Machine Learning

Kernel-PCA

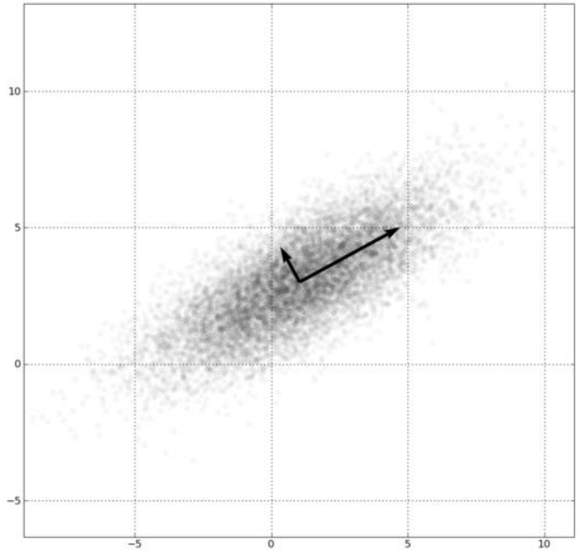
Principal Component Analysis

Problem – high feature dimension:

Especialy in Computer Vision

- Feature is the (5x5) patch → feature vector is in $\mathbb{R}^{25}$
- SIFT is composed of 16 histograms of 8 directions → vector in $\mathbb{R}^{128}$

Idea – the feature space is represented in another **basis**.



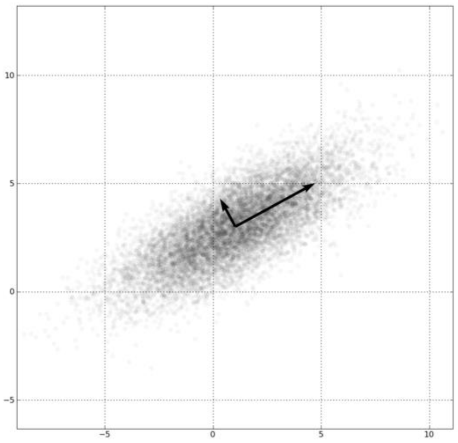
Assumption: the directions of small variances correspond to noise and can be neglected

Approach: project the feature space onto a linear subspace so that the variances in the projected space are maximal

Principal Component Analysis

A simplified example: data are centered, the subspace is one-dimensional, i.e. it is represented by a vector $\|e\|^2=1$. Projection of an x on e is $\langle x,e\rangle$. Hence, the task is

$$\sum_l \langle x^l, e\rangle^2 \to \max_e \qquad \text{s.t. } \|e\|^2=1$$



Lagrangian:

$$\sum_l \langle x^l, e\rangle^2 + \lambda\big(\|e\|^2-1\big) \to \min_\lambda \max_e$$

Gradient with respect to e:

$$\sum_l 2\langle x^l, e\rangle \cdot x^l + 2\lambda e = 2e\sum_l x^l\otimes x^l + 2\lambda e = 0$$

$$e\cdot cov = \lambda e$$

Principal Component Analysis

$$e \cdot cov = \lambda e$$

→ e is an eigenvector and λ is the corresponding eigenvalue of the **covariance matrix**. Which one?

For a pair e und λ the variance is

$$\sum_l \langle x^l, e \rangle^2 = e \cdot \sum_l x^l \otimes x^l \cdot e = e \cdot cov \cdot e = \|e\|^2 \cdot \lambda = \lambda$$

→ chose the eigenvector corresponding to the **maximal** eigenvalue.

Similar approach: project the feature space into a subspace so that the summed squared distance between the points and their projections is minimal → the result is the same.

Principal Component Analysis

Summary (for higher-dimensional subspaces):

1. Compute the covariance matrix $cov = \sum_l x^l \otimes x^l$
2. Find all eigenvalues and eigenvectors
3. Sort the eigenvalues in decreasing order
4. Choose m eigenvectors for the m first eigenvalues (in the order)
5. The $n \times m$ projection matrix consists of m columns, each one is the corresponding eigenvector.

Are projections onto a **linear** subspace good? Alternatives?

Do it with scalar products

The optimal direction vector can be expressed as a linear combination of data points, i.e. it is contained in the subspace that is **spanned** by the data points.

$$e = \sum_i \alpha_i x_i$$

Note: in high-dimensional spaces it may happen that all data points lie in a linear subspace, i.e. do not span the whole space (e.g. the dimension of the space is higher as the number of the data points). It will be important for the feature spaces (later).

Why is it so? Proof by contradiction: Assume, it is not the case – the optimal vector is not contained in the subspace that is spanned by the data points. Project it into the subspace – the subject become better.

Do it with scalar products

Let us do the task a bit more complicated ☺ – consider projections of the direction vector onto all data vectors (instead of the vector itself):

$$\lambda \cdot e = cov \cdot e \qquad \leftrightarrow \qquad \lambda \cdot (x_k^T \cdot e) = x_k^T \cdot cov \cdot e \;\; \forall k$$

The right side follows from the left one directly.

The opposite is less trivial (see the board). It holds only if e can be represented as a linear combination of x_i – here it is just the case (see the previous slide).

Do it with scalar products

All together:

$$cov = \frac{1}{l}\sum_j x_j x_j^T \qquad e = \sum_i \alpha_i x_i \qquad \lambda\cdot(x_k^T\cdot e) = x_k^T\cdot cov\cdot e \quad \forall k$$

$$\lambda\cdot(x_k^T\cdot\sum_i x_i\alpha_i) = x_k^T\cdot\frac{1}{l}\sum_j x_j x_j^T\cdot\sum_i x_i\alpha_i \quad \forall k$$

$$\lambda l\sum_i \alpha_i x_k^T x_i = \sum_i \alpha_i \sum_j x_k^T x_j x_j^T x_i \quad \forall k$$

Let K be the matrix of all pair-vise scalar products, i.e. $K_{ij} = x_i^T x_j$ then (in the matrix form)

$$\lambda l K\alpha = K^2\alpha$$

and finally

$$\lambda l\alpha = K\alpha$$

The PCA can be expressed by scalar products only!!!

Do it with scalar products

$$\lambda l \alpha = K \alpha$$

With an unknown vector $\alpha = (\alpha_1, \alpha_2 \ldots \alpha_l)$
→ basically the same task – find eigenvalues (this time however of the matrix K instead of the cov).

Let the α be given (already found). At the test time new data points are projected onto e , the length of the projection is

$$\langle x, e \rangle = \langle x, \sum_i \alpha_i x_i \rangle = \sum_i \alpha_i \langle x, x_i \rangle$$

→ the quantity of interest can be computed using scalar products, the direction vector is not explicitly necessary.

Using kernels

Now we would like to find a direction vector in the **feature space**.

All the matter remains exactly the same, but with the **Kernel matrix**

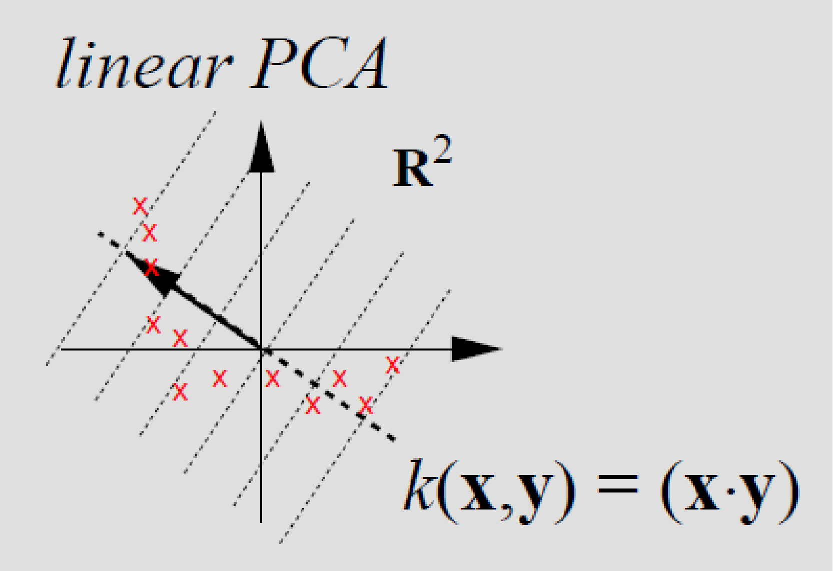
$$K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = \kappa(x_i, x_j)$$

The projection onto the optimal direction is

$$\langle \Phi(x), e \rangle = \langle \Phi(x), \sum_i \alpha_i \Phi(x_i) \rangle = \sum_i \alpha_i \kappa(x, x_i)$$

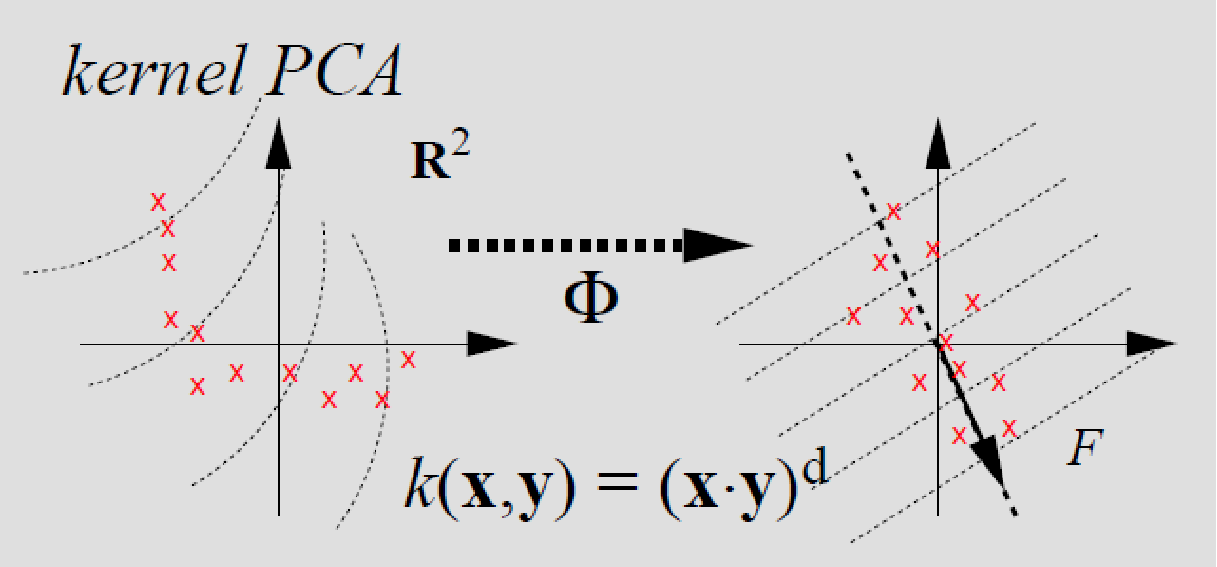
→ **Kernel-PCA**

An illustration



$$\langle \Phi(x), e \rangle = \sum_i \alpha_i \kappa(x, x_i)$$

A linear function in the feature space ↔ a non-linear function in the input space.

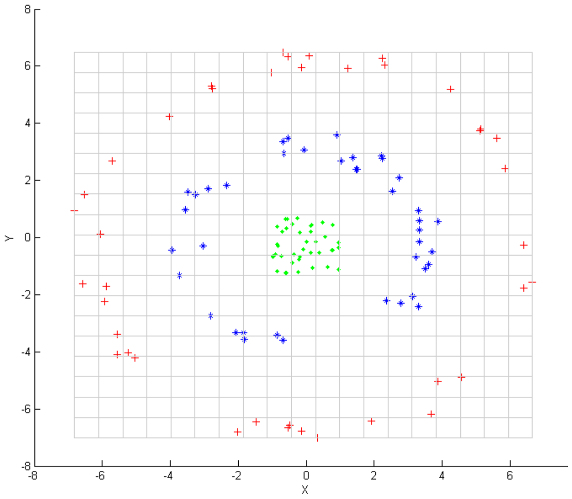


Literature

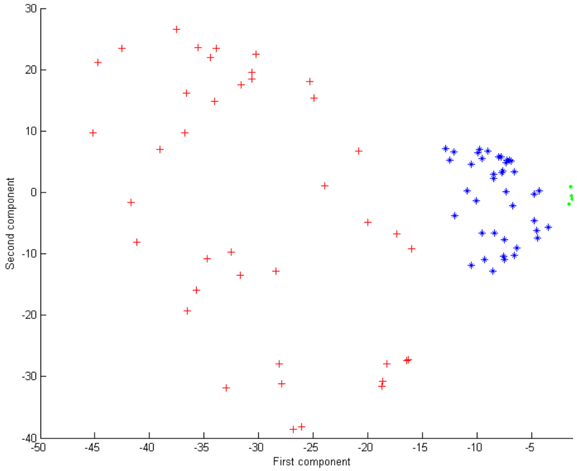
Bernhard Schölkopf, Alexander Smola, Klaus-Robert Müller
Kernel Principal Component Analysis (1997).

In the paper all the matter is presented immediately for the feature spaces and kernels.

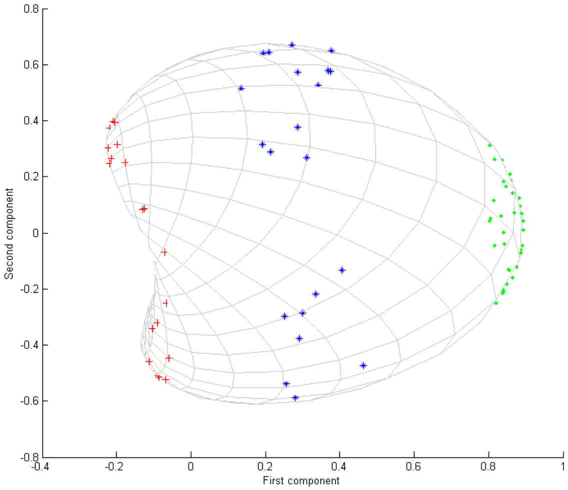
☺



Input space



Polynomial kernel



Gaussian kernel