Kernel-Based Learning Algorithms

- Linear classifiers
- Margins
- Feature space
- Kernels functions
- Support vector machines

- Literature
Linear Classifiers

- Assume: training samples \((\vec{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}, 1 \leq i \leq n\), are separable by hyperplane.

- Linear separator \(f(\vec{x}) = (\vec{w} \cdot \vec{x}) + b\).
  - \(\vec{w} \cdot \vec{x} = \sum_i w_i x_i\).

- Decision boundary \(\{\vec{x} \mid (\vec{w} \cdot \vec{x}) + b = 0\}\).
  - Two classes \(f(\vec{x}) = \text{sign}((\vec{w} \cdot \vec{x}) + b)\), where
    \[
    \text{sign}(z) = \begin{cases} 
      1 & \text{if } z > 0, \\
      -1 & \text{otherwise}.
    \end{cases}
    \]
Margins

- Margin $m$: minimal distance of a sample to the decision surface.

- Finding a linear separator consists of solving the constraint satisfaction problem:

$$
\begin{align*}
\vec{x}_i \cdot \vec{w} + b & \geq m \quad \text{if } y_i = 1, \\
\vec{x}_i \cdot \vec{w} + b & \leq -m \quad \text{if } y_i = -1,
\end{align*}
$$

or, equivalently,

$$
y_i(\vec{x}_i \cdot \vec{w} + b) \geq m
$$

for all training samples $i$.

- Rescale $\vec{w}$ and $b$ such that $m = 1$: canonical representation.

- Margin is related to the size of $\vec{w}$: $m = \frac{2}{\|\vec{w}\|}$, where

$$
\|\vec{w}\| = \sqrt{\vec{w} \cdot \vec{w}}.
$$

- Learning is searching for hyperplanes (determined by $\vec{w}$ and $b$) with maximum margin.

- How limiting is the choice of linear functions as separators?
Feature Spaces

▶ Assume: samples are not separable by hyperplane.

▶ Idea: map the problem into higher dimensional space.
  ▶ Let \( \mathcal{F} \) be a potentially much higher dimensional feature space.
  ▶ Let \( \Phi : \mathcal{X} \rightarrow \mathcal{F}, \vec{x} \mapsto \Phi(\vec{x}) \).

▶ Learning problem now works with samples

\[
(\Phi(\vec{x}_1), y_1), \ldots, (\Phi(\vec{x}_k), y_k) \in \mathcal{F} \times \mathcal{Y}.
\]

▶ Can the mapped problem be classified in a “simple” way?

▶ Statistics: difficulty increases drastically with the dimension of the space.

▶ Statistical learning theory: the contrary can be true (sometimes).
Example

- Let $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{F} = \mathbb{R}^3$.

- $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. 

14 Kernel-Based Learning Algorithms
Kernel Functions

- Training depends only on dot products of the form $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$!

- Here we find:

$$
\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) = x_{i1}^2 x_{j1}^2 + \sqrt{2} x_{i1} x_{i2} \sqrt{2} x_{j1} x_{j2} + x_{i2}^2 x_{j2}^2 \\
= (x_{i1} x_{j1} + x_{i2} x_{j2})^2 \\
= (\vec{x}_i \cdot \vec{x}_j)^2 \\
= k(\vec{x}_i, \vec{x}_j).
$$

- $k$ is called kernel function.

- Surprise: The dot product can be computed without explicitly using or even knowing $\Phi$ if we know $k$!

- Theorem Every algorithm that only uses scalar products can implicitly be executed in $\mathcal{F}$ by using kernels.
Typical Kernel Functions

- Polynomial classifier of degree $p$:
  \[ k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p. \]

- Gaussian radial basis function classifier:
  \[ k(\vec{x}_i, \vec{x}_j) = e^{-\frac{||\vec{x}_i - \vec{x}_j||}{c}}. \]

- Sigmoidal classifier:
  \[ \tanh(\kappa(\vec{x}_i \cdot \vec{x}_j) + \theta). \]
Support Vector Machines (1)

- Margin determined by just a few \( (n) \) examples from \((X, Y)\) called support vectors.

- Recall conditions:
  \[ y_i (\vec{x}_i \cdot \vec{w} + b) \geq 1. \]

- Going into feature space \( \mathcal{F} \):
  \[ y_i (\Phi(\vec{x}_i) \cdot \vec{w} + b) \geq 1. \]
  
  ▶ Observe that now \( \vec{w} \in \mathcal{F}! \)

- Task: Learn \( \vec{w} \) and \( b \) such that \( \frac{||\vec{w}||^2}{2} \) is minimized.
  
  ▶ Background: this maximizes the margin and minimizes the expected risk.

- How to solve minimization problem if we can access the feature space only via dot-products computed by the kernel?
Support Vector Machines (2)

Consider the dual optimization problem: Minimize

\[ \frac{||\vec{w}||^2}{2} - \sum_i \alpha_i (y_i ((\Phi(\vec{x}_i) \cdot \vec{w}) + b) - 1) \] 

wrt \( \vec{w} \) and \( b \) and maximize it wrt \( \alpha_i \), where \( \alpha_i \geq 0 \).

Optimal points wrt \( b \) and \( \vec{w} \) yield:

\[ \sum_i \alpha_i y_i = 0 \quad \text{and} \quad \vec{w} = \sum_i \alpha_i y_i \Phi(\vec{x}_i). \]

Substituting these equations into (*) yields

\[ \max \alpha \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)). \]

Using kernel function \( k(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) \) we obtain

\[ \max \alpha \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\vec{x}_i, \vec{x}_j). \]
Support Vector Machines (3)

- From the above maximization problem we obtain values for $\alpha_i$, $1 \leq i \leq n$.

- Remember, classes were defined by

$$f(\bar{x}) = \text{sign}((\bar{w} \cdot \bar{x}) + b).$$

- Going into feature space yields:

$$f(\bar{x}) = \text{sign}((\bar{w} \cdot \Phi(\bar{x})) + b).$$

- Using $\bar{w} = \sum_i \alpha_i y_i \Phi(\bar{x}_i)$ yields

$$f(\bar{x}) = \text{sign}(\sum_i \alpha_i y_i (\Phi(\bar{x}_i) \cdot \Phi(\bar{x})) + b).$$

- Using kernel function $k(\bar{x}_i, \bar{x}) = \Phi(\bar{x}_i) \cdot \Phi(\bar{x})$ we obtain

$$f(\bar{x}) = \text{sign}(\sum_i \alpha_i y_i k(\bar{x}_i, \bar{x}) + b).$$
Summary

▶ Approach:
  ▶ Project examples into high dimensional space.
  ▶ Learn linear separators with maximum margin.
  ▶ Learning as minimizing the expected risk.

▶ Advantages:
  ▶ Good empirical result on various classification tasks (character recognition, text classification, segmentation etc.).
  ▶ PAC-style theoretical grounding.
  ▶ Global optimization method, no local minima.