Lecture 5, November 22, 2016

Machine Learning Theory

Linear classifiers

Ruth Urner

We have seen in the last lectures that, for binary hypothesis classes, learnability is equivalent to having bounded VC-dimension. Moreover, we have seen that, for classes of bounded VC-dimension, *any* ERM learning algorithm is a successful learner (that is, learns the class according to Definition 1, Lecture 2).

Note that, "ERM" for is not an learning algorithm. Rather it is a paradigm. Namely, it asks to output a classifier of minimal empirical error. Different learning methods can be ERM for some fixed hypothesis class. In this lecture, we will see a first concrete learning algorithm for the class of linear predictors: The perceptron algorithm is a very basic learner that, as we will see, successfully learns the class of linear predictors in the realizable case.

This chapter is based on the (first part of) Chapter 9 in the textbook [1].

1 Linear Classifiers

We consider the domain $\mathcal{X} = \mathbb{R}$ and label set $\mathcal{Y} = \{-1, 1\}$. The class $\mathcal{H}_{\text{g-lin}}$ of general linear classifiers (or the class of linear halfspaces) is defined as follows:

$$\mathcal{H}_{\text{g-lin}} = \{h_{\boldsymbol{w},b} \ : \ \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\},$$

where

$$h_{\boldsymbol{w},b}(\boldsymbol{x}) = \text{sign}\left(\langle \boldsymbol{x}, \boldsymbol{w}\rangle + b\right) = \text{sign}\left(\left(\sum_{i=1}^{d} x_i w_i\right) + b\right)$$

General linear halfspaces in $\mathbb{R}^d$ can be viewed as *homogeneous* linear halfspaces in $\mathbb{R}^{d+1}$ via a simple transformation. Recall that the class $\mathcal{H}_{\text{lin}}$ of homogeneous linear halfspaces is defined as

$$\mathcal{H}_{\text{lin}} = \{h_{\boldsymbol{w}} \ : \ w \in \mathbb{R}^d\},$$

where

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \text{sign}\left(\langle \boldsymbol{x}, \boldsymbol{w}\rangle\right) = \text{sign}\left(\sum_{i=1}^{d} x_i w_i\right).$$

Let $\boldsymbol{x} = h_{\boldsymbol{w},b}(\boldsymbol{x}) \in \mathbb{R}^d, \boldsymbol{w} = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ and $b \in \mathbb{R}$ be given. We define

$$\boldsymbol{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$$

and

$$\boldsymbol{w}' = (1, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}.$$

Then we get $\langle \boldsymbol{x}, \boldsymbol{w} \rangle + b = \langle \boldsymbol{x}', \boldsymbol{w}' \rangle$ for all $\boldsymbol{x} \in \mathbb{R}$, and thus

$$h_{\boldsymbol{w},b}(\boldsymbol{x}) = h_{\boldsymbol{w}'}(\boldsymbol{x}').$$

We will thus now focus our analysis of learning linear classifiers to learning homogeneous linear classifiers. This is not a restriction of generality. For the problem of learning general linear classifiers we can employ the following simple reduction:

- Given a dataset $S = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ of data points $\boldsymbol{x}_i$ in $\mathbb{R}^d$, transform them to data set $S' = (\boldsymbol{x}'_1, \boldsymbol{x}'_2, \ldots, \boldsymbol{x}'_m)$ of points in $\mathbb{R}^{d+1}$.
- Learn a homogeneous linear classifier $\boldsymbol{w}' \in \mathbb{R}^{d+1}$ from S'.
- Interpret the first component w'_1 in $\boldsymbol{w}'$ as the term b and the remaining entries $(\boldsymbol{w}'_2, \ldots, \boldsymbol{w}'_d)$ as the vector $\boldsymbol{w}$ defining the halfspace classifier $h_{\boldsymbol{w},b}$ in $\mathbb{R}^d$.

2 The VC-dimension of halfspaces

Theorem 1. *The VC-dimension of homogeneous linear halfspaces in $\mathbb{R}^d$ is d.*

Proof. We first show that homogeneous linear halfspaces can shatter d points. Consider the set of vectors $\boldsymbol{e}_1, \ldots \boldsymbol{e}_d$, where $\boldsymbol{e}_i$ is defined as the vector with entry 1 in the i-th component and entries 0 else. Let $U \subseteq [d]$ be a subset of indices and let $y_1^U, \ldots, y_d^U$ be the induced labeling, that is

$$y_i^U = \begin{cases} 1 \text{ if } i \in U \\ -1 \text{ else} \end{cases}$$

We now define $\boldsymbol{w}$ by

$$w_i = \begin{cases} 1 \text{ if } i \in U \\ -1 \text{ else} \end{cases}$$

Then we get

$$h_{\boldsymbol{w}}(\boldsymbol{e}_i) = y_i$$

for all $i \in [d]$. Thus, the d unit vectors are shattered.

Next, we prove that homogeneous linear halfspaces can not shatter more than d points in $\mathbb{R}^d$. Let some points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1} \in \mathbb{R}^d$ be given and assume by way of contradiction, that this set of $d+1$ points is shattered. Since they are also linearly dependent, there exists a set of scalars $a_1, \ldots a_{d+1}$ such that $\sum_{i=1}^{d+1} a_i \boldsymbol{x}_i = 0$.

Let $I = \{i : a_i > 0\}$ and $J = \{i : a_i > 0\}$. At least one of these sets of indices is non-empty. We assume for now that both I and J are not empty. Then we get

$$\sum_{i \in I} a_i \boldsymbol{x}_i = \sum_{j \in J} |a_j| \boldsymbol{x}_j$$

Since the $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1}$ are shattered, there exists a vector $\boldsymbol{w}$ with

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \ > \ 0 \text{ for all } i \in I$$

and

$$\langle \boldsymbol{w}, \boldsymbol{x}_j \rangle \ < \ 0 \text{ for all } j \in J$$

This yields

$$0 < \sum_{i \in I} a_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle = \left\langle \boldsymbol{w}, \sum_{i \in I} a_i \boldsymbol{x}_i \right\rangle = \left\langle \boldsymbol{w}, \sum_{j \in J} |a_j| \boldsymbol{x}_j \right\rangle = \sum_{j \in J} |a_j| \langle \boldsymbol{w}, \boldsymbol{x}_j \rangle < 0,$$

a contradiction. Finally note that if either $I = \emptyset$ or $J = \emptyset$, then one of the inequalities above is an equality, but the contradiction still follows. □

Similarly, one can show that the VC-dimension of general linear classifiers in $\mathbb{R}^d$ is $d + 1$.

Theorem 2. *The VC-dimension of general linear classifiers in $\mathbb{R}^d$ is $d + 1$.*

Proof. Left as excecise. □

The theory of learning classes of bounded VC-dimension (see Lectures 2 and 3) now tells us that there is a constant $C > 0$, such that general halfspaces in $\mathbb{R}^d$ are learnable from samples of size larger than

$$m(\epsilon, \delta) = C \cdot \frac{(d + 1) + \ln(1/\delta)}{\epsilon^2},$$

in the agnostic case and samples of sizes at least

$$m(\epsilon, \delta) = C \cdot \frac{(d + 1) + \ln(1/\delta)}{\epsilon},$$

in the realizable case.

3 The Perceptron Algorithm

The perceptron algorithm is a simple learner that implements an ERM rule for $\mathcal{H}_{\text{lin}}$ in the realizable case.

Note that a trainingsexample $(\boldsymbol{x}_i, y_i)$ from S is misclassified by a (homogeneous) halfspace $h_{\boldsymbol{w}}$ if and only if

$$y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle < 0.$$

In the realizable case, a learner is an ERM learner if it always outputs a classifier that makes no errors on the traning data. Thus an ERM for homogeneous halfspaces needs to output a vector $\boldsymbol{w}$ such that

$$y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0$$

for all $(\boldsymbol{x}_i, y_i) \in S$. The perceptron algorithm is defined as follows:

We will next show that, on realizable data, the perceptron will always find a classifier that makes no error (that is, the above algorithm will always halt). The following theorem furthermore gives a bound on the number of iterations the algorithm makes before finding such a classifier. It thus provides a bound on the computation time that the Perceptron algorithm requires.

Algorithm 1 Perceptron

Input $S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$
Initialize: $\boldsymbol{w}^1 = (0, 0, \ldots, 0)$
for $t = 1, 2, \ldots$ **do**
  if there exists an i such that $y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \leq 0$ **then**
    $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + y_i \boldsymbol{x}_i$
  else
    return $\boldsymbol{w}^t$
  end if
end for

Theorem 3. *Assume that the data generating distribution P is realizable by the class of homogeneous linear classifiers in $\mathbb{R}^d$. Let $S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$ be a sample generated by P. Let*

$$B = \min\{\|\boldsymbol{w}\| \ : \ y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq 1 \text{ for all } i \in [m]\}$$

and let

$$R = \max_{i \in [m]} \|\boldsymbol{x}_i\|.$$

Then the Perceptron algorithm stopts after $t \leq (RB)^2$ iterations.

Note that, by the definition of the algorithm, we have

$$y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0$$

for all $(\boldsymbol{x}_i, y_i) \in S$, after $t \leq (RB)^2$ iterations. That is, at that point the algorithm found a classifier that minimizes the empirical risk on the sample S.

Proof. The proof can be found as the proof of Theorem 9.1, Chapter 9, of the textbook [1]. □

In the tutorials, it was shown that the bound $(RB)^2$ in the above theorem can be replaced with the quantity $\frac{R^2}{\gamma^2}$, where γ is the *margin* of the data. That is, γ is the distance of the closest point to the hyperplane defininig the halfspace classifier, for the classifier that maximizes that distance. The margin, by which a dataset is separable, can be viewed as a parameter of niceness or easiness of the data. Theorem 3 quantifies how the runtime of the perceptron algorithm depends on this easyness parameter. The "easier" the dataset, that is, the larger the margin by which it is separable with a homogeneous linear halfspace, the faster the algorithm will converge to finding such a seaparating halfspace.

References

[1] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning.* Cambridge University Press, 2014.