Foundations for Machine Learning

L. Y. Stefanus

TU Dresden, June-July 2019

Reference

- Shai Shalev-Shwartz and Shai Ben-David. UNDERSTANDING MACHINE LEARNING: From Theory to Algorithms. Cambridge University Press, 2014.

A Formal Model of Learning

1. The learner's input: In the basic statistical learning setting, the learner has access to the following:
   - Domain set: An arbitrary set, X. This is the set of objects that we may wish to recognize or label. For example, in the papaya learning problem, the domain set will be the set of all papayas, represented by a vector of features (like the papaya's color and softness).

- Label set: For our current discussion, we will restrict the label set Y to be a two-element set, usually {0, 1}. For our papayas example, let 1 represents being tasty and 0 stands for being not-tasty.
- Training data: $S = ((x_1, y_1), \ldots, (x_m, y_m))$ is a finite sequence of pairs in $X \times Y$: that is, a sequence of labeled domain points. This is the input that the learner has access to (like a set of papayas that have been tasted, represented by their color, softness, and tastiness).

2. The learner's output: The learner is requested to output a prediction rule, $h : X \rightarrow Y$. This function is also called a predictor, a hypothesis, or a classifier. The predictor can be used to predict the label of new domain points. In our papayas example, it is a rule that our learner will employ to predict whether future papayas in the market are going to be tasty or not. We use the notation $A(S)$ to denote the hypothesis that a learning algorithm, A, returns upon receiving the training sequence S.

3. A data-generation model: How is the training data generated? First, we assume that the instances (the papayas) are generated by some probability distribution (in this case, representing the environment). Let us denote that probability distribution over X by D. It is important to note that we do not assume that the learner knows anything about this distribution. This could be any arbitrary probability distribution. As to the labels, in the current discussion we assume that there is some "correct" labeling function, $f : X \rightarrow Y$, and that $y_i = f(x_i)$ for all i. This assumption will be relaxed later. The labeling function is unknown to the learner. In fact, this is just what the learner is trying to figure out. In summary, each pair in the training data S is generated by first sampling a point x_i according to D and then labeling it by f.

4. Measures of success: The error of h is the probability to draw a random instance x, according to the distribution D, such that h(x) does not equal f(x). Given a domain subset, $A \subset X$, the probability distribution, D, assigns a number, D(A), which determines how likely it is to observe a point $x \in A$. We refer to A as an event and express it using a function $\pi : X \to \{0,1\}$, namely, $A = \{x \in X : \pi(x) = 1\}$. In that case, we also use the notation $P_{x\sim D}[\pi(x)]$ to express D(A). Now, we can define the error of a prediction rule, $h : X \to Y$, to be

$$L_{\mathcal{D},f}(h) \stackrel{\text{def}}{=} \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}(\{x : h(x) \neq f(x)\}).$$

$L_{D,f}(h)$ has several synonymous names such as the generalization error, the risk, or the true error of h.

- Notice that the learner is blind to the underlying distribution D over the world and to the labeling function f. The only way the learner can interact with the environment is through observing the training set.

Empirical Risk Minimization

- A learning algorithm receives as input a training set S, sampled from an unknown distribution D and labeled by some target function f, and should output a predictor $h_S : X \rightarrow Y$ (the subscript S emphasizes the fact that the output predictor depends on S). The goal of the algorithm is to find h_S that minimizes the error with respect to the unknown D and f.

Empirical Risk Minimization

- Since the learner does not know what D and f are, the true error is not directly available to the learner. A useful notion of error that can be calculated by the learner is the training error (the error the classifier incurs over the training sample):

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

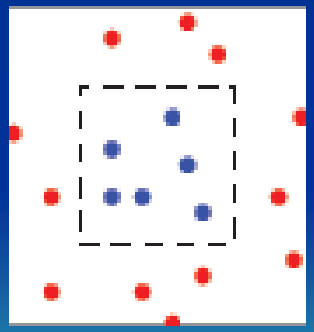
where [m] = {1, …, m}.

Empirical Risk Minimization

- The terms empirical error and empirical risk are often used interchangeably for training error.
- Since the training sample is the snapshot of the world that is available to the learner, it makes sense to search for a solution that works well on that data.
- This learning paradigm, which tries to come up with a predictor h that minimizes $L_S(h)$ is called Empirical Risk Minimization or ERM.

Overfitting

- Although the ERM rule seems very natural, without being careful, this approach may fail miserably.
- To demonstrate such a failure, let us go back to the problem of learning to predict the taste of a papaya on the basis of its softness and color.
- Consider a sample as depicted in the following:



Overfitting

- Assume that the probability distribution D is such that instances are distributed uniformly within the larger square and the labeling function f determines the label to be 1 if the instance is within the inner square, and 0 otherwise. The area of the larger square in the picture is 2 and the area of the inner square is 1.
- Consider the following predictor:

$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases}$$

Overfitting

- No matter what the sample is, $L_S(h_S) = 0$, and therefore this predictor may be chosen by an ERM algorithm.
- On the other hand, the true error of any predictor that predicts the label 1 only on a finite number of instances is, in this case, 1/2. Thus, $L_{D,f}(h_S) = 1/2$.
- So we have found a predictor whose performance on the training set is excellent, yet its performance on the true "world" is very poor. This phenomenon is called overfitting.
- That is, being correct by chance.
- Intuitively, overfitting occurs when our hypothesis fits the training data "too well".

Empirical Risk Minimization with Inductive Bias

- We have just demonstrated that the ERM rule might lead to overfitting.
- Rather than giving up on the ERM paradigm, we will look for ways to rectify it. We will search for conditions under which there is a guarantee that ERM does not overfit, namely, conditions under which when the ERM predictor has good performance wrt the training data, it is also highly likely to perform well over the underlying data distribution.
- A common solution is to apply the ERM learning rule over a restricted search space.

Empirical Risk Minimization with Inductive Bias

- Formally, the learner should choose in advance (before seeing the data) a set of predictors. This set is called a hypothesis class and is denoted by H. Each $h \in H$ is a function from X to Y. For a given class H, and a training sample, S, the ERM_H learner uses the ERM rule to choose a predictor $h \in H$, with the lowest possible error over S. Formally,

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h)$$

where argmin stands for the set of hypotheses in H that achieve the minimum value of $L_S(h)$ over H.

Empirical Risk Minimization with Inductive Bias

- By restricting the learner to choosing a predictor from H, we bias it toward a particular set of predictors. Such restrictions are often called an inductive bias.
- Since the choice of such a restriction is determined before the learner sees the training data, it should ideally be based on some prior knowledge about the problem to be learned. For example, for the papaya problem we may choose the class H to be the set of predictors that are determined by axis aligned rectangles. We will later show that ERM_H over this class is guaranteed not to overfit. On the other hand, the example of overfitting that we have seen previously, demonstrates that choosing H to be a class of predictors that includes all functions that assign the value 1 to a finite set of domain points does not suffice to guarantee that ERM_H will not overfit.

- A fundamental question in learning theory is, over which hypothesis classes ERM_H learning will not result in overfitting.
- Intuitively, choosing a more restricted hypothesis class better protects us against overfitting but at the same time might cause us a stronger inductive bias.
- We will study this fundamental tradeoff.

Finite Hypothesis Classes

- The simplest type of restriction on a class is imposing an upper bound on its size (that is, the number of predictors h in H).
- We will show that if H is a finite class then ERM_H will not overfit, provided it is based on a sufficiently large training sample (this size requirement will depend on the size of H).

Finite Hypothesis Classes

- Limiting the learner to prediction rules within some finite hypothesis class may be considered as a reasonably mild restriction.
- For example, H can be the set of all predictors that can be implemented by a Python program written in at most 10^9 bits of code.

Finite Hypothesis Classes

- Another example of H is the class of axis aligned rectangles for the papaya learning problem, with discretized representation.

Performance Analysis of ERM_H

- H is a finite class.
- For a training sample, S, labeled according to some $f : X \to Y$, let h_S denote a result of applying ERM_H to S, namely,

$$h_S \in \operatorname*{argmin}_{h \in H} L_S(h)$$

Performance Analysis of ERM_H

The Realizability Assumption:

There exists $h^* \in H$ such that $L_{(D,f)}(h^*) = 0$.

Note that this assumption implies that with probability 1 over random samples, S, where the instances of S are sampled according to D and are labeled by f,

we have $L_S(h^*) = 0$.

Performance Analysis of ERM_H

- Any guarantee on the error with respect to the underlying distribution D, for an algorithm that has access only to a sample S, should depend on the relationship between D and S.
- The common assumption in statistical machine learning is that the training sample S is generated by sampling points from the distribution D independently of each other.
- Expressed formally:

the i.i.d assumption

The examples in the training set are independently and identically distributed (i.i.d.) according to the distribution D. That is, every x_i in S is freshly sampled according to D and then labeled according to the labeling function, f. We denote this assumption by $S \sim D^m$ where m is the size of S, and D^m denotes the probability over m-tuples induced by applying D to pick each element of the tuple independently of the other members of the tuple.

- Intuitively, the training set S is a window through which the learner gets partial information about the distribution D over the world and the labeling function, f. The larger the sample gets, the more likely it is to reflect more accurately the distribution and labeling used to generate it.

Confidence Parameter (1-δ)

- Since the training set S is picked by a random process, it is not realistic to expect that with full certainty S will suffice to direct the learner toward a good predictor (from the point of view of D), as there is always some probability that S happens to be very nonrepresentative of D.
- In the papaya tasting example, there is always some chance that all the papayas we have happened to taste were not tasty, in spite of the fact that, say, 75% of the papayas in our island are tasty. In such a case, $ERM_H(S)$ may be the constant function that labels every papaya as not tasty (and has 75% error on the true distribution of papayas in the island). Therefore …

Confidence Parameter (1-δ)

- Therefore, we will address the probability to sample a training set for which $L_{(D,f)}(h_S)$ is not too large. Usually, we denote the probability of getting a non-representative sample by δ, and call $(1 - \delta)$ the confidence parameter of our prediction.

Accuracy Parameter ϵ

- Furthermore, since we cannot guarantee perfect label prediction, we need another parameter for the quality of prediction, the accuracy parameter, commonly denoted by ϵ.
- We interpret the event $L_{(D,f)}(h_S) > \epsilon$ as a failure of the learner, while if $L_{(D,f)}(h_S) \leq \epsilon$ we view the output of the algorithm as an approximately correct predictor.
- Therefore …

Accuracy Parameter ϵ

- Therefore, we are interested in upper bounding the probability to sample m-tuple of instances that will lead to failure of the learner. The labeling function f : X → Y is fixed.
- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set. We would like to upper bound

$$D^m(\{S|_x : L_{(D,f)}(h_S) > \epsilon\})$$

- Let H_B be the set of bad hypotheses:

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

and M be the set of misleading samples:

$$M = \{S|_x : \exists h \in H_B, L_S(h) = 0\}$$

- *M* is misleading, because $\forall S|_x \in M$, there is a bad hypothesis that looks like a "good" hypothesis on $S|_x$.

Upper Bounding the Probability of Learner's Failure

- We want to bound the probability of the event $L_{(D,f)}(h_S) > \epsilon$.
- Since the realizability assumption implies that $L_S(h_S) = 0$, it follows that the event $L_{(D,f)}(h_S) > \epsilon$ can only happen if for some $h \in H_B$ we have $L_S(h) = 0$. In other words, this event will only happen if our sample is in the set of misleading samples, *M*. So, formally, we have shown that

$$\{S|_x : L_{(D,f)}(h_S) > \epsilon\} \subseteq M.$$

- Rewriting *M* as

$$M = \bigcup_{h \in H_B} \{S|_x : L_S(h) = 0\}$$

we have …

Upper Bounding the Probability of Learner's Failure

we have

$$D^m\left(\left\{S|_x : L_{(D,f)}(h_S) > \epsilon\right\}\right) \leq D^m(M) = D^m\left(\bigcup_{h \in H_B}\{S|_x : L_S(h) = 0\}\right)$$

- Applying the union bound property from the Probability Theory to the right-hand side of the preceding equation yields

$$D^m\left(\left\{S|_x : L_{(D,f)}(h_S) > \epsilon\right\}\right) \leq \sum_{h \in H_B} D^m(\{S|_x : L_S(h) = 0\}) \qquad (*)$$

- Next, let us bound each summand of the right-hand side of the preceding inequality. Fix some "bad" hypothesis $h \in H_B$. The event $L_S(h) = 0$ is equivalent to the event $\forall i, h(x_i) = f(x_i)$. Since the examples in the training set are sampled i.i.d. we get

$$\begin{aligned}\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) &= \mathcal{D}^m(\{S|_x : \forall i, h(x_i) = f(x_i)\}) \\ &= \prod_{i=1}^{m} \mathcal{D}(\{x_i : h(x_i) = f(x_i)\}).\end{aligned} \qquad (**)$$

Upper Bounding the Probability of Learner's Failure

- For each individual sampling of an element of the training set we have

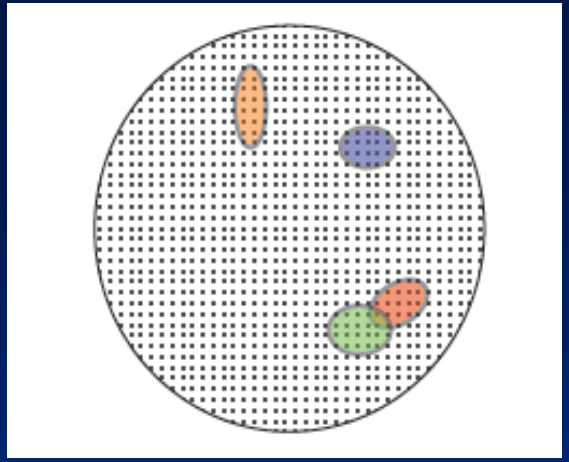
$$D(\{x_i : h(x_i) = f(x_i)\}) = 1 - L_{(D,f)}(h) \leq 1 - \epsilon,$$

where the last inequality follows from the fact that $h \in H_B$ such that $L_{(D,f)}(h) > \epsilon$. Combining the previous equation with Equation (**) and using the inequality $1 - \epsilon \leq e^{-\epsilon}$ we obtain that for every $h \in H_B$,

$$D^m(\{S|_x : L_S(h) = 0\}) \leq (1 - \epsilon)^m \leq e^{-\epsilon m}.$$

Combining this inequality with Inequality (*) we conclude that $D^m(\{S|_x : L_{(D,f)}(h_S) > \epsilon\}) \leq |H_B|e^{-\epsilon m} \leq |H|e^{-\epsilon m}$.

A graphical illustration of the union bound result



Each point in the large circle represents a possible m-tuple of instances. Each colored oval represents the set of misleading m-tuple of instances for some bad predictor $h \in H_B$. The ERM can potentially overfit whenever it gets a misleading training set S. That is, for some $h \in H_B$ we have $L_S(h) = 0$. The result of the union bound guarantees that for each individual bad hypothesis, at most $(1-\epsilon)^m$-fraction of the training sets would be misleading. In particular, the larger m is, the smaller each of these colored ovals becomes. The union bound formalizes the fact that the area representing the training sets in M is at most the sum of the areas of the colored ovals. Therefore, it is bounded by $|H_B|$ • (the maximum size of a colored oval). Any sample S outside the colored ovals cannot cause the ERM to overfit.

So we have derived the following theorem about learnability.

Theorem 1

Let *H* be a finite hypothesis class. Let $\delta \in (0,1)$ and $\epsilon > 0$ and let *m* be an integer that satisfies

$$m \geq \frac{\ln(|\mathrm{H}|/\delta)}{\epsilon}.$$

Then, for any labeling function *f*, and for any distribution *D* for which the realizability assumption holds (that is, for some $h \in H, L_{(D,f)}(h) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample *S* of size *m*, we have that for every ERM hypothesis h_S, it holds that

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

- Theorem 1 tells us that for a sufficiently large sample m, the ERM_H rule with a finite hypothesis class H will be probably (with confidence $1 - \delta$) approximately (up to an error of ϵ) correct.