ECE 0402 - Pattern Recognition

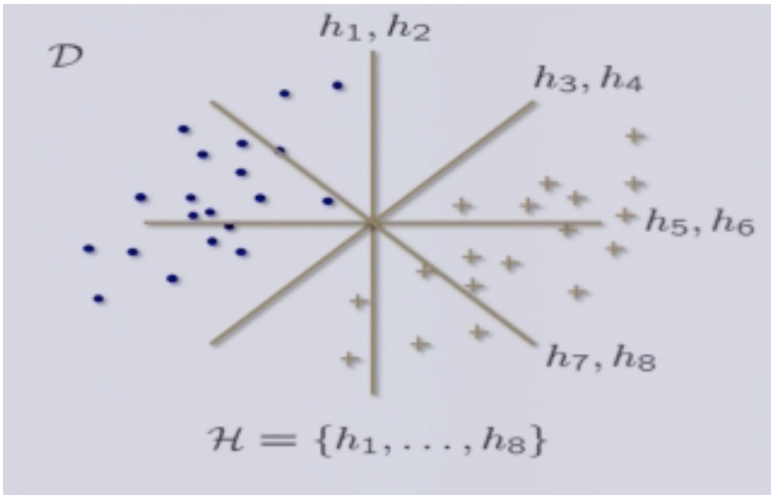
Lecture 3

Review:

Training data: $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ where each $x_i \in \mathbb{R}^d$.

Hypothesis set: $\mathcal{H} = \{h_1, ..., h_m\}$ $h(x) = y$

We will use training data to pick best classifier.



$$true \;\; risk : R(h_j) := \mathbb{P}[h_j(X \neq Y]$$

$$empirical \;\; risk : \hat{R}_n(h_j) := 1/n \sum_{i=1}^{n} 1_{\{i:h_j(x_i) \neq y_i\}}(i) \;\; \text{where} \;\; j = 1, 2, .., m$$

How many times it's not right

We want to choose a hypothesis from $\mathcal{H}$ that returns a small risk.

LLN: if n is large enough, empirical risk is accurate repr. of true risk

A common strategy is to pick

$h^* = \arg\min_{h_j \in \mathcal{H}} \hat{R}_n(h_j)$

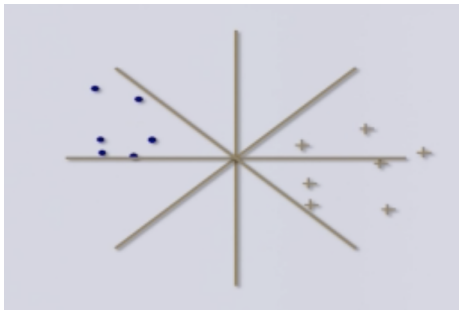
since $\hat{R}_n(h_j)$ is suppose to be a good estimate of $R(h_j)$. As long as n big enough, we expect $\hat{R}_n(h_j) \approx R(h_j)$.

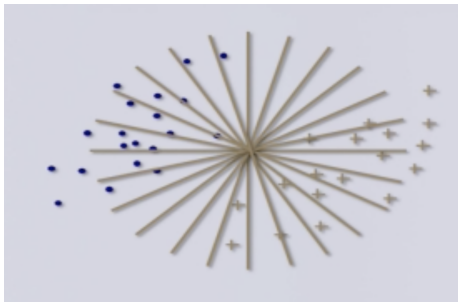
unknown risk

However, if m is very large, then there are some h_k for which $\hat{R}_n(h_k) \ll R(h_k)$, or $\hat{R}_n(h_k) \gg R(h_k)$. In other words, there might be some hypothesis they look much better than they

n small

m large





really are, just by chance. We should worry about this because empirical risk minimization is going to be vulnerable to this problem.

Empirical risk could be small,

- this could be because the true risk is small
- or, could be because $\hat{R}_n(h_k) \ll R(h_k)$ for some h_k.

Which one is more likely? This all depends how big m is.

To get a quantitative resolution to the question how big n needs to be, we have relied on concentration inequalities.

Hoeffding's inequality:

Let $X_1, ..., X_n$ be independent bounded RVs with sum S_i, then for any $\epsilon > 0$,

$$\mathbb{P}[|\ S_n - \mathbb{E}[S_n]\ | \geq \epsilon] \leq 2\ e^{-\frac{2\epsilon^2}{n(b-a)^2}}$$

In our setting this gives us the bound:

$$\mathbb{P}[|\ \hat{R}_n(h_j) - R(h_j)\ | \geq \epsilon] \leq 2\ e^{-2\epsilon^2 n}$$

However, we are ultimately interested in h^*, not just a single h_j. One way to argue that $|\ \hat{R}_n(h^*) - R(h^*)\ |$ is to ensure that $|\ \hat{R}_n(h_j) - R(h_j)\ |$ simultaneously for $\forall j$.

$$\begin{aligned}\mathbb{P}[|\ \hat{R}_n(h^*) - R(h^*)\ | \geq \epsilon] \ \leq\ & \mathbb{P}[\ |\ \hat{R}_n(h_1) - R(h_1)\ | \geq \epsilon \\ & \text{or}\ \ |\ \hat{R}_n(h_2) - R(h_2)\ | \geq \epsilon \\ & \ \ \vdots \\ & \text{or}\ \ |\ \hat{R}_n(h_m) - R(h_m)\ | \geq \epsilon]\end{aligned}$$

$P[E_1 \cup E_2 \cup E_3] \leq P[E_1]$ OR $P[E_2]$ OR $P[E_3]$ union bound

And union bound got us to this point:

$$\mathbb{P}[\ \ |\ \hat{R}_n(h^*) - R(h^*)\ | \geq \epsilon] \leq\ \ 2m\ e^{-2\epsilon^2 n}$$

- linearly increasing with m
- exponentially decreasing with n
- for fixed n how big m can actually be?

When can we be confident that $\hat{R}_n(h^*) \approx R(h^*)$?

Note that $2m\ e^{-2\epsilon^2 n} = e^{\ log(2m)\ -\ 2\epsilon^2 n}$

As long as m isn't too big ($m < e^n$), then we can be confident that $\hat{R}_n(h^*) \approx R(h^*)$

This is good. But not quite enough to have $\hat{R}_n(h^*) \approx R(h^*)$ (learning)

Ideally, we would like to have $R(h^*) \approx 0$

Note that if $\hat{R}_n(h^*) \approx R(h^*)$, then $\hat{R}_n(h^*) \approx 0$ implies $R(h^*) \approx 0$.

We don't know how to go about making this training error always small

The Learning Problem:

1. Can we ensure that $R(h^*)$ is close to $\hat{R}_n(h^*)$? ✓
2. Can we make $\hat{R}_n(h^*)$ small enough?

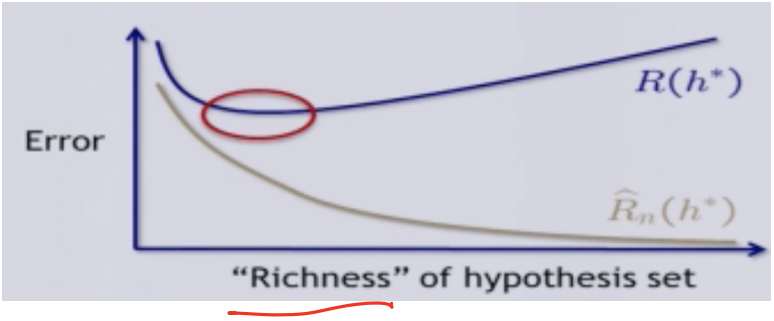
We want to make $\hat{R}_n(h^*)$ as small as possible and our initial intuition was to pick

$$h^* = \arg\min_{h_j \in \mathcal{H}} \hat{R}_n(h_j) \qquad \text{ERM}$$

This will make $\hat{R}_n(h^*)$ small only if there is some $h_j \in \mathcal{H}$ such that actually does well on our training data set. We need a rich set of possible hypotheses...

And we already understood the **fundamental tradeoff** here: more hypothesis ultimately sacrifices our guarantee that $\hat{R}_n(h^*) \approx R(h^*)$.

- if we have rich set of hypothesis training error, $\hat{R}_n(h^*)$ goes down– no news.
- unfortunately at the same time $\hat{R}_n(h^*) - R(h^*)$ goes up.



What is a good hypothesis?

Ideally, we would like to have a small number of hypotheses so that

$$\hat{R}_n(h^*) \approx R(h^*)$$

while also being lucky (smart) enough to have $\hat{R}_n(h^*) \approx 0$. Together these imply $R(h^*) \approx 0$.

In general this may not be possible, since there may not be **any** function f with $R(f) \approx 0$.

Why not?

$$\text{Noise} : Y = f(X) + N$$

Suppose we **knew** the joint distribution of our data,

- what is the optimal classification rule f?
- what are the fundamental limits on how small $R(f^*)$ can be?

Consider (X, Y) pair where

- X is a random vector in $\mathbb{R}^d$
- $Y \in \{0, 1, ..., K-1\}$ is a random variable that depends on X.

Let $f : \mathbb{R}^d \to \{0, 1, ..., K-1\}$ be a some classifier with probability of error/risk given by

$$R(f) \;:=\; \mathbb{P}[f(X) \neq Y]$$

Let's denote **a posteriori class probabilities** by

$$\eta_k(x) \;:=\; \mathbb{P}[Y = k | X = x]$$

after seeing data, what is likelihood of classes

for $k = 0, ..., K-1$.

Optimal function is the one that maximizes posterior probability → Bayesian classifier

Theorem: The classifier $f^*(x) := \arg\max_k \eta_k(x)$ satisfies

$$R(f^*) = \min R(f)$$

where the minimum is over all possible classifiers (every possible k).

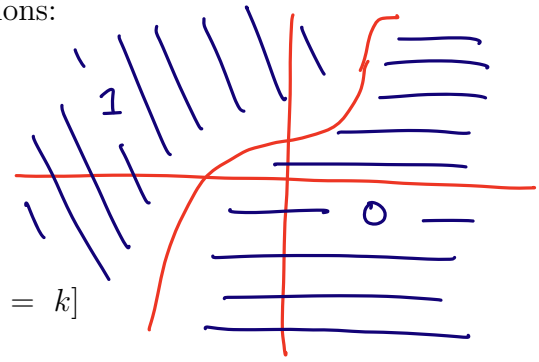
Terminology:

- f^* is called the **Bayes classifier**
- $R(f^*)$ is called the **Bayes risk**

For convenience, we will assume $X|Y = k$ is a continuous RV with density $g_k(x)$ We also need to keep track of **a priori class probabilities**, let's denote that by $\pi_k = P[Y = k]$–how likely is each class.

Consider an arbitrary classifier f, and denote the decision regions:

$$\Gamma_k(f) := \{x : f(x) = k\}$$



Rather than looking at risk, let's consider:

$$\begin{aligned} 1 - R(f) &= \mathbb{P}[f(X) = Y] \\ &= \sum_{k=0}^{K-1} \pi_k \times \mathbb{P}[f(X) = k \mid Y = k] \\ &= \sum_{k=0}^{K-1} \pi_k \int_{\Gamma_k(f)} g_k(x) dx \end{aligned}$$

If we want to maximize this expression, we should design our classifier f such that,

$$x \in \Gamma_k(f) \leftrightarrow \pi_k g_k(x) \text{ is maximized}$$

MAP

Therefore, the optimal f has

$$f^*(x) = \arg\max_k \pi_k g_k(x)$$

Just for fun, I am going to write this expression slightly different

$$f^*(x) = \arg\max_k \frac{\pi_k g_k(x)}{\sum_{l=0}^{K-1} \pi_l g_l(x)}$$

By writing like this I can express this what I have start out (a posteriori class probabilities)– using Bayes rule

$$f^*(x) = \arg\max_k \ \mathbb{P}[Y = k \mid X = x]$$

This classifier makes common sense!

There are lots of ways of expressing the Bayes classifier. Let's look at the couple variations because most likely you have seen this before.

- $f^*(x) = \arg\max_k \ \eta_k(x)$ Bayes
- $f^*(x) = \arg\max_k \ \pi_k g_k(x)$
- When $K = 2$, we can do **likelihood ratio test**

$$\frac{g_1(x)}{g_0(x)} \underset{1}{\overset{0}{\lessgtr}} \frac{\pi_0}{\pi_1}$$

- When $\pi_0 \ = \pi_1 = ... = \pi_{K-1}$

$$f^*(x) = \arg\max_k \ g_k(x)$$

This has a special name "ML-classifier", maximum likelihood classifier/detector.

Let's look at a $2-d$ example: suppose that $K = 2$ and that

$$X|Y = 0 \sim \mathcal{N}(0,1) \text{ and } X|Y = 1 \sim \mathcal{N}(1,1)$$

Here we are randomly choosing a class according to some distribution. And given which class we belong to, there is some class conditional distribution for X. What is the likelihood ration test say:

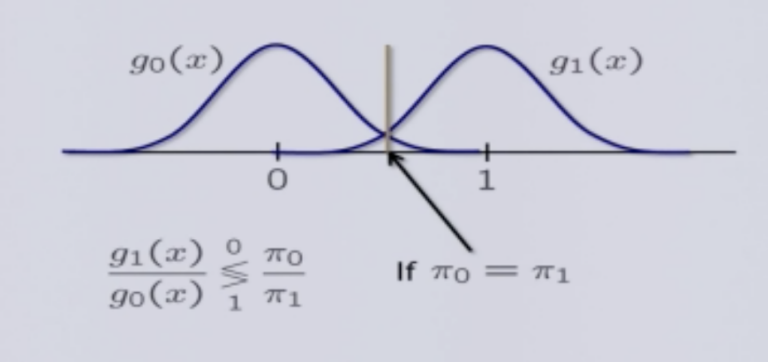
$$\frac{g_1(x)}{g_0(x)} \underset{1}{\overset{0}{\lessgtr}} \frac{\pi_0}{\pi_1}$$

If $\pi_0 = \pi_1$, threshold will be 1. And how do we calculate the Bayes risk in this case?

$$\begin{aligned} R(f^*) &= \mathbb{P}[f^*(X) \neq Y] \\ &= \mathbb{P}[\textit{declare } 0 | Y = 1]\pi_1 \\ &\quad + \mathbb{P}[\textit{declare } 1 | Y = 0]\pi_0 \end{aligned}$$

In the case where $\pi_0 = \pi_1 = 1/2$, our test reduced to declaring 1 iff $x \geq \frac{1}{2}$ (Let's plot how this looks like).

$P[X=0 \mid Y=1]\,P[Y=1] + P[X=1 \mid Y=0]\,P[Y=0]$



Thus,

$$\begin{aligned} R(f^*) &= \mathbb{P}[X < \frac{1}{2}|Y = 1] + \mathbb{P}[X > \frac{1}{2}|Y = 0] \\ &= \frac{1}{2}\int_{-\infty}^{\frac{1}{2}} g_1(t)dt + \frac{1}{2}\int_{\frac{1}{2}}^{\infty} g_0(t)dt \\ &= \Phi\left(-\frac{1}{2}\right) \end{aligned}$$

End note: Note that we don't know any of the g_ks, π_ks, and it is going to super hard in a lot of cases. Everything we did here, assumed we know the full distribution–what generated our data. If we know π_ks and all g_ks, then we can actually do the optimal thing. Not today, we will talk about this next lecture, we will look at the methods that are inspired by this approach.We might assume distributions as a start (Gaussian is always a good one), and estimate π_ks...

Assumed that we know the data's full distribution - which is not always the case