# ECE 0402 - Pattern Recognition

Lecture 6

**Reference reading:** Please also refer to Learning from Data 2.1.3 and 2.1.4 for further discussion of the topic discussed in this lecture note.

**Review:** We showed that for a given $\mathcal{H}$, if we know that $k$ is a **break point** (meaning that no data set of size k can be **shattered**), then the growth function is bounded by this sum:

$$m_{\mathcal{H}}(n) \leq \sum_{i=0}^{k-1} \binom{n}{i} \implies \text{polynomial with leading term } n^{k-1}$$

- Positive rays ($k = 2$):

$$m_{\mathcal{H}}(n) = n + 1 \leq n + 1$$

- Positive intervals ($k = 3$):

$$m_{\mathcal{H}}(n) = \frac{1}{2}n^2 + \frac{1}{2}n + 1 \leq \frac{1}{2}n^2 + \frac{1}{2}n + 1$$

- Linear Classifiers in $\mathbb{R}^2$ (k=4):

$$m_{\mathcal{H}}(n) \leq \frac{1}{6}n^3 + \frac{5}{6}n + 1$$

Bottom line if is we have a set of classifier, all we need is a break point to exist – growth function is a polynomial $n^{k-1}$

we can actually replace $\mid \mathcal{H} \mid$ with $m_{\mathcal{H}}(n)$ to obtain an inequality along the lines of

$$R(h^*) \leq \hat{R}_n(h^*) + \sqrt{\frac{1}{2n} \; log \; \frac{2m_{\mathcal{H}}(n)}{\delta}}$$

We won't be able to quite show this for technical reasons (which we will soon see). We will only be able to show that with probability $\geq 1 - \delta$

$$\boxed{R(h^*) \leq \hat{R}_n(h^*) + \sqrt{\frac{8}{n} \; log \; \frac{4m_{\mathcal{H}}(2n)}{\delta}}}$$

This is called **VC generalization bound**. VC stands for Vapnik and Chervonenkis who proved in 1971.

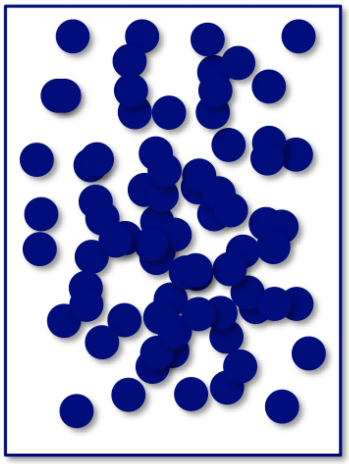


Figure 1: UB

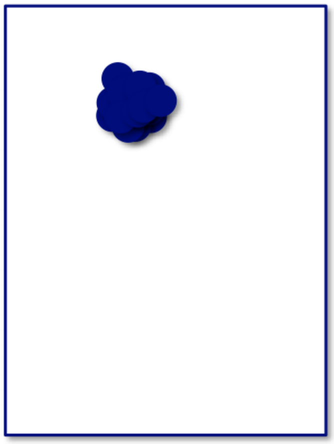


Figure 2: VC

Mathematically, using Hoeffding's inequality together with a union bound, we were able to show that

$$\mathbb{P}[max_{h\in\mathcal{H}} \mid \hat{R}_n(h) - R(h) \mid > \epsilon] \leq \mid \mathcal{H} \mid \cdot 2e^{-2\epsilon^2 n}$$

What the VC bound gives us is a generalization of the form:

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} \mid \hat{R}_n(h) - R(h) \mid > \epsilon\right] \leq 2 \cdot m_{\mathcal{H}}(2n) \cdot 2e^{-\frac{1}{8}\epsilon^2 n}$$

**Supremum**: The supremum of a set $\mathcal{S} \subset T$ is the least element of $T$ that is greater than of equal to all elements of $\mathcal{S}$. (This is sometimes called the least-upper-bound which probably makes a lot more sense). Here are several examples:

- $sup\{1, 2, 3\} = 3$
- $sup\{x : 0 \leq x \leq 1\} = 1$
- $sup\{x : 0 < x < 1\} = 1$
- $sup\{1 - 1/n : n > 0\} = 1$

The magic in the proof of the VC bound is to realize that we can relate the supremum over all $h \in \mathcal{H}$ to the maximum over a finite set of $h \in \mathcal{H}$ using a really cool trick!

### VC Bound

$$R(h) \leq \hat{R}_n(h) + \sqrt{\frac{8}{n}\ log\ \frac{4m_{\mathcal{H}}(2n)}{\delta}}$$

**Role of Growth Function** We aim to get a bound on (no matter what "h" we picked from our hypothesis set, we wanna believe that the training error is a good predictor of the true risk. )

$$\mathbb{P}\left[\mid \hat{R}_n(h) - R(h) \mid \quad > \epsilon \right]$$

that hold for any $h \in \mathcal{H}$, i.e., a bound on

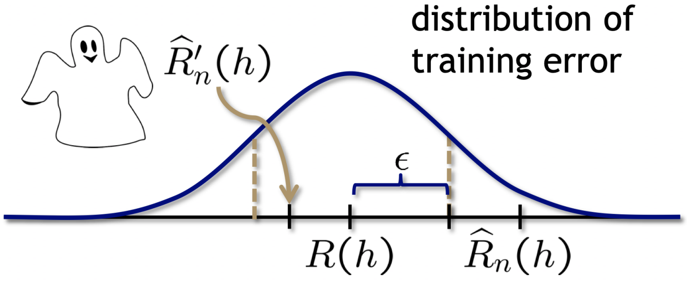
$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} \ \mid \hat{R}_n(h) - R(h) \mid \quad > \ \epsilon\right]$$

Perhaps it is not surprising that we can understand $\hat{R}_n(h)$ using growth function...

There may be infinitely many $h \in \mathcal{H}$, but $\mathcal{H}$ can only generate $m_{\mathcal{H}}(n)$ **unique dichotomies** for $n$ data points. Thus, empirical risk $\hat{R}_n(h)$ can only take finitely many – at most $m_{\mathcal{H}}(n)$ – different values.

Unfortunately, $R(h)$ can still take infinitely many different values, and so there are infinitely many $\mid \hat{R}_n(h) - R(h) \mid$.

**Fundamental insight**: The key trick is to consider two different datasets!–this is for sake of analysis. We will imagine that in addition to our training data, we have access to a second independent dataset (of size $n$), which we call the **ghost sample**. Here the blue line is the distribution of empirical risk, and $\hat{R}'_n(h)$ is the second estimate of $R(h)$.



Can we relate $\mathbb{P}\left[\mid \hat{R}_n(h) - R(h) \mid \quad > \epsilon \right]$ to some $\mathbb{P}\left[\mid \hat{R}'_n(h) - \hat{R}_n(h) \mid \quad > \epsilon \right]$?

Suppose (for the moment) that the empirical estimates $\hat{R}_n(h)$ and $\hat{R}'_n(h)$ are RVs that are drawn from a symmetric distribution with mean $R_n(h)$.

Consider the following events:

- A: the event that $\mid \hat{R}_n(h) - R(h) \mid \quad > \epsilon$
- B: the event that $\mid \hat{R}_n(h) - \hat{R}'_n(h) \mid \quad > \epsilon$

**Claim**: $\mathbb{P}\left[B|A\right] \geq \frac{1}{2}$

Because of this symmetry assumption: the probability $B|A$ bigger than a half. So why is that true?

Thus $\mathbb{P}\left[B\right] \geq \mathbb{P}\left[B|A\right] \cdot \mathbb{P}\left[A\right] \geq \frac{1}{2}\mathbb{P}\left[A\right]$

$$\implies \mathbb{P}\left[\mid \hat{R}_n(h) - R(h) \mid \ \ > \epsilon\right] \leq 2\ \mathbb{P}\left[\mid \hat{R}_n(h) - \hat{R'}_n(h) \mid \ \ > \epsilon\right]$$

Unfortunately the distribution of $\hat{R}_n(h)$ and $\hat{R'}_n(h)$ is binomial (not symmetric) so this exact statement doesn't hold in general, but the intuition is valid.

Instead, we have the following bound:

**Lemma 1 (Ghost sample)**

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} \ \mid \hat{R}_n(h) - R(h) \mid \ \ > \ \epsilon\right]$$
$$\leq 2\mathbb{P}\left[\sup_{h\in\mathcal{H}} \ \mid \hat{R}_n(h) - \hat{R'}_n(h) \mid \ \ > \ \frac{\epsilon}{2}\right]$$

We wanna understand worse-case deviation between our empirical risk and the true risk. But rather than analyzing that directly, we are gonna instead upper-bound that by looking at the worst case deviation between pairs of two empirical estimates.

**Lemma 2 (Where the magic happens)**

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} \ \mid \hat{R}_n(h) - \hat{R'}_n(h) \mid \ \ > \ \frac{\epsilon}{2}\right]$$
$$\leq m_{\mathbb{H}}(2n) \cdot \sup_{\mathcal{S}}\ \sup_{h\in\mathcal{H}}\ \mathbb{P}\left[\mid \hat{R}_n(h) - \hat{R'}_n(h) \mid \ \ > \ \frac{\epsilon}{2}\middle|\mathcal{S}\right]$$

$\mathcal{S}$ is a sample of 2n observations.

(If we only looking at two empirical estimates deviate by a large amount, we can use a union bound again. Because we now only have two datasets so there is only finitely many hypothesis we need to consider. This gave us the Lemma 2– derive on board).

But, still this looks ugly. What's the worst-case probability (over datasets and overall hypothesis) of our two empirical risks deviate by more than $\epsilon/2$ conditioned on what the dataset actually was. Conditioned on $\mathcal{S}$, what is random here? There is still only one little thing, we have $2n$ points but we haven't said which of the hypothesis they go to. If you give me $2n$ points and we split them to half, what is the probability that these two empirical estimates gives us very different averages.

**Lemma 3**

For **any** fixed classifier $h \in \mathcal{H}$ and **any** fixed set of $2n$ data points $\mathcal{S}$,

$$\mathbb{P}\left[ \mid \hat{R}_n(h) - \hat{R}'_n(h) \mid \ \ > \ \frac{\epsilon}{2} |\mathcal{S} \right] \leq 2e^{-\frac{\epsilon^2 n}{8}}$$

where again the probability is w.r.t. a random partitioning of **any** $\mathcal{S}$ into 2 training sets of size $n$. (this is a version/variant of Hoeffding inequality).

One big thing to note in Lemma 3 is that, this is a statement that holds for any possible hypothesis h, and any possible sets of data. There is nothing about hypothesis or the dataset–more like "I have bunch of numbers, I split them in half and averaged them do I get similar numbers"...

Final Step: Putting all of this together, we get

$$\mathbb{P}\left[ \text{sup}_{h \in \mathcal{H}} \ \mid \hat{R}_n(h) - R(h) \mid \ \ > \ \epsilon \right] \leq 2 \ m_{\mathcal{H}}(2n) \ 2e^{-\frac{\epsilon^2 n}{8}}$$

This was the main result we were after. We can also state this as confidence bound. For any $h \in \mathcal{H}$, we have that with probability $\geq 1 - \delta$

$$R(h) \leq \hat{R}_n(h) + \sqrt{\frac{8}{n} \ log \frac{4 m_{\mathcal{H}}(2n)}{\delta}}$$

**Summary so far...** there are really 2 interesting ideas in this whole thing:

- one was that you can understand deviation bounds (the probability that my empirical risk deviates from the true risk) and kind of relate to that saying "what's the probability that two empirical estimates are very different from each other"
- and once we are only looking at 2 empirical estimates, you can just use the same kind of union bound as before. And the growth function (max ways of labelling 2n points) pops out in a very natural way. So you get rid of having to think about "m" instead you have this $m_{\mathcal{H}}$.

In English, if you tell me the training risk, we know that true risk has to be bounded by it plus that log term in the VC bound. What that means is, if we know the training error (which we do), we can use this as an upper bound on the True-error (which we truly do not know).

**The VC Dimension**

The confidence bound we had with VC bound is stated in terms of growth function–it is actually going to be useful to go one more step an not talk about growth functions all the

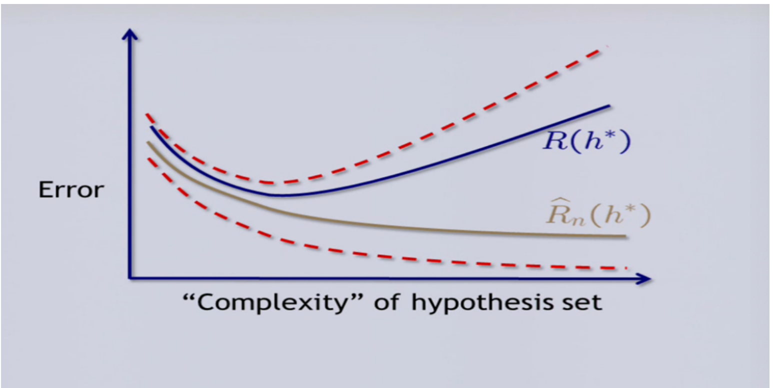


Figure 3: Interpretation of VC Bound

time.. remember we went to a lot of work to show if there is a break pt, we can think of the growth function as being a polynomial...

We showed that if $k$ is a break point for $\mathcal{H}$ then $m_{\mathcal{H}}(n) \leq \sum_{i=0}^{k-1} \binom{n}{i} \leq n^{k-1} + 1$.

$$\begin{aligned} \implies R(h) &\leq \hat{R}_n(h) + \sqrt{\frac{8}{n} \log \frac{4((2n)^{k-1}+1)}{\delta}} \\ &\lessapprox \hat{R}_n(h) + \sqrt{\frac{8(k-1)}{n} \log \frac{8n}{\delta}} \end{aligned}$$

This approximation is true if $k \geq 3$.

So what we are seeing here is, if k is a break point, $k-1$ being an suddenly interesting number in terms of controlling how big this confidence bound is going to be! In fact $k-1$ is important enough that it is given a special name, it is called the “VC dimension”

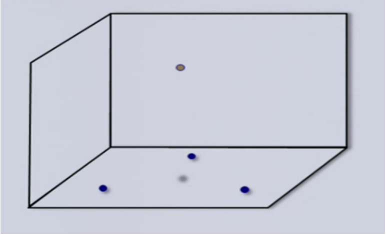
**Definition:** The **VC dimension** of $\mathcal{H}$ is the largest $n$ for which $m_{\mathcal{H}}(n) = 2^n$. The denotation for the VC dimension of a hypothesis set $\mathcal{H}$ is, $d_{VC}(\mathcal{H})$. In other words, $d_{VC}(\mathcal{H})$ is the maximum number of points that our hypothesis shatters. In other other words, $d_{VC}(\mathcal{H})$ is 1 less than the smallest break point.

$$R(h) \lessapprox \hat{R}_n(h) + \sqrt{\frac{8d_{VC}}{n} \log \frac{8n}{\delta}}$$

**Examples:**

- Positive rays: $d_{VC} = 1$.
- Positive intervals: $d_{VC} = 2$.
- Convex sets: $d_{VC} = \infty$.

- Linear classifiers in $\mathbb{R}^2$ : $d_{VC} = 3$.



- How about linear classifier in $\mathbb{R}^3$?

So the fact that the linear classifiers had a break point of $k = 4$ was only due to the fact that we were only looking at $\mathbb{R}^2$. In the higher dimensions it changes.

In general $d_{VC} = d + 1$ for linear classifiers in $\mathbb{R}^d$ .

- You can prove that by showing $d_{VC} \leq d + 1$ and $d_{VC} \geq d + 1$, usual tricks! If you haven't seen, this is very often and the easiest ways to proof things of same sort.

proof:

- One Direction: Let's first show that there exists a set of $d + 1$ points in $\mathbb{R}^d$ that are shattered.

  This is relatively straightforward:

$$X = \begin{bmatrix} - & \tilde{x}_1^T & - \\ - & \tilde{x}_2^T & - \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ - & \tilde{x}_{d+1}^T & - \end{bmatrix}_{(d+1)\times(d+1)} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ & \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix}$$

  I chose a particular data points. This choice makes $X$ an invertible matrix. Then the question is, can we shatter this data set? In other words,

  - For any

$$y = \begin{bmatrix} y_1 \\ y2 \\ \vdots \\ y_{d+1} \end{bmatrix} = \begin{bmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{bmatrix}$$

    can we find a vector $sign(X\theta) = y$ ?

  - If we make $\theta = X^{-1}y \implies sign(X\theta) = sign(y) = y$

    We can shatter $d{+}1$ pts. This $\implies d_{VC} \geq d{+}1$ – all we know that VC-dimension is at least $d + 1$.

- Other Direction: we need to show $d_{VC} \leq d+1$. In order to show that, we need to show,
  - We can not shatter any set of $d+2$ points
  - Take any $d+2$ points $x_1, ..., x_{d+2}$. This has to be true no matter what these pts are!
  - We have more points than dimensions, so there must be some $j$ for which

$$x_j = \sum_{i \neq j} \alpha_i x_i$$

    where not all $\alpha_i = 0$

  - consider the dichotomy where the $x_i$ with $\alpha_i \neq 0$ are labeled $y_i = sign(\alpha_i)$, and $y_j = -1$
  - No linear classifier can implement such a dichotomy

$$x_j = \sum_{i \neq j} \alpha_i x_i \implies \theta^T x_j = \sum_{i \neq j} \alpha_i \theta^T x_i$$

    If $y_i = sign(\theta^T x_i) = sign(\alpha_i)$, then $\alpha_i \theta^T x_i > 0$

    - This means that $\theta^T x_j = \sum_{i \neq j} \alpha_i \theta^T x_i > 0$
    - Thus, $y_j = sign(\theta^T x_j) = +1$

We have just shown that for a linear classifier in $\mathbb{R}^d$

$$\begin{aligned} d_{VC} &\geq d+1 \\ d_{VC} &\leq d+1 \\ &\implies d_{VC} = d+1 \end{aligned}$$

How many parameters does a linear classifier in $\mathbb{R}^d$ have?

$$\begin{aligned} w &\in \mathbb{R}^d \\ b &\in \mathbb{R} \implies d+1 \text{ parameters} \end{aligned}$$

And the $d_{VC} = d+1$, is this a coincidence? Let's look at our other examples:

- Positive rays:
  - $d_{VC} = 1$
  - 1 parameter
- Positive intervals:

  - $d_{VC} = 2$
  - 2 parameters

- Convex sets:
  - $d_{VC} = \infty$
  - as many as you want

So VC dimension is “effective number of parameters” – meaning that additional parameters do not always contribute additional degrees of freedom.

You can introduce parameters that do nothing, that have no actual effect on the VC dimension. As an example of this: take the output of a linear classifier, and then feed this into another linear classifier

$$y_i = sign(\omega'(sign(\theta^T x_i) + b')$$

This is adding no additional degrees of freedom. The parameters $\omega'$ and $b'$ are redundant –they do not allow us to create any new dichotomies.

**How big does our training set need to be?**

$$R(h) \lessapprox \hat{R}_n(h) + \sqrt{\frac{8d_{VC}}{n} \log \frac{8n}{\delta}}$$

Just to see how this bound behaves, we can ignore the constants and look at:

$$\epsilon \sim \sqrt{\frac{d_{VC}}{n} \log n}$$

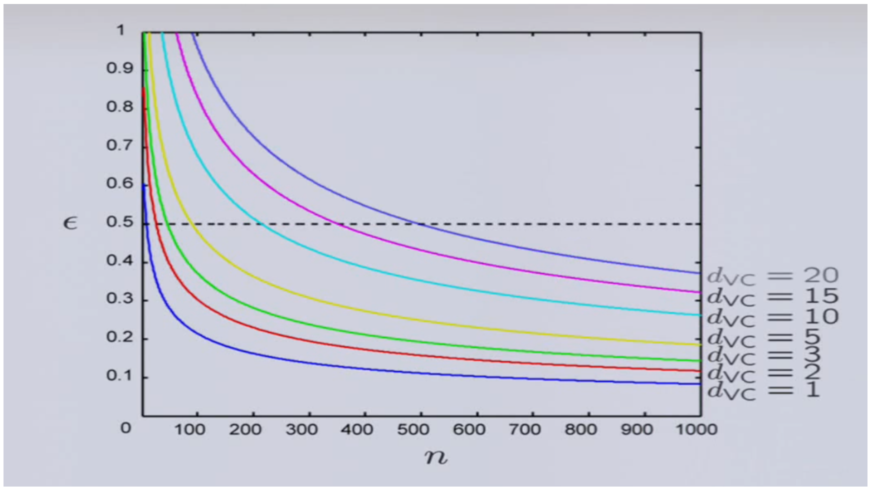


Figure 4: VC tightness versus data size

**Rule of thumb (in practice):** $n \geq 10 d_{VC}$

**Recap**:

The learning Problem:

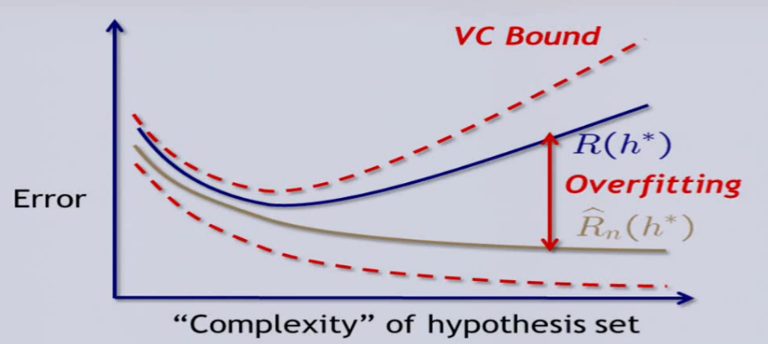
Given a set $\mathcal{H}$, find a function $h \in \mathcal{H}$ that minimizes $R(h)$.

In the case of classification, we can also think of this as trying to find $h \in \mathcal{H}$ that approximates the Bayes classifier $f*$.

- More complex $\mathcal{H} \implies$ better chance of **approximating** $f^*$.
- Less complex $\mathcal{H} \implies$ better confidence bound/ better chance of **generalizing** to out of sample

$\downarrow$

**"Approximation-generalization tradeoff"**



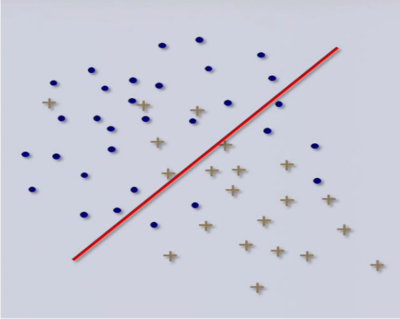


Figure 5: okay

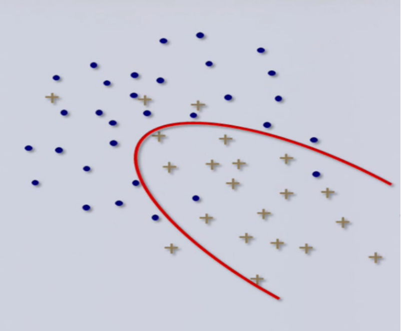


Figure 6: maybe better

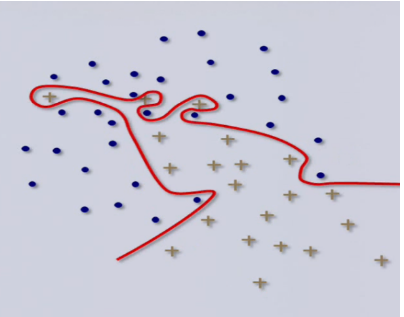


Figure 7: stupid

### Beyond classification

In supervised learning problems we are given training data

$$(x_1, y_1), ..., (x_n, y_n)$$

where $x_i \in \mathbb{R}^d$, and so far we have only considered the case $y_i \in \{+1, -1\}$ (or $y_i \in \{0, ..., K-1\}$.

What if $y_i \in \mathbb{R}$? This problem is usually called **regression**. $y_i$'s are dependent variables.

We can think of regression as being an extension of classification as the number of classes grows to $\infty$.

### Regression

A regression model typically posits that our training data are realizations of a random pair $(X, Y)$ where

$$Y = f(X) + E$$

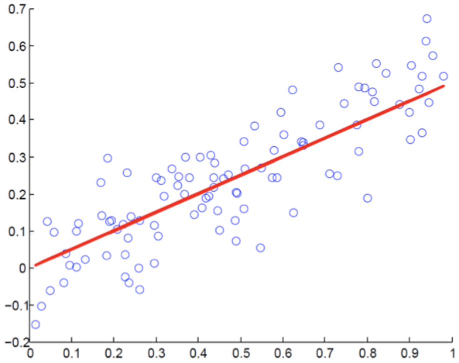
with $E$ representing noise and $f$ belonging to some class of functions.

Example class functions:

- polynomials
- sinusoids/ trigonometric polynomials
- exponentials
- kernels

**Linear Regression**: In linear regression, we assume that $f$ is an **affine** function, i.e.,

$$f(x) = \beta^T x + \beta_0$$



where $\beta \in \mathbb{R}^d$ and $\beta_0 \in \mathbb{R}$.

The question now is basically: how can we estimate parameters $\beta, \beta_0$ from training data?

**Least Squares**: In least squares linear regression, we select $\beta, \beta_0$ to minimize the sum of squared errors

$$SSE(\beta, \beta_0) := \sum_{i=1}^{n} \left( y_i - \beta^T x_i - \beta_0 \right)^2$$

And we like to minimize this...

Legendre (1805), and of course Gauss (1795,1809).

**Example**: Suppose $d = 1$, so that $x_i, \beta$ are scalars.

$$SSE(\beta, \beta_0) = \sum_{i=1}^{n} \left( y_i - \beta^T x_i - \beta_0 \right)^2$$

How to minimize?

$$\frac{\partial SSE}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left( y_i - \beta^T x_i - \beta_0 \right) = 0$$
$$\frac{\partial SSE}{\partial \beta} = -2 \sum_{i=1}^{n} x_i \left( y_i - \beta^T x_i - \beta_0 \right) = 0$$

Rearranging these equations,

$$n\beta_0 + \sum_{i=1}^{n} \beta x_i = \sum_{i=1}^{n} y_i$$
$$\sum_{i=1}^{n} \beta_0 x_i + \sum_{i=1}^{n} \beta {x_i}^2 = \sum_{i=1}^{n} y_i x_i$$

or in matrix form

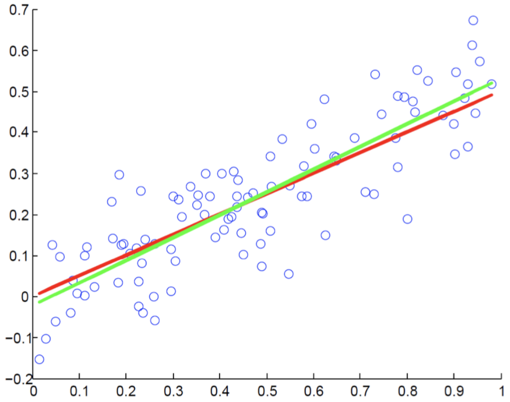
$$\begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i {x_i}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

inverting the matrix

$$\begin{bmatrix} \beta_0 \\ \beta \end{bmatrix} = \begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i {x_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

Setting $\bar{x} = \frac{1}{n} \sum_i x_i$ and $\bar{y} = \frac{1}{n} \sum_i y_i$,

$$\begin{bmatrix} \beta_0 \\ \beta \end{bmatrix} = \frac{1}{\sum_i {x_i}^2 - n\bar{x}^2} \begin{bmatrix} \bar{y}\left(\sum_i x_i^2\right) - \bar{x} \sum_i x_i y_i \\ \sum_i x_i y_i - n\bar{x}\bar{y} \end{bmatrix}$$



**General Least Squares** Suppose $d$ is arbitrary. Set

$$\theta = \begin{bmatrix} \beta_0 \\ \beta(1) \\ \vdots \\ \beta(d) \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad A = \begin{bmatrix} 1 & x_1(1) & \dots & x_1(d) \\ 1 & x_2(1) & \dots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \dots & x_n(d) \end{bmatrix}$$

Then

$$SSE(\theta) = \sum_{i=1}^{n} \left( y_i - \beta^T x_i - \beta_0 \right)^2 = \|y - A\ \theta\|^2$$

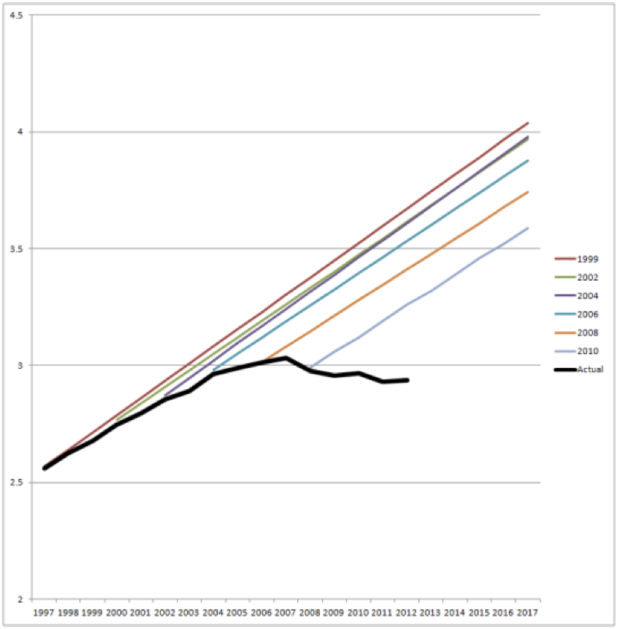
The minimizer $\hat{\theta}$ of this quadratic objective function is

$$\boxed{\hat{\theta} = \left(A^T A\right)^{-1} A^T y}$$

provided that $A^T A$ is nonsingular.

**“Proof”**:

$$\begin{aligned} \|y - A\theta\|^2 &= (y - A\theta)^T (y - A\theta) \\ &= y^T y - 2y^T A\theta + \theta^T A^T A\theta \\ \nabla_\theta \|y - A\theta\|^2 &= -2A^T y + 2A^T A\theta = 0 \\ &\Downarrow \\ \hat{\theta} &= \left(A^T A\right)^{-1} A^T y \end{aligned}$$



Here is US DOT forecast of road traffic, compared to actual :)

**Does LR always make sense?**

Sometimes linear methods (regression and classification) just don't work. One way to create nonlinear estimators (or classifiers) is to first transform the data via a **nonlinear feature map**.

$$\Phi : \mathbb{R} \to \mathbb{R}^{d'}$$

After applying $\Phi$, we can then try a linear method to the transformed data $\Phi(x_1), ..., \Phi(x_n)$. In the case of regression, our model becomes

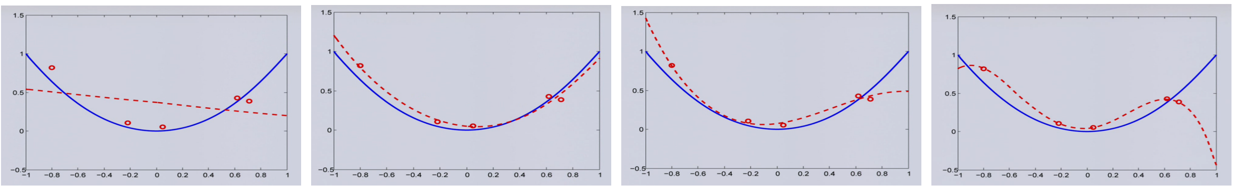
$$f(x) = \beta^T \Phi(x) + \beta_0$$

where now $\beta \in \mathbb{R}^{d'}$. We have more features but we can still use standard least squares.

**Example**: Suppose $d = 1$ but $f(x)$ is cubic polynomial. How do we find a least squares estimate of $f$ from training data.

$$\Phi_k(x) = x^k \to A = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix}$$

### Overfitting

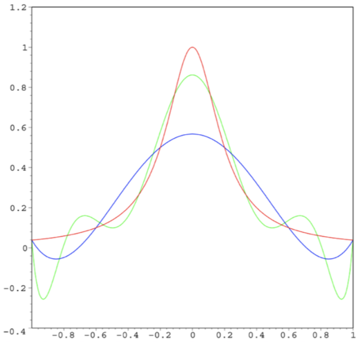


linear, quadratic, cubic and quartic fits

Noise in the observations can make overfitting a big problem, but even if there is no noise, fitting a higher order polynomial (interpolation) can be incredibly unstable.

For example, one called “Runge’s phenomenon”(you may of encountered this in a numerical analysis class). When you take a smooth function:

- not exactly polynomial
- well approximated by a polynomial
- but what order?



VC generalization bound said “overfitting” can be judged by looking at how complicated our $\mathcal{H}$ and how many training data we have:

$$R(h) \lessapprox \hat{R}(h) + \epsilon(\mathcal{H}, n)$$

This is one way of **<u>quantifying this tradeoff</u>**. When we are talking about regression, it is going to turn out that, there is a more natural way and definitely much cleaner bound for understanding how our error behaves. It is going to be something called “bias-variance decomposition”.

An alternative approach: **Bias-variance decomposition**

- **bias**: how well can $\mathcal{H}$ actually approximates true underlying function $f^*$

  –So if your true function, is not exactly a polynomial and you are trying to represent it using a polynomial, then there is some inherent mismatch between your model and the real. Or If you had a non-bandlimited signal and you are trying to represent it using only finitely many fourier series coefficients, there is some inherent bias to your representation.basically no matter what you do, that’s gonna be a problem. This bias is one thing to contributes to our error.

- **variance**: how well can we pick a good $h \in \mathcal{H}$

$$R(h) = \text{bias} + \text{variance}$$

Bias-variance decomposition is useful because it is more easily generalizes to regression problem.

**Bias-variance decomposition**: In regression, we will assume real-valued observations (i.e., regression) and consider squared error for the risk.

- $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$ where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$
- $f : \mathbb{R}^d \to \mathbb{R}$: unknown target function
- $h_{\mathcal{D}} : \mathbb{R}^d \to \mathbb{R}$: function in $\mathcal{H}$ we pick using $\mathcal{D}$

$$R(h_{\mathcal{D}}) = \mathbb{E}_X \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right]$$

Setting up the decomposition:

$$\begin{aligned} R(h_{\mathcal{D}}) &= \mathbb{E}_X \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right] \\ \mathbb{E}_{\mathcal{D}}[R(h_{\mathcal{D}})] &= \mathbb{E}_{\mathcal{D}} \left[ \mathbb{E}_X \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right] \right] \\ &= \mathbb{E}_X \left[ \mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right] \right] \end{aligned}$$

To evaluate $\mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right]$, we will break up into two terms.

- We define the **average hypothesis** as: $\bar{h}(X) = \mathbb{E}_D \left[ h_{\mathcal{D}}(X) \right]$
- Think about drawing many datasets $\mathbb{D}_1, ..., \mathbb{D}_p$

$$\bar{h}(X) \approx \frac{1}{p} \sum_{i=1}^{p} h_{\mathcal{D}_i}(X)$$

- decomposition:

$$\begin{aligned} &\mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right] \\ &\quad = \mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - \bar{h}(X) + \bar{h}(X) - f(X))^2 \right] \\ &\quad = \mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - \bar{h}(X))^2 + (\bar{h}(X) - f(X))^2 + 2(h_{\mathcal{D}}(X) - \bar{h}(X))(\bar{h}(X) - f(X)) \right] \\ &\quad = \mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - \bar{h}(X))^2 \right] + (\bar{h}(X) - f(X))^2 \end{aligned}$$

Plugging this back into the original expression, we get

$$\begin{aligned} R(h_{\mathcal{D}}) &= \mathbb{E}_X \left[ \mathbb{E}_D \left[ (h_{\mathcal{D}}(X) - f(X))^2 \right] \right] \\ &= \mathbb{E}_X \left[ \text{bias}(X) + \text{variance}(X) \right] \\ &= \text{bias} + \text{variance} \end{aligned}$$