ECE 0402 - Pattern Recognition

Lecture 10

Recap:

The learning Problem:

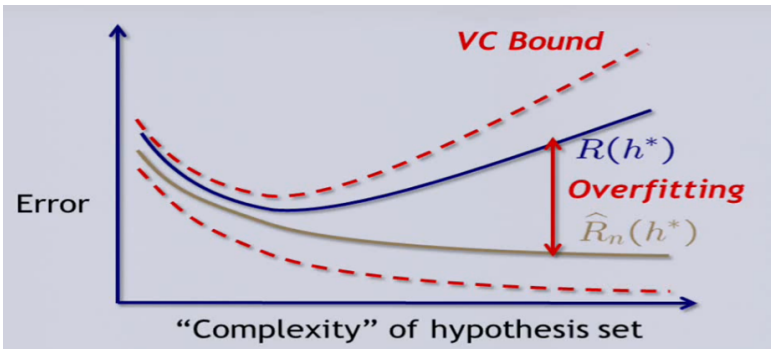
Given a set $\mathcal{H}$, find a function $h \in \mathcal{H}$ that minimizes $R(h)$.

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"



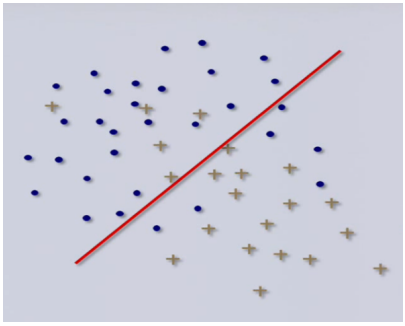


Figure 1: okay

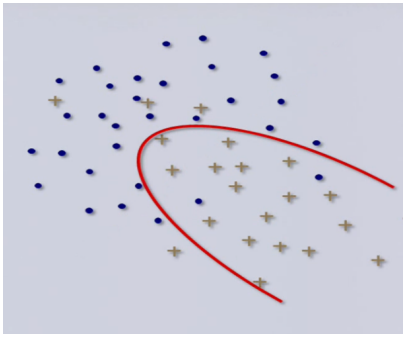


Figure 2: maybe better

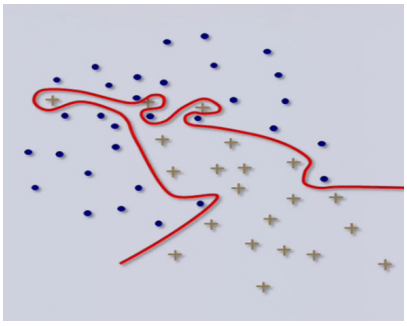


Figure 3: stupid

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 ∞. $K \to \infty$

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 $\beta_0 \in \mathbb{R}$.

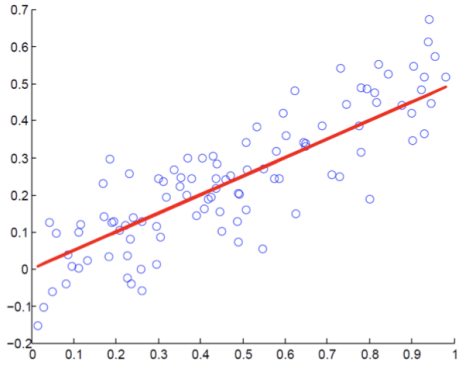
The question now is basically: how can we estimate parameters β, β_0 from training data?

Least Squares: In least squares linear regression, we select β, β_0 to minimize the sum of squared errors

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

And we like to minimize this...

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



Example: Suppose $d = 1$, so that x_i, β 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

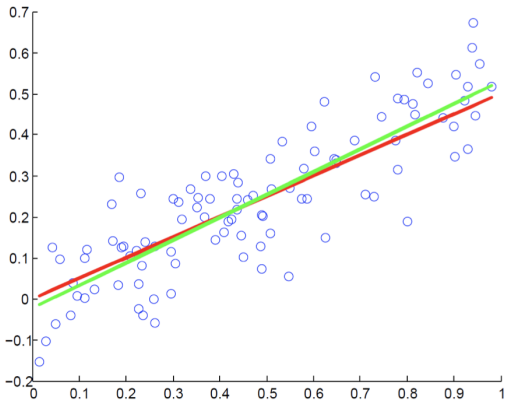
$$\begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i {x_i}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

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

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

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

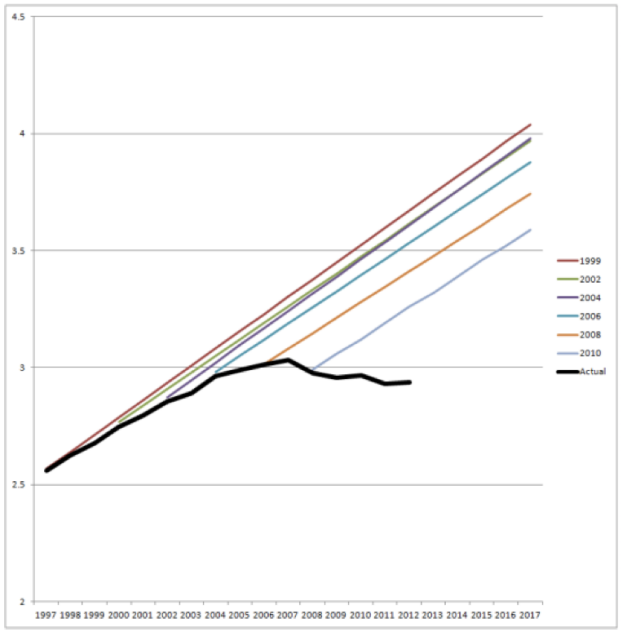
$$\boxed{\hat{\theta} = \left(A^T A\right)^{-1} A^T y}$$

provided that $A^T A$ is nonsingular.

"Proof":

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

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 Φ, 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

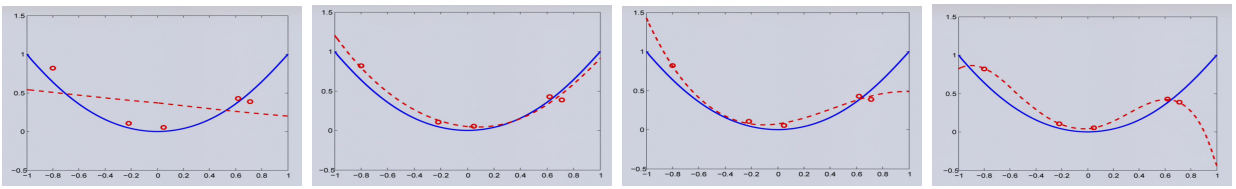
$$f(x) = \beta^T \Phi(x) + \beta_0$$

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

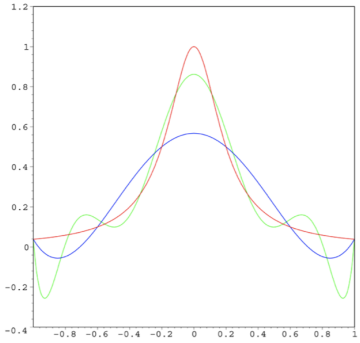


linear, quadratic, cubic and quartic fits

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 **quantifying this tradeoff**. And alternative approach: **Bias-variance decomposition**

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

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