Machine Learning

Nando de Freitas

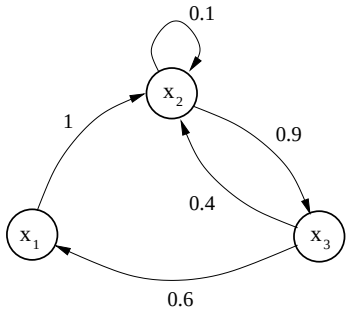
January 16, 2007

Lecture 2 - *Google's PageRank: Why math helps*

OBJECTIVE: Motivate linear algebra and probability as important and necessary tools for understanding large datasets. We also describe the algorithm at the core of the Google search engine.

◇ PAGERANK

Consider the following mini-web of 3 pages (the data):



The nodes are the webpages and the arrows are links. The

numbers are the normalised number of links. We can re-write this directed graph as a **transition matrix**:

T is a **stochastic matrix**: its columns add up to 1, so that

$$T_{i,j} = P(x_j|x_i)$$

$$\sum_j T_{i,j} = 1$$

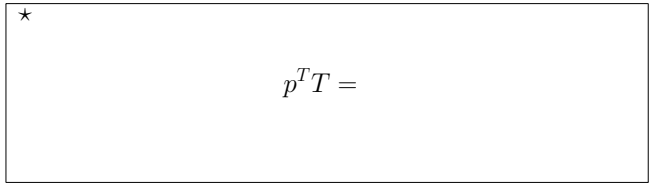
In information retrieval, we want to know the "relevance" of each webpage. That is, we want to compute the probability of each webpage: $p(x_i)$ for $i = 1, 2, 3$.

Let's start with a random guess $\pi = (0.5, 0.2, 0.3)^T$ and "crawl the web" (multiply by T several times). After, say

$N = 100$, iterations we get:

$$\pi^T T^N = (0.2, 0.4, 0.4)$$

We soon notice that no matter what initial π we choose, we always converge to $p = (0.2, 0.4, 0.4)^T$. So



The distribution p is a measure of the relevance of each page. Google uses this. But will this work always? When does it fail?

The **Perron-Frobenius Theorem** tell us that for any starting point, the chain will converge to the invariant distribution p, as long as T is a stochastic transition matrix that obeys the following properties:

1. **Irreducibility**: For any state of the Markov chain, there is a positive probability of visiting all other states.

   That is, the matrix T cannot be reduced to separate smaller matrices, which is also the same as stating that the transition graph is connected.

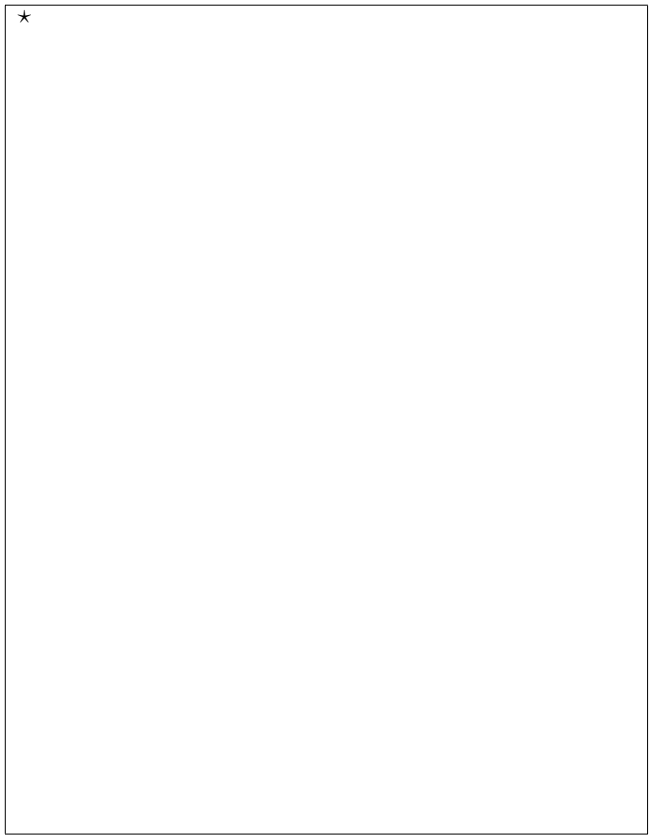
2. **Aperiodicity**: The chain should not get trapped in cycles.

Google's strategy is to add am matrix of uniform noise E to T:

$$L = T + \epsilon E$$

where ϵ is a small number. L is then normalised. This ensures irreducibility.

How quickly does this algorithm converge? What determines the rate of convergence? Again matrix algebra and spectral theory provide the answers:



Lecture 3 - *The Singular Value Decomposition (SVD)*

OBJECTIVE: The SVD is a matrix factorization that has many applications: e.g., information retrieval, least-squares problems, image processing.

◇ EIGENVALUE DECOMPOSITION

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$. If we put the eigenvalues of $\mathbf{A}$ into a diagonal matrix $\mathbf{\Lambda}$ and gather the eigenvectors into a matrix $\mathbf{X}$, then the eigenvalue decomposition of $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}.$$

But what if $\mathbf{A}$ is not a square matrix? Then the SVD comes to the rescue.

◇ FORMAL DEFINITION OF THE SVD

Given $\mathbf{A} \in \mathbb{R}^{m \times n}$, the SVD of $\mathbf{A}$ is a factorization of the form

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{u}$ are the left **singular vectors**, σ are the **singular values** and $\mathbf{v}$ are the right singular vectors.

$\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal with positive entries (singular values in the diagonal).
$\mathbf{U} \in \mathbb{R}^{m \times n}$ with orthonormal columns.
$\mathbf{V} \in \mathbb{R}^{n \times n}$ with orthonormal columns.
($\Rightarrow \mathbf{V}$ is orthogonal so $\mathbf{V}^{-1} = \mathbf{V}^T$)

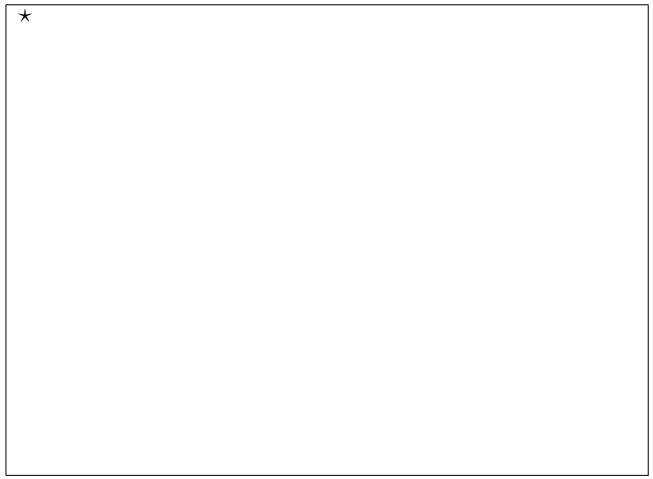
The equations relating the right singular values $\{\mathbf{v}_j\}$ and the left singular vectors $\{\mathbf{u}_j\}$ are

$$\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j \qquad j = 1, 2, \ldots, n$$

i.e.,

$$\mathbf{A} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

or $\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$.



1. There is no assumption that $m \geq n$ or that $\mathbf{A}$ has full rank.

2. All diagonal elements of $\mathbf{\Sigma}$ are non-negative and in non-increasing order:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$$

where $p = \min(m, n)$

Theorem 1 *Every matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ has singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$*
Furthermore, the singular values $\{\sigma_j\}$ are uniquely determined.
If $\mathbf{A}$ is square and $\sigma_i \neq \sigma_j$ for all $i \neq j$, the left singular vectors $\{\mathbf{u}_j\}$ and the right singular vectors $\{\mathbf{v}_j\}$ are uniquely determined to within a factor of ± 1.

◇ EIGENVALUE DECOMPOSITION

Theorem 2 *The nonzero singular values of $\mathbf{A}$ are the (positive) square roots of the nonzero eigenvalues of $\mathbf{A}^T\mathbf{A}$ or $\mathbf{A}\mathbf{A}^T$ (these matrices have the same nonzero eigenvalues).*

⋆ Proof:

◇ LOW-RANK APPROXIMATIONS

Theorem 3 $\|\mathbf{A}\|_2 = \sigma_1$, *where* $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|\neq 1} \|\mathbf{Ax}\|$.

⋆ Proof:

Another way to understand the SVD is to consider how a matrix may be represented by a sum of rank-one matrices.

Theorem 4

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where r *is the rank of* $\mathbf{A}$.

⋆ Proof:

What is so useful about this expansion is that the ν^{th} *partial sum captures as much of the "energy" of* $\mathbf{A}$ *as possible by a matrix of at most rank-*ν. In this case, "energy" is defined by the 2-norm.

Theorem 5 *For any* ν *with* $0 \leq \nu \leq r$ *define*

$$\mathbf{A}_\nu = \sum_{j=1}^{\nu} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

If $\nu = p = \min(m, n)$*, define* $\sigma_{\nu+1} = 0$.
Then,

$$\|\mathbf{A} - \mathbf{A}_\nu\|_2 = \sigma_{\nu+1}$$

Lecture 4 - *Fun with the SVD*

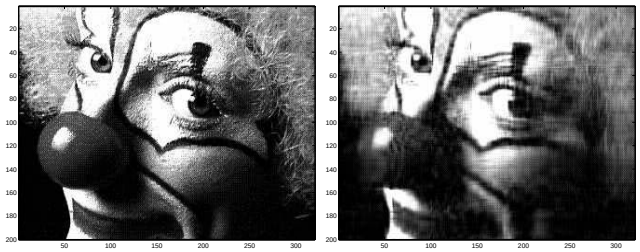
OBJECTIVE: Applications of the SVD to image compression, dimensionality reduction, visualization, information retrieval and latent semantic analysis.

◇ IMAGE COMPRESSION EXAMPLE

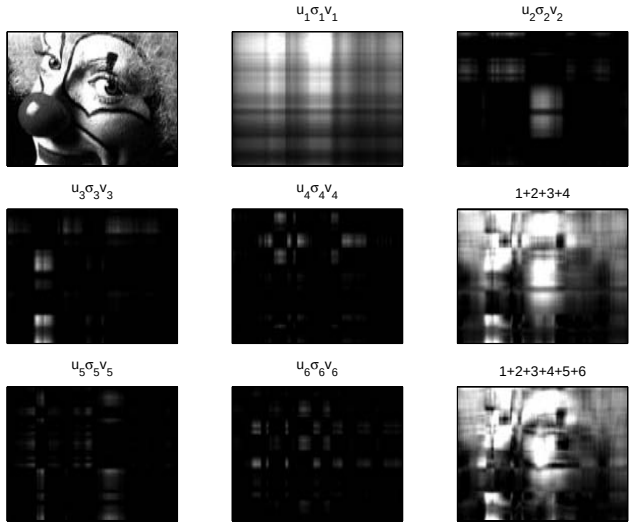
```
load clown.mat;
figure(1)
colormap('gray')
image(A);

[U,S,V] = svd(A);
figure(2)
k = 20;
colormap('gray')
image(U(:,1:k)*S(1:k,1:k)*V(:,1:k)');
```

The code loads a clown image into a 200×320 array $\mathbf{A}$; displays the image in one figure; performs a singular value decomposition on $\mathbf{A}$; and displays the image obtained from a rank-20 SVD approximation of $\mathbf{A}$ in another figure. Results are displayed below:



The original storage requirements for $\mathbf{A}$ are $200 \cdot 320 = 64,000$, whereas the compressed representation requires $(200+300+1) \cdot 20 \approx 10,000$ storage locations.



Smaller eigenvectors capture high frequency variations (small brush-strokes).

◇ TEXT RETRIEVAL - LSI

The SVD can be used to cluster documents and carry out information retrieval by using concepts as opposed to word-matching. This enables us to surmount the problems of synonymy (car,auto) and polysemy (money bank, river bank). The data is available in a term-frequency matrix

If we truncate the approximation to the k-largest singular values, we have

$$\mathbf{A} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

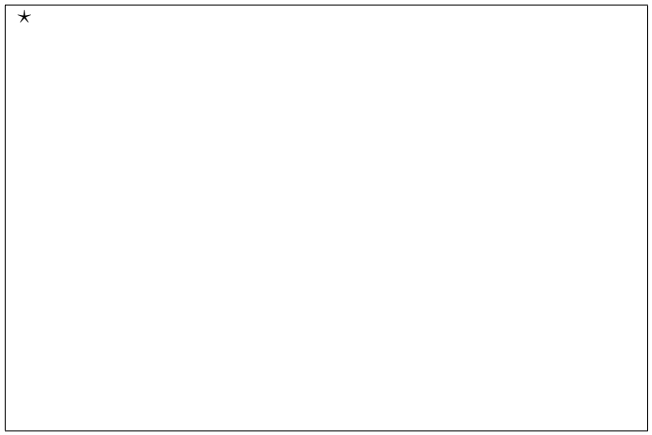
So

$$\mathbf{V}_k^T = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{A}$$

In English, $\mathbf{A}$ is projected to a lower-dimensional space spanned by the k singular vectors $\mathbf{U}_k$ (eigenvectors of $\mathbf{A}\mathbf{A}^T$). To carry out **retrieval**, a **query** $\mathbf{q} \in \mathbb{R}^n$ is first projected to the low-dimensional space:

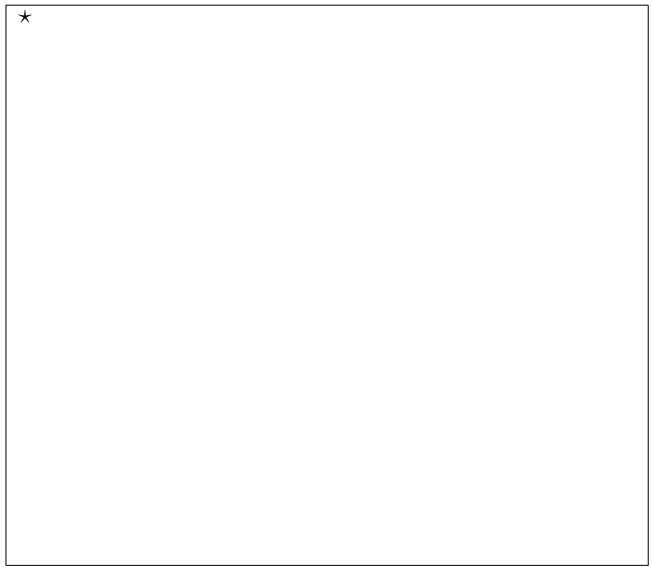
$$\widehat{\mathbf{q}}_k = \mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{q}$$

And then we measure the angle between $\widehat{\mathbf{q}}_k$ and the $\mathbf{v}_k$.

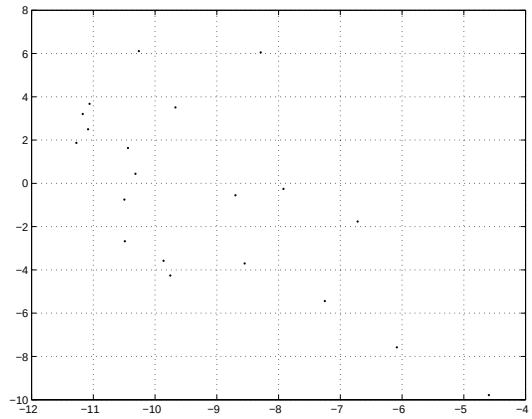


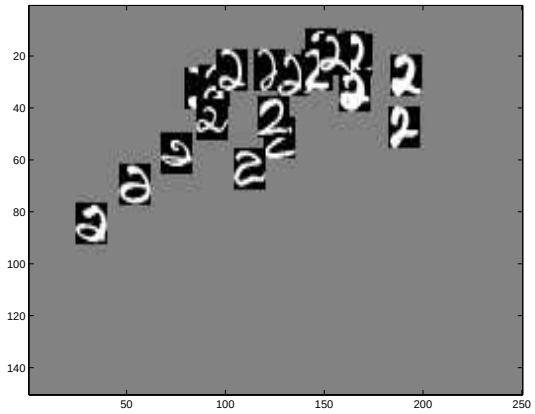
◇ PRINCIPAL COMPONENT ANALYSIS (PCA)

The columns of $\mathbf{U\Sigma}$ are called the **principal components** of $\mathbf{A}$. We can project high-dimensional data to these components in order to be able to visualize it. This idea is also useful for cleaning data as discussed in the previous text retrieval example.



For example, we can take several 16×16 images of the digit 2 and project them to 2D. The images can be written as vectors with 256 entries. We then from the matrix $\mathbf{A} \in \mathbb{R}^{n \times 256}$, carry out the SVD and truncate it to $k = 2$. Then the components $\mathbf{U}_k \mathbf{\Sigma}_k$ are 2 vectors with n data entries. We can plot these 2D points on the screen to visualize the data.





Lecture 5 - *Probability Revision*

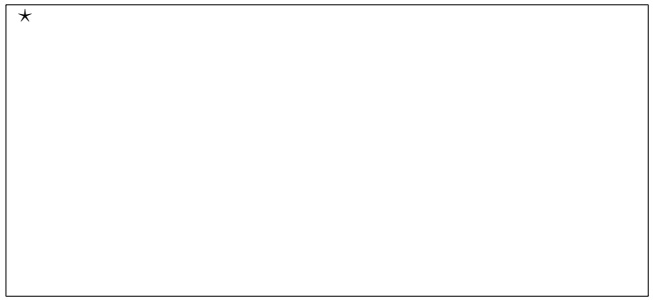
OBJECTIVE: Revise the fundamental concepts of probability, including marginalization, conditioning, Bayes rule and expectation.

◇ PROBABILITY

Probability theory is the formal study of the laws of chance. It is our tool for dealing with uncertainty. Notation:

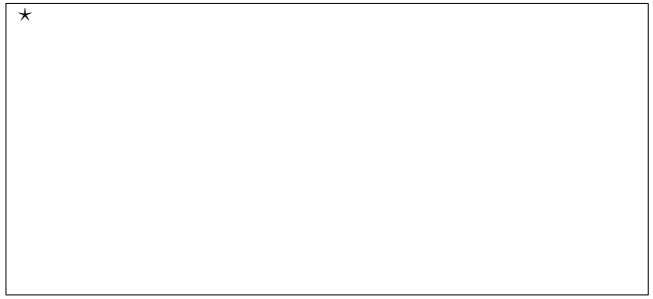
- **Sample space:** is the set Ω of all outcomes of an experiment.
- **Outcome:** what we observed. We use $\omega \in \Omega$ to denote a particular outcome. *e.*g. for a die we have $\Omega = \{1, 2, 3, 4, 5, 6\}$ and ω could be any of these six numbers.
- **Event:** is a subset of Ω that is well defined (measurable). *e.g.* the event $A = \{even\}$ if $w \in \{2, 4, 6\}$

Why do we need measure?



Frequentist Perspective

Let probability be the frequency of events.



Axiomatic Perspective

The frequentist interpretation has some shortcomings when we ask ourselves questions like

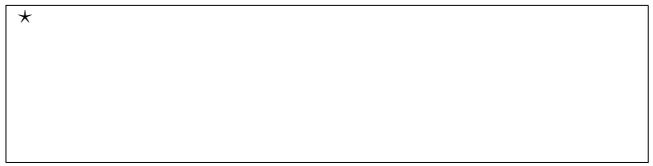
- *what is the probability that David will sleep with Anne?*
- *What is the probability that the Panama Canal is longer than the Suez Canal?*

The axiomatic view is a more elegant mathematical solution. Here, a **probabilistic model** consists of the triple $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the sigma-field (collection of measurable events) and P is a function mapping $\mathcal{F}$ to the interval $[0, 1]$. That is, with each event $A \in \mathcal{F}$ we associate a probability $P(A)$.

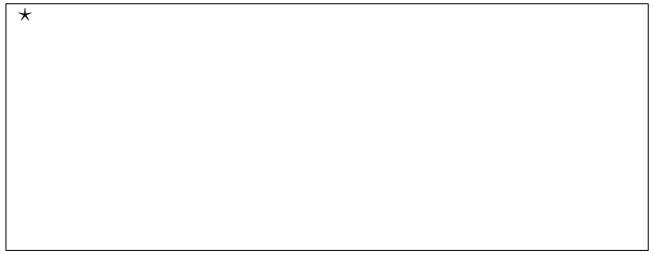
Some outcomes are not measurable so we have to assign probabilities to $\mathcal{F}$ and not Ω. Fortunately, in this course everything will be measurable so we need no concern ourselves with measure theory. We do have to make sure the following two axioms apply:

1. $P(\emptyset) = 0 \leq p(A) \leq 1 = P(\Omega)$
2. For **disjoint sets** A_n, $n \geq 1$, we have

$$P\left(\sum_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n)$$



If the sets overlap:



$$P(A + B) = P(A) + P(B) - P(AB)$$

If the events A and B are **independent**, we have $P(AB) = P(A)P(B)$.

⋆ Let $P(HIV) = 1/500$ be the probability of contracting HIV by having unprotected sex. If one has unprotected sex twice, the probability of contracting HIV becomes:

What if we have unprotected sex 500 times?

Conditional Probability

$$P(A|B) \triangleq \frac{P(AB)}{P(B)}$$

where $P(A|B)$ is the **conditional probability** of A given that B occurs, $P(B)$ is the **marginal probability** of B and $P(AB)$ is the **joint probability** of A and B. In general, we obtain a **chain rule**

$$P(A_{1:n}) = P(A_n|A_{1:n-1})P(A_{n-1}|A_{1:n-2}) \dots P(A_2|A_1)P(A_1)$$

⋆ Assume we have an urn with 3 red balls and 1 blue ball: $U = \{r, r, r, b\}$. What is the probability of drawing (without replacement) 2 red balls in the first 2 tries?

Marginalisation

Let the sets $B_{1:n}$ be disjoint and $\bigcup_{i=1}^{n} B_i = \Omega$. Then

$$P(A) = \sum_{i=1}^{n} P(A, B_i)$$

⋆ Proof:

⋆ What is the probability that the second ball drawn from our urn will be red?

Bayes Rule

Bayes rule allows us to reverse probabilities:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Combinining this with marginalisation, we obtain a powerful tool for statistical modelling:

$$P(model_i|data) = \frac{P(data|model_i)P(model_i)}{\sum_{j=1}^{M} P(data|model_j)P(model_j)}$$

That is, if we have **prior** probabilities for each model and generative data models, we can compute how likely each model is **a posteriori** (in light of our prior knowledge and the evidence brought in by the data).

Discrete random variables

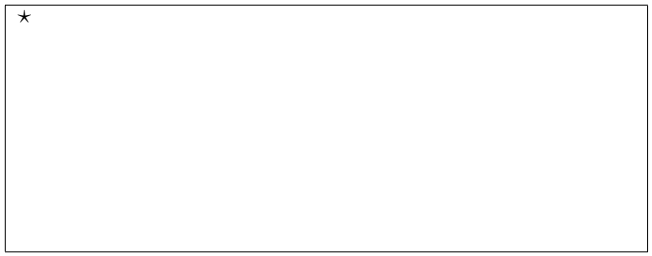
Let E be a discrete set, e.g. $E = \{0, 1\}$. A **discrete random variable** (r.v.) is a map from Ω to E:

$$X(w) : \Omega \mapsto E$$

such that for all $x \in E$ we have $\{w|X(w) \leq x\} \in \mathcal{F}$. Since $\mathcal{F}$ denotes the measurable sets, this condition simply says that we can compute (measure) the probability $P(X = x)$.

⋆ Assume we are throwing a die and are interested in the events $E = \{even, odd\}$. Here $\Omega = \{1, 2, 3, 4, 5, 6\}$. The r.v. takes the value $X(w) = even$ if $w \in \{2, 4, 6\}$ and $X(w) = odd$ if $w \in \{1, 3, 5\}$. We describe this r.v. with a **probability distribution** $p(x_i) = P(X = x_i) = \frac{1}{2}$, $i = 1, \ldots, 2$

The **cumulative distribution function** is defined as $F(x) = P(X \leq x)$ and would for this example be:



Bernoulli Random Variables

Let $E = \{0, 1\}$, $P(X = 1) = \lambda$, and $P(X = 0) = 1 - \lambda$.
We now introduce the *set indicator variable*. (This is a very useful notation.)

$$\mathbb{I}_A(w) = \begin{cases} 1 \;\; if & w \in A; \\ 0 \;\; otherwise. & \end{cases}$$

Using this convention, the probability distribution of a **Bernoulli** random variable reads:

$$p(x) = \lambda^{\mathbb{I}_{\{1\}}(x)}(1-\lambda)^{\mathbb{I}_{\{0\}}(x)}.$$

Expectation of Discrete Random Variables

The expectation of a discrete random variable X is

$$\mathbb{E}[X] = \sum_E x_i p(x_i)$$

The expectation operator is linear, so $\mathbb{E}(ax_1 + bx_2) = a\mathbb{E}(x_1) + b\mathbb{E}(x_2)$. In general, the expectation of a function $f(X)$ is

$$\mathbb{E}[f(X)] = \sum_E f(x_i)\, p(x_i)$$

Mean: $\mu \triangleq \mathbb{E}(X)$

Variance: $\sigma^2 \triangleq \mathbb{E}[(X - \mu)^2]$

⋆ For the set indicator variable $\mathbb{I}_A(\omega)$,

$$\mathbb{E}[\mathbb{I}_A(\omega)] =$$

Continuous Random Variables

A continuous r.v. is a map to a continuous space, $X(w) : \Omega \mapsto \mathbb{R}$, under the usual measurability conditions. The **cumulative distribution function** $F(x)$ (cdf) is defined by

$$F(x) \triangleq \int_{-\infty}^{x} p(y)\, dy = P(X \leq x)$$

where $p(x)$ denotes the **probability density function** (pdf). For an infinitesimal measure dx in the real line, distributions F and densities p are related as follows:

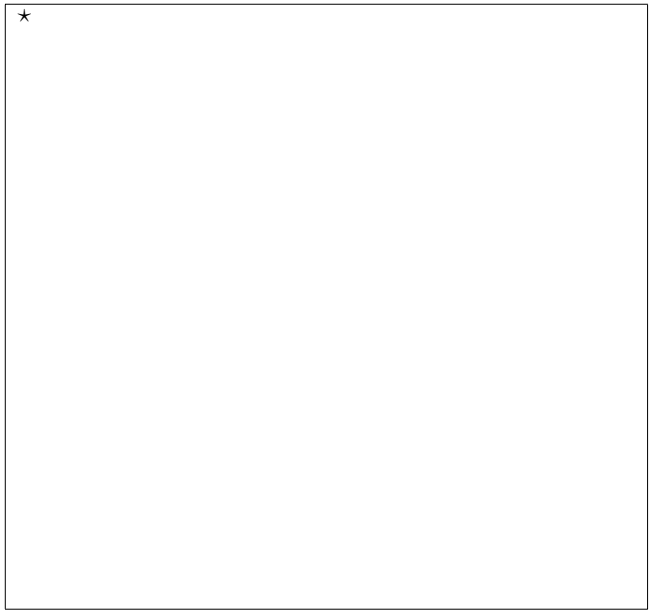
$$F(dx) = p(x)dx = P(X \in dx).$$

⋆

Univariate Gaussian Distribution

The pdf of a Gaussian distribution is given by

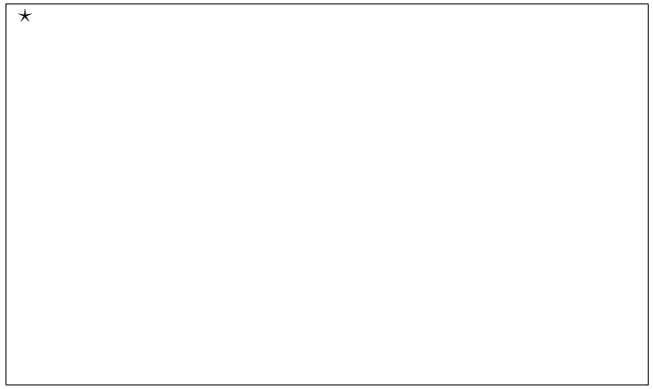
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$



Our short notation for Gaussian variables is $X \sim \mathcal{N}(\mu, \sigma^2)$.

Univariate Uniform Distribution

A random variable X with a uniform distribution between 0 to 1 is written as $X \sim \mathcal{U}_{[0,1]}(x)$



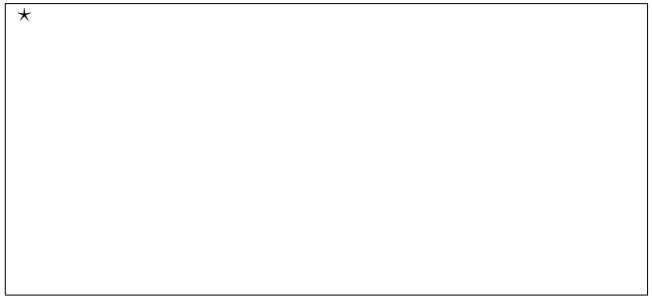
Multivariate Distributions

Let $f(u, v)$ be a pdf in 2-D. The cdf is defined by

$$F(x, y) = \int_{-\infty}^{x} \int_{-\infty}^{y} f(u, v)\, du\, dv = P(X \leq x, Y \leq y).$$

1 Bivariate Uniform Distribution

$$X \sim \mathcal{U}_{[0,1]^2}(x)$$



Multivariate Gaussian Distribution

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

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

where

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

and

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

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

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.

⋆ What is $\int \cdots \int e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}\, dx$?

Linear Operations

Let $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$ be given matrices, and $X \in \mathbb{R}^n$ be a random variable with mean $\mathbb{E}(X) = \mu_x \in \mathbb{R}^n$ and covariance $cov(X) = \Sigma_X \in \mathbb{R}^{n \times n}$. We define a new random variable

$$Y = AX + b$$

If $X \sim N(\mu_x, \Sigma_x)$, then $Y \sim N(\mu_y, \Sigma_y)$ where

⋆

$\mu_y = \mathbb{E}(Y) =$

$\Sigma_y =$

Finally, we define the **cross-covariance** as

$$\Sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)'].$$

X and Y are **uncorrelated** if $\Sigma_{XY} = 0$. So,

$$\Sigma = \begin{pmatrix} \Sigma_{XX} & 0 \\ 0 & \Sigma_{YY} \end{pmatrix}.$$

Lecture 6 - *Linear Supervised Learning*

OBJECTIVE: 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 ideas in machine learning.

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.

⋆

Mathematically, the linear model is expressed as follows:

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

We let $x_{i,0} = 1$ to obtain

$$\widehat{y}_i = \sum_{j=0}^{d} x_{ij}\theta_j$$

In matrix form, this expression is

$$\widehat{Y} = X\theta$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{10} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots \\ x_{n0} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

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

⋆

We will present several approaches for computing θ.

◇ OPTIMIZATION APPROACH

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

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

⋆

We will need the following result from matrix differentiation: $\frac{\partial A}{\partial \theta} = A^T$.

⋆

$$\frac{\partial C}{\partial \theta} =$$

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

$$\widehat{\theta} =$$

The corresponding predictions are

$$\widehat{Y} = HY =$$

where H is the "hat" matrix.

◇ GEOMETRIC APPROACH

⋆

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

Maximum Likelihood

⋆

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

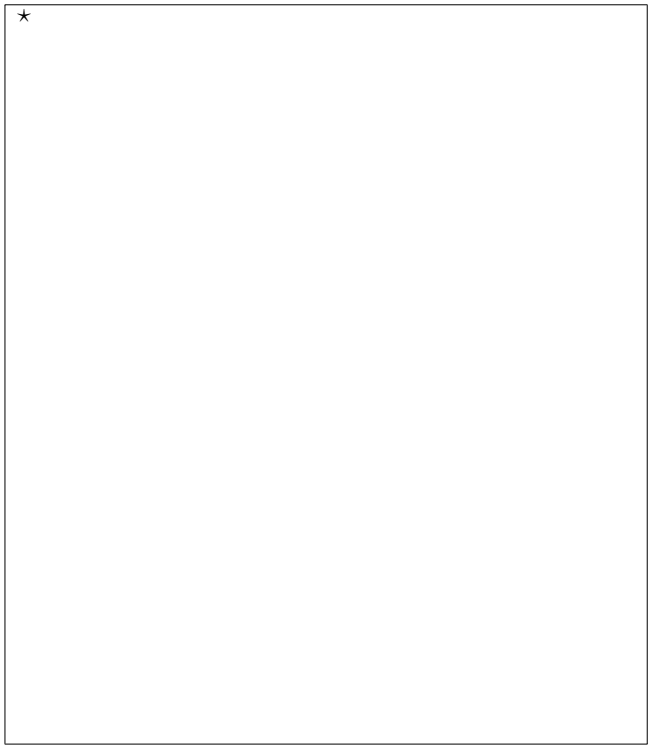
$$Y = X\theta + \mathcal{N}(0, \sigma^2 I)$$

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:

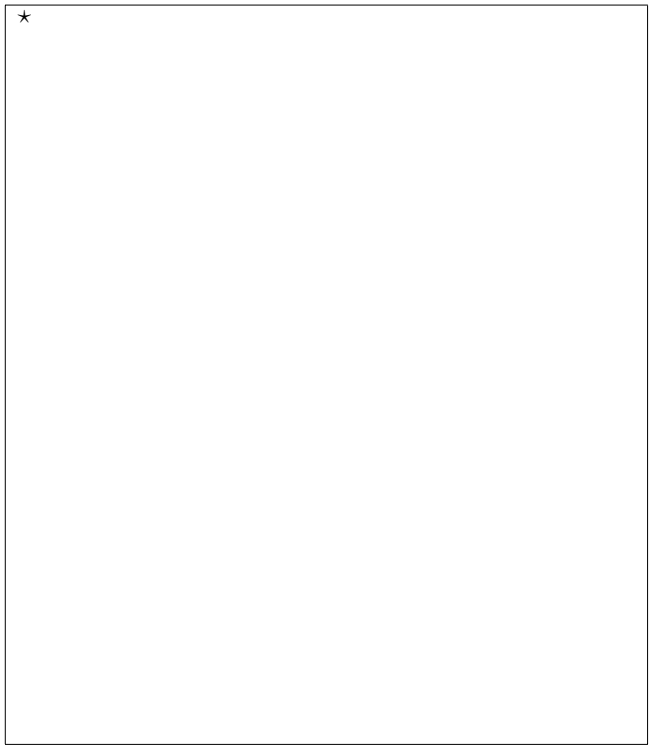
$$p(Y|X, \theta, \sigma) = \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)}$$

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

The ML estimate of θ is:



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



Lecture 7 - *Ridge Regression*

OBJECTIVE: Here we learn a cost function for linear supervised learning that is more stable than the one in the previous lecture. We also introduce the very important notion of **regularization**.

All the answers so far are of the form

$$\widehat{\theta} = (XX^T)^{-1}X^TY$$

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:

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

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

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

⋆ Proof:

It is useful to visualise the quadratic optimisation function and the constraint region.

⋆

That is, we are solving the following **constrained optimisation** problem:

$$\min_{\theta\,:\,\theta^T\theta \leq t} \left\{(Y - X\theta)^T(Y - X\theta)\right\}$$

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.**

Spectral View of LS and Ridge Regression

Again, let $X \in \mathbb{R}^{n \times d}$ be factored as

$$X = U\Sigma V^T = \sum_{i=1}^{d} u_i \sigma_i v_i^T,$$

where we have assumed that the rank of X is d.

⋆ The least squares prediction is:

$$\widehat{Y}_{LS} = \sum_{i=1}^{d} u_i u_i^T Y$$

⋆ Likewise, for ridge regression we have:

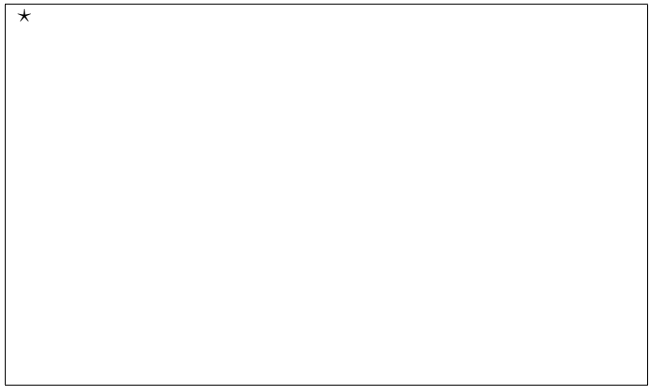
$$\widehat{Y}_{ridge} = \sum_{i=1}^{d} \frac{\sigma_i^2}{\sigma_i^2 + \delta^2} u_i u_i^T Y$$

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).

Also, by increasing δ^2 we are penalising the weights:

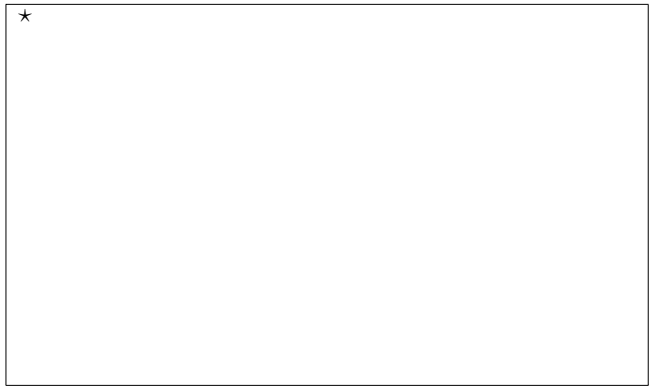


Small eigenvectors tend to be wobbly. The Ridge filter factor f_i gets rid of the wobbly eigenvectors. Therefore, the predictions tend to be more stable (smooth, regularised).

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

Minimax and cross-validation

Cross-validation is a widely used technique for choosing δ. Here's an example:



Lecture 8 - *Maximum Likelihood and Bayesian Learning*

OBJECTIVE: In this chapter, we revise maximum likelihood (ML) for a simple binary model. We then introduce Bayesian learning for this simple model and for the linear-Gaussian regression setting of the previous chapters. The key difference between the two approaches is that the frequentist view assumes there is one true model responsible for the observations, while the Bayesian view assumes that the model is a random variable with a certain prior distribution. Computationally, the ML problem is one of optimization, while Bayesian learning is one of integration.

◇ MAXIMUM LIKELIHOOD

Frequentist Learning assumes that there is a true model (say a parametric model with parameters θ_0). The estimate is denoted $\widehat{\theta}$. It can be found by maximising the **likelihood**:

⋆

$$\widehat{\theta} = \arg\max_{\theta} \quad p(x_{1:n}|\theta)$$

For **identical and independent distributed** (i.i.d.) data:

$$p(x_{1:n}|\theta) =$$

$$\mathcal{L}(\theta) = \log p(x_{1:n}|\theta) =$$

Let's illustrate this with a coin-tossing example.

⋆ Let $x_{1:n}$, with $x_i \in \{0, 1\}$, be i.i.d. Bernoulli:

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

With $m \triangleq \sum x_i$, we have

$$\mathcal{L}(\theta) =$$

Differentiating, we get

◇ BAYESIAN LEARNING

Given our **prior** knowledge $p(\theta)$ and the data **model** $p(\cdot|\theta)$, the Bayesian approach allows us to update our prior using the new data $x_{1:n}$ as follows:

$$p(\theta|x_{1:n}) = \frac{p(x_{1:n}|\theta)p(\theta)}{p(x_{1:n})}$$

where $p(\theta|x_{1:n})$ is the **posterior distribution**, $p(x_{1:n}|\theta)$ is the likelihood and $p(x_{1:n})$ is the **marginal likelihood** (evidence). Note

$$p(x_{1:n}) = \int p(x_{1:n}|\theta)p(\theta)d\theta$$

⋆

Bayesian Prediction

We predict by marginalising over the posterior of the parameters

$$\begin{aligned} p(x_{n+1}|x_{1:n}) &= \int p(x_{n+1}, \theta|x_{1:n})d\theta \\ &= \int p(x_{n+1}|\theta)p(\theta|x_{1:n})d\theta \end{aligned}$$

Bayesian Model Selection

For a particular model structure M_i, we have

$$p(\theta|x_{1:n}, M_i) = \frac{p(x_{1:n}|\theta, M_i)p(\theta|M_i)}{p(x_{1:n}|M_i)}$$

Models are selected according to their posterior:

$$P(M_i|x_{1:n}) \propto P(x_{1:n}|M_i)p(M_i) = P(M_i)\int p(x_{1:n}|\theta, M_i)p(\theta|M_i)d\theta$$

The ratio $P(x_{1:n}|M_i)/P(x_{1:n}|M_j)$ is known as the **Bayes Factor**.

⋆ Let $x_{1:n}$, with $x_i \in \{0, 1\}$, be i.i.d. Bernoulli: $x_i \sim \mathcal{B}(1, \theta)$

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} p(x_i|\theta) = \theta^m(1-\theta)^{n-m}$$

Let us choose the following **Beta** prior distribution:

$$p(\theta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

where Γ denotes the Gamma-function. For the time being, α and β are fixed **hyper-parameters**. The posterior distribution is proportional to:

$$p(\theta|x) \propto$$

with normalisation constant

Since the posterior is also Beta, we say that the Beta prior is **conjugate** with respect to the binomial likelihood. Conjugate priors lead to the same form of posterior.

Different hyper-parameters of the Beta $\mathcal{B}e(\alpha, \beta)$ distribution give rise to different prior specifications:

The generalisation of the Beta distribution is the Dirichlet distribution $\mathcal{D}(\alpha_i)$, with density

$$p(\theta) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

where we have assumed k possible thetas. **Note that the Dirichlet distribution is conjugate with respect to a Multinomial likelihood.**

◇ BAYESIAN LEARNING FOR LINEAR-GAUSSIAN MODELS

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

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

Our task is to rearrange terms in the exponents in order to obtain a simple expression for the posterior distribution.

⋆

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

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**:

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

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.

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

2 Full Bayesian Model

In Bayesian inference, we're interested in the full posterior:

$$p(\theta,\sigma^2,\delta^2|X,Y) \propto p(Y|\theta,\sigma^2,X)p(\theta|\sigma^2,\delta^2)p(\sigma^2)p(\delta^2)$$

where

$$\begin{aligned}Y|\theta,\sigma^2,X &\sim \mathcal{N}(X\theta,\sigma^2 I_n)\\ \theta &\sim \mathcal{N}\left(0,(\sigma^2\delta^2 I_d)\right)\\ \sigma^2 &\sim \mathcal{IG}\left(a/2,b/2\right)\\ \delta^2 &\sim \mathcal{IG}\left(\alpha,\beta\right)\end{aligned}$$

where $\mathcal{IG}\left(\alpha,\beta\right)$ denotes the **Inverse-Gamma distribution**.

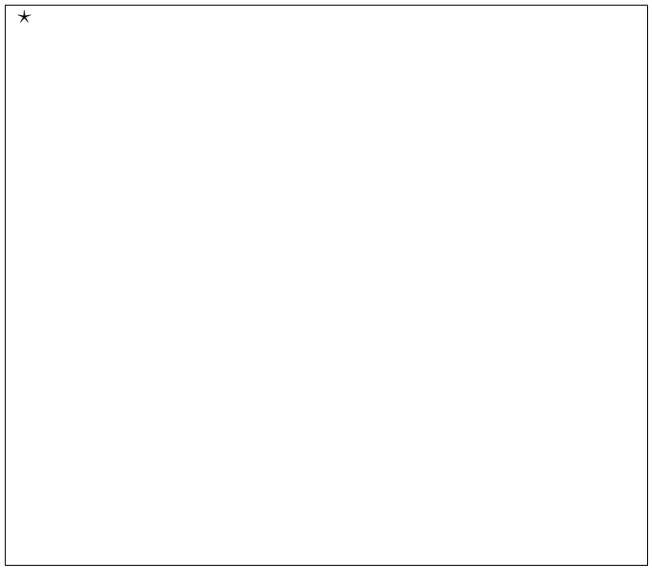
$$\delta^2 \sim \mathcal{IG}\left(\alpha,\beta\right) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta/\delta^2}(\delta^2)^{-\alpha-1}\mathbb{I}_{[0,\infty)}(\delta^2)$$

This is the conjugate prior for the variance of a Gaussian. The generalization of the Gamma distribution, i.e. the conjugate prior of a covariance matrix is the **inverse**

Wishart distribution $\Sigma \sim IW_d(\alpha, \alpha\Sigma^*)$, admitting the density

$$p(\Sigma|\alpha, \Sigma^*) \propto |\Sigma|^{-(\alpha+d+1)/2} \exp\{-(1/2)tr(\alpha\Sigma^*\Sigma^{-1})\}$$

We can visualise our hierarchical model with the following graphical model:



The product of likelihood and priors is:

$$\begin{aligned} p(\theta, \sigma^2, \delta^2|X, Y) \;\propto\; & \left(2\pi\sigma^2\right)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)} \\ & \times \left(2\pi\sigma^2\delta^2\right)^{-\frac{d}{2}} e^{-\frac{1}{2\delta^2\sigma^2}\theta^T\theta} \\ & \times (\sigma^2)^{-a/2-1} e^{-\frac{b}{2\sigma^2}} (\delta^2)^{-\alpha-1} e^{-\frac{\beta}{\delta^2}} \end{aligned}$$

We know from our previous work on computing the posterior for θ that:

$$\begin{aligned} p(\theta, \sigma^2, \delta^2|X, Y) \;\propto\; & (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}Y^TPY} \\ & \times (2\pi\sigma^2\delta^2)^{-d/2} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^TM^{-1}(\theta-\mu)} \\ & \times (\sigma^2)^{-a/2-1} e^{-\frac{b}{2\sigma^2}} (\delta^2)^{-\alpha-1} e^{-\frac{\beta}{\delta^2}} \end{aligned}$$

where

$$\begin{aligned} M^{-1} &= X^TX + \delta^{-2}I_d \\ \mu &= MX^TY \\ P &= I_n - XMX^T \end{aligned}$$

From this expression, it is now obvious that

$$p(\theta|\sigma^2, X, Y) = \mathcal{N}(\mu, \sigma^2 M)$$

Next, we integrate $p(\theta, \sigma^2, \delta^2|X, Y)$ over θ in order to get an expression for $p(\sigma^2, \delta^2|X, Y)$. This will allow us to get an expression for the marginal posterior $p(\sigma^2|X, Y)$.

⋆

$$p(\sigma^2|X, Y) \sim \mathcal{IG}\left(\frac{a+n}{2}, \frac{b+Y'PY}{2}\right)$$

Integrating over σ^2 gives us an expression for $p(\delta^2|X, Y)$

⋆

Unfortunately this is a nonstandard distribution, thus making it hard for us to come up with the normalizing constant. So we'll make use of the fact that we know θ and σ^2 to derive

a conditional distribution $p(\delta^2|\theta, \sigma^2, X, Y)$.

$\star$ We know that:

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

In summary, we can

- Obtain $p(\theta|\sigma^2, \delta^2, X, Y)$ analytically.
- Obtain $p(\sigma^2|\delta^2, X, Y)$ analytically.
- Derive an expression for $p(\delta^2|\theta, \sigma^2, X, Y)$.

Given δ^2, we can obtain analytical expressions for θ and σ^2. But, let's be a bit more ambitious. Imagine we could run the following sampling algorithm (known as the **Gibbs Sampler**)

⋆

1. LOAD data (X, Y).
2. Compute $X^T Y$ and $X^T X$.
3. Set, e.g., $a = b = 0$, $\alpha = 2$ and $\beta = 10$.
4. Sample $\delta^{2(0)} \sim \mathcal{IG}(\alpha, \beta)$.
5. FOR $i = 1$ to N:
   (a) Compute M, P and $\mu^{(i)}$ using $\delta^{2(i-1)}$.
   (b) Sample $\sigma^{2(i)} \sim \mathcal{IG}\left(\frac{a+n}{2}, \frac{b+Y'PY}{2}\right)$.
   (c) Sample $\theta^{(i)} \sim \mathcal{N}(\mu^{(i)}, \sigma^{2(i)} M)$.
   (d) Sample $\delta^{2(i)} \sim \mathcal{IG}\left(\frac{d}{2} + \alpha, \beta + \frac{\theta^{(i)T}\theta^{(i)}}{2\sigma^{2(i)}}\right)$.

We can use these samples in order to approximate the integrals of interest with Monte Carlo averages.

For example, the predictive distribution

$$p(y_{n+1}|X_{1:n+1}, Y) = \int p(y_{n+1}|\theta, \sigma^2, x_{n+1}) p(\theta, \sigma^2, \delta^2|X, Y) d\theta d\sigma^2 d\delta^2$$

can be approximated with:

$$\widehat{p}(y_{n+1}|X_{1:n+1}, Y) = \frac{1}{N} \sum_{i=1}^{N} p(y_{n+1}|\theta^{(i)}, \sigma^{2(i)}, x_{n+1})$$

That is,

⋆

$\widehat{p}(y_{n+1}|X_{1:n+1}, Y) =$

In the next lecture, we will derive the theory that justifies the use of this algorithm as well a many other **Monte Carlo** algorithms.