

## LECTURE 10: PERCEPTRON AND LINEAR SVM

We start with the perceptron algorithm which is probably one of the simplest linear classifiers. We then introduce the margin maximization idea and derive the linear SVM classifier.

Assume that  $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$  where  $\mathcal{X} = \mathbb{R}^p$  and  $\mathcal{Y} = \{\pm 1\}$ . For simplicity we only consider the linear classifiers without intercept here, i.e.  $\mathcal{H} = \{x \mapsto w^T x : w \in \mathbb{R}^p\}$ . Furthermore, we assume that the data are linearly separable, i.e. there exists some  $w_*$  which can correctly classify all examples  $\text{sign}(w_*^T x_i) = y_i$  for  $i = 1, \dots, n$ . The perceptron algorithm works as follows.

1. Start  $w_0 = 0$  and  $t = 0$ ;
2. While  $w_t$  has training error  $> 0$ :
  - (a) Pick one observation  $(x_i, y_i)$  which is misclassified by  $w_t$ ;
  - (b) Update  $w_{t+1} = w_t + y_i x_i$ ;
  - (c)  $t = t + 1$
3. Let  $T = t$ ,  $w = w_T$  and return  $\hat{h}_n(x) = \text{sign}(w^T x)$ .

**Theorem 10-1 (Novikov).** Define  $r = \max_i \|x_i\|$  and  $\delta = \min_i \frac{w_*^T x_i y_i}{\|w_*\|}$ , where  $w_*$  is some classifier which can linearly separate  $\mathcal{D}_n$ . Then it terminates after  $T \leq r^2/\delta^2$  steps.

PROOF.

First, note that  $\delta$  has the meaning of the “margin”: the minimum distance of an example to the decision hyperplane. So the larger the margin, the smaller number of steps we need to converge. The basic idea of the proof is to show that  $w_t$  are getting closer and closer to  $w_*$ . Since  $\|w_t - w_*\|^2 = \|w_t\|^2 + \|w_*\|^2 - 2w_t^T w_*$ , essentially we need to upperbound  $\|w_t\|^2$  and lowerbound  $w_t^T w_*$  and then combine the results.

First we have  $w_0^T w_* = 0$  and

$$\begin{aligned} w_{t+1}^T w_* &= w_t^T w_* + y_i x_i^T w_* \\ &\geq w_t^T w_* + \delta \|w_*\|. \end{aligned}$$

Clearly we have  $w_t^T w_* \geq t\delta \|w_*\|$  by induction. Second, since  $\|w_0\| = 0$  and

$$\begin{aligned} \|w_{t+1}\|^2 &= \|w_t + y_i x_i\|^2 \\ &= \|w_t\|^2 + \|x_i\|^2 + 2y_i x_i^T w_t \\ &\leq \|w_t\|^2 + r^2. \end{aligned}$$

So we have  $\|w_t\|^2 \leq tr^2$ . Thus we have

$$t\delta \|w_*\| \leq w_t^T w_* \leq \|w_t\| \|w_*\| \leq \sqrt{tr} \|w_*\|.$$

Then it follows that  $t \leq r^2/\delta^2$  for any  $t$ , and thus  $T \leq r^2/\delta^2$ .

□

## Maximum Margin Classifier: Support Vector Machines (SVM)

We consider the set of linear classifiers  $\mathcal{H} = \{h(x) = w^T x + b, w \in \mathbb{R}^p, b \in \mathbb{R}\}$ . Suppose the training examples are linearly separable, i.e. there exist some linear classifier which has 0 training error. Consider the following optimization problem:

$$\begin{aligned} \min_{w,b} \quad & \|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \quad \forall i = 1, \dots, n. \end{aligned}$$

This is a constrained optimization where the objective is a quadratic function and the constraints are linear. So it is a convex optimization problem (quadratic programming, to be more specific).

Given a hyperplane (classifier), define *margin* as the minimum distance between the plane to any of the example. Now we show that the above optimization essentially tries to find a classifier which maximizes the margin. First, assume that there are two examples  $x_+$  and  $x_-$ , both are on the margin boundary (see Figure 1). Then we know that the margin equals half of the distance between  $(x_+ - x_-)$ 's projection along the direction that is perpendicular to the hyperplane. So we have

$$\text{margin} = \frac{1}{2}(x_+ - x_-)^T \frac{w}{\|w\|}$$

Using the fact that  $x_+$  and  $x_-$  lie on margin, we have  $w^T x_+ + b = 1$  and  $w^T x_- + b = -1$ . Thus  $w^T(x_+ - x_-) = 2$ . So we conclude that the margin is  $1/\|w\|$ . Thus minimizing  $\|w\|^2$  subject to the linear constraints is equivalent to maximizing  $1/\|w\|^2$  subject to the same constraints.

Since in practice examples may not be linearly separable, we introduce the concept of slack variables. For each example, define  $\xi_i \geq 0$  to be the slack variable which measures how much this example violates the margin condition. Instead of minimizing  $\|w\|^2$ , we will also add a term  $\sum_{i=1}^n \xi_i$  which penalizes violation of the margin condition. The relaxed optimization problem can be written as:

$$\begin{aligned} \min_{w,b} \quad & \sum_{i=1}^n \xi_i + \lambda \|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n \\ & \xi_i \geq 0 \quad \forall i = 1, \dots, n. \end{aligned}$$

where  $\lambda > 0$  is a tuning parameter which controls the balance between training error and the margin. Note that equivalently, we can write down the optimization problem as

$$\min_{w,b} \sum_{i=1}^n (1 - y_i(w^T x_i + b))_+ + \lambda \|w\|^2$$

where  $(t)_+ = \max(t, 0)$ .