Introduction to Support Vector Machines

Starting from slides drawn by Ming-Hsuan Yang and Antoine Cornuéjols
SVM Bibliography

http://citeseer.nj.nec.com/burges98tutorial.html


SVM — The Main Idea

**Given** a set of data points which belong to either of two classes, find an **optimal separating hyperplane**

- maximizing the distance (from closest points) of either class to the separating hyperplane, and

- minimizing the risk of misclassifying the training samples and the *unseen* test samples.

**Approach:** Formulate a constraint-based optimisation problem, then solve it using **quadratic programming (QP)**.
O schemă generală pentru învățarea automată

training data -> machine learning algorithm -> data model

predicted classification

test/generalization data
Plan

1. Linear SVMs
   The primal form and the dual form of linear SVMs
   Linear SVMs with soft margin
2. Non-Linear SVMs
   Kernel functions for SVMs
   An example of non-linear SVM
1. Linear SVMs: Formalisation

Let $S$ be a set of points $x_i \in \mathbb{R}^d$ with $i = 1, \ldots, m$. Each point $x_i$ belongs to either of two classes, with label $y_i \in \{-1, +1\}$. The set $S$ is linear separable if there are $w \in \mathbb{R}^d$ and $w_0 \in \mathbb{R}$ such that

$$y_i(w \cdot x_i + w_0) \geq 1 \quad i = 1, \ldots, m$$

The pair $(w, w_0)$ defines the hyperplane of equation $w \cdot x + w_0 = 0$, named the separating hyperplane.

The signed distance $d_i$ of a point $x_i$ to the separating hyperplane $(w, w_0)$ is given by $d_i = \frac{w \cdot x_i + w_0}{\|w\|}$.

It follows that $y_id_i \geq \frac{1}{\|w\|}$, therefore $\frac{1}{\|w\|}$ is the lower bound on the distance between points $x_i$ and the separating hyperplane $(w, w_0)$. 
Optimal Separating Hyperplane

Given a linearly separable set $S$, the optimal separating hyperplane is the separating hyperplane for which the distance to the closest (either positive or negative) points in $S$ is maximum, therefore it maximizes $\frac{1}{||w||}$. 
\[ D(x) = w \cdot x + w_0 \]
Linear SVMs: The Primal Form

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}||w||^2 \\
\text{subject to} & \quad y_i(w \cdot x_i + w_0) \geq 1 \quad \text{for} \quad i = 1, \ldots, m
\end{align*}
\]

This is a constrained quadratic problem (QP) with \(d + 1\) parameters \((w \in \mathbb{R}^d \text{ and } w_0 \in \mathbb{R})\). It can be solved by quadratic optimisation methods if \(d\) is not very big \((10^3)\).

For large values of \(d\) \((10^5)\): due to the Kuhn-Tucker theorem, because the above objective function and the associated constraints are convex, we can use the method of Lagrange multipliers \((\alpha_i \geq 0, i = 1, \ldots, m)\) to put the above problem under an equivalent “dual” form.

Note: In the dual form, the variables \((\alpha_i)\) will be subject to much simpler constraints than the variables \((w, w_0)\) in the primal form.
2. Nonlinear Support Vector Machines

- Note that the only way the data points appear in (the dual form of) the training problem is in the form of dot products $x_i \cdot x_j$.
- In a higher dimensional space, it is very likely that a linear separator can be constructed.
- We map the data points from the input space $\mathbb{R}^d$ into some space of higher dimension $\mathbb{R}^n$ ($n > d$) using a function $\Phi : \mathbb{R}^d \to \mathbb{R}^n$.
- Then the training algorithm would depend only on dot products of the form $\Phi(x_i) \cdot \Phi(x_j)$.
- Constructing (via $\Phi$) a separating hyperplane with maximum margin in the higher-dimensional space yields a **nonlinear decision boundary** in the input space.
General Schema for Nonlinear SVMs
Introducing Kernel Functions

• But the dot product is computationally expensive...

• If there were a “kernel function” $K$ such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, we would only use $K$ in the training algorithm.

• All the previous derivations in the model of linear SVM hold (substituting the dot product with the kernel function), since we are still doing a linear separation, but in a different space.

• Important remark: By the use of the kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the higher space.
Some Classes of Kernel Functions for SVMs

- **Polynomial:** \( K(x, x') = (x \cdot x' + c)^q \)

- **RBF (radial basis function):** \( K(x, x') = e^{-\frac{||x-x'||^2}{2\sigma^2}} \)

- **Sigmoid:** \( K(x, x') = \tanh(\alpha x \cdot x' - b) \)
Concluding Remarks: SVM — Pros and Cons

Pros:

- Find the **optimal** separation hyperplane.
- Can deal with very high dimensional data.
- Some kernels have infinite Vapnik-Chervonenkis dimension (see Computational learning theory, ch. 7 in Tom Mitchell’s book), which means that they can learn very elaborate concepts.
- Usually work very well.

Cons:

- Require both positive and negative examples.
- Need to select a good kernel function.
- Require lots of memory and CPU time.
- There are some numerical stability problems in solving the constrained QP.