Indian Institute of Information Technology, Allahabad

Department of Information Technology

Machine Learning Assignment

as

part of C2 assessment

Submitted By

Mrityunjaya Tiwari
IIT2019239

1 Consider a learning problem with n boolean attributes. Let the hypothesis class be H. Let c $\in H$ be the target concept, and D be a set of m independent, randomly drawn examples from c. A hypothesis is said to be consistent with D if it has zero prediction error on the examples in D. Let H' denote the subset of hypotheses such that $\forall h \in H'$ generalization error $\epsilon(h) > \gamma$. Give an upper bound on the probability that $\exists h \in H'$ such that h is consistent with D. You can use the fact that $P(|\epsilon(h) - \hat{\epsilon}(h)| > \gamma) \leq 2exp(-2\gamma^2 m)$.

Solution: Let us assume that $H' \in H$ is finite. Consider $h_1 \in H'$. We know that $\epsilon(h_1) > \gamma$. We need to find the probability that h_1 correctly classifies m training examples randomly sampled from D.

Suppose we draw one training example (x_1, y_1). The probability of h_1 classifying it correctly is $\mathrm{P}[\mathrm{h}_1(x_1) = y_1] \leq (1 - \gamma)$

The probability of h_1 being right m times - $\mathrm{P}_D^m[h_1(x_i) = y_i] \leq (1 - \gamma)^m$

Now consider a second hypothesis $h_2 \in H'$. The probability of either of them classifying the m training examples correctly - $\mathrm{P}^m D[h_1 \cup h_2] = P^m D[h_1] + P^m D[h_2] - P^m D[h_1 \cap h_2]$
$\leq P^m D[h_1] + P^m D[h_2]$
$\leq 2(1 - \gamma)^m$

Hence, if there are k hypotheses, the probability of any one of them correctly classifying all examples is $\leq k(1 - \gamma)^m$. In this case, $k < |H'|$, hence $\mathrm{P}_D^m[h(x_i) = y_i] \leq |H'|(1 - \gamma)^m \quad \exists h \in H'$

Now, we know that when $0 \leq \gamma \leq 1, (1 - \gamma) \leq e^{-\gamma}$. Therefore, —H'—(1 - $\gamma)^m \leq |H'|e^{-\gamma m}$

Hence, for a finite hypothesis space $H' \in H$, given a set of m training examples drawn randomly and independently according to D, where $c \in H$ is the target concept, the probability that there exists a hypothesis $h \in H'$ with generalization error $\epsilon(h) > \gamma$ consistent with the training examples is less than $|H'|e^{-\gamma m}$. This upper bound is also known as the Blumer Bound.

---

2 Overfitting refers to the phenomenon of an algorithm overtly fitting the patterns in the training data which may not generalize well. On the other hand, underfitting refers to the phenomenon of an algorithm not being able to capture even the desirable patterns existing in the data (due to limitations on its representational power). In the description below, training data is the data over which the model is learned, and test data is some unseen data coming from the same distribution. Consider learning three different classifiers $C_1; C_2; C_3$ on a given data set such that C_1 has high training as well as test accuracies, C_2 has high training accuracy but low test accuracy, whereas C_3 has low training as well as test accuracies. Which one of the following statements is correct?

1. C_1 **is overfitting whereas** C_2 **is underfitting.**
2. C_1 **is overfitiing whereas** C_3 **is underfitiing.**
3. C_2 **is overfitting whereas** C_3 **is underfitting.**
4. C_2 **is underfitting whereas** C_3 **is overfitting.**
5. **None of the above**.

Justify your answer.

Solution: c) is the correct statement. C_2 is overfitting whereas C_3 is underfitting, while C_1 neither overfits nor underfits.

Generalization Error has three componenets, namely - irreducible error, square of the bias, and variance. Irreducible error refers to the error due to noise in the data distribution which can't be removed my any model. Bias is the difference between the true underlying data signal $f(x)$ and the estimated hypothesis $\hat{f}(x)$, and variance is the variance of the predictions made by the estimated hypothesis. Since we don't have access to $f(x)$, it is not possible to calculate the three components analytically. Hence, we use heuristics.

The general heuristic for calculating bias and variance uses the training and the test errors. We say a model has high bias, or underfits the data, when it has both high training as well as test errors. On the other hand, a model has high variance, or overfits the data, when it has low training error but high test error. This is quite clear, given that a model with high bias will not be even able to fit

to the training set, and hence will have high errors on both training as well as the test set, whereas a model with high variance will fit to the training data too well and will have low training error, but will have high generalization error due to the variance component being high. Hence, it performs poorly and has high error on the test set. A model with low bias and variance, performs reasonably well both on the training as well as the test error and has relatively low generalization error.

---

3 Suppose that you are working with a supervised machine learning (classification) problem and you have access to m examples $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ coming from some (unknown) underlying distribution. Assume that these are the only examples that you have access to. You would like to build a model to predict the ys from $x's$ so as to generalize well to future (unseen) data. You have a bunch of learning algorithms at hand and you would like to find out which of these is most suited for this problem. Since, you don't know yet about much about machine learning, the only thing that you can do is train each algorithm on a subset of the available data (called the training data), predict the accuracy of the learned model on a (possibly different) subset of the available data (called the test data) and finally declare as algorithm of choice the one having the highest accuracy on the test data. Now, consider the following scenarios to decide the train/test subsets:

1. **You choose the entire set of examples for training as well as for testing.**
2. **You randomly choose half of the examples as the training set and then independently (at random) choose another half as the test set. The two sets could be overlapping since they are chosen independent of each other.**
3. **You decide a number $m' \leq m$. You randomly choose m' examples as your training set. The remaining m' - m examples become your test set. The two sets are disjoint in this case.**

Answer the following:

- **Which of the above scenarios is likely to give you best performance on the unseen data? Argue for each case.**

- **If one decides to go for alternative (c), describe the considerations that you will make in choosing m'. Note that choosing m' decides the size of the training as well as the test set (since the total number of examples is fixed).**

Solution:

- c) gives the best performance on unseen data.

  a) gives the worst performance on unseen data. This is because we are effectively not using any test data whatsoever. The model is evaluated based on the data it has already seen, so it is bound to have high variance when fit to the data properly after fine-tuning and repeated training (unless the model itself has high bias to begin with).

  b) does not give good performance either, mainly because of two reasons. Firstly, the training and test sets could overlap, resulting in some of the data in the test set already been trained on by the model. This makes the test error lower than it should be, and gives a false estimate of generalization error. Secondly, it is normally not good practice to choose half of the data as training as well as the test set. The exact ratio depends on the total available data, since we might be missing out on a lot of training data if the dataset size is large, and it might not be a good representation of the true underlying data distribution. This automatically makes the generalization error of the trained model large.

  c) is the best and commonly used technique for reducing generalization error. We choose a percentage of the total data to be the training set, and the rest becomes the test set. In this case, the test set is from the same distribution as the training data, but the model hasn't seen it before, hence providing a good estimate of generalization error. Also, there is flexibility in choosing m', so we can decide exactly how much data we want for training as well as testing and make sure both of the processes have enough data to be properly executed.

- The choice of m' depends on the size of the dataset available for the problem. If we have a small dataset of size, say, around 1000 data instances, then we can assign m' to be 70% of the size of the dataset, and the size of the test dataset becomes 30% - in other words, a 70/30 split. If we have a larger

dataset of above 10,000 instances, we can assign a greater percentage of data for training, like a 80/20 split. With the advent of big data and data-hungry sample inefficient techniques like deep learning, it is recommended to do a 95/5 split, since we often have over a million data instances, and even 5% of the total data is enough to get an estimate of the generalization error.
