargo, es un hecho conocido (reportado en ([9])) que en los datos predefinidos como de prueba del conjunto USPS *existen algunas pocas observaciones que son difíciles o imposibles de clasificar debido a errores de segmentación o mala asignación de etiquetas* ([9]). Así surge la posibilidad de explotar este hecho como prueba para el método propuesto en esta tesis. El experimento consiste aquí entonces en intercambiar los conjuntos de entrenamiento (7291 observaciones) y prueba (2007 observaciones) con el fin de explotar el hecho conocido de la existencia de observaciones atípicas en el conjunto de tamaño 2007; eliminando también la necesidad de submuestrear los datos y hacer los resultados fácilmente reproducibles. Al igual que en ([9]) se utilizó el kernel polinomial $k(x, y) =< x, y >^d$.

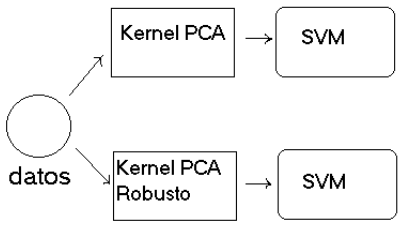

Figura 6-22: Esquema de la extracción de características para un esquema de clasificación de imágenes de dígitos. Figura tomada de ([9]).

Resultados clasificación						
# comp / d	2	3	4	5	6	7
16	6.9 / *6.9*	7.7 / *7.4*	8.1 / *8.1*	8.8 / *8.8*	10.6/ *10.5*	13.3/ *12.4*
32	6.1 / *5.6*	6.4 / *5.8*	6.6 / *6.5*	7.5 / *6.9*	7.9 / *7.6*	8.5 / *8.2*
64	5.5 / *5.4*	5.9 / *4.9*	6.4 / *5.8*	6.8 / *6.8*	7.3 / *7.3*	8.0 / *8.0*
128	5.4 / *5.3*	4.8 / *4.7*	**5.0** / *5.1*	6.2/ *5.9*	7.5 / *7.3*	**8.5** / *8.7*

Cuadro 6.1: Resultados de extracción de caracteriísticas versión no-robusta y *robusta*. Las columnas indican para cada grado del kernel polinomial la versión no-robusta / versión robusta (y filas el número de componentes) el porcentaje de error de clasificación sobre los datos de prueba. En negritas se muestra los valores en los cuales la versión no-robusta tuvo un menor error de prueba (dos casos).

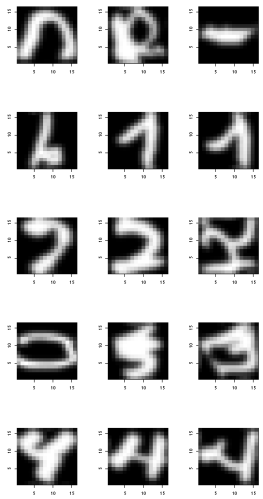

Figura 6-23: Outliers en el conjunto USPS identificados. Cada fila representa tres observaciones de los dígitos del 0 al 4.

Se aplicó el método propuesto por separado a cada una de las clases correspondientes a los 10 dígitos, con esto se pretende eliminar observaciones que sean atípicas al dígito en cuestión. Las Figuras (6-23) y (6-24) muestran algunas de las observaciones con mayor distancia de Mahalanobis para cada una de las clases usando el kernel polinomial de grado 3. Con el método propuesto se asignó peso cero al uno por ciento de aquellas observaciones con mayor distancia de Mahalanobis de cada dígito. Después, considerando todas las clases juntas, se procedió a la obtención de características y clasificación. Se entrenó una máquina de soporte vectorial lineal (con kernel identidad, el parámetro de costo para clasificar datos linealmente separables se fijó en $C = 10$) tanto a las características robustas como a las caracterísitcas obtenidas por Kernel PCA no-robusto.

Efectivamente, debido a la capacidad de el método propuesto en esta tesis para identificar (algunas de) aquellas observaciones que se alejan más de la calidad de ser

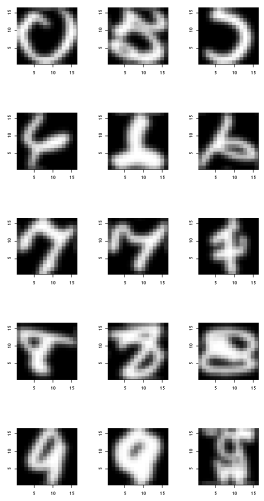

Figura 6-24: Outliers en el conjunto USPS identificados. Cada fila representa tres observaciones de los dígitos del 5 al 9.

el número i se puede obtenerse un menor error de prueba. La Tablas (6.1) presenta algunos resultados numéricos comparando el método no-robusto y robusto respectivamente para diferentes grados del kernel polinomial y de número de componentes. Aunque la ganancia es marginal, en aplicaciones de clasificación tan elaboradas como esta cualquier disminución en el error de prueba es aceptable. En dos situaciones el método no-robusto obtiene un menor error de prueba. A esto se puede mencionar que, si es que existen outliers en el conjunto de prueba (y que sean parecidos a los eliminados del conjunto de entrenamiento) muy posiblemente serán mal clasificados por el método robusto.

Capítulo 7

Conclusiones y Perspectivas

Hoy en día existe un gran número de aplicaciones donde la dimensión de los datos N supera al número de observaciones disponible m. La mayoría de los métodos se orienta claramente hacia la situación donde $m >> N$, con muchos resultados basados en situaciones limítrofes donde $m \to \infty$ con N fijo. Surge entonces la necesidad de métodos apropiados para manejar las aplicaciones de hoy en día. Tal es el caso de Kernel PCA, que tiene como filosofía realizar una transformación implícita de los datos tal que la dimensión puede ser arbitrariamente mayor al número de ejemplos. Lo presentado en esta tesis es un acercamiento a enfrentar este tipo de problemas incluyendo aspectos de la distribución de los datos en métodos provenientes del aprendizaje máquina.

Kernel PCA - una extensión nolineal de PCA - tiene profundas ventajas: en especial si el número de observaciones es menor a su dimensión, es capaz de vencer la maldición de la dimensionalidad, ya que aunque la búsqueda de componentes principales se haga en espacio de dimensión muy alta las soluciones siempre están en el espacio generado por los datos.

En este trabajo se propuso y estudió un método para realizar Kernel PCA de manera robusta. PCA es una técnica muy estudiada en el sentido de la robustez, existen numerosos trabajos al respecto y aquí se pretendió brindar un panorama muy general sobre ellos. Dado que Kernel PCA es una generalización directa de PCA, se buscó en los métodos de PCA robusto los aquellos que fueran asequibles para aplicar el *truco del kernel*. Finalmente, uno de los más sencillos fue lo propuesto y se demostró cómo incluir en Kernel PCA los estimadores ponderados de covarianza y media, así también la forma en que la distancia de Mahalanobis se puede kernelizar. A través de diversos experimentos se observó y comparó el comportamiento del método con la versión no-robusta de Kernel PCA. Se espera que lo propuesto en esta tesis presente al practicante que desee realizar Kernel PCA una opción para obtener mejores resultados y al investigador nuevos temas de reflexión en cuanto al análisis de datos moderno.

Apéndice A

Algunos Fundamentos Teóricos

A.1. Fundamentos relacionados con Kernel PCA

La siguientes dos definiciones son necesarias para entender cómo es que es posible considerar un sistema de ecuaciones alternativo a $Cv = \lambda v$.

Definición A.1.1 (Ortonormalizacion de Gram-Schmidt) *Supóngase que* $\{v_i\}_\Lambda$ *es un conjunto linealmente independiente de vectores en un espacio* H*. Una base ortonormal* $e_1, e_2, \ldots$ *puede ser construida como*

$$e_1 := v_1/\|v_1\|,$$

$$e_2 := (v_2 - P_1 v_2)/\|v_2 - P_1 v_2\|,$$

$$e_3 := (v_3 - P_2 v_3)/\|v_3 - P_2 v_3\|,$$

$$\vdots$$

donde P_n *es tal que*

$$P_n x := \sum_{i=1}^{n} (e_i \cdot x) e_i.$$

Si $v_1, v_2, \ldots$ *no son linealmente independientes puede ser que* $v_{n+1} - P_n v_{n+1}$*, en este caso no se considera* v_{n+1} *y se procede con* v_{n+2}*, recorriendo todos los indices.*

Definición A.1.2 (Expansiones y relación de Parseval) *Sea* e_i *una base ortonormal en* H*, entonces para cada* x *en* H

$$x = \sum_i \langle e_i, x \rangle \, e_i$$

y

$$\|x\|^2 = \sum_i \langle e_1, x\rangle^2 .$$

La posibilidad de trabajar con un nuevo conjunto de ecuaciones presentado en el capítulo donde se formula Kernel PCA se justifica a continuación:

Proposición A.1.1 (Nuevo conjunto de ecuaciones para Kernel PCA) *Considerese que $x_1, x_2, \ldots, x_m$ son ortonormales (si no lo son, aplicar la ortonormalizacion de Gram-Schmidt y construir $e_1, e_2, \ldots, e_n$). Estos $e_1, e_2, \ldots, e_n$ son una base de x_i, pero tambien cada e_i puede ser escrito como una combinación lineal de x_j (por construcción).*

Por lo tanto

$$\langle x_n, v_1\rangle = \langle x_n, v_2\rangle , \quad \textit{para todo } n = 1, \ldots, m$$

donde

$$v_1 = \lambda v, \ \textit{y} \ v_2 = Cv$$

es equivalente a la correspondiente aseveracion sobre el conjunto ortonormal $e_1, e_2, \ldots, e_n$.

En el caso ortonormal, la relación de Parseval aplicada al complemento del generado de x_i, implica que si se reemplaza $x = v_1 - v_2$ entonces

$$\|v_1 - v_2\| = \sum_{i=1}^{m} (\langle x_i, v_1\rangle - \langle x_i, v_2\rangle)^2.$$

Entonces $v_1 = v_2$ si y solo si

$$\langle x_n, v_1\rangle = \langle x_n, v_2\rangle , \quad \textit{para todo } n = 1, \ldots, m.$$

Cuando la distancia de Mahalanobis se quiere calcular en $\mathcal{H}$ utilizando estimadores ponderados, es necesario considerar la siguiente observación:

Observación A.1.1 (Distancia de Mahalanobis para datos no centrados) *Para el caso general con estimadores ponderados se tiene que*

$$d = m \sum_{k=1}^{m} \frac{(\tilde{\phi}(x) \cdot \tilde{u_k})^2}{\lambda_K}$$

y

$$\tilde{\phi}(x) \cdot \tilde{u_k} = \sum_{i=1}^{m} \alpha_i^k \tilde{\phi}(x) \cdot (\omega_i \tilde{\phi}(x_i))$$

Entonces

$$\tilde{\phi}(x)\cdot\tilde{u_k}=\sum_{i=1}^{m}\alpha_i^k(\phi(x)-\frac{\sum_{m=1}^{m}\omega_m\phi(x_m)}{\sum_{p=1}^{m}\omega_p})\cdot\omega_i(\phi(x_i)-\frac{\sum_{n=1}^{m}\omega_n\phi(x_n)}{\sum_{p=1}^{m}\omega_p})$$

$$=\sum_{i=1}^{m}\alpha_i^k\omega_i(\phi(x)\cdot\phi(x_i)-\frac{\sum_{n=1}^{m}\omega_n\phi(x)\cdot\phi(x_n)}{\sum_{p=1}^{m}\omega_p}-\frac{\sum_{m=1}^{l}\omega_m\phi(x_m)\cdot\phi(x_i)}{\sum_{p=1}^{l}\omega_p}+$$

$$+\frac{\sum_{m=1}^{l}\sum_{n=1}^{l}\omega_n\omega_m\phi(x_n)\cdot\phi(x_m)}{(\sum_{p=1}^{l}\omega_p)^2}).$$

Que en términos del kernel centrado con media robusta se expresa como

$$\tilde{\phi}(x)\cdot\tilde{u_k}=\sum_{i=1}^{l}\alpha_i^k\omega_i\tilde{k}(x,x_i).$$

O bien

$$\tilde{\phi}(x)\cdot\tilde{u_k}=\sum_{i=1}^{l}\beta_i^k\tilde{k}(x,x_i).$$

A.2. Kernels y sus propiedades

Algunas definiciones básicas y resultados se presentan a continuación. Se entiende que los índices i, j corren desde 1 a m, donde m es el número de observaciones de la muestra.

Definición A.2.1 (Matriz de Gram) *Dada una función $k:\mathcal{X}^2\to\mathbb{K}$ (donde $\mathbb{K}=\mathbb{C}$ o $\mathbb{K}=\mathbb{R}$) y observaciones $x_1,\cdots,x_m\in\mathcal{X}$, la matriz de $m\times m$ K con elementos*

$$K_{ij}=k(x_i,x_j)$$

se conoce como matriz de Gram (o matriz de kernel) de k con respecto a $x_1,\cdots,x_m$.

Definición A.2.2 (Kernel (Semi Positivo Definido)) *Sea $\mathcal{X}$ un conjunto no-vació. Una función k en $\mathcal{X}\times\mathcal{X}$ para la cual $m\in\mathbb{N}$ genera una matriz semi definida positiva de Gram es llamado(a) un kernel semi positivo definido. Al que comúnmente, se referirá simplemente como* kernel.

La definición de kernels semi positivos definidos y matrices semi positivas definidas difieren en el hecho de que para los kernels se tiene la libertad de elegir los puntos en los cuales evaluar el kernel - para toda elección, el kernel induce una matriz semi positiva definida -.

El hecho de un kernel ser semi positivo definido implica no-negatividad en la diagonal

$$k(x, x) \geq 0, \text{ para todo } x \in \mathcal{X}$$

y simetría

$$k(x_i, x_j) = \overline{k(x_j, x_i)}.$$

donde se incluye el caso de valores complejos, al representar la simetría el complejo conjugado.

Observación A.2.1 (Terminología) *El término kernel se deriva del uso de este tipo de funciones en el área de operadores integrales como los estudiados por Hilbert y otros. Una función k la cual genera un operador T_k mediante*

$$(T_k f)(x) = \int_{\mathcal{X}} k(x, x\prime) f(x\prime) dx\prime$$

es llamado el kernel de T_k.

Como ejemplos de kernels comúnmente utilizados se tienen:

- Kernel polinomial: $k(x, y) = \gamma(x \cdot y + c)^d$
- Kernel de base radial: $k(x, y) = exp(-\|x - y\|^2/(2\sigma^2))$
- Kernel sigmoide: $k(x, y) = tanh(\kappa(x \cdot y) + \Theta)$

Las demostraciones de las siguientes propiedades se puede encontrar en ([10]). Si k_1 y k_2 son kernels entonces:

- $k_1 + k_2$ es un kernel.
- ck_1 es un kernel si $c > 0$.
- $c_1 k_1 + c_2 k_2$ es un kernel si $c_1 > 0,\ c_2 > 0$.

Las siguientes definiciones ayudarán a justificar los pasos necesarios que se dan para introducir Kernel PCA Robusto.

Definición A.2.3 (Matriz Condicionalmente Positiva Definida) *Una matriz simétrica de $m \times m$ K $(m \geq 2)$ tomando valores en $\mathbb{K}$ y que satisface*

$$\sum_{i,j=1}^{m} c_i \overline{c_j} K_{ij} \geq 0$$

para todo $c_i \in \mathbb{K}$, con $\sum_{i=1}^{m} c_i = 0$, se llama condicionalmente semi positiva definida.

Definición A.2.4 (Kernel Condicionalmente Positivo Definido) *Sea $\mathcal{X}$ un conjunto no-vacío. Una función $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$ para la cual todo $m \geq 2$, $x_1, \cdots, x_m \in \mathcal{X}$ genere una matriz condicionalmente semi positiva definida se conoce como kernel condicionalmente semi positivo definido.*

La siguiente demostración se puede encontrar también en ([10]).

Proposición A.2.1 *Sea $x_0 \in \mathcal{X}$, y sea k un kernel simétrico en $\mathcal{X} \times \mathcal{X}$. Entonces*

$$\tilde{k}(x, x\prime) = \frac{1}{2}(k(x, x\prime) - k(x, x_0) - k(x_0, x\prime) + k(x_0, x_0))$$

es semi positivo definido si y sólo si k es condicionalmente semi positivo definido.

Bibliografía

[1] N. A. Campbell. Robust procedures in multivariate analysis I: robust covariance estimation. *Appl. Statist.*, 29(3):231–237, 1980.

[2] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics. The Approach Based On Influence Functions.* Wiley Series In Probability And Mathematical Statistics. John Wiley & Sons, 1985.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements Of Statistical Learning.* Springer-Verlag, 2001.

[4] P. J. Huber. *Robust Statistics.* Wiley Series In Probability And Mathematical Statistics. John Wiley & Sons, 1981.

[5] M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 2005.

[6] J. E. Jackson. *A User's Guide To Principal Components.* Wiley Series In Probability And Mathematical Statistics. John Wiley & Sons, 1991.

[7] I. T. Jolliffe. *Principal Component Analysis.* Springer Series In Statistics. Springer-Verlag, 1986.

[8] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* Wiley Series In Probability And Mathematical Statistics. John Wiley & Sons, 1987.

[9] B. Schölkopf and A. Smola. *Learning With Kernels.* MIT Press, 2002.

[10] B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[11] J. Shawe-Taylor and N. Cristianini. *Kernel Methods For Pattern Analysis.* Cambridge University Press, 2004.

[12] F. Torre and M. Black. Robust principal component analysis for computer vision, 2001.