

## Lecture 1 — Feb 25

Scribe: Julian Yarkony

Lecturer: Deva Ramanan

**Note:** These lecture notes are still rough, and have only have been mildly proofread.



This is the danger environment. A very very scary place.

## 1.1 Probability Review

Union bound. This defines an upper bound on  $P(a \text{ or } b)$ . Note this is an upper bound not an answer; to determine the max possible area  $P(a)$  and  $P(b)$  cover different regions and hence upper bound  $=P(a)+p(b)$ .

## 1.2 Chertoff bound

The chertoff bound is a tool used to determine how many trials are needed to establish bias of a coin or OTHER BINARY RANDOM EVENTS. The chertoff bound can be thought of as a function, given a coin which is supposed to be head at a rate of  $p$  ( $p \neq 1/2$ ); if bias is to be established the must be at least  $n$  trials (for a given level of significance indicated with epsilon

$$n \geq 5 \frac{1}{(p - .5)^2} * \ln\left(\frac{1}{\sqrt{\epsilon}}\right)$$

Let  $z_1, z_2, z_3 \dots z_n$  be independent bernouli events w/bias  $\Phi$

let the experimental mean of  $\Phi$  or  $\Phi^e$  be  $\frac{1}{m} \sum z_i$

$$p(|\Phi^e - \Phi| > \delta) < 2 * e^{-2*\delta^2*m}$$

which means that the prob that random variable lies far away from its mean is bounded by its variance

Given a training set  $S$  of  $x$   $y$  pairs. we assume samples are iid from  $D=p(xy)$  the joint distribution. ( $p(y)$  is bernouli and  $p(x|y)$  is normal

Let  $h$  be a particular predictor. The error often the training set can be described the following which is called the empirical loss

$$= \frac{1}{m} \sum I(h(x_i) \neq y_i)$$

### 1.3 Generalization error

$$E(h) = \sum_{x,y} p_{x,y} d(h(x) \neq y)$$

Our hope is that

$$\lim_{m \rightarrow \infty} \text{empirical}(E(h)) = E(h)$$

as more points are added the experimental equals the actual. The fundamental point is that the the loss can understood and bounded given the model and the number of data points. This is described below

$$E(\text{h empirical}) \leq E(\text{h or best hypothesis}) + 2 * \sqrt{.5 * m * \log\left(\frac{2 * k}{\delta}\right)}$$

The last term is the factor which is minimized. Often times there is not a discrete number of hypothesis. Logistic regression is an example (infinite number of places for the bound to be located. This is not always true decision trees with boolean inputs.

### 1.4 Empirical risk minimization

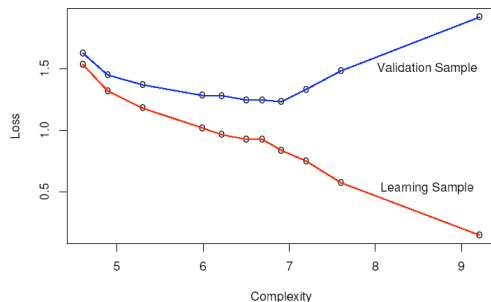
Empirical risk minimization is an important concept in machine learning denoting the goal of minimizing the risk or loss function. The risk associated with a particular function is simply the weighted sum (by probabilities) of the loss or

$$R(f) = \sum L(f(x_i), g(x_i)) * p(x_i)$$

Risk in this case can be thought of in many ways. We could use the square of the amount off if we want to penalize more and more as distance increases. More interestingly one could penalize more but at a decreasing rate.

### 1.5 Shattering

Shattering is a concept from support vector machines. It simply states whether a particular decision boundary separates the data (x's from o's on either side of the boundary). This leads to an important point in there may be no way not SHATTER a set of points in a given space. For example if there are 4 points plotted in a 2 dimensional space there may be no way to separate them. You can demonstrate this for yourself, draw one X and 2 O's in a line with the x in the middle and slightly above the O's. You now notice only one line separates them, next plot an X below one of the O's. You had already constrained the line as much as it could go then you placed one additional point. Note is separation can be made



**Figure 1.1.** Picture of the Bias-variance problem. note is stole this photo from the internet

then shattering has occurred.

Shattering relates in a nuanced way to the generalization bound

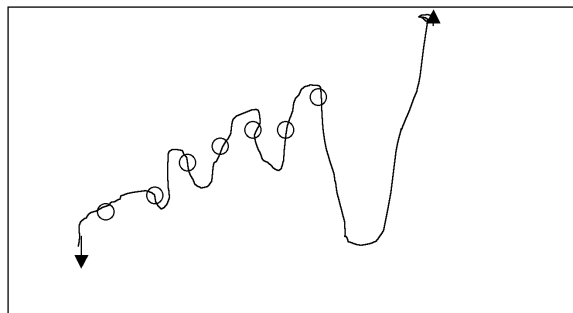
$$E(h_{\text{empirical}}) \leq \min_{h \text{ or best hypothesis}} + 2 * \sqrt{.5 * m * \log\left(\frac{2 * k}{\delta}\right)}$$

in this case  $k$  is the dimensionality. Higher  $k$ 's allow for higher variance as they are in a higher space.

## 1.6 Introduction to bias/variance

In artificial intelligence and machine learning one is confronted with the problem of how complex to make a model. If one makes a model too simple it can not capture the variability of the data. However if a model is too complex relations which are coincidental will be found, and the model will have no explanatory value (some machine learning algorithms can not only predict the data but also tell you what they did). Furthermore more complex models require more data and often more LABELED (data where a person gave the answers for the neural network, for example labeling digits for a neural network digit recognizer) data both of which are expensive. However there is no easy answer for how complex to make a model. One crucial thing to keep in mind is that when drawing a decision boundary all the the regularities of the problem and concepts underlying the problem need not be taken into account for classification particularly binary classification.

For example: Suppose one is making mathematical model to classify whether a person should vote for Obama or McCain in the general election. Great amounts of effort could be taken in to determine underlying ideology, special cases, policy positions which have sequential properties or joined properties (policy B can not be implemented before policy A; or policy A or B is good but not both). Dozens of questions with user defined weights could be used. HOWEVER A BINARY DECISION IS BEING MADE a simple model using equal weights on a few questions in all likelihood may be far more that sufficient to predict the vote of a person with high accuracy.



**Figure 1.2.** An example of overfitting

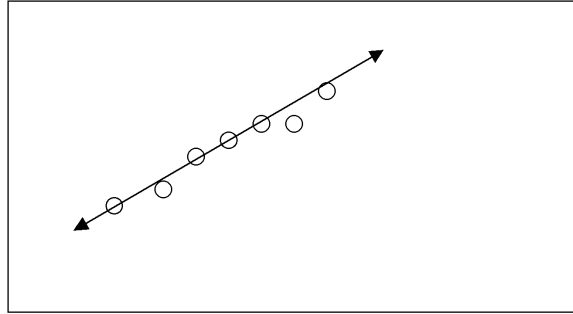
In this overview we will first discuss the foundational problem using intuition and graphics. Second the mathematics will be explored. Next regularization and different loss functions and a simple solution called cross validation will be brought it and how it is applied to neural networks will be discussed. All of the work in this review can be summarized as *entia non sunt multiplicanda praeter necessitatem* or entities should not be multiplied beyond complexity or the simplest explanation which fits the data is usually the right one.

## 1.7 Over-fitting vs Under-fitting

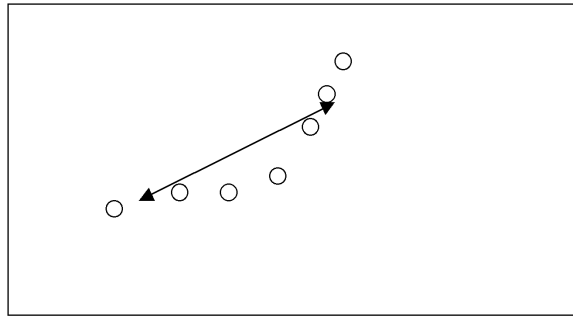
When discussing the quality of a model in regards to a set of data two commonly used terms are over-fitting and under-fitting. A model which is over-fitted is a model which has an excess of parameters. The added complexity may and often does help the model perform well on a set of training data BUT it inhibits prediction of future points. Figure 2 on over-fitting illustrates this. While it is clear from the picture that we are looking at data generated by a linear function plus noise (Is the human brain not a powerful machine that it can determine that on the fly) the overfitted example gains improved accuracy on the training set (the points which it learns on or the points which are drawn on the graph, while missing the overall message or pattern in the data). Noise, hidden factors, and difficult high level relations are primary causes of variability in data that can not or is difficult to capture with statistical models.

Under-fitting is the opposite of over-fitting. This occurs when the model is incapable of capturing the variability of the data. For example suppose one is training a LINEAR ( $y=ax+b$  not polynomial  $a$  and  $b$  are constants) classifier on a data set that is a parabola. The resultant classifier will have no predictive power nor will it be able to properly map the training data. This is the result of under-fitting, or attempting to use a model which is too simple to describe a given set of data.

Understanding these two phenomena allows one to thread the needle and go into the space between the two extremes. It is in this gap where the model has predictive power in the validation set lies.



**Figure 1.3.** An example of a good fit to the data



**Figure 1.4.** An example of under-fitting

$$\begin{aligned}
 E\{(f_i - y_i)^2\} &= E\{(f_i - E\{y_i\} + E\{y_i\} - y_i)^2\} \\
 &= E\{(f_i - E\{y_i\})^2\} + E\{(E\{y_i\} - y_i)^2\} + 2E\{(E\{y_i\} - y_i)(f_i - E\{y_i\})\} \\
 &= \text{bias}^2 + \text{Var}\{y_i\} + 2(E\{f_i E\{y_i\}\} - E\{E\{y_i\}^2\} - E\{y_i f_i\} + E\{y_i E\{y_i\}\})
 \end{aligned}$$

Note:  $E\{f_i E\{y_i\}\} = f_i E\{y_i\}$  since  $f$  is deterministic and  $E\{E\{z\}\} = z$

:  $E\{E\{y_i\}^2\} = E\{y_i\}^2$  since  $E\{E\{z\}\} = z$

:  $E\{y_i f_i\} = f_i E\{y_i\}$

:  $E\{y_i E\{y_i\}\} = E\{y_i\}^2$

: Thus the last term in the expectation above cancels to zero.

$$E\{(f_i - y_i)^2\} = \text{bias}^2 + \text{Var}\{y_i\}$$

Thus the decomposition of the MSE in expectation becomes:

$$E\{(t_i - y_i)^2\} = \text{Var}\{\text{noise}\} + \text{bias}^2 + \text{Var}\{y_i\}$$

**Figure 1.5.** The proof for the mathematics as stolen from the internet

## 1.8 Proof about bias and variance

An important thing to note is that an over-fitted function which has perfect accuracy on the training set and poor accuracy in the validation set can not be used in describing the ideal function. The ideal function does not have perfect accuracy on the training data, instead it has optimal predictive accuracy, it maps all of the underlying mathematical terms in the true function and none of the noise.

The ideal function output is denoted as  $f_i$

while the data is denoted with  $y_i$ .

It should be noted that in the below picture. One can phrase over-fitting and under-fitting in the context of bias/variance. We desire our model  $m$  of the ideal function  $f$  do have as little bias (average manhattan or absolute value distance of a series of values of  $m$  corresponding to a single answer

$f_i$

from the answer  $f$  gives for a given point) from  $f$  at any given point, furthermore we want the variance of points with a given value in  $f$  to differ as little as possible in  $m$ . In practice however this is extremely difficult. One can minimize variance by making the model  $m$  equal to a constant over all space (minimize any variance) but this will have a huge problem with bias. On the opposite extreme one can fit a polynomial to the data hit every point but this extreme over-fitting will produce high variance as the model will have learned to map noise. Learning to minimize both is difficult but cross validation and regularization can significantly help. Note the proof is in figure 1.5 and is stolen from the internet.

Bayes risk: another way to say weighted loss is risk function, an example of a risk function is square loss as defined above. The bayes risk is in the equations in figure 1.5 but in a rather subtle way. The equations in 1.5 refer to  $\text{bias}^2 + \text{variance}(\text{data}) + (\text{variance of the noise})$  this term is the inherent difficulty of the problem and is called the bayes risk, note this is not dependent on hypothesis.

## 1.9 Cross validation

Cross validation is a simple technique for determining when to stop increasing the complexity of the model. It involves splitting the training data into two parts, one which is used to train the model and another which is used to validate it. Essentially the model is trained on a series of levels of complexity, at each level performance on the training set and on the validation set are evaluated. Since more parameters are always being added the accuracy on the training set should always improve with added complexity however the added complexity should have a negative effect on the accuracy of the validation set. Once the increase in complexity fails to improve accuracy in the validation set (negative or zero derivative or even a very small positive derivative of the accuracy of the validation set with respect to model complexity) the model training is halted and an approximation of the ideal complexity for the model

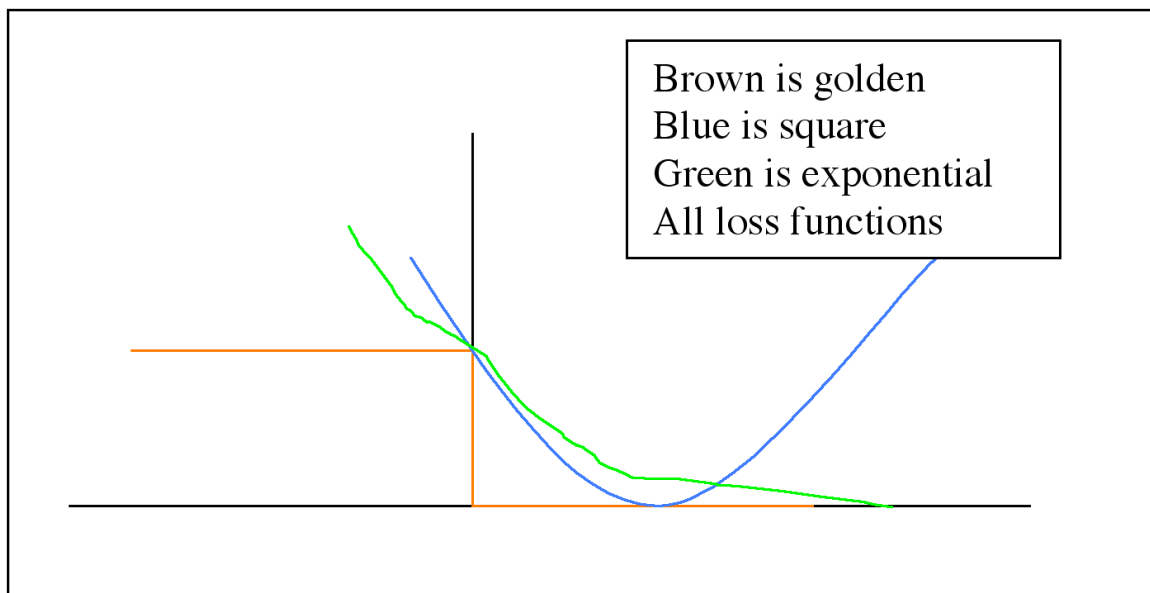
is determined. Adding complexity can take many forms and is model dependent, in neural networks adding hidden layer or hidden neurons could be a way of increasing complexity.

## 1.10 Regularization

Regularization is a means of controlling the variability of models by controlling their weights. These methods (we used one of these in homework 3) work by penalizing weight vectors which are inappropriate (inappropriate often means having too big weights). A simple version of this is taking the dot product of the weights with THEMSELVES. All this means is the bigger the weights (or bigger negative) the more the model is penalized. Methods like this penalize complexity and essentially force down variance at the expense of bias. Too high a constant in front of the regularization term will force the weights to all take on zero as their value

## 1.11 Loss functions

when evaluating models the term margin is often used. Margin is equal to actual\*(predicted). In the context of classification the value for actual (desired result) is 1 or -1. If both terms are negative a positive result is produced just as if both terms are positive. Only when a disagreement occurs does a negative margin occur. There are many ways of evaluating how bad a point is or how good a model is. One could say a model which get 95 percent correct get a score of .95 (gold standard) but there are many other options (note the gold standard is not differentiable, and hence substitutes are used such as the logistic). One could penalize the square of the amount the model is off. This method is not used in CLASSIFICATION because one pays a penalty even when a point is properly classified. support vector machines minimize hinge loss or maximize the minimum margin. when minimizing the exponential loss boosting is applied (this is when classifiers are continually retrained using extra copies of the data they classified incorrectly during the last iteration. DEPENDING ON WHAT KIND OF LOSS ONE WANTS TO MINIMIZE DIFFERENT LOSS FUNCTIONS OR MACHINE LEARNING METHODS ARE USED



**Figure 1.6.** The proof for the mathematics as stolen from the internet