**CS 596: Theoretical Machine Learning**

Lecturer: Pranjal Awasthi
Scribe: Jingru Yi

Lecture #9
Oct. 19, 2016

# 1 Linear Model

In linear models, we assume the output $y$ bears a linear algebraic relation with the input $x \in \Re^d$. Specifically, we have $H = \{\text{sgn}(w \cdot x),\ w \in \Re^d\}$. For a target coefficient vector $w^* \in \Re^d$, the goal is to learn $w^*$ given a training data set $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$, where $x_i \in R^d$ is sampled from distribution $D$ and $y_i = \text{sgn}(w^* \cdot x_i) \in \{-1, 1\}$. From VC theory we know the following

**Theorem 1.** *If we can find hypothesis $w \in \Re^d$ such that the training error $err_S(w) = 0$, then $\forall \delta > 0$, with probability $\geq 1 - \delta$, the true error satisfies*

$$err(w) \leq \frac{VC\text{-}dim(H) log(\frac{1}{\delta})}{m},$$

Note that, the smaller $d$ is, the smaller $\frac{dlog(\frac{1}{\delta})}{m}$ tends to be, implying that it is easier to achieve lower true error in low-dimensional space. Next, we give the relationship between the VC dimension and the number of unknown parameters.

**Lemma 1.** *Let $H$ be the function class homogeneous*[1] *linear separators in $\Re^d$. Then, VCdim(H)=d.*

We can set up the learning problem of finding a zero error function as a linear programming (LP) optimization problem. Specifically, for each training sample $(x_i, y_i)$, we want to find $w$ such that:

$$\begin{cases} (w \cdot x_i) > 0 & \text{if } y_1 = +1 \\ (w \cdot x_i) < 0 & \text{if } y_1 = -1 \end{cases} \quad \equiv \quad y_i(w \cdot x_i) > 0.$$

Thus, the LP problem reduces to finding $w$ such that $y_i(w \cdot x_i) > 0, \forall\ i$. However, LP usually behaves like a black box, and we do not have much control over it. This motivates us to explore more effective solution.

## 1.1 Better bounds for better data

Given a collection of data belonging to two different classes, we can find many hyperplanes separating them. Intuitively, the hyperplane with the largest margin should generalize better than others. In this section, we will discuss the maximum margin classifier, in particular, *Support Vector Machines* (SVMs). First, we give the definition of margin.

**Definition 1.** Margin is the minimum distance of data points from the separating hyperplane, i.e.,

$$\text{Margin}(w, S) = \min_{x \in S} \frac{w \cdot x}{||w||}.$$

[1] Here we assume that the linear separator passes through the origin. If not, one can add an extra dimension that is always constant and do learning in that space.

We first look at an online algorithm to learn the coefficient vector $w$, along with a theorem which bounds the number of mistakes made by this algorithm.

**Algorithm 1** Perceptron

**Input**: training set $S$
**Initialize**: $w_0 = \vec{0}$
**for** $t = 0, 1, 2, \ldots, T$ **do**
    **if** mistake at time $t$ on data $(x_t, y_t)$ **then**
        $w_{t+1} \leftarrow w_t + \frac{x_t \cdot y_t}{||x_t||}$
    **end if**
**end for**
**Output**: $w_{T+1}$

**Theorem 2.** *If* $\exists\ w^* \in R^d$ *with* $\|w^*\| = 1$ *such that* $\text{Margin}(w^*, S) \geq \gamma$*, then the number of mistakes made by perceptron* $\leq \frac{R^2}{\gamma^2}$*, where* $R = \max_{i \in S} ||x_i||$.

*Proof.* At time $t$, suppose perceptron makes an mistake on $(x_t, y_t)$, then

$$w_{t+1} \leftarrow w_t + \frac{x_t \cdot y_t}{||x_t||} \tag{1}$$

$$\begin{aligned} \Rightarrow w_{t+1} \cdot w^* &= w_t \cdot w^* + \frac{(x_t \cdot w^*) y_t}{||x_t||} \\ &\geq w_t \cdot w^* + \frac{\gamma}{||x_t||} \text{ [margin of every example is } \geq \gamma] \\ &\geq w_t \cdot w^* + \frac{\gamma}{R} \text{ [using the fact that } \|x_t\| \leq R]. \end{aligned} \tag{2}$$

Squaring both sides of (1), we have:

$$\begin{aligned} ||w_{t+1}||^2 &= ||w_t||^2 + 1 + \frac{2(w_t \cdot x_t) y_t}{||x_t||} \\ &\leq ||w_t||^2 + 1. \text{ [because } w_t \text{ made a mistake on } x_t, (w_t \cdot x_t) y_t < 0 \text{ ]} \end{aligned} \tag{3}$$

Thus, after $T$ mistakes, we have:

$$T \geq ||w_{T+1}||^2 \geq (w_{T+1} w^*)^2 \geq T^2 \frac{\gamma^2}{R^2},$$

where the first inequality comes from equation (3), and the third inequality comes from equation (2). Finally, we have:

$$T \leq \frac{R^2}{\gamma^2}.$$

□

### 1.2 Support Vector Machine

One can construct cases when the data is separable but Perceptron takes exponential steps to converge. Can we retain the Perceptron guarantee and always run in polynomial time? This question leads us to support vector machines (SVM), as introduced below.

An SVM is a discriminative classifier defined by a separating hyperplane. It combines LP and perceptron to find hyperplane with large margin while subject to all given constraints. Formally, assume $||x_i|| \leq 1$, then the problem of finding $w$ can be described as follows:

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & y_i(w \cdot x_i) \geq \gamma, \ \forall i \in S \\ & ||w|| = 1. \end{aligned}$$

We can further transform this problem to make the optimization easy to carry out. Define $v = w/\gamma$, then we have:

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & y_i(v \cdot x_i) \geq 1, \ \forall i \in S \\ & \gamma||v|| = 1. \end{aligned}$$

This is equivalent to

$$\begin{aligned} \text{maximize} \quad & \frac{1}{||v||} \\ \text{subject to} \quad & y_i(v \cdot x_i) \geq 1, \ \forall i \in S. \end{aligned}$$

Rewriting we get,

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||v||^2 \\ \text{subject to} \quad & y_i(v \cdot x_i) \geq 1, \ \forall i \in S. \end{aligned}$$

The above optimization problem is convex, and the optimized hyperplane forms the SVM. For this problem, we can use *Lagrangian duality* to find its optimal value, and we call this problem as the *primal* problem. Here the basic idea is to augment the objective function with a weighted sum of the constraint functions. Specifically, we define the *Lagrangian* as:

$$L(v, \alpha) = \frac{1}{2}||v||^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(v \cdot x_i)),$$

where $\alpha_i$ is the *Lagrange multipliers* associated with the $i$th inequality constraint. Then according to *weak duality*, we have:

$$\max_{\alpha \succeq 0} \min_{v} L(v, \alpha) \leq \min_{v} \max_{\alpha \succeq 0} L(v, \alpha), \tag{4}$$

where $\max_{\alpha \succeq 0} \min_v L(v, \alpha)$ is called the *dual* problem. Let $p^*$ be the optimal value of primal problem, and $(v^*, \alpha^*)$ be the optimal variables of dual problem. When *strong duality* holds, we have $L(v^*, \alpha^*) = p^*$. Compared with the primal problem, this dual problem is unconstrained, and therefore easy to optimize.

Now fixed $\alpha$ and minimize $L(v, \alpha)$. To do this, we can set the derivative of $L$ with

respect to $v$ to zero:

$$\begin{aligned}
\min_v L(v,\alpha) &= \min_v \frac{1}{2}||v||^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i(v \cdot x_i)) \\
\bigtriangledown_v L(v,\alpha) &= v - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \\
\Rightarrow v &= \sum_{i=1}^{m} \alpha_i y_i x_i. \qquad (5)
\end{aligned}$$

Now we are left with optimizing $\alpha$. We then substitute the expression of $v$ into equation (5):

$$\begin{aligned}
&\max_{\alpha \succeq 0} \frac{1}{2}||\sum_{i=1}^{m} \alpha_i y_i x_i||^2 + \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i y_i (\sum_{j=1}^{m} \alpha_j y_j x_j) \cdot x_i \\
\Rightarrow \quad &\max_{\alpha \succeq 0} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j (x_i \cdot x_j) y_i y_j.
\end{aligned}$$

The output of SVM are $\alpha_1^*, \alpha_2^*, \cdots \alpha_n^*$, and the optimal hyperplane $v^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i$. Besides, the penalty will be 0, i.e., $\forall\ i,\ \alpha_i^*(1 - y_i(v^* \cdot x_i)) = 0$. So, for every i, either $\alpha_i^* = 0$ or $y_i(v^* \cdot x_i) = 1$. If $\alpha_i \neq 0$, then $y_i(v^* \cdot x_i) = 1$. Such points are called *support vectors* since the optimal hyperplane is supported on them and they are the closest points to it. In the next lecture we will see how to argue about generalization ability of SVMs.