Lecture 2. Bayes Decision Theory

Prof. Alan Yuille

Spring 2014

Outline

1. Bayes Decision Theory
2. Empirical risk
3. Memorization & Generalization; Advanced topics

1 How to make decisions in the presence of uncertainty?

There are different examples of applications of the Bayes Decision Theory (BDT). BDT was motivated by problems arising during the 2^{nd} World War: Radar for aircraft detections, code-breaking and decryption. The task is to estimate the state but we only have a noisy, or corrupted, observation.

1.1 Likelihood Function

The likelihood is a function of the parameters of a statistical model. It can be a conditional probability. Given:
Observed Data $x \in \mathbf{X}$
State $y \in \mathbf{Y}$
$p(x|y)$ - conditional distribution – the probability of observing x if state is y.
$y \in \langle -1, 1 \rangle$ e.g., 1: Airplane / -1: Bird (No airplane)

Example:
$p(x|y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp^{\frac{-(x-\mu_y)^2}{2{\sigma_y}^2}}$, x is the length/brightness of the fish
mean=μ_y, variance=${\sigma_y}^2$. See figure (1).

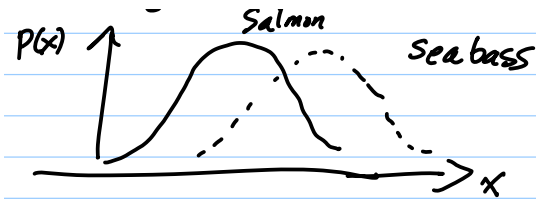

Figure 1: Gaussian distributions for the lengths of Salmon and Sea Bass.

1.2 The Maximum Likelihood (ML) Estimator

How to decide if a blip on the radar is Aircraft or Bird (No Airplane)? One way is to use the Maximum Likelihood (ML) estimator:
$\hat{y}_{ML}(x) = \arg\max_y p(x|y)$ i.e. $\forall y,\ p(x|\hat{y}_{ML}) \geq p(x|y)$
If $P(x|y=1) > P(x|y=-1)$
decide $\hat{y}_{ML}(x) = 1$, otherwise $\hat{y}_{ML}(x) = -1$
Equivalently,
decide $y = 1$ if $\log \frac{P(x|y=1)}{p(x|y=-1)} > 0$: *log-likelihood ratio* test.

The ML estimator seems a reasonable way to make a decision. But what if birds are much more likely than airplanes? Surely we should take the expected frequency, or *prior*, of birds and airplanes into account?

1.3 The Maximum a Posteriori (MAP) Estimator

The prior probability $p(y)$ is the probability of the state before we have observed it.(Priori means previous in Latin and posteriori means after). The prior would be the probability of having an airplane and the probability of not having it, without using the radar:
$p(y=1), p(y=-1)$

We combine the prior $p(y)$ with the likelihood $p(x|y)$ to obtain the *posterior* probability $p(y|x)$, which is the probability of the state y given (i.e. conditioned on) the observation x.

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

This is the Bayes Rule. It follows from the identity $p(x|y)p(y) = p(x,y) = p(y|x)p(x)$.
$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ is the probability of y conditioned on observation x.
If $p(y=1|x) > p(y=-1|x)$
then decide $y = 1$, otherwise decide $y = -1$
Maximum a Posteriori (MAP):

$$\hat{y}_{MAP}(x) = \arg\max_y p(y|x)$$

1.4 The Decision Rules and the Loss Function

What does it cost if you make the wrong decision?
i.e. suppose you decide $y = 1$, but the state is really $y = -1$,
i.e. you may pay a big penalty if you decide it is a bird when it is a plane.
(Pascal's wager: Bet on God – you may pay a big loss if God exists but you do not worship God.)

A *decision rule* $\alpha(.)$ takes input x and outputs a decision $\alpha(x)$. We will usually require that $\alpha(.)$ lies in a *class of decision rules* $\mathcal{A}$, i.e. $\alpha(.) \in \mathcal{A}$. $\mathcal{A}$ is sometimes called the *hypothesis class*. In Bayes Decision Theory there are usually no restrictions placed on $\mathcal{A}$ (i.e. all rules $\alpha(.)$ are allowed). In Machine Learning, we will usually put restrictions on $\mathcal{A}$ to ensure that we have have enough data to learn them (see later lectures).

The *loss function* $L(\alpha(x), y)$ is the cost you pay if you make decision $\alpha(x)$, but the true state is y.

Example: All wrong answers are penalized the same:
$L(\alpha(x), y) = 0, \ if \, \alpha(x) = y$ (correct decision)
$L(\alpha(x), y) = 1, \ if \, \alpha(x) \neq y$ (incorrect decision)

Alternatively, we can set $L(\alpha(x) = 1, y = -1) = 1$ and $L(\alpha(x) = -1, y = -1) = 100$. In other words, we can penalize *false negatives* ($\alpha(x) = -1$ and $y = 1$) much more than *false positives*. E.g., it is much worse to think that a radar blip is caused by a bird, if it is really caused by a plane, than the other way round.

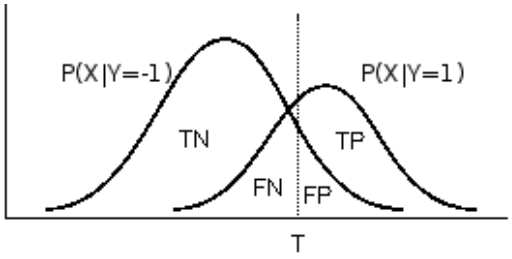

Figure 2: The impact of the threshold T on the number of true positives, false positives, true negatives and false negatives.

By thresholding the log-likelihood ratio T, we can vary the number of *false negatives* ($\alpha(x) = -1, y = 1$), *false positives* ($\alpha(x) = 1, y = -1$), *true negatives* ($\alpha(x) = y = -1$) and *true positives* ($\alpha(x) = y = 1$). Figure (2) shows an example of how the threshold T

modifies a Gaussian likelihood function whose log-likelihood ratio is used as the decision rule:
If $\log \frac{P(x|y=1)}{p(x|y=-1)} > T$, then $y = 1$, otherwise, $y = -1$.

Note that if $T = -\infty$, then we maximize the number of true positives at the cost of maximizing the number of false positives too, and having no true negatives.

1.5 The Risk and Bayes Decision Theory

To put everything together, we have:
likelihood function: $p(x|y) \quad x \in \mathbf{X}, y \in \mathbf{Y}$
prior: $p(y)$
decision rule: $\alpha(x) \quad \alpha(x) \in \mathbf{Y}$
loss function: $L(\alpha(x), y)$ cost of making decision $\alpha(x)$ when true state is y.

The risk function combines the loss function, the decision rule, and the probabilities. More precisely, the *risk* of a decision rule $\alpha(.)$ is the expected loss $L(.,.)$ with respect to the probabilities $p(.,.)$.

$$R(\alpha) = \sum_{x,y} L(\alpha(x), y)P(x, y)$$

(Note: if x takes continuous values (instead of discrete values) then we replace $\sum_{x,y}$ by $\sum_y \int dx$.)

According to *Bayes Decision Theory* one has to pick the decision rule $\hat{\alpha}$ which minimizes the risk.
$\hat{\alpha} = \arg\min_{\alpha \in \mathbf{A}} R(\alpha)$, i.e. $R(\hat{\alpha}) \geq R(\alpha) \quad \forall \alpha \in \mathcal{A}$ (set of all decision rules).

$\hat{\alpha}$ is the *Bayes Decision*
$R(\hat{\alpha})$ is the *Bayes Risk*.

1.6 MAP and ML as special cases of Bayes Decision Theory

We can re-express the Risk function as
$R(\alpha) = \sum_x \sum_y L(\alpha(x), y)p(x, y) = \sum_x P(x)\{\sum_y L(\alpha(x), y)p(y|x)\}$
Hence, for each x, the best decision is
$\alpha(x) = \arg\min_{\alpha(x)} \sum_y L(\alpha(x), y)p(y|x)$
Note the decision now depends on the posterior $p(y|x)$.

Suppose the loss function penalizes all errors equally:
$L(\alpha(x), y) = 1$, if $\alpha(x) \neq y$,
$L(\alpha(x), y) = 0$, if $\alpha(x) = y$.
$y \in \{-1, 1\}$
then $\sum_y L(\alpha(x), y)p(y|x) = P(y \neq \alpha(x)|x) = 1 - P(y = \alpha(x)|x)$,

hence, $\hat{\alpha}(x) = \arg\max_{\alpha(x)} p(y = \alpha(x)|x)$, which is the MAP estimate.
If, in addition, $p(y = 1) = p(y = -1)$,
then $\hat{\alpha}(x) = \arg\max_{\alpha(x)} p(x|y = \alpha(x))$, which is the ML estimate.

In summary, Bayes decision is MAP estimator if the loss function penalizes all errors by the same amount. If the loss function penalizes all the errors by the same amount *and* the prior is uniform (i.e. $p(y = 1) = p(y = -1)$), then the Bayes decision is the ML estimator.

1.7 The log-likelihood ratio and thresholds

For the binary classification case – $y \in \{\pm 1\}$ – the decision depends on the *log-likelihood ratio* $\log \frac{p(x|y=1)}{p(x|y=-1)}$ and on a threshold T. This threshold is determined by the prior and the loss function.

To understand this, we can express the loss function as a 2×2 matrix with components $\{L_{a,i} : \ a = 1, 2 \ i = 1, 2\}$ where $L_{a,i} = L(\alpha(x) = a, y = i)$.

The decision rule, for input x, requires making the decision $\hat{\alpha}(x)$ which minimizes the expected loss. The expected loss for decision $\alpha(x) = 1$ is given by $L_{1,1}p(y = 1|x) + L_{1,-1}p(y = -1|x))$. The expected loss for decision $\alpha(x) = -1$ is $L_{-1,1}p(y = 1|x) + L_{-1,-1}p(y = -1|x))$.

Hence,
if $L_{1,1}p(y = 1|x) + L_{1,-1}p(y = -1|x) < L_{-1,1}p(y = 1|x) + L_{-1,-1}p(y = -1|x)$,
then $\hat{\alpha}(x) = -1$
otherwise $\hat{\alpha}(x) = 1$ (ignore special case with equality)

This reduces to (after algebra).
$\hat{\alpha}(x) = 1$ if $\frac{p(y=1|x)}{p(y=-1|x)} > T_L$, $\hat{\alpha}(x) = -1$ otherwise
where $T_L = \frac{L_{1,-1} - L_{-1,-1}}{L_{-1,1} - L_{1,1}}$.

Now express $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ (Bayes rule) which implies that

$$\log \frac{p(y = 1|x)}{p(y = -1|x)} = \log \frac{p(x|y = 1)}{p(x|y = -1)} + \log \frac{p(y = 1)}{p(y = -1)},$$

which combine the log-likelihood ratio with the log ratio of the prior.

Putting all this together gives a decision rule: $\hat{\alpha}(x) = 1$ provided:

$$\log \frac{p(x|y = 1)}{p(x|y = -1)} > T_L + T_P,$$

where $T_P = -\log \frac{p(y=1)}{p(y=-1)}$.

In summary, the Bayes decision reduces to thresholding the log-likelihood ratio by a threshold $T = T_L + T_P$ which depends on the loss function (T_L) and the prior (T_P). Hence

the *form of the decision rule* (i.e. its dependence on x), and hence the form of the decision boundary, is specified by the likelihood function. The loss function and prior determine the precise position of the decision boundary (but not its form).

1.8 Examples of Bayes Decisions

Let $p(x|y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp^{-\frac{(x-\mu_y)^2}{2\sigma_y{}^2}} \quad y \in \{-1, 1\}, \quad p(y) = 1/2$
$L(\alpha(x), y) = 1$, if $\alpha(x) = y$
$L(\alpha(x), y) = 0$, if $\alpha(x) \neq y$
Bayes Rule
$\alpha(x) = \arg\min_{y \in \{-1,1\}} (x - \mu_y)^2$

The decision boundary is at $x_{ML} = 1/2(\mu_{-1} + \mu_1)$. Decision rule $\hat{y}_{ML}(x) = 1$, if $x > x_{ML}$, otherwise $\hat{y}_{ML}(x) = -1$. See figure (3).

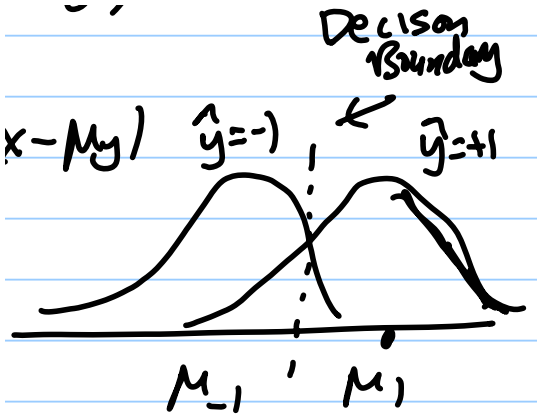

Figure 3: Decision boundary for Salmon and Sea Bass.

Suppose $\vec{x}$ is a vector in two dimensions
$p(\vec{x}|y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}|\vec{x}-\mu_y|^2}$
The decision boundary is a line specified by the points $\vec{x}$ such that
$2\vec{x} \cdot (\mu_1 - \mu_{-1}) = |\mu_1|^2 - |\mu_2|^2$,
which follows from solving $(\vec{x} - \mu_1)^2 = (\vec{x} - \mu_{-1})^2$.
The decision rule classifies all points $\vec{x}$ above the line (i.e. $2\vec{x} \cdot (\mu_1 - \mu_{-1}) > |\mu_1|^2 - |\mu_2|^2$) as $\hat{y} = 1$. All points below the line are classified as $\hat{y} = -1$. See figure (4).

Now suppose that the distributions are Gaussian but with different covariances $\sum_y$:
$p(\vec{x}|y) = \frac{1}{2\pi|\sum_y|^{1/2}} \exp -\frac{1}{2}(x - \mu_y)^T \sum_y^{-1} (x - \mu_y)$.

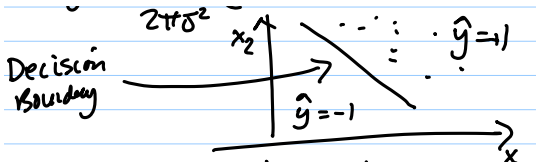

Figure 4: Decision boundary is a line if the distributions $P(\vec{x}|y)$ are both Gaussians with the same covariance σ^2 (times identity matrix).

Then the decision boundary is defined by a curved surface which obeys: $\frac{1}{2}(x-\mu_1)^T \sum_1^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_{-1})^T \sum_{-1}^{-1}(x-\mu_{-1}) + \frac{1}{2}\log|\sum_1| - \frac{1}{2}\log|\sum_{-1}| = 0$. See figure (5).

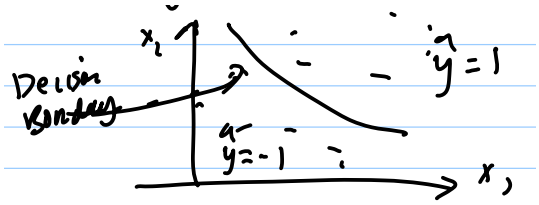

Figure 5: Decision boundary is a curve (a quadratic) if the distributions $P(\vec{x}|y)$ are both Gaussians with different covariances.

1.9 Bayes Decision Theory: multi-class and regression

Bayes Decision Theory also applies when y is not a binary variable, e.g. y can take M discrete values or y can be continuous valued. In this course, usually
$y \in \{-1, 1\}$: classification
$y \in \{1, 2, ..., M\}$: multi-class classification
$y \in \mathbb{R}^1$: regression

Bayes decision theory is the ideal decision procedure – but in practice it can be difficult to apply because of the limitations described in the next subsection.

Note, Bayes Decision Theory (and Machine Learning) can also be used if $\vec{y}$ is a vector-valued. I.e. if $\vec{y}(= y_1, y_2, ..., y_i, ..., y_M)$ where each y_i is binary-, discrete-, or continuous-valued (as above). But we will not discuss these situations in this course.

1.10 The Strengths and Weaknesses of Bayes Decision Theory

Bayes Decision theory is applicable, and the Bayes Risk is the best you can do, provided:
(a) you know $p(x|y), p(y)$, $L(.,.)$
(b) you can compute $\hat{\alpha} = \arg\min_{\alpha} R(\alpha)$

(c) you can afford the losses (e.g., gambling, poker)
(d) the future is the same as the past. I.e. the data samples $\{(x_i, y_i) : i = 1, ..., n\}$ are representative of an underlying unchanging (with time) distribution $P(x, y)$ of samples (more on this later).

Bayes Decision theory is not applicable if:
(i) if you are playing a game against an intelligent opponent – they will predict your "best move" and counter it. It is best to make your decision slightly random to keep your opponent guessing. This leads to Game Theory.
(ii) if any of the assumption (a), (b), (c), (d) are wrong.

Comments:
Bayes Decision Theory has been proposed as a rational way for humans to make decisions. Cognitive scientists perform experiments to see if humans really do make decisions by minimizing a Risk function. Tversky and Kahneman concluded that humans do not and instead rely on heuristics and biases. They proposed *Prospect Theory* as an alternative. But this is disputed and humans may use Bayes Decision theory (without knowing it) in certain types of situations. For example, a bookmaker (who takes bets on the outcome of horse races) would rapidly go bankrupt if he did not use Bayes Decision theory.

2 Empirical risk

The fundamental problem with Bayes Decision Theory (BDT) and Machine Learing (ML) is that we usually do not know the distribution $p(x|y)p(y)$. Instead we have a set of labeled examples $\mathcal{X}_N = (x_1, y_1), ..., (x_N, y_N)$

We define the *empirical risk* $R_{emp}(\alpha : \mathcal{X}_N) = \frac{1}{N} \sum_{i=1}^{N} L(\alpha(x_i), y_i)$
This is the risk of using decision rule $\alpha(x_i)$ averaged over the labeled examples $\mathcal{X}_N$.

A fundamental assumption of BDT and ML is that the observed data $\mathcal{X}$ consists of independent identically distributed i.i.d samples from an (unknown) distribution $p(x, y) = p(x|y)p(y)$

Then, as $N \to \infty, R_{emp}(\alpha : \mathcal{X}_N) \to R(\alpha)$ (in probability)
Hence we recover the risk if we have enough data – i.e. if N is big enough. But how big is enough? (We will discuss this issue in the advanced material of this lecture, and later in the course.).

This suggests several strategies. The first strategy consist in learning the probabilities:
Use $\mathcal{X} = \{(x_i, y_i) : i = 1 : N\}$
to learn the distributions $p(x|y)$ & $p(y)$
then apply Bayes Decision Theory.
i.e. estimate $\hat{\alpha}(x) = \arg\min_{\alpha(x)} \sum_y (\alpha(x), y)p(y|x)$
Note: this is the classic Statistics Strategy.

The second strategy is the discriminative approach: Estimate the best decision rule $\hat{\alpha}(x)$ directly from the empirical risk $R_{emp}(\alpha : \mathcal{X}_N)$

Note: this is the classic Machine Learning Strategy.
Motivation: why estimate the probabilities when you only care about the decision? Some probability distributions – e.g., Gaussians – are notoriously non-robust. This means if the data is contaminated, or does not follow a Gaussian model, then estimates of the parameters of the Gaussian (mean and variance) can be seriously wrong (see Huber: Robust Statistics) which means that the Bayes rule will not be good.

So why try to learn/estimate the parameters of two Gaussians (for $p(x|y=1), p(x|y=-1)$) if we only want to learn the decision boundary? Particularly if the estimates of the Gaussians may be corrupted by data that is a long way from the decision boundary? (Note this is the machine learning argument, there are other reasons why learning the distributions may be better – they may be more insightful, they might enable you to transfer your knowledge from one domain to another).

There is also a third strategy – learn the posterior distribution $p(y|x)$ directly.
This is called *regression* in Statistics (and has a history of over 200 years, starting with Gauss). This has close relations to some forms of Machine Learning, because ML researchers often want to give a confidence to their decisions.

2.1 Generative Methods and Inverse Inference

The first approach is sometimes called *generative* or *inverse inference*. It is called generative because if you know the distributions $P(x|y)$ and $p(y)$ then you can sample from them (stochastically) to generate samples $x_1, x_2, ..., x_m$ which, ideally, should look similar to the observed data. It is called inverse inference because it means that you are inverting the generative process to estimate which state (i.e. $y = 1$ or $y = -1$ in the classification case) is most likely to have generated the data.

A disadvantage of the generative approach is that the space $\mathbf{X}$ of observations is typically much bigger than the space $\mathbf{Y}$ of the atates. This makes it harder to learn $p(x|y)$ than $p(y|x)$ (the smaller the space, the easier it is to learn a distribution over it).

This disadvantage is enhanced because in many Machine Learning applications it is not clear how to *represent* the observations. Consider the example of discriminating sea bass and salmon. There are many ways to represent these fish. How do we know to represent them in terms of their length and brightness? There are many other properties/features that we could consider. The generative approach, taken to extremes, would require a model that could generate the full visual appearance of the fish (e.g., a computer graphics model). This is often unnecessary and instead a limited number of features (length and brightness) seem sufficient. But this highlights an important problem – how to represent the observations (e.g., by selecting features) which can be used as input to either generative, or discriminative, or regression models. We will return to this issue later in the course.

But generative models have several advantages. Firstly, we can generate stochastic samples from them which enables us to gain intuition about the model and see what

aspects of the data they capture (this is particularly useful for vision). Secondly, we can evaluate how well the models generate the observed data which helps us select between different models (see later lectures). Thirdly, and most importantly, generative models allow us to transfer knowledge from one domain to another (in principle, sometimes very hard in practice). For example, suppose have a generative model of a cow, then this makes it easier to learn a generative model of a yak (by exploiting the similarities between these two animals). Cognitive Science researchers argue that generative models enable us to account for an important aspect of human learning – namely the ability of humans to learn from a very small number of examples. This is only possible if humans have sophisticated representations and prior models (e.g., like a detailed model of a cow in terms of its parts) which can be adapted to new data and new problems. But this is beyond the scope of this course.

To re-emphasize this: learning $p(y|x)$ is often easier than learning the distributions $p(x|y)$ & $p(y)$ and then using Bayes rule to compute $p(y|x)$. The reason is that the space of decisions $\mathbf{Y}$ is often much smaller than the space of observations $\mathbf{X}$ (recall $y \in \mathbf{Y}$ and $x \in \mathbf{X}$). For example, suppose $y \in \{Face, NonFace\}$ and x is an image. It is much easier to learn the conditional distributions $p(y = Face|x), p(y = NonFace|x)$ than learn the models $p(x|Face)$ &$P(x|NonFace)$, which is like making a computer graphics systems to generate all possible Face and Non-Face images.

3 Memorization & Generalization

3.1 Finiteness of Data

Suppose we have $R_{emp}(\alpha : \mathcal{X}_N)$ with N samples (i.e. $N = |\mathcal{X}_n|$). We want to learn a rule $\alpha(x)$ that will give good results for data that you have not seen yet, but which comes from the same source as your samples.

Assume $\mathcal{X}_n = \{(x_i, y_i) : i = 1 : N\}$ are samples from an unknown distribution $p(x, y)$. Want to learn a decision rule using the observed samples $\mathcal{X}_N$ that will also apply to other samples from $p(x, y)$.

Probably Approximately Correct (PAC) – Vapnik, Valiant:
You do not want a rule that works perfectly on $\{(x_i, y_i) : i = 1 : N\}$ but fails to *generalize* to new (unknown) samples from $p(x, y)$. I.e. you want the rule to work for new samples $(x_{N+1}, y_{N+1}), (x_{N+2}, y_{N+2}), ...$ from $p(x, y)$.

3.2 Comparision

3.2.1 Memorization

This is like having a Decision Rule: $\hat{\alpha} = \arg_\alpha \min R_{emp}(\alpha : \mathcal{X}_N)$, where $R_{emp}(\hat{\alpha} : \mathcal{X}_N)$ is small on the training data $\mathcal{X}_N$, but where the Bayes risk $R(\alpha)$ may be big.
This is like an intelligent parrot that can memorize everything that the professor says, but

does not really understand the material. For example, the professor says that $2*2=4$ and $3*3=6$ – but does not say what $*$ means – and then asks "what is $4*4$"?. The parrot says "I don't know, you never told me". The student who has understood the material says $4*4=8$, guessing that $*$ probably means addition $+$. Note that the student cannot really be sure that this is the correct answer – only that it is very probable (we will return to this point later).

3.2.2 Generalization

We want a decision rule $\hat{\alpha}$ so that $R_{emp}(\hat{\alpha}:\mathcal{X}_N)$ is small on the training data $\mathcal{X}_N$, but the risk $R(\hat{\alpha})$ is also small. This is like a student who realizes that the professor sometimes makes a mistake and so tolerates some mistakes in the training data but still understands the material. For example, the professor says $2.1*2.0=4$ and $1.8*3.1=6$ – so there are small mistakes in the data, but the student still realizes that $*=+$. Or the professor says $2*2=4$, $5*4=9$, $2*6=12$ and $3*7=5$ – the last example is an 'outlier' (a big mistake) and the student should reject it, and realize that $*=+$. Note: rejecting outliers is an important topic. Some models are very sensitive to outliers and other models are more robust (be careful of using Gaussians, look what happened to the financial industry in 2008)..

3.2.3 Cross-validation

In practice, we will check for generalization by cross-validation:
Use a *training set* $\mathcal{X}=\{(x_i,y_i):i=1:N\}$ to learn the rule $\hat{\alpha}$
Use a *testing set* $\mathcal{X}_{test}=\{(x_j,y_j):j=1:N\}$ to test the rule $\hat{\alpha}$
Choose $\hat{\alpha}$ so that $R_{emp}(\hat{\alpha}:\mathcal{X}_N)$ is small on both the training set and test set. (More about this in later lectures).

How? By restricting the set $\mathcal{A}$ of possible decision rules $\hat{\alpha}(.)$. If we allow very complex rules $\alpha(.)$, then we can obtain almost perfect results on the training dataset, but these will usually give poor results on the testing dataset (because we can find a rule which, by coincidence, performs well on the training data). But if we can find a simple rule $\alpha(.)$ that explains the training data, then it is more likely to perform well on the testing dataset. We will return to these issues later in the course. Sometime this will involve adding an extra "regularization term" $E_R(\alpha)$ to the empirical risk $R_{emp}(\alpha:\mathcal{X}_N)$. We will also discuss cross-validation in a later lecture from the perspective of regression.

3.3 Mathematics of Memorization and Generalization: Advanced Material

Here is an analysis of the difference between memorization and generalization. It is the simplest example that I know. The course will mention other analysis by Vapnik, and Smale. (Also put a discussion of PAC-Bayes?).

Single Decision Rule Case

First, suppose we are consider a single decision rule $\alpha(.)$ to classify input data x as a binary output $y \in \{0,1\}$ (so $\alpha(x) \in \{0,1\}$). The loss function $L(\alpha(x), y) \in \{0,1\}$. We have N samples $\mathcal{X}_N = \{(x_i, y_i) : i = 1, ..., N\}$. We assume that they are drawn from an unknown probability distribution $p(x, y)$.

The empirical risk $R_{emp}(\alpha : \mathcal{X}_N)$ and the risk $R(\alpha)$ are given by:

$$R_{emp}(\alpha : \mathcal{X}_N) = \frac{1}{N}\sum_{i=1}^{N} L(\alpha(x_i), y_i), \quad R(\alpha) = \sum_{x,y} L(\alpha(x), y)P(x, y). \tag{1}$$

The problem with learning is that we can measure $R_{emp}(\alpha : \mathcal{X}_N)$ for any decision rule $\alpha(.) \in \mathcal{A}$ but we cannot compute $R(\alpha)$ because we do not know the distribution $p(x, y)$. So a decision rule $\alpha(.)$ may look good on the data – i.e. $R_{emp}(\alpha : \mathcal{X}_N)$ can be very small – but it may work badly in general – i.e. $R(\alpha)$ may be big. So we need to know how $R_{emp}(\alpha : \mathcal{X}_N)$ relates to $R(\alpha)$.

By the law of large numbers, $R_{emp}(\alpha : \mathcal{X}_N) \mapsto R(\alpha)$ as $N \mapsto \infty$. Hence if N is big enough then the two risks will become very similar so that if $R_{emp}(\alpha : \mathcal{X}_N)$ is small then $R(\alpha)$ will also be small – and so the rule will *generalize* to new data drawn from $p(x, y)$. But the question is how big must N be before we can be almost certain that $R_{emp}(\alpha : \mathcal{X}_N) \approx R(\alpha)$?

There are standard theorems from large deviation theory (Chernoff, Sanov, Cremers) which can be used to give results like *Result 0*:

$$\Pr\{|R_{emp}(\alpha : \mathcal{X}_N) - R(\alpha)| > \epsilon\} < \exp\{-N\epsilon\}. \tag{2}$$

Suppose we require that $\exp\{-N\epsilon\} < \delta$ where δ is a small number. This is equivalent to requiring that $N > \frac{-\log \delta}{\epsilon}$ (note that $-\log \delta > 0$ if $0 < \delta < 1$). Then we can restate the result in equation (2) as *Result 1*:

$$\text{If } N > \frac{-\log \delta}{\epsilon} \text{ then with prob } > 1 - \delta \;\; |R_{emp}(\alpha : \mathcal{X}_N) - R(\alpha)| < \epsilon. \tag{3}$$

Result 1 is an example of a Probably Approximately Correct (PAC) theorem. With *probability* $> 1 - \delta$ we know that we can *approximate* $R(\alpha)$ by $P_{emp}(\alpha : \mathcal{X}_N)$ provided $N = |\mathcal{X}_N|$ is sufficiently big (as a function of δ and ϵ. The result shows that the number of examples we need increases with the greater precision we require (i.e. as ϵ gets smaller) and with the greater certainty that the result is correct (i.e. as δ tends to 0). Note that we can never be completely certain that $|R_{emp}(\alpha : \mathcal{X}_N) - R(\alpha)| < \epsilon$ because there is always a chance that our samples $\mathcal{X}_N = \{(x_1, y_1), ..., (x_N, y_N)\}$ are unrepresentative of $p(x, y)$. But this chance decreases exponentially with N, as shown in equation (2). So we can only be almost certain that $|R_{emp}(\alpha : \mathcal{X}_N) - R(\alpha)| < \epsilon$ and "almost" is quantified by δ.

Result 1 gives us a 'learnability condition' for a single classifier $\alpha(.)$. But, in practice, we have a hypothesis set $\mathcal{A}$ of many classifiers $\alpha^\nu : \nu \in \mathcal{A}$. We need to be almost sure

that $|R_{emp}(\alpha^\nu : \mathcal{X}_N) - R(\alpha^\nu)| < \epsilon$ for all $\nu \in \mathcal{A}$. The bigger the number $|\mathcal{A}|$ of classifiers then the larger the chance that one of them has an empirical risk that differs from its risk. (Note: here we consider the case when there are only a finite number of classifiers – we will return to the case where there are an infinite number of classifiers later).

Multiple Decision Rule Case.

To extend Result 1 to cases where there are a large number of classifiers we first recall Boole's inequality:

$$Pr(A^1 \text{ or } ... \text{ or} A^{|\mathcal{A}|}) \leq \sum_{\nu \in \mathcal{A}} Pr(A^\nu). \tag{4}$$

We set $Pr(A^\nu) = Pr\{|R_{emp}(\alpha^{nu} : \mathcal{A}) - R(\alpha^\nu)| > \epsilon\}$. Then, using Boole we can extend Result 1 to the following:

$$\Pr\{|R_{emp}(\alpha^\nu : \mathcal{X}_N) - R(\alpha^\nu)| > \epsilon \text{ for at least one } \nu \in \mathcal{A}\} < |\mathcal{A}| \exp\{-N\epsilon\}. \tag{5}$$

Using the same reasoning as before (to get from Result 0 to Result 1) we obtain *Result 2*:

$$\text{If } N > \frac{\log|\mathcal{A}| - \log\delta}{\epsilon} \text{ then with prob } > 1-\delta \;\; |R_{emp}(\alpha : \mathcal{X}_N) - R(\alpha)| < \epsilon \; \forall \nu \in \{1, ..., |H|\}. \tag{6}$$

Result 2 is a more realistic example of a (very simple) PAC theorem. It illustrates the key ideas of generalization. First, we can never be completely certain that we have generalized but we can become almost certain by requiring that δ be very small. Second, our empirical risk can never equal the risk exactly, but we can control the precision by making ϵ small. Third, the number of data samples we need N will grow with the number of hypotheses $|\mathcal{A}|$ (the number of different decision rules we consider), with the amount of certainty we require (δ) and with the degree of precision ϵ.

The classic mistake, if we only have a small amount of data, is that we allow ourselves to try using a very large number of hypotheses to explain it. The mathematics shows that with high probability we may find a classifier which has low empirical risk but which has big risk – then we overgeneralize. Recall that memorization means that $R_{emp}(\alpha : \mathcal{X}_N)$ is small and generalization means that $R(\alpha)$ is small.

What happens if the size of the hypothesis set is infinite? Then we need to approximate the hypothesis set by a finite set, so that each element of the hypothesis set is "close" to some member of the finite set. Then we can use results like Result 2. There are roughly speaking two ways to obtain a finite hypothesis set in this way. The first is by use of the *Vapnik-Chevronenkis* (VC) dimension, which we will discuss later this lecture. The other is for functional approximation (i.e. the output is continuous-valued, which we may not have time to discuss in this course).

3.3.1 The Vapnik-Chervonenkis Dimension

Suppose we have a set of N points $\mathcal{X}_N = \{x_i : i = 1, ..., N\}$ in d-dimensional space. These points are assumed to be in *general position*, which means that they do not lie in low-dimensional subspaces. E.g., if there are $N = 3$ point in $d = 2$ dimensional space, then the three points are not allowed to lie in a straight line.

Consider all the possible *dichotomies* of the data. These are the number of ways we divide the dataset into two different sets, which we can call the positives and the negatives. There are 2^N possible dichotomies, because each data point can be assigned two possible labels (e.g., $y = 1$ or $y = -1$). Hence each dichotomy corresponds to a set of labelled points $\{(x_i, y_i) : i = 1, ..., N\}$. Different dichotomies have the same x_i's but different y_i's.

Now suppose we have a hypothesis set $\mathcal{A}$ of decision rules. For example, the set of all possible separating hyper-planes (e.g., $\alpha(x) = 1$ if $\vec{a} \cdot \vec{x} + b > 0$, $\alpha(x) = -1$ otherwise, – this is one rule in $\mathcal{A}$, the rules are parameterized by $\vec{a}, b$).

We say that this hypothesis set $\mathcal{A}$ *shatters* a dataset of size N in d dimensions provided, for any dichotomy of the dataset, we can find a rule $\alpha(.) \in \mathcal{A}$ which gives perfect classification. In other words, even before we see the data – the points x_i and their labels y_i – we know that there is a classifier in our hypothesis set $\mathcal{A}$ which can achieve perfect results. This means that we know that we can fit the dataset perfectly, but *it says nothing about generalization.*

This defines the VC-dimension. For any hypotheses set $\mathcal{A}$ (defined over a d-dimensional space), *the VC-dimension h is the maximum number of points (in general position) which can be shattered by the hypothesis class.* For example, it should be easy to convince yourself that the VC dimension for the set of separating hyperplanes in two-dimensions ($d = 2$) is $h = 3$. More generally, $h = d + 1$ for the set of separating hyperplanes.

The concept of VC-dimension allows theorists to prove PAC theorems. The strategy is broadly similar to the method described earlier in this lecture, but the theorems apply to hypotheses sets which contain an infinite number of classifiers (e.g., separating hyperplanes). The insight is that if we fix the positions of the data points, then many classifiers give identical results (if you move a hyperplane by a small amount then you will not change its classification rules). Hence the "effective" number of classifiers is finite and can be quantified by the VC-dimension.

Typical PAC theorems are of form: *With probabaility $> 1 - \delta$:*

$$R(\alpha) \leq R_{emp}(\alpha : \mathcal{X}_N) + \sqrt{\frac{h \log(2N/h) - \log(\delta/4)}{N}}$$

for all $\alpha \in \mathcal{A}$, where $\mathcal{A}$ has VC-dimension h, for any dataset $\mathcal{X}_N$ of size N.

As before, to get decent bounds – $R(\alpha) \approx R_{emp}(\alpha : \mathcal{X}_N)$ for all $\alpha \in \mathcal{A}$ – you need the number of examples $N >> h$ and $N >> -\log \delta$ in order to have a high probability of generalizing from $\mathcal{X}_N$.

Note: these PAC bounds are usually far too conservative and are rarely useful in

practice. This is partly because they are "worst case" bounds. PAC-bayes gives another way to get bounds, based on estimating a posterior distribution $q(\alpha)$ over the classifiers. This gives tighter bounds (Add PAC-Bayes to next draft!).

3.3.2 Perceptron Capacity: Cover's Result

Perceptrons are separating hyper-planes (they also have learning algorithms which we will discuss later). Cover (Stanford) analyzed the *capacity* of perceptrons. His analysis gives an alternative perspective on the idea of shattering. And also derives the probability of finding dichotomies if the amount of data is bigger than the VC dimenision (i.e. if we cannot shatter the data).

Perceptron Capacity: suppose we have n samples in a d-dimensional space. Assume the points are in general position (i.e. no subject of $d+1$ point lies in a $d-1$ dimensional subspace, e.g. in two dimensions $d=2$, no more than two datapoints like on any straight line).

Let $f(n,d)$ be the fraction of the 2^n dichotomies of the n points that can be expressed by linear separation. A dichotomy means that each of the n points is labeled as either 'positive' or 'negative'. So you have a set of points and consider all the possible ways you can label them.

It can be shown that $f(n,d)=1$ for $n<d+1$ – i.e. in this case we know that *we can always find a plane separating the positive and negative data before we have even seen the data!* This is bad, because we know that we can find a plane.

Otherwise, if $n \geq d+1$, then $f(n,d) = \frac{2}{2^n}\sum_{j=0}^{d}\frac{(n-1)!}{j!(n-1-j)!}$. So for $n \geq d+1$, then it is not certain that we can find a plane which separates the positive and negative data points. So if we do find a plane that can separate them – then this means something. It might just be coincidence (i.e. we got lucky) or it may mean that we have found structure in the data (i.e. that we can generalize). The bigger n is, then the more likely it is that we have found structure and the less likely that this is a coincidence. See the plot of this function as a function of n in figure (6).

There is a critical value at $2(d+1)$
$f(n,d)=1, for n \ll 2(d+1)$
$f(n,d)=0, for n \gg 2(d+1)$

The probability of finding a separating hyperplane by chance decreases rapidly for $n>2(d+1)$

Perceptrons can only represent a restricted set of decision rules (e.g. separation by hyperplane). This is a limitation and a virtue. If we can find a separating hyperplane, then it is probably not due to chance alignment of the data (provided $n>(d+1)$), and so it is likely to generalize. In Learning Theory (Vapnik) the quantity $(d+1)$ is the VC dimension of perceptrons and is a measure of the *capacity* of perceptrons. There is a hypothesis space of classifiers – the set of all perceptrons in this case – and this hypothesis space has a *capacity* which is $d+1$ for perceptrons.

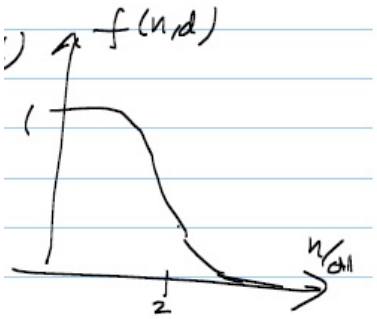

Figure 6: Plot the function $f(n, d)$ as a function of $n/(d+1)$. For small $n < d+1$ the value is 1. For $n >> 2(d+1)$ the value is almost 0, so there is also no chance that we can find a separating plane by 'luck'. If we find a separating plane, then it probably corresponds to structure in the data and will enable us to generalize.