04_PAC_Learning_and_VC_Dimension

April 14, 2020

1 PAC-LEARNING AND VC-DIMENSION

In this chapter we will consider two important aspect of the statistical learning theory. We will focus our attention on the problem of inductively learning an unknown target function given only training examples and a space of candidate hypothesis. We already know that overfitting can occurs since the training error is a bad estimate of the generalization error and also when the learner doesn't see enough samples.Our goal is to answer question such as:

- Sample complexity : how many examples are needed for a learner to converge to a successful hypothesis?
- Computational complexity : how much computational effort is needed for a learner to converge to a successful hypothesis?
- Mistake bound : how many training examples will the learner misclassify before converging to a successful hypothesis?

1.1 Probably learning and approximately correct hypothesis

For simplicity we will restrict our discussion to the case of learning bolean-valued concepts from noise-free data. Despite this restriction most of the results can be generalized to other cases. Let's consider the set of all possible instances as $\mathbf{X}$ and the set of target concepts that our learner will try to learn as $\mathbf{C}$. In the case of boolean-valued function we have that each concept c corresponds to $c : \mathbf{X} \to \{0, 1\}$. Let $\mathcal{D}$ the training data drawn from $\mathbf{X}$ according to some stationary distribution $\mathcal{P}$. The learner L will try to find an hypothesis h, among all the hypothesis contained in the hypothesis space $\mathcal{H}$, that is a good approximation of the concept c.

$$h^* = arg \min_{h \in \mathcal{H}} error_{train}(h)$$

We define the error of an hypothesis, as the probability that such hypothesis will misclassify an instance.

$$error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}}[h(x) \neq c(x)]$$

This is the definition of the train error since x is defined from the data set. The true error is the one in which the input can be drawn from the true, unknown probability.

$$error_{true}(h) = \Pr_{x \in \ P}[h(x) \neq c(x)]$$

This, the true error, is the one that we expect to see when we will use the learner L onto unseen samples. The most of the analysis on the complexity of learning centers around the question *"how probable is that the the training error for h gives a misleading estimate of the true error?"*. We say that when the training error is higher than the true error the model is overfitting.

Should be noted that the error stricly depends on the probability.

In our analysis we will not require rhat the learner L output a zero error hypothesis, we wil just require that the error of the learner is bounded by some constant ε, that can be made arbitrarily small. Moreover, we will not require that the learner succeed for every training set, we will require that its probability of failure is bounded by a constant δ, that can be made arbitrarily small.

We want a learner L that probably learn a hypothesis that is approximately correct.

1.1.1 Version space

We have now understood that one the most critical aspect of a machine learning problem in the sample size, indee, in most practical settings the factor that limits the success of a learner is the limited number of samples. This problem is usually called *sample complexity.*

We define a learner **consistent** if it outputs hypotheses that perfectly fit the training data.

$$Consistent(h, \mathcal{D}) = \forall \langle x, c(x) \rangle \in \mathcal{D}, h(x) = c(x)$$

We now derfine the **version space** as the set of all hypotheses $h \in \mathcal{H}$ that correctly classify the training examples

$$VS_{\mathcal{H},\mathcal{D}} = \{h \in \mathcal{H} | Consistent(h, \mathcal{D})\}$$

Every consistent learner outputs a hypothesis belonging to the version space since, by definition, the version space constains every consistent hypothesis in $\mathcal{H}$. Now the question, can we bound the $error_{true}$ of a consistent learner?

Theorem

*If the hypothesis space $\mathcal{H}$ is **finite** and $\mathcal{D}$ is a sequence of $N \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that $VS_{\mathcal{H},\mathcal{D}}$ containts a hypothesis erorr greater than ϵ is less than $|\mathcal{H}|e^{-\epsilon N}$:*

$$\Pr(\exists h \in \mathcal{H} : L_{train}(h) = 0 \wedge L_{true} \geq \epsilon) \leq |\mathcal{H}|e^{-\epsilon N}$$

With the theorem above we are bounding the *probability* that a bad event occur. We have found an upper bound that relates the error on the training set of consistent learner to the true error. This bound depends on the number of samples, on the allowed error and on the size of the hypothesis space. We have explicitly found the parameters that govern the generalization capability of a model.

Proof

$$\Pr((L_{train}(h_1) = 0 \wedge L_{true}(h_1) \geq \epsilon) \vee ... \vee (L_{train}(h_{|\mathcal{H}|}) = 0 \wedge L_{true}(h_{|\mathcal{H}|}) \geq \epsilon$$

$$
\begin{aligned}
&\leq \sum_{h \in \mathcal{H}} \Pr(L_{train}(h) = 0 \wedge L_{true}(h) \geq \epsilon) && \text{(union bound)} \\
&\leq \sum_{h \in \mathcal{H}} \Pr(L_{train}(h) = 0 | L_{true}(h) \geq \epsilon) && \text{(bound using Bayes' rule)} \\
&\leq \sum_{h \in \mathcal{H}} (\ -\epsilon)^N && \text{(bound on individual } h_i s) \\
&\leq |\mathcal{H}|(1-\epsilon)^N && (k \leq |\mathcal{H}|) \\
&\leq |\mathcal{H}| e^{-\epsilon N} && (1 - \epsilon \leq e^{-\epsilon} \text{ for } 0 \leq \epsilon \leq 1)
\end{aligned}
$$

We can now use this result to determine the number of training samples required to reduce the probability of failure below some desired level δ:

$$|\mathcal{H}| e^{-\epsilon N} < \delta$$

Fixing, the accuracy ϵ and the confidence δ we can found the required number of samples:

$$N \geq \frac{1}{\epsilon}\Big(\ln |\mathcal{H}| + \ln\Big(\frac{1}{\delta}\Big)\Big)$$

The expression above provides a general bound on the number of training points sufficient for any consistent learner to successfully learn any taget concept in the hypothesis space, for any desired level of confidence and accuracy. In a similar manner, given the number of samples and a level of confidence, we can calculate the accuracy:

$$\epsilon \geq \frac{1}{N}\Big(\ln |\mathcal{H}| + \ln\Big(\frac{1}{\delta}\Big)\Big)$$

Example: Learning conjunctions of boolean literals A boolean literal is any boolean variable, e.g. "$Flat$", or its negation "$\neg Flat$". Conjuction of boolean literals includes concepts as "$Flat \wedge \neg Eart$". Now consider the hypothesis space $\mathcal{H}$ defined by conjuctions of literals based on M boolean variables. The size of $\mathcal{H}$ is 3^M, where the number 3 is due to the possibility to include the variable as a literal in the hypothesis, include its negation as a literal and its absence. The question is, how many examples are sufficient to ensure with probability at least $(1-\delta)$ that every h in $VS_{\mathcal{H},\mathcal{D}}$ satisfy $L_{true} \leq \epsilon$?

$$N \geq \frac{1}{\epsilon}\Big(\ln 3^M + \ln\Big(\frac{1}{\delta}\Big)\Big)$$

For example, if the boolean literals are 10 and we want a confidence of 95% with an error less than 0.1, the number of samples should be at lest 140.

Consider now a class C of possible target concepts and a learner L using hypothesis space $\mathcal{H}$, we will say that the concept class C is PAC-learnable by L using $\mathcal{H}$ if, for any target concept c in C, L will output a hypothesis h with $error_{\mathcal{D}}(h) < \epsilon$ with probability $(1-\delta)$, after observing a reasonable number of training examples and performing a reasonable amount of computation.

Definition

C is **PAC-learnable** if there exists an algorithm L such that for every $f \in C$, for any distribution $\mathcal{P}$, for any ϵ such that $0 \leq \epsilon \leq 1/2$, and δ such that $0 \leq \delta \leq 1$, algorithm L, with probability at least $1 - \delta$, outputs a concept h such that $L_{true}(h) \leq \epsilon$ using a n umber of samples that is polynomial of $1/\epsilon, 1/\delta$.

Definition

C is **efficiently PAC-learnable** by L using $\mathcal{H}$ iff for all $c \in C$, distributions $\mathcal{P}$ over $\mathbf{X}$, ϵ such that $0 \leq \epsilon \leq 1/2$, and δ such that $0 \leq \delta \leq 1$, algorithm L, with probability at least $1 - \delta$, outputs a concept $h \in \mathcal{H}$ such that $L_{true}(h) \leq \epsilon$, in a time that is **polynomial** in $1/\epsilon, 1/\delta, M$ and $size(c)$.

1.1.2 Agnostic Learning

In all the formulation just presented, the root assumption is that, in $\mathcal{H}$ exists a concepts c such that the training error is zero. Obviously, this assumption cannot always be satisfied. A learner that makes no assumption that the target concept is in the hypothesis spac, and that simply finds the hypothesis with minimum error on the training set, is called **agnostic** learner. Agnostic learner will always output an hypothesis h such that $error_{\mathcal{D}}(h) > 0$, so, can we bound $error_{true}(h)$ given $error_{\mathcal{D}}(h)$? Explicitly, we want that:

$$error_{true}(h) \leq error_{\mathcal{D}}(h) + \epsilon$$

We can use the **Hoeffding bound** which allows to characterize the deviation between the true probability of some event and its probability observed frequency over N independent trials. For any N i.i.d. coin flips $X_1, ..., X_N$, where $X_i \in \{0, 1\}$ and $0 \leq \epsilon \leq 1$, we define the empirical mean as $\bar{X}$, obtaining the following bound:

$$\Pr(\mathbb{E}[\bar{X}] - \bar{X} > \epsilon) < e^{-2N\epsilon^2}$$

Theorem

*If the hypothesis space $\mathcal{H}$ is finite and the dataset $\mathcal{D}$ is a sequence of i.i.d. examples of some concept c, than for any $0 \leq \epsilon \leq 1$, then for any learned hypothesis h in $\mathcal{H}$:

$$\Pr(L_{true}(h) - L_{train}(h) > \epsilon) \leq |\mathcal{H}| e^{-2N\epsilon^2}$$

We can bound the true error as:

$$L_{true}(h) \leq L_{train}(h) + \sqrt{\frac{\ln |\mathcal{H}| + \ln(1/\delta)}{2N}}$$

We found again the bias-variance decomposition since the first term on the right-hand-side of the equation is the bias and the second one is the variance. So:

- For large $|\mathcal{H}|$
  - we have low bias, assuming we can find a good hypothesis
  - we have high variance, because the bound is loser

- For small $|\mathcal{H}|$
  - we have high bias
  - we have lowe variance

Given δ and ϵ, how large should be N?

$$N \geq \frac{1}{2\epsilon^2}\Big(\ln|\mathcal{H}| + \ln\Big(\frac{1}{\delta}\Big)\Big)$$

This bound is nothing but the generalization of the previous equation when the learner picks the best hypothesis but, the best hypothesis, may have nonzero training error.

1.2 Sample complexity for infinite hypothesis spaces

In the above section we were able to bound the sample complexity in the case in which the cardinality of the hypothesis space was finite. But, what about continuous hypothesis spaces? Using the same equation as before we will obtain an infinite bound and so an infinite variance. Actually no, the bound previously provided is just a pessimistic bound.

Example: Learning axes aligned rectangle

We want to learn an unknown target axes-aligned rectangle R. We randomly drawn samples with a labele that indicate wheter or not the point is contained in R. Consider the hypothesis corresponding to the tightest rectangle R' around positive samples. The error region is the difference between R and R'. This region can be seen as the union of four rectangular regions.

In each of these regions we want an error less than $\epsilon/4$. When N samples are drawn, a bad event for a single rectangular region is when the probability of all the N samples of being outside this region, which is at most $(1-\epsilon/4)^N$. The same holds for the other three regions. The same holds also for the other three regions and by union bound we obtain $4(1-\epsilon/4)^N$. If we want that the probability of a bad event is less than δ we simply impose:

$$(1-\epsilon/4)^N \leq \delta$$

By exploiting the inequality $(1-x) \leq e^{-x}$, we get:

$$N \geq (4/\epsilon)\ln(4/\delta)$$

It is important in the example above the number of points that can be correctly classified. Can we get a bound error as a function of the number of points that can be completely labeled? Here we will consider another measure of the complexity of $\mathcal{H}$, called the **VApnik-Chervonenkis dimension** which allows to state bounds on sample complexity that use the $VC(\mathcal{H})$ instead of $|\mathcal{H}|$.

1.2.1 Shattering a set of instances

The VC dimension measures the complexity of the hypothesis space not by the number of distinct hypotheses, but by the number of distinct instances from $\mathcal{D}$ that can be completely discriminated using $\mathcal{H}$.

Definition (Dichotomy)

A **dichotomy** of a set S is a partition of S into two disjoint subsets

Definition (Shattering)

A set of instances S is **shattered** by hypothesis space $\mathcal{H}$ iff for every dichotomy of S there exists some hypothesis $h \in \mathcal{H}$ consistent wuth this dichotomy

It is important, whene we use the VC dimension, to specify the definition of the dichotomy and of the shuttering!

Let's now consider, as an example, the set X of instances corresponding to points on the x, y plane.

Let $\mathcal{H}$ be the set of all linear decision surfaces in the plane, whast is the VC dimension of this hypothesis space? Any two disting points in the plane can be shattered by $\mathcal{H}$, because we can find four linear surfaces that the plane include neither, either, or both points. In the case of three points we will be able to find 2^3 linear surfaces that shatter them. Of course three colinear points cannot be shattered..so, what is the VC dimension in this case? It will be at least three. The definition of the VC dimension requires that, if we can find a configuration of instances (points) of size d for which, we can correctly shatter any possible dichotomy of these instances, than the VC dimension is at least d. To upper bound the VC dimension we just have to show that $VC(\mathcal{H}) < d$

Definition

The **Vapnik-Chervonenkis dimension**, $VC(\mathcal{H})$, of hypothesis space $\mathcal{H}$ defined over instance space X is the size of the largest finite subset of X shattered by $\mathcal{H}$. If arbitrarily large finite sets of X can be shattered, then $VC(\mathcal{H}) \equiv \infty$

How many points can a linear boundary classify exactly in $M-$dimension? The solution is $M + 1$. The rule of thumb is that the number of parameters in the model often mathces the maximum number of points. However, in general, these two numbers can be completely different. There are problems where the number of parameters is infinite but the VC dimension is finite and others in which with just 1 parameter we have a VC dimension which is infinite.

Some examples of VC dimension:

- Linear classifier: $VC(\mathcal{H}) = M + 1$, for M features plus the constant term
- Neural networks: $VC(\mathcal{H}) =$ number of parameters
- 1-Nearest neighbor: $VC(\mathcal{H}) = \infty$
- SVM with Gaussian kernel: $VC(\mathcal{H}) = \infty$

How many randomly drawn examples suffice to guarantee error of at most ϵ with probability at least $(1 - \delta)$?

$$N \geq \frac{1}{\epsilon}\left(4 \log_2\left(\frac{2}{\delta}\right) + 8VC(\mathcal{H}) \log_2\left(\frac{13}{\epsilon}\right)\right)$$

We can now express the PAC bound using VC dimension as:

$$L_{true}(h) \leq L_{train}(h) + \sqrt{\frac{VC(\mathcal{H})\Big(\ln \frac{2N}{VC(\mathcal{H})} + 1\Big) + \ln \frac{4}{\delta}}{N}}$$

Also in this case we can recognize the bias-variance decomposition and we have stated in statistical terms the intuition behind the dependence of the error on the training size. In **structural risk minimization** we choose the hypothesis space to minimize the above bound on expected true error.

Theorem

The VC dimension of a hypothesis space $|\mathcal{H}| < \infty$ is bounded from above:

$$VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$$

Proof

If $VC(\mathcal{H}) = d$ then there exist at least 2^d functions in $\mathcal{H}$, since there are at least 2^d possible labelings: $|\mathcal{H}| \geq 2^d$

Theorem

Concept class C with $VC(C) = \infty$ is not PAC-learneble.

1.3 References:

1. *Mithcell T.M., "Machine Learning", chapters: 7.1, 7.2, 7.3, 7.4*
2. *Restelli M., Course slides*