Learning with Kernels

Proseminar Data Mining

Matthieu Bulté
Fakultät für Mathematik
Technische Universität München
Email: matthieu.bulte@tum.de

***Abstract*— Text der Kurzfassung**

***Index Terms*— support vector machines, machine learning, statistics**

I. Introduction

todo

II. First steps: hard margin classifier

The hard margin classifier learns the parameters $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ of a hyperplane $\mathscr{H} \subset \mathbb{R}^n$, separating two classes of observations $\mathbf{x}$ of dimension n. To train it, the algorithm takes a set of m observations $\mathbf{x_i}$ together with a vector of labels $y_i = \pm 1$ for each of the observations:

$$(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \dots, (\mathbf{x_m}, y_m)$$

The learned parameters are then used to define a decision function f, assigning to points of $\mathbb{R}^n$ a label ± 1 corresponding to the side of the hyperplane on which the points stands:

$$f(\mathbf{x}) = sgn(\langle \mathbf{w}, \mathbf{x} \rangle + b) \tag{1}$$

A. Margin maximization

One particularity about the learning algorithm used for Support Vector Machines is the loss function the algorithm tries to optimize. Other learning algorithms typically chose a loss function based on the empirical risk, defined as following for a function f and a training set $(\mathbf{x}, \mathbf{y})$ of size m:

$$R_{emp}[f] = \frac{1}{m} \sum_{i=0}^{m} \frac{1}{2} |f(\mathbf{x_i}) - y_i| \tag{2}$$

The margin maximization algorithm instead, searches within the space of hyperplanes properly classifying the training set, for the hyperplane with the largest distance from the training points to the hyperplane. This can be translated in the following constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}}{\text{maximize}} \quad & M := \frac{1}{\|\mathbf{w}\|} \underset{i}{min}\; y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq M \end{aligned} \tag{3}$$

Assuming that a solution to this optimization problem exisits, it still has the issue that although the solution hyperplane is unique, there is an inifity of parameter vectors $\|\mathbf{w}\|$ resulting in the solution hyperplane, only differing in their length. Uniqueness of $\mathbf{w}$ can be ensured by adding to its length the follwoing contraint:

$$M\|\mathbf{w}\| = 1$$

We can thus reformulate the optimization problem into the following quadratic optimization problem which is known to have a numerical solution:

$$\begin{aligned} \underset{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}}{\text{minimize}} \quad & \|\mathbf{w}\| \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 \end{aligned} \tag{4}$$

B. Support vectors

Although the previous formulation of the problem helped us to better understand the ideas of support vector machines, the resulting quadratic problem is in practice too hard to solve for a very large m. We will, in this section, introduce another formulation of the optimization problem that will not only make it computationally more practical, but will also give us more insights about the resulting hyperplane.

III. Non linearly separable

A. Noisy training data

B. Non linear target boundary

IV. Kernel methods

A. The kernel trick

B. Some useful kernels

V. Zusammenfassung und Ausblick

blabla