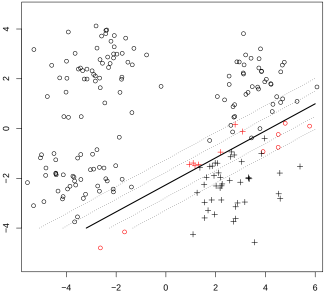
HT2015: SC4
Statistical Data Mining and Machine Learning

Dino Sejdinovic
Department of Statistics
Oxford

http://www.stats.ox.ac.uk/~sejdinov/sdmml.html

Non-linear methods

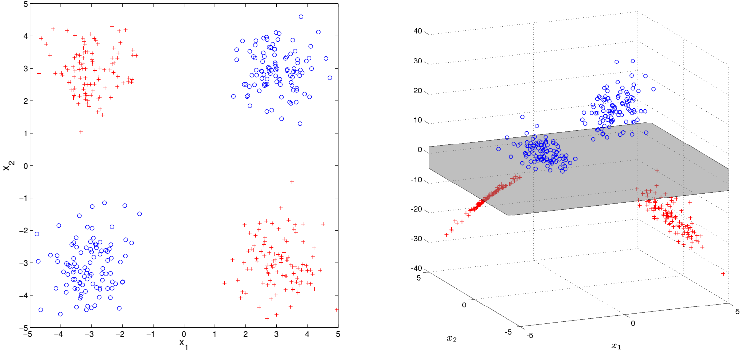
- Linear methods (LDA, logistic regression, naïve Bayes) are simple and effective techniques to learn from data "to first order".
- To capture more intricate information from data, non-linear methods are often needed:
  - Explicit non-linear transformations $x \mapsto \varphi(x)$.
  - Local methods like kNN.
- **Kernel methods**: introduce non-linearities through **implicit** non-linear transforms, often local in nature.



Kernel Methods

slides based on Arthur Gretton's Advanced Topics in Machine Learning course

XOR example



- No linear classifier separates red from blue.
- Linear separation after mapping to a **higher dimensional feature space**:

$$\mathbb{R}^2 \ni \begin{pmatrix} x^{(1)} & x^{(2)} \end{pmatrix}^\top = x \;\mapsto\; \varphi(x) = \begin{pmatrix} x^{(1)} & x^{(2)} & x^{(1)}x^{(2)} \end{pmatrix}^\top \in \mathbb{R}^3$$

Kernel SVM

- Back to the dual C-SVM with explicit non-linear transformation $x \mapsto \varphi(x)$:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \varphi(x_i)^\top \varphi(x_j) \quad \text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \preceq \alpha \preceq C \end{cases}$$

- Suppose $p = 2$, and we would like to introduce quadratic non-linearities,

$$\varphi(x) = \left(1, \sqrt{2}x^{(1)}, \sqrt{2}x^{(2)}, \sqrt{2}x^{(1)}x^{(2)}, \left(x^{(1)}\right)^2, \left(x^{(2)}\right)^2\right)^\top$$

  Then

$$\varphi(x_i)^\top \varphi(x_j) = 1 + 2x_i^{(1)}x_j^{(1)} + 2x_i^{(2)}x_j^{(2)} + 2x_i^{(1)}x_i^{(2)}x_j^{(1)}x_j^{(2)} + \left(x_i^{(1)}\right)^2 \left(x_j^{(1)}\right)^2 + \left(x_i^{(2)}\right)^2 \left(x_j^{(2)}\right)^2 = (1 + x_i^\top x_j)^2$$

- Since only dot-products are needed in the objective function, non-linear transform need not be computed explicitly - inner product between features is often a simple function (**kernel**) of x_i and x_j: $k(x_i, x_j) = \varphi(x_i)^\top \varphi(x_j) = (1 + x_i^\top x_j)^2$
- Generally, m-order interactions can be implemented simply by $k(x_i, x_j) = (1 + x_i^\top x_j)^m$ (**polynomial kernel**).

Kernel SVM: Kernel trick

- Kernel SVM with $k(x_i, x_j)$. Non-linear transformation $x \mapsto \varphi(x)$ still present, but **implicit** (coordinates of the vector $\varphi(x)$ are never computed).

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad \text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \preceq \alpha \preceq C \end{cases}$$

- Prediction? $f(x) = \operatorname{sign}\left(w^\top \varphi(x) + b\right)$, where $w = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)$ and offset b obtained from a margin support vector x_j with $\alpha_j \in (0, C)$.
  - No need to compute w either! Just need

$$w^\top \varphi(x) = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)^\top \varphi(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x).$$

  - Get offset from

$$b = y_j - w^\top \varphi(x_j) = y_j - \sum_{i=1}^{n} \alpha_i y_i k(x_i, x_j)$$

  for any margin support-vector x_j ($\alpha_j \in (0, C)$).
- Fitted a separating hyperplane in a high-dimensional feature space without ever mapping explicitly to that space.

Kernel trick in general

- In a learning algorithm, if only inner products $x_i^\top x_j$ are explicitly used, rather than data items x_i, x_j directly, we can replace them with a kernel function $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$, where $\varphi(x)$ could be **nonlinear, high- and potentially infinite-dimensional** features of the original data.
  - Kernel ridge regression
  - Kernel PCA
  - Kernel K-means
  - Kernel FDA

Gram matrix

- The **Gram matrix** is the matrix of dot-products, $\mathbf{K}_{ij} = \varphi(x_i)^\top \varphi(x_j)$.

$$\mathbf{K} = \begin{pmatrix} - \; \varphi(x_1)^\top \; - \\ \vdots \\ - \; \varphi(x_i)^\top \; - \\ \vdots \\ - \; \varphi(x_n)^\top \; - \end{pmatrix} \cdot \begin{pmatrix} | & & | & & | \\ \varphi(x_1) & \cdots & \varphi(x_j) & \cdots & \varphi(x_n) \\ | & & | & & | \end{pmatrix}$$

- Since $\mathbf{K} = \Phi\Phi^\top$, it is symmetric and positive semidefinite.
- Recall: Gram matrix closely related to the distance matrix (MDS)
- Assuming features are centred, the sample covariance of features is $\Phi^\top\Phi$.
- Many kernel methods, e.g. kernel PCA, make use of the duality between the Gram and the sample covariance matrix.

Kernel: an inner product between feature maps

Definition (kernel)

Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **kernel** if there exists a **Hilbert space** and a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k(x, x') := \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}} .$$

- Almost no conditions on $\mathcal{X}$ (eg, $\mathcal{X}$ itself need not have an inner product, e.g., documents).
- Think of kernel as **similarity measure between features**

What are some simple kernels? E.g., for text documents? For images?

- A single kernel can correspond to multiple sets of underlying features.

$$\varphi_1(x) = x \qquad \text{and} \qquad \varphi_2(x) = \begin{pmatrix} x/\sqrt{2} & x/\sqrt{2} \end{pmatrix}^\top$$

Positive semidefinite functions

If we are given a “measure of similarity” with two arguments, $k(x, x')$, how can we determine if it is a valid kernel?

1. Find a feature map?
    - Sometimes not obvious (especially if the feature vector is infinite dimensional)
2. A simpler direct property of the function: positive semidefiniteness.

Positive semidefinite functions

Definition (Positive semidefinite functions)

A symmetric function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite if $\forall n \geq 1,\ \forall (a_1, \ldots a_n) \in \mathbb{R}^n,\ \forall (x_1, \ldots, x_n) \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \kappa(x_i, x_j) \geq 0.$$

- Kernel $k(x, y) := \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$ for a Hilbert space $\mathcal{H}$ is positive semidefinite.

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \langle a_i \varphi(x_i), a_j \varphi(x_j) \rangle_{\mathcal{H}} \\ &= \left\| \sum_{i=1}^{n} a_i \varphi(x_i) \right\|_{\mathcal{H}}^2 \geq 0. \end{aligned}$$

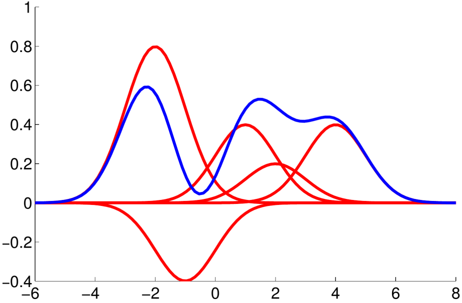
Positive semidefinite functions are kernels

Moore-Aronszajn Theorem

Every positive semidefinite function is a kernel for some Hilbert space $\mathcal{H}$.

- Often, $\mathcal{H}$ is a space of functions (**Reproducing kernel Hilbert space - RKHS**)

Gaussian RBF kernel $k(x, x') = \exp\left(-\frac{1}{2\gamma^2}\left\|x - x'\right\|^2\right)$ has an infinite-dimensional $\mathcal{H}$ with elements $h(x) = \sum_{i=1}^{m} a_i k(x_i, x)$ (recall that $w^\top \varphi(x)$ in SVM has exactly this form!).



Examples of kernels

- **Linear**: $k(x, x') = x^\top x'$.
- **Polynomial**: $k(x, x') = (c + x^\top x')^m$, $c \in \mathbb{R}$, $m \in \mathbb{N}$.
- **Gaussian RBF**: $k(x, x') = \exp\left(-\frac{1}{2\gamma^2}\left\|x - x'\right\|^2\right)$, $\gamma > 0$.
- **Laplacian**: $k(x, x') = \exp\left(-\frac{1}{2\gamma^2}\left\|x - x'\right\|\right)$, $\gamma > 0$.
- **Rational quadratic**: $k(x, x') = \left(1 + \frac{\left\|x - x'\right\|^2}{2\alpha\gamma^2}\right)^{-\alpha}$, $\alpha, \gamma > 0$.
- **Brownian covariance**: $k(x, x') = \frac{1}{2}\left(\|x\|^\gamma + \|x\|^\gamma - \|x - x'\|^\gamma\right)$, $\gamma \in [0, 2]$.

New kernels from old: sums, transformations

The great majority of useful kernels are built from simpler kernels.

Lemma (Sums of kernels are kernels)

Given $\alpha > 0$ and k, k_1 and k_2 all kernels on $\mathcal{X}$, then αk and $k_1 + k_2$ are kernels on $\mathcal{X}$.

To prove this, just check inner product definition. A difference of kernels may not be a kernel (**why?**)

Lemma (Mappings between spaces)

Let $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ be sets, and define a map $s : \mathcal{X} \to \widetilde{\mathcal{X}}$. Define the kernel k on $\widetilde{\mathcal{X}}$. Then the kernel $k(s(x), s(x'))$ is a kernel on $\mathcal{X}$.

Example: $k(x, x') = x^2 \, (x')^2$.

New kernels from old: products

Lemma (Products of kernels are kernels)

Given k_1 on $\mathcal{X}_1$ and k_2 on $\mathcal{X}_2$, then $k_1 \times k_2$ is a kernel on $\mathcal{X}_1 \times \mathcal{X}_2$.

Proof.

Sketch for finite-dimensional spaces only. Assume $\mathcal{H}_1$ corresponding to k_1 is $\mathbb{R}^m$, and $\mathcal{H}_2$ corresponding to k_2 is $\mathbb{R}^n$. Define:

- $k_1 := u^\top v$ for $u, v \in \mathbb{R}^m$ (e.g.: kernel between two images)
- $k_2 := p^\top q$ for $p, q \in \mathbb{R}^n$ (e.g.: kernel between two captions)

Is the following a kernel?

$$K\left[(u, p); (v, q)\right] = k_1 \times k_2$$

(e.g. kernel between one image-caption pair and another)

□

New kernels from old: products

Proof.

(continued)

$$
\begin{aligned}
k_1 k_2 &= \left(u^\top v\right)\left(q^\top p\right) \\
&= \mathrm{trace}\left(u^\top v q^\top p\right) \\
&= \mathrm{trace}\left(p u^\top v q^\top\right) \\
&= \langle A, B \rangle \,,
\end{aligned}
$$

where $A := pu^\top$, $B := qv^\top$ (features of image-caption pairs)
Thus $k_1 k_2$ is a valid kernel, since inner product between $A, B \in \mathbb{R}^{m \times n}$ is

$$\langle A, B \rangle = \mathrm{trace}(AB^\top).$$

□

Kernel Methods – Discussion

- Kernel methods allows for very flexible and powerful machine learning models.
- **Nonparametric** method: parameter space (e.g., of parameter w in SVM) can be infinite-dimensional
- Kernels can be defined over more complex structures than vectors, e.g. graphs, strings, images, probability distributions.
- Computational cost at least quadratic in the number of observations, often $O(n^3)$ computation and $O(n^2)$ memory (various approximations - hot research topic!)
- Further reading:
  - Bishop, Chapter 6.
  - UCL course by Arthur Gretton on Advanced Topics in Machine Learning.
  - Schölkopf and Smola, Learning with Kernels, 2001.
  - Rasmussen and Williams, Gaussian Processes for Machine Learning, 2006.