Machine Learning

Lecture 7: Representation Learning

7.1 Principal Component Analysis

7.2. Metric Learning

Marc Sebban & Amaury Habrard

Hubert Curien lab, UMR CNRS 5516
University of Jean Monnet Saint-Étienne (France)

Academic year 2016-2017

Representation Learning

A training set $S = \{z_i = (\mathbf{x_i}, y_i)\}_{i=1}^{m}$ is composed of m examples ($\mathbf{x_i} \in \mathcal{X}$ and $y \in \mathcal{Y}$) independently and identically distributed (i.i.d.) according to an unknown distribution $\mathcal{D}_{\mathcal{Z}}$ over the joint space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

Representation learning

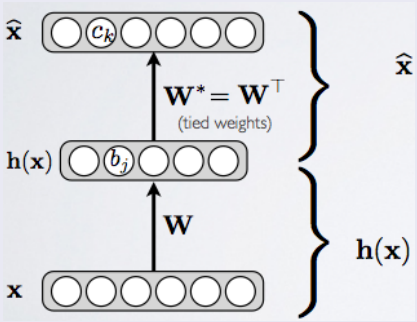
Representation learning refers to techniques which automatically learn an (implicit or explicit) feature space $\mathcal{X}'$ which

- is computationally convenient to process,
- allows us to overcome the limitations of hand-crafted features $\mathbf{x_i} \in \mathcal{X}$ which often require expensive human expertise,
- improves the machine learning tasks by better representing the problem at hand.

Representation Learning

Already studied representation learning techniques

- **Deep learning**, where each layer can be interpreted as an intermediate representation (autoencoder, CNN). **Success stories**: *computer vision, speech recognition, NLP, Go, googlecar, etc.*

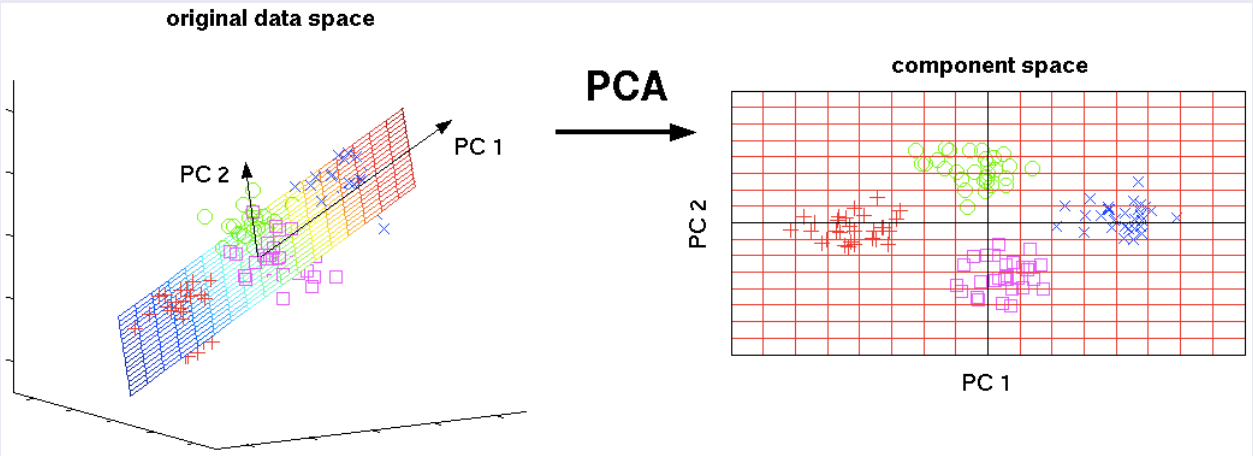


- **SVM**, where data are projected onto a (potentially) infinite space via the kernel trick. The final classifier takes the form of a linear combination of similarities to support vectors ($\approx$ new features).

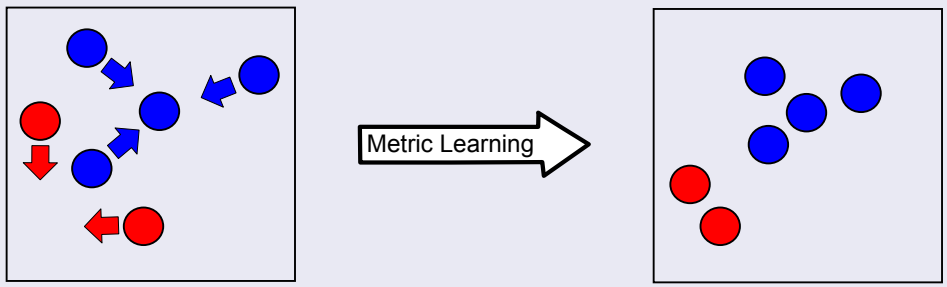
Representation Learning

Today

- Unsupervised representation learning: **Principal Component Analysis**



- Supervised representation learning: **Metric Learning**



Introduction on PCA

Definition

Principal component analysis (PCA), also known as the *Karhunen-Love* transform, is widely used for:

- **Dimensionality reduction**: it projects data points living in a d-dimensional space onto a M-dimensional subspace, where $M < d$. If $M = 2$, PCA allows **Data visualization** while preserving the variance of the original data.
- **Feature extraction**: it generates new uncorrelated (i.e. without redundancies) meaningful features.

Example of dimensionality reduction

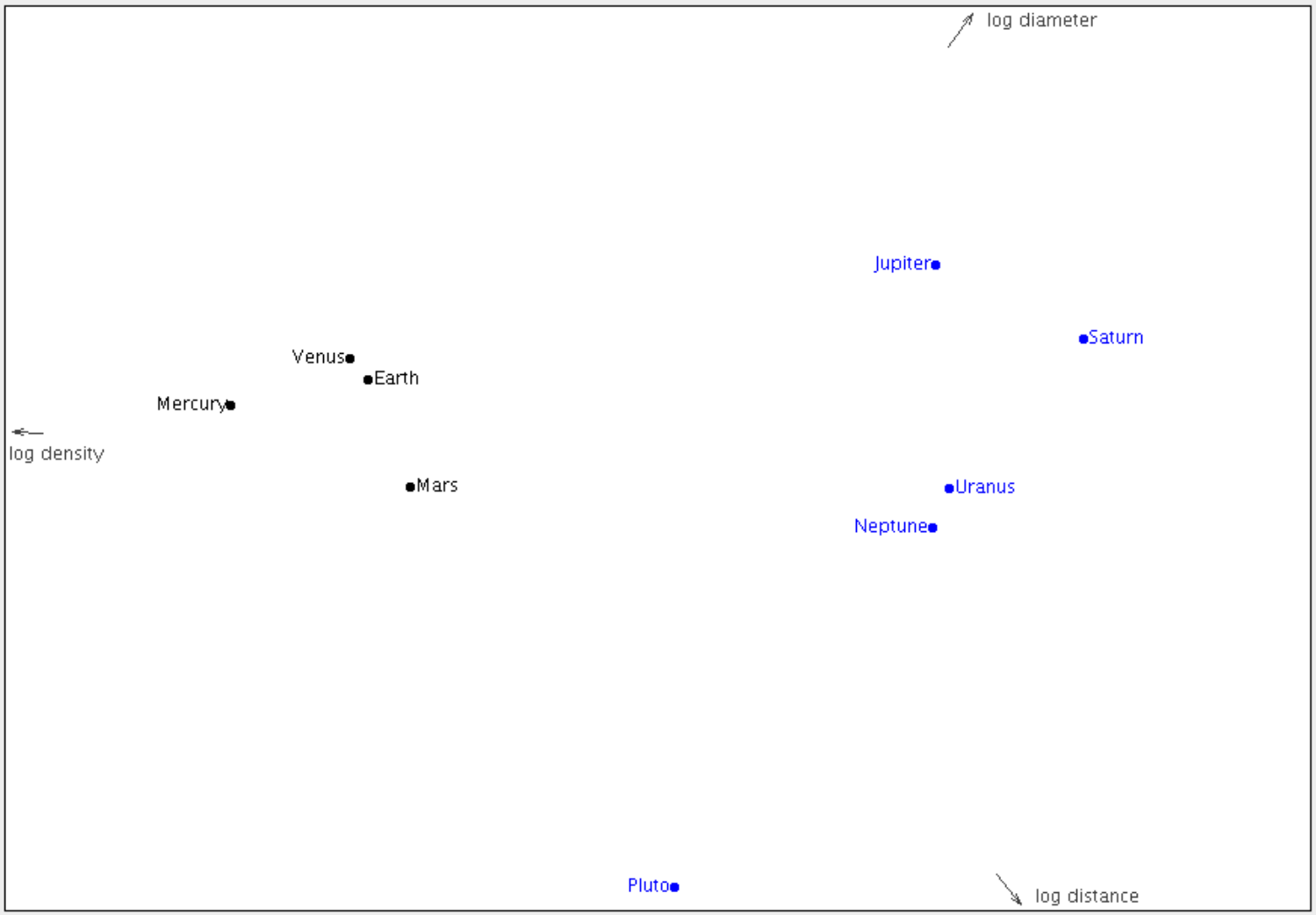
Dimensionality Reduction

Three important features of the planets in our solar system are (i) the **distance** to the Sun, (ii) the equatorial **diameter** and (iii) the **density**.

	Distance	Diameter	Density
Mercury	0.387	4878	5.42
Venus	0.723	12104	5.25
Earth	1.000	12756	5.52
Mars	1.524	6787	3.94
Jupiter	5.203	142800	1.31
Saturn	9.539	120660	0.69
Uranus	19.18	51118	1.29
Neptune	30.06	49528	1.64
Pluto	39.53	2300	2.03

Since a 3D-plot is not always very readable, can we find a 2D-plot of the data such that **close points in that new space mean similar planets in the original 3D-space**?

Example of dimensionality reduction

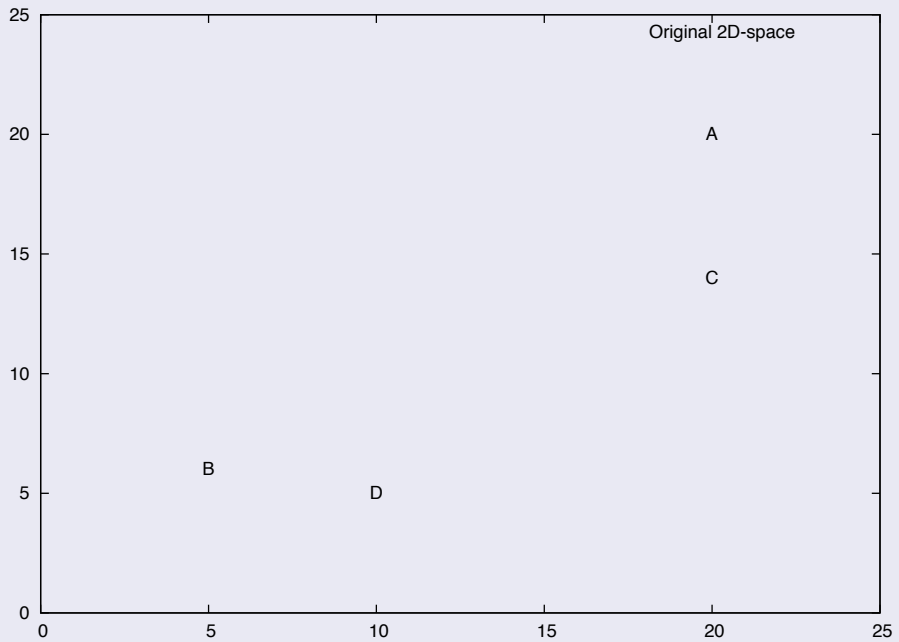


Example of Feature Extraction

Feature Extraction

Let us assume that four students (A,B,C,D) got the following grades for 2 exams.

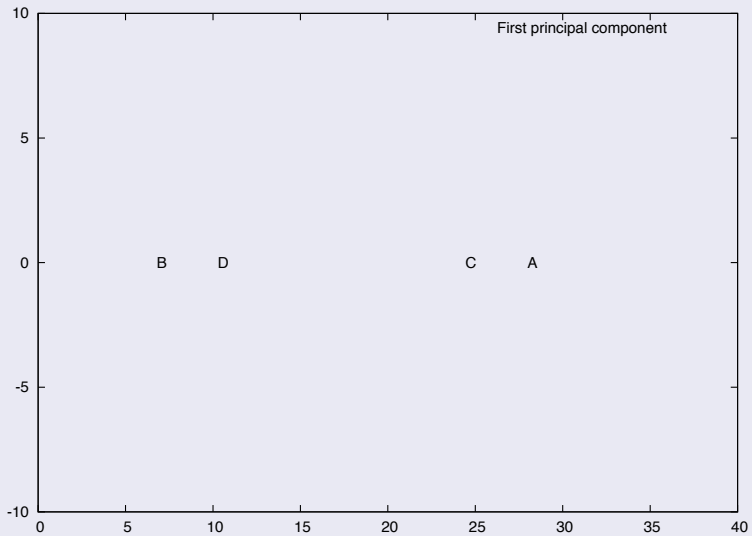
	Exam 1	Exam 2
A	20	20
B	5	6
C	20	14
D	10	5

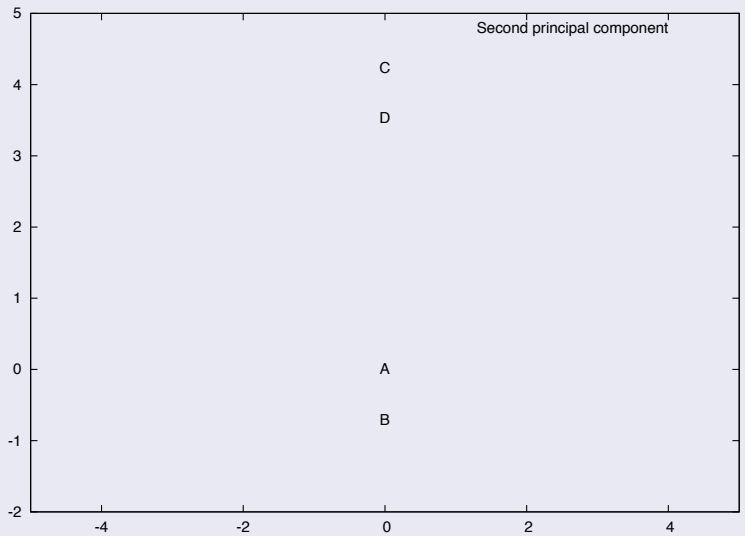


Feature Extraction

	Exam 1	Exam 2
A	20	20
B	5	6
C	20	14
D	10	5

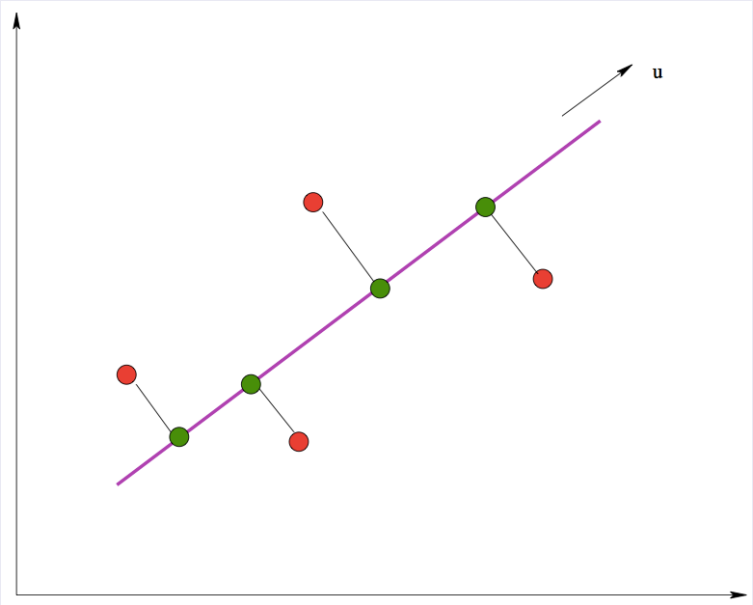
PCA allows us to generate two **new uncorrelated features** (i.e. orthogonal vectors) bringing useful information: the first component expresses most of the information contained in the original 2D-space (**students who PASS or FAIL**) while the second groups together students having put in the same effort (**a lot of OR a small**) for both the exams.





Goal of PCA

The goal of PCA is to **linearly** project the data $\mathbf{x}_i \in \mathbb{R}^d$ onto a space having dimensionality $M < d$ such that **close points in that new M-space mean similar examples in the original d-space**.



Here, $d = 2$ and $M = 1$. We have to define the direction of this space using a 2-dimensional vector $\mathbf{u}$.

Maximization of the variance of the projected data

Let us suppose that the training data are **zero mean** (that is, $\forall i$, $\mathbf{x}_i$ is changed into $\mathbf{x}_i - \bar{\mathbf{x}}$).

PCA seeks a new space of size $M < d$ by applying a linear transformation $\mathbf{U}$ on the original data. The new representation of a training data $\mathbf{x}_i$, denoted by $\mathbf{t}_i$, is given by:

$$\mathbf{t}_i = \mathbf{U}^T \mathbf{x}_i$$

where $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_M)$ is a $d \times M$ matrix of new bases and $\mathbf{u}_j \in \mathbb{R}^d$. We impose that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, that is:

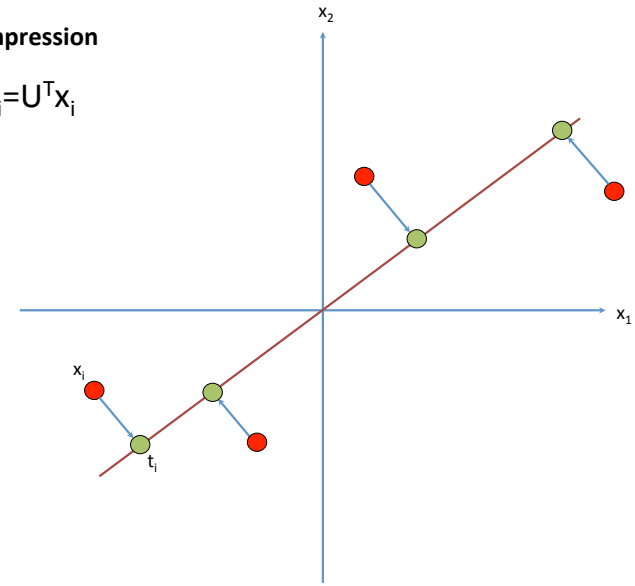
- every new feature $\mathbf{u}_i$ is linearly independent from the others,
- $\forall j$, ${\mathbf{u}_j}^T \mathbf{u}_j = 1$.

Note that each $\mathbf{t}_i$ is a linear combination of the original features

Reconstruction from compressed representation

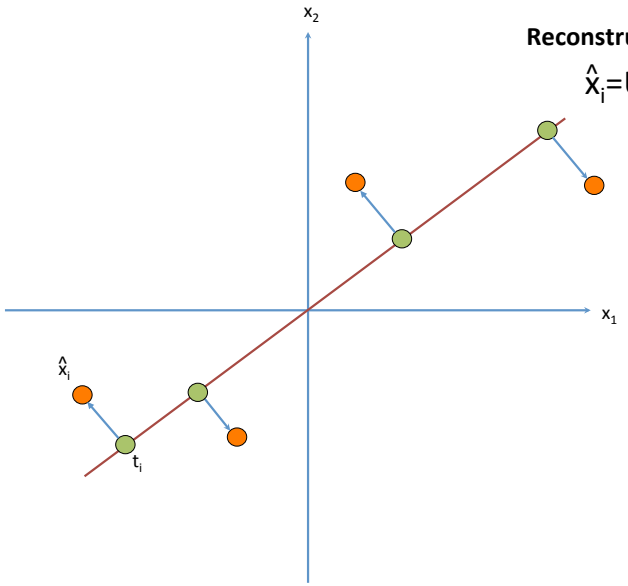
Compression

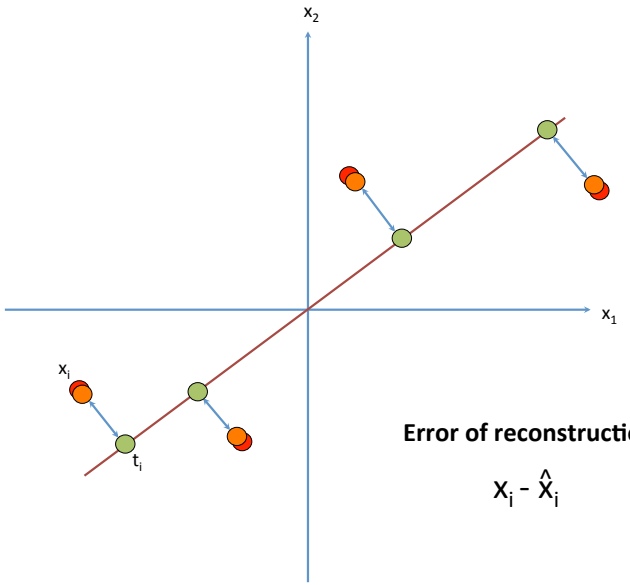
$t_i = U^T x_i$



Reconstruction

$\hat{x}_i = U t_i$



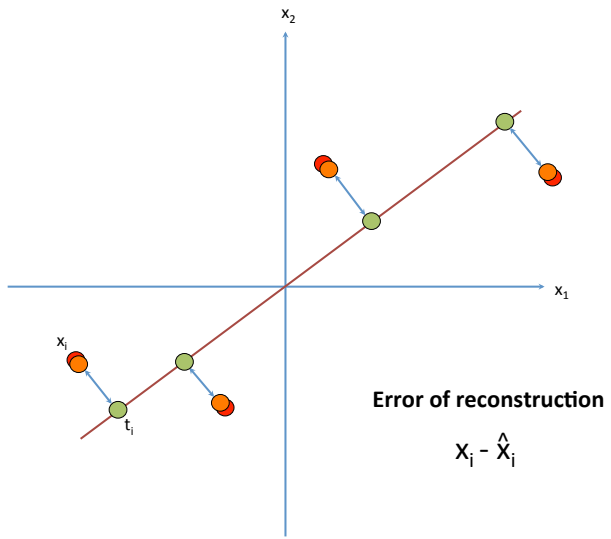


Error of reconstruction

$x_i - \hat{x}_i$

Reconstruction from compressed representation

- If $x_i \in \mathbb{R}^d$, then PCA can generate, at the most, d new components, i.e. $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_d)$ is a $d \times d$ matrix of new bases.
- If $\mathbf{U}$ is composed of d new bases, then it is possible to perfectly reconstruct the data (i.e. $\mathbf{U}$ is bijection).
- If $\mathbf{U}$ is only composed of M new bases ($M < d$), then some information is lost by the projection onto the M-dimensional space and then the reconstruction is not "perfect".
- The aim of PCA is to minimize this error of reconstruction.



Maximization of the variance of the projected data

Let $\hat{\mathbf{x}}_i = \mathbf{U}\mathbf{t}_i$ (with $\mathbf{t}_i = \mathbf{U}^T\mathbf{x}_i$) be the reconstruction of the original vector $\mathbf{x}_i$ using the transformation $\mathbf{U}$. The objective of PCA is to optimize $\mathbf{U}$ such that the mean square error $J(\mathbf{U})$ between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ is as small as possible.

$$
\begin{aligned}
\min_{\mathbf{U}} J(\mathbf{U}) &= \min_{\mathbf{U}} \frac{1}{n} \sum_i (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \\
&= \min_{\mathbf{U}} \frac{1}{n} \sum_i (\mathbf{x}_i - \mathbf{U}\mathbf{U}^T\mathbf{x}_i)^T(\mathbf{x}_i - \mathbf{U}\mathbf{U}^T\mathbf{x}_i) \\
&= \min_{\mathbf{U}} \frac{1}{n} \sum_i ({\mathbf{x}_i}^T\mathbf{x}_i - 2{\mathbf{x}_i}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i + {\mathbf{x}_i}^T\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i) \\
&= \min_{\mathbf{U}} \frac{1}{n} \sum_i ({\mathbf{x}_i}^T\mathbf{x}_i - {\mathbf{x}_i}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i) \\
&= \min_{\mathbf{U}} \frac{1}{n} \sum_i {\mathbf{x}_i}^T\mathbf{x}_i - \frac{1}{n} \sum_i {\mathbf{x}_i}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i
\end{aligned}
$$

Maximization of the variance of the projected data

$$
\begin{aligned}
\min_{\mathbf{U}} J(\mathbf{U}) &= \min_{\mathbf{U}} \frac{1}{n} \sum_i \mathbf{x}_i{}^T \mathbf{x}_i - \frac{1}{n} \sum_i \mathbf{x}_i{}^T \mathbf{U}\mathbf{U}^T \mathbf{x}_i \\
&= \min_{\mathbf{U}} \frac{1}{n} \sum_i \mathbf{x}_i{}^T \mathbf{x}_i - \frac{1}{n} \sum_i \mathbf{t}_i{}^T \mathbf{t}_i \\
&= \min_{\mathbf{U}} tr \left(\frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i{}^T - \frac{1}{n} \sum_i \mathbf{t}_i \mathbf{t}_i{}^T \right) \\
&= \min_{\mathbf{U}} tr \left(\frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i{}^T - \frac{1}{n} \sum_i \mathbf{U}^T \mathbf{x}_i \mathbf{x}_i{}^T \mathbf{U} \right) \\
&= \min_{\mathbf{U}} tr \left(\frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i{}^T \right) - tr \left(\frac{1}{n} \sum_i \mathbf{U}^T \mathbf{x}_i \mathbf{x}_i{}^T \mathbf{U} \right)
\end{aligned}
$$

Maximization of the variance of the projected data

$$\begin{aligned}\min_{\mathbf{U}} J(\mathbf{U}) &= \min_{\mathbf{U}} tr\left(\frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i{}^T\right) - tr\left(\mathbf{U}^T(\frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i{}^T)\mathbf{U}\right) \\ &= \min_{\mathbf{U}} tr\left(\Sigma\right) - tr\left(\mathbf{U}^T\Sigma\mathbf{U}\right)\end{aligned}$$

where Σ is the covariance matrix of the original data and $\mathbf{U}^T\Sigma\mathbf{U}$ is the **covariance matrix in the new space**. Since $tr(\Sigma)$ does not depend on $\mathbf{U}$, minimizing $J(\mathbf{U})$ boils down to maximizing $\mathbf{U}^T\Sigma\mathbf{U}$, that is,

$$\max_{\mathbf{U}} \mathbf{U}^T\Sigma\mathbf{U}$$

$$\text{s.t } \forall j = 1..M,\ \mathbf{u}_j{}^T\mathbf{u}_j = 1.$$

Closed-Form Solution

Lagrange Multipliers

$$\max_{\mathbf{U}} \mathbf{U}^T \Sigma \mathbf{U}$$

$$\text{s.t } \forall j = 1..M,\ \mathbf{u}_j{}^T \mathbf{u}_j = 1.$$

Introducing Lagrange multipliers (denoted by the feature vector $\lambda = (\lambda_1, ..., \lambda_M)$), we get the unconstrained maximization problem:

$$\max_{\mathbf{U}} \mathbf{U}^T \Sigma \mathbf{U} + \lambda(1 - \mathbf{U}^T \mathbf{U}).$$

Let us consider the first component $\mathbf{u}_1$ of the new space. Finding $\mathbf{u}_1$ requires to solve:

$$\frac{\partial \mathbf{U}^T \Sigma \mathbf{U} + \lambda(1 - \mathbf{U}^T \mathbf{U})}{\partial \mathbf{u}_1} = 0$$

Closed-Form Solution

Derivatives of matrices and vectors

Let $v \in \mathbb{R}^d$ a vector and M a $d \times d$ matrix:

$$\frac{\partial v^T M v}{\partial v} = (M + M^T)v.$$

If M is symmetric, then $M = M^T$ and

$$\frac{\partial v^T M v}{\partial v} = 2Mv.$$

Derivatives

Applying the previous property on $\frac{\partial \mathbf{U}^T \Sigma \mathbf{U} + \lambda(1 - \mathbf{U}^T \mathbf{U})}{\partial \mathbf{u}_1} = 0$, we get:

$$\Sigma \mathbf{u}_1 = \lambda_1 \mathbf{u}_1,$$

which says that $\mathbf{u}_1$ **must be an eigenvector of** Σ.

Eigenvectors and Eigenvalues

Eigenvalue

The eigenvalues of Σ are the solutions λ to the following *characteristic equation*:

$$det(\Sigma - \lambda \mathbf{I}) = 0.$$

Eigenvector

An eigenvector $\mathbf{u}_1$ of Σ corresponding to the eigenvalue λ_1 is defined as follows:

$$\Sigma \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

Thus, to get $\mathbf{u}_1$, one has to solve the following equation $(\Sigma - \lambda_1 \mathbf{I})\mathbf{u}_1 = 0$.

Closed-Form Solution

We aim at maximizing $\mathbf{U}^T \Sigma \mathbf{U}$

Derivatives

The solution is given by $\Sigma \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$. We deduce that:

$$\begin{aligned} & \Sigma \mathbf{u}_1 = && \lambda_1 \mathbf{u}_1 \\ \Leftrightarrow \quad & \mathbf{u}_1{}^T \Sigma \mathbf{u}_1 = && \mathbf{u}_1{}^T \lambda_1 \mathbf{u}_1 \\ \Leftrightarrow \quad & \mathbf{u}_1{}^T \Sigma \mathbf{u}_1 = && \lambda_1 \mathbf{u}_1{}^T \mathbf{u}_1 \\ \Leftrightarrow \quad & \mathbf{u}_1{}^T \Sigma \mathbf{u}_1 = && \lambda_1 \end{aligned}$$

So the variance will be maximum when we set $\mathbf{u}_1$ equal to the eigenvector having the largest eigenvalue λ_1. **This eigenvector is known as the first principal component**.

Closed-Form Solution

Generalization to an M-dimensional projection space

Considering the general case of an M-dimensional projection space, we get:

$$\Sigma \mathbf{U} = \mathbf{U}\lambda \Leftrightarrow \mathbf{U}^T \Sigma \mathbf{U} = \lambda,$$

where $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_M)$ and $\lambda = (\lambda_1, ..., \lambda_M)$. The optimal linear projection for which the variance of the projected data is maximized is now defined by the M eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_M$ of the data covariance matrix Σ corresponding to the M largest eigenvalues $\lambda_1, ..., \lambda_M$.

Properties of the components

- The eigenvalues of Σ are always positive because Σ is PSD.
- The number of components is equal to the number of non zero eigenvalues.
- The total variance of the original data is $V = tr(\Sigma)$. We can show that:

$$V = tr(\Sigma) = tr(\lambda) = \lambda_1 + \lambda_2 + ... + \lambda_d.$$

- When we project the data onto a M-dimensional space corresponding to the first M eigenvectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_M$ associated with the first M largest eigenvalues $\lambda_1, \ldots, \lambda_M$, we get a new covariance matrix $\mathbf{U}\Sigma\mathbf{U}^T$ whose total variance $\hat{V} = \lambda_1 + ... + \lambda_M$.
- Therefore, we can compute the ratio of variance "explained" by the projected data:

$$\frac{\lambda_1 + ... + \lambda_M}{\lambda_1 + ... + \lambda_d}.$$

The higher the ratio, the better the projection.

Maximum variance formulation

Algorithmic complexity of PCA

PCA involves evaluating the mean $\bar{\mathbf{x}}$ and the covariance matrix Σ of the data set and then finding the M eigenvectors of Σ corresponding to the M largest eigenvalues.

- The computational cost of computing the full eigenvector decomposition for a matrix of size $d \times d$ is $\mathcal{O}(d^3)$.
- However, if we are only interested in the the projection onto the first M principal components, efficient techniques exist, such as the *power method* that scale like $\mathcal{O}(Md^2)$, or alternatively we can make use of the EM algorithm.

kernel PCA

kPCA

kernel PCA is an extension of the standard PCA which allows (by the use of the kernel trick) the capture of non-linearity by performing the linear transformation in a Hilbert space.

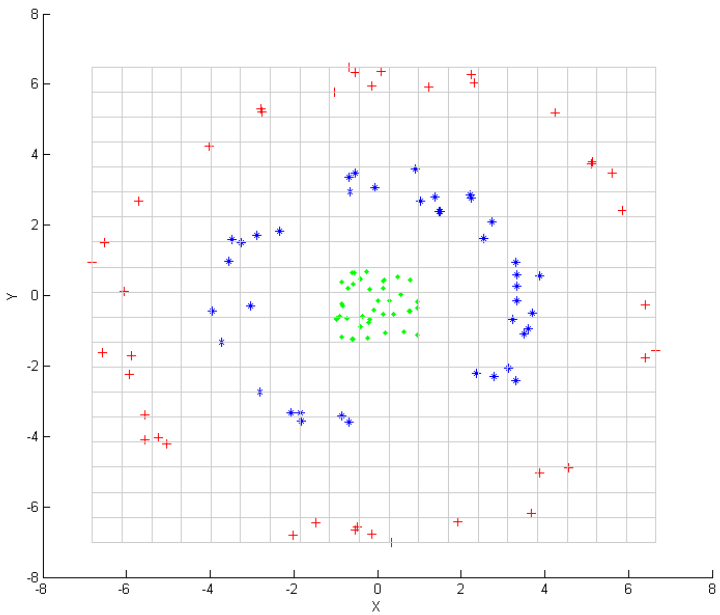
Let us suppose that the training set $S = \{z_i = (\mathbf{x_i}, y_i)\}_{i=1}^{m}$ is composed of m examples, where $\mathbf{x_i} \in \mathbb{R}^d$. While it is in general impossible to get all possible dichotomies of S when $m \geq d$, we know that the points can almost always be linearly separated in a $d \geq m$-dimensional space (related to the notion of shattered points and VC-dimension).

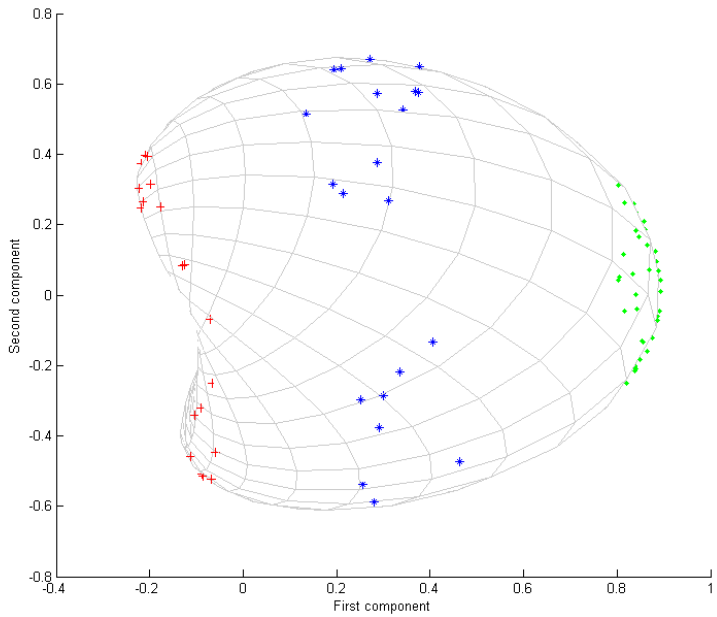
kPCA consists in performing a PCA from the $m \times m$ Gram matrix built w.r.t. a given kernel.

kernel PCA

Illustration

Consider the training data on the left coming from three concentric sets of points. If we (i) use a gaussian kernel (i.e. $K(x, x') = e^{\frac{-||x-x'||^2}{2\sigma^2}}$), (ii) build the gram matrix, and (iii) perform a linear PCA, we get the results on the right which require only one component to linearly retrieve the three groups.





Exercise

The objective is to perform a PCA on the following 8 unlabeled examples lying in a 2-dimensional space

$$X = \{x_i\}_{i=1}^{8} = \{(1,2),(3,3),(3,5),(5,4),(5,6),(6,5),(8,7),(9,8)\}$$

and project them onto the first principal component.

1. Compute the covariance matrix Σ from the zero mean values
2. Solve the characteristic equation $det(\Sigma - \lambda \mathbf{I}) = 0$ to get the two eigenvalues λ_1 and λ_2.
3. Deduce the first unit eigenvector $\mathbf{u}_1$ from the largest eigenvalue λ_1 by solving $(\Sigma - \lambda_1 \mathbf{I})\mathbf{u}_1 = 0$.
4. Compute the part of the total variance explained by this 1-dimensional space.
5. Use $\mathbf{u}_1$ to find the projection $t_1 = \mathbf{u}_1{}^T \mathbf{x}_1$ for the the first training data $\mathbf{x}_1$.