9 Linear Predictors

Here, we will introdduce several different hypothesis classes of the linear family: halfspaces, linear regression and logistic regression predictors.

Affine function is a function between affine spaces that perserves points, straight lines and planes. Parralel lines remain parallel after affine transformations in affine spaces. First, define a class of affine functions:

$$L_d = \{h_{\mathbf{w},b} : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$$

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

Also, to shorten notation it is possible to incorporate the constant b and and dummy weight to have more compact notation:

$$h_{\mathbf{w}}(x) = \langle \mathbf{w}, \mathbf{x} \rangle$$

where $w \in \mathbb{R}^{d+1}$.

9.1 Halfspaces

First, we consider halfspaces in the context of binary classification. We compose the hypothesis class of a sign function and the linear function

$$HS_d = sign \circ L_d = \{\mathbf{x} -> sign(h_{\mathbf{w}}(x))\}$$

Geometrically, in 2 dimensions, this can be percieved as a line with values above the directed line (have a sharp angle with the gradient $\mathbf{w}$) being positive, all other negative. VCdim is $d + 1$ which will be showed afterwards, which means that halfspaces can be learned as long as the sample size is $\frac{d+\log(1/\delta)}{\epsilon}$ There are (at least) two ways to finding an ERM halfspace in the realizable case of halfspaces. In this context, it is called the “linearly separable” case.

- this is fantastic separable and realizable

In the non-separable case, the problem is much harder to solve, and usually a surrogate function is introduced instead of the 0-1 loss.

9.1.1 Linear Programming for the Class of Halfspaces

Linear programs (LP) are problems that can be expressed as maximizing a linear functions subject to linear inequalities:

$$\max_{\mathbf{w} \in R^d} \langle \mathbf{u},\ \mathbf{w} \rangle$$

subject to $A\mathbf{w} \geq \mathbf{v}$. In words, maximize weights, but satisfy the condition with the matrix. Linear programs can be solved in polynomial time. So, we just need to express halfspaces as a linear program problem. We wish that for every i, $sign(\langle \mathbf{w}, \mathbf{x} \rangle) = y_i$. Which is equivalent to say that we are looking for

$$y_i \langle \mathbf{w}, \mathbf{x} \rangle > 0, \forall i = 1, ..., m$$

Let **w*** be a vector satisfying that condition. Define $\gamma = min_i(y_i \langle \mathbf{w*}, \mathbf{x} \rangle)$ and let $\mathbf{w'} = \frac{\mathbf{w*}}{\gamma}$ Therefore, for all i, we have

$$y_i \langle \mathbf{w'}, x_i \rangle = \frac{1}{\gamma} y_i \langle \mathbf{w*}, \mathbf{x}_i \rangle \geq 1$$

Now, we try to set this equation in the form $A\mathbf{w} \geq v$ and use some LP solver to solve it. We don't actually do any maximization, so we can set our objective u as all zero.

9.1.2 Perceptron for Halfspaces

An alternative to linear programming for implementing the ERM rule for halfspaces is to use the Perceptron algorithm.

- fun fact: Rosenblatt was a psychologist: he did rat experiments where he would train rats to do something, transplant their brain into other ones–> this failed utterly. His perceptron was invented and applied on machines in 1957 (at this age of 29), died at age 43 in an accident

The idea of the perceptron algorithm is to adjust weights for examples which have been missed and leave them as is for examples which are correct. The missed ones adjust the weights by $y_i \mathbf{x}_i$

** Theorem 9.1. ** Assume that $(x_1, y_1)...(x_m, y_m)$ is separable. Let $B = \min\{||\mathbf{w} : \forall i \in [m], y_i \langle \mathbf{w}, \mathbf{x} \rangle \geq 1\ \}$ and let $R = \max_i ||\mathbf{x}||$

Then, the perceptron algorithm stops after $(RB)^2$ iterations.

Let $w*$ be the separating set of weights, with the minimal norm B. Now, we need to prove that for T iterations, the cosine between the current (t) weights and $w*$ is at least $\frac{\sqrt{T}}{RB}$, mathematically said:

$$\frac{\langle w*, w^{T+1}\rangle}{||w*||||w^{T+1}||} \geq \frac{\sqrt{T}}{RB}$$

Now, we look at all the iterations where there is a correction $y_i\langle w, x_i\rangle \leq 0$. $\langle w*, w^{T+1}\rangle \geq T$, by looking through T iterations of separating angle. Second, we upper bound the norm $||w^{T+1}|| \leq TR^2$.

- TODO how to do this recursively for T iterations?
- how can B be exponentially large?

9.1.3 The VC dimension of Halfspaces

Theorem 9.2 The VC dimension of halfspaces in R^d is d.

Proof Consider a set of vectors $\mathbf{e}_1, ..., \mathbf{e}_d$, one hot encoded at location i. This set is shattered by homogenous halfspaces, since the inner product of every possible labelling is y_i. Now, we go above d by taking $d+1$ for dimension of $\mathbf{x}$. There must exist some non zero numbers that yield $\sum_{i=1}^{d+1} a_i x_i = 0$. Now take sets $I = i : a_i > 0$ and $J = j : a_j < 0$. One of them is nonempty.

- TODO don't get this part: come back to it

9.2 LINEAR REGRESSION

Linear regression is a statistical tool for modelling relationships between explanatory variables and a real-valued outcome. We wish to learn a function $h : \mathbb{R}^d \to \mathbb{R}$ mapping from the input space $\mathcal{X}$ to the real-value y.

The hyp. class of linear regression predictors is the class of linear functions. A typical loss function is squared-loss $l(h, (\mathbf{x}, y)) = (h(\mathbf{x}) - y)^2$ The ERM is called Mean Squared loss since it averages on the number of samples

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

VC-dimension analysis of linear regression is not possible due to VC calculation restriction to binary functions. One could rely on the "discretization trick" which makes the class become finite (at most 2^{64(d + 1)})

9.2.1 Least Squares

The last squares algorithm is the ERM for linear regression predictors. The ERM tries to find the minimum weights such that the mean squared error is the smallest:

$$arg \min_w L_S(h_w) = argmin_w \frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

We calculate the gradient with respect to w:

$$\frac{2}{m}(\langle w, x_i \rangle - y_i)x_i = 0$$

We rewrite the problem as $Aw = b$ where $A = \sum_{i=1}^{m} x_i x_i^T$ and $B = \sum_{i=1}^{m} y_i x_i$

If A is invertible then the solution is $w = A^{-1} b. In case A is not invertible

- TODO b is in the range of A? needs proof

9.2.2 Linear Regression for Polynomial Regression Tasks

Polynomial hypothesis classes is parametrized by degree and a vector of coefficients:

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

ERM is also Least Squares Algorithm, just need to define a mapping $\theta : \mathbb{R} \to \mathbb{R}^{n+1}$.

I Googled this, and there seems to be 2 ways. One needs to make partial derivates on $a_0, a_1, ..., a_n$. Then you have an equation system with each coefficient and you end up with a Vandermonde matrix.

9.3 LOGISTIC REGRESSION

The hypothesis class associated with logistic regression is a composition between a sigmoid and the linear class. A sigmoid is defined as :

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

It represents probability.