Machine Learning - Notes

Dom Hutchinson

December 17, 2019

Contents

General

Lecturer - Carl Henrik Ek
Course Website - `http://carlhenrik.com/COMS30007/`
Course Repo - `https://github.com/carlhenrikek/COMS30007`
Course Subreddit - `https://www.reddit.com/r/coms30007/`

1 Introduction

1.1 Motivation

Definition 1.1 - *Deductive Reasoning*
A method of reasoning in which the premieses are viewed as supplying all the evidence for the truth of the conclusion.

Definition 1.2 - *Inductive Reasoning*
A method of reasoning in which the premieses are viewed as supplying some evidence for the truth of the conclusion, rather than all the evidence. This allows for the conclusion of the *Inductive Reasoning* to be false.

Remark 1.1 - *Free-Lunch Theorem*
There are infinite number of hypotheses that perfectly explain the data. Adding a data point removes an infinite number of possibilities, but still leaves infinite possibilities.

Remark 1.2 - *The Task of Machine Learning*
When proposing to use machine learning on a task, one should consider the following questions:

i) How can we formulate beliefes ad assumptions mathematically?

ii) How can we connect our assumptions with data?

iii) How can we update our beliefs?

Remark 1.3 - *Useful Models are not always True*
Our goal is to understand realisations of a system. If we can then we can equate our model to the system. It is important to note thtat our model does not need to be perfectly true to be useful.

1.2 Probability Theory

Definition 1.3 - *Stochastic/Random Variable*
A variable whose value depends on outcomes of random phenomona.
e.g. $x \sim \mathcal{N}(0,1)$.

Definition 1.4 - *Probability Measure,*$\mathbb{P}$
A function with signature $\mathbb{P} : \mathcal{F} \to [0,1]$, where $\mathcal{F}$ is a sample space of rv X, and fulfils $\int_{-\infty}^{\infty} \mathbb{P}(x)dx = 1$.

Definition 1.5 - *Joint Probability Distribution*
A *Probability Measure* for multiple variables, $\mathbb{P} : X \times Y \to [0,1]$.
Let n_{ij} be the number of outcomes where $X = x_i$ and $Y = y_j$ then

$$\mathbb{P}(X = x_i, Y = y_j) = \frac{n_{ij}}{\sum_{i,j} n_{ij}}$$

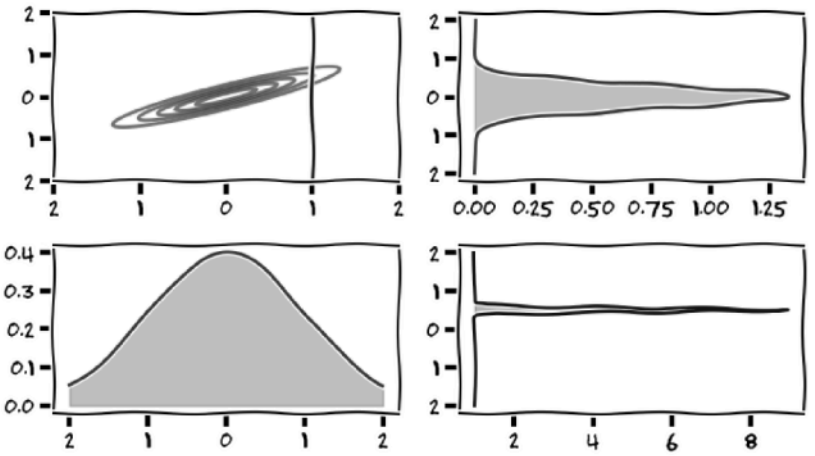
Definition 1.6 - *Marginal Probability Distribution*
A *Probability Measure* for one variable when the sample space is over multiple variables.
Let n_{ij} be the number of outcomes where $X = x_i$ and $Y = y_j$ then

$$\mathbb{P}(X = x_i) = \frac{\sum_j n_{ij}}{\sum_{i,j} n_{ij}}$$

Definition 1.7 - *Conditional Probability Distribution*
A *Probability Measure* for a variable, given another variable has a defined value. Let n_{ij} be the number of outcomes where $X = x_i$ and $Y = y_j$ then

$$\mathbb{P}(Y = y_j | X = x_i) = \frac{n_{ij}}{\sum_j n_{ij}}$$

Example 1.1 - *Joint, Marginal & Conditional Probability*
The below image shows two marginals distributions in the bottom-left, X, & top-right, Y, their joint distribution in the top-left and a conditional in the bottom right $\mathbb{P}(Y|X = 1)$.



Theorem 1.1 - *Product Rule*
For random variables X & Y

$$\mathbb{P}(X = x, Y = Y) = \mathbb{P}(Y = y | X = x)\mathbb{P}(X = x)$$

Theorem 1.2 - *Sum Rule*
For random variables X & Y

$$\mathbb{P}(X = x) = \sum_j \mathbb{P}(X = x, Y = y_j)$$

Theorem 1.3 - *Baye's Theorem*
For random variables X & Y

$$\mathbb{P}(X = x | Y = y) = \frac{\mathbb{P}(Y = y | X = x)\mathbb{P}(X = x)}{\mathbb{P}(Y = y)}$$

Definition 1.8 - *Elements of Bayes' Theorem*
The elements of *Bayes' Theory* can be broken down to explain parts of the model.

$$\underbrace{\mathbb{P}(\theta|Y)}_{\text{Posterior}} = \frac{\overbrace{\mathbb{P}(Y|\theta)}^{\text{Likelihood}}\overbrace{\mathbb{P}(\theta)}^{\text{Prior}}}{\underbrace{\mathbb{P}(Y)}_{\text{Evidence}}}$$

Posterior	Which parameters of the model do I belive produce distributions have generated the data Y
Likelihood	How likley is the data to come from the model specifely indexed by θ
Prior	What distribution do I think parameter θ has
Evidence	How likely do I think data Y is for all models.

N.B. The *Evidence* normalises this function.

Definition 1.9 - *Expectaction Value,* $\mathbb{E}$
The mean value a random variable will produce from a large number of samples.

Continuous	Discrete
$\mathbb{E}(X) = \int_{-\infty}^{\infty} x\mathbb{P}(X)dx$	$\mathbb{E}(X) = \sum_{-\infty}^{\infty} x\mathbb{P}(X)dx$
$\mathbb{E}(f(X)) = \int_{-\infty}^{\infty} f(x)\mathbb{P}(X)dx$	$\mathbb{E}(f(X)) = \sum_{-\infty}^{\infty} f(x)\mathbb{P}(X)dx$

Definition 1.10 - *Variance*
Describes the amount of spread in the values a single random variable will produce.

$$\text{var}(X) = \mathbb{E}\left(x - \mathbb{E}(x))^2\right) = \mathbb{E}(X^2) - \Big(\mathbb{E}(X)\Big)^2$$

Definition 1.11 - *Covariance*
Describes the joint variability between two random variables.

$$\text{cov}(X, Y) = \mathbb{E}\Big(\big(X - \mathbb{E}(X)\big)\big(Y - \mathbb{E}(Y)\big)\Big)$$

Definition 1.12 - *Marginalisation*
The process of summing out the probability of one random variable using its joing probability with another rando variable.

$$\begin{array}{rrcl} \text{Continuous} & \mathbb{P}(X = x) & = & \int (X = x, Y = y)dy \\ \text{Discrete} & \mathbb{P}(X = x) & = & \sum_i \mathbb{P}(X = x, Y = y_i) \end{array}$$

Definition 1.13 - *Likelihood Function*
Define $\mathbf{X} \sim f_n(\cdot; \theta^*)$ for some unknown $\theta^* \in \Theta$ and let $\mathbf{x}$ be an observation of $\mathbf{X}$.
A *Likelihood Function* is any function, $L(\cdot; \mathbf{x}) : \Theta \to [0, \infty)$, which is proportional to the PMF/PDF of the obeserved realisation $\mathbf{x}$.

$$L(\theta; \mathbf{x}) := C f_b(\mathbf{x}; \theta) \ \forall \ C > 0$$

N.B. Sometimes this is called the *Observed* Likelihood Function since it is dependent on observed data.

Definition 1.14 - *Log-Likelihood Function*
Let $\mathbf{X} \sim f_n(\cdot; \theta^*)$ for some unknown $\theta^* \in \Theta$ and $\mathbf{x}$ be an observation of $\mathbf{X}$.
The *Log-Likelihood Function* is the natural log of a *Likelihood Function*

$$\ell(\theta; \mathbf{x}) := \ln f_n(\mathbf{x}; \theta) + C, \ C \in \mathbb{R}$$

Definition 1.15 - *Maximum Likelihood Estimation*
The *Maximum Likelihood Estimate* is an estimate for a parameter of a probability distribution which is the value which maximises the *Likelihood Function* (or the *Likelihood Function*).

$$\hat{\theta} := \text{argmax}_\theta L(\theta; \mathbf{x})$$

Definition 1.16 - *Central Limit Theorem*
The distribution of the sum (or mean) of a large number of independent, identically distributied random variables can be approximated to a normal distribution, regardless of the distributions of the random variables.

1.3 Conjugate Priors

Definition 1.17 - *Conjugate Prior*
If we have a *Likelihood Function*, $\mathbb{P}(X|\theta)$, with a known distribution (*e.g.* Normal) we can choose our *Prior*, $\mathbb{P}(\theta)$, to be from a distribution which is *Conjugate* to the distribution of the *Likelihood Function*.
These are defined in *tables*

Remark 1.4 - *Why use Conjugate Priors?*
If we have a *Conjugate Prior* then the *Posterior*, $\mathbb{P}(\theta|X)$, will be in the same distribution family as the Prior too. We can then work out the distribution of the *Posterior* by passing the parameters of the *Prior* through pre-derived functions

$$\begin{array}{rcl} \text{Posterior} & \propto & \text{Likelihood} \times \text{Prior} \\ \mathbb{P}(\theta|X) & \propto & \mathbb{P}(X|\theta) \times \mathbb{P}(\theta) \end{array}$$

N.B. - `https://en.wikipedia.org/wiki/Conjugate_prior#Table_of_conjugate_distributions`

Example 1.2 - *Conjugate Priors*
Consider a scenario where we are flipping a coin. We may have *Likelihood Function* $\theta^x(1-\theta)^{n-x}$.
If we choose our *Prior* to be $\theta^{a-1}(1-\theta)^{b-1}$ which is a *Beta Distribution*.
Then (after some maths) we find the *Posterior*

2 Distributions

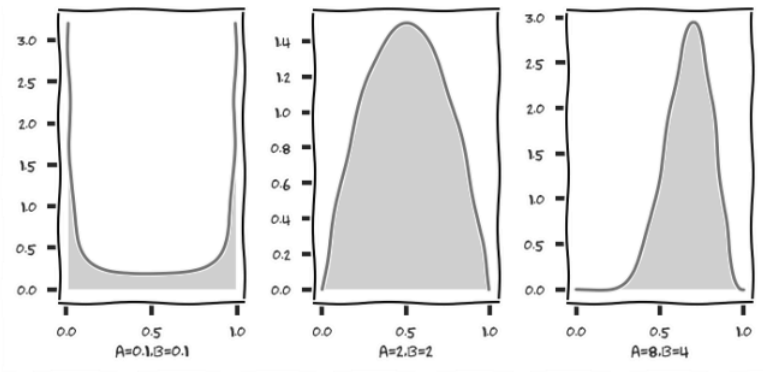
Definition 2.1 - *Bernoulli Distribution*
Models an event with a binary outcome (0 or 1) with parameter p st $\mathbb{P}(X=1)=p$ Let $X \sim \text{Bernoulli}(p)$. Then

$$\begin{array}{rcl} f_X(x) & = & \begin{cases} p & ,x=1 \\ 1-p & ,x=0 \\ 0 & \text{otherwise} \end{cases} \\ F_X(x) & = & \begin{cases} 0 & ,x<0 \\ 1-p & 0 \le x < 1 \\ 1 & x \ge 1 \end{cases} \\ \mathbb{E}(X) & = & p \\ \text{Var}(X) & = & p(1-p) \end{array}$$

Definition 2.2 - *β-Distribution*
A *β-Distribution* is a continuous distribution over interval $[0,1]$ which is parameterised by two positive *shape parameters*, α & β. A *β-Distribution* can be used to encode assumptions as a *Prior*.
Let $X \sim \beta(\alpha,\beta)$. Then

$$f_X(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

Example 2.1 - *β-Distribution*



Definition 2.3 - *Direchlet Distribution*
Let $X \sim \text{Dir}(\boldsymbol{\alpha})$. Then

$$f_X(x) := \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \times \cdots \times \Gamma(\alpha_N)} \prod_{i=1}^{N} x_i^{\alpha_{i-1}}$$

Definition 2.4 - *Exponential Distribution Family*
The *Exponential Distribution Family* is a set of probability distributions which fit the form.

$$\mathbb{P}(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x})g(\boldsymbol{\theta})e^{\boldsymbol{\theta}^T \mathbf{u}(\mathbf{x})}$$

With conjugate prior

$$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{\chi}, \nu) = f(\boldsymbol{\chi}, \nu)g(\boldsymbol{\chi})^{\nu} e^{\nu \boldsymbol{\theta}^T \boldsymbol{\chi}}$$

Definition 2.5 - *Multivariate Normal Distribution*
Let $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

N.B. Also known as *Gaussian Distribution.*

3 Regression

Definition 3.1 - *Supervised Learning*
Learning the relationship $f(\cdot)$ between pairs of data x_i and y_i where $y_i = f(x_i)$.

Remark 3.1 - *Summary of Regression*

Linear Regression We are limited to lines

unctions Regression We can use non-linear functions, but it is hard to determine how many & what basis functions should be used. The prior is hard to interpret.

Kernel Regression The complexity is defined by the data (Good) but there is no uncertainty in our estimate.

3.1 Linear Regression

Definition 3.2 - *Linear Regression*
Linear Regression is the process of taking a set of data points & producing a linear relationship between a dependent varaible & one of more explanatory variables.
Let $\mathbf{x} \in \mathbb{R}^n$ be a set of observed values from n explanatory variables & $\mathbf{a} \in \mathbb{R}^n + 1$ be a set of parameters. Then we predict the value of the dependent variable to be

$$y(\mathbf{x}, \mathbf{a}) = a_0 + \sum_{i=0}^{n} a_{i+1} x_i$$

Remark 3.2 - *Limitation of Linear Regression*
The formula defined in **Definition 3.1** is a linear function of the coefficients defined by **a and** the observed values of $\mathbf{x}$ this limits the relationships we can model between elements of $\mathbf{x}$. The model can be extended to avoid this using *Basis Functions*.

Definition 3.3 - *Linear Regression - Basis Functions*
We can extend *Linear Regression* to include *Basis Functions* so that relationships between explanatory variables can be modelled.
Let $\mathbf{x} \in \mathbb{R}^n$ be a set of observed values from explanatory variables, $\mathbf{a} \in \mathbb{R}^m$ be a set of coefficients (weightings), and $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^{m-1}$ be a set of basis functions. Then we can predict the dependent variable to be

$$y(\mathbf{x}, \mathbf{a}) = a_0 + \sum_{i=1}^{m} a_i \phi_{i-1}(\mathbf{x})$$

Remark 3.3 - *Linear Regression - Basis Function*
To simplfy the equation used in **Definition 3.2** we can define $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^m$ with $\phi_0(\mathbf{x})$. Then

$$y(\mathbf{x}, \mathbf{a}) = \sum_{i=0}^{m} a_i \phi_i(\mathbf{x}) = \mathbf{a}^T \boldsymbol{\phi}(\mathbf{x})$$

Proposition 3.1 - *Noise*
We will often introduce the concept of *Noise* into a *Linear Regression* model. Typically we assume noise to be modelled by a zero-mean Normal distribution with precision β, $\varepsilon \sim \text{Normal}(0, \beta^{-1})$, so

$$t := y(\mathbf{x}, \mathbf{a}) = \mathbf{a}^T \boldsymbol{\phi}(\mathbf{x}) + \varepsilon$$

From this we can derive a likelihood

$$\mathbb{P}(t|\mathbf{x}, \mathbf{a}, \beta) \sim \text{Normal}(t|\mu = y(\mathbf{x}, \mathbf{a}), \sigma^2 = \beta^{-1}) = \text{Normal}(t|\mu = \mathbf{a}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2 = \beta^{-1})$$

If we have a series of sets of observations, $\mathbf{X} \in \mathbb{R}^{m \times n}$, then

$$\mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta) \sim \prod_{i=1}^{m} \text{Normal}(t_i|\mu = \mathbf{a}^T \boldsymbol{\phi}(\mathbf{x}_i), \sigma^2 = \beta^{-1})$$

Definition 3.4 - *Maximum Likelihood Estimate*
A *Maximum Likelihood Estimate* is estimating the value of a parameter to be the most likely, according to our *Likelihood Function*.

Remark 3.4 - *Finding Maximum Likelihood Estimate*
Suppose we have a defined *Likelihood Function* $\mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta)$ and we want to find *Maximum Likelihood Estimates* for parameters $\mathbf{a}$. Then

i) Define the *Likelihood Function*, $\mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta)$.

ii) Take the natural log, $\ln \mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta)$.

iii) Take the derivative wrt $\mathbf{a}$, $\frac{\partial}{\partial \mathbf{a}} \ln \mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta)$.

iv) Set the derivatite to 0, $\frac{\partial}{\partial \mathbf{a}} \ln \mathbb{P}(\mathbf{t}|\mathbf{X}, \mathbf{a}, \beta) = 0$.

v) Solve to find the stationary point of $\mathbf{a}$.

vi) Check this stationary point is a maximum, if it is then it is a *Maximum Likelihood Estimate*

Example 3.1 - *Maximum Likelihood Estimate*
Here I shall find the *Maximum Likelihood Estimate* for $\mathbf{a}$

$$\begin{aligned}
\mathbb{P}(\mathbf{t}|\mathbf{X},\mathbf{a},\beta) &\sim \prod_{i=1}^{m} \text{Normal}(t_i|\mathbf{a}^T\boldsymbol{\phi}(\mathbf{x}_i),\beta^{-1}) \\
&= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\beta^{-1}}} e^{-\frac{1}{2}\beta(t_i-\mathbf{a}^T\boldsymbol{\phi}(x_i))^2} \\
&= \left(\frac{\beta}{2\pi}\right)^{\frac{m}{2}} e^{-\frac{\beta}{2}\sum_{i=1}^{m}(t_i-\mathbf{a}^T\boldsymbol{\phi}(x_i))^2} \\
\implies \ln \mathbb{P}(\mathbf{t}|\mathbf{X},\mathbf{a},\beta) &= \tfrac{m}{2}(\underbrace{\ln(\beta)}_{\text{Noise Precision}} - \underbrace{\ln(2\pi)}_{\text{Constant}}) - \underbrace{\frac{\beta}{2}\sum_{i=1}^{m}(t_i-\mathbf{a}^T\boldsymbol{\phi}(x_i))}_{\text{Error}} \\
\implies \frac{\partial}{\partial \mathbf{a}} \ln \mathbb{P}(\mathbf{t}|\mathbf{X},\mathbf{a},\beta) &= \beta \sum_{i=1}^{2}(\mathbf{t}_i-\mathbf{a}^T\boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}(x_i)^T \\
\text{Setting} \quad 0 &= \frac{\partial}{\partial \mathbf{a}} \ln \mathbb{P}(\mathbf{t}|\mathbf{X},\mathbf{a},\beta) \\
\implies 0 &= \beta \sum_{i=1}^{2}(\mathbf{t}_i-\mathbf{a}^T\boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}(x_i)^T \\
&= \left(\sum_{i=1}^{m}\mathbf{t}_i\boldsymbol{\phi}(\mathbf{x}_i)^T\right) - \mathbf{a}^T\left(\sum_{i=1}^{m}\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_i)^T\right) \\
\implies \mathbf{a}_{MLE} &= \left(\boldsymbol{\phi}(\mathbf{X})^T\boldsymbol{\phi}(\mathbf{X})\right)^{-1}\boldsymbol{\phi}(\mathbf{X})^T\mathbf{t}
\end{aligned}$$

Theorem 3.1 - *Variance of Posterior*
Let α be the parameter of the prior, β be the parameter for the likelihood and $\mathbf{X}$ be the observed values from the predictor variables.

$$\begin{aligned}
s_n &= (I\alpha+\beta\mathbf{X}^T\mathbf{X})^{-1} \\
&= \left(I\alpha+\beta\begin{pmatrix}\sum_{i=1}^{n}1 & \sum_{i=1}^{n}x_i \\ \sum_{i=1}^{n}x_i & \sum_{i=1}^{n}x_i^2\end{pmatrix}\right)^{-1} \\
&= \begin{pmatrix}\beta n+\alpha & \beta\sum_{i=1}^{n}x_i \\ \beta\sum_{i=1}^{n}x_i & \alpha+\beta\sum_{i=1}^{n}x_i^2\end{pmatrix}^{-1} \\
&= \frac{1}{(\beta n+\alpha)\left(\alpha+\beta\sum_{i=1}^{n}x_i^2\right)-\left(\beta\sum_{i=1}^{n}x_i\right)^2}\begin{pmatrix}\alpha+\beta\sum_{i=1}^{n}x_i^2 & -\beta\sum_{i=1}^{n}x_i \\ -\beta\sum_{i=1}^{n}x_i & \beta n+\alpha\end{pmatrix} \\
&\quad \text{Assume data is centred so } \sum_{i=1}^{n}x_i = 0 \\
&= \frac{1}{(\beta n+\alpha)\left(\alpha+\beta\sum_{i=1}^{n}x_i^2\right)}\begin{pmatrix}\alpha+\beta\sum_{i=1}^{n}x_i^2 & 0 \\ 0 & \beta n+\alpha\end{pmatrix} \\
&= \begin{pmatrix}\frac{1}{\beta n+\alpha} & 0 \\ 0 & \frac{1}{\alpha+\beta\sum_{i=1}^{n}x_i^2}\end{pmatrix}
\end{aligned}$$

Theorem 3.2 - *Mean of Posterior* Let α be the parameter of the prior, β be the parameter for the likelihood, $\mathbf{X}$ be the observed values from the predictor variables and $\mathbf{t}$ be the observed

values for the dependent variable.

$$\begin{aligned}
m_n &= (\alpha I + \beta \mathbf{X}^T\mathbf{X})^{-1}\beta\mathbf{X}^T\mathbf{t} \\
&= s_n\beta\mathbf{X}^T\mathbf{t} \\
&= \beta s_n \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} \\
&= \beta s_n \begin{pmatrix} \sum_{i=1}^{n} t_i \\ \sum_{i=1}^{n} t_i x_i \end{pmatrix} \\
&\quad \text{Assume data is centred so } \sum_{i=1}^{n} x_i = 0 \\
&= \beta \begin{pmatrix} \frac{1}{\beta n + \alpha} & 0 \\ 0 & \frac{1}{\alpha + \beta \sum_{i=1}^{n} x_i^2} \end{pmatrix} \begin{pmatrix} \sum_{i=1}^{n} t_i \\ \sum_{i=1}^{n} t_i x_i \end{pmatrix} \\
&= \begin{pmatrix} \frac{\beta \sum_{i=1}^{n} t_i}{\beta n + \alpha} \\ \frac{\beta \sum_{i=1}^{n} t_i x_i}{\alpha + \beta \sum_{i=1}^{n} x_i^2} \end{pmatrix}
\end{aligned}$$

Proposition 3.2 - *Prediction*
Suppose we are given as inputs to the model: $\mathbf{X}$ observations from the predictor variables, $\mathbf{t}$ observations from the dependent variable; α, parameter for the prior; and β, parameter for the likelihood.
If we now want to predict the value $\hat{t}$ at position $\hat{\mathbf{x}}$ we want to solve

$$\mathbb{P}(\hat{t}|\hat{\mathbf{x}}, \mathbf{t}, \mathbf{X}, \alpha, \beta) = \int \mathbb{P}(\hat{t}|\hat{\mathbf{x}}, \mathbf{a}, \beta)\mathbb{P}(\mathbf{a}|\mathbf{t}|\mathbf{X}, \alpha, \beta)d\mathbf{a}$$

where $\mathbf{a}$ is the coefficient for weighting each parameter.

3.2 Dual Linear Regression

Definition 3.5 - *Kernel*
A *Kernel* is a function that defines an inner-produce in some space.
Let $\mathbf{x}$ be a vector in the orignal space & $\phi(\cdot)$ map from the original space to the kernel space. Then the *Kernel Function* is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) := \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

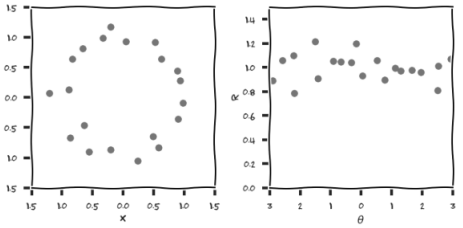
Remark 3.5 - *Usefulness of Kernels*
It is generally easier to define the inner-product of a space than to define a space & kernels allow us to never have to realise a space. This allows us to work with infinite dimensional spaces.
N.B. The space defined by the *Kernel* is called the *Induced Space*.

Definition 3.6 - *Kernel Regression*
Kernel Regression is the act of performing a linear regression in an *Induced Space*.
Let $\mathbf{X}$ & $\mathbf{t}$ be training data, λ be a parameter for noise, $\hat{\mathbf{x}}$ be an unseen data point which we wish to predicit a value $\hat{y}$ for. Then

$$\hat{y}(\hat{\mathbf{x}}) = k(\hat{\mathbf{x}}, \mathbf{x})(k(\mathbf{X}, \mathbf{X}) + \lambda I)^{-1}\mathbf{t}$$

Remark 3.6 - *Usefulness of Kernel Regression*
Note that the problem is linear in the *Induced Space* but not in the original space, thus allowing us to learn non-linear functions using lines.

Remark 3.7 - *Changing Basis*
Some data is clearly non-linear & it may be helpful to transform it into another basis where linear regression is possible.
e.g. If the data appearas to fit a circle in a cartesian basis, it can be translated into polar co-ordinates which should be linear.



Proposition 3.3 - *Changing Basis*
Let $\mathbf{a}$ be weightings for observed parameters and $\mathbf{x}$ be a set of observed parameters.
Suppose we want to change the basis of $\mathbf{x}$, if we have a function $\phi : \mathcal{X} \to \mathcal{Z}$ which can do this then we predict y as

$$t = \mathbf{a}^T \phi(\mathbf{x}) = \mathbf{a}^T \mathbf{z}$$

Definition 3.7 - *Dual Linear Regression*
Standard *Linear Regression* is defined as a linear combination of columns. *Dual Linear Regression* is a linear combination of the inner product of a new data point with each of the training data, this allows it to consider combinations of data points.

Remark 3.8 - *Intution of Dual Regression*
Dual Regression can be considered as describing an unseen data points as a combination of seen ones. *i.e.* Has the shape of a rhino, fur of a tiger, ...

Proposition 3.4 - *Dual Linear Regression Steps*
To perform a *Dual Linear Regression* perform the following

i) Formulate Posterior, $\mathbb{P}(\theta|\mathbf{X})$;

ii) Find stationary point of posterior;

iii) Re-write the coefficients $\mathbf{a}$ in terms of the data;

iv) Perform Kernel regression.

Remark 3.9 - *Useful Kernels*
Not all functions can be used as *Kernels*. Some that can, and can be useful,

i) Kernelised Euclidean Distance $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j)$

ii) Exponented Quadratic $k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 e^{-\frac{1}{2\ell^2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}$.

Remark 3.10 - *Limitations of Linear/Dual Regression*

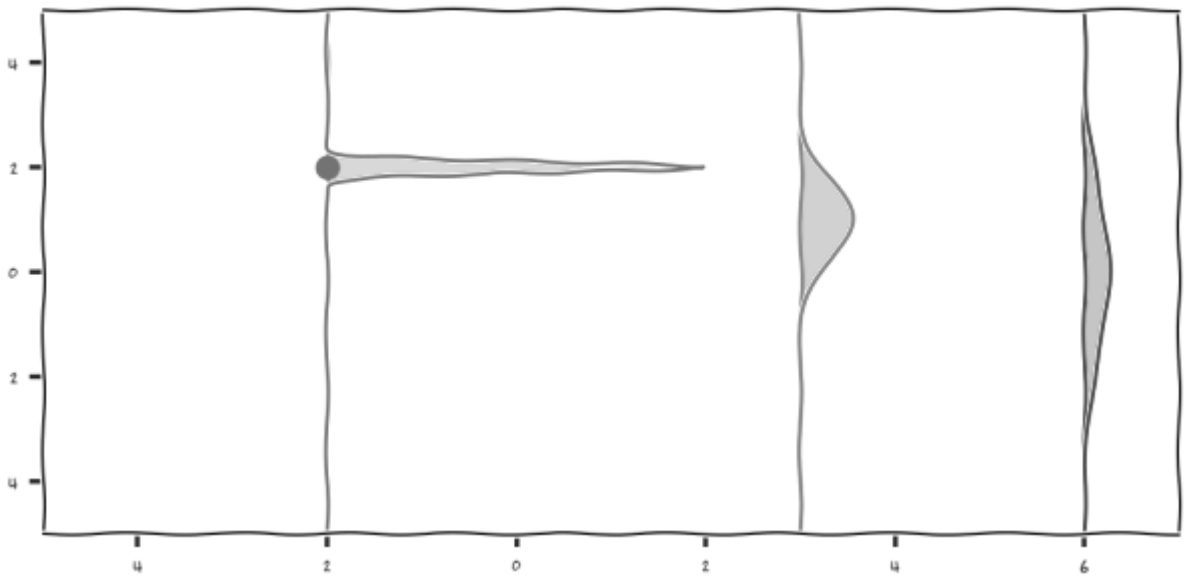
i) No uncertainty in our observed outputs;

ii) No uncertainity in our mapping;

iii) We have to make assumptions over the space of functions

3.3 Gaussian Processes

Remark 3.11 - *Motivation*
Here we want to introduce uncertainty into our observed outputs & mappings. This means that instead of outputting a discrete value we return a probability distribution. Now we can consider a few more features for our observations, such as how much does observing a value at $\mathbf{x}_0$ tell us about an observation at $\mathbf{x}_1$.
N.B. In the image before we have an observation for $x = -2$ and three marginals for $x = -2, 3, 6$ which our observation has a decreasing affec on, as distance increases.



Definition 3.8 - *Gaussian Process*
A *Gaussina Process* is a generalisation of random variables into an infinite number of *Gaussian Distributions*. The specific process is defined by a mean function $\mu(\cdot)$ and a co-variance function function $k(\cdot)$.

$$\mathbb{P}(f_1, f_2, \dots | \mathbf{x}, \boldsymbol{\theta}) \sim \mathcal{GP}\left(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x})\right) = \text{Normal}\left(\begin{pmatrix}\mu(x_1)\\ \mu(x_2)\\ \vdots\end{pmatrix}, \begin{pmatrix}k(x_1,x_1) & k(x_1,x_2) & \dots\\ k(x_2,x_1) & k(x_2,x_2) & \dots\\ \vdots & \vdots & \ddots\end{pmatrix}\right)$$

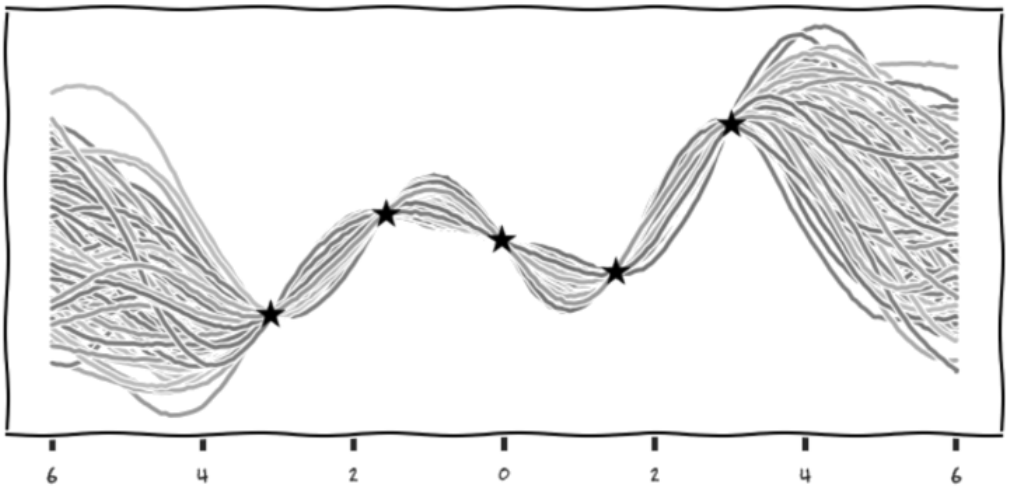
N.B. Gaussian Processes is non-parameteric.

Remark 3.12 - *Covariance Function*
The *Covaraince Function* of a *Gaussian Process* defines how much an observation at $\mathbf{x}_0$ affects our prediction for $\mathbf{x}_1$. The greater the covariance values (Not on the main diagonal) the more an observation tells us. We can define the *Covariance Functions* to vary with distance & other factors.

$$\text{Very little effect} = \begin{pmatrix}1 & .1\\ .1 & 1\end{pmatrix}. \quad \text{A lot of effect} = \begin{pmatrix}1 & .9\\ .9 & 1\end{pmatrix}$$

Remark 3.13 - *Sampling from a Gaussian Process*
When we take a sample from a *Gaussian Process* we are given a function which fits the distributions defined the *Gaussian Process*.

Example 3.2 - *Sampling from a Gaussian Process*



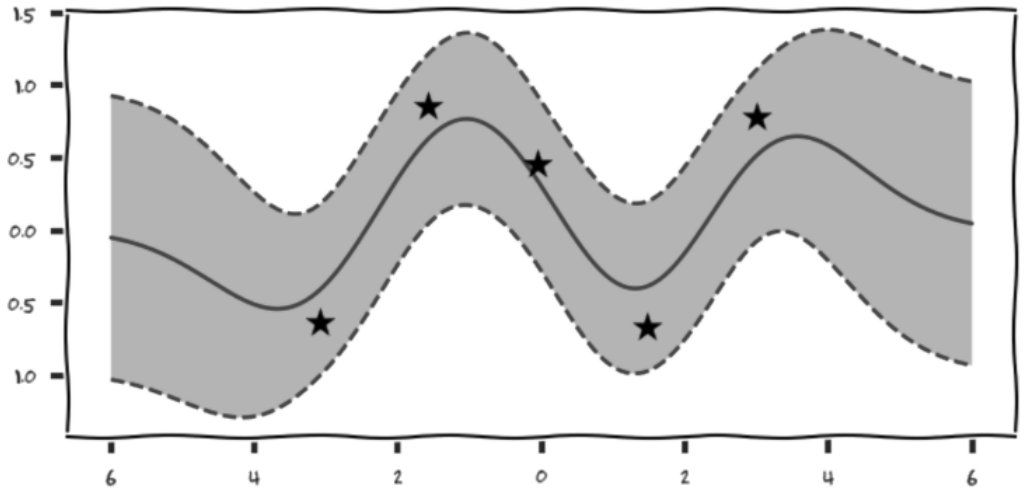
Proposition 3.5 - *Gaussian Process - Posterior, No Noise*
Let $\mathbf{f}, \mathbf{X}$ be training data, $f^*, \mathbf{x}^*$ be training data and k be the co-variance function. We have

$$\begin{aligned}\begin{pmatrix}\mathbf{f}\\ f^*\end{pmatrix} &\sim \text{Normal}\left(\begin{pmatrix}\mathbf{0}\\ 0\end{pmatrix}, \begin{pmatrix}k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{x}^*)\\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*)\end{pmatrix}\right)\\ \mathbb{P}(f^*|\mathbf{x}^*,\mathbf{X},\mathbf{f}) &\sim \text{Normal}\Big(k(\mathbf{x}^*,\mathbf{X})^T k(\mathbf{x}^*,\mathbf{X})^{-1}\mathbf{f},\ k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,\mathbf{X})^T k(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\mathbf{x}^*)\Big)\end{aligned}$$

Proposition 3.6 - *Gaussian Process - Posterior, Noise*
Let $\mathbf{f}, \mathbf{X}$ be training data, $f^*, \mathbf{x}^*$ be training data and k be the co-variance function.
Define $\mathbf{y}_i = f_i + \varepsilon$ where $\varepsilon \sim \text{Normal}(0, \sigma^2 I)$ We have

$$\begin{aligned}\begin{pmatrix}\mathbf{y}\\ f^*\end{pmatrix} &\sim \text{Normal}\left(\begin{pmatrix}\mathbf{0}\\ 0\end{pmatrix}, \begin{pmatrix}k(\mathbf{X},\mathbf{X}) + \sigma^2 I & k(\mathbf{X},\mathbf{x}^*)\\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*)\end{pmatrix}\right)\\ \mathbb{P}(f^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\sigma^2) &\sim \text{Normal}\Big(k(\mathbf{x}^*,\mathbf{x})^T (k(\mathbf{x},\mathbf{x}) + \sigma^2 I)^{-1}\mathbf{y},\ k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,\mathbf{x})^T (k(\mathbf{x},\mathbf{x}) + \sigma^2 I)^{-1}k(\mathbf{x},\mathbf{x}^*)\Big)\end{aligned}$$

Example 3.3 - *Noisy Gaussian Process*



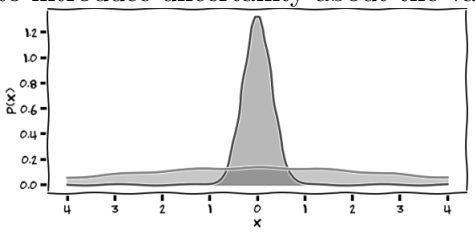
4 Unsupervised Learning

Definition 4.1 - *Unsupervised Learning*
In *Unsupervised Learning* we are given only the ouput data & are tasked with deriving the

underlying properties of each event.
i.e. Given $y = f(x)$ recover both $f(\cdot)$ & x.

5 Bayesian Optimisation

Remark 5.1 - *Importance of Uncertainty*
Note that we can have $\hat{x} = argmax_x p(x) = \text{argmax}_x q(x)$ but $p(\hat{x}) \neq q(\hat{x})$. Due to this we need to introduce uncertainty about the value of x so that we can weight out outcomes accordingly.



Definition 5.1 - *Optimisation*
Optimisation is the process of finding the best outcome for a problem.

Remark 5.2 - *Optimisation*
Classically *Optimisation* is seen as $\hat{x} = \text{argmin}_x f(x)$ however we typically have an objective function that we do not know explicitly, but are able to test. Since testing in real life situations (*e.g.* Medical Trials) are expensive we want to minimise the number required to achieve a good level of certainty.

Definition 5.2 - *Global Optimisation*
Global Optimisation is the set of techniques for finding $x_M = \text{argmin}_{x \in \mathcal{X}} f(x)$ where $\mathcal{X}$ is a bounded domain (reducing the amount of testing required) and f is not known explicitly. It is possible that evaluations of f are noisy.

Proposition 5.1 - *Bayesian Optimisation*

i) Choose a prior over the space of possible objective functions, f.

ii) Combine the prior & likelihood to get a posterior over the space.

iii) Use the posterior to choose a set of evaluations according to a *strategy*.

iv) Add new data, update posterior & re-evaluate.

v) Repeat until budget is gone

Proposition 5.2 - *Naïve Strategies for Global Optimisation*
Below are some naïve *Global Optimisation* stategies

i) Implicity knowledge, ask a SME;

ii) Grid Search, test the domain at regular intervals; and,

iii) Random Sampling.

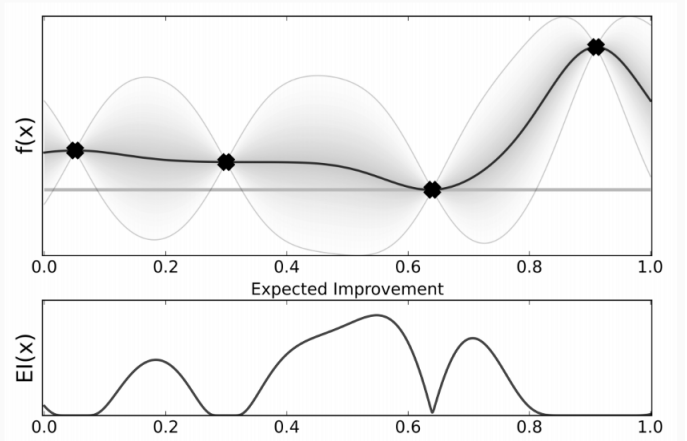
Remark 5.3 - *Interpretting Bayesian Optimisation*
We cannot solve the direct problem $x_M = \text{argmin}_{x \in \mathcal{X}} f(x)$ but can solve $x_{n+1} = \text{argmin}_{x \in \mathcal{X}} \alpha(x; D_n, M_n)$ where α is an *acquisition function*, D_n is the samples taken in the first n steps & M_n is
Remark 5.4 - *Exploration v Exploitation*
When considering strategies for *Bayesian Optimisation* we need to consider how we approach

Exploration (Testing new areas) & *Explotation* (Investigating areas which seem good). A *Acquisition Function* can be defined which returns the expected gain in information if a sample was to be taken at x.

Definition 5.3 - *Expected Improvement*
Expected Improvement is an *Acquistion Function*.
Let $\mathbf{x}$ be a point in the domain, θ be parameters of the distribution and X be data we already know. Then

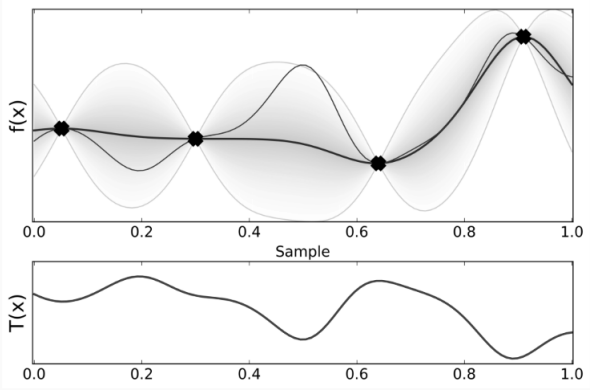
$$EI(\mathbf{x};\theta,X) := \int \max(0, y_{best} - y)\mathbb{P}(y|\mathbf{x},\theta,X)dy$$

We should then sample at $\mathbf{x}$ where $EI(\mathbf{x}) \geq EI(\mathbf{x}') \ \forall \ x' \in \mathcal{X}$ since this point offers the greatest information gain.
N.B. y_{best} is the best value for y we have found so far. *i.e.* The value we are trying to improve.



Definition 5.4 - *Thomson Sampling*
Thomson Sampling is an *Acquistion Function*.
Let $\mathbf{x}$ be a point in the domain, θ be parameters of the distribution and X be data we already know. Then

$$T(\mathbf{x};\theta,X) := \mathbb{P}(y|\mathbf{x},\mathbf{x},\theta,X)$$



6 Evidence

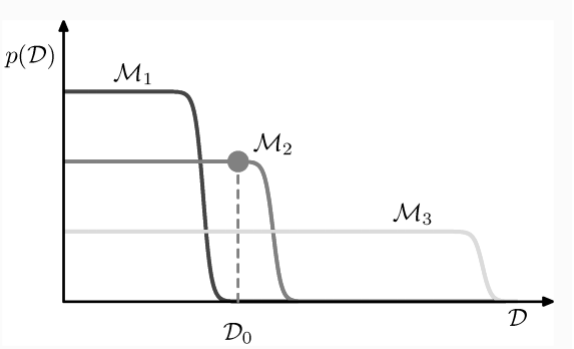
Definition 6.1 - *Evidence*
Evidence is part of *Bayes' Theorem*, if models the likelihood of seeing the data we have been given, regardless of parameters.

$$\mathbb{P}(X) = \int \mathbb{P}(Y|\theta)\mathbb{P}(\theta)d\theta$$

Proposition 6.1 - *Using the Evidence*
If we have multiple models we can test them against out *Evidence* & then choose the model with the greatest accuracy.

Remark 6.1 - *Evidence & Regression Models*
The way we model the *Evidence* depends upon what sort of regression we are doing. Evidence is strongest at points where the lines intersect.

Remark 6.2 - *Model Selection - Rule of Thumb*
When choosing a model you should the simplest model which can explain the data.
N.B. Essentially Occam's Razor.



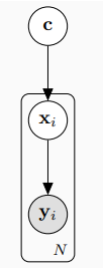
7 Graphical Models

Definition 7.1 - *Graphical Models*
Graphical Models are graphics which show dependency between elements of a model. This dependecy is the minimal factorisation of the joint distribution. *Graphical Models* have several elements

- Node - Random Variable or realisation;
- Edge - A stochastic relationship
- Plate - A product

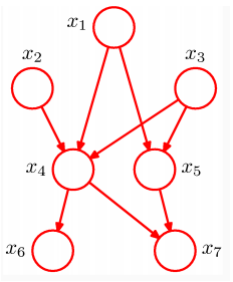
Graphical Models can be directed graphs, often known as *Bayesian Networks*, or undirected, known as *Markov Random Field*.

Example 7.1 - *Graphical Model*
The below graphic model shows that $\mathbf{x}$&$\mathbf{y}$ form a product (ie x_i relates to y_i but not y_j for $i \neq j$) and that $\mathbf{x}$ depends on c & $\mathbf{y}$ depends on $\mathbf{x}$.



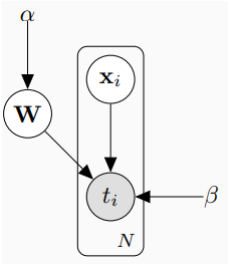
Example 7.2 - *Factorisation of a Directed Graph*
The below *Graphical Model* encodes the factorisation

$$\mathbb{P}(x_1, \dots, x_7) = \mathbb{P}(x_1)\mathbb{P}(x_2)\mathbb{P}(x_3)\mathbb{P}(x_4|x_1, x_2, x_3)\mathbb{P}(x_5|x_1, x_3)\mathbb{P}(x_6|x_4)\mathbb{P}(x_7|x_5, x_4)$$



N.B. We pair each node with its direct parents.

Example 7.3 - *Directed Graph with Constants*
We can include *Constants* in a *Graphical Model* as nodes without a circle.



Proposition 7.1 - *Explaining Away*
Explaining Away is the process of breaking down a feature into multiple features in such a way that you isolate a particular variance of that feature. This is useful as we may not know much about the original feature but a lot about the sub-features.

Example 7.4 - *Explaining Away*
Say you have an image you may wish to break it down into objects, positions & orientations so that the object variable *explains aways* variance associated with objects from the image. Positions & orientations will contain no information about the objects. This can be considered as an integral

$$\mathbb{P}(\text{Image}) = \int \mathbb{P}(\text{Image}|\text{Object,Position,Orientation})\mathbb{P}(\text{Object})\mathbb{P}(\text{Position})\mathbb{P}(\text{Orientation})$$

Proposition 7.2 - *Hierarchical Knowledge*
We can extent the idea of *Explaining Away* by applying it to the sub-features we have produced. If we keep applying this (in a linear fashion) until we find a sub-...-sub-feature that we have knowledge about then we have utilised a *Hierarchical Knowledge*.
This is analogous to defining lots of relationships between features st many features can be explained by a few (in a linear fashion).

8 Non-Parametric Models

Definition 8.1 - *Non-Parameteric Models*
Non-Parameteric Models do not have a specified *a Priori* but are instead determined by the data. This does not mean *Non-Parameteric Models* have no parameters but rather the number they have is not fixed & their values are not pre-defined. *Non-Parameteric Models* cope in an infinite dimenional parameter space.

Definition 8.2 - *Nearest-Neighbour Classifier*
A *Nearest-Neighbour Classifier* takes in a series of training observations & then when given a

test observation simply assigns it to the class of its nearest-neighbour from the training observations.
*N.B. Nearest-Neighbour Classifier*s consider all features equally, at high dimension irrelevant features may be given too much weight.

Definition 8.3 - *k-Nearest-Neighbour Classifier*
A *k-Nearest Neighbour Classifier* is an extension of the *Nearest Neighbour Classifier*. Instead of just considering the nearest neighbour, it considers the k nearest neighbours to the test observation & assigns the test observation to the majority class of these k training observations.

Definition 8.4 - *Gaussian Mixtures Method*
Gaussian Mixtures Method is a *Non-Parameteric Model* which implements *Soft Clustering*. By specifing the number of clusters we want to produce, k, we can apply the *Expectation-Maximisation* algorithm to produce k gaussian distributions which represent k different clusters within the data.

1) Initialise random means for each cluster $\mu_1(1), \dots, \mu_k(1)$.

2) *Estimation Step*
$\forall\ x_i\ \forall\ j \in [1,k]$ set $z_{ij}(t) := \mathbb{E}\big(z_{ij}|x_j, \mu_j(t)\big)$.
Normalise $z_{i1}, \dots, z_{ik}$ st $\sum_{i=1}^{k} z_{ij} = 1$

3) *Maximisation Step*
$\forall\ j \in [1,k]$ set $\mu_j(t+1) := \dfrac{\sum_{i=1}^{n} z_{ij}(t)x_i}{\sum_{i=1}^{n} z_{ij}(t)}$

Remark 8.1 - *Gaussian Processes*
Gaussian Processes are *Non-Parameteric*.

Definition 8.5 - *Generative Model*
A *Generative Model* is a model which given a set of outcomes y it wishes to derive the data which formed it, X.

$$\mathbb{P}(X|Y = y)$$

Example 8.1 - *Constructing a Generative Model*
Consider a scenario where we want to model the topics which occur in a text.
We will define a "topic distribution" which gives the likelihood of each topic, independent of the data.
To consider words we define a "word-topic distribution" which gives the likelihood of a given topic being discussed given a certain word has occurred.
We can specify a generative model for text data by

i) Choose a random distribution over topics

ii) Randomly choose a topic from the topic distribution

iii) Randomly choose a word from word distribution of that topic

Let $w_{d,n}$ be the n^{th} word in the d^{th} document, β_k be the topic-word distribution of the k^{th} topic, θ_d be the topic distribution for the d^{th} document and $z_{d,b}$ be the true topic for n^{th} word in the d^{th} document.

Then we have joint distribution

$$\mathbb{P}(w,z,\theta,\beta)=\underbrace{\prod_{k=1}^{K}\mathbb{P}(\beta_k)\underbrace{\prod_{d=1}^{D}\mathbb{P}(\theta_k)\underbrace{\prod_{n=1}^{N}\mathbb{P}(w_{d,n}|\beta,z_{d,n})\mathbb{P}(z_{d,n}|\theta_d)}_{\text{word}}}_{\text{document}}}_{\text{corpus}}$$

8.1 Dirichlet Process

Definition 8.6 - *Dirichlet Distribution*
The *Dirichlet Distribution* is a generalisation of the β distibution & is the conjugate prior of the *Multinominal Distribution.*

$$\text{Dir}(\boldsymbol{\mu}|\alpha)=\frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdot\cdots\cdot\Gamma(\alpha_K)}\prod_{k=1}^{K}\mu_k^{\alpha_k-1}$$

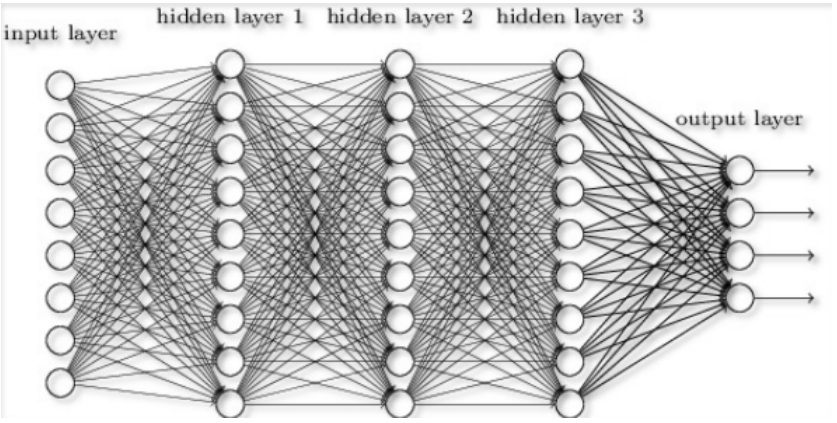
Definition 8.7 - *Dirichlet Process*
A *Dirichlet Process* is an infinite dimensional generalisation of a *Dirichlet Distribution.* It generates a partioning of (possibly) infinite number of elements. It is unintuitive to define a *Dirichlet Process* mathematically & thus we write it constructively.

Definition 8.8 - *Chinese Restaurant Process*
Consider having an infinite number of tables (clusters) & dishes (labels for clusters) andgit customers (data points). Any number of customers can sit at a table but only one dish is allowed per table. We consider how to distribute customers between tables.
Consider when a new customer arrives, the probability they start to a new table is $\frac{\alpha}{N-1-\alpha}$ where N is the number of customers already in the restaurant & α is a sensitivity value we have set.
If they choose to not sit at a new table they sit at table i with probailitiy $\frac{n_i}{N}$ where n_i is the number of people already at table i.
By varying N & α we perform simulations to determine how many tables are used & how many people sit at each one.

9 Neural Networks

Remark 9.1 - *Motivation*
The motivation for the theory of *Neural Networks* is to mimic "brain structure" by having a series of nodes which take in signals and only pass on a signal if a certain threshold is met.
It has been theorised that all intelligence could be modelled by logical rules this appears to be untrue and thus a probabilisitc approach has been taken, *Machine Learning*, where data is used to make inferences.

Definition 9.1 - *Neural Networks*
Neural Networks were first described in 1940 by *Walter Pitts* where the main idea was to composite a function into several functions. *Neural Networks* have a directed graph structure and is made of layers which are split into the input layer, output layer and hidden layer*s*. Each node in one layer connects to all the nodes in the next layer.



Proposition 9.1 - *Function for Hidden Layers*
Below is a general function used to determine whether a neuron in a hidden layer is fired

$$z_j = h(a_j) \text{ where } a_j = \sum_{i=1}^{D} w_{ji} x_i + w_{j0}$$

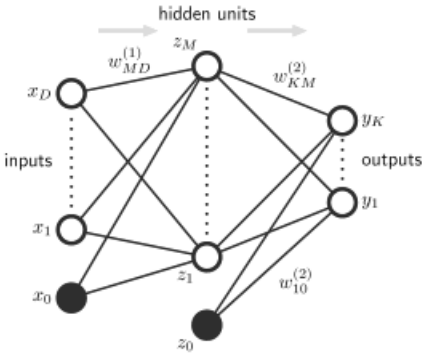
where w_{ji} is the weight applied to the i^{th} neuron by the j^{th}, x_i is the value of the i^{th} neuron, w_{j0} is a bias & $h(\cdot)$ is an activation function which acts to threshold a_j.

Proposition 9.2 - *Function for Ouptut Layer*
Whether a neuron on the *Output Layer* is fired, to signal that this input should have this output, is determined by the below functions

$$y_k = f(a_k) = \sigma(a_k) \text{ where } a_k = \sum_{j=1}^{M} w_{kj} z_j + w_{k0}$$

where w_{kj} is the weight applied to the j^{th} neuron by k^{th}, w_{k0} is a bias and z_j is the value of the j^{th} neuron.

Proposition 9.3 - *Nesting Functions*
Since the functions for the *Output Layer* depends on the vales on the hidden layers, and the hidden layers values depend on the previous layers to themselves it is clear to see how these functions become nested, composite functions.



$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_j^M w_{kj}^{(2)} h\left(\sum_j^D w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

Definition 9.2 - *Kernel of a Function*
The *Kernel* of a function is the set of values which map to themselves when the function is applied to them

$$\text{Ker}(f) = \{x \in X : f(x) = x\} \subseteq X\}$$

Definition 9.3 - *Image of a Function*
The *Image* of a function is the set of all possible values which the function can output

$$\text{Im}(f) = \{y \in Y : \exists\ x \in X \text{ st } f(x) = y\}$$

Theorem 9.1 - *Rank-Nullity Theorem*
The *Rank-Nullity Theorem* states that dimensionalities of the image and the kernel of a function will always sum to the dimensionality of the input space

$$\dim(\text{Im}(f)) + \dim(\text{Ker}(f)) = \dim(X)$$

Proposition 9.4 - *Shrinking Image*
We can use the *Rank-Nullity Theorem* and some intuition to realise that if we keep applying a function to itself its *Image* will never grow and thus, the *Kernel* can only grow.
The intuition here is that elements in the *Kernel* can never leave it, by definition, by values may be mapped to a *Kernel* value and thus leave the *Image* and grow the *Kernel*, where they are now stuck.

Remark 9.2 - *Usefulness of Shrinking Image*
By shrinking the *Image* we are reduction what we are required to know, thus simplifying problems.

Theorem 9.2 - *Change of Variable*
Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ be a random vector with a probability density function given by $P_{\mathbf{X}}(\mathbf{x})$ and let $\mathbf{t} \in \mathcal{Y} \subseteq \mathbb{R}^n$ be a random vector st $\phi(\mathbf{y}) = \mathbf{x}$ where $\pi : \mathcal{Y} \to \mathcal{X}$ is bijective and $|\nabla\phi(\mathbf{y})| > 0\ \forall\ \mathbf{y} \in \mathcal{Y}$. Then the probabiltiy density funcyion $p_Y(\cdot)$ induced in $\mathcal{Y}$ is given by

$$p_Y(\mathbf{y}) = p_X(\phi(\mathbf{y}))|\nabla\phi(\mathbf{y})|$$

where $\nabla\phi(\cdot)$ denotes the Jacobian of $\phi(\cdot)$ and $|\cdot|$ denotes the determinant operator.

Proposition 9.5 - *Neural Network Learning*
Neural Networks "Learn" by, first, *Forward Propogation* where the hidden & output layers apply their functions; then an error is calcualted for the set of weights, W, which were used during *Forward Propogation*

$$E(W) = \sum_{i=1}^{n} \frac{1}{2}(g(\mathbf{x}_i, W) - y_i)^2 + \text{reg}(W)$$

where $g(\cdot,\cdot)$ is the function which represents all the composite functions (here it outputs what the network belives the answer is), y_i is the true result for y_i and reg(W) is . The error is then back propogated to update the weights.

Definition 9.4 - *Backpropogation*
We update weights of neurons using by considering the gradient of the error (*i.e.* how error changes between neurons).

$$W_t = W_{t-1} + \nu\frac{\partial}{\partial W}E(W)$$

where ν is the *Learning Rate*.
We know the error on the output layer. The formula below allows us to propgate this eror back through the layers

$$\delta_{\equiv}\frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial a_j} = \frac{\partial h(a_j)}{\partial a_j}\sum_j w_{kj}\delta_k$$

Proposition 9.6 - *Process of Backpropagation*

i) Randomly initialise weights.

ii) Forward propogate.

iii) Compute Error.

iv) Compute gradients for one step back.

v) Iteratively push gradients back.

vi) Update each layer.

vii) Repeat **ii)-vi)** until convergence

Proposition 9.7 - *Possible Acitvation Functions*

- Sigmoid, $f(x) = \dfrac{1}{1+e^{-x}}$
- Tanh, $f(x) = \tanh(x) = \dfrac{2}{1+e^{-2x}} - 1$
- ReLu, $f(x) = max(0, x) \approx \ln(1+e^x)$

Proposition 9.8 - *Neural Network Architectures*
There are many possible architectures for *Neural Networks* and if we know *a priori* then we can use that to reduce the search space.

- Convolution Neural Networks - Takes a weighted average within a spatial region (*i.e.* blurring). Logic is that pixels depend on their neighbours.
- Recurrent Neural Networks - Designed ot model sequential data

Remark 9.3 - *Neural Networks $\neq$ Models*
Neural Networks are *Decision Machines*, not *Models*. This means they do not understand the data & have no concept of uncertainty, thus new data cannot be generated from them. Further, random noise will always be assigned a label.

Proposition 9.9 - *Combatting Limitations of Neural Networks being Decision Machines*

- Early stopping;
- Layer-wise training;
- Denoising;
- Afverserial training;
- Drop out

10 Reinforcement Learning & Decisions

Definition 10.1 - *Reinforcement Learning*
Reinforcement Learning tries to learn a solution without specifying the task explicitly, instead a system of providing awards for good actions is used.

Proposition 10.1 - *Reinforcement Learning Process*

i) Agent takes an action in the environment.

ii) The environment changes as a result.

iii) The agent may be rewarded.

iv) Repeat **i)-iii)** until a time limit is reached

N.B. The strategy determined for the agent is called the *policy*.

Remark 10.1 - *Optimal Behaviour wrt Time*
Different strategies are required to find an *optimal behaviour* depending upon the time scale

- Finite time horizon, $\mathbb{E}\left(\sum_{t=0}^{h} r_t\right)$
- Average Reward (over all times), $\lim_{h\to\infty} \mathbb{E}\left(\frac{1}{h}\sum_{t=0}^{h} r_t\right)$
- Infinite Horizon (discounted reward) $\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_t\right)$ for $\gamma \in [0, 1]$

Definition 10.2 - *Markov Decision Process*
Suppose we have a fully observable system we can determine the optimal policy using dynamic programming. *i.e.* We know the *Transition Matrix*, $T(s, a, s')$, and *Reward Matrix*, $R(s, a, s')$, explicitly.

$$T(s, a, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$$
$$R(s, a, s') = \mathbb{E}(s_t = s, a_t = a, s_{t+1} = s')$$

where s is the current state, s' is another state & a is the action taken.

Remark 10.2 - *We might not be able to observe a state*

Proposition 10.2 - *Model Free*
Learning a mdel can be hard due to uncertainty propagation. Directly learning a function from state-action sets to reward can be easier

$$Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

N.B. This does not build a model, rather a decision system.

Remark 10.3 - *Tricky part of Reinforcement Learning*
How to get the data.

Remark 10.4 - *Exploration v Explotation*
When thinking of a strategy to come up with a policy we need to balance: exploring the environement & exploiting what we know.

11 Classification

Definition 11.1 - *Classification Task*
Consider a data set $\mathcal{D}$ where each item belongs to a class in $\mathcal{C}$.
The *Classification Task* is given a set of observations $\{\mathcal{D}, \mathcal{C}\}$ can we correctly associate new data to the correct classes?

Remark 11.1 - *Generative Models*
Suppose we have an image of a dog. The fact the appearance of image is a dog does not make the object a dog, the fact that the object is a dog makes the image have the appearance of a dog. Further we can have images that have the appearance of a dog but are not of a dog, such as drawings.

Remark 11.2 - *Classification is an Iverse Problem*
Further to **Remark 11.1** we note that multiple classes of objects can produce the same image. Thus when performing *Classification* on an image we might need to return multiple possible classes.

Proposition 11.1 - *Formulating Classification*
Let $\mathbf{x}$ be a set of observed values & $c_1 \in \mathcal{C}$ is one of the possible classes we can assign to $\mathbf{x}$. By *Bayes' Rule* we have

$$\mathbb{P}(c_1|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\mathbb{P}(\mathbf{x})}$$

When the cardinality of $\mathcal{C}$ is low we can calculate the *Evidence* explicitly

$$\mathbb{P}(\mathbf{x}) = \sum_{c\in\mathcal{C}} \mathbb{P}(\mathbf{x}|c)\mathbb{P}(c)$$

Meaning the *Posterior* can be formulatede as

$$\mathbb{P}(c_1|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\sum_{c\in\mathcal{C}} \mathbb{P}(\mathbf{x}|c)\mathbb{P}(c)}$$

Proposition 11.2 - *Binary Classification*
Suppose we are classifying to one of two classes, *i.e.* $\mathcal{C} := \{c_1, c_2\}$. Then we have

$$\begin{aligned}
\mathbb{P}(\mathbf{x}) &= \mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1) + \mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2) \\
\implies \mathbb{P}(c_1|\mathbf{x}) &= \frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1) + \mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)} \\
&= \frac{\left(\frac{1}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}\right)}{\left(\frac{1}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}\right)} \frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1) + \mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)} \\
&= \frac{1}{1 + \frac{\mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}} \\
&= \frac{1}{1 + \exp\left(\ln\left(\frac{\mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)}{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}\right)\right)} \\
&= \frac{1}{1 + \exp\left(-\ln\left(\frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)}\right)\right)} \\
&= \frac{1}{1 + \exp\left(-\ln\left(\frac{\mathbb{P}(\mathbf{x},c_1)}{\mathbb{P}(\mathbf{x},c_2)}\right)\right)}
\end{aligned}$$

We note the *Sigmoid Function* has the form

$$y(t) = \frac{1}{1 + e^{-t}}$$

By setting $t = \ln\left(\frac{\mathbb{P}(\mathbf{x},c_1)}{\mathbb{P}(\mathbf{x},c_2)}\right)$ we can use the *Sigmoid Function* for *Binary Classification*.

Proposition 11.3 - *Binary Classification - Gaussian*
In **Proposition 11.2** we did not specify the model for the data.
Suppose $\mathbb{P}(\mathbf{x}|c_i) \sim_{Normal} (x|\mu_i, \sigma_i^2)$ then

$$\begin{aligned} t &:= \ln\left(\frac{\mathbb{P}(\mathbf{x}|c_1)\mathbb{P}(c_1)}{\mathbb{P}(\mathbf{x}|c_2)\mathbb{P}(c_2)}\right) \\ &= \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma_1^2}}e^{-\frac{(\mathbf{x}-\mu_1)^2}{2\sigma_1^2}}\mathbb{P}(c_1)}{\frac{1}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{(\mathbf{x}-\mu_2)^2}{2\sigma_2^2}}\mathbb{P}(c_2)}\right) \\ &= -\tfrac{1}{2}\ln\left(\frac{2\pi\sigma_1^2}{2\pi\sigma_2^2}\right) + \ln\left(\frac{\mathbb{P}(c_1)}{\mathbb{P}(c_2)}\right) - \frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2} \\ &= \ln\left(\frac{\sigma_2}{\sigma_1}\right) + \ln\left(\frac{\mathbb{P}(c_1)}{\mathbb{P}(c_2)}\right) - x^2\left(\frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_2^2}\right) + x\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right) - \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right) \end{aligned}$$

We can analyse this expression of t to see how the relationship between the distributions of c_1 & c_2 affect classification.

i) If $\sigma_1 = \sigma_2$ the posterior is linear in $\mathbf{x}$.

ii) If $\mu_1 = \mu_2$ and $\sigma_1 = \sigma_2$ then the posterior is independent of $\mathbf{x}$ since $\mathbb{P}(c_1|\mathbf{x}) = \mathbb{P}(c_1)$

11.1 Logisitic Regression

Definition 11.2 - *Logistic Regression*
Logisitic Regression is a process of deriving an $f(\cdot)$ st

$$\mathbb{P}(c_1|\mathbf{x}) = \frac{1}{1+e^{-f(\mathbf{x})}} = \sigma(\mathbf{x})$$

is a good binary classifier.
We aim to define $f(\cdot)$ such that objects in c_1 produced large *positive* values & objects in c_2 produce large *negative* values.

Remark 11.3 - *Linear Classification*
If we use a *Linear Classifier* with *Logistic Regression* we define $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ and our problem is to find $\mathbf{w}$ which maximises positive values when $\mathbf{x}$ is of class c_1, and negative values when $\mathbf{x}$ is of class c_2.

$$\hat{\mathbf{w}}_{\text{MLE}} = \text{argmax}_{\mathbf{w}}\mathbb{P}(\mathcal{C}|\mathcal{D}) = \text{argmin}_{\mathbf{w}} - \ln\mathbb{P}(\mathcal{C}|\mathcal{D})$$

Remark 11.4 - *Usefullness of Logistic Regression*
If our feature data has high dimensionality then the mean & co-variance matrices for each class will be high. This means that *Classifiers* will have lots of parameters which need to be found. *Logisitic Regression* requires only 1 more than the dimensionality of the feature data (a bias).

Example 11.1 - *Remark 11.4*
Suppose $\mathbf{x} \in \mathbb{R}^{100}$ then each Gaussian (μ_i, Σ_i) would have dimensionality $\mu_i \in \mathbb{R}^{100}$ & $\Sigma_i \in \mathbb{R}^{100\times100}$. For a binary classifier this would require $2(100 + 100 \times 100) = 20200$ parameters. Whereas a *Logistic Regression* model requires 101.

Remark 11.5 - *Limitations of Logistic Regression*
If a *Logistic Regression* model is given an outlier point it will define it with almost absolute certainty to the class it is closest to. Other *Models* which use a *Gaussian* distribution will classify it to the same class as the *Logistic Regression* model but with lower certainty as the two tails will have a similar value at the outlier.

11.2 Laplace Approximation

Definition 11.3 - *Laplace Approxmation*
Laplace Approximation is used to approximate the integral of functions which have most their mass concentrated in a small area (and thus have rapidly decreasing tails).
Let $x_0 := \text{argmax}_x g(x)$ for some continuous function g.
Define $h(x) := \ln g(x)$. Then

$$\begin{aligned}\int_a^b g(x)dx &= \int_a^b \exp(h(x)dx \\ &\approx \int_a^b \exp\left(h(x_0) + h'(x_0)(x-x_0) + \frac{1}{2}h''(x_0)(x-x_0)^2\right)dx \text{ by Taylor Expansion}\end{aligned}$$

Since we define $h(x_0)$ to be a maximum, $h'(x_0) = 0$. Thus

$$\int_a^b \exp(h(x))dx \approx \int_a^b \exp\left(h(x_0) + \frac{1}{2}h''(x_0)(x-x_0)^2\right)$$

TODO - Read book on this

12 Approximative Inference

Remark 12.1 - *Motivation*
Suppose we are trying to implement a model which depends on a distribution $p(x)$, but that $p(x)$ is intractable. We need to consider ways to inferring the distribution of $p(x)$. The technique we use for this depends on the scenarion. There are three main groups of techniques

i) Sampling.

ii) Laplace Approximation

iii) Variational Inference

Definition 12.1 - *Markov Random Field*
A *Markov Random Field* is a graph of random variables where edges show dependency between random variables.
Images can be considered as *Markov Random Fields* as adjacent pixels depend upon each other.
N.B. Also known as a *Markov Network*.

Remark 12.2 - *Markov Random Field Marginal Likelihood*
Let $\{x_1, x_2, \dots\}$ be the set of all binary images & y be a class that an image can be assigned to. Then

$$\mathbb{P}(y) = \int \mathbb{P}(y|x)\mathbb{P}(x)dx = \sum_i \mathbb{P}(y|x_i)\mathbb{P}(x_i)$$

i.e. Integrating over all binary images. The number of possible binary images is finite so we can sum over it, but it is a big number so obviously a fouls errend.

Remark 12.3 - *Techniques for sampling from distributions we do not know the form of*

i) Rejection Sampling.

ii) Importance Sampling.

iii) Markov Chain Monte Carlo.

Remark 12.4 -
Let $p(\cdot)$ be a distribution we do not know how to sample from & $\tilde{p}(\cdot)$ be one we do

$$p(\cdot) = \frac{1}{Z}\tilde{p}(z)$$

Definition 12.2 - *Rejection Sampling*
Rejection Sampling is a tecnique used to sample from distributions which we do not know explicitly.
Suppose $Z \sim f_Z(\cdot)$ is a distribution we do not know the form of.

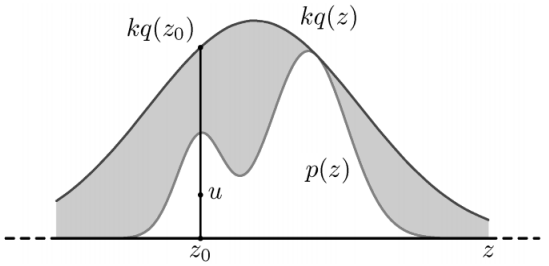
i) Pick an approximate distribution, $h_Z(\cdot)$.

ii) Choose a $k \in \mathbb{R}^{\geq 0}$ such that $kh_Z(z) \geq f_Z(z)\ \forall\ z$.

iii) Pick a random location using the approximate distribution $z_0 \sim h_Z(\cdot)$.

iv) Pick a random number from the uniform distribution $u_0 \sim \text{Uniform}[0, kh_Z(z_0)]$.

v) If $u_0 > f_Z(z_0)$ then reject z_0 otherwise sample it.



Remark 12.5 - *Reject Sampling*

- We can use the distribution of the samples as a *proposed distribution* for f_Z.
- If the bound between kh_Z and f_Z is tight then the sampler is efficient. Often it is nt though as we have to set k so high.
- Does not scale well to multiple dimension.
- Lots of samples get rejected.

Definition 12.3 - *Importance Sampling*
Suppose $Z \sim f_Z(\cdot)$ is a distribution we do not know the form of and h_Z be a distribution we do know explicitly.
Importance Sampling is a technique used estimate teh expected value of Z.

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}}(f_Z) &= \int f_Z(z)\mathbb{P}(z)dz \\
&= \int f_Z(z)\mathbb{P}(z)\tfrac{h_Z(z)}{h_Z(z)}dz \\
&= \int f_Z(z)\tfrac{\mathbb{P}(z)}{h_Z(z)}h_Z(z)dz \\
&= \mathbb{E}_{h_Z}\left(f_Z(z)\tfrac{\mathbb{P}(z)}{h_Z(z)}\right) \\
&\approx \frac{1}{L}\sum_{i=1}^{L} f_Z(z^{(i)})\frac{\mathbb{P}(z^{(i)})}{h_Z(z^{(i)})}
\end{aligned}$$

where $z^{(i)} \sim h_Z(\cdot)$.
TODO - pretty sure I am confused about f_Z and h_Z.

Remark 12.6 - *Importance Sampling*

- Accepts all samples.
- $\frac{\mathbb{P}(z^{(i)})}{f_Z(z^{(i)})}$ corrects bias in sampling from the wrong distribution.
- It is not always possible to evaluate $\mathbb{P}(z)$.

Definition 12.4 - *Markov Chain Monte Carlo*
Markov Chain Monte Carlo is a set of methods used for sampling from a probability distribution which is not known explicitly.
Let z_t be the state at time t and $h_Z(\cdot)$ be a proposed conditional distribution which we can sample from.

i) Initialise a state z_0.

ii) Sample from the proposed conditional distribution $h_Z(z^*|z_0)$.

iii) Compute an acceptance probability

$$A(z^*, z_0) = \min\left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z_0)}\right)$$

iv) Sample $u \sim \text{Uniform}(0, 1)$.

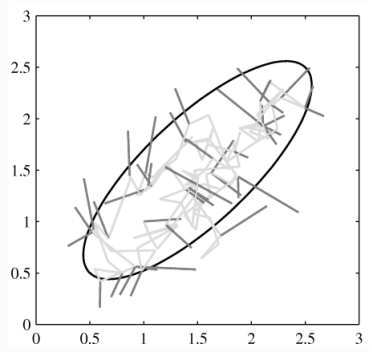
(a) If $A(z^*, z_0)$ ¿0 then $z_1 = z^*$.

(b) Else $z_1 = z_0$.

v) Repeat 2-4 until time limit is reached.

N.B. Also known as *Metropolis Sampling.*

Example 12.1 - *Markov Chain Monte Carlo*



Remark 12.7 - *Markov Chain Monte Carlo*

- Has the *Markov Property* - Remembers its state from the last sample & uses it.
- Good for exploration due to *Markov Property.*

Definition 12.5 - *Gibbs Sampling*
Gibbs Sampling exploits the fact that 1-dimension samples are often easy to get, in order to create a very simple markov chain.

i) Intialise a state $\mathbf{x}_0$.

ii) Pick a single variable from that state, $x_i \in \mathbf{x}$.

iii) Formulate a posterior, $\mathbb{P}(x_i|\mathbf{x}_{\neg i})$. $\mathbf{x}_{\neq i}$ is the state x without variable x_i.

iv) Sample from the posterior $x_{(1,i)} \sim \mathbb{P}(x_i|\mathbf{x}_{\neq i})$.

v) Repeat 2-4 for all variables.

N.B. The general idea is to sample each variable in turn.
Remark 12.8 - *Summary - Stochastic Inference*

i) Hard to know how well we are doing.

ii) Usually slow.

13 Deterministic Approximative Inference

Remark 13.1 - *Motivation*
Can we reformulate the problem of infering an intractable conditional distribution $\mathbb{P}(y|\mathbf{x})$ as an *optimisation* problem?
We want to find an approximate model, $q_\theta(\mathbf{x})$, which is approximately equal to this conditional distribution where θ are parameters we tune. Not that $q_\theta(\mathbf{x})$ is independent of y.

Remark 13.2 - *What is actually intractable?*
Suppose that we want to model the distribution $p(y)$ but it is intractable.
We have from *Bayes' Theorem* that

$$p(y) = \frac{p(y|x)p(x)}{p(x|y)}$$

Both $p(y|x)$ and $p(x)$ are known thus $p(x|y)$ is intractable.
This means the problem of approximating $p(y)$ is equivalent to approximating $p(x|y)$ and this is the problem we try to solve using *Variational Bayes*.
N.B. Since $p(y)$ is intractable it is impossible to derive a mathematical definition for it, thus our answer will always be wrong.

Remark 13.3 - *Scenario*
Suppose we are given a set of data $\{y_i\}$ where $y_i \in \{0, 1\}$ and each value represents the value of a pixel in a monochromatic image. Images however are noisy and thus y_i is just a realisation of an underlying value for pixel i where the true value is x_i. Consider the task where we want to infer the true value of x_i from $\{y_i\}$.
N.B. The distribution in this case is intracable thus we must use inference.

Theorem 13.1 - *Jensen's Inequality* The function of the expected value is a lower bound to the expected value of the function.

$$\begin{aligned} \mathbb{E}[f(X)] &\geq f(\mathbb{E}(X)) \\ \int f(x)p(x)dx &\geq f\left(\int xp(x)dx\right) \end{aligned}$$

N.B. Often we use this for $f(x) = \ln x$ as logs make many problems easier.

Definition 13.1 - *Kullback-Leibler Divergence*
Kullback-Leibler Divergence measures the divergence between two distributions.
It is not symmetric & thus is not a metric.
Let $p(X)$ and $q(X)$ be probability distributions we wish to compare. Then

$$KL(q(X)||p(X)) := \int q(X) \ln \frac{q(X)}{p(X)} dX$$

N.B. The KL measure requires that $q(X)$ & $p(X)$ be zero-matching. *i.e.* Whenever one of them equals 0 the other must too. This causes some limitations.

Proposition 13.1 - *Features of Kullback-Leibler Divergence*
Here are some features of values of Kullback-Leibler Divergence which are pertinant to this partical problem

i) $KL(q(x)||p(x)) = 0 \iff q(x) = p(x)$.

ii) $KL(q(x)||p(x)) \geq 0$.

Proof 13.1 - $KL(q(x)||p(x|y)) = 0 \iff q(x) = p(x|y)$ Let $p(x)$ & $q(x)$ be two probability distributions.

$$\begin{aligned} KL(q(x)||p(x|y)) &= \int q(x) \ln \frac{q(x)}{p(x|y)} dx \\ &\quad \text{Assume } q(x) = p(x|y) \\ &= \int p(x|y) \ln \frac{p(x|y)}{p(x|y)} dx \\ &= \int p(x|y) \ln 1 dx \\ &= \int p(x|y) \times 0 dx \\ &= 0 \end{aligned}$$

Proof 13.2 - $KL(q(x)||p(x|y)) \geq 0$ Let $p(x)$ & $q(x)$ be two probability distributions.

$$\begin{aligned} KL(q(x)||p(x|y)) &= \int q(x) \ln \frac{q(x)}{p(x|y)} dx \\ &= -\int q(x) \ln \frac{p(x|y)}{q(x)} dx \\ &= -\int q(x)(\ln p(x|y) - \ln q(x)) dx \\ &= -\int q(x) \ln p(x|y) dx + \int q(x) \ln q(x) dx \\ &\geq -\ln \int p(x|y) dx + \ln \int q(x) dx \text{ by Jensen's Inequality} \\ &= -\ln 1 + \ln 1 \\ &= 0 - 0 = 0 \end{aligned}$$

Proposition 13.2 - *Variational Bayes*
Let $p(\mathbf{Y})$ be an intractable distribution we are trying to approximate& $q(\mathbf{X})$ be a distribution we are choosing (and thus is tractable).
The idea is to choose $q(\mathbf{X})$ st it is approximately equal to $p(\mathbf{X})$.
Thus

$$\begin{aligned} \ln p(\mathbf{Y}) &= \ln \left(\int p(\mathbf{Y}, \mathbf{X}) d\mathbf{X} \right) \text{ by definition of evidence} \\ &= \ln \left(\int p(\mathbf{X}|\mathbf{Y}) p(\mathbf{Y}) d\mathbf{X} \right) \\ &= \ln \left(\int \frac{q(\mathbf{X})}{q(\mathbf{X})} p(\mathbf{X}|\mathbf{Y}) p(\mathbf{Y}) d\mathbf{X} \right) \\ &\geq \int q(\mathbf{X}) \ln \left(\frac{p(\mathbf{X}|\mathbf{Y}) p(\mathbf{Y})}{q(\mathbf{X})} \right) d\mathbf{X} \text{ by Jensen's Inequality} \\ &= \int q(\mathbf{X}) \ln \frac{p(\mathbf{X}|\mathbf{Y})}{q(\mathbf{X})} d\mathbf{X} + \int q(\mathbf{X}) \ln p(\mathbf{Y}) d\mathbf{X} \text{ by log rules} \\ &= \int q(\mathbf{X}) \ln \frac{p(\mathbf{X}|\mathbf{Y})}{q(\mathbf{X})} d\mathbf{X} + \underbrace{\int q(\mathbf{X}) d\mathbf{X}}_{=1} \ln p(\mathbf{Y}) \\ &= -KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y})) + \ln p(\mathbf{Y}) \text{ by definition of KL} \end{aligned}$$

Consider the *Kullback-Leibler Divergence* element of this derivation as this is the term we are seeking to minimise (*i.e.* make as close to zero as possible).

$$\begin{aligned}
KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y})) &:= \int q(\mathbf{X}) \ln\left(\frac{q(\mathbf{X})}{p(\mathbf{X}|\mathbf{Y})}\right) d\mathbf{X} \\
&= \int q(\mathbf{X}) \ln\left(\frac{q(\mathbf{X})p(\mathbf{Y})}{p(\mathbf{X},\mathbf{Y})}\right) d\mathbf{X} \text{ by definition of joint distribution} \\
&= \int q(\mathbf{X}) \ln q(\mathbf{X}) d\mathbf{X} - \int q(\mathbf{X}) \ln p(\mathbf{X},\mathbf{Y}) d\mathbf{X} + \int q(\mathbf{X}) \ln p(\mathbf{Y}) d\mathbf{X} \text{ by log rules} \\
&= H(q(\mathbf{X})) - \mathbb{E}_{q(\mathbf{x})}[\ln p(\mathbf{X},\mathbf{Y})] + \underbrace{\int q(\mathbf{X}) d\mathbf{X}}_{=1} \ln p(\mathbf{Y}) \\
&= H(q(\mathbf{X})) - \mathbb{E}_{q(\mathbf{x})}[\ln p(\mathbf{X},\mathbf{Y})] + \ln p(\mathbf{Y}) \text{ by definitions} \\
\implies \ln(\mathbf{Y}) &= KL(q(\mathbf{X})||p(\mathbf{X},\mathbf{Y})) + \underbrace{\mathbb{E}_{q(\mathbf{X})}[\ln p(\mathbf{X},\mathbf{Y}) - H(q(\mathbf{X}))}_{\text{Evidence Lower Bound}} \text{ by rearrangement} \\
&\geq \mathbb{E}_{q(\mathbf{X})}[\ln p(\mathbf{X},\mathbf{Y})] - H(q(\mathbf{X})) \text{ since } KL(\cdot||\cdot) \geq 0 \\
&=: L(q(\mathbf{X}))
\end{aligned}$$

We have now derived a likelihood function, $L(q(\mathbf{X}))$, for our proposed approximate-distribution which we seek to maximise.
Note the following are valid expressions of our likelihood function

$$L(q(\mathbf{X})) := \mathbb{E}_{q(\mathbf{x})}[\ln p(\mathbf{X},\mathbf{Y})] - H(q(\mathbf{X})) = \int q(\mathbf{X}) \ln \frac{p(\mathbf{Y},\mathbf{X})}{q(\mathbf{X})} d\mathbf{X} \text{ by log rules}$$

N.B. Maximising $L(q(\mathbf{X}))$ is equivalent to minimising $KL(q(\mathbf{X})||p(\mathbf{X},\mathbf{Y}))$ which in turns implies $q(\mathbf{X})$ and $p(\mathbf{X},\mathbf{Y})$ are very similar, as desired.

13.1 Mean Field Approximation

Remark 13.4 - *Motivation*
Now we have managed to redefine the inference problem as an optimisation problem we need to consider possible distributions for $q(\mathbf{X})$ which we can then test. *Mean Field Approximation* assumes that each data point is independent and update the distribution of each point over a series of cycles.

$$q(\mathbf{X}) := \prod_i q_i(X_i)$$

Proposition 13.3 - *Mean Field Approximation*
Consider the likelihood function for *variational inference* derived in **Proposition 13.2** and the definition of $q(\mathbf{X})$ we are using

$$\begin{aligned}
L(q(\mathbf{X})) &= \int q(\mathbf{X}) \ln \frac{p(\mathbf{Y},\mathbf{X})}{q(\mathbf{X})} d\mathbf{X} \\
&= \int \prod_i q_i(X_i) \ln \frac{p(\mathbf{Y},\mathbf{X})}{\prod_k q_k(X_k)} d\mathbf{X} \\
&= \int \prod_i q_i(X_i) \left(\ln p(\mathbf{Y},\mathbf{X}) - \sum_k \ln q_k(X_k)\right) d\mathbf{X}
\end{aligned}$$

Since we want to update the distribution of each data point we note that our likelihood function can be rewritten as

$$L(q) = L(q_j) + L(q_{\neg j})$$

where j is a data point we are interested in at this point in time.
Thus we can derive the likelihood function for proposed distributions of this form

$$\begin{aligned}
L(q) &= \int \prod_i q_i(X_i) \left(\ln p(\mathbf{Y}, \mathbf{X}) - \sum_k \ln q_k(X_k) \right) d\mathbf{X} \\
&= \int_j \int_{\neg j} q_j(X_j) \prod_{i \neq j} q_i(X_i) \left(\ln p(\mathbf{Y}, \mathbf{X}) - \sum_k \ln q_k(X_k) \right) d\mathbf{X}_{\neg j} dX_j \\
&= \int_j q_j(X_j) \underbrace{\int_{\neg j} \prod_{i \neq j} q_i(X_i) \ln p(\mathbf{Y}, \mathbf{X}) d\mathbf{X}_{\neg j}}_{=: \ln f_j(X_j)} dX_j \\
&- \int_j q_j(X_j) \int_{\neg j} \prod_{i \neq j} q_i(X_i) \left(\ln q_j(X_j) + \sum_{k \neq j} \ln q_k(X_k) \right) d\mathbf{X}_{\neq j} dX_j \text{ by log rules \& def. of summation} \\
&= \int_j q_j(X_j) \ln f_j(X_j) dX_j - \int_j q_j(X_j) \left(\ln q_j(X_j) \underbrace{\int_{\neg j} \prod_{i \neq j} q_i(X_i) d\mathbf{X}_{\neg j}}_{=1} + \underbrace{\int_{\neg j} \prod_{i \neq j} q_i(X_i) \sum_{k \neq j} \ln q_k(x_k) d\mathbf{X}_{\neg k}}_{\text{constant wrt } q_j} \right) \\
&= \int_j q_j(X_j) \ln f_j(X_j) dX_j - \int_j q_j(X_j) \ln q_j(X_j) dX_j + c \underbrace{\int_j q_j(X_j) dX_j}_{=1} \\
&= \int_j q_j(X_j) \ln \left(\frac{f_j(X_j)}{q_j(X_j)} \right) dX_j + c \text{ by log rules} \\
&= -\int_j q_j(X_j) \ln \left(\frac{q_j(X_j)}{f_j(X_j)} \right) dX_j + c \\
&= -KL(q_j(X_j) || f_j(X_j)) + c
\end{aligned}$$

Again our task is to minimise $KL(\cdot||\cdot)$. We know that $KL(\cdot||\cdot) \geq 0$ the minimal value for KL is 0 (the case where $q_j = f_j$).
Since we are free to choose $q_j(X_j)$ we can simply equate it with $f_j(X_j)$.
Consider the defintion of $f_j(X_j)$ from the previous derivation

$$\ln f_j(X_j) := \int_{\neg j} \prod_{i \neq j} q_i(X_i) \ln p(\mathbf{Y}, \mathbf{X}) d\mathbf{X}_{\neg j} = \mathbb{E}_{q_{\neg j}(X_{\neq j})}[\ln p(\mathbf{Y}, \mathbf{X})]$$

Thus we need to choose a distribution, $q_j(X_j)$, where the above expectation is tractable.

Proposition 13.4 - *Mean Field Approximation - Ising Model*
Now we can consider possible distributions for $q_j(X_j)$.
In the *Ising Model* we define the following prior

$$p(\mathbf{X}) = \frac{1}{Z_0} e^{\mathbb{E}_0(X)} \text{ where } Z_0 \text{ is a normalising term \& } \mathbb{E}_0(\mathbf{X}) = \sum_{i=1}^{N} \sum_{j \in N_i} w_{ij} x_i x_j$$

note that N_i is the neighbourhood of the i^{th} value, w_{ij} is the weighting that i gives the value of x_j and $x_i, x_j \in \{-1, 1\}$.
Since $x_i, x_j \in \{-1, 1\}$ we find that $e^{\mathbb{E}_0(\mathbf{X})}$ only grows if $x_i = x_j$ since that is the onlly case where $x_i x_j$ is positive.
We definethe likelihood

$$\mathbb{P}(\mathbf{Y}|\mathbf{X}) = \prod_i p(y_i|x_i) = \frac{1}{Z_1} \prod_i e^{L_i(x_i)}$$

where L_i gives a large value if it is likely that x_i generated y_i.
Now we make our assumptions about our proposed approximate-distribution $q(\mathbf{X})$. We will assume that the distribution over each latent variable is independent

$$q(\mathbf{X}) = \prod_i q_i(x_i, \mu_i) \text{ where } \mathbb{E}_{q_i(X_i)}(X_i) = \mu_i$$

We now need to get the joint distribution of this model

$$\begin{aligned}
\ln p(\mathbf{X}, \mathbf{Y}) &= \ln p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}) \\
&= \ln\left(\prod_i e^{L_i(x_i)} \frac{1}{Z_0} e^{\sum_{j\in N_i} w_{ij}x_i x_j}\right) \\
&= \sum_i \left(L_i(x_i) + \sum_{j\in N_i} w_{ij}x_i x_j\right) + c \text{ by application of } \ln
\end{aligned}$$

We can now compute the expectation to get the approximative posterior

$$\begin{aligned}
\ln q_i(x_i) &= \ln f_i(x_i) \\
&= \int \prod_{j\neq i} q_j(X_j) \ln p(\mathbf{X}, \mathbf{Y}) d\mathbf{X}_{\neg i} \\
&= \int \prod_{j\neq i} q_j(x_j) \left(L_i(x_i) + \sum_{k\in N_i} w_{ik}x_i x_k + c\right) d\mathbf{X}_{\neg i} \text{ since we consider only one term} \\
&= \underbrace{\int \prod_{j\neq i} q_j(x_j) d\mathbf{X}_{\neg i}}_{=1} L_i(x_i) + \int \prod_{j\neq i} q_j(x_j) \sum_{k\in N_i} w_{ik}x_i x_k d\mathbf{X}_{\neg i} + c
\end{aligned}$$

Consider the second integral

$$\int \prod_{j\neq i} q_j(x_j) \sum_{k\in N_i} w_{ik}x_i x_k d\mathbf{X}_{\neg i} = x_i \sum_{k\in N_i} w_{ik} \int \left(\prod_{j\neq i} q_j(x_j)\right) x_k d\mathbf{X}_{\neg i}$$

Consider expanding this integration over each term

$$\begin{aligned}
x_i \sum_{k\in N_i} w_{ik} \int \left(\prod_{j\neq i} q_j(x_j)\right) x_k d\mathbf{X}_{\neg i} &= x_i \sum_{k\in N_i} w_{ik} \int [q_1(x_1) \times \cdots \times q_N(x_N)] x_k dX_1 \dots dX_N \\
&= x_i \sum_{k\in N_i} w_{ik} \underbrace{\int q_1(x_1) dX_1}_{=1} \cdots \int q_k(x_l) x_k dX_k \dots \underbrace{\int q_N(x_N) dX_N}_{=1} \\
&= x_i \sum_{k\in N_i} w_{ik} \int q_k(x_k) dX_k \\
&= x_i \sum_{k\in N_i} w_{ik} \mathbb{E}_{q_k(x_k)}[X_k] \\
&= x_i \sum_{k\in N_i} w_{ik}\mu_k
\end{aligned}$$

Substituting this into the first set of integrals

$$\ln q_i(x_i) = \ln f_i(x_i) = L_i(x_i) + x_i \underbrace{\sum_{k\in N_i} w_{ik}\mu_k}_{=:m_i} + C = L_i(x_i) + x_i m_i + C$$

This means that

$$q(\mathbf{X}) \propto \prod_i e^{L_i(x_i)+x_i m_i}$$

We need to ensure this definition of $q(\mathbf{X})$ is an actual distribution (*i.e.* it integrates to 1). Since we have that $x_i \in \{1, -1\}$ only we can solve this easily

$$\begin{aligned}
\hat{q}_(x_i = 1) &= \frac{q(x_i = 1)}{q(x_i = 1) + q(x_i = -1)} \\
&= \frac{e^{m_i + L_i(1)}}{e^{m_i + L_i(1)} + e^{-m_i + L_i(-1}} \\
&= \frac{1}{1 + e^{-2m_i - L_i(1) + L_i(-1)}} \\
&= \frac{1}{1 + e^{-2\underbrace{\left(m_i + \frac{1}{2}L_i(1) - \frac{1}{2}L_i(-1)\right)}_{=:a_i}}} \\
&= \frac{1}{1 + e^{-2a_i}} \\
&= \text{Sigmoid}(2a_i) \\
\implies q_i(x_i = -1) &= \text{Sigmoid}(-2a_i)
\end{aligned}$$

Thus our proposed $q(\cdot)$ is a distribution.
We now need to consider how to undate μ_i

$$\begin{aligned}
\mu_i &= \mathbb{E}_{q_i(x_i)}[x_i] \\
&= \sum_{x_i \in [1,-1]} x_i q_i(x_i) \\
&= 1 \times q_i(x_i = 1) + (-1) \times q_i(x_i = -1) \\
&= \frac{1}{1 + e^{-2a_i}} - \frac{1}{1 + e^{2a_i}} \\
&= \frac{e^{a_i} - e^{-a_i}}{e^{a_i} + e^{-a_i}} \\
&= \tanh(a_i) \\
&= \tanh\left(m_i + \tfrac{1}{2}[L_i(1) - L_i(-1)]\right)
\end{aligned}$$

Note that tanh is similar in shape to a sigmoid but runs between -1 & 1 rather than 0 & 1.

0 Appendix

0.1 Definitions

Definition 0.1 - *Memory-Based Methods*
Memory-Based Methods for classification store the entire training set in order to make predictions for future data points. *e.g.* Nearest-Neighbours.
N.B. These generally require a distance measure to be defined.

Definition 0.2 - *Multinomial Distribution*
Multinomial Distribution is a generalisation of the *Binomial Distribution.*
It considers n independent trials where each results in one of k categories with the probability of each category being fixed.

$$\mathbb{P}(\mathbf{x};\mathbf{p}) = \begin{cases} \dfrac{n!}{x_1!\cdot\cdot\cdot\cdot\cdot x_k!} p_1^{x_1}\cdot\cdot\cdot\cdot\cdot p_k^{x_k}, & \text{if } \sum_{i=1}^k x_i = n \\ 0 & \text{otherwise} \end{cases} \quad \text{for } \mathbf{x} \in \mathbb{R}^k$$

N.B. Consider rolling a k sided dice n times

0.2 Proofs

Proof 0.1 - *Deriving Gaussian Marginal Distribution*
NOTE - This is dense as fuck & uses quite a bit of bullshit.

Let $\mathbf{X} \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\Lambda_{11} & \Lambda_{12}\\ \Lambda_{21} & \Lambda_{22}\end{pmatrix}^{-1}\right)$.
$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ can be considered as two parts of the mean vector $\boldsymbol{\mu}$.
Let $\mathbf{x}$ be a realisation of $\mathbf{X}$ where $\mathbf{x} := (\mathbf{x}_1, \mathbf{x}_2)$ with $\mathbf{x}_1$ & $\mathbf{x}_2$ representing the same partition as $\boldsymbol{\mu}_1$ & $\boldsymbol{\mu}_2$ respecitvely.
Define $D := \dim(\mathbf{x}),\ D_1 := \dim(\mathbf{x}_1)$ & $D_2 := \dim(\mathbf{x}_2)$.

Here we want to get from $\mathbb{P}(\mathbf{x}_1, \mathbf{x}_2)$ to $\mathbb{P}(\mathbf{x}_1)$.
Consider the exponent of the joint distribution

$$E = -\frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\Lambda_1 1(\mathbf{x}_1-\boldsymbol{\mu}_1) - \frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\Lambda_{12}(\mathbf{x}_2-\boldsymbol{\mu}_2) - \frac{1}{2}(\mathbf{x}_2-\boldsymbol{\mu}_2)^T\Lambda_{21}(\mathbf{x}_1-\boldsymbol{\mu}_1) - \frac{1}{2}(\mathbf{x}_2-\boldsymbol{\mu}_2)^T\Lambda_{22}(\mathbf{x}_2-\boldsymbol{\mu}_2)$$

To produce the marginal for x_1 we want to isolate the terms involving x_2 so they are easy to remove.

$$\begin{aligned}
E &= -\tfrac{1}{2}\Big[\left(\mathbf{x}_2^T\Lambda_{22}\mathbf{x}_2 - 2\mathbf{x}_2^T\Lambda_2 2(\boldsymbol{\mu}_2 - \Lambda_{22}^{-1}\Lambda_{21}(\mathbf{x}_1-\boldsymbol{\mu}_1))\right) \\
&- 2\mathbf{x}_1^T\Lambda_{12}\boldsymbol{\mu}_2 + 2\boldsymbol{\mu}_1^T\Lambda_{12}\boldsymbol{\mu}_2 + \boldsymbol{\mu}_2^T\Lambda_{22}\boldsymbol{\mu}_2 + \mathbf{x}_1^T\Lambda_{11}\mathbf{x}_1 \\
&- 2\mathbf{x}_1^T\Lambda_1 1\boldsymbol{\mu}_1 + \boldsymbol{\mu}_1^T\Lambda_{11}\boldsymbol{\mu}_1\Big] \\
&= \underbrace{-\frac{1}{2}\left(\mathbf{x}_2 - (\boldsymbol{\mu}_2 - \Lambda_{22}^{-1}\Lambda_{21}(\mathbf{x}_1-\boldsymbol{\mu}_1))\right)^T\Lambda_{22}\left(\mathbf{x}_2 - (\boldsymbol{\mu}_2 - \Lambda_{22}^{-1}\Lambda_{21}(\mathbf{x}_1-\boldsymbol{\mu}_1))\right)}_{E_1} \\
&+ \underbrace{\frac{1}{2}\left(\mathbf{x}_1^T\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\mathbf{x}_1 - 2\mathbf{x}_1^T\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_1^T\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\boldsymbol{\mu}_1\right)}_{A} \\
&- \underbrace{\frac{1}{2}\left(\mathbf{x}_1^T\Lambda_1 1\mathbf{x}_1 - 2\mathbf{x}_1^T\Lambda_{11}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_1\Lambda_{11}\boldsymbol{\mu}_1\right)}_{B}
\end{aligned}$$

Note that A & B do not contain any x_2 terms.
Since the co-variance matrix is symmetric we have $\Lambda_{12} = \Lambda_{21}^T$ we have

$$\mathbf{x}_1^T \Lambda_{12} \boldsymbol{\mu}_2 = \mathbf{x}_1^T \Lambda_{21}^T \boldsymbol{\mu}_2 = (\Lambda_{21}\mathbf{x}_1)^T \boldsymbol{\mu}_2 = \boldsymbol{\mu}_2^T \Lambda_{21} \mathbf{x}_1$$

We shall not rewrite A & B as quadratic expressions

$$\begin{aligned}
A &= \frac{1}{2}\left(\mathbf{x}_1^T \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\mathbf{x}_1 - 2\mathbf{x}_1^T\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_1^T\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\boldsymbol{\mu}_1\right) \\
&= \frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T(\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21})(\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
B &= \frac{1}{2}\left(\mathbf{x}_1^T\Lambda_{11}\mathbf{x}_1 - 2\mathbf{x}_1^T\Lambda_{11}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_1\Lambda_{11}\boldsymbol{\mu}_1\right) \\
&= \frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T\Lambda_{11}(\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
\implies \quad A - B &= \tfrac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T(\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21} - \Lambda_{11})(\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
\text{Let } E_2 &:= A - B
\end{aligned}$$

Now the exponent has been orgainised we can consider the whole gaussian expression.

$$\begin{aligned}
\mathbb{P}(\mathbf{x}_1, \mathbf{x}_2) &= \frac{e^{E_1}e^{E_2}}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \\
\mathbb{P}(\mathbf{x}_1) &= \int \mathbb{P}(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 \\
&= \int \frac{e^{E_1}e^{E_2}}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} d\mathbf{x}_2 \\
&= \frac{e^{E_2}}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \int e^{E_1} d\mathbf{x}_2 \quad \text{Since } E_2 \text{ is independent of } \mathbf{x}_2
\end{aligned}$$

Now we consider $\int e^{E_1} d\mathbf{x}_2$.
Since we know a gaussian must intergrate to 1 over the whole domain we deduce that

$$\begin{aligned}
\int \frac{1}{(2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}}} e^{E_1} d_{\mathbf{x}_2} &= 1 \\
\implies \int e^{E_1} d\mathbf{x}_2 &= (2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}}
\end{aligned}$$

N.B. Λ_{22}^{-1} is the variance of $\mathbf{x}_2$.
Using the result of this intergal we have

$$\begin{aligned}
\mathbb{P}(\mathbf{x}_1) &= (2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}} \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} e^{E_2} \\
&= \frac{e^{E_2}}{(2\pi)^{\frac{D-D_2}{2}}|\Lambda_{22}^{-1}|^{-\frac{1}{2}}|\Sigma|^{\frac{1}{2}}}
\end{aligned}$$

The Schur complement of Λ_{22} is $\Lambda_{22}^{-1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$.
Thus

$$\begin{aligned}
|\Lambda_{22}^{-1}|^{-\frac{1}{2}}|\Sigma|^{\frac{1}{2}} &= |\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}|^{-\frac{1}{2}}|\Sigma_{11}|^{\frac{1}{2}}|\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}|^{\frac{1}{2}} \\
&= |\Sigma_{11}|^{\frac{1}{2}}
\end{aligned}$$

Now we have a full expression

$$\begin{aligned}
\mathbb{P}(\mathbf{x}_1) &= \frac{e^{E_2}}{(2\pi)^{\frac{D-D_2}{2}}|\Lambda_{22}^{-1}|^{-\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \\
&= \frac{1}{(2\pi)^{\frac{D_1}{2}}|\Sigma_{11}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T\Sigma_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1)}
\end{aligned}$$

□

Proof 0.2 - *Deriving Gaussian Conditional Distribution*

Let $\mathbf{X} \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}\right)$.
$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ can be considered as two parts of the mean vector $\boldsymbol{\mu}$.
Let $\mathbf{x}$ be a realisation of $\mathbf{X}$ where $\mathbf{x} := (\mathbf{x}_1, \mathbf{x}_2)$ with $\mathbf{x}_1$ & $\mathbf{x}_2$ representing the same partition as $\boldsymbol{\mu}_1$ & $\boldsymbol{\mu}_2$ respecitvely.
Define $D := \dim(\mathbf{x})$.

We want to find the distribution of $\mathbb{P}(\mathbf{x}_1|\mathbf{x}_2)$.
From the product rule we know that $\mathbb{P}(\mathbf{x}_1, \mathbf{x}_2) = \mathbb{P}(\mathbf{x}_1|\mathbf{x}_2)\mathbb{P}(\mathbf{x}_2)$ and we already know the joint & marginal distributions for a gaussian.
We have that

$$\mathbb{P}(\mathbf{x}_1, \mathbf{x}_2) \propto e^{-\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}}$$

We now want to factor the marginal distribution out of this expression.

$$\mathbb{P}(\mathbf{x}_2) \propto e^{-\frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\mu}_2)^T \Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)}$$

Lets look at the exponent of the joint distribution.
N.B. About to use a lot of Schur Complements

$$\begin{aligned}
E &= -\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}\\
&= -\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}I & 0\\ \Sigma_{22}^{-1}\Sigma_{21} & I\end{pmatrix}^T \begin{pmatrix}(\Sigma/\Sigma_{22})^{-1} & 0\\ 0 & \Sigma_{22}^{-1}\end{pmatrix} \begin{pmatrix}I & -\Sigma_{12}\Sigma_{22}^{-1}\\ 0 & I\end{pmatrix} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}\\
&= -\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}(\Sigma/\Sigma_{22})^{-1} & -(\Sigma/\Sigma_{22})^{-1}\Sigma_{12}\Sigma_{22}^{-1}\\ -\Sigma_{21}\Sigma_{22}^{-1}(\Sigma/\Sigma_{22})^{-1} & \Sigma_{22}^{-1}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1\\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}\\
&= -\frac{1}{2}\Big[\mathbf{x}_1 - (\boldsymbol{\mu}_1 + \Sigma_{21}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))\Big]^T (\Sigma/\Sigma_{22})^{-1}\Big[\mathbf{x}_1 - (\boldsymbol{\mu}_1 + \Sigma_{21}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))\Big]\\
&\underbrace{-\frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\mu}_2)^T\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)}_{E_2}
\end{aligned}$$

Note that E_2 is exactly the exponent for the marginal distribution of $\mathbf{x}_2$ and thus what we want to factory out in order to get to the conditional distribution.

$$\mathbb{P}(\mathbf{x}_1|\mathbf{x}_2) \propto e^{-\frac{1}{2}\left[\mathbf{x}_1 - \underbrace{(\boldsymbol{\mu}_1 + \Sigma_{21}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))}_{\text{mean}}\right]^T \underbrace{(\Sigma/\Sigma_{22})}_{\text{covariance}}{}^{-1}\left[\mathbf{x}_1 - (\boldsymbol{\mu}_1 + \Sigma_{21}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))\right]}$$

□

0.3 Remarks

Remark 0.1 - *Worlds*
We can consider 3 different when answering an ml question.

i) Deterministic, $x = 4$;

ii) Point Estimate, $\text{argmax}_x p(x) = 4$;

iii) Stochastic, $p(x) \sim \text{Normal}(4, 10^2)$.