Lecture 7:
Linear supervised learning

Nando de Freitas
www.cs.ubc.ca/~nando/340-2009/
September 2009

Outline

Linear regression is a supervised learning task. It is of great interest because:

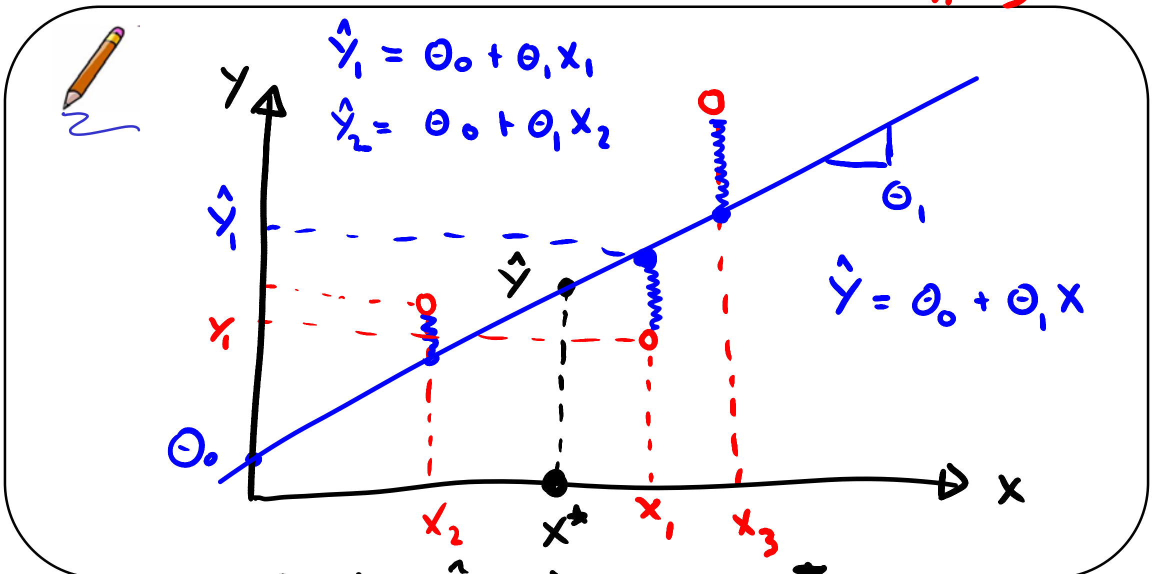
- Many real processes can be approximated with linear models.
- Linear regression appears as part of larger problems.
- It can be solved analytically.
- It illustrates many of the approaches to machine learning.

Least squares

Given the data $\{x_{1:n}, y_{1:n}\}$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we want to fit a hyper-plane that maps x to y.

$d = 1$

$n = 3$



Pred: $\hat{y}(x^*) = \theta_0 + \theta_1 x^*$

Least squares

$$\min \left\{ (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 \right\}$$

$$= \min_{\theta_0, \theta_1} \left\{ (y_1 - \theta_0 - \theta_1 x_1)^2 + (y_2 - \theta_0 - \theta_1 x_2)^2 + (y_3 - \theta_0 - \theta_1 x_3)^2 \right\}$$

Learning and prediction with least squares

Learning

Given $x_{1:n}, y_{1:n} \Longrightarrow \hat{\Theta}$

Prediction

$x^* \longrightarrow \boxed{\hat{y} = \Theta x^*} = \hat{y}(x^*)$

in d dimensions

$1 \rightarrow \theta_0$
$x_1 \rightarrow \theta_1$
$\vdots \rightarrow \theta$
$x_d \rightarrow \theta_d$

$\longrightarrow \hat{y} = 1\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots$

Least squares

Mathematically, the linear model is expressed as follows:

$i = 1, 2, \cdots, n$

$$\begin{bmatrix} x_1, y_1 \\ \vdots \\ x_n, y_n \end{bmatrix}$$

$$\hat{y}_i = 1\theta_0 + \sum_{j=1}^{d} x_{ij}\theta_j$$

eg. i ≡ index over images

j ≡ index over features

eg. (r, g, b)

d = 3

We let $x_{i,0} = 1$ to obtain $\boxed{\hat{y}_i = \sum_{j=0}^{d} x_{ij}\theta_j}$

In matrix form, this expression is $\boxed{\hat{Y} = X\theta}$

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{10} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots \\ x_{n0} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

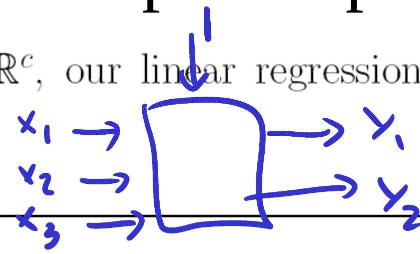
$y_1 = x_1\theta_1 + \theta_0$

$y_1 = x_1\theta_1 + 1\theta_0$

$y_1 = \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$

Least squares with multiple outputs

If we have several outputs $y_i \in \mathbb{R}^c$, our linear regression expression becomes:



$$Y = X\Theta \qquad o = 2,\; d = 3$$

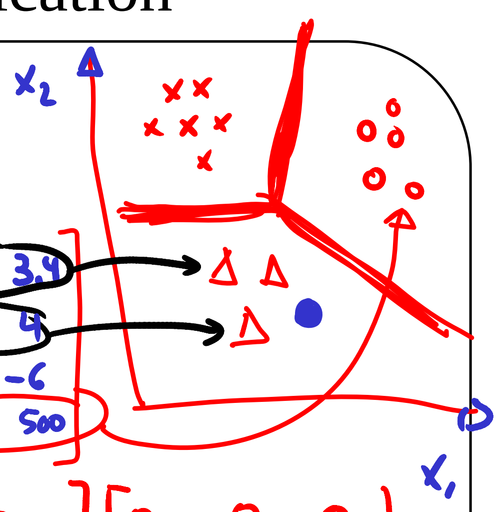
$$\underbrace{\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ \vdots & \\ y_{n1} & y_{n2} \end{bmatrix}}_{n \times o} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & & & & \vdots \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}}_{n \times (d+1)} \underbrace{\begin{bmatrix} \theta_{01} & \theta_{02} \\ \theta_{11} & \theta_{12} \\ \vdots & \\ \theta_{d1} & \theta_{d2} \end{bmatrix}}_{(d+1)\, o}$$

Linear classification

$$\underset{n\times 3}{\hat{Y}} = \underset{n\times 3}{X}\,\underset{3\times 3}{\Theta}$$

data:

$$Y = \begin{array}{c} \begin{array}{ccc} \Delta & x & o \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad X = \begin{bmatrix} 1.2 & 3.4 \\ 1.3 & 4 \\ -2 & -6 \\ -20 & 500 \end{bmatrix}$$



$$\hat{Y} = X\Theta = \begin{bmatrix} 1 & 1.2 & 3.4 \\ 1 & 1.3 & 4 \\ & \vdots & \end{bmatrix} \begin{bmatrix} \theta_{10} & \theta_{11} & \theta_{12} \\ \theta_{20} & \theta_{21} & \theta_{22} \\ \theta_{30} & \theta_{31} & \theta_{32} \end{bmatrix}$$

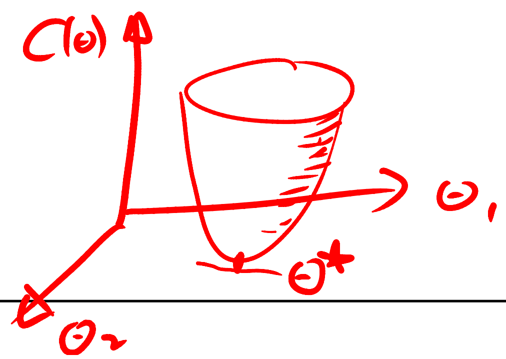
$$\hat{y}(\bullet) = [0.8 \quad 0.2 \quad 0] \xrightarrow{\text{argmax}} [1 \quad 0 \quad 0] \;\; \Delta$$

Optimization approach

Our aim is to mininimise the quadratic cost between the output labels and the model predictions

$$C(\theta) = (Y - X\theta)^T (Y - X\theta)$$

find θ^* (minimum)



Optimization approach

We will need the following results from matrix differentiation: $\frac{\partial A\theta}{\partial \theta} = A^T$ and $\frac{\partial \theta^T A \theta}{\partial \theta} = 2A^T\theta$ EXAM

$$\frac{\partial C}{\partial \theta} = \frac{\partial}{\partial \theta}(Y - X\theta)^T(Y - X\theta)$$

$$= \frac{\partial}{\partial \theta}\left(Y^T Y - 2\underbrace{Y^T X \theta}_{1\times 1} + \theta^T X^T X \theta\right)$$

$$= 0 - 2X^T Y + 2X^T X\theta \to 0$$

$$\boxed{\hat{\theta} = (X^T X)^{-1} X^T Y}$$

Optimization approach

θ has $d+1$ parameters. For simplicity, I will say d.

These are the **normal equations**. The solution (estimate) is:

$$\underbrace{\widehat{\theta}}_{d\times 1} = \underbrace{(\underbrace{X^T}_{d\times n}\underbrace{X}_{n\times d})^{-1}}_{d\times d}\underbrace{\underbrace{X^T}_{d\times n}\underbrace{Y}_{n\times 1}}_{d\times 1}$$

The corresponding predictions are

$\hat{Y} = X_k\hat{\Theta}$ $\quad 1\times d$

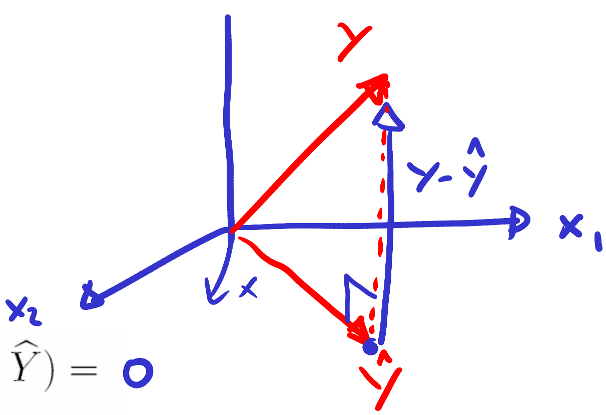
$$\underbrace{\widehat{Y}_k}_{1\times 1} = HY = \underbrace{X_k(X^TX)^{-1}X^T}_{H}Y$$

where H is the "hat" matrix.

X_k is a new point for which I want a prediction.

Geometric approach

$\hat{Y} = X\Theta$



$X^T(Y - \widehat{Y}) = 0$

$X^TY - X^T\hat{Y} = 0$

$X^TY - X^TX\Theta = 0$

$\Theta = (X^TX)^{-1}X^TY$

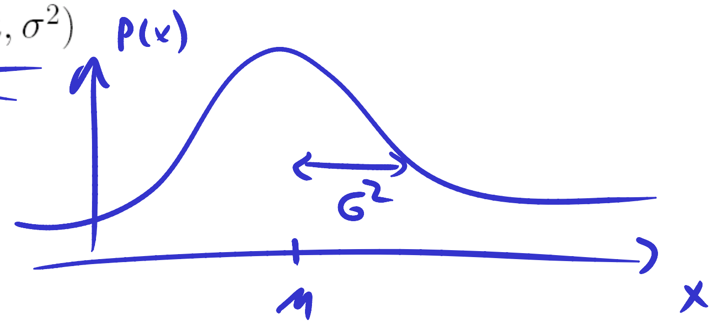
Probability approach: Univariate Gaussian distribution

The probability density function of a Gaussian distribution is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$

where μ is the mean or center of mass and σ^2 is the variance.

$X \sim \mathcal{N}(\mu, \sigma^2)$

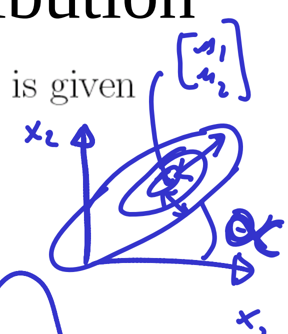


Multivariate Gaussian distribution

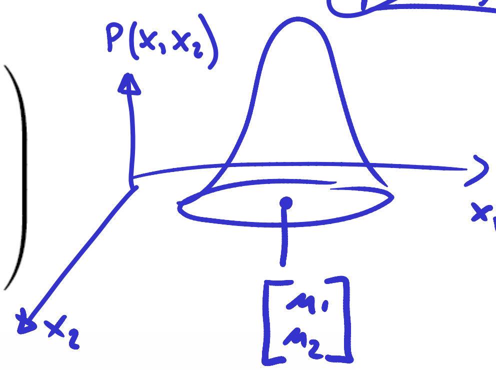
Let $x \in \mathbb{R}^n$. The pdf of an n-dimensional Gaussian is given by

$$\Sigma = \begin{bmatrix} \sigma_{11} & \rho \\ \rho^T & \sigma_{22} \end{bmatrix}$$

$$p(x) = \frac{1}{2\pi^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$



$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} \mathbb{E}(x_1) \\ \vdots \\ \mathbb{E}(x_n) \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} \sigma_{11} \cdots \sigma_{1n} \\ \cdots \\ \sigma_{n1} \cdots \sigma_{nn} \end{pmatrix} = \mathbb{E}[(X-\mu)(X-\mu)^T]$$

$$\sigma_{ij} = \mathbb{E}[X_i - \mu_i)(X_j - \mu_j)^T]$$

Multivariate Gaussian distribution

We can interpret each component of x, for example, as a feature of an image such as colour or texture. The term $\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$ is called the **Mahalanobis distance**. Conceptually, it measures the distance between x and μ.

$$P(x) = |2\pi\Sigma|^{-1/2} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

$$-\log P(x) = \text{const} + \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$$

If $\Sigma = I$:

$$-\log P(x) = \text{const} + \frac{1}{2}(x-\mu)^T(x-\mu)$$

Maximum likelihood approach

If our errors are Gaussian distributed, we can use the model

$$Y = X\theta + \mathcal{N}(0, \sigma^2 I)$$

$$|\sigma^2 I_n|^{-1/2} = (\sigma^2)^{-\frac{n}{2}}$$

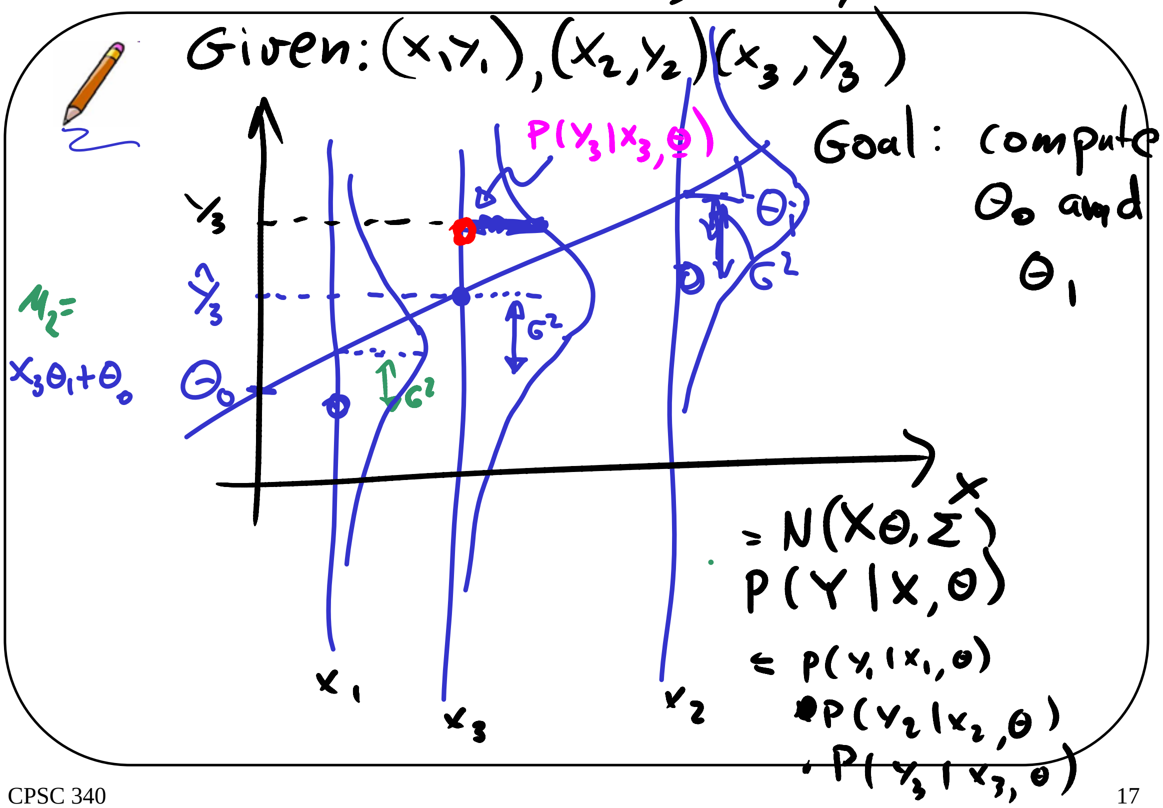
Note that the mean of Y is $X\theta$ and that its variance is $\sigma^2 I$. So we can equivalently write this expression using the probability density of Y **given** X, θ and σ: $\Sigma = \sigma^2 I$

$[y_1 \ y_2 \ y_3]^T$ height r.v.

$$p(Y|X,\theta,\sigma) = \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)}$$

$\bar{\eta}$ time

Maximum ~~likelihood~~ independent



Maximum likelihood

The maximum likelihood (ML) estimate of θ is obtained by taking the derivative of the log-likelihood, $\log p(Y|X, \underline{\theta, \sigma})$. The idea of maximum likelihood learning is to maximise the likelihood of seeing some data Y by modifying the parameters (θ, σ).

Maximum likelihood

$$P(y|x,\theta,\sigma^2) = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}(y-x\theta)^T(y-x\theta)}$$

The ML estimate of θ is:

$=$

$$\mathcal{L}(\theta) = \log P(y|x,\theta,\sigma^2) = \underbrace{-\frac{n}{2}\log(2\pi\sigma^2)}_{\text{Const. (ind. of } \theta)} - \frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = -\frac{1}{2\sigma^2}\frac{\partial}{\partial \theta}\left[(Y-X\theta)^T(Y-X\theta)\right] \to 0$$

$=$

$$\boxed{\hat{\theta}_{LS} = \hat{\theta}_{ML} = (X^T X)^{-1} X^T Y}$$

Maximum likelihood

Proceeding in the same way, the ML estimate of σ is:

$$\mathcal{L}(\sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)$$

$$\frac{\partial \mathcal{L}(\sigma^2)}{\partial \sigma} = -\frac{n}{\cancel{2}}\,\frac{\cancel{2}\,\cancel{2\pi}\,\sigma}{\cancel{2\pi}\,\sigma^{\cancel{2}}} + \frac{\cancel{2}}{\cancel{2}\sigma^3}(Y-X\theta)^T(Y-X\theta) \to 0$$

$$\frac{n}{\sigma} = \frac{1}{\sigma^3}(Y-X\theta)^T(Y-X\theta)$$

$$\sigma^2 = \frac{1}{n}(Y-X\theta)^T(Y-X\theta)$$

Lecture 8:
Regularization and ridge regression

Nando de Freitas
www.cs.ubc.ca/~nando/340-2009/
September 2009

All the answers so far are of the form

$$\widehat{\theta} = (X^{T}X^{\cancel{T}})^{-1}X^{T}Y$$

They require the inversion of XX^T. This can lead to problems if the system of equations is poorly conditioned. A solution is to add a small element to the diagonal:

$$\widehat{\theta} = (X^{T}X^{\cancel{T}} + \delta^2 I_d)^{-1}X^{T}Y$$

This is the ridge regression estimate. It is the solution to the following **regularised quadratic cost function**

$$C(\theta) = (Y - X\theta)^T(Y - X\theta) + \delta^2\theta^T\theta$$

$$① \quad \frac{\partial Ax}{\partial x} = A^T$$

Proof

$$\frac{\partial x^TAx}{\partial x} = 2A^Tx$$

$$C(\theta) = (Y - X\theta)^T(Y - X\theta) + \delta^2\theta^T\theta$$

$$C(\theta) = Y^TY - Y^TX\theta - \theta^TX^TY + \theta^TX^TX\theta + \delta^2\theta^TI\theta$$

$$= Y^TY - 2Y^TX\theta + \theta^T(X^TX + \delta^2 I)\theta$$

$$\frac{\partial C(\theta)}{\partial \theta} = 0 - (2Y^TX)^T + 2(X^TX + \delta^2 I)^T\theta$$

$$= -2X^TY + 2(X^TX + \delta^2 I)\theta$$

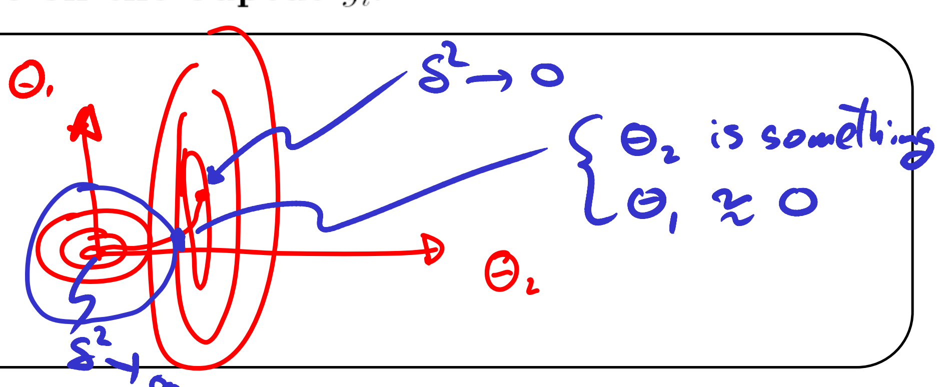
Equate to zero:

$$(X^TX + \delta^2 I)\theta = X^TY$$

Ridge as constrained optimization

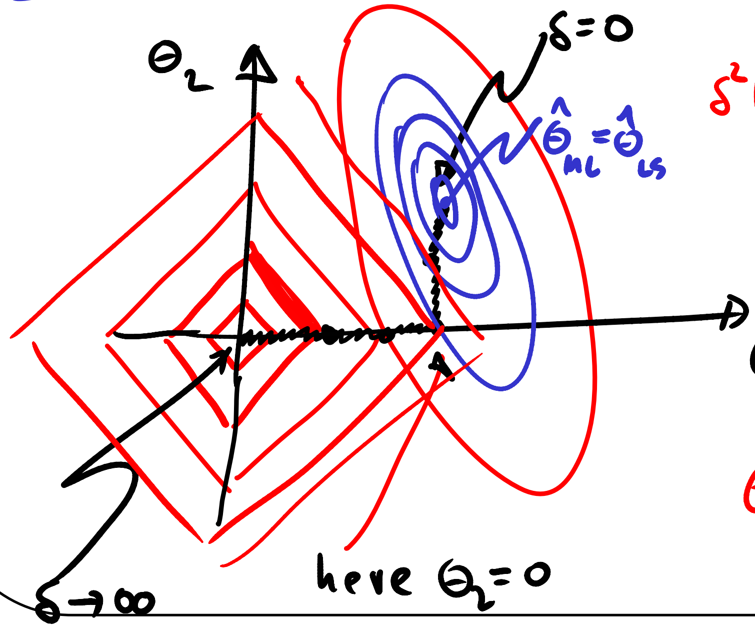
$$\min_{\theta:\ \theta^T\theta \le t} \{(Y - X\theta)^T(Y - X\theta)\}$$

Large values of θ are penalised. We are **shrinking** θ towards zero. This can be used to carry out **feature weighting**. **An input $x_{i,d}$ weighted by a small θ_d will have less influence on the ouptut y_i.**



The Lasso

$$\min_\theta \underbrace{(y - X\theta)^T(y - X\theta)}\ + \delta^2 \sum_i |\theta_i|$$



in 2D

$\delta^2(|\theta_1| + |\theta_2|) = \text{const}$

$-\theta_1 \quad -\theta_2 = \text{const}$

$-\theta_1 \quad +\theta_2 = \text{const}$

$+\theta_1 \quad -\theta_2 = \text{const}$

$+\theta_1 \quad +\theta_2 = \text{const}$

$\theta_2 = -\theta_1 + \text{const}$

Spectral view of ridge regression

Again, let $X \in \mathbb{R}^{n \times d}$ be factored as $X = U\Sigma V^T = \sum_{i=1}^{d} u_i \sigma_i v_i^T$,

The least squares prediction is: $\widehat{Y}_{LS} = \sum_{i=1}^{d} u_i u_i^T Y$

$$(X^T X)\theta = X^T y$$

$$V\Sigma U^T U \Sigma V^T \theta = V\Sigma U^T y$$

$$\cancel{V}\Sigma^{\cancel{2}} V^T \theta = \cancel{V}\cancel{\Sigma} U^T y$$

$$\boxed{\theta = V\Sigma^{-1}U^T y} \quad \star$$

$$\hat{y} = X\theta = UU^T y$$

Likewise, for ridge regression we have:

$$\widehat{Y}_{ridge} = \sum_{i=1}^{d} \frac{\sigma_i^2}{\sigma_i^2 + \delta^2} u_i u_i^T Y$$

in terms of SVD, what is

$\hat{\Theta}_{ridge} = ?$

Regularization and noise filtering

The filter factor

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \delta^2}$$

penalises small values of σ^2 (they go to zero at a faster rate).

Regularization and noise filtering

Small eigenvectors tend to be wobbly. The Ridge filter factor f_i gets rid of the wobbly eigenvectors. Therefore, the predictions tend to be more stable (smooth, regularised).

The smoothness parameter δ^2 is often estimated by cross-validation or Bayesian hierarchical methods.

$\hat{y} = X\theta$

Minimax and cross-validation

	δ^2	Error train $\sum_{i \in train}(y_i - \hat{y}_i)^2$	Error test	max	min max	avg.
	0.1	100	2	100		
16	1	10	11	11	x	
6	10	1	19	19		x
	50	20	0	20		x
	100	100	1000	1000		

min max $\delta = 1$ || best avg $\delta = \{10, 50\}$

Lecture 9:
Bayesian learning for linear models

Nando de Freitas
www.cs.ubc.ca/~nando/340-2009/
September 2009

Bayesian linear-Gaussian supervised learning

In the Bayesian linear prediction setting, we focus on computing the posterior:

$$\begin{aligned} p(\theta|X,Y) &\propto p(Y|X,\theta)p(\theta) \\ &= \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} p(\theta) \end{aligned}$$

We often want to maximise the posterior — that is, we look for the *maximum a poteriori* (MAP) estimate. In this case, the choice of prior determines a type of constraint! For example, consider a Gaussian prior $\theta \sim \mathcal{N}(0, \delta^2\sigma^2 I_d)$. Then

$$p(\theta|X,Y) \propto \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} \left(2\pi\sigma^2\delta^2\right)^{-\frac{d}{2}} e^{-\frac{1}{2\delta^2\sigma^2}\theta^T\theta}$$

$$
\begin{aligned}
p(\theta|X,Y) &= \left|2\pi\sigma^2 M\right|^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^T M^{-1}(\theta-\mu)} \\
&\propto \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} \left(2\pi\sigma^2\delta^2\right)^{-\frac{d}{2}} e^{-\frac{1}{2\delta^2\sigma^2}\theta^T\theta}
\end{aligned}
$$

Bayesian posterior

So the posterior for θ is Gaussian:

$$p(\theta|X,Y) = \left|2\pi\sigma^2 M\right|^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^T M^{-1}(\theta-\mu)}$$

with **sufficient statistics**:

$$\mathbb{E}(\theta|X,Y) = (XX^T + \delta^{-2}I_d)^{-1}X^TY$$
$$var(\theta|X,Y) = (XX^T + \delta^{-2}I_d)^{-1}\sigma^2$$

Bayesian estimates, ridge and ML

The MAP point estimate is:

$$\widehat{\theta}_{MAP} = (XX^T + \delta^{-2}I_d)^{-1}X^TY$$

It is the same as the ridge estimate (except for a trivial negative sign in the exponent of δ), which results from the L_2 constraint. A flat ("vague") prior with large variance (large δ) leads to the ML estimate.

$$\widehat{\theta}_{MAP} = \widehat{\theta}_{ridge} \quad \xrightarrow{\delta^2\to 0} \quad \widehat{\theta}_{ML} = \widehat{\theta}_{SVD} = \widehat{\theta}_{LS}$$