Lecture 3.

Prof. Alan Yuille

Summer 2014

Outline

1. Precision and Recall Curves. Receiver Operating Characteristic Curves (ROC). What to do if we do not fix the loss function?
2. The Curse of Dimensionality. Why human intuitions are missleading in high-dimensional spaces? Why we rarely have enough dataset in high dimensional spaces, and what we can do about it.
3. The Bias-Variance Dilemma. The classic statistical perspective on generalization. How to use cross-validation to check for generalization.

1 Precision/Recall and ROC curves

What if we do not know the loss function? Or if we want a more "sophisticated" decision process? For example, we do not want to diagnose patients as "cancer" or "not-cancer". Instead we want to separate them into groups for further testing.

The basic idea is that decision processes can often be characterized by a function $f(x)$ and a threshold T. This gives rise to a one-dimensional family of decision rules $\alpha_T(x)$, where $\alpha_T(x) = 1$ if $f(x) > T$ and $\alpha_T(x) = -1$ otherwise. By altering the threshold T we change the decision rule and, alter the number of true positives and false positives. For example, suppose the decision function $f(x)$ is the log-likelihood ratio $\log \frac{p(x|y=1)}{p(x|y=-1)}$ then altering T corresponds to changing the prior $p(y)$ and the loss function $L(.,.)$. We can summarize the performance of this family of decision rules $\alpha_T(x)$ by plotting how performance (different for Precision/Recall and ROC) change with T.

Examples: Detecting Cats in the Pascal Object Detection Challenge. The data consists of $20,000$ images. There are roughly $1,000$ cats. The cats occur in a range of sizes and can occur anywhere in an image. There can be one or more cats in an image. The positions of cats and specified by bounding boxes which surround them, see figure 1. So the task is to determine which of all the possible bounding boxes in the images contain cats. There are

roughly $1,000$ bounding boxes in an image. Hence there are $20,0000,0000$ possible bounding boxes of which only $1,000$ are cats. Hence there are far more "backgrounds/negatives" (non-cats) than "targets/positives" (cats).

1.1 Precision and Recall

Figure 1: The positions of the targets (cats) are specified by bounding boxes. Images can contain one target, several targets, or none.

The Precision/Recall curve is motivated by detecting "targets/positives" in the presence of a much larger number of "backgrounds/negatives". Let the dataset $\mathcal{X} = \{(x_i, y_i) : i = 1, ..., N\}$. Suppose this has n_1 targets and n_2 backgrounds: $n_1 = \sum_{i=1}^{N} I(y_i = 1)$ and $n_2 = \sum_{i=1}^{N} I(y_i = -1)$. Note that $I(.)$ is the indicator function, i.e., $I(y = 1) = 1$ if $y = 1$ and $I(y = 1) = 0$ if $y \neq 1$.

There is a one-dimensional parameterized decision functions $f(x)$ and decision rules $\alpha_T(x)$. The decision rule is of the form

$$\alpha_T(x) = 1, \text{ if } f(x) > T \text{ and } \alpha_T(x) = -1, \text{ if } f(x) < T$$

Define

$$m_1^T = \text{true positives} = \text{number of targets (cats) detected}$$

$$m_2^T = \text{false positives} = \text{number of backgrounds (non-cats) detected}$$

I.e.

$$m_1^T = \sum_{i=1}^{N} I(\alpha_T(x_i) = 1) I(y_i = 1)$$

$$m_2^T = \sum_{i=1}^{N} I(\alpha_T(x_i) = 1) I(y_i = -1)$$

Precision at threshold T is $p(y = 1|\hat{y} = 1) = \frac{m_1^T}{m_1^T + m_2^T} = \frac{\text{No. of true positives}}{\text{No. of true positives + No. of false positives}}$.

Recall at threshold T is $p(\hat{y} = 1|y = 1) = \frac{m_1^T}{n_1} = \frac{\text{No. of true positives}}{\text{No. total targets}}$.

The Precision-Recall curve has characteristic shape shown in figure 2. The idea is that at large threshold T the detector will only detect the targets (if the detector is well designed) but will probably only detect a few of them (because a high threshold means that most of then will be rejected). But at low threshold, we will probably detect most of the targets but also (incorrectly) a large number of the distactors. In summary, the precision will tend to decrease and the recall will increase as we reduce the size of the threshold T.

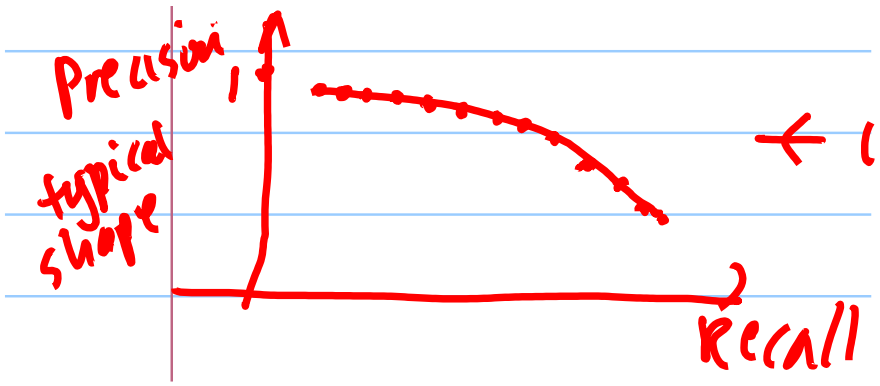

Figure 2: The precision recall curve.

1.2 Receiver Operating Characteristic Curve

The ROC curve is used in situations where the number of targets and backgrounds is roughly balanced. Unlike the previous section, we do not assume that the number of distactors is much bigger than the number of targets. It was originally motivated as a way of characterizing the properties of a physical device for detecting targets, e.g., a radar system which has parameters to be tuned. Hence the device outputs a function $f(x)$ of the input x and a decision $\alpha_T(x)$ is made by selecting a threshold T (i.e. T is a tuning parameter, of course most devices also have other tuning parameters).

The ROC curve usually, but not necessarily, assumes that the decision rule $\alpha_T(.)$ is specified by the log-likelihood ratio test $\log \frac{P(x|y=1)}{P(x|y=-1)}$ with threshold T. The ROC curve plots the true positive ratio as a function of the false positive ratio as the threshold T varies, see figure 3. The true positive ratio is the ratio of No. of true positives to No. of total positives, or $p(\hat{y} = 1|y = 1)$. And the false positive ratio is the ratio of No. of false positives to No. of total negatives, or $p(\hat{y} = 1|y = -1)$.

Here $\hat{y}$ is the decision made by the decision rule $\hat{\alpha}_T(x)$. At threshold T, $\hat{y}(x) = 1$, if $\log \frac{p(x|y=1)}{p(x|y=-1)} > T$, then

$$p(\hat{y} = 1 \mid y) = \sum_x p(\hat{y} \mid x) p(x \mid y)$$

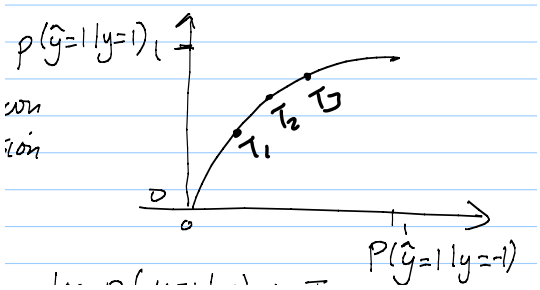

Figure 3: ROC curve. True positives and false negatives as a function of the decision threshold.

Now:

$$\begin{cases} p(\hat{y}=1 \mid x)=1, & \text{if } \log \frac{p(x|y=1)}{p(x|y=-1)} > T \\ p(\hat{y}=1 \mid x)=0, & \text{otherwise.} \end{cases}$$

$$\begin{cases} p(\hat{y}=-1 \mid x)=1, & \text{if } \log \frac{p(x|y=1)}{p(x|y=-1)} < T \\ p(\hat{y}=-1 \mid x)=0, & \text{otherwise.} \end{cases}$$

Hence:

$$p(\hat{y}=1|y=1) = \sum_{x:\ \log \frac{p(x|y=1)}{p(x|y=-1)}>T} p(x|y=1)$$

$$p(\hat{y}=1|y=-1) = \sum_{x:\ \log \frac{p(x|y=1)}{p(x|y=-1)}>T} p(x|y=-1)$$

$$p(\hat{y}=1 \mid y) = \sum_x p(\hat{y}=1 \mid x) p(x \mid y) = \sum_{x:\ \log \frac{p(x|y=1)}{p(x|y=-1)}>T} p(x|y)$$

The ROC curve plots the number of true positives as a function of the false positives. Each value of T gives a point on this curve. As the threshold becomes very large, i.e. $T \to \infty$, the proportions of true positives and false positives tend to zero (because the threshold is so high that the decision rule decides that there are no positives). Similarly as the threshold becomes very low, i.e. $T \to -\infty$, then the proportion of true and false positives become equal to one (because the decision rule decides that everything is positive). The Bayes decision is a specific point on the curve at threshold T^* (determined by the prior and the loss function).

ROC curves are used in Signal Detection Theory to characterize properties of the human perceptual system – e.g., our ability to hear and see. They have several interesting properties. For example suppose a human subject is given two stimuli x_1 and x_2 and told to decide which is the target and which is the distactor (i.e. the human knows there is one

of each), then the best achievable performance is characterized by the area under the ROC curve (check, this is the right ROC!).

2 The Curse of Dimensionality

Visual illustrations of Machine Learning and Bayes Decision Theory used in textbooks and lectures are often misleading. The figures are two-dimensional, but the data typically lies in a much higher-dimensional space. I.e. the observations x often lie in a high-dimensional space (e.g., 100 dimensions) and not in two-dimensions, as the examples in textbooks (or these lectures) would suggest.

Our geometrical intuition is based on living in a three-dimensional world and seeing two-dimensional figures. So our intuition for higher dimensional spaces is often wrong. This section will give some examples. More seriously, learning probability distributions in high-dimensional spaces requires an enormous amount of data. This is the *curse of dimensionality* (or one aspect of it anyway). This is a major problem for Machine Learning.

2.1 Intuitions of Geometry in High-Dimensions

Our geometric intuitions are often wrong in high-dimensions. It is good to be aware of this. Here are a few examples.

Example 1

Consider the volume of a sphere of radius $r = 1$ in D dimension. What fraction of its volume lies in the region between $1 - \epsilon < r < 1$?

To investigate this, we calculate the volume $V_d(r)$ of a sphere of radius r in d-dimensional space. This scales as r^d (we use K_R to specify the constant). Then we do a Taylor series expansion of $V_d(r)$ to estimate the volume in the small ring $1 - \epsilon \leq r \leq 1$. This gives

$$V_d(r) = K_d r^d \qquad \frac{V_d(1) - V_d(1-\epsilon)}{V_d(1)} = 1 - (1-\epsilon)^D$$

For larger d, the fraction of the volume in this ring tends to 1 even for small ϵ (because $(1-\epsilon)^d$ tends to 0 exponentially fast as d increases). So almost all the volume of the sphere is very close to the boundary! See figure 4.

Example 2

Consider the behaviour of a zero-mean Gaussian distribution in high-dimensions, see figure 5.

In one-dimension the Gaussian is $p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-x^2}{2\sigma^2}$ and its probability mass is peaked near $x = 0$. In two-dimensions, we express a Gaussian (rotationally symmetric) as $p(x_1, x_2) = \frac{1}{2\pi\sigma^2} \exp \frac{-(x_1^2+x_2^2)}{2\sigma^2}$. This can be re-expressed in polar coordinates, setting $r = \sqrt{x_1^2 + x_2^2}$, to give a distribution $p(x_1, x_2) = p(\theta)p(r)$ where $p(\theta) = 1/(2\pi)$ and

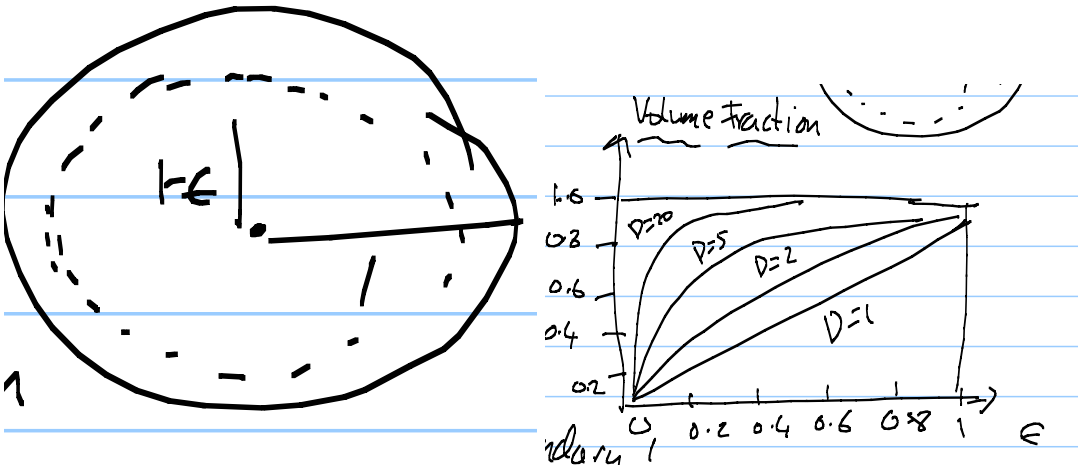

Figure 4: The volume of a sphere lies (left) almost entirely at the boundary in high dimensional spaces (right).

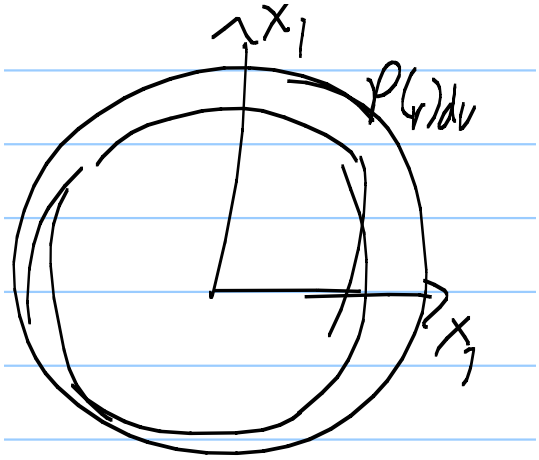

Figure 5: Where does the probability mass of a Gaussian? Near the center for low dimensions. But it moves out from the center as the dimension increases.

$p(r) = \frac{r}{\sigma^2} \exp \frac{-r^2}{2\sigma^2}$. Note that this takes value 0 at the center (where $r = 0$).

In d-dimensions, we can also express a (rotationally symmetric) Gaussian distribution is terms of a distribution $p(r)$ which is of form: $p(r) = \frac{r^{d-1}}{k_d} \exp \frac{-r^2}{2\sigma^2}$, where k_d is a constant. This shows that the probability mass of the Gaussian moves further away from the origin as d increases. Here we define the probability mass to be the amount of probability in an infinitesimal ring of size r, see figure 5.

So in high dimensions, most of the probability mass of the Gaussians is concentrate on a thin shell away from the center of the Gaussian. See figure 6.

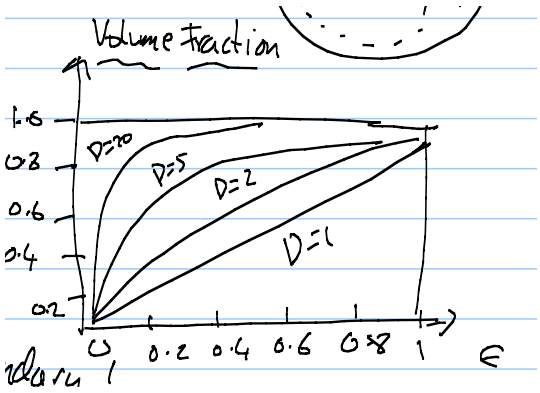

Figure 6: For higher dimensions most of the mass of a Gaussian is clustered on a thin shell away from the center.

2.2 Learning distributions and decision rules is hard in high-dimensions

Example 1

Consider a Gaussian distribution in d dimensions. It is parameterized by its mean μ, with d degrees of freedom, and its covariance Σ with $\frac{d(d-1)}{2}$ degrees of freedom. This gives a total of $\frac{d(d+1)}{2}$ parameters which need to be estimated/learnt.

This will require $k \times \frac{d(d+1)}{2}$ data examples in order to estimate the parameters accurately (i..e. dataset $\mathcal{X}$ must be roughly this size). The values of k is uncertain – $k = 5$, $k = 10$, bigger values of k will give more accurate estimates of the parameters (see later section of this lecture).

In this example, the amount of data required grows quadratically with the number of dimensions d. This implies that learning this distribution in 100-dimensional space will require of the order of $5 \times 10,000$ training examples.

Example 2

Consider learning a non-parametric distribution, like a histogram. Suppose the histogram has B bins per dimension, see figure 7. This has B bins in one-dimension, B^2 bins in two-dimensions, and B^d bins in d-dimensions. We will need $k \times B^d$ pieces of data in d dimensions.

So the amount of data required by this histogram model grows exponentially fast as d increases. This rapidly becomes impractical. Suppose the number of bins is $B = 10$. Then even in ten-dimensions, $d = 10$, we need $k \times 10^{10}$ data examples.

Example 3

Suppose we want a decision rule that corresponds to a separating hyperplane – e.g., $\alpha(\vec{x}) = 1$ if $\vec{a} \cdot \vec{x} + b > 0$ and $\alpha(\vec{x}) = -1$ otherwise. Then we need to estimate a $(d+1)$-dimensional vector $\vec{a}, b$. (This ignores the scale issue). Hence we need $k \times (d+1)$ data examples. This grows linearly with the dimension of the space.

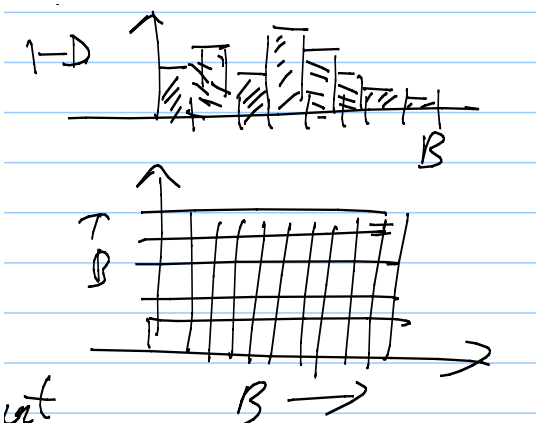

Figure 7: B bins in 1-dimension. B^2 bins in two dimensions. B^D bins in D dimensions. Exponential growth in amount of data needed in high dimensions.

2.3 How to deal with the curse of dimensionality?

Firstly, we may get lucky and know that the data is generated by a parameterized model (like a Gaussian) and there is enough data to learn the parameters.

Secondly, it may be sufficient to learn a simple decision rule – like a separating hyperplane – which only involves a limited number of parameters which scales slowly (e.g., linearly) with the dimension.

Thirdly, we can apply *dimension-reduction techniques*. These assume that the data lies on some low-dimensional surface/manifold in the high dimensional space. See figure (8)

So, the effective dimension of the data may be the dimension of the surface/manifold, which may be a lot smaller than the dimension of the space. But how to find this surface?

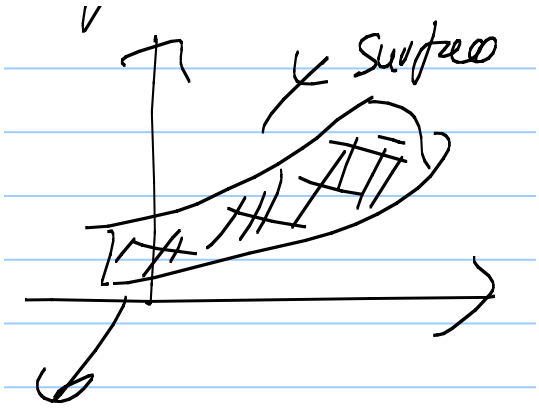

Figure 8: In practice, data in high dimensional spaces often lies in a low dimensional surface. If we know this surface, then we need much less data.

Later in this course we will describe several dimension reduction methods.

3 Estimators, Bias and Variance, and Cross-Validation

The *bias variance dilemma* was first articulated by Statisticians. It gives an alternative perspective on generalization (versus memorization) and leads to *cross-validation* which is a practical way to test for generalization.

First, we describe *statistical estimators*. These are used to estimate properties of a dataset. For example, we may want to estimate the mean and variance of a set of data, or fit a regression model to the data, or more generally estimate the parameters of a model. Bayes decision theory gives a criterion for selecting an "optimal estimator". But, historically, statistical estimators were developed by Statisticians who were sceptical about Bayesian methods (partly because of the difficulty of specifying priors) and so do not depend

3.1 Statistical Estimators

We want to estimate a continuous quantity θ. The estimator is based on a set $\mathcal{X} = \{x_1, ..., x_N\}$ of examples. It is assumed that these are i.i.d. samples from an (unknown) distribution $p(x|\theta)$. Hence the probability of the set $P(\mathcal{X}) = \prod_{i=1}^{N} p(x_i|\theta)$.

An *estimator* of θ is a function $g(\mathcal{X})$ which gives an estimate $\hat{\theta} = g(\mathcal{X})$ of θ. The estimate is a random variable that depends on the data set $\mathcal{X}$. If we have a different set of samples from $\mathcal{X}$ we would get a different estimate of θ (this will be important when we

consider generalization).

For example, to estimate the mean of X (random variable) from dataset $\mathcal{X}$ we set $g_1(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N} x_i$. To estimate the variance of X, we first estimate $\sum_x x^2 p(x)$ by the estimator $g_2(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N} x_i^2$ and then estimate the variance by $g_2(\mathcal{X}) - \{g_1(\mathcal{X})\}^2$.

3.2 Evaluating an Estimator

To evaluate an estimator $g(\mathcal{X})$ of θ we can measure how much it differs from θ. It is attractive to use a quadratic error (this simplifies the analysis) such as $(g(\mathcal{X}) - \theta)^2$, but this depends on the dataset $\mathcal{X}$. So we need to compute the *expected error* of the estimator with respect to $p(\mathcal{X})$:

$$r(g, \theta) = E_{\mathcal{X}}[g(\mathcal{X}) - \theta]^2 = \int (g(\mathcal{X}) - \theta)^2 P(\mathcal{X}) d\mathcal{X}.$$

We also compute the *bias* of the estimator:

$$b_\theta(g) = E_{\mathcal{X}}[g(\mathcal{X})] - \theta.$$

We say that $g(.)$ is an unbiased estimator of θ if $b_\theta(g) = 0$ for all θ. For example, $g_1(.)$ (previous section) is an unbiased estimator of the mean of X. This can be shown because

$$\sum_{\mathcal{X}} P(\mathcal{X}) g_1(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{\mathcal{X}} P(\mathcal{X}) x_i$$

$$= \frac{1}{N}\sum_{i=1}^{N} p(x_i) x_i = \frac{1}{N} \times N \sum_x x p(x) = \sum_x x p(x).$$

Now consider how the estimate depends on the dataset. There are many possible datasets (size N) that can be sampled from $p(x)$: $\mathcal{X}_1 = (x_1, ..., x_N)$, $\mathcal{X}_2 = (x_{N+1}, ..., x_{2N})$, $\mathcal{X}_3 = (x_{2N+1}, ..., x_{3N})$, and so on. From each dataset we get a different estimate $g(\mathcal{X}_1)$, $g(\mathcal{X}_2)$, ..., $g(\mathcal{X}_n)$ of θ. We can calculate the mean of these estimates – to get a better estimate of θ – and we can compute the variance of these estimates to see how the estimates differ from one dataset to another. If the variance is small, then this means that the estimates are the same for each dataset and hence we have good generalization.

To explore this, we compute the variance of the estimator $g_1(.)$ (the estimator of the mean described at the top of the page) with respect to $p(\mathcal{X})$ and show that it tends to 0 as the size N of the dataset tends to ∞.

$$Var_{\mathcal{X}}(g_1) = \frac{1}{N^2} Var_{\mathcal{X}}(\sum_{i=1}^{N} x_i) = \frac{1}{N^2}\sum_{i=1}^{N} Var_{x_i}(x_i) = \frac{\sigma^2}{N},$$

where σ^2 is the variance of X.

Hence the variance of the estimator (with respect to the dataset) tends to 0 as $N \to \infty$ (with fall-off rate $1/N$). Note this result is true for any linear estimator. Therefore the estimator becomes perfectly accurate as $N \to \infty$. We say that $g_1(.)$ is a consistent estimator.

Note that estimators do not have to be perfectly unbiased. The maximum likelihood estimator of the variance (see next lecture) is $(N-1)/N\sigma^2$ and hence is biased for finite N (here σ^2 is the true variance). But the amount of bias decreases rapidly for large N and we say the estimator is asymptotically unbiased.

3.3 The Bias Variance Dilemma

Assume the parameter θ of distribution $p(x|\theta)$ is a random variable. Then given parameter θ, the probability of dataset is

$$p(\mathcal{X}|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

$p(\theta)$ denotes the prior distribution of parameter θ. And $p(\theta|\mathcal{X})$ denotes the posterior distribution of parameter θ given the dataset. Let $g(\mathcal{X})$ be an estimator of θ. We have the following results.

Result 1. Given dataset $\mathcal{X}$

$$< (\theta - g(\mathcal{X}))^2 >_{p(\theta|\mathcal{X})} = < (\theta - < \theta >_{p(\theta|\mathcal{X})})^2 >_{p(\theta|\mathcal{X})} + (< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2,$$

where $< . >_{p(\theta|\mathcal{X})}$ is the expectation with respect to $p(\theta|\mathcal{X})$.

Proof.

$$(\theta - g(\mathcal{X}))^2 = (\theta - < \theta >_{p(\theta|\mathcal{X})} + < \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2 = (\theta - < \theta >_{p(\theta|\mathcal{X})})^2 + (< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2$$

$$-2(\theta - < \theta >_{p(\theta|\mathcal{X})})(< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X})).$$

Now take the expectation with respect to $p(\theta|\mathcal{X})$. The quantity $(< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))$ is independent of θ, the expectation of $(\theta - < \theta >_{p(\theta|\mathcal{X})})$ is 0. The result follows.

This result states that the expected error (w.r.t. $p(\theta|\mathcal{X})$) has two terms. The first term is the inherent variance of the process which is independent of the estimator $g(.)$) and the second term is the squared error.

Next we study how the expectation of the squared error $(< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2$ depends on the particular dataset $\mathcal{X}_1, ..., \mathcal{X}_m$.

Result 2.

$$< (< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2 >_{p(\mathcal{X})} = (< \theta >_{p(\theta|\mathcal{X})} - < g(\mathcal{X}) >_{p(\mathcal{X})})^2 + < (g(\mathcal{X}) - < g(\mathcal{X}) >_{p(\mathcal{X})})^2 >_{p(\mathcal{X})} .$$

This expresses the error in terms of the expected squared bias and the variance of the estimator $g(.)$.

Proof.

$$(< \theta >_{p(\theta|\mathcal{X})} - g(\mathcal{X}))^2 = (< \theta >_{p(\theta|\mathcal{X})} - < g(\mathcal{X}) >_{p(\mathcal{X})} + < g(\mathcal{X}) >_{p(\mathcal{X})} - g(\mathcal{X}))^2 =$$

$$(< \theta >_{p(\theta|\mathcal{X})} - < g(\mathcal{X}) >_{p(\mathcal{X})})^2 + (< g(\mathcal{X}) >_{p(\mathcal{X})} - g(\mathcal{X}))^2$$
$$-2(< \theta >_{p(\theta|\mathcal{X})} - < g(\mathcal{X}) >_{p(\mathcal{X})})(< g(\mathcal{X}) >_{p(\mathcal{X})} - g(\mathcal{X}).$$

The result follows by taking the expectation with respect to $p(\mathcal{X})$.

What does this mean? The expected error is the sum of two terms. The first depends on the bias of the estimator and the second is its variance. To get good generalization we want the variance to be small. But we also want the bias to be small. In practice, there is a trade-off between the bias and the variance. A complex classifier can give a good fit to the dataset and hence have small bias. But it may have large variance because it tends to over-fit the data and so gives different results on different datasets.

Examples: The bias and variance dilemma is better illustrated using a slightly more complex setting where the dataset is $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^N$. Given any x, we are to predict y. Suppose the data is generated by $y = 2\sin(1 - 5x) + \epsilon$, where ϵ is a random sample from a zero mean Gaussian with variance 1. A more complex models gives better fit to the data (i.e. to underlying model), see figure 9. Hence we need to have a model which balances the bias and the variance.

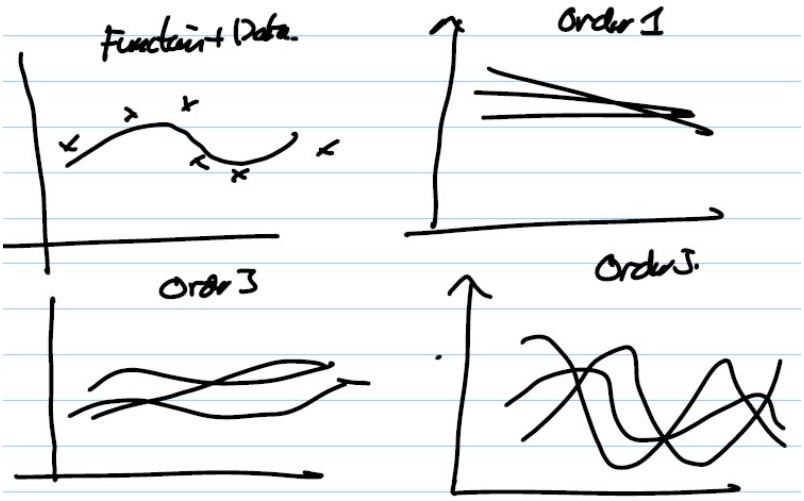

Figure 9: More complex models give better fits to the data – reduce the bias – but small changes in the dataset may lead to big variations – i.e. large variance.

3.4 Cross-validation and Regularization

Cross-validation divides the dataset into two part as training & validation set. Train models of different complexity and test their error on the validation set. As model complexity increases, the error on training set decreases. But the error on validation set decreases then increases, see figure 10.

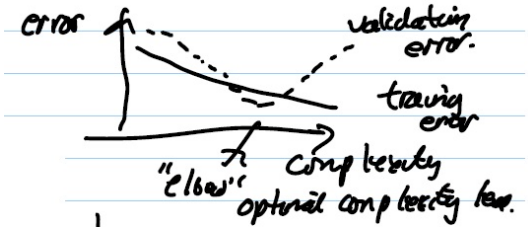

Figure 10: The training error always decreases as we increase the model complexity, but the error on the validation set decreases and then increases.

Regularization add a regularization term to penalize the model complexity. This gives an augmented error function $E' =$ error on data $+ \lambda \times$ model complexity. The λ is optimized using cross-validation. See also, structured risk minimization (Vapnik).