Chapters 2-4

Chapter 2: A gentle start

Terminology:

- Domain set (instance space): $\mathcal{X}$, all the objects (instances) we may wish to label. Represented as a vector of features.
- Label set: $\mathcal{Y}$, set of all possible labels, generated by some unknown *true* labelling function f
- Training data $(x, y) \in D, \quad |D| = m$

Empirical risk minimization

The *training error* is defined over the training set (sample) S

$$L_S(h) \stackrel{def}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

Overfitting: a trivial classifier that achieves zero training error simply copies labels from examples in the training set, and outputs a random label if the input vector is not from the training set.

A **Hypothesis class** is a set of predictors (family of models) $\mathcal{H}$. Each $h \in \mathcal{H}$ is a function mapping $\mathcal{X}$ to $\mathcal{Y}$.

Formally, the predictor chosen by the ERM rule is the one that minimizes the training error.

$$ERM_{\mathcal{H}}(S) \in \arg\min_{h \in \mathcal{H}} L_S(h)$$

The choice to restrict the hypothesis space is called an *inductive bias*. The choice of a family of predictors should be based on some prior knowledge about the problem. Ideally, we would want guarantees that the chosen family will not overfit.

DEFINITION 2.1 (The realizability assumption): There exists $h^* \in \mathcal{H}$ s.t. $L_{(\mathcal{D},f)}(h^*) = 0$

Note: this definition holds for any S – S can be any random sample from the true data distribution D labelled by f.

IID assumption All samples are *independently* and *identically* distributed according to $\mathcal{D}$: $S \sim \mathcal{D}^m$.

Two new parameters are introduced, the *confidence* of the distribution sample and *accuracy* of the classifier.

- **Confidence**: parameter δ denotes the probability of getting a **nonrepresentative** sample of the true distribution. Thus, we call $(1-\delta)$ the *confidence parameter*
- **Accuracy**: the accuracy parameter ϵ determines what we consider as failure of the classifier. If $L_{(\mathcal{D},f)}(h_s) > \epsilon$ we consider this a failure of the learner, while $L_{(\mathcal{D},f)}(h_s) \leq \epsilon$ we consider the algorithm an *approximately correct* predictor.

With these parameters, we can formally express the *probability* to sample an $m-$tuple of instances that will lead to failure of teh learner. We define the *bad* hypotheses as the subset of the hypothesis space which has error greater than ϵ on the *true* distribution.

$$\mathcal{H}_b = \{h \in \mathcal{H} : L_{(D,f)}(h) > \epsilon\}$$

Also, we define the *miseleading* samples as the data samples which allow a hypothesis from the set of *bad* hypotheses to minimize the training error (obtain error of 0).

$$M = \{S_x : \exists h \in \mathcal{H}_B s.t. L_S(h) = 0\} \tag{1}$$

We want to upper bound the number of training samples which produce *bad* hypotheses (which obtain error lager then ϵ) as the minimizers of the *training* error.

$$\mathcal{D}^m(\{S_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\})$$

We can rewrite (1) as

$$M = \bigcup_{h \in \mathcal{H}_B} \{S_x : L_s(h) = 0\}$$

the number of failures is bounded by the number of misleading samples

$$\mathcal{D}^m(\{S_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \leq \mathcal{D}^m(M) = \mathcal{D}^m(\cup_{h \in \mathcal{H}_B}\{S_x : L_s(h) = 0\}) \tag{2}$$

Union bound: for two sets A, B and a distribution $\mathcal{D}$, the union bound states that the size of the union of two sets is at most the sum of the sizes of both set (holds with equality when both sets are disjoint):

$$\mathcal{D}(A \cup B) \leq \mathcal{D}(A) + \mathcal{D}(B)$$

By union bounding the RHS of (2), we obtain

$$\mathcal{D}^m(\{S_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S_x : L_s(h) = 0\})$$

COROLLARY 2.3 $\mathcal{H}$ is a finite hypothesis class. Let $\delta \in (0,1)$, $\epsilon > 0$ and m is an integer satisfying

$$m \geq \frac{log(|\mathcal{H}|/\delta)}{\epsilon}$$

Then, for **any** labeling function f and **any** distribution $\mathcal{D}$ for which the realizability assumption holds, with probability of *at least* $1-\delta$ over the choice of an i.i.d. sample S of size m, for **every** ERM hypothesis h_S, the following holds:

$$L_{(D,f)}(h_S) \leq \epsilon$$

For a sufficiently large m, the ERM_H rule over a finite hypothessi class will be *probably* $(1-\delta)$ *approximately* (up to error ϵ) correct (PAC).

Chapter 3: A formal learning model

PAC Learning

TODO: PAC learnability definition

Sample complexity is the *minimal** number of examples required to guarantee a PAC solution. $m_H : (0,1)^2 \to \mathbb{N}$.

COROLLARY 3.2 Every finite hypothesis class is PAC learnable with sample complexity

$$m_H(\epsilon, \delta) \leq \left\lceil \frac{log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

Relaxations: Two relaxations with respect to our original problem are necessary for real-world problems:

- Removing the realizability assumption (*agnostic* PAC)
- Learning problems beyond binary classification

Agnostic PAC Learning: practically, it is not realistic that there will exist a classifier which perfectly approximates the labeling function over the whole data distribution. In place of the absolute labelling function f, we introduce

the conditional probability $D((x,y)\|x)$ indicating the probability of an input sample x to have the class label y.

Revising the empirical and true error

$$L_D(h) \stackrel{def}{=} \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \stackrel{def}{=} \mathcal{D}(\{(x,y) : h(x) \neq y\})$$

The definition of empirical risk remains the same as before.

TODO: Bayes optimal predictor formula

The Bayes optimal predictor assigns class label 1 to samples taht have probability of having label 1 larger than 0.5, and 0 otherwise.

TODO: Agnostic PAC learnability definiton

Agnostic PAC defines the *relative* distance from the best classifier in the chosen hypothesis class rather then the absolute minimal error.

Generalized loss functions

Given any set $\mathcal{H}$ and a domain set $\mathcal{Z}$, l is any function mapping from HxZ to nonnegative real numbers.