Revista Colombiana de Estadística
Junio 2008, volumen 31, no. 1, pp. 19 a 40

Introducción a kernel ACP y otros métodos espectrales aplicados al aprendizaje no supervisado

Introduction to Kernel PCA and other Spectral Methods Applied to Unsupervised Learning

Luis Gonzalo Sánchez[a], Germán Augusto Osorio[b], Julio Fernando Suárez[c]

Departamento de Matemáticas y Estadística, Facultad de Ciencias Exactas y Naturales, Universidad Nacional de Colombia, Manizales, Colombia

Resumen

En el presente trabajo, se introducen las técnicas de kernel ACP (KACP) y conglomeramiento espectral con algunos ejemplos ilustrativos. Se pretende estudiar los efectos de aplicar ACP como preproceso sobre las observaciones que se desean agrupar, para lo cual se hacen experimentos con datos reales. Entre las tareas adicionales que requieren estos procedimientos está la sintonización de parámetros (ajuste de valores); el alineamiento del kernel se presenta como alternativa de solución. La técnica de alineamiento del kernel presenta buenos resultados al contrastar las curvas de alineamiento con los índices de Rand obtenidos para los datos evaluados. Finalmente, el estudio muestra que el éxito de ACP depende del problema y que no se tiene un criterio general para decidir.

***Palabras clave*:** método kernel, análisis de conglomerados, teoría de grafos, selección de modelo.

Abstract

In this work, the techniques of Kernel Principal Component Analysis (Kernel PCA or KPCA) and Spectral Clustering are introduced along with some illustrative examples. This work focuses on studying the effects of applying PCA as a preprocessing stage for clustering data. Several tests are carried out on real data to establish the pertinence of including PCA. The use of these methods requires of additional procedures such as parameter tuning; the kernel alignment is presented as an alternative for it. The results of kernel alignment expose a high level of agreement between the tuning curves their respective Rand indexes. Finally, the study shows that the success of PCA is problem-dependent and no general criteria can be established.

***Key words*:** Kernel method, Cluster analysis, Model selection, Graph theory.

[a]Estudiante de maestría. E-mail: lgsanchezg@unal.edu.co
[b]Estudiante de maestría. E-mail: gaosorioz@unal.edu.co
[c]Profesor asociado. E-mail: jfsuarezc@unal.edu.co

1. Introducción

El análisis de componentes principales (ACP) y la búsqueda de conglomerados están entre las técnicas de mayor uso en el análisis exploratorio de datos. Estas han sido aplicadas a una variedad de problemas de las ciencias sociales, la biología, la visión artificial, entre muchos otros. Básicamente, el ACP se enfoca en la representación de los datos reduciendo su dimensión. Este hecho da lugar a una cantidad de propiedades interesantes (*i.e.* la reducción de ruido), que también pueden aprovecharse (Jolliffe 2002). Aunque, ACP no supone conocimiento previo sobre la distribución de los datos, funciona mejor sobre cierto tipo de configuraciones. Diferentes extensiones no lineales de este método han sido propuestas a lo largo de los años, y una de las aproximaciones que ha recibido mayor acogida está enmarcada en los métodos kernel, que presentan una solución aparentemente simple, pero conceptualmente amplia.

El análisis de conglomerados, o *clustering*, intenta determinar si la muestra que se tiene está conformada por elementos de diferentes poblaciones a través de la búsqueda de agrupaciones de puntos. Determinar la estructura del grupo no es trivial porque no existe información previa sobre su forma. Entre los métodos de conglomeramiento, el algoritmo de k-medias ha sido, tal vez, el más difundido. Recientemente, los algoritmos de conglomeramiento basados en las propiedades espectrales de los grafos han ganado auge por su fácil implementación al utilizar conceptos de álgebra lineal que son de uso extensivo.

En este trabajo se estudian los efectos de aplicar el análisis de componentes principales lineales y no lineales (utilizando kernel ACP) como etapa previa a la búsqueda de conglomerados. Diferentes consideraciones son de especial atención al aplicar los algoritmos; por ejemplo, la sintonización de los parámetros cuando no existe un punto de referencia *ground truth* sobre el cual ajustar variables. Por tanto, se estudia el método de alineamiento del kernel como posible alternativa al problema.

El trabajo está desarrollado de la siguiente forma: breve introducción teórica de los métodos aplicados y ejemplos que ayudan a su comprensión básica, pruebas realizadas sobre conjuntos de datos reales para establecer si la combinación de técnicas refleja mejoras en la obtención de resultados, y comentarios y conclusiones. Todos los algoritmos fueron desarrollados en MatLab®.

2. Análisis de componentes principales (ACP)

El análisis de componentes principales es una técnica de representación de datos enfocada a la reducción de dimensión. El ACP ha sido la tendencia dominante para el análisis de datos en un gran número de aplicaciones. Su atractivo recae en la simplicidad y capacidad de reducción de dimensión, minimizando el error cuadrático de reconstrucción producido por una combinación lineal de variables latentes, conocidas como componentes principales, las cuales se obtienen a partir de una combinación lineal de los datos originales. Los parámetros del modelo pueden calcularse directamente de la matriz de datos centralizada $\mathbf{X}$, bien sea

por descomposición en valores singulares o por la diagonalización de la matriz de covarianza (positiva semidefinida) (Jolliffe 2002). Sea $\mathbf{x}_i$ el i-ésimo vector de observación (vector columna) de tamaño c, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$. La matriz de rotación $\mathbf{U}$ permite calcular las p componentes principales $\mathbf{z}$ que mejor representan $\mathbf{x}$.

$$\mathbf{z} = \mathbf{U}^T\mathbf{x} \tag{1}$$

$\mathbf{U}$ puede obtenerse al solucionar un problema de valores propios, y está definida como los p mayores vectores propios de $\mathbf{X}^T\mathbf{X}$, esto es,

$$\mathbf{X}^T\mathbf{X}\mathbf{U} = n\mathbf{U}\mathbf{\Lambda} \tag{2}$$

La matriz $\mathbf{X}^T\mathbf{X}$ está asociada a la matriz de covarianza $C = \frac{1}{n}\mathbf{X}^T\mathbf{X}$; además, puede calcularse como[1]

$$C = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T \tag{3}$$

El problema de valores propios $C\mathbf{u} = \lambda\mathbf{u}$ implica que todas las soluciones de $\mathbf{u}$ deben estar en el espacio generado por el conjunto de vectores $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$; por lo cual (Schölkopf et al. 1996),

$$\lambda\langle\mathbf{x}_i, \mathbf{u}\rangle = \langle\mathbf{x}_i, C\mathbf{u}\rangle, \quad \forall i = 1, \ldots, n. \tag{4}$$

A continuación se describe una adaptación del procedimiento anterior cuando la reducción del espacio no se hace sobre los datos originales sino sobre un mapeo o transformación de los mismos.

3. Kernel ACP

Es posible sustituir el espacio original de las observaciones $\mathcal{X}$, que en general corresponde a $\mathbb{R}^p$, por un espacio provisto de producto punto $\mathcal{H}$ mapeado a través de $\boldsymbol{\phi} : \mathcal{X} \mapsto \mathcal{H}$ (Schölkopf & Smola 2002). Partiendo de la misma suposición sobre la cual se construyó la matriz de covarianza C, la cual implica que los datos están centralizados en $\mathcal{H}$, procedemos a construir la matriz de covarianza en el nuevo espacio:

$$C_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_i)^T \tag{5}$$

Si $\mathcal{H}$ es de dimensión infinita, se puede pensar de $\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_i)^T$ como el operador lineal que transforma $h \in \mathcal{H}$ a $\boldsymbol{\phi}(\mathbf{x}_i)\langle\boldsymbol{\phi}(\mathbf{x}_i), h\rangle$. En este caso debemos encontrar los valores propios no nulos ($\lambda > 0$) y sus respectivos vectores propios $\mathbf{u}_{\mathcal{H}}$ que satisfacen

$$\lambda\mathbf{u}_{\mathcal{H}} = C_{\mathcal{H}}\mathbf{u}_{\mathcal{H}} \tag{6}$$

[1]En realidad se calcula el estimado de la matriz de covarianza de los datos.

De la misma forma, las soluciones de $\mathbf{u}_{\mathcal{H}}$ deben estar dentro del espacio generado por $\{\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)\}$. Entonces,

$$\lambda\langle\boldsymbol{\phi}(\mathbf{x}_k), \mathbf{u}_{\mathcal{H}}\rangle = \langle\boldsymbol{\phi}(\mathbf{x}_k), C_{\mathcal{H}}\mathbf{u}_{\mathcal{H}}\rangle, \quad \forall k = 1, \ldots, n \tag{7}$$

Además, es posible definir los vectores propios en términos de los datos mapeados en $\mathcal{H}$:

$$\mathbf{u}_{\mathcal{H}} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{x}_i) \tag{8}$$

Combinando (7) y (8), se obtiene

$$\lambda \sum_{i=1}^{n} \alpha_i \big\langle \boldsymbol{\phi}(\mathbf{x}_k), \boldsymbol{\phi}(\mathbf{x}_i) \big\rangle = \frac{1}{n} \sum_{i=1}^{n} \alpha_i \Big\langle \boldsymbol{\phi}(\mathbf{x}_k), \sum_{j=1}^{n} \boldsymbol{\phi}(\mathbf{x}_j) \Big\rangle \big\langle \boldsymbol{\phi}(\mathbf{x}_j), \boldsymbol{\phi}(\mathbf{x}_i) \big\rangle, \qquad \forall k = 1, \ldots, n \tag{9}$$

Definiendo la matriz K como $k_{ij} = \big\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \big\rangle$, se obtiene

$$n\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha} \tag{10}$$

donde $\boldsymbol{\alpha}$ denota el vector columna que sintetiza la representación de $\mathbf{u}_{\mathcal{H}}$ dada en (8), a través del conjunto de observaciones mapeadas por $\boldsymbol{\phi}$. Debido a la simetría de K, sus vectores propios generan el espacio completo; por tanto

$$n\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \tag{11}$$

genera las soluciones de la ecuación (10). De esta forma, los valores propios asociados a $\boldsymbol{\alpha}$ corresponden a $n\lambda$; en consecuencia, cada uno de los $\mathbf{u}_{\mathcal{H}}$ corresponden al mismo ordenamiento de los $\boldsymbol{\alpha}$. Es necesario trasladar la restricción de $\|\mathbf{u}_{\mathcal{H}}\| = 1$ a los correspondientes vectores propios de K:

$$1 = \sum_{i,j=1}^{n} \alpha_i \alpha_j \big\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \big\rangle = \lambda\langle\boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle \tag{12}$$

Para la extracción de componentes principales, deben proyectarse los datos mapeados a $\mathcal{H}$ sobre los respectivos vectores propios seleccionados. Podemos hacer uso de

$$\big\langle \mathbf{u}_{\mathcal{H}}, \boldsymbol{\phi}(\mathbf{x}) \big\rangle = \sum_{i=1}^{n} \alpha_i \big\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \big\rangle \tag{13}$$

La centralización de los datos es posible al reemplazar la matriz K por su correspondiente versión centralizada:

$$\widetilde{K} = K - \mathbf{1}_n K - K\mathbf{1}_n + \mathbf{1}_n K \mathbf{1}_n \tag{14}$$

donde $\mathbf{1}_n$ es una matriz cuadrada de tamaño $n \times n$ cuyas entradas son $1/n$. Para los m datos de prueba en el vector $\mathbf{t}$ que deben ser mapeados se tiene:

$$k_{ij}^{test} = \left\langle \boldsymbol{\phi}(\mathbf{t}_i), \boldsymbol{\phi}(\mathbf{x}_j) \right\rangle \tag{15}$$

y su versión centralizada:

$$\widetilde{K}^{test} = K^{test} - \mathbf{1}'_n K - K^{test}\mathbf{1}_n + \mathbf{1}'_n K \mathbf{1}_n \tag{16}$$

donde $\mathbf{1}_n$ es una matriz de tamaño $m \times n$ cuyas entradas son $1/m$.

3.1. Algunos kernels

Ya se ha planteado el problema de ACP en términos de los productos punto entre las observaciones mapeadas a un espacio $\mathcal{H}$. Ahora bien, no es necesario hacer explícito este mapeo ϕ; en vez de esto, podemos usar una función $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ la cual, en ciertas condiciones, representa el producto punto en $\mathcal{H}$, es decir, $k(x, x') = \langle \phi(x), \phi(x') \rangle$. Entre las funciones kernel más comunes están:

- **Kernel lineal:** corresponde al producto punto en el espacio de entrada:

$$k(x, x') = \langle x, x' \rangle \tag{17}$$

- **Kernel polinomial:** representa la expansión a todas las combinaciones de monomios de orden d, y está dado por

$$k(x, x') = \langle x, x' \rangle^d \tag{18}$$

- **Kernel gaussiano:** está contenido dentro de las funciones de base radial:

$$k(x, x') = \exp\left(- \frac{\|x - x'\|^2}{2\sigma^2} \right) \tag{19}$$

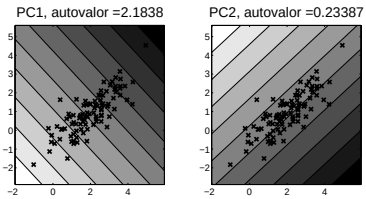
- **Kernel hiperbólico:** está asociado a las funciones de activación de las redes neuronales:

$$k(x, x') = \tanh\left(\xi \langle x, x' \rangle + b\right) \tag{20}$$

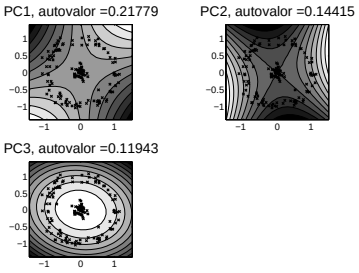
  Los parámetros $\xi > 0$ y $b < 0$ denotan escala y corrimiento, respectivamente.

En la figura 1 se presentan algunos ejemplos de los mapeos que se pueden lograr al aplicar ACP con la representación del producto punto utilizando las funciones kernel descritas. Las intensidades de gris en los planos dibujados son proporcionales a los valores que toma cada componente principal en cada punto evaluado del plano; por ejemplo, en la figura 1(a) el plano izquierdo corresponde a la primera componente principal utilizando el kernel lineal, lo cual es equivalente al ACP convencional. Note que los niveles de gris están asociados a las proyecciones ortogonales de todos los puntos del plano sobre el mayor eje principal de la elipsoide que contiene los datos (puntos negros). En la figura 1(b) se observa el efecto de

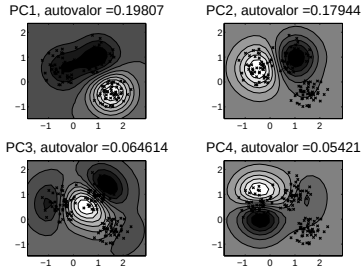
mapeo no lineal al aplicar un producto punto modificado, en particular, el kernel polinomial de orden 2; en este ejemplo, es fácil apreciar que el mapeo no lineal corresponde a la proyección de los puntos ubicados en el plano a un cono en $\mathbb{R}^3$. La primera y segunda componentes son aproximadamente proyecciones de los puntos en $\mathbb{R}^3$ al plano cuyo vector normal es paralelo al eje principal del cono; por tanto, la tercera componente tiene la dirección del eje principal del cono. Los casos de las figuras 1(c) y 1(d) son de naturaleza más abstracta y se pueden ver como mapeos no lineales a espacios de funciones (dimensión infinita). Es posible observar que el kernel gaussiano descompone la estructura de agrupaciones presente en los datos del la figura 1(c) y el kernel hiperbólico trata de desdoblar la estructura de variedad no lineal presente en los puntos de la figura 1(d).



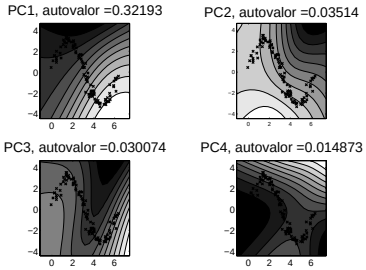
(a) Kernel lineal



(b) Kernel polinomial de orden 2



(c) Kernel gaussiano $\sigma = 0.5$



(d) Kernel hiperbólico $\xi = 0.08$ y $b = -\pi/2$

FIGURA 1: Ejemplos de mapeos conseguidos con KACP para diferentes configuraciones de los datos y diferentes kernel

A continuación se presentan las técnicas de conglomeramiento que se desea utilizar en conjunto con ACP (lineal y no lineal).

4. Conglomeramiento por *k*-medias

En el caso del algoritmo de k-medias, cada una de las k agrupaciones de observaciones que se desea revelar está representada por un punto μ_l para $l = 1, \ldots, k$, denominado centroide. Cada una de las observaciones x_i de la muestra se asigna al grupo de puntos C_l para el que se cumple que $\min_l (\text{dist}(x_i,\ \mu_l))$ (Peña 2002).

En esta aproximación se pretende minimizar la suma de las distancias cuadráticas promedio entre los puntos pertenecientes a una partición y el centroide que la define.

$$SC = \sum_{l=1}^{k} \frac{1}{M_l} \left(\sum_{\{i|\ x_i \in C_l\}} \mathrm{dis}(x_i, \mu_l)^2 \right) \tag{21}$$

donde M_l es el número de puntos que conforman C_l. Una de las distancias cuadráticas más comunes es $\|x_i - \mu_l\|_2^2$.

El algoritmo de k-medias se puede dividir en dos fases:

- La primera parte se conoce como reasignación por lotes, en la cual, partiendo de un conjunto de centroides establecidos por algún criterio de inicialización (por ejemplo: puntos aleatorios de la muestra, las k observaciones de la muestra más alejadas entre ellas, etc), se agrupan los puntos a la vez de acuerdo con las distancias a los centroides y se recalculan como la media de los puntos asignados al mismo grupo. El procedimiento se itera hasta que las medias no cambian y, por tanto, la partición generada por las distancias a las medias se mantiene constante.

- En seguida, se trata de reasignar cada uno de los puntos a otro conglomerado, tal que al recalcular μ_l (21), este sea menor que el actual. Este proceso se repite hasta que ninguno de los puntos tenga que reasignarse, lo que asegura un mínimo local para SC.

5. Conglomeramiento espectral

Antes de introducir los algoritmos de conglomeramiento espectral, es necesario dar algunos conceptos asociados a su construcción, como la representación con grafos, en particular los grafos de similitud.

5.1. Preliminares sobre los grafos

Dado un conjunto de observaciones $x_1, \ldots, x_n$ y alguna noción de similitud s_{ij} entre pares de puntos x_i y x_j, una forma intuitiva de agrupar las observaciones es unir los puntos en diferentes subconjuntos disyuntos, tal que la similitud entre puntos pertenecientes a diferentes subconjuntos sea baja, mientras que para pares de puntos pertenecientes a un mismo subconjunto se mantenga un alto grado de similitud. Partiendo solo de estos elementos (el conjunto y la función de similitud), podemos representar la muestra a través de un *grafo de similitud* $G = (V, E)$ que conecta el conjunto de vértices V mediante un conjunto de aristas E. Un grafo está conformado por los vértices v_i, los cuales representan los puntos x_i, y las conexiones entre ellos, conocidas como aristas del grafo, las cuales están pesadas por w_{ij}, si s_{ij} toma valores positivos o mayores que cierto umbral. Con esta definición del grafo de similitud es posible construir un criterio de agrupamiento con base en la partición del grafo G (Luxburg 2006). Sea $G = (V, E)$ un grafo no dirigido, es

decir $w_{ij} = w_{ji}$, la matriz de adyacencia $W = (w_{ij})_{i,j=1,\ldots,n}$ representa las aristas que conectan los vértices v_i y v_j cuando $w_i j = 0$, se dice que los vértices no están conectados. El grado de un vértice v_i está dado por

$$d_i = \sum_{j=1}^{n} w_{ij}$$

La matriz D de grados está definida como la matriz diagonal cuyos elementos son los grados $d_i, \ldots, d_n$. Dado un subconjunto de vértices $A \subset V$, denotando su complemento $\overline{A} = V \setminus A$, se define el vector indicador $\mathbf{1}_A = (f_1, \ldots, f_n)^T \in \mathbb{R}^n$ como aquel cuyas entradas $f_i = 1$, si $v_i \in A$, y $f_i = 0$ de otra forma. Pueden considerarse dos formas de medir el tamaño de un subconjunto $A \subset V$:

$$|A| := \text{el número de vértices en } A$$
$$\text{vol}(A) := \sum_{\{i|\ v_i \in A\}} d_i$$

Un subconjunto $A \subset V$ de un grafo está conectado si cualquier par de vértices en A puede juntarse a través de un camino de vértices conectados que también pertenezcan a A. Un subconjunto A se denomina componente conectado si no existen conexiones entre los vértices de A y $\overline{A}$. Los conjuntos $A_1, \ldots, A_k$ forman una partición del grafo G, si $A_i \cap A_j = \varnothing$ y $A_1 \cup \cdots \cup A_k = V$.

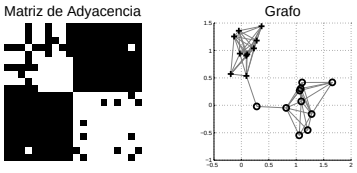
5.2. Grafos de similitud

Existen diferentes concepciones sobre cómo transformar un conjunto de puntos $x_1, \ldots, x_n$, asociados a una función de similitud o distancia en un grafo. El objetivo principal es modelar las relaciones en las vecindades de los puntos. En este trabajo se consideran los siguientes tipos de grafo:

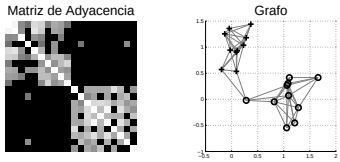
- **El ϵ-vecindario.** Todas las parejas de puntos cuya distancia sea menor a un umbral ϵ son conectados en el grafo. Se dice que este grafo es no pesado, ya que la matriz de adyacencia toma valores de 0 o 1.
- **Los k-vecinos más cercanos.** Se conectan al vértice v_i las k parejas más cercanas a este. Note que esta situación genera un grafo dirigido; por tanto, si se quiere que la matriz de adyacencia sea simétrica, se debe aplicar otro criterio para la generación de conexiones. Una forma consiste en conectar los vértices v_i y v_j si uno de los puntos, x_i o x_j, está entre los k vecinos más cercanos del otro. La segunda forma consiste en unir los vértices v_i y v_j, solo si x_i es uno de los vecinos más cercanos de x_j, y viceversa. La primera aproximación se denomina *grafo de los k-vecinos más cercanos*; la segunda, *grafo de los k-vecinos más cercanos mutuos*. Las conexiones están ponderadas con el valor de la similitud s_{ij}.
- **El grafo completamente conectado.** En este caso, simplemente se conectan todos los vértices, es decir, cualquier vértice del grafo está conectado

directamente con el resto de vértices. Una medida de similitud recomendable podría asociarse al kernel gaussiano porque esta función define vecindarios de manera implícita.

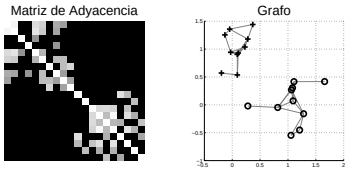
La figura 2 presenta las diferentes configuraciones de grafos y sus matrices de adyacencia asociadas; los puntos más claros sobre la matriz de adyacencia indican pesos más grandes en las conexiones entre pares de vértices del grafo, mientras que los puntos totalmente negros corresponden a la ausencia de conexiones entre los pares de vértices.



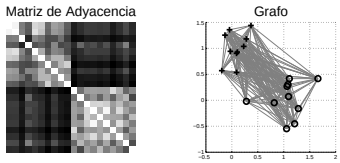
(a) ϵ-vecindario para $\epsilon = 0.7$.



(b) k-vecinos más cercanos con $k = 5$.



(c) k-vecinos más cercanos mutuos con $k = 5$.



(d) Grafo completamente conectado.

FIGURA 2: Diferentes configuraciones de grafos para un conjunto de puntos.

Para los *kernels* definidos positivos, que representan un producto punto en el espacio transformado $\mathcal{H}$, es posible obtener distancias entre pares de puntos:

$$\text{dis}(x, x') = \sqrt{k(x, x) - 2k(x, x') + k(x', x')} \tag{22}$$

5.3. Laplaciano de un grafo

Una de las herramientas principales del conglomeramiento espectral son los laplacianos de los grafos (un tratado más detallado puede encontrarse en Chung (1997)). En este trabajo se consideran tres tipos de laplacianos:

- **Laplaciano no normalizado.** La matriz del laplaciano no normalizado está definida como

$$L = D - W \tag{23}$$

Esta matriz se distingue por ser semidefinida positiva, lo cual implica que tiene valores propios reales no negativos; si V es completo, entonces $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ y el vector propio asociado a λ_1 es un vector constante de unos multiplicado por algún factor de escala. En el caso de que la matriz L se pueda ordenar en k bloques completos, el valor propio 0 tiene multiplicidad

k. Una propiedad muy importante que puede asociar este laplaciano con la partición de un grafo está dada por

Para todo vector $f \in \mathbb{R}^n$, se tiene que

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \tag{24}$$

- **Laplacianos normalizados.** En este caso se consideran dos tipos de normalizaciones:

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{25}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W \tag{26}$$

En este caso se satisface que

$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \tag{27}$$

Además, existe una relación muy estrecha entre ambos laplacianos: λ es un valor propio de L_{rw} con su respectivo vector propio u, si y solo si λ es un valor propio de L_{sym} con el correspondiente vector propio $w = D^{1/2}u$; 0 es un valor propio de L_{rw} con vector propio $\mathbf{1}$, entonces 0 es un valor propio de L_{sym} con vector propio $D^{1/2}\mathbf{1}$. Ambas matrices son semidefinidas positivas y, por tanto, tienen valores propios reales no negativos.

5.4. Algoritmos de conglomeramiento espectral

Los algoritmos presentados a continuación tienen como entrada el número k de conglomerados por construir y la matriz de similitudes $S \in \mathbb{R}^{n \times n}$.

Algoritmo 1: conglomeramiento espectral para laplaciano no normalizado y laplaciano normalizado L_{rw}.

1: Construir el grafo utilizando alguno de los métodos expuestos en 5.2 para obtener la matriz de adyacencia W.
2: Calcular el laplaciano no normalizado L dado por (23).
3: **if** (laplaciano no normalizado = true) **then**
4: Calcular los primeros k vectores propios $u_1, \ldots, u_k$ de L.
5: **else if** (laplaciano normalizado L_{rw}= true) **then**
6: Calcular los primeros k vectores propios $u_1, \ldots, u_k$ del problema de vectores propios generalizado $Lu = \lambda Du$.
7: **end if**
8: Construir la matriz $U \in \mathbb{R}^{n \times k}$ que contiene los k primeros vectores propios como sus columnas.
9: Hacer que y_i, para $i = 1, \ldots, n$, sea cada una de las filas de U.
10: Aplicar el algoritmo de k-medias a los puntos $(y_i)_{i=1,\ldots,n}$ en $\mathbb{R}^k$ y agruparlos en los conglomerados $C_1, \ldots, C_k$.

Algoritmo 2: conglomeramiento espectral para laplaciano normalizado L_{sym}.

1: Construir el grafo utilizando alguno de los métodos expuestos en 5.2 para obtener la matriz de adyacencia W.
2: Calcular el laplaciano normalizado L_{sym} dado por (25).
3: Calcular los primeros k vectores propios $u_1, \ldots, u_k$ de L_sym.
4: Construir la matriz $U \in \mathbb{R}^{n \times k}$ que contiene los k primeros vectores propios como sus columnas.
5: Normalizar la matriz U por filas para que cada una tenga norma 1, es decir, $\widehat{u}_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.
6: Hacer que y_i, para $i = 1, \ldots, n$, sea cada una de las filas de $\widehat{U}$.
7: Aplicar el algoritmo de k-medias a los puntos $(y_i)_{i=1,\ldots,n}$ en $\mathbb{R}^k$ y agruparlos en los conglomerados $C_1, \ldots, C_k$.

En los dos casos, los puntos $x_1, \ldots, x_n$ se ordenan de acuerdo con las agrupaciones hechas en $C_1, \ldots, C_k$.

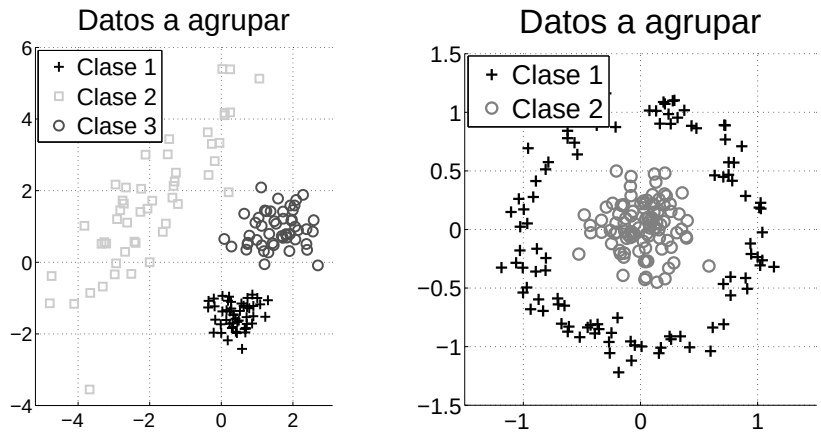
6. Selección de algoritmos de conglomeramiento espectral

Para evaluar los algoritmos de conglomeramiento espectral se trabajan dos conjuntos de datos artificiales en dos dimensiones, que se describen a continuación:

- **Conglomeramiento con diferente variabilidad intragrupo:** consiste en 3 distribuciones gaussianas con diferentes medias y matrices de covarianza. En el caso particular, cada uno de los grupos tiene 50 puntos (figura 3(a)). Este tipo de configuración es común en los datos reales donde el comportamiento de la perturbación intraclase no es homocedástico para todas las clases presentes. En este caso deben ser más apreciables los efectos de tener en cuenta la normalización del laplaciano del grafo construido.

- **Conglomerados no linealmente separables:** en este conjunto de datos, el primer conglomerado se deriva de una distribución gaussiana isotrópica; el segundo conglomerado es una corona circular que encierra al primer conglomerado. Cada conglomerado cuenta con 100 datos (figura 3(b)). El propósito de este conjunto es vislumbrar las propiedades de conectividad de los diferentes tipos de grafos.

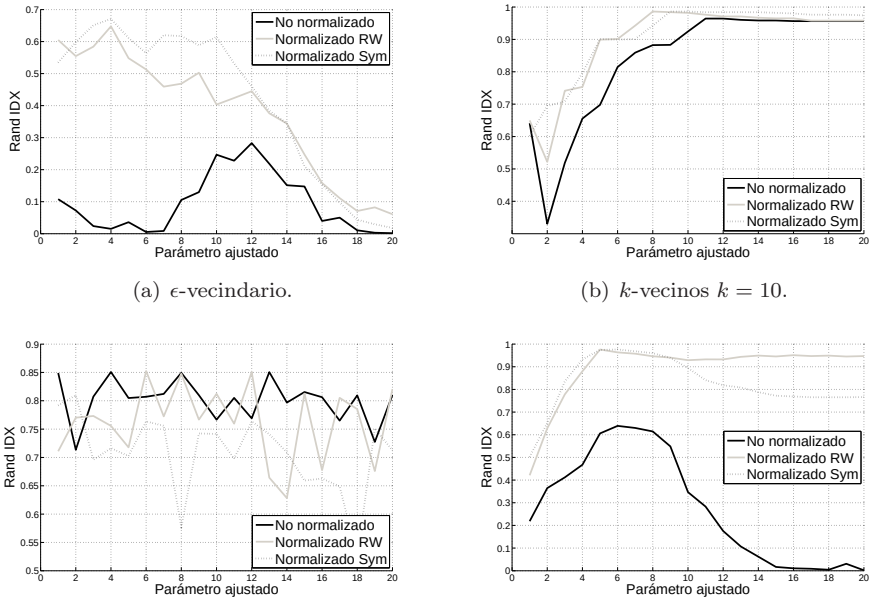
Como medida de similitud y forma de dar pesos a las aristas del grafo, se utiliza el kernel gaussiano, ya que cumple las restricciones de no negatividad. La comparación de los algoritmos de conglomeramiento es independiente de la sintonización del parámetro del kernel (o en el caso del grafo ϵ-vecindario del parámetro ϵ), es decir, suponiendo que existe algún criterio de sintonización, se comparan los índices de Rand ajustados (Yeung & Ruzzo 2000) para diferentes valores de σ o ϵ, según el caso. Las figuras 4 y 5 ilustran el comportamiento de los algoritmos de conglomeramiento espectral considerando las variaciones del parámetro de sintonización



(a) Diferentes varianzas intragrupo. (b) Linealmente no separables.

FIGURA 3: Datos artificiales para las pruebas de conglomerado espectral.

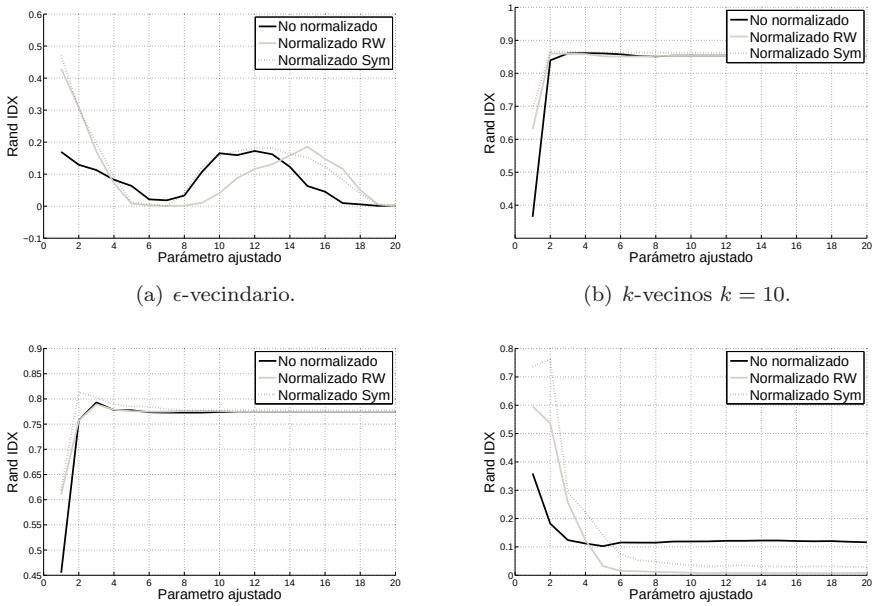
(kernel o vecindario, dependiendo del grafo). Las curvas presentadas se obtienen de promediar 10 iteraciones independientes de los algoritmos de conglomeramiento espectral, tal que cada muestra de prueba es independiente de las otras, pero se genera a partir de la misma distribución subyacente.



(a) ϵ-vecindario. (b) k-vecinos $k = 10$.

(c) k-vecinos mutuos $k = 10$. (d) Grafo completamente conectado.

FIGURA 4: Desempeño de los algoritmos de conglomeramiento espectral para el conjunto de datos artificiales con tres clases.



(a) ϵ-vecindario.

(b) k-vecinos $k = 10$.

(c) k-vecinos mutuos $k = 10$.

(d) Grafo completamente conectado.

FIGURA 5: Desempeño de los algoritmos de conglomeramiento espectral para el conjunto de datos no linealmente separables.

Aunque la selección de un modelo de conglomeramiento espectral particular sigue siendo un problema abierto, los resultados obtenidos con las muestras artificiales sugieren utilizar:

- El grafo de k-vecinos, porque los índices de Rand obtenidos parecen insensibles a la variación del parámetro del kernel. Esto podría entenderse como la propiedad de los k-vecinos para establecer un vecindario cuyo radio es adaptativo, dependiendo del punto que se desee conectar al grafo.
- El grafo completamente conectado en conjunto con los laplacianos normalizados puede obtener muy buenos resultados si se logra una sintonización apropiada del parámetro por ajustar. Los cambios que experimenta el índice de Rand al variar, en este caso, el σ del kernel parecen más suaves; por tanto, algún criterio que logre establecer un valor cercano al de mejor desempeño podría funcionar relativamente bien.

7. Alineamiento del kernel

Un criterio para sintonización de un kernel utilizado en el conglomeramiento espectral, que en un principio fue introducido dentro del contexto de aprendizaje supervisado, se presenta en Cristianini et al. (2001). La idea básica del aprendizaje consiste en establecer una relación directa entre la entrada que se tiene y la salida

que se busca. Se espera encontrar observaciones asociadas como más cercanas entre sí, es decir, que tomen valores similares en sus medidas. Una forma de medir lo anterior es mediante la correlación existente entre lo que se mide y lo que se desea. En el caso particular, se desea medir la correlación entre los valores arrojados por la utilización de un kernel y utilizarlos como predictores de agrupaciones. En este trabajo el alineamiento consiste en conseguir un alto grado de correlación entre la matriz K, obtenida a través de la aplicación de un kernel, y las particiones obtenidas por los algoritmos de conglomeramiento presentados en la sección 5.

Definición 1. Alineamiento: el alineamiento empírico de un kernel k_1 con un kernel k_2, con respecto a una muestra S, está dado por la cantidad

$$\widehat{A}(S, k_1, k_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \tag{28}$$

donde K_i es la matriz kernel para la muestra S utilizando el kernel k_i; $\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{n} k_1(x_i, x_j) k_2(x_i, x_j)$ (esta definición de producto interno induce la norma de Frobenius).

Es de notar que las matrices K deben satisfacer $\sum_{i,j=1}^{n} k_1(x_i, x_j) = 0$; así se puede calcular la correlación entre k_1 y k_2. Si se considera k_1 como el kernel por ajustar y K_2 una matriz que representa la partición hecha por el algoritmo de conglomeramiento construida de la siguiente forma,

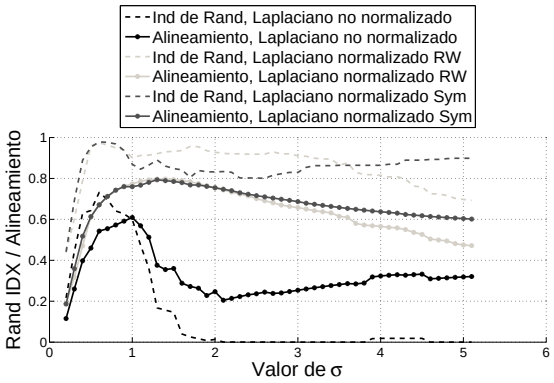
$$k_2(x_i, x_j) = \begin{cases} 1, & \text{si } x_i \text{ y } x_j \text{ están en } C_a \forall a; \\ 0, & \text{si } x_i \text{ y } x_j \text{ no están en } C_a \forall a. \end{cases} \tag{29}$$

En esta implementación del alineamiento de conglomerado no se pretende ajustar un umbral para una partición binaria, como se propone en Cristianini et al. (2002). La idea fundamental para el caso aquí propuesto consiste en ajustar los parámetros del kernel que mejor se ajusten a la partición obtenida por un algoritmo convencional como el k-medias, buscando el punto de mayor concordancia entre los valores que inducen la partición y la partición en sí.

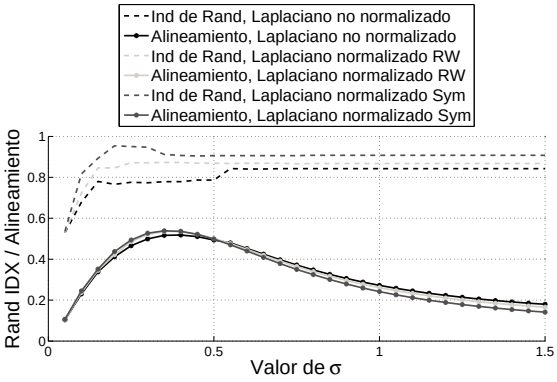
La figura 6 presenta los valores de alineamiento obtenidos al variar el parámetro σ del kernel gaussiano para las mismas 10 iteraciones del algoritmo de conglomeramiento espectral utilizando el grafo completamente conectado para los datos artificiales correspondientes a 3 clases, y el grafo de los k-vecinos más cercanos para los datos linealmente no separables. Note que los picos del alineamiento están localizados cerca de los puntos donde se obtienen los máximos índices de Rand sobre el rango de variación del parámetro a sintonizar.

8. Bases de datos reales

En esta parte del trabajo se pretende indagar, de manera empírica, la efectividad de aplicar ACP como etapa de preproceso a los algoritmos de conglomeramiento espectral. Para este fin se utilizan dos bases de datos reales disponibles en línea para la comparación de resultados (*benchmarks data sets*). A continuación se presenta una breve descripción de cada una de ellas.



(a) Resultados para datos artificiales provenientes de 3 clases utilizando el grafo completamente conectado.



(b) Resultados para datos artificiales no linealmente separables utilizando el grafo de los k-vecinos $k = 10$.

FIGURA 6: Alineamiento para sintonización del σ del kernel gaussiano en los datos artificiales.

8.1. Lirios de Fisher

Esta base de datos fue introducida por Sir R. A. Fisher en 1936 para estudiar los métodos de análisis discriminante. Los datos corresponden a una muestra de 150 observaciones con 3 clases que corresponden a *Lirio setosa*, *Lirio virginica* y *Lirio versicolor* (clases balanceadas). Cuatro variables (longitud del pétalo, ancho del pétalo, largo del sépalo, ancho del sépalo) describen cada observación.

8.2. Dígitos manuscritos (MNIST *Database*)

Está formada por imágenes de números manuscritos segmentadas y escaladas a igual tamaño (LeCun & Cortés 2008). Originalmente, la base de datos se encuentra conformada por dos muestras independientes: la muestra de entrenamiento, con

60000 observaciones repartidas en 10 clases que corresponden a los dígitos del 0 al 9 (figura 7); la muestra de validación, que contiene 10000 observaciones de las 10 clases. Como espacio de características se cuenta con las intensidades de grises de cada uno de las píxeles de las imágenes previamente filtradas con una máscara gaussiana de tamaño 9×9. Dado que las imágenes son de 28×28, píxeles el espacio de entrada consta de 784 dimensiones. Para efectos de facilitar la visualización de resultados, este trabajo considera 3 de las 10 clases para hacer el conglomerado. Las clases de interés son los dígitos 0, 1 y 4.

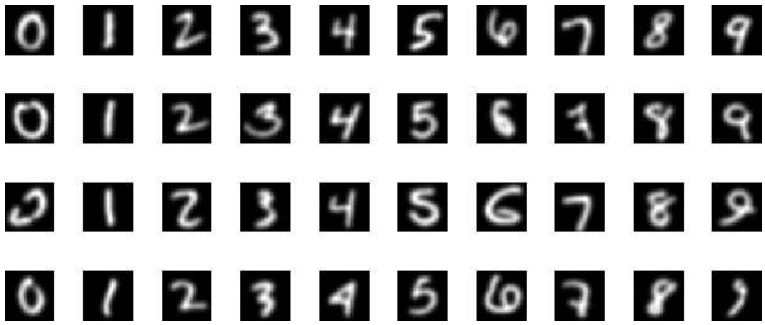


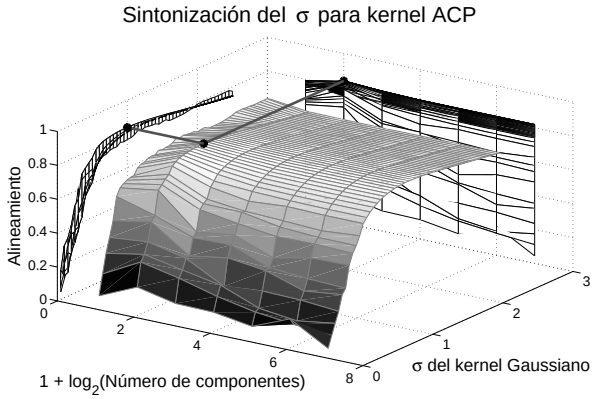
Figura 7: Algunos ejemplos de dígitos manuscritos (*MNIST Database*).

9. Resultados y discusión

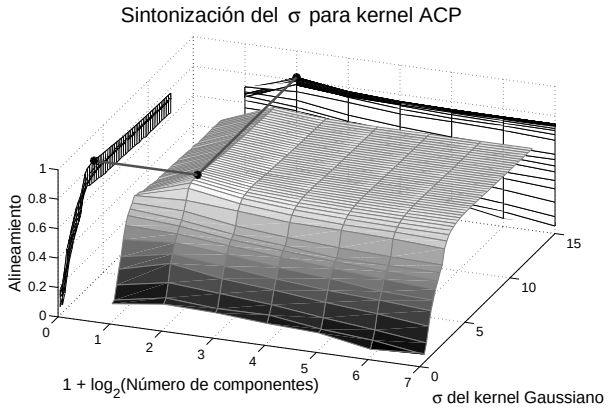
Los resultados por comparar corresponden a los índices de Rand obtenidos por los algoritmos de conglomeramiento sin aplicar, y luego de aplicar dos tipos de preproceso (el primero basado en ACP lineal y el segundo basado en ACP no lineal utilizando el kernel gaussiano). Los algoritmos de conglomeramiento empleados en la comparación son k-medias con distancia $\|\cdot\|_2$ y conglomerado espectral con grafo de los k-vecinos para $k = 10$, y el grafo completamente conectado, utilizando los dos laplacianos normalizados. La versión kernel del algoritmo de k-medias, que también se incluye en el análisis, es equivalente a evaluar este algoritmo utilizando todas las componentes principales no nulas extraídas por kernel ACP. Para considerar las variaciones de inicialización en el algoritmo de k-medias, este se corre con 10 inicializaciones aleatorias para cada conjunto de datos, escogiendo la partición con menor valor para SC. En el caso de los lirios, las pruebas se realizan una sola vez sobre todas las observaciones de la base de datos, mientras que los dígitos manuscritos requieren un tratamiento diferente debido a la cantidad de observaciones disponibles. En particular, los resultados de las pruebas sobre MNIST se obtienen al promediar los resultados de 30 corridas, escogiendo en cada una de ellas y de manera aleatoria 50 observaciones por clase.

Como se observa en la figura 8, los máximos valores de alineamiento para el KACP se obtienen con 2 componentes principales para $\sigma = 1$ y $\sigma = 4$ en los lirios y los dígitos, respectivamente. Para efectos de simplificar el ajuste de

parámetros, estos valores obtenidos de sigma se utilizan en el preproceso con KACP indiferentemente del algoritmo de conglomeramiento que se aplique. Los algoritmos de conglomerados espectrales requieren la sintonización adicional del kernel que define la similitud en los grafos.



(a) Lirios.



(b) Dígitos 0, 1 y 4.

FIGURA 8: Sintonización del σ para el kernel gaussiano en KACP utilizando el alineamiento con k-medias.

En la tabla 1 se observa un comportamiento muy particular en los índices de Rand obtenidos para los diferentes espacios de entrada considerando 1, 2, 3 y 4 componentes. Los mejores resultados de conglomeramiento, incluso comparados contra los obtenidos al evaluar el espacio completo, se logran al considerar la primera componente principal. Al hacer una inspección más detallada de cómo se encuentran distribuidas las clases en el espacio de características, se observa que las covarianzas de las clases problema (*versicolor* y *virginica*) son similares y el vector que une sus medias no se encuentra alineado con ninguna de las direcciones

Tabla 1: Resultados de los diferentes algoritmos de conglomeramiento para los datos de los lirios sin preprocesar el espacio inicial y preprocesados con ACP lineal. Las componentes son escogidas en orden descendente de sus valores propios asociados.

Espacio de entrada	Algoritmo de conglomeramiento				
	k-medias	Grafo completamente conectado		Grafo de los k-vecinos, para $k = 10$	
		L_{rw}	L_{sym}	L_{rw}	L_{sym}
Sin transformar	0.7302	0.7562	0.7163	0.7445	0.7445
1 CP lineal	0.7726	0.8022	0.7726	0.7720	0.8340
2 CP lineales	0.7163	0.7424	0.7163	0.7302	0.7302
3 CP lineales	0.7302	0.7711	0.7163	0.7445	0.7445
4 CP lineales	0.7302	0.7866	0.7163	0.7445	0.7445

Tabla 2: Resultados de los diferentes algoritmos de conglomeramiento para los datos de los lirios preprocesados con kernel ACP a través del kernel gaussiano siendo el parámetro ajustado $\sigma = 4$. Las componentes son escogidas en orden descendente de sus valores propios asociados. El total de componentes no nulas es 58.

Espacio de entrada	Algoritmo de conglomeramiento				
	k-medias	Grafo completamente conectado		Grafo de los k-vecinos, para $k = 10$	
		L_{rw}	L_{sym}	L_{rw}	L_{sym}
1 CP no lineal	0.5128	0.5388	0.5132	0.5027	0.5010
2 CP no lineales	0.8015	0.8015	0.8015	0.7312	0.7302
4 CP no lineales	0.7437	0.7287	0.8015	0.7312	0.7302
8 CP no lineales	0.7437	0.8015	0.8015	0.7455	0.7455
16 CP no lineales	0.7437	0.8176	0.8015	0.7592	0.7445
32 CP no lineales	0.7437	0.8176	0.8015	0.7445	0.7445
64 CP no lineales	0.7437	0.8176	0.8015	0.7445	0.7445

principales de sus matrices de covarianza, lo cual explica la degradación en los resultados del conglomerado al agregar la segunda componente principal. Considerando el mapeo no lineal del espacio de características a través de KACP, se construyó la tabla 2. En general, para el algoritmo de k-medias, se logra el mejor rendimiento con 2 componentes principales, como se había predicho en el gráfico de alineamiento del kernel (figura 8(a)). También se observa que los algoritmos de conglomeramiento espectral alcanzan los mayores valores en los índices de Rand considerando conjuntos de CP no lineales más grandes. Esta situación puede deberse a las particiones no lineales obtenidas con el conglomeramiento espectral[2]. Básicamente los resultados obtenidos con ACP lineal y kernel ACP no difieren mucho para la muestra de lirios, dada la naturaleza de las clases, el introducir mapeos no lineales en el problema no aporta poder discriminante a los procedimientos para la construcción de los conglomerados.

[2]Desde el punto de vista del espacio de entrada al algoritmo de conglomeramiento. Por ejemplo, el algoritmo de k-medias combinado con KACP puede lograr particiones no lineales del espacio inicial que fue transformado con KACP; sin embargo, si el espacio inicial es introducido directamente al algoritmo de conglomeramiento, este no puede conseguir particiones no lineales.

Tabla 3: Resultados de los diferentes algoritmos de conglomeramiento para los dígitos manuscritos del 0, 1, y 4 de la base de datos MNIST sin preprocesar el espacio inicial y preprocesados con ACP lineal. Las componentes son escogidas en orden descendente de sus valores propios asociados.

	Algoritmo de conglomeramiento				
Espacio de entrada	k-medias	Grafo completamente conectado		Grafo de los k-vecinos, para $k = 10$	
		L_{rw}	L_{sym}	L_{rw}	L_{sym}
1 CP lineal	0.5299 ±0.1021	0.5692 ±0.1119	0.5037 ±0.0940	0.6329 ±0.0591	0.6590 ±0.0557
2 CP lineal	0.8105 ±0.0778	0.8333 0.0687	0.8228 ±0.0802	0.8939 ±0.0642	0.9004 ±0.0619
4 CP lineal	0.8236 ±0.0693	0.8828 0.0534	0.8511 ±0.0566	0.9102 ±0.0908	0.9265 ±0.0465
8 CP lineal	0.8412 ±0.0711	0.8921 0.0669	0.8630 ±0.0658	0.91256 ±0.0960	0.9410 ±0.0384
16 CP lineal	0.8448 ±0.0722	0.8511 0.1134	0.8674 ±0.0613	0.9151 ±0.1026	0.9436 ±0.0391
32 CP lineal	0.8538 ±0.0594	0.7823 0.1647	0.8696 ±0.0576	0.9060 ±0.1057	0.9423 ±0.0399
64 CP lineal	0.8538 ±0.0594	0.7498 0.1614	0.8671 ±0.0620	0.9097 ±0.1084	0.9405 ±0.0409
128 CP lineal	0.8538 ±0.0594	0.7465 0.1624	0.8671 ±0.0620	0.9122 ±0.1027	0.9405 ±0.0409

Las pruebas hechas sobre la base de datos de dígitos manuscritos presentan un comportamiento diferente al expuesto por los lirios. Entre los factores a los que se les puede atribuir este comportamiento se encuentran la dimensión y la distribución de los conglomerados en el espacio. Al calcular la distancia euclídea entre las medias de cada una de las clases, se encuentra que estas no están alineadas como en el caso de la base de datos de lirios, ya que las distancias son similares. Los resultados obtenidos al aplicar ACP lineal (tabla 3) en general son asintóticos, porque agregan componentes; sin embargo, los efectos de supresión de ruido lucen similares a los obtenidos en Schölkopf et al. (1999), donde ACP tiene un efecto benéfico de filtrado hasta cierto número de componentes considerado; cuando se agregan más componentes, el ruido aparece de nuevo. Aunque en el problema tratado en el presente trabajo no se contaminan las imágenes con perturbaciones comunes en imágenes como ruido sal y pimienta (perturbaciones de máximos o mínimos en el rango dinámico de la imagen) o el ruido gaussiano, puede considerarse que el modelo de una clase es distorsionado por los detalles que agrega cada uno de los participantes en la muestra. La sintonización del σ presentada en la figura 8(b) da buenos resultados con el algoritmo de k-medias, y aunque al agregar más componentes aumenta el índice de Rand, el incremento no es tan notorio como en los algoritmos de conglomeramiento espectral. Los resultados obtenidos sí son superiores a los reportados con ACP lineal. En las pruebas realizadas (tablas 3 y 4), KACP no ofrece un mejoramiento conjunto en los algoritmos de conglomeramiento espectral, puesto que los resultados siguen siendo altamente dependientes del laplaciano y el grafo empleados. Es posible que dicho resultado también se deba

Tabla 4: Resultados de los diferentes algoritmos de conglomeramiento para los dígitos manuscritos del 0, 1, y 4 de la base de datos MNIST preprocesados con kernel ACP a través del kernel gaussiano siendo el parámetro ajustado $\sigma = 4$. Las componentes son escogidas en orden descendente de sus valores propios asociados.

	Algoritmo de conglomeramiento				
Espacio de entrada	k-medias	Grafo completamente conectado		Grafo de los k-vecinos, para $k = 10$	
		L_{rw}	L_{sym}	L_{rw}	L_{sym}
1 CP no lineal	0.6458 ±0.0618	0.6582 ±0.0653	0.6539 ±0.0732	0.6060 ±0.0965	0.6414 ±0.0930
2 CP no lineal	0.8811 ±0.0349	0.8749 0.0405	0.8817 ±0.0376	0.89536 ±0.0540	0.9014 ±0.0382
4 CP no lineal	0.8850 ±0.0405	0.8881 0.0380	0.8923 ±0.0372	0.8721 ±0.1579	0.8683 ±0.1529
8 CP no lineal	0.8936 ±0.0396	0.8751 0.0583	0.9080 ±0.0349	0.9231 ±0.0799	0.9357 ±0.0333
16 CP no lineal	0.9013 ±0.0426	0.8060 0.1676	0.9111 ±0.0359	0.9434 ±0.0340	0.9454 ±0.0339
32 CP no lineal	0.8994 ±0.0421	0.4243 0.0497	0.9111 ±0.0360	0.9426 ±0.0331	0.9472 ±0.0315
64 CP no lineal	0.9007 ±0.0422	0.7223 0.2032	0.9118 ±0.0375	0.9426 ±0.0350	0.9439 ±0.0345
128 CP no lineal	0.8994 ±0.0418	0.5296 0.1058	0.9118 ±0.0375	0.9354 ±0.0353	0.9380 ±0.0354

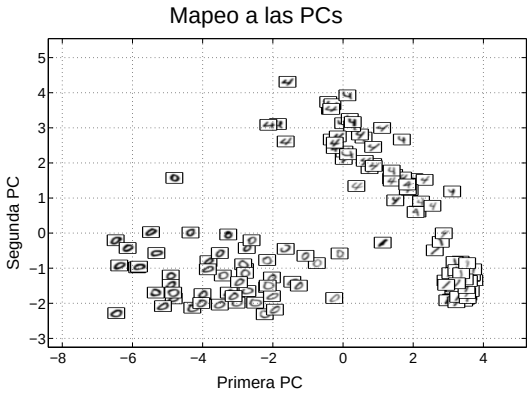
a la relación existente entre los métodos de conglomeramiento espectral y KACP, reportada en diferentes trabajos (Bengio et al. 2003, Bengio et al. 2004, Alzate & Suykens 2006), y al criterio con que fue sintonizado el parámetro de KACP (alineamiento sobre conglomerados por k-medias).

Como se había mencionado, uno de los procedimientos referentes a la búsqueda de estructura en los datos está asociado a la posibilidad de visualizar los puntos que confirman la muestra, de modo que dejen ver alguna estructura de agrupamiento. ACP y/o KACP son formas básicas de conseguirlo (figura 9).

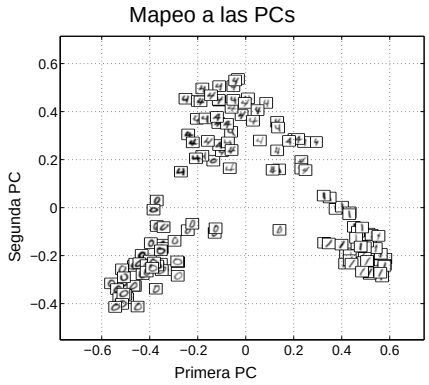
10. Conclusiones

Se presentaron las técnicas de KACP y conglomeramiento espectral, y cómo el uso de los *kernels* posibilita mapeos y particiones no lineales en ambos procedimientos. En los experimentos realizados con kernel ACP se encuentran ventajas principalmente al utilizar el algoritmo de k-medias ya que la generación de las particiones lineales sobre los mapeos no lineales introducidos por KACP genera algoritmos de conglomeramiento con particiones no lineales en el espacio de entrada.

El alineamiento del kernel proporciona un buen criterio para sintonización de parámetros, como se observó al contrastar las curvas de alineamiento con los índices de Rand obtenidos.



(a) Mapeo con ACP lineal.



(b) Mapeo con KACP, kernel gaussiano $\sigma = 4$.

FIGURA 9: Mapeos de las imágenes de 28×28 píxeles a las primeras 2 componentes principales lineales y no lineales.

Para los experimentos realizados, la combinación KACP-conglomeramiento espectral no presenta efectos muy claros de mejor desempeño comparados con ACP lineal. Los resultados muestran ser más dependientes del tipo de grafo y laplaciano empleados. Sin embargo, se pudo observar el efecto de eliminación de filtrado que tiene el aplicar ACP y KACP, sobre todo en las imágenes de dígitos manuscritos, donde un número moderado de componentes mejora los resultados notoriamente y el continuar agregando más componentes degrada el resultado.

Como se ha mostrado en otros trabajos que tratan aproximaciones similares, aplicar exitosamente ACP como una etapa de preproceso es altamente dependiente del problema, y poder catalogar todos los escenarios posibles donde ACP puede o no funcionar, no se ha resuelto todavía.

[Recibido: octubre de 2007 — Aceptado: febrero de 2008]

Referencias

Alzate, C. & Suykens, J. A. K. (2006), A Weighted Kernel PCA Formulation with Out-of-Sample Extensions for Spectral Clustering Methods, *in* 'International Joint Conference on Neural Networks', pp. 138–144.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J. F., Vincent, P. & Ouimet, M. (2004), 'Learning eigenfunctions links spectral embedding and kernel PCA', *Neural Computation* **16**(10), 2197–2219.

Bengio, Y., Vincent, P., Paiement, J. F., Delalleau, O., Ouimet, M. & Le Roux, N. (2003), Spectral Clustering and Kernel PCA are Learning Eigenfunctions, Technical report, Département d'Informatique et Recherche Opérationnelle Centre de Recherches Mathématiques, Université de Montréal.

Chung, F. R. K. (1997), *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*, American Mathematical Society.

Cristianini, N., Elisseeff, A., Shawe-Taylor, J. & Kandola, J. (2001), On kernel-target alignment, *in* 'Proceedings Neural Information Processing Systems'.

Cristianini, N., Shawe-Taylor, J. & Kandola, J. (2002), Spectral kernel methods for clustering, *in* 'Proceedings Neural Information Processing Systems'.

Jolliffe, I. T. (2002), *Principal Component Analysis*, Springer Series in Statistics, second edn, Springer.

LeCun, Y. & Cortés, C. (2008), 'The MNIST Database'. Tomado en octubre de 2007 de la página web.
*http://yann.lecun.com/exdb/mnist/

Luxburg, U. V. (2006), A Tutorial on Spectral Clustering, Technical report, Max-Planck-Institut für biologische Kybernetik.

Peña, D. (2002), *Análisis de datos multivariantes*, McGraw-Hill.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K. R., Ratsch, G. & Smola, A. J. (1999), 'Input Space vs. Feature Space in Kernel-Based Methods', *IEEE Transactions on Neural Networks* **10**(5), 1000–1017.

Schölkopf, B. & Smola, A. (2002), *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, United States.

Schölkopf, B., Smola, A. & Müller, K. R. (1996), Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Technical Report 44, Max-Planck-Institut für biologische Kybernetik.

Yeung, K. Y. & Ruzzo, W. L. (2000), An empirical study on Principal Component Analysis for Clustering Gene Expression Data, Technical report, Department of Computer Science and Engineering, University of Washington.