

ECE 695 Lecture Notes

Prof. Shai Ben-David

December 3, 2003

1 Review

The basic idea of SVMs is the following: embed the data in a high dimensional Euclidean space and search for separating half-spaces in it. The hope is that by embedding in a high dimensional space we can make the training data linearly separable.

Problems: Half Spaces in \mathfrak{R}^n where n is large have high VC-dimension and therefore the generalization guarantees for the resulting classifier will not be as good. In addition, searching for separating hyperplanes in a high dimensional space is a difficult task.

Solution: Can solve the generalization guarantee problem by looking for alternatives to VC-dimension to bound $|E^s(h) - E^p(h)|$

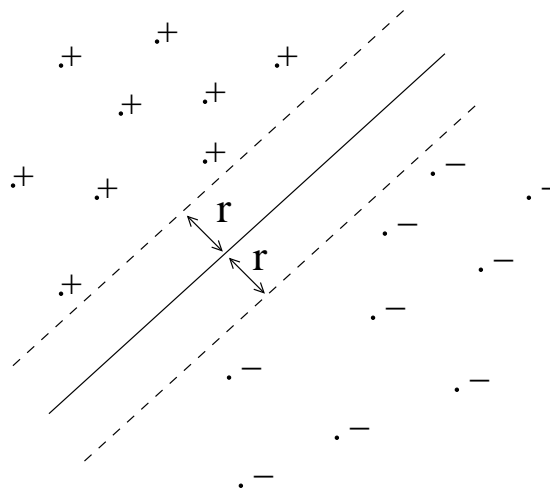


Figure 1: Margins

Best Candidate: If we know that the positive and negative samples can be separated by a hyperplane such that every point is at a distance greater than r from the hyperplane,

then r is called the margin. If there is such a margin r then we can get good error bounds in terms of it even if the hyperplane is in a high dimensional space. In order to show how margins can replace VC-dimension to provide good generalization guarantees we shall look at a different model of learning called online learning.

2 Online Learning

The major points that define this model are:

- The examples come one at a time.
- Upon seeing a new datapoint x_i the learner has to predict its label y_i .
- After making the prediction the learner is shown the correct label of x_i .
- x_{i+1} is introduced.
- The measure of success is the number of mistakes that were made (“mistake bound” model.)
- We assume that all the points are labelled by some function $h : X \mapsto \{+1, -1\}$ that comes from some family H which is known to the learner (we need to have a fixed set of labels and a family of functions because otherwise the learner can be forced to be always wrong.)

Example:

$$X = \{1, \dots, n\}$$

$$H = \{h_k : 0 \leq k \leq N\}$$

$$h_k(i) = \begin{cases} 1 & i \leq k \\ 0 & i > k \end{cases}$$

Consider a simple learner’s strategy: use minimal consistent hypothesis,

$$L(x_i) = \begin{cases} 1 & \text{if some } x_j, j < i \text{ was labelled 1 and } x_j \geq x_i \\ 0 & \text{otherwise} \end{cases}$$

Can we bound the number of mistakes for this learner? If we have N labels and we set $x_i = i$ then the learner makes N mistakes.

Improved strategy: binary search

$$L(x_i) = \begin{cases} 1 & \text{if } x_i \leq \frac{\max_{y_j=1} x_j + \min_{y_j=0} x_j}{2} \\ 0 & \text{otherwise} \end{cases}$$

Now, whenever binary search errs the interval of uncertainty (between rightmost 1 and

leftmost 0) decreases by at least a factor of 2. Regardless of which h is used for labelling and in which order the input is presented, binary search makes $\leq \log N$ mistakes.

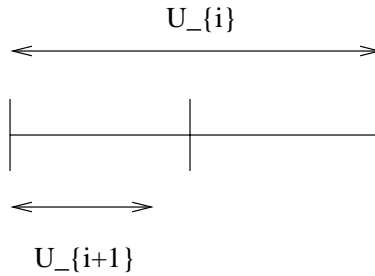


Figure 2: $|U_{i+1}| \leq |U_i|/2$

Claim 1 *This bound is the best possible for the H of initial segments.*

Theorem 1 *If H has VC-dimension $\geq d$ then any learner can be forced to make $\geq d$ mistakes.*

Proof: Choose $A \subseteq X$, $|A| = d$ and H shatters A (i.e. all labellings on A are allowed.) As an adversary just choose the label opposite to what the learner predicts. Formally, Let $x_1 \dots x_d$ be points of A in any order. Given any learner L , let $h \in H$ be such that

$$\forall i (L((x_1, h(x_1)), \dots, (x_i, h(x_i)), x_{i+1}) \neq h(x_{i+1}))$$

□

We will use this observation to show that given any $B \subseteq \mathfrak{R}^n$ if H is a collection of half-spaces s.t. $\forall h \in H, h$ has margin $\geq r$ over B , then VC-dimension of $H \leq \left(\frac{2R}{r}\right)^2$ where R is upper bound on $\|X\|$ for $X \in B$. Note that n does not affect the VC-dimension. We will prove this VC-dimension result by showing an online algorithm for H that is guaranteed to make fewer than $\leq \left(\frac{2R}{r}\right)^2$ mistakes.

2.1 The Perceptron Algorithm (Rosenblat, 1953)

The perceptron algorithm is an algorithm for online learning of separating half-spaces. Let $X = \mathfrak{R}^n$ and $H =$ half-spaces over \mathfrak{R}^n . Every $h \in H$ can be represented by a vector of weights $\vec{w}_h \in \mathfrak{R}^n$ and a bias $b_h \in \mathfrak{R}$, where

$$h(x) = \text{sign}(\vec{x} \cdot \vec{w}_h + b_h)$$

We initialize by setting $\vec{w}_0 = \vec{0}$ and $b_0 = 0$. We are given a stream of input points $((x_1, y_1), (x_2, y_2), \dots)$. Upon seeing (x_i, y_i) we update the weights as follows:

$$\vec{w}_{i+1} = \begin{cases} \vec{w}_i & \text{sign}(\vec{x}_i \cdot \vec{w}_i + b_i) = y_i \\ \vec{w}_i + y_i \vec{x}_i & \text{otherwise} \end{cases}$$

$$b_{i+1} = \begin{cases} b_i & \text{sign}(\vec{x}_i \cdot \vec{w}_i + b_i) = y_i \\ b_i + y_i R^2 & \text{otherwise} \end{cases}$$

Theorem 2 (Novikoff) *If $\exists \vec{w}^*, b^*$ s.t. $\|\vec{w}^*\| = 1$ and $\forall i y_i(\vec{w}^* \cdot \vec{x}_i + b^*) \geq r$ (for some fixed $r > 0$), then the perceptron algorithm makes at most $\left(\frac{2R}{r}\right)^2$ mistakes.*

Proof: For simplicity, let us assume that $R = 1$. We extend the \vec{x}_i 's by one more coordinate and set these to be 1. Also \vec{w}^ includes an $(n + 1)^{th}$ coordinate b_i .*

*Claim 1: Whenever the algorithm errs, \vec{w}_{i+1} gets closer to \vec{w}^**

$$\begin{aligned}\vec{w}_{i+1} \cdot \vec{w}^* &= (\vec{w}_i + y_i \vec{x}_i) \cdot \vec{w}^* \\ &= \vec{w}_i \cdot \vec{w}^* + y_i \vec{x}_i \cdot \vec{w}^* \\ &\geq \vec{w}_i \cdot \vec{w}^* + r\end{aligned}$$

Claim 2:

$$\begin{aligned}\|\vec{w}_{i+1}\|^2 &= \vec{w}_{i+1} \cdot \vec{w}_{i+1} \\ &= \|\vec{w}_i\|^2 + 2y_i \vec{x}_i \cdot \vec{w}_i + \|\vec{x}_i\|^2 \\ &\leq \|\vec{w}_i\|^2 + 2\end{aligned}$$

After making k mistakes $\vec{w}_{i+1} \cdot \vec{w}^ \geq kr$ and $\|\vec{w}_{i+1}\| \leq \sqrt{2k}$. On the other hand since $\|\vec{w}^*\| = 1$,*

$$\begin{aligned}\vec{w}_{i+1} \cdot \vec{w}^* &\leq \|\vec{w}_{i+1}\| \\ kr &\leq \sqrt{2k} \\ k &\leq \left(\frac{2}{r}\right)^2\end{aligned}$$

□