Kernel PCA:
keep walking ... in informative directions

Johan Van Horebeek, Victor Muñiz, Rogelio Ramos
CIMAT, Guanajuato, GTO

Kernel PCA:
keep walking ... in informative directions

Johan Van Horebeek, Victor Muñiz, Rogelio Ramos
CIMAT, Guanajuato, GTO

Contents:

1. Kernel based methods

- as a computational trick
- to define (nonlinear) extensions
- for data with no natural vector representation

2. Some issues of interest

- robustness
- detecting influential variables
- KPCA and random projections

1.Kernel based methods

1.1. Principal Component Analysis (PCA)

Given $X = (X_1, \cdots, X_d)^t$,

we look for a direction u such that the projection $\langle u, X \rangle$ is informative.

In PCA, informative means maximun variance : $\arg\max_u Var(\langle u, X \rangle)$.

Solution: u is the first eigenvector of $Cov(X)$.

1.Kernel based methods

1.1. Principal Component Analysis (PCA)

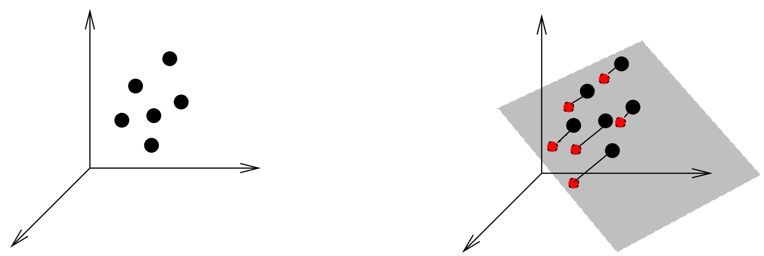
Given $X = (X_1, \cdots, X_d)^t$,

we look for a direction u such that the projection $\langle u, X \rangle$ is informative.

In PCA, informative means maximun variance : $\arg\max_u Var(\langle u, X \rangle)$.

Solution: u is the first eigenvector of $Cov(X)$.

Repeating this k times and imposing decorrelation with previous found projections, we obtain a k-dimensional space spanned by the first k eigenvectors of $Cov(X)$.



Many nice properties (esp. if multinormal distributed);

e.g. characterization as best linear k-dim. predictor.

1.Kernel based methods

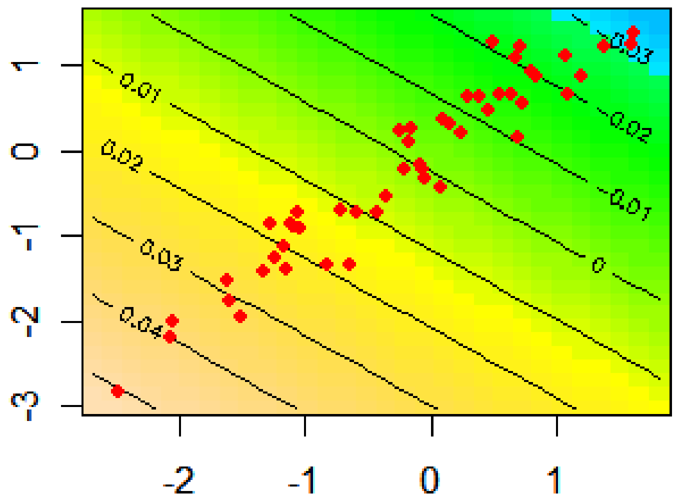
1.1. Principal Component Analysis (PCA)

Given $X = (X_1, \cdots, X_d)^t$,

we look for a direction u such that the projection $\langle u, X \rangle$ is informative.

In PCA, informative means maximun variance : $\arg\max_u Var(\langle u, X \rangle)$.

Solution: u is the first eigenvector of $Cov(X)$.



Define projection function: $f(x) := \langle u, x \rangle$ with u solution of PCA.

We show the contour lines of f;

the gradient(s) mark the direction of the most informative walk;

an order is also obtained.

1.Kernel based methods

Example

Suppose objects are texts:

	word$_1$	**word**$_2$	. . . . . .	**word**$_d$
doc$_1$	.	.	. . . . . .	.
.	.	.	. . . . . .	.
doc$_n$	.	.	. . . . . .	.

1.Kernel based methods

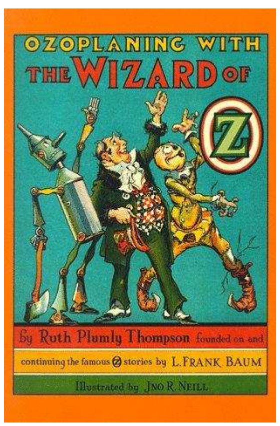
Example

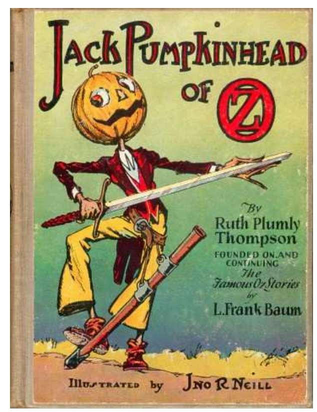
Suppose objects are texts:

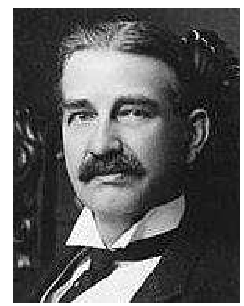
	word$_1$	word$_2$	. . . . . .	word$_d$
doc$_1$	.	.	. . . . . .	.
.	.	.	. . . . . .	.
doc$_n$	.	.	. . . . . .	.

Stylometry

Books of the Wizard of Oz (X): some written by Thompson, others by Baum.







Define the 50 most used words.

Define $(X_1, \cdots, X_{50})$ with X_i the (relative) frecuency of occurrence of word i in a *chapter*.

1.Kernel based methods

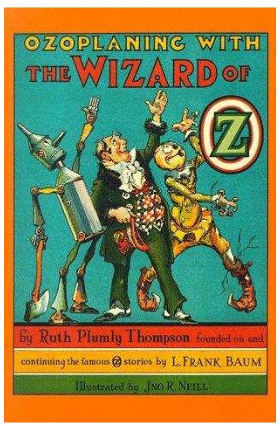
Example

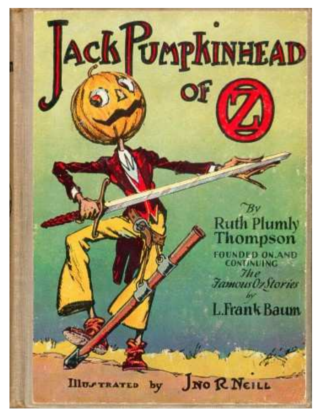
Suppose objects are texts:

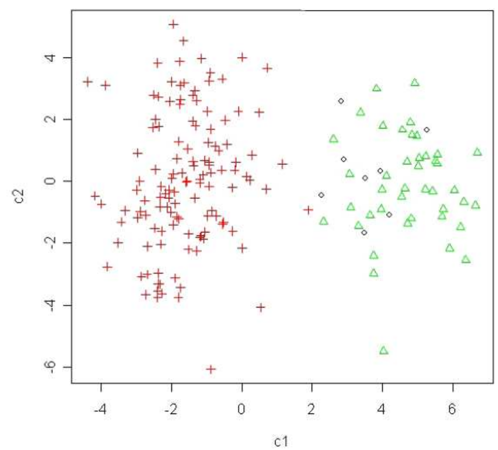
	word_1	word_2	. . . . . .	word_d
doc_1	.	.	. . . . . .	.
.	.	.	. . . . . .	.
doc_n	.	.	. . . . . .	.

Stylometry

Books of the Wizard of Oz (X): some written by Thompson, others by Baum.







Define the 50 most used words.

Define $(X_1, \cdots, X_{50})$ with X_i the (relative) frecuency of occurrence of word i in a *chapter*.

1.Kernel based methods

1.1. Principal Component Analysis (PCA)

Solution: u is the first eigenvector of $Cov(X)$.

Suppose we estimate $Cov(X)$ by the sample covariance

$$\widehat{Cov(X)} \sim \mathbf{X}^t\mathbf{X} \quad \textbf{with } \mathbf{X} \textbf{ the centered data matrix},$$

1.Kernel based methods

1.1. Principal Component Analysis (PCA)

Solution: u is the first eigenvector of $Cov(X)$.

Suppose we estimate $Cov(X)$ by the sample covariance

$$\widehat{Cov(X)} \sim \mathbf{X}^t\mathbf{X} \quad \textbf{with X the centered data matrix},$$

Property

If $\{u_j\}$ are eigenvectors of $\mathbf{X}^t\mathbf{X}$; and $\{v_j\}$ eigenvectors of $\mathbf{X}\mathbf{X}^t$, then

$$u_j \sim \mathbf{X}^t v_j := \sum_i \alpha_i^j x_i.$$

Hence, if $n < d$, it is convenient to calculate eigenvectors of $\mathbf{X}\mathbf{X}^t = [\langle x_i, x_j \rangle]_{i,j}$:

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j \langle x_i, x \rangle, \quad \alpha \textbf{ depends on eigenvectors of } \mathbf{X}\mathbf{X}^t$$

this leads to the *Kernel trick*

$\boxed{\text{1.Kernel based methods}}$

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j \langle x_i, x \rangle, \;\; \alpha \textbf{ depends on eigenvectors of } \mathbf{XX}^t = [\langle x_i, x_j \rangle]_{i,j} :$$

1. **If $n < d$ we have a computational convenient way (trick) to get $f(x)$.**

2. **Only internal products of the observations are necessary.**

   This can be interesting for complex objects (see later).

This forms the basis of Kernel PCA.

In the same way: Kernel LDA, Kernel Ridge, etc.: how to *kernelize* known methods?

1.Kernel based methods

$$f(x) = \langle u_j, x\rangle = \sum_i \alpha_i^j \langle x_i, x\rangle, \quad \alpha \text{ depends on eigenvectors of } \mathbf{XX}^t = [\langle x_i, x_j\rangle]_{i,j}:$$

1. If $n < d$ we have a computational convenient way (trick) to get $f(x)$.
2. Only internal products of the observations are necessary.

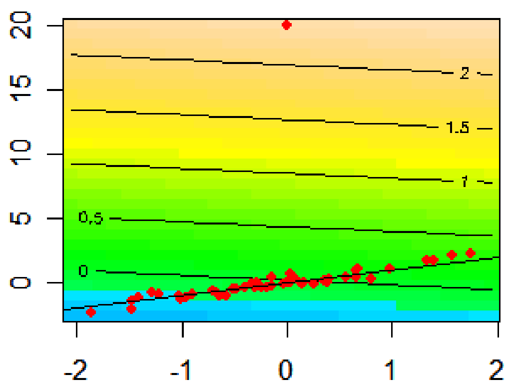
   This can be interesting for complex objects (see later).

This forms the basis of Kernel PCA.

In the same way: Kernel LDA, Kernel Ridge, etc.: how to *kernelize* known methods?

Many questions of interest; e.g.:

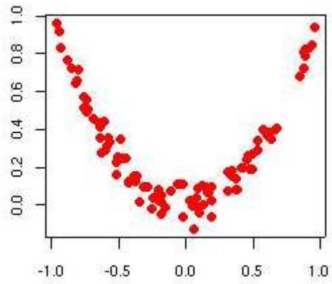
1. What if the sample covariance matrix is a bad estimator?

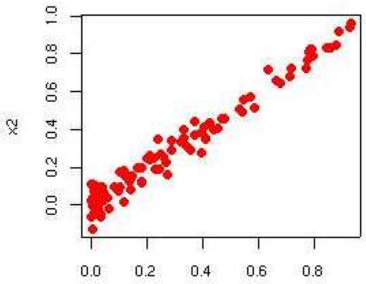


2. How to obtain insight about which variables are influential?

1.Kernel based methods

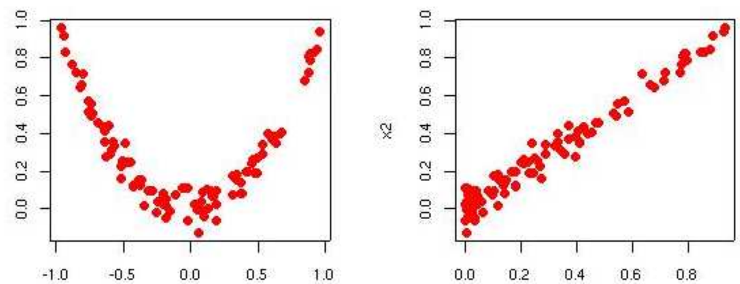
1.2. (Nonlinear) Extensions of Principal Component Analysis (PCA)





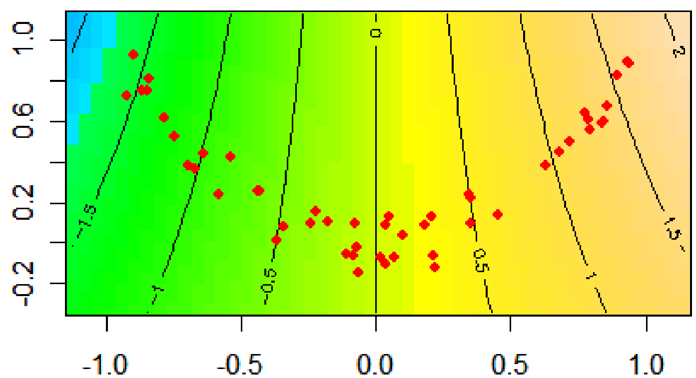
1.Kernel based methods

1.2. (Nonlinear) Extensions of Principal Component Analysis (PCA)



Solution: transform $X = (X_1, X_2)$ into $\Phi(X) = (X_1^2, X_2)$; apply PCA on $\{\Phi(x_i)\}$.

Projection function $f(x)$ in original space looks like:



Contour lines defined by: $\langle u, \Phi(x)\rangle =$ *constant*.

How to define $\Phi()$?

1.Kernel based methods

1.2. (Nonlinear) Extensions of Principal Component Analysis (PCA)

For some transformations it is computationaly convenient to work with kernels.

Before:

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j \langle x_i, x \rangle, \;\; \alpha \textbf{ depends on eigenvectors of } \mathbf{XX}^t = [\langle x_i, x_j \rangle]_{i,j} :$$

Suppose we transform x **into** $\Phi(x)$ **and define** $K_\Phi(x, y) :=< \Phi(x), \Phi(y) >$**:**

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j K_\Phi(x_i, x), \;\; \alpha \textbf{ depends on eigenvectors of } [K_\Phi(x_i, x_j)]_{i,j}$$

1.Kernel based methods

1.2. (Nonlinear) Extensions of Principal Component Analysis (PCA)

For some transformations it is computationaly convenient to work with kernels.

Before:

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j \langle x_i, x \rangle, \quad \alpha \textbf{ depends on eigenvectors of } \mathbf{X}\mathbf{X}^t = [\langle x_i, x_j \rangle]_{i,j} :$$

Suppose we transform x **into** $\Phi(x)$ **and define** $K_\Phi(x,y) := < \Phi(x), \Phi(y) >$**:**

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j K_\Phi(x_i, x), \quad \alpha \textbf{ depends on eigenvectors of } [K_\Phi(x_i, x_j)]_{i,j}$$

Example:

If

$$\Phi(z = (z_1, z_2)) = (1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1 z_2, z_2^2)$$

$$K_\Phi(x,y) = (1+ < x, y >)^2 \qquad \textbf{more general: } K_\Phi(x,y) = (1+ < x, y >)^k$$

This is easier to calculate then $\Phi(x)$**,** $\Phi(y)$ **and afterwards** $< \Phi(x), \Phi(y) >$ **!**

Observe: only $\Phi(x)$ **should belong to a vector space, not necessary** x**.**

Useful for objects with no natural vector representation.

1.Kernel based methods

1.2. (Nonlinear) Extensions of Principal Component Analysis (PCA)

For some transformations it is computationally convenient to work with kernels.

Before:

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j \langle x_i, x \rangle, \;\; \alpha \textbf{ depends on eigenvectors of } \mathbf{XX}^t = [\langle x_i, x_j \rangle]_{i,j} :$$

Suppose we transform x **into** $\Phi(x)$ **and define** $K_\Phi(x, y) :=< \Phi(x), \Phi(y) >$**:**

$$f(x) = \langle u_j, x \rangle = \sum_i \alpha_i^j K_\Phi(x_i, x), \;\; \alpha \textbf{ depends on eigenvectors of } [K_\Phi(x_i, x_j)]_{i,j}$$

Example:

Suppose x **and** y **are strings of length** d **over the alphabet** $\mathcal{A}$**, i.e.** $x, y \in \mathcal{A}^d$

Define $= (\Phi_s(x))_{s \in \mathcal{A}^d}$ **with** $\Phi_s(x)$ **the number of occurrences of substring** s **in** x**.**

Much easier to calculate $\langle \Phi(x), \Phi(y) \rangle$ **directly:**

$$\langle \Phi(x), \Phi(y) \rangle = \sum_{s \in S(x,y)} \Phi_s(x) \Phi_s(y) \textbf{ with } S(x, y) \textbf{ substrings of } x \textbf{ and } y.$$

1.Kernel based methods

How to choose $K(\cdot,\cdot)$?

1. **For which $K(\cdot,\cdot)$ exists a $\Phi()$ such that $K_\Phi(y, x_i) :=< \Phi(y), \Phi(x_i) >$?**

2. **How to understand it in data space? (and how to tune the parameters?)**

Problem

We do not have a good intuition to think in terms of inner products.

Much easier to think in terms of distances.

E.g. $K(x, y) = P(x)P(y)$ **leads to** $dist_\Phi(x, y) = (P(x) - P(y))^2$

1.Kernel based methods

1.3. The very particular case of kernel PCA with a Radial Base Kernel

Define

$$K(x, y) = \exp(-||x - y||^2/\sigma).$$

What can we say about $\Phi()$?

1.Kernel based methods

1.3. The very particular case of kernel PCA with a Radial Base Kernel

Define

$$K(x, y) = \exp(-||x - y||^2/\sigma).$$

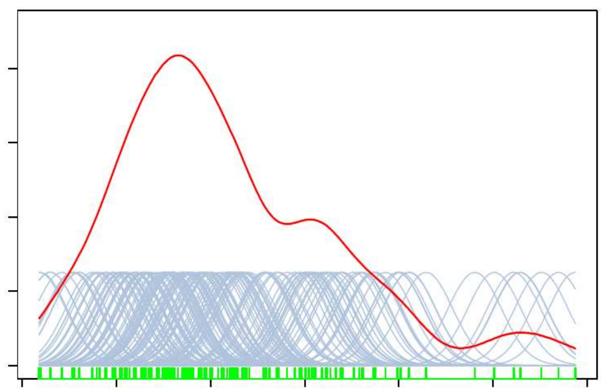
What can we say about $\Phi()$?

$||\Phi(x)||^2 = K(x, x) = 1$

i.e, we map x on a hypersphere ... of infinite dimension, $\Phi(x) \in \mathcal{R}^\infty$.

Define the mean $\overline{m}_\Phi = \sum_i \Phi(x_i)/n$, and $\widehat{p(x)} = \sum_j K(x_j, x)/n$

$$||\Phi(x_i) - \overline{m}_\Phi||^2 \sim c - 2\widehat{p(x_i)}$$



1.Kernel based methods

1.3. The very particular case of kernel PCA with a Radial Base Kernel

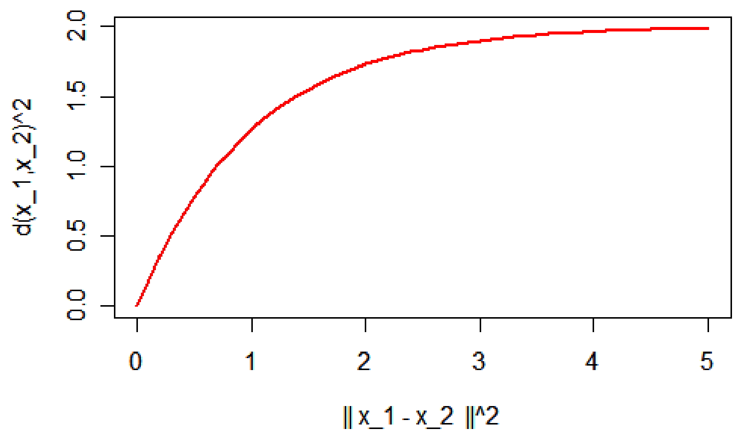
Define

$$K(x, y) = \exp(-||x - y||^2/\sigma).$$

What can we say about $\Phi()$?

The corresponding distance function:

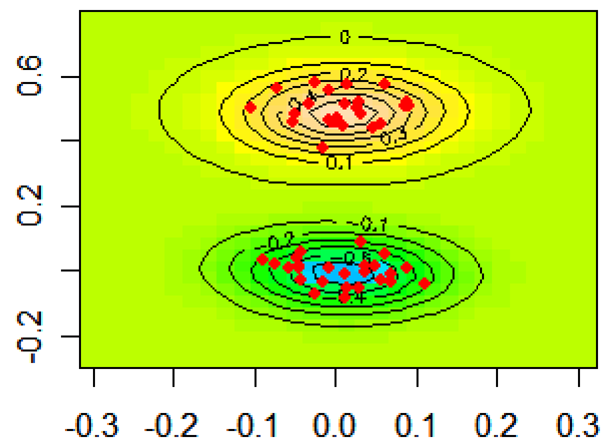
$$d_\Phi(x_1, x_2)^2 = 2(1 - exp(-||x_1 - x_2||^2/\sigma)).$$



Observe: the distance can not be arbitrarly large.

Useful to understand it using the link with *Classical Dimensional Scaling***.**

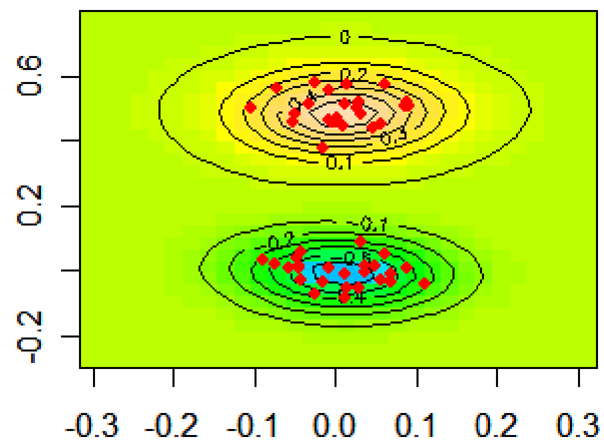
1.3. The very particular case of kernel PCA with a Radial Base Kernel



Not obvious what kernel PCA stands for in this case.

1.Kernel based methods

1.3. The very particular case of kernel PCA with a Radial Base Kernel



Not obvious what kernel PCA stands for in this case.

In the following we motivate that KPCA is sensitive to the densities of the observations.

1.Kernel based methods

1.3. The very particular case of kernel PCA with a Radial Base Kernel

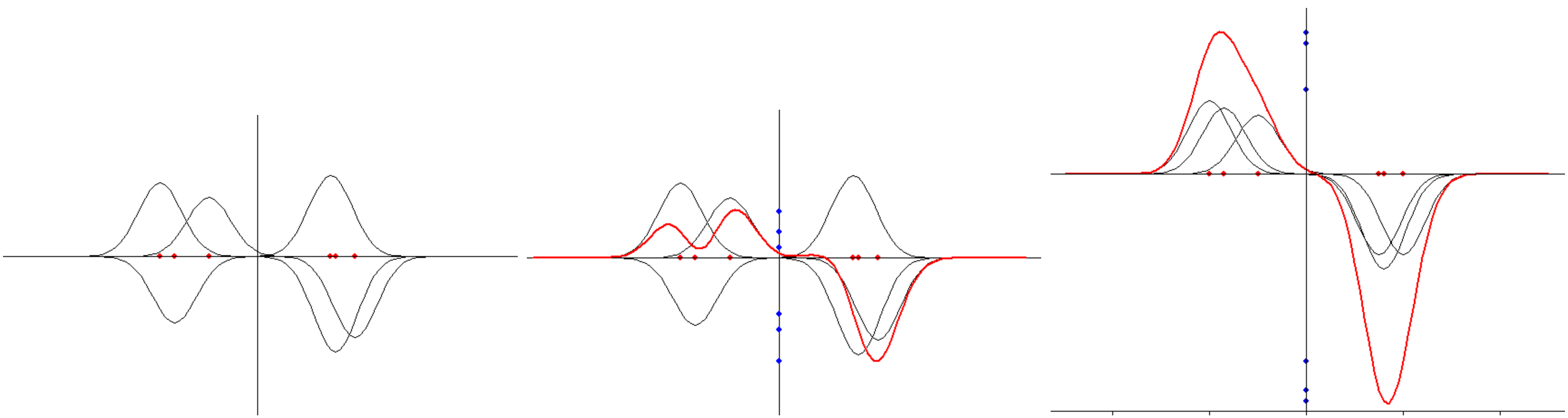
Property

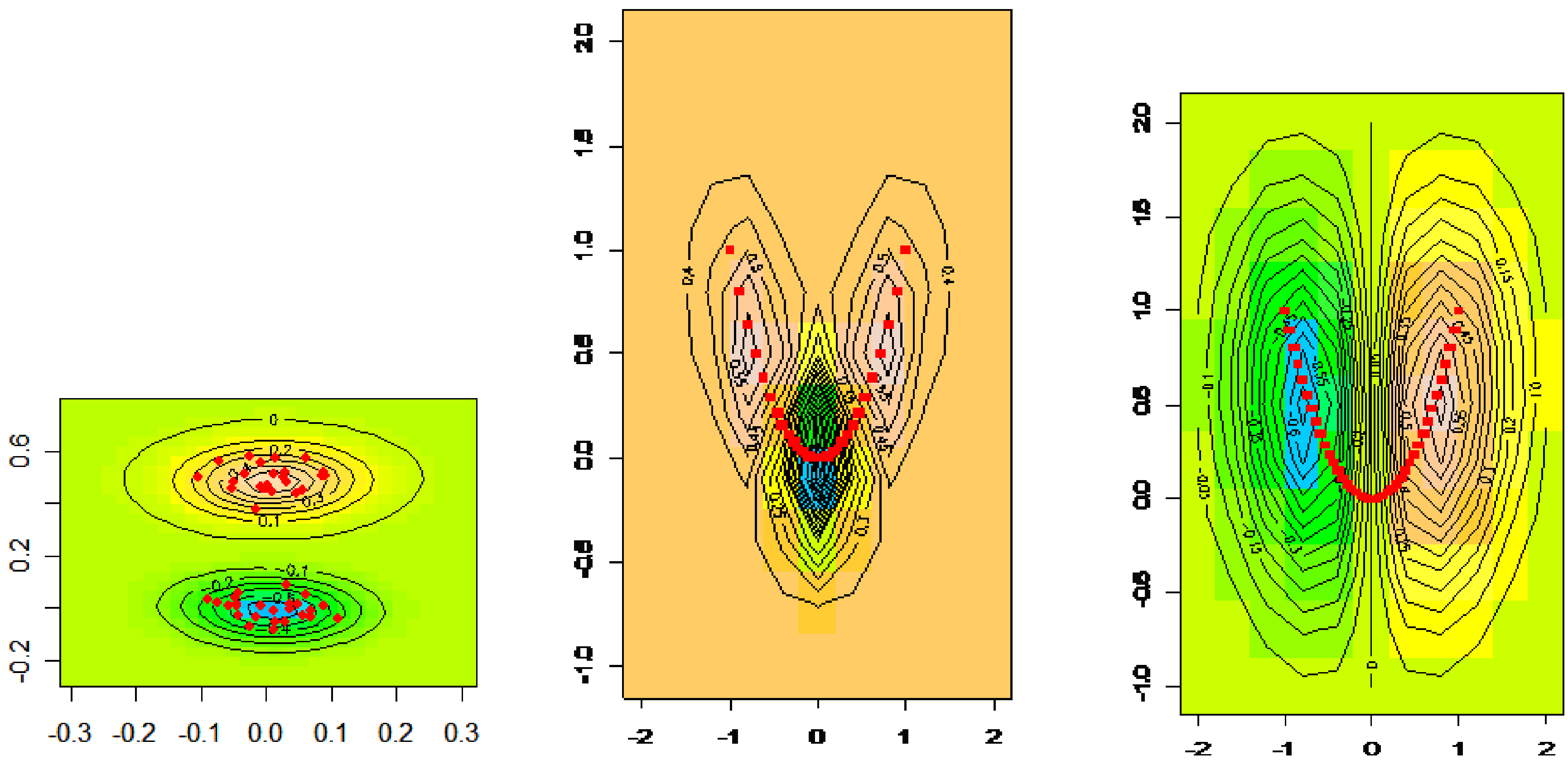
Define:

$$f_\alpha(x) := \sum_i \alpha_i K(x_i, x),$$

The projection function of the first principal component of KPCA (no centered) is the solution $f_\alpha(\cdot)$ of::

$$\max_\alpha \sum_j (f_\alpha(x_j))^2 \text{ with appropiate boundry conditions}$$





2. Issues related to KPCA

2.1. Need for robust versions (work with M. Debruyne, M. Hubert)

2. Issues related to KPCA

2.1. Need for robust versions (work with M. Debruyne, M. Hubert)

The influence function can be calculated and is not always bounded.

Good idea to work with bounded kernels.

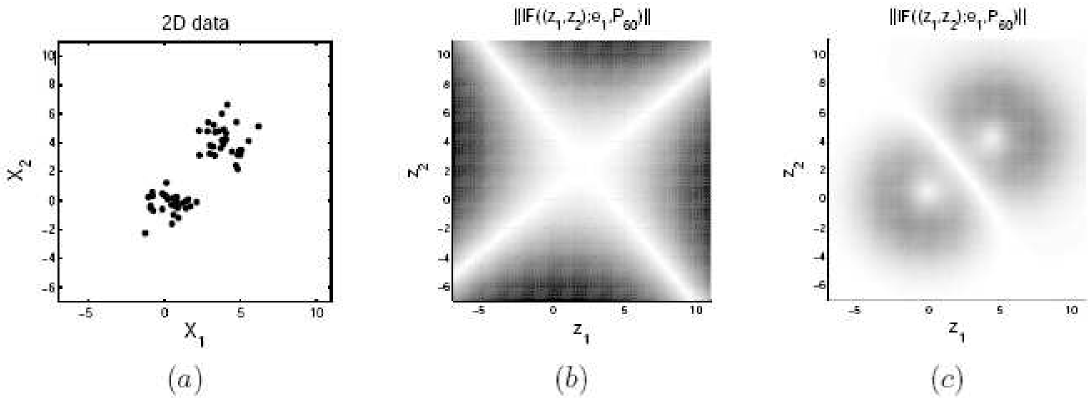
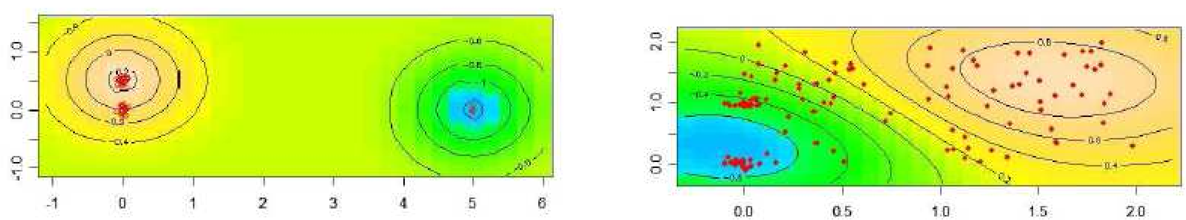


Figure 1: (a): Simple 2D data example; $(b)-(c)$: $||IF(z;e_1,P_{60})||$ as a function of z. White represents values equal to 0, large values tend to black. (b): linear kernel; (c) RBF kernel.

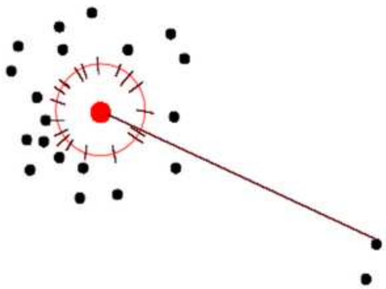
Even with RBK we can have problems:

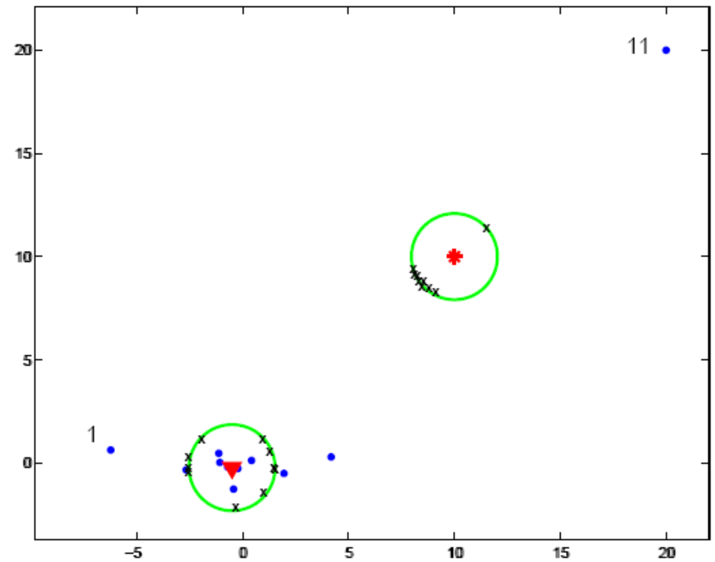


Many robust methods for PCA; how to transpose them to KPCA?

Spherical KPCA

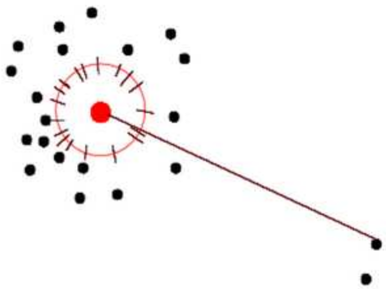
We adapt Spherical PCA (Marron et al.)

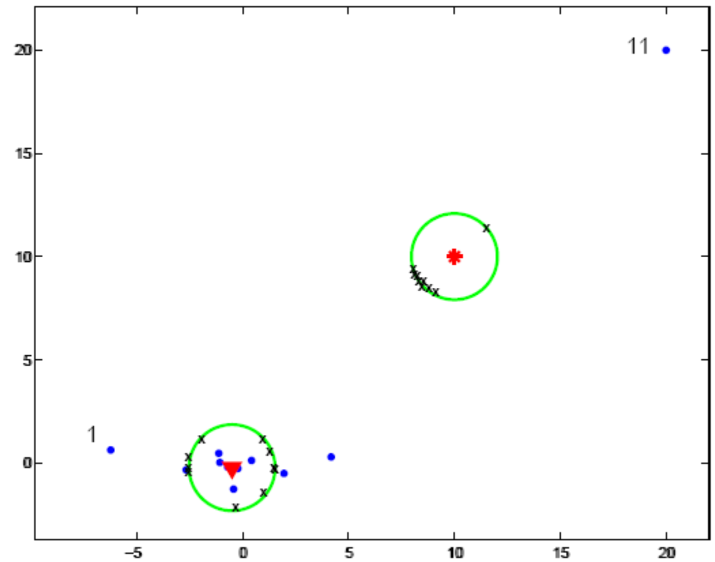




Spherical KPCA

We adapt Spherical PCA (Marron et al.)





Idea

1. **Look for θ such that $\{\frac{x_i-\theta}{||x_i-\theta||}\}$ equals 0**

   To obtain θ: iterate

$$\theta^{(m)} = \frac{\sum_i w_i x_i}{\sum w_i} \textbf{ con } w_i = \frac{1}{||x_i - \theta^{(m-1)}||}$$

2. **Apply PCA to $\{\frac{x_i-\theta}{||x_i-\theta||}\}$.**

1. Look for θ such that $\{\frac{x_i-\theta}{||x_i-\theta||}\}$ equals 0.

To obtain θ, iterate

$$\theta^{(m)} = \frac{\sum_i w_i x_i}{\sum w_i} \textbf{ con } w_i = \frac{1}{||x_i - \theta^{(m-1)}||}$$

2. Apply PCA to $\{\frac{x_i-\theta}{||x_i-\theta||}\}$.

1. Look for θ such that $\{\frac{x_i-\theta}{||x_i-\theta||}\}$ equals 0.

To obtain θ, iterate

$$\theta^{(m)} = \frac{\sum_i w_i x_i}{\sum w_i} \textbf{ con } w_i = \frac{1}{||x_i - \theta^{(m-1)}||}$$

Observe: the optimal θ is of the form:

$$\sum_i \gamma_i x_i.$$

Rewrite the calculations in terms of γ:

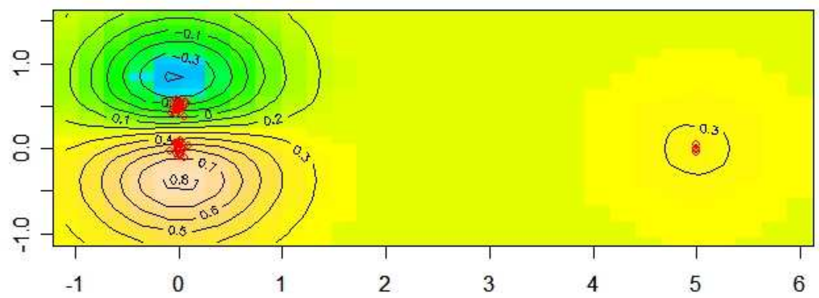
$$\gamma^{(m)} = \frac{w}{\sum w_i} \qquad \textbf{con } w_i^{-1} = \sqrt{K(x_i, x_i) - 2\sum_k \gamma_k^{(m-1)} K(x_i, x_k) + \sum_{k,l} K(x_k, x_l)}$$

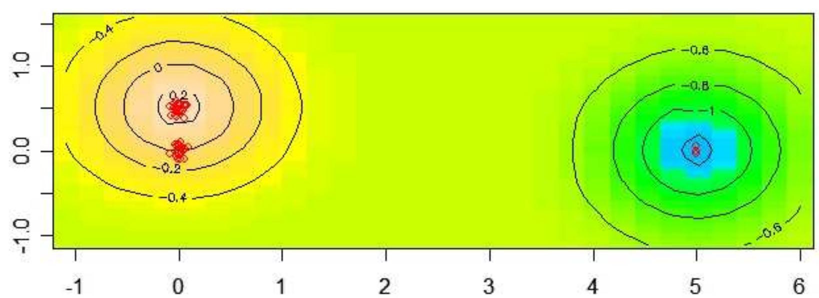
2. Apply PCA to $\{\frac{x_i-\theta}{||x_i-\theta||}\}$.

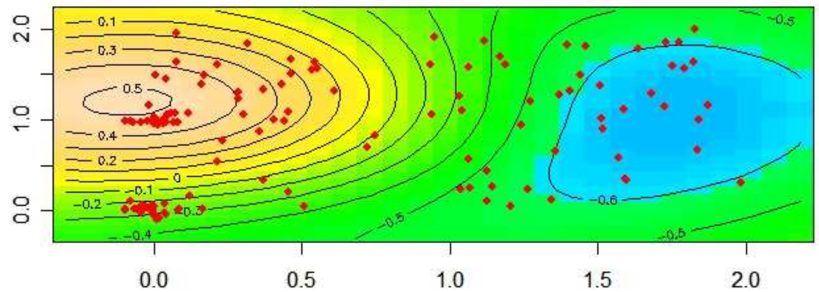
Use the kernel

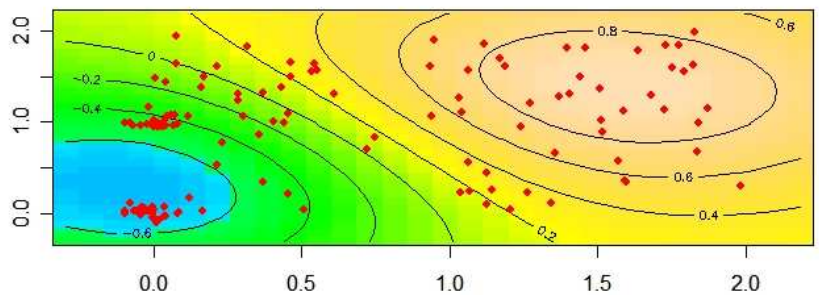
$$K^*(x_i, x_j) =$$
$$\frac{K(x_i, x_j) - \sum_k \gamma_k K(x_i, x_k) - \sum_k \gamma_k K(x_j, x_k) + \sum_{k,l} K(x_k, x_l)}{\sqrt{K(x_i, x_i) - 2\sum_k \gamma_k K(x_i, x_k) + \sum_{k,l} K(x_k, x_l)}\sqrt{K(x_j, x_j) - 2\sum_k \gamma_k K(x_j, x_k\rangle + \sum_{k,l} K(x_k, x_l)}}$$

Examples






0.1
0.3
0.5
0.4
0.2
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
0.0
1.0
2.0
0.0
0.5
1.0
1.5
2.0


0.8
0.6
0
-0.2
-0.4
-0.6
0.6
0.4
0.2
0.0
1.0
2.0
0.0
0.5
1.0
1.5
2.0

Weighted KPCA

Idea: introduce fake transformations

$$
\begin{tikzcd}
K(\cdot,\cdot) \arrow[r] & \Phi(\cdot) \arrow[r] & Cov(\Phi(X)) \\
K^{*}(\cdot,\cdot) \arrow[r] & \Phi^{*}(\cdot) \arrow[r] & X^{\Phi^{*t}} X^{\Phi^{*}} \arrow[u, bend right, "\text{Robust estimator for}"']
\end{tikzcd}
$$

Weighted KPCA

Idea: introduce fake transformations

$$
\begin{tikzcd}
K(\cdot,\cdot) \arrow[r] & \Phi(\cdot) \arrow[r] & Cov(\Phi(X)) \\
K^*(\cdot,\cdot) \arrow[r] & \Phi^*(\cdot) \arrow[r] & X^{\Phi^{*t}} X^{\Phi^*} \arrow[u, bend right, "\text{Robust estimator for}"']
\end{tikzcd}
$$

E.g. one can use introduce weights (e.g. by means of Mahalanobis distance):

$$K^* = W(K - 1_w WK - KW1_w + 1_w WKW1_w)W$$

$W = Diag(\{w_i\})$, $1_w = \frac{1}{\sum w_i}1$ **and** w_i **is a function of:**

$$d^2_{mah.}(x_i, \bar{x}) = n \sum_k \frac{(\sum_j \alpha_j^k k(x_i, x_j))^2}{\lambda_k},$$

KPCA using K^* corresponds to PCA with the *Campbell weighted covariance estimator* **using the kernelized Mahalanobis distance.**

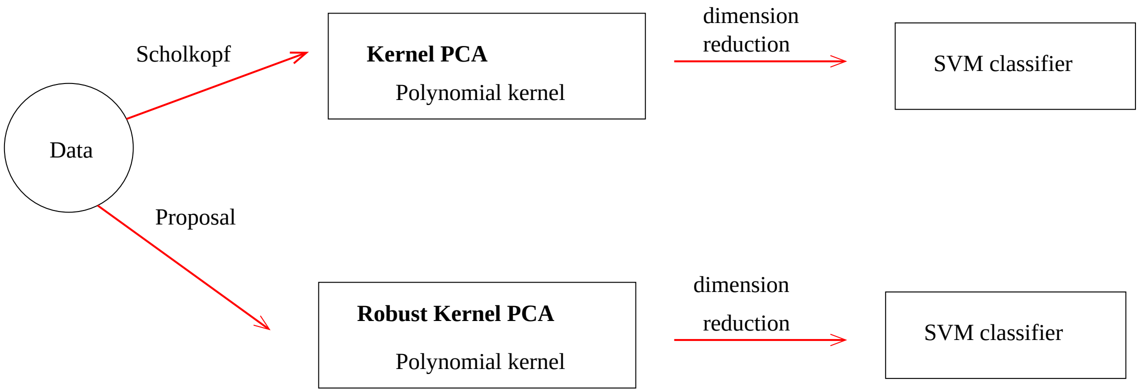
Examples

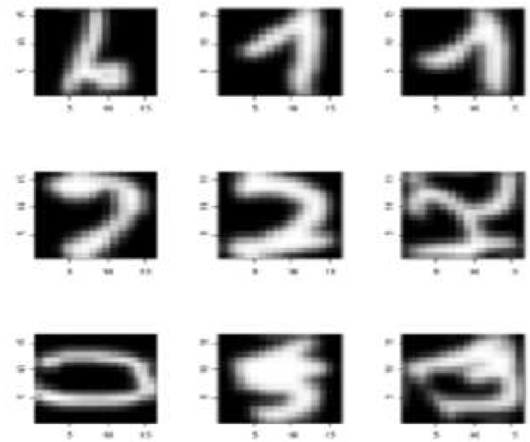
Projecting images on a subspace using (K)PCA to extract background.

Second column: ordinary KPCA;

Third column: robust version.

(K)PCA as a preprocessor for a classifier (SVM) of digits (USPS).





Some outliers.

Classification error standard KPCA vs robust KPCA:

	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*
120	5.4 / *5.3*	4.8 / *4.7*	**5.0** / *5.1*	6.2/ *5.9*	7.5 / *7.3*	**8.5** / *8.7*

Rows: # of components used; Columns: degree of polynomial kernel

2. Issues related to KPCA

2.2. Detecting influential variables

Anova KPCA

(Inspired by work of Yoon Lee for classification)

Instead of

$$K(x,y) = \exp(-||x-y||^2/\sigma),$$

we use

$$K_{\beta}(x,y) = \beta_1 \exp(-(x_1-y_1)^2/\sigma) + \cdots + \beta_d \exp(-(x_d-y_d)^2/\sigma).$$

Anova KPCA

(Inspired by work of Yoon Lee for classification)

Instead of

$$K(x,y) = \exp(-||x-y||^2/\sigma),$$

we use

$$K_{\beta}(x,y) = \beta_1 \exp(-(x_1-y_1)^2/\sigma) + \cdots + \beta_d \exp(-(x_d-y_d)^2/\sigma).$$

The optimization problem:

$$\max_{\alpha,\beta} \sum_j (f_{\alpha,\beta}(x_j))^2 \ \textbf{con} \ f_{\alpha,\beta}(x) = \sum_i \alpha_i K_{\beta}(x_i,x), \ \textbf{s.a.} \ ||\beta||_1 \le c, \ ||\beta||_2 = 1.$$

To get a solution we alternate:

- **optimize over α, fixing β:**

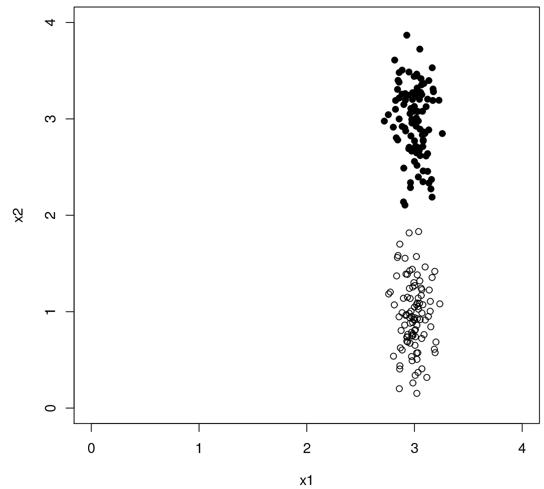
  leads to KPCA;

- **optimize over β, fixing α:**

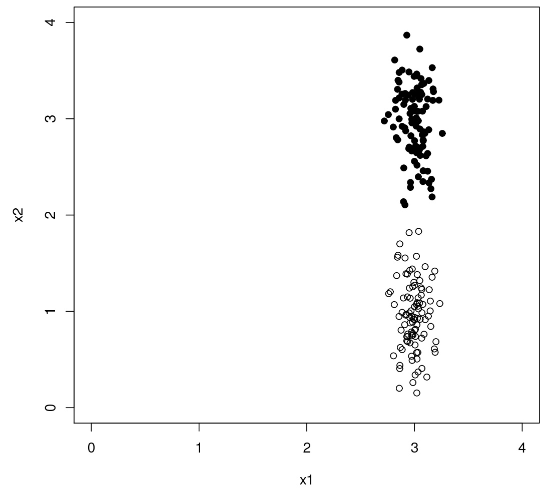
  leads to a cuadratic optimization problem with restrictions.

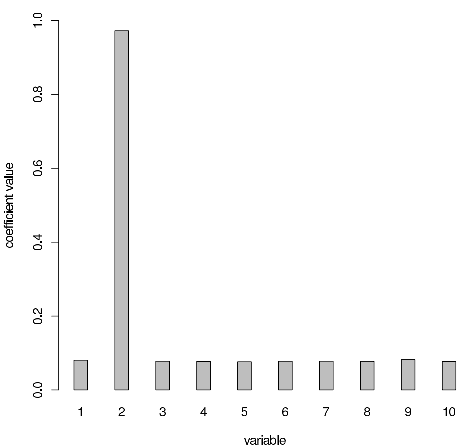
Example 1

10 dimensional data set; $(x_3, \cdots, x_{10})$ **de** $\mathcal{N}(0, 3.5^2)$ **y** (x_1, x_2)**:**



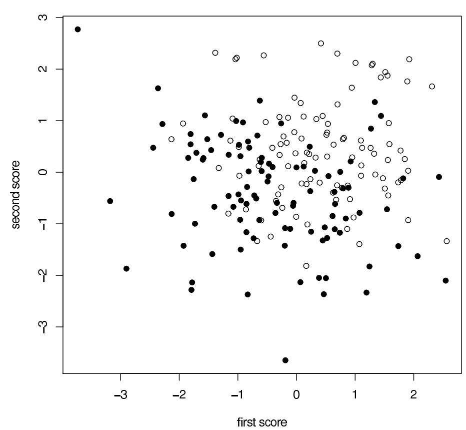
10 dimensional data set; $(x_3, \cdots, x_{10})$ de $\mathcal{N}(0, 3.5^2)$ y (x_1, x_2):





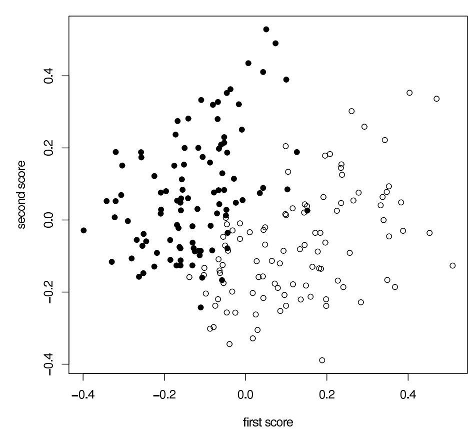
Weightings

(β_i's)



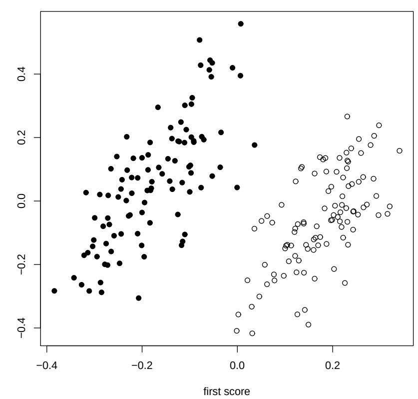
Projections

PCA



Projections

Kernel PCA

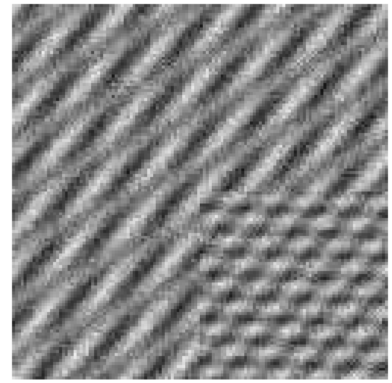


Projections

proposed method

Example 2: segmentation of fringe patterns

Task: assign each pixel to the pattern it belongs to.



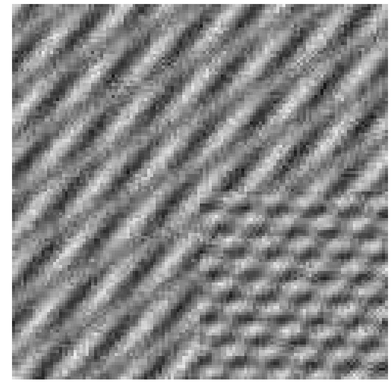
Variables: magnitud of the response to 16 (=d) filters tuned at different frequencies.

$n = 128 \times 128$ **pixels.**

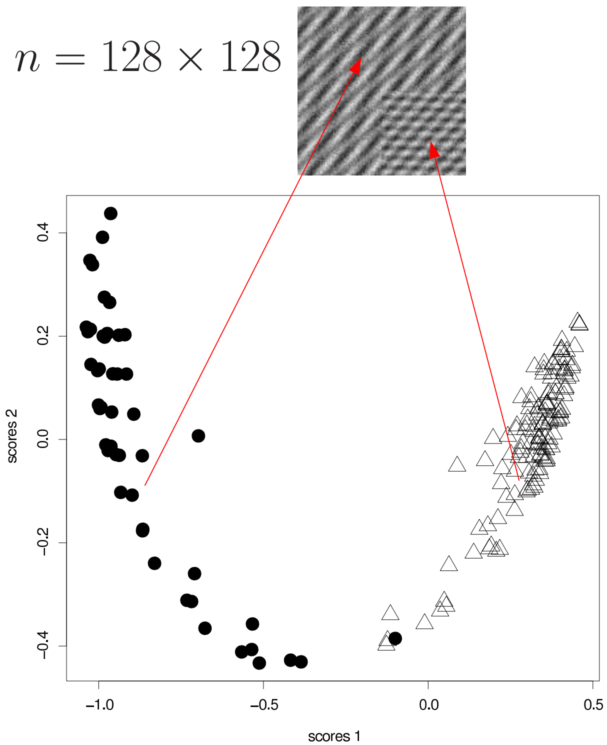
Example 2: segmentation of fringe patterns

Task: assign each pixel to the pattern it belongs to.

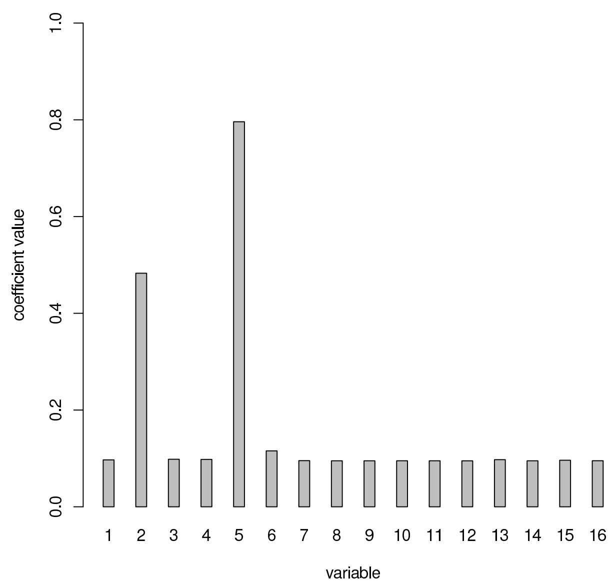
Variables: magnitud of the response to 16 (=d) filters tuned at different frequencies.



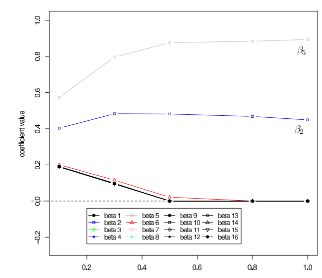
$n = 128 \times 128$



Projections



Weightings



Effect of c

2. Issues related to KPCA

2.3. KPCA and random projections

Motivation:

In case of many observations, because of its dimension, working with K becomes computationally intractable.

2. Issues related to KPCA

2.3. KPCA and random projections

Motivation:

In case of many observations, because of its dimension, working with K becomes computationally intractable.

Idea:

Generate a new (low dimensional) data matrix $\mathbf{Z}$ and apply PCA on $\mathbf{Z}$.

Choose $\mathbf{Z}$ such that $K_z := \mathbf{Z}\mathbf{Z}^t$ is a good approximation of K: e.g $E(K_z) = K$.

2. Issues related to KPCA

2.3. KPCA and random projections

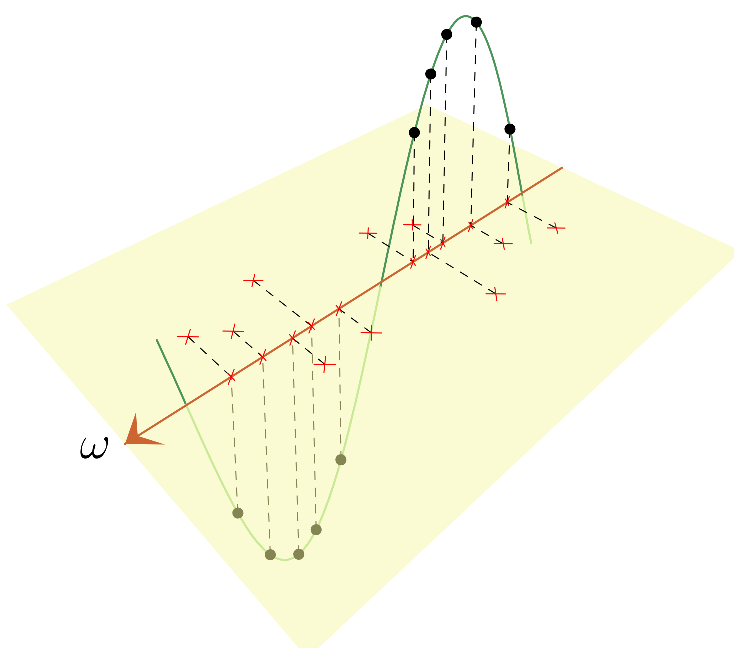
Motivation:

In case of many observations, because of its dimension, working with K becomes computationally intractable.

Idea:

Generate a new (low dimensional) data matrix $\mathbf{Z}$ and apply PCA on $\mathbf{Z}$.

Choose $\mathbf{Z}$ such that $K_z := \mathbf{Z}\mathbf{Z}^t$ is a good approximation of K: e.g $E(K_z) = K$.



for different w_k, b_k, calculate: $z_{i,k} = \sqrt{2}cos(w_k^t x_i + b_k)$

Final remark

Although kernel based methods have been around for a while, many open questions.

If the choice of first names is a good trend detector, ...

4.12 **Control óptimo de una epidemia** (Reporte de Tesis)

Kernel Prieto Moreno, kernel@ciencias.unam.mx *(IIMAS, UNAM)*
Coautor: María de Lourdes Esteva Peralta

El virus de la influenza causa problemas médicos y sociales sustancial
tramos en una pandemia ocasionada por el virus de influenza AH1N1. E
estrategias para mitigar una epidemia usando teoría de control óptimo y e

En el primer modelo se uso vacunación, en el segundo campaña educa
educativa con administración de medicamentos.

... kernels have a promising future!

Thanks

References/preprints can be found at http://www.cimat.mx/~horebeek