

Linear Predictors

Peng Lingwei

July 30, 2019

Contents

9 Linear Predictors	1
9.1 HALFSPACES	1
9.1.1 Linear Programming for the Class of Halfspaces	2
9.1.2 Perception for Halfspaces	2
9.1.3 The VC Dimension of Halfspaces	3
9.2 LINEAR REGRESSION	3
9.2.1 Least Squares	4
9.2.2 Linear Regression for Polynomial Regression Tasks	4
9.3 LOGISTIC REGRESSION	4

9 Linear Predictors

This chapter is focused on learning linear predictors using the ERM approach; however, in later chapters we will see alternative paradigms for learning these hypothesis classes.

The class of affine functions :

$$L_d = \{h_{\vec{w},b} = \langle \vec{w}, \vec{x} \rangle + b : \vec{w} \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

Rewrite into homogeneous linear function. Let $\vec{w}' = (b, w_1, \dots, w_d) \in \mathbb{R}^{d+1}$, $\vec{x}' = (1, x_1, \dots, x_d) \in \mathbb{R}^{d+1}$. Therefore,

$$h_{\vec{w},b}(\vec{x}) = \langle \vec{w}, \vec{x} \rangle + b = \langle \vec{w}', \vec{x}' \rangle.$$

9.1 HALFSPACES

The class of *Halfspaces* is :

$$HS_d = \text{sign} \circ L_d = \{\vec{x} \mapsto \text{sign}(h_{\vec{w},b}(\vec{x})) : h_{\vec{w},b} \in L_d\}.$$

The $VCdim(HS_d) = d + 1$, and the sample size is $\Omega\left(\frac{d+\log(1/\delta)}{\epsilon}\right)$.

In the context of halfspaces, the realizable case is often referred to as the "separable" case.

Implementing the ERM rule in the nonseparable case is known to be computationally hard. (Ben-David and Simon, 2001).

The most popular approach of learning nonseparable data is use *surrogate loss functions*(ch12), namely, to learn a halfspace that does not necessarily minimize the empirical risk with the 0-1 loss, but rather with respect to a different loss function.

9.1.1 Linear Programming for the Class of Halfspaces

Linear programs :

$$\max_{\vec{w} \in \mathbb{R}^d} \langle \vec{u}, \vec{w} \rangle, \quad \text{s.t. } A\vec{w} \geq \vec{v}.$$

Change the ERM problem for halfspaces in the realizable case can be expressed as a LP:

$$\max_{\vec{w} \in \mathbb{R}^d} \langle \vec{u}, \vec{w} \rangle, \quad \text{s.t. } \vec{u} = \vec{0}, \quad A\vec{w} \geq \vec{v}, \{A_{i,j}\} = y_i x_{i,j}, \quad \vec{v} = (1, \dots, 1) \in \mathcal{R}^m \quad (9.1)$$

9.1.2 Perception for Halfspaces

$$y_i \langle \vec{w}^{(t+1)}, x_i \rangle = y_i \langle \vec{w}^{(t)} + y_i \vec{x}_i, x_i \rangle = y_i \langle \vec{w}^{(t)}, x_i \rangle + \|\vec{x}_i\|^2.$$

Because $\|\vec{x}_i\| \geq 0$, so the Perception guides the solution to be "more correct" on i'th example. "More correct" doesn't mean make i'th example exactly correct.

Algorithm 1 Batch Perception

Require: A training set $(\vec{x}_1, y_1), \dots, (\vec{x}_m, y_m)$

Ensure: $\vec{w}^{(1)} = (0, \dots, 0)$

```

for t=1,2,... do
  if ( $\exists i$  s.t.  $y_i \langle \vec{w}^{(t)}, \vec{x}_i \rangle \leq 0$ ) then
     $\vec{w}^{(t+1)} = \vec{w}^{(t)} + y_i \vec{x}_i$ 
  else
    return  $\vec{w}^{(t)}$ 
  end if
end for

```

Theorem 9.1. Assume that $(\vec{x}_1, y_1), \dots, (\vec{x}_m, y_m)$ is saperable, let $B = \min\{\|\vec{w}\| : \forall i \in [m], y_i \langle \vec{w}, \vec{x}_i \rangle \geq 1\}$, and let $R = \max_i \|\vec{x}_i\|$. Then, the Perception algorithm stops after at most $(RB)^2$ iterations.

Proof. let $\vec{w}^* = \arg \min_{\vec{w}} \{\|\vec{w}\| : \forall i \in [m], y_i \langle \vec{w}, \vec{x}_i \rangle \geq 1\}$. Our mean goal is to proof:

$$\frac{\sqrt{T}}{RB} \leq \frac{\langle \vec{w}^*, \vec{w}^{(T+1)} \rangle}{\|\vec{w}^*\| \|\vec{w}^{(T+1)}\|} \leq 1 \quad (9.2)$$

$$\vec{w}^{(1)} = (0, \dots, 0) \Rightarrow \langle \vec{w}^*, \vec{w}^{(1)} \rangle = 0.$$

$$\langle \vec{w}^*, \vec{w}^{(t+1)} \rangle - \langle \vec{w}^*, \vec{w}^{(t)} \rangle = \langle \vec{w}^*, y_i \vec{x}_i \rangle \geq 1 \Rightarrow \langle \vec{w}^*, \vec{w}^{(T+1)} \rangle \geq T \quad (9.3)$$

$$\|\vec{w}^{(t+1)}\|^2 = \|\vec{w}^{(t)} + y_i \vec{x}_i\|^2 \leq \|\vec{w}^{(t)}\|^2 + R^2 \quad (9.4)$$

$$\|\vec{w}^{(T+1)}\|^2 \leq TR^2 \quad (9.5)$$

□

9.1.3 The VC Dimension of Halfspaces

Theorem 9.2. *The VC dimension of the class of homogenous halfspaces in \mathbb{R}^{d+1} is $d+1$.*

Proof. First, consider the set of vectors $\vec{e}_1, \dots, \vec{e}_{d+1} \in \mathbb{R}_{d+1}$, then, $\forall \{y_1, \dots, y_{d+1}\}$, set $\vec{w} = (y_1, \dots, y_{d+1})$, we get $\forall i, \langle \vec{w}, \vec{e}_i \rangle = y_i$. So $VCdim(HS_d) \geq d+1$.

Second, suppose that $\exists X = (\vec{x}_1, \dots, \vec{x}_{d+2})$ are shattered by HS_d . We can get none zero vector $\vec{a} = (a_1, \dots, a_{d+2})$ s.t. $\vec{a}^T X = \vec{0}$. Let $I = \{i : a_i > 0\}$ and $J = \{j : a_j < 0\}$, then $\sum_{i \in I} a_i \vec{x}_i = -\sum_{j \in J} a_j \vec{x}_j$.

Because X is shattered by HS_d , so $\exists \vec{w}$ such that $\forall i \in I, \langle \vec{w}, \vec{x}_i \rangle > 0$ and $\forall j \in J, \langle \vec{w}, \vec{x}_j \rangle < 0$. It follows that

$$0 < \sum_{i \in I} a_i \langle \vec{x}_i, \vec{w} \rangle = -\sum_{j \in J} a_j \langle \vec{x}_j, \vec{w} \rangle < 0.$$

which leads to a contradiction. \square

Theorem 9.3. *The VC dimension of the class of nonhomogeneous halfspaces in \mathbb{R}^d is $d+1$.*

Proof. First, the set of vectors $\vec{0}, \vec{e}_1, \dots, \vec{e}_d$ is shattered by the class of nonhomogeneous halfspaces.

Second, if $\exists \vec{x}_1, \dots, \vec{x}_{d+2}$ are shattered by the class of nonhomogeneous halfspaces, it will contradict former theorem. \square

9.2 LINEAR REGRESSION

The hypothesis class of linear regression predictors is simply the set of linear function

$$\mathcal{H}_{reg} = L_d = \{\vec{x} \mapsto \langle \vec{w}, \vec{x} \rangle + b : \vec{w} \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

Squared-loss function

$$l_{sq}(h, (\vec{x}, y)) = (h(\vec{x}) - y)^2.$$

Mean Squared Error

$$L_S(h) = \frac{1}{m} \sum_{i=1}^m (h(\vec{x}_i) - y_i)^2.$$

Absolute value loss function

$$l(h, (\vec{x}, y)) = |h(\vec{x}) - y|.$$

Note that since linear regression is not a binary prediction task, we cannot analyse its sample complexity using the VC-dimension. One possible analysis of the sample complexity of linear regression is by relying on the "discretization trick" (namely, use 64 bits floating point representation to represent \vec{w}, b .) But we also need that the loss function will be bounded.

The rigorous means to analyze the sample complexity of regression problems is coming later.

9.2.1 Least Squares

Let $\mathbf{A} = \mathbf{X}\mathbf{X}^T$ and $\vec{b} = \mathbf{X}\vec{y}$.

If \mathbf{A} is invertible then the solution to the ERM algorithm is

$$\vec{w} = \mathbf{A}^{-1}\vec{b}.$$

Otherwise,

$$\hat{\vec{w}} = \mathbf{A}^+\vec{b}.$$

9.2.2 Linear Regression for Polynomial Regression Tasks

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n.$$

$$\mathcal{H}_{poly}^n = \{x \mapsto p(x)\}.$$

Let $\Psi(x) = (1, x, x^2, \dots, x^n)$

$$p(x) = \langle \vec{a}, \Psi(x) \rangle.$$

9.3 LOGISTIC REGRESSION

logistic function :

$$\Phi_{sig}(z) = \frac{1}{1 + \exp(-z)} \quad (9.6)$$

Sigmoid hypothesis class

$$\mathcal{H}_{sig} = \phi_{sig} \circ L_d = \{\vec{x} \mapsto \phi_{sig}(\langle \vec{w}, \vec{x} \rangle) : \vec{w} \in \mathbb{R}^d\}.$$

Sigmoid loss function:

$$l_{sig}(h_{\vec{w}}, (\vec{x}, y)) = \log(1 + \exp(-y\langle \vec{w}, \vec{x} \rangle)).$$

The ERM problem associated with logistic regression is

$$\arg \min_{\vec{w} \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-y_i \langle \vec{w}, \vec{x}_i \rangle)) \quad (9.7)$$

which is identical to the problem of finding a *Maximum Likelihood Estimator*.