Chapter 13

Probably Approximately Correct Learning

Empirical Risk Minimization. Decision theory. Probably Approximately Correct Learning. VC dimension and shattering.

In Chapter 11 we saw how there is a trade-off between the ability of a model to fit the training data and the ability of the model to generalize from the training sample to the population of examples whose features we wish to predict. We did this by instantiating the model into different hypotheses, using different training sets (by *Bootstrapping*) and then measuring the *Bias*, which is the error of the mean prediction for each example, and the *Variance*, the dispersion of the predictions for each example. We saw how reducing *Bias* leads to an eventual increase in *Variance* due to *overfitting*. In this chapter we will look at the *Bias-Variance tradeoff* in more detail, with a more formal and grounded approach.

13.1 Empirical Risk Minimization

In brief, *Empirical Risk Minimization* consists in minimizing the training error. Or, more generally, minimizing a loss function measured on the training set, such as the classification error or the quadratic error in regression. This is what we have been doing when training regression or classification models in supervised learning. The name comes from trying to minimize the risk, which is the expected loss, and this is an empirical risk because we measure it on the training set. This contrasts with the true risk, or the average loss over all possible data, which we cannot measure directly. Furthermore, if we adjust the parameters to minimize the empirical risk, then the empirical risk becomes a biased estimate of the true risk (for example, the true error, if that is our loss function). However, we can use probability theory to find a probable upper bound on the true error based on the empirical error we minimized.

First, we note that, if $A_1, A_2, ..., A_k$ are random events, then the probability of at least one of them occurring cannot be larger than the sum of their probabilities:

$$P(A_1 \cup A_2 \cup ...A_k) \leq P(A_1) + P(A_2) + ... + P(A_k)$$

This is the *union bound*, an upper bound on the probability of the union of a set of random events. Furthermore, if $B_1, B_2, ..., B_m$ are independent random events following the Bernoulli distribution, which is the distribution of a random variable that can take the values 0 or 1 with probabilities ϕ and

$1 - \phi$ respectively, with $\hat{\phi}$ defined as:

$$P(B_i = 1) = \phi \qquad \hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} B_i$$

Then, the following *Hoeffding's inequalities* hold:

$$P(\phi - \hat{\phi} > \gamma) \leq e^{-2\gamma^2 m}$$

$$P(\hat{\phi} - \phi > \gamma) \leq e^{-2\gamma^2 m}$$

In other words, the probability that the mean of a set of random Bernoulli variables with the same probability $P(B_i = 1) = \phi$ deviating from ϕ by more than γ decreases exponentially with γ and the number of examples on the sample.We can rewrite this as the *Hoeffding's inequality*:

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2e^{-2\gamma^2 m}$$

This is useful because, in classification, we can consider the classification error for each example to be a Bernoulli random variable, with values of 0 or 1, and ϕ to be the probability of the classifier committing an error. In this case, $\hat{\phi}$ is the observed error rate on the training set, or the *empirical error*, and we train the classifier by finding the set of parameters that minimizes this error. Thus, we are doing *empirical risk minimization*(ERM) because the empirical error, which we try to minimize, is the risk of misclassification for examples in the training set. However, what we would really like would be to minimize the ϕ, the true error, which we cannot measure but is related to the empirical error $\hat{\phi}$ by Hoeffding's inequality. This gives us a probable upper bound on the true error and is the rationale behind the notion of *Probably Approximately Correct Learning*.

13.2 Probably Approximately Correct Learning

Let us consider $\mathcal{X}$ be the population of all possible examples and $c : \mathcal{X} \to \{0, 1\}$ the target function to learn, assigning each example to one of two possible classes. $\mathcal{H}$ is the hypothesis class the learner will explore and $\mathcal{D}$ is the probability distribution according to which examples are drawn from $\mathcal{X}$, and according to which the training sample S is obtained. Our learner will draw S from $\mathcal{X}$ according to distribution $\mathcal{D}$ and then find an hypothesis $\hat{h}$ that minimizes the empirical error, $\hat{E}_s$, measured on the sample S:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{E}_S(h)$$

This is the empirical error, which is the average error on the sample, while the true error of an hypothesis h is the probability of error for any example drawn from $\mathcal{X}$ according to distribution $\mathcal{D}$. In other words, the true error corresponds to the set of possible instances for which the learned hypothesis differs from the target function c:

$$E(h) = P_{x \sim D}\left(h(x) \neq c(x)\right)$$

The true error is not accessible to the learner, who can only compute the empirical error.

In general, it is not reasonable to assume that the true error will be zero, since we cannot include all possible examples in the training set and different hypotheses may seem correct on all the training set while making mistakes outside it. So we need a more realistic set of requirements for our learner. We

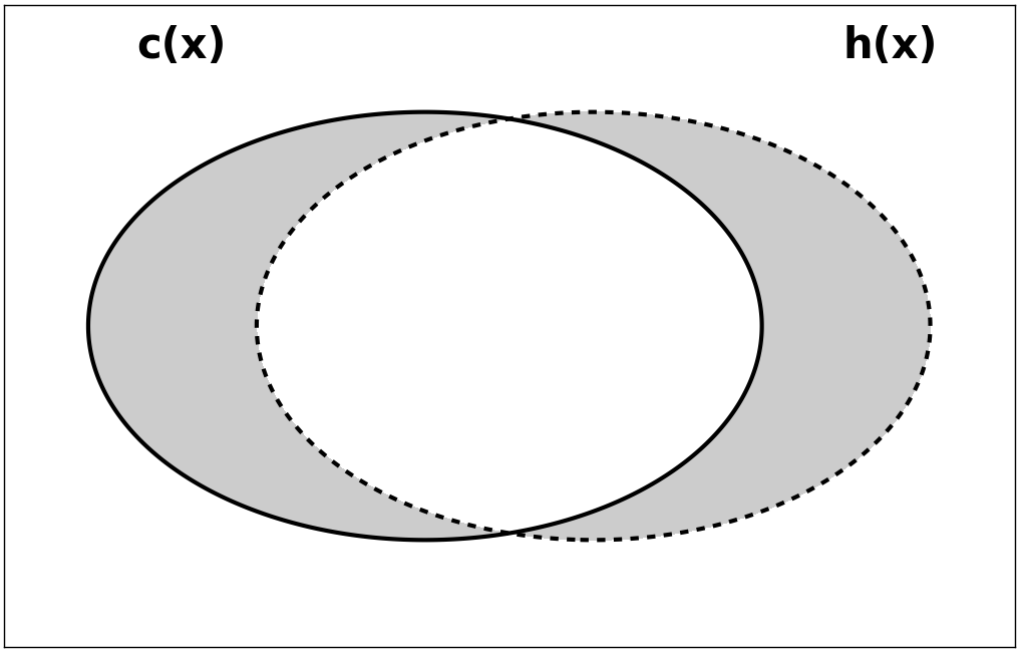

Figure 13.1: The true error of an hypothesis is the difference between the classifications given by that hypothesis and the classifications given by the function c providing the true classes of all points.

can demand that the result is *approximately correct*, in the sense that the true error of the hypothesis we find be below some threshold ϵ, instead of zero:

$$E(\hat{h}) \leq \epsilon$$

Furthermore, since we are training our classifier on a random subset of all possible examples, the training set may mislead our classifier into finding a hypothesis whose true error is not even bound by ϵ. So we require that our learner is *Probably Approximately Correct* (PAC):

$$P\left(E(\hat{h}) \leq \epsilon\right) \geq 1 - \delta$$

with $\epsilon < 1/2$ and $\delta < 1/2$. That is, there is a probability $1 - \delta$, with a small (below 0.5) δ, that the true error of the resulting hypothesis is some ϵ below 0.5. A learner is an *Efficient PAC learner* if it can learn hypothesis $\hat{h}$ in a time that is polynomial on $1/\epsilon$ and $1/\delta$.

Let us now suppose that we have a PAC learner, able to learn an hypothesis with a true error of ϵ or less with a probability of $1 - \delta$ or more, and let us assume that the hypothesis space $\mathcal{H}$ is finite and contains at least one hypothesis with $E(h) \leq \epsilon$, which must be true for there to be a chance of finding such hypotheses. Training and testing examples will all be drawn from $\mathcal{X}$ according to distribution $\sim \mathcal{D}$. Let us also define the *version space* as the set of *consistent hypotheses*, which are those hypotheses for which the empirical error is zero. This means that any consistent hypothesis (any hypothesis in the version space) minimizes the empirical error, since the empirical error cannot be less than zero.

We say that the version space is ϵ-*exhausted* if all hypotheses in the version space have a true error of at most ϵ:

$$\forall h \in \mathcal{V} \qquad E(h) < \epsilon$$

It is important to note that the learner cannot tell this, since the true error is not measurable by the learner. Conversely, the version space is not ϵ-*exhausted* if at least one hypothesis has a true error greater than ϵ.

What is the probability that no hypothesis in the version space has a true error larger than ϵ? In other words, what is the probability that the version space is ϵ-exhausted? If we suppose $h_1, h_2, ..., h_k$ are hypotheses with a true error greater than ϵ, $E(h_i) > \epsilon$, then the probability that h_i is consistent with one example is smaller than $1 - \epsilon$, since that is the probability of correct classification for a hypothesis with error ϵ. Thus, the probability of the hypothesis being consistent with all examples in a set of m examples is below $(1 - \epsilon)^m$. Using the *union bound* relation we saw previously:

$$P(A_1 \cup A_2 \cup ...A_k) \leq P(A_1) + P(A_2) + ... + P(A_k)$$

we know that the probability that any hypothesis h_i of the k hypotheses with $E(h_i) > \epsilon$ is consistent with m examples is $\leq k(1 - \epsilon)^m$, which is the sum of the probabilities of each hypothesis h_i being consistent with the set of examples. Although we do not know the value of k, which is the total number of such hypotheses, we know that k cannot be larger than the total number of hypotheses, $|\mathcal{H}|$. That is, $k(1 - \epsilon)^m \leq |\mathcal{H}|(1 - \epsilon)^m$. And since $(1 - \epsilon) \leq e^{-\epsilon}$ for $0 < \epsilon < 1$, the probability of an hypothesis with a true error greater than ϵ being in the version space (that is, being compatible with the training set) is bounded by:

$$P\left(\exists h \in \mathcal{V} : E(h) \geq \epsilon\right) \leq |\mathcal{H}|e^{-\epsilon m}$$

Let us now choose a value δ that is an upper bound on the probability that an hypothesis in the version space has a true error greater than ϵ. In this case, for $P\left(E(h \in \mathcal{V}) > \epsilon\right) \leq \delta$,

$$|\mathcal{H}|e^{-\epsilon m} \leq \delta \Leftrightarrow m \geq \frac{1}{\epsilon}\left(\ln \frac{|\mathcal{H}|}{\delta}\right)$$

This gives us a lower bound on the number of examples needed to have a probability of at least $1 - \delta$ of learning an hypothesis with a true error of at most ϵ.We can also compute the lower bound on ϵ as a function of the size of the training set, m, and the probability δ that the learner produces an hypothesis with an error greater than ϵ:

$$m \geq \frac{1}{\epsilon}\left(\ln \frac{|\mathcal{H}|}{\delta}\right) \Leftrightarrow \epsilon \leq \frac{1}{m}\left(\ln \frac{|\mathcal{H}|}{\delta}\right)$$

This assumes that the learner is a *consistent learner*. That is, a learner that learns hypothesis with zero empirical error, $\hat{E_S}(\hat{h}) = 0$. To extend this reasoning for $\hat{E_S} \geq 0$, we can consider the empirical (training) error to be the mean of Bernoulli variables corresponding to the classification error of each training example:

$$\hat{E}(h_i) = \frac{1}{m}\sum_{i=1}^{m} 1\{h(x^{(i)} \neq c(x^{(i)})\} = \frac{1}{m}\sum_{i=1}^{m} Z_i$$

Applying the Hoeffding inequalities we saw before:

$$P(\phi - \hat{\phi} > \gamma) \leq e^{-2\gamma^2 m}$$

$$P(\hat{\phi} - \phi > \gamma) \leq e^{-2\gamma^2 m}$$

gives us the following bounds:

$$P\left(\hat{E} - E > \epsilon\right) \leq e^{-2m\epsilon^2}$$

$$P\left(E - \hat{E} > \epsilon\right) \leq e^{-2m\epsilon^2}$$

Thus, the probability of the true error of hypothesis h being more than ϵ above the empirical error of h is bounded by:

$$P\left(E(h) > \hat{E}_S(h) + \epsilon\right) \leq e^{-2m\epsilon^2}$$

Extending this for all hypotheses $h \in \mathcal{H}$:

$$P\left(\exists h \in \mathcal{H} : E(h) > \hat{E}_S(h) + \epsilon\right) \leq |\mathcal{H}|e^{-2m\epsilon^2}$$

Calling this probability δ and solving for m, we obtain:

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

This gives us the lower bound on the size of the training set to guarantee a maximum probability of $delta$ that the true error of the hypothesis we find is greater than the sum of the empirical error and ϵ. This lower bound increases with the square of $1/\epsilon$ and with the logarithm of the total number of hypotheses, $|\mathcal{H}|$.

This result gives us an important insight into a major problem of machine learning, which is the *inductive bias*. We mentioned before that it is always necessary to assume something about the hypothesis space we are learning in order to be able to generalize from the training set to future examples. This assumption that restricts the hypothesis space is the inductive bias. For example, that the best regression curve will be a polynomial of some degree or the the best classifier will be a hyperplane with a specified number of dimensions. Let us now look at what happens with a classifier that has no inductive bias. For example, suppose that our hypothesis space $\mathcal{H}$ is the set of all subsets of $\mathcal{X}$. This means that our classifier can split $\mathcal{X}$ into two classes in any combination of examples by finding a subset of $\mathcal{X}$ defining one class and placing in the other class any example not in that subset. If this is the case, then the size of our hypothesis space is two raised to the number of possible examples, since each example may or may not belong to each subset: $|\mathcal{H}| = 2^{|\mathcal{X}|}$. Let us further assume that each example is described by a vector of n boolean features, for simplification, which means that $\mathcal{X}$ is the set of all 2^n combinations of features and the cardinality of our hypothesis space is:

$$|\mathcal{H}| = 2^{|\mathcal{X}|} = 2^{2^n}$$

Using this, we can compute the lower bound on the size of the training set for some value of ϵ and δ:

$$m \geq \frac{1}{2\epsilon^2}(\ln \frac{|\mathcal{H}|}{\delta}) \Leftrightarrow m \geq \frac{1}{2\epsilon^2}(2^n \ln \frac{2}{\delta})$$

The lower bound of m grows exponentially in the number of features, n, and since for an approximately correct learning we want ϵ to be below 0.5, this means that m will be greater than $|\mathcal{X}|$, the total number of all possible examples. In other words, without inductive bias we have no probably approximately correct learning when trying to extrapolate from the training set to new examples.

Let us now extend this analysis to the hypotheses obtained by empirical risk minimization (ERM). Recall that an ERM learner selects the hypothesis from $\mathcal{H}$ that minimizes the empirical error:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{E}(h)$$

Let us define the *generalization error* as the difference between the true error and the empirical error:

$$E(\hat{h}) - \hat{E}(\hat{h})$$

and h^* be the best hypothesis, in the sense of being the hypothesis with the smallest true error.

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

Let $1 - \delta$ be the probability that the true error of the ERM hypothesis is not greater than the empirical error plusϵ, $P(E(\hat{h}) \leq \hat{E}(\hat{h}) + \epsilon) = 1 - \delta$. Furthermore, the empirical error for the ERM hypothesis, given our training set S, cannot be greater than the empirical error of the best hypothesis, since the ERM hypothesis was obtained by minimizing the empirical error. Thus, $\hat{E}(\hat{h}) \leq \hat{E}(h^*)$. This means that the true error of the best hypothesis must also be bounded by the sum of the empirical error of the best hypothesis and ϵ with a probability of at least $1 - \delta$, because the best hypothesis, by definition, is the hypothesis with the lowest true error: $\hat{E}(h^*) \leq E(h^*) + \epsilon$ with $P \geq 1 - \delta$.

Combining all these, we find that, with a probability of at least $1 - \delta$, the true error of the ERM hypothesis we obtain by minimizing the empirical error cannot be greater than the true error of the best hypothesis plus two times ϵ: and $P(E(\hat{h}) \leq E(h^*) + 2\epsilon) \geq 1 - \delta$. Using the previous bounds, we can decompose the true error of the ERM hypothesis into these two terms:

$$E(\hat{h}) = \left(\arg\min_{h \in \mathcal{H}} E(h) \right) + 2\sqrt{\frac{1}{2m} \ln \frac{|\mathcal{H}|}{\delta}}$$

The first term is the smallest true error of any hypothesis in the hypothesis space $\mathcal{H}$, which corresponds to the *Bias* of our model, and the larger this term, the less the model is able to fit the data adequately. Thus, when this term dominates the true error, we say that our model is underfitting. The second term is a function of the size of the hypothesis space and the size of the training set, and corresponds to the *Variance* of our model. In general, the larger the hypothesis space the greater the variance of the predictions of the hypotheses obtained by training with different training sets. If this term dominates the true error, the model is overfitting since now the critical problem is not the model's inability to adjust to the points but rather its excessive freedom in adapting to the training set.

13.3 Shattering and the V-C Dimension

So far, we have assumed that the hypothesis space $\mathcal{H}$ is finite, which allowed us to obtain a lower bound for the size of the training set given the values of ϵ and δ:

$$m \geq \frac{1}{2\epsilon^2} \left(\ln \frac{|\mathcal{H}|}{\delta} \right)$$

This can be true for some classifiers, such as decision trees with a fixed limited depth, but is not true in general, as it is often the case that classifiers use continuous parameters and thus have an infinite number of possible hypotheses. For example, logistic regression, SVM, neural networks and so forth. In these cases, the previous expression is no longer useful and we need another approach.

We can start by thinking that, for a classifier with continuous parameters, there can be many hypotheses that result in the same set of labels for a given set of examples. A logistic regression, for example, can divide the same set of points into the same two subsets with an infinitude of lines, as long as the lines are placed between the sets to separate, as illustrated in Figure 13.2.

So what is relevant is how a classifier divides the set of examples into different subsets and not how many different decision frontiers it can express. This leads us to the following definition:

Hypothesis class $\mathcal{H}$ shatters set of points $\mathcal{S}$ if, for any labelling S of $\mathcal{S}$, there is a $h \in \mathcal{H}$ that is consistent with S

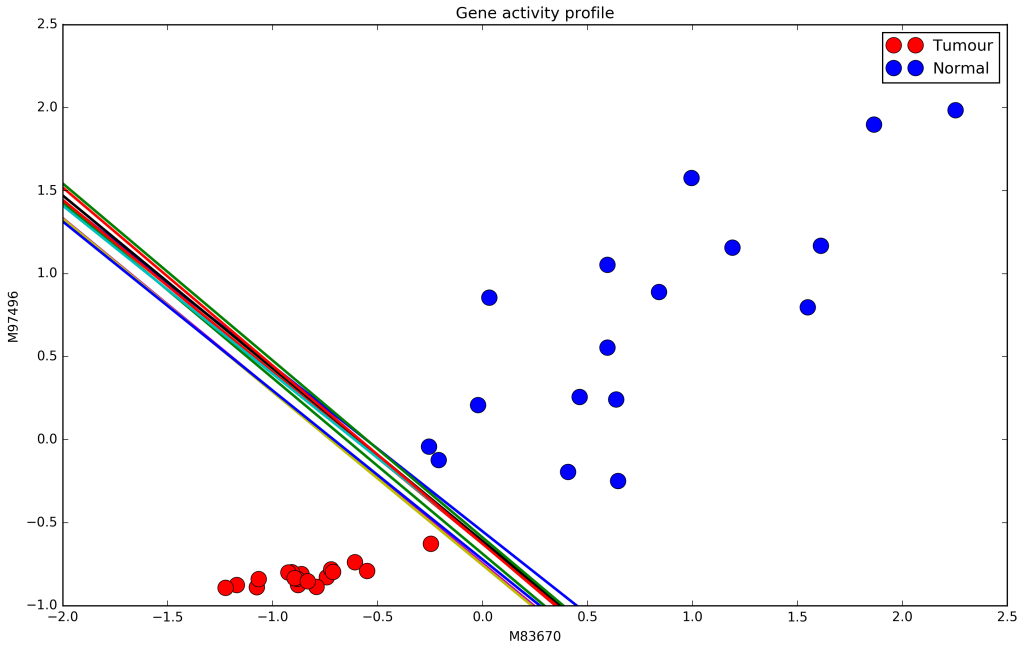

Figure 13.2: For classifiers with continuous parameters, although the size of the hypothesis space is infinite, there are also infinite hypothesis resulting in the same classifications for all points.

In other words, $\mathcal{H}$ shatters a set of points if it can provide hypotheses that can classify all those points correctly whatever the class each point belongs to. For example, a linear classifier in two dimensions can shatter a set of 3 points forming a triangle, as shown in Figure 13.3.

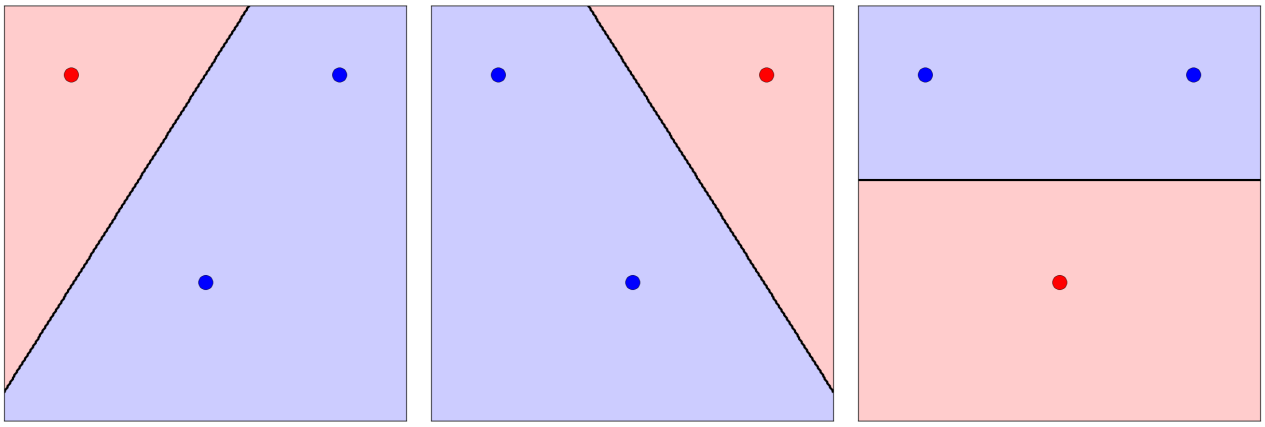

Figure 13.3: A linear classifier in two dimensions can shatter this set of 3 points by correctly classifying them whatever their labels. Note that two other cases, where all points belong to the same class, were omitted for being trivial.

Using this notion of *shattering*, we can define the *Vapnik-Chervonenkis dimension* of an hypothesis space $\mathcal{H}$, or, for short, the *V-C dimension* of $\mathcal{H}$, $VC(\mathcal{H})$, as the size of the largest set of points that $\mathcal{H}$ can shatter. Note that the points can be placed in the most adequate way to facilitate shattering and that there may be sets with fewer than $VC(\mathcal{H})$ points that $\mathcal{H}$ cannot shatter. For example, if two points overlap in the same coordinates no hypothesis can distinguish them. What matters is that some set of points exists for which the hypothesis space can provide hypotheses for correct classification whatever

the labels may be. Vapnik et. al. demonstrated that, with a probability of $1 - \delta$:

$$E(\hat{h}) \leq \hat{E}(\hat{h}) + \mathcal{O}\left(\sqrt{\frac{VC(\mathcal{H})}{m} \ln \frac{m}{VC(\mathcal{H})} + \frac{1}{m} \ln \frac{1}{\delta}}\right)$$

That is, the true error of the ERM hypothesis is bounded by the empirical error plus a term that is approximately proportional to the VC dimension of the hypothesis space ($VC(\mathcal{H})$) and approximately inversely proportional to the size of the training set (m). In other words, to keep the true error within some bounds, the size of the training set must increase as $VC(\mathcal{H})$ increases.

This has implications for the approach of using linear discriminants in higher dimensions to classify non linearly separable sets. The VC dimension of a linear classifier is $D + 1$, where D is the dimension of the feature vectors. As we increase the dimension D, we increase the VC dimension of the hypothesis space and thus we require a larger sample for the training set to prevent overfitting and an increase in the generalization error.

13.4 Summary

The probabilistic and statistical foundation of machine learning provides us with a good intuition about important aspects, even though, in practice, methods such as validation and testing provide better estimates of the true error of our models or hypotheses. In this chapter we saw how inductive bias is an important requirement for machine learning, since without it the hypothesis space becomes too large for allowing generalization from a data set to all possible points. We also saw how the true error results from a contribution of the error of the best hypothesis, corresponding to the bias of the model, and the generalization error due to the size of the hypothesis space, corresponding to the variance of the model. This is the source of the Bias-Variance tradeoff, since improving the best hypothesis of the model generally requires increasing the size of the hypothesis space. Most importantly, we saw the notion of Probably Approximately Correct learning. In machine learning we cannot guarantee that the prediction error will be zero but we can make it probable that it will be small, as long as we have enough data.

13.5 Further Reading

1. Alpaydin [2], Sections 2.1 through 2.3
2. Mitchell [18], Chapter 7 up to section 7.4

Bibliography

[1] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[3] David F Andrews. Plots of high-dimensional data. *Biometrics*, pages 125–136, 1972.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, New York, 1st ed. edition, oct 2006.

[5] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of www image search results using visual. Association for Computing Machinery, Inc., October 2004.

[6] Guanghua Chi, Yu Liu, and Haishandbscan Wu. Ghost cities analysis based on positioning data in china. *arXiv preprint arXiv:1510.08505*, 2015.

[7] Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[8] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning. Stanford CA Morgan Kaufmann*, pages 231–238, 2000.

[9] Hakan Erdogan, Ruhi Sarikaya, Stanley F Chen, Yuqing Gao, and Michael Picheny. Using semantic analysis to improve speech recognition performance. *Computer Speech & Language*, 19(3):321–343, 2005.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[11] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[12] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[13] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. Dna visual and analytic data mining. In *Visualization'97., Proceedings*, pages 437–441. IEEE, 1997.

[14] Chang-Hwan Lee, Fernando Gutierrez, and Dejing Dou. Calculating feature weights in naive bayes with kullback-leibler measure. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1146–1151. IEEE, 2011.

[15] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[16] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[17] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC, 1st edition, 2009.

[18] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[19] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[20] Roberto Valenti, Nicu Sebe, Theo Gevers, and Ira Cohen. Machine learning techniques for face analysis. In Matthieu Cord and Pádraig Cunningham, editors, *Machine Learning Techniques for Multimedia*, Cognitive Technologies, pages 159–187. Springer Berlin Heidelberg, 2008.

[21] Giorgio Valentini and Thomas G Dietterich. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *The Journal of Machine Learning Research*, 5:725–775, 2004.

[22] Jake VanderPlas. Frequentism and bayesianism: a python-driven primer. *arXiv preprint arXiv:1411.5018*, 2014.