Machine Learning - Reviewed Notes

Dom Hutchinson

January 8, 2020

Contents

1 General

Remark 1.1 - *Machine Learning*
Machine Learning is a class of techniques which use algorithms & statistical models with the goal of producing systems which can perform a specific task without being given explicit instructions for how to do so. *Machine Learning* problems can be broken down into three sub-problems

i) How can we formulate beliefs & assumptions about a scenario mathematically?

ii) How can we connect our assumptions with observed data?

iii) How can we update our beliefs using the observed data?

Definition 1.1 - *Reasoning*
Reasoning is the process of applying premises which are known to be true to infer whether an unseen statement is true.

i) *Deductive Reasoning* assumes that the premises supply all the information required to make a definite conclusion about the truefullness of the unseen statement.
e.g. When it rains things outside get wet. Grass is outside therefore when it rains outside grass gets wet.

ii) *Inductive Reasoning* assumes that the premises supply some of the information and thus does not produce a definite conclusion.
e.g. The grass has gott wet when it rains, numerous times, therefore when it rains grass gets wet.

iii) *Abductive Reasoning* takes a conclusion & a rule and tries to infer the premises for them.
e.g. The grass is wet. Grass gets wet when it rains, therefore it has rained outside.

Definition 1.2 - *Bayesian Inference*
Bayesian Inference is a method of inference in which *Bayes' Theorem* is used to update a *posterior* distribution as more information about the *prior* & *likelihood* is discovered.

Definition 1.3 - *Intractable*
A problem is said to be *Intractable* if there is no known algorithm for solving it. There are two common reasons for a problem to be *Intractable*

i) *Computationally Intractable* - A given formulation is impossible to solve (typically due to an integration).

ii) *Analytically Intractable* - Due to some limitation in observation techniques we are not able to fully define a scenario.

Remark 1.2 - *Free-Lunch Theorem*

> *There are an infinite number of hypotheses that perfectly explain all the observed data. Adding a data point removes an infinite number of these possibilities, but still leaves an infinite number.*

Remark 1.3 - *Models do not need ot be perfectly true to be useful.*

Definition 1.4 - *Training Data*
Let $\mathbf{X} \in \mathbb{R}^{D\times N}$ represent N D-dimensional sets of independent variables and $\mathbf{y} \in \mathbb{R}^N$ be the outcomes (dependent variables) for each of these sets.
Training Data is the pair $(\mathbf{X}, \mathbf{y})$ which we use to learn the relationship between the domain and

output space of a scenario.
N.B. Test Data has the same structure as *Training Data* except we apply our learned process to the parameters and then compare our learned outcomes to the true outcomes in order to assess our model.

Definition 1.5 - *Model*
A *Model* is a system that models relationships between several variables. Generally this is occurs when consider independent & dependent variables.
Models are built upon two types of variables

i) *Observable Variables* - A variable whoses value can be observed directly.

ii) *Latent Variables* - A variable that cannot be directly observed, but a model is dependent upon it, thus we must infer their values from *Observable Variables*.

Definition 1.6 - *Discriminative Model*
Let X be a set of observed values.
A *Discriminative Model* aims to find the distribution for the outcomes, Y produced by X.

$$p(Y|X)$$

e.g. k-Nearest Neighbours, Neural Networks, Logistic Regression.

Definition 1.7 - *Generative Model*
Let Y be a set of observed outcomes.
A *Generative Model* aims to find a distribution for the variables, X, which produced Y.

$$p(X|Y)$$

e.g. Gaussian Mixture Model, Latent Dirichlet Allocation, everything else discussed here.

Definition 1.8 - *Mixture Model*
Mixture Models represent the presence of subpopulations within an overall population, without requiring the training data to specify which subpopulation each data point belongs to.
Sometimes it makes sense to interpret these subpopulations as clusters (*e.g.* k-means clustering), or as discrete latent variables which assign data points to specific subpopulations (*e.g.* Gaussian Mixtures Model).

1.1 Supervised Learning

Definition 1.9 - *Supervised Learning*
Let $(\mathbf{X}, \mathbf{y})$ be *Training Data*.
Supervised Learning is the process of learing a function, $f(\cdot)$, which maps the domain space to the output space.

$$y_i = f(\mathbf{x}_i)$$

N.B. Often we assume that there is some *Additive Noise*, $y_i = f(\mathbf{x}_i) + \varepsilon$ with $\varepsilon \sim \text{Normal}(0, 1)$.

1.1.1 Neural Network

Definition 1.10 - *Neural Network*
Neural Networks are a system that aims to learn how to perform a task without the rules being defined explicitly. A *Neural Network* is built layers of *neurons* where each neuron has a uni-directional edge to every node in the next layer. Each *neuron* assigns a weight to the edges

entering it & has a threshold function which determines whether it fires.

Proposition 1.1 - *Neural Network Layers*
The layers of a *Neural Network* are split into three categories

- *Input Layer* - We use inputted data to decide which of these *neurons* fires. There is one *neuron* for each possibility in the input space (*e.g.* one per pixel in an image).
- *Hidden Layers* - These lie between the *input* & *output layers* and do not necessarily have the same number of *neurons*. Each *neuron* a function which defined how much signal it sends to the next layer, dependent upon the signals it recieves.
- *Output Layer* - Has one *neuron* for each possible output class. The class of the *neuron* with the greatest value is returned as the result of the computation.

Proposition 1.2 - *Functions for Hidden Layer*
The function assigned to each *neurons* in the hidden layer should be defined st if they get enough stimuli from its connected *neurons* then it sends a signal to the next layer.
A common way to do this is to calculate

$$t := w_0 + \sum_{i=1}^{n} w_i z_i$$

where w_0 is a bias, w_i is the weight assigned to the i^{th} edge & z_i is the signal recived from the i^{th} edge, and then to pass a linear transformation of this through the *sigmoid* function.

Proposition 1.3 - *Training a Neural Network*
To train a *Neural Network* we typically define an *Error Function*, $E(\mathbf{w})$, which assesses how good our classifier is for a given set of weights. We then seek the set of weights, $\hat{\mathbf{w}}$, which minimises the *Error Function.*
A popular *Error Function* is the *Sum-of-Squares Error* function

$$\hat{\mathbf{w}} := \text{argmin}_{\mathbf{w}} E(\mathbf{w}) \text{ where } E(\mathbf{w}) := \frac{1}{2}\sum_{i=1}^{N} \|y(\mathbf{x}_i, \mathbf{w}) - \mathbf{t}_i\|^2$$

where $\{(\mathbf{x}_i, \mathbf{t}_i)\}$ is our *training data* and $y(\cdot)$ is the output our neural network gives.
N.B. Gradient Descent is a popular minimisation technique.

Remark 1.4 - *In many cases the error function is independent for each training item* $E(\cdot) = \sum_{n=1}^{N} E_n(\cdot)$

Proposition 1.4 - *Backpropogation*
Let $E(\cdot)$ be an *Error Function* used to evaluate our *Neural Network.*
Backpropogation is a technique for training a *Neural Network* and is based on evaluating the gradient of the *Error Function*, $\nabla E(\cdot)$.
Consider the simple case were

$$y_k(\mathbf{x}, \mathbf{w}) := \sum_i w_{ki} x_i \text{ and } E_n(\mathbf{w}) := \frac{1}{2}\sum_k (y_k(\mathbf{x}_n, \mathbf{w}) - t_{nk})^2$$

The gradient of the error function wrt a specific weight, w_{ji}, is

$$\frac{\partial E_n}{\partial w_{ji}} = (y_j(\mathbf{x}_n, \mathbf{w}) - t_{nj}) x_{ni}$$

This can be calculated for every edge & then the weight applied to each edge can be updated accordingly by adding/subtracting a certain amount of this gradient to its weight.

1.2 Un-Supervised Learning

Definition 1.11 - *Un-Supervised Learning*
Unsupervised Learning is the set of *Machine Learning* algoirthms which seek to make inferences from unlabeled data. Our task is to take a set of observations, $\mathbf{Y}$ and infer a *Latent Representation* for the model, $\mathbf{X}$, & a function, f, that maps from the *Latent Space* to the *Observed Space.*
N.B. Techinically defining $f(x) = x$ would suffice so we need to add some supervision (direction) to produce a meaningful result.

Remark 1.5 - *Latent Space,* $\boldsymbol{X}$
Since we are seeking the *Latent Sapce* we can make assumptions about its distribtion. Typically this is that each variable is centred around 0, is independent of all others & has variance 1.

$$\mathbf{X} \sim \text{Normal}(\mathbf{0}, I)$$

Remark 1.6 - *Preference*
In *Unsupervised Learning* we often talk about *Preference* rather than *belief.* The idea being that if we get two equally good solutions we choose the one that we belive is more likely, *i.e.* give greater preference to more likely solutions.
N.B. Preference is analoguous to *Priors.*

1.2.1 Principle Component Analysis

Definition 1.12 - *Principle Component Analysis*
Principle Component Analysis is an *unsupervised learning* technique which projects observations into the D dimensional plane which causes the greatest spread in the data.
Let $\mathbf{Y} \in \mathbb{R}^{N \times D}$ be a set of observed data and $\mathbf{X} \in \mathbb{R}^{N \times k}$ be the latent representation we wish to find

i) Calculate the co-variance matrix for $\mathbf{Y}$.

ii) Calculate the eigenvalues & vectors for the covaraince matrix.

iii) Pick the k largest eigenvalues & use their corresponding eigenvectors as rows of $\mathbf{W} \in \mathbb{R}^{D \times k}$.

iv) Calculate $\mathbf{X}$ st $\mathbf{Y} = \mathbf{W}\mathbf{X}^T$.

Proposition 1.5 - *Derivation of Principle Component Analysis*
Let $\mathbf{Y}$ be a set of observations, $\mathbf{X}$ be the set of latent variables we wish to infer & $\mathbf{W}$ be a set of weightings which define a linear relationship between $\mathbf{X}$ & $\mathbf{Y}$.
We wish to derive a posterior distribution for $\mathbf{W}$ & $\mathbf{X}$, our unknown parameters.
We define a *Likelihood Function*

$$p(\mathbf{Y}|\mathbf{W}, \mathbf{X}) = \text{Normal}\left(\mathbf{X}\mathbf{W} + \mu, \frac{1}{\beta}I\right)$$

where μ is a constant offest.
This leads to the following joint distribution

$$p(\mathbf{Y}, \mathbf{W}, \mathbf{X}) = p(\mathbf{Y}|\mathbf{W}, \mathbf{X})p(\mathbf{X})p(\mathbf{W})$$

We can now marginalise our $\mathbf{X}$ & $\mathbf{W}$

$$p(\mathbf{Y}) = \int p(\mathbf{Y}|\mathbf{W}, \mathbf{X})p(\mathbf{X})p(\mathbf{W})$$

This is intractable (*i.e.* we cannot solve for both $\mathbf{W}$ & $\mathbf{X}$) so we must take a point estimate of one & intergrate out the other.

Since we wish to be able to take unseen data & infer its latent location we need to be able to calculate $p(\mathbf{x}|\mathbf{y})$ and this can only be reached if we marginalise out $\mathbf{X}$ (*i.e.* estimate $\mathbf{W}$).

$$p(\mathbf{X},\mathbf{Y}|\mathbf{W})=p(\mathbf{Y}|\mathbf{W},\mathbf{X})p(\mathbf{X})$$

We know both the left hand terms

$$\begin{array}{rcl}p(\mathbf{Y}|\mathbf{W},\mathbf{X}) & = & \text{Normal}\left(\mathbf{XW}+\mu,\frac{1}{\beta}I\right)\\ p(\mathbf{X}) & = & \text{Normal}(\mathbf{0},I)\end{array}$$

Consider a pair of points $(\mathbf{x},\mathbf{y})$ then.

$$p(\mathbf{y},\mathbf{x}|\mathbf{W})=\text{Normal}\left(\begin{pmatrix}\mathbb{E}(\mathbf{y})\\ \mathbb{E}(\mathbf{x})\end{pmatrix}\begin{pmatrix}\mathbb{E}[(\mathbf{y}-\mathbb{E}(\mathbf{y}))(\mathbf{y}-\mathbb{E}(\mathbf{y}))^T] & \mathbb{E}[(\mathbf{y}-\mathbb{E}(\mathbf{y}))(\mathbf{x}-\mathbb{E}(\mathbf{x}))^T]\\ \mathbb{E}[(\mathbf{x}-\mathbb{E}(\mathbf{x}))(\mathbf{y}-\mathbb{E}(\mathbf{y}))^T] & \mathbb{E}[(\mathbf{x}-\mathbb{E}(\mathbf{x}))(\mathbf{x}-\mathbb{E}(\mathbf{x}))^T]\end{pmatrix}\right)$$

Note that

$$\mathbb{E}(\mathbf{y})=\mathbb{E}[\mathbf{Wx}+\mu+\varepsilon]=\mathbf{W}\mathbb{E}(\mathbf{x})+\mu+\mathbb{E}(\mu)=\mathbf{W}\times 0+\mu+0=\mu$$

Thus

$$\begin{array}{rcl}\mathbb{E}[(\mathbf{x}-\mathbb{E}(\mathbf{x}))(\mathbf{y}-\mathbb{E}(\mathbf{y}))^T] & = & \mathbb{E}[\mathbf{x}(\mathbf{y}-\mu)^T]\\ & = & \mathbb{E}[\mathbf{x}(\mathbf{Wx}+\mu+\varepsilon-\mu)^T]\\ & = & \mathbb{E}[\mathbf{x}(\mathbf{Wx})^T+\mathbf{x}\varepsilon^T]\\ & = & \mathbb{E}(\mathbf{xx}^T\mathbf{W}]+\mathbb{E}(\mathbf{x})\mathbb{E}(\varepsilon)\\ & = & \mathbb{E}(\mathbf{xx}^T)\mathbf{W}^T+0\\ & = & I\mathbf{W}^T\\ & = & \mathbf{W}^T\\ \implies \mathbb{E}[(\mathbf{y}-\mathbb{E}(\mathbf{y}))(\mathbf{x}-\mathbb{E}(\mathbf{x}))^T] & = & \mathbf{W}\\ \text{and } \mathbb{E}[(\mathbf{y}-\mathbb{E}(\mathbf{y}))(\mathbf{y}-\mathbb{E}(\mathbf{y}))^T] & = & \mathbb{E}[(\mathbf{Wx}+\mu+\varepsilon-\mu)(\mathbf{Wx}+\mu+\varepsilon-\mu)^T]\\ & = & \mathbb{E}[\mathbf{Wx}(\mathbf{Wx})^T+\mathbf{Wx}\varepsilon+\varepsilon\mathbf{Wx}^T+\varepsilon\varepsilon^T]\\ & = & \mathbb{E}[\mathbf{Wxx}^T\mathbf{W}]+\mathbb{E}[\mathbf{Wx}\varepsilon^T]+\mathbb{E}[\varepsilon\mathbf{x}^T\mathbf{W}^T]+\mathbb{E}(\varepsilon\varepsilon^T)\\ & = & \mathbf{W}I\mathbf{W}^T+\mathbf{W}\times 0+0\times\mathbf{W}^T+\sigma^2 I\\ & = & \mathbf{WW}^T+\sigma^2 I\\ \implies p(\mathbf{y},\mathbf{x}|\mathbf{W}) & = & \text{Normal}\left(\begin{pmatrix}\mu\\ \mathbf{0}\end{pmatrix},\begin{pmatrix}\mathbf{WW}^T+\sigma^2 I & \mathbf{W}\\ \mathbf{W}^T & I\end{pmatrix}\right)\\ \implies p(\mathbf{y}|\mathbf{W}) & = & \text{Normal}(\mu,\mathbf{WW}^T+\sigma^2 I)\\ \text{and } p(\mathbf{x}|\mathbf{y},\mathbf{W}) & = & \text{Normal}(\mathbf{W}^T(\mathbf{WW}^T+\sigma^2 I)^{-1}(\mathbf{y}-\mu),\ I-\mathbf{W}^T(\mathbf{WW}^T+\sigma^2 I)^{-1}\mathbf{W})\end{array}$$

Now we need to make estimate $\mathbf{W}$.
We can estimate the maximum likelihood solution for $\mathbf{W}$ by finding the eigenvalues & vectors for the covariance function of the observed data, $\mathbf{Y}$, and then using the eigenvectors with the D largest associated eigenvalues as rows of $\mathbf{W}$.[1].
Where D is the dimensionality we wish our result to have.

1.3 Parametric Models

1.3.1 Topic Model

Remark 1.7 - *Motivation*
Topic Models are based on the premise that certain types of data can be well desribed by atomic units within them. (*i.e.* A document can be well described by the words within it).

[1] This works since we assume that precision tends to infty, $\beta\to\infty$

Definition 1.13 - *Topic Models*
Topic Models are a class of models designed to discover the *topics* which occur in a collection of documents. *Latent Dirichlet Allocation* is a common topic model.

Remark 1.8 - *See **Proposition 2.6** for an implementation.*

1.4 Non-Parametric Models

1.4.1 k-Nearest Neighbours

Definition 1.14 - *k-Means Clustering*
k-Means Clustering produces k clusters by partitioning a set of n observations. *k-Means Clustering* assumes the covariance of each cluster is similar. *k-Means Clustering* aims to find a configuration of clusters that minimise *Within-Cluster Scatter.* A typical technique for *k-Means Clustering* is

1) Randomly initialise k vectors, $\{\mu_1, \dots, \mu_k\}$.
2) (a) Assign each training instance to its nearest μ_i.
   (b) Move each μ_i to be the centroid of the instances assigned to it.
3) Repeat 2) until no $\{\mu_1, \dots, \mu_k\}$ moves.

N.B. This finds a local minimum, not necessarily a global minimum.

Definition 1.15 - *k-Medoids Clustering*
k-Medoids Clustering is a variation on *k-Means Clustering* where, instead of using any point as a centre, it uses training observations as the centre of a cluster.

1.4.2 Gaussian Mixture Models

Definition 1.16 - *Mixture of Gaussians*
A *Mixture of Gaussians* is a superposition of K Gaussian distributions of the form

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \text{ where } \sum_{i=1}^{K} \pi_i = 1$$

where each distribution has its own *mean* & *co-variance* and is given a relative weighting, *Mixing Coefficient*, π_i.
N.B. Each distribution is referred to as a *Component* of the mixture.

Proposition 1.6 - *Gaussian Mixture Models*
Let $\mathbf{x}$ be distributed according to a *Mixture of Gaussians.*
Consider a binary k-dimensional random variable $\mathbf{z}$ where all but one dimension is zero-valued and define the marginal distribution $p(\mathbf{z})$ st $p(z_i = 1) = \pi_i$ (*N.B.* This is the same *Mixing Coefficient*).

Since $\mathbf{z}$ is zero-valued in all bar one dimension

$$\begin{aligned} p(\mathbf{z}) &= \prod_{i=1}^K \pi_i^{z_i} \\ p(\mathbf{x}|\mathbf{z}) &= \prod_{i=1}^K \text{Normal}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)^{z_i} \\ p(\mathbf{x}, \mathbf{z}) &= p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \\ &= \prod_{i=1}^K \pi_i^{z_i}\text{Normal}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)^{z_i} \end{aligned}$$

N.B. $\mathbf{z}$ represents the cluster that a data point belongs to.

Proposition 1.7 - *Fitting a Gaussian Mixtures Model*
Consider wishing to fit K D-dimensional components where $\mathbf{Y}_k \sim \text{Normal}(\boldsymbol{\mu}_k, \Sigma_k)$ for data points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$.
(In this case we the covariance matrices are diagonal).

i) $\forall\ k \in [1, K]$ random initalise $\mathbf{Y}_k$.

ii) $\forall\ i \in [1, n]$ associate $\mathbf{x}_i$ to component k where $\mathbb{P}(\boldsymbol{\mu} = \boldsymbol{\mu}_k, \Sigma = \Sigma_k|\mathbf{x}_i)$ is greatest.

iii) $\forall\ k \in [1, K],\ d \in [1, D]$ set $\mu_{k,d} = \dfrac{1}{n_k}\sum_{c_i=k} x_{i,d}$ and $\Sigma_{k,dd} = \dfrac{1}{n_k}\sum_{c_i=k}(x_{i,d} - \mu_{k,d})^2$.
where n_k is the number of data points assigned to component k.

iv) Repeat until convergence of $\mathbf{Y}_1, \dots, \mathbf{Y}_K$.

N.B. These are the sample mean & covariance. $\boldsymbol{\mu} \in \mathbb{R}^D,\ \Sigma \in \mathbb{R}^{D\times D}$.

2 Probability

2.1 General

Definition 2.1 - *Frequentist Probabiltiy*
Frequentist Probabiltiy is an interpretation of *Probability* where *Probability* refers to the limiting relative frequences of events. *Probabilities* are objective properties of the world.

$$P(X = x) = \lim_{n\to\infty} \frac{k}{n}$$

where k is the number of times an observation of x occurs in n samples.

Definition 2.2 - *Bayesian Probability*
Bayesian Probability is an interpretation of *Probability* where *Probability* is a reasonable expectation given our beliefs of the system. We encode these beliefs using the different components of *Bayes' Theorem.*

Remark 2.1 - *In this course we mainly deal with Bayesian Probability*

Definition 2.3 - *Random Variable*
A *Random Variable* is a function which maps an event in the sample space to a value.
X is a *Random Variable* if it satisfies the signature

$$X : \Omega \to \mathbb{R}$$

Definition 2.4 - *Probability Measure*
Probability Measures are functions which maps sets of events in a probability space to a value in $[0, 1]$. The returned value is a measure of how likely it is for a realisation of the random variable to fulfil the criteria placed upon it.

$$p(\cdot) : \mathcal{F} \to [0, 1]$$

and must fulfil the following criteria

i) $\forall\ A_0, \dots, A_n \in \mathcal{F} \implies \left(\bigcup_{i=0}^{n} A_i\right) \in \mathcal{F}$.

ii) $p(\Omega) = 1$.

iii) $p\left(\bigcup_{i=0}^{n}\right) = \sum_{i=0}^{n} p(A_i)$ for any n disjoint $A_0, \dots, A_n$.

Definition 2.5 - *Joint Probability Measure*
A *Joint Probability Measure* is a *Probabiltiy Measure* which is defined over multiple random variables and measures how likely it is that a realisation of the random variables fulfils <u>all</u> the criteria placed upon them.

$$p(\cdot, \cdot) : X \times Y \to [0, 1]$$

$$p(X = x, Y = Y) \equiv p(X = x \text{ and } Y = y)$$

Definition 2.6 - *Marginal Probability Measure*
A *Marginal Probability Measure* is a *Probabiltiy Measure* for a proper subset of the random variables in a scenario.
Suppose we have a scenario which is modelled by random variables X & Y then $p(X)$ is a *Marginal Probabiltiy Measure.*
Marginalisation is the process of reformulating the *Mariginal Distribution* of a single random variable to incorporate other random variables in the scenario

$$p(X) = \int p(X, Y = y)dy \text{ or } p(X) = \sum_i p(X, Y = y_i)$$

Definition 2.7 - *Conditional Probability Distribution*
A *Conditional Probability Distribution* is a *Probability Distribution* which measures the likelihood of a realisation of a random variable fulfilling a criteria placed upon it **given** that the realisation has already fulfilled a criteria placed upon another random variable.

$$p(X = x | Y = Y) \equiv p(X = x \text{ given we know that } Y = y)$$

N.B. If X & Y are independent then $p(X = x|Y = y) = P(X = x)$.

Definition 2.8 - *Independence*
Let X & Y be *Random Variables* and $p(\cdot)$ be a *Probabiltiy Distribution.*
X & Y are said to be *independent* of one another if

$$p(X = x, Y = y) = p(X = x).p(Y = y)$$

Definition 2.9 - *Expected Value*
Let X be a random variable, $p(\cdot)$ be a probability distribution for X and $f(\cdot) : \mathbb{R} \to \mathbb{R}$.
The *Expected Value* of a *Probability Distribution* is the mean value X.
If X is <u>continuous</u> then

$$\mathbb{E}(X) := \int_{-\infty}^{\infty} xp(x)dx \text{ and } \mathbb{E}(f(X)) := \int_{-\infty}^{\infty} f(x)p(x)dx$$

If X is discrete then

$$\mathbb{E}(X) := \sum_{x=-\infty}^{\infty} xp(x) \text{ and } \mathbb{E}(f(X)) := \sum_{x=-\infty}^{\infty} f(x)p(x)$$

Theorem 2.1 - *Linear Transformations of Expected Value*
Let X be a random variable and $a, b \in \mathbb{R}$. Then

$$\mathbb{E}(aX + b) = a\mathbb{E}(X) + b$$

Definition 2.10 - *Variance*
Let X be a random variable.
Variance measures how far a set of random numbers are spread from their expected value.

$$\text{Var}(X) := \mathbb{E}(X - \mathbb{E}(X))^2 = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2$$

where σ^2 & Σ are measures of variance.

Theorem 2.2 - *Linear Transformations of Variance*
Let X be a random variable and $a, b \in \mathbb{R}$. Then

$$\text{Var}(aX + b) = a^2\text{Var}(X)$$

Definition 2.11 - *Precision*
Precision is the reciprical of *Variance*.

$$\lambda := \frac{1}{\sigma^2}$$

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} & \dots & \Lambda_{1D} \\ \Lambda_{21} & \Lambda_{22} & \dots & \Lambda_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{D1} & \Lambda_{D2} & \dots & \Lambda_{DD} \end{pmatrix} := \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1D} \\ \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \dots & \Sigma_{DD} \end{pmatrix}^{-1}$$

Definition 2.12 - *Covariance*
Let X & Y be random variables.
Covariance measures the joint variability of two random variables.

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}(X))(Y - \mathbb{E}(Y))] = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$$

If X & Y are independent then $\text{Cov}(X, Y) = 0$.
N.B. By definition of *Covaraince* $\text{Cov}(X, X) = \text{Var}(X)$.

Definition 2.13 - *Conjugacy*
A *Prior* is said to be *Conjugate* if its distribution is in the same family of distributions as the distribution of the *Posterior*, in a given scenario.
N.B. Tables of *Conjugate Pairs* exist online.

Remark 2.2 - *Why use Conjugate Priors?*
If we have a *Conjugate Prior* then we can determine the distribution of the *Posterior* by passing the parameters of the *Prior* though pre-derived functions and thus avoid computing the *Evidence*, which is often difficult.

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

Remark 2.3 - *Usefulness of Conjugacy*
It is useful to use conjugacy as it allows us to avoid computing the *Evidence* (from Bayes' Rule) and simply multipling the prior & likelihood and then normalising to find the posterior.
However, in many cases, this is not possible and we have to perform a full calculation to reach the posterior.

2.2 Theorems

Theorem 2.3 - *Basic Rules*
Let X & Y be *Random Variables* and $p(\cdot)$ be a *Probabiltiy Distribution.*

Product Rule	$p(X=x,Y=Y)=p(Y=y\|X=x)p(X=x)$
Sum Rule	$p(X=x)=\sum_j p(X=x,Y=y_j)$

Theorem 2.4 - *Bayes' Theorem*
Let X & Y be *Random Variables* and $p(\cdot)$ be a *Probabiltiy Distribution.*
Bayes' Theorem states that

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

Remark 2.4 - *Components of Bayes' Theorem*
Let X & Y be *Random Varaibles* where Y is observed data & X represents the parameters of a theorised model of X, then the components of *Baye's Theorem* can each be considered to explain a different part of a model

$$\underbrace{p(X|Y)}_{\text{Posterior}}=\frac{\overbrace{p(Y|X)}^{\text{Likelihood}}\overbrace{p(X)}^{\text{Prior}}}{\underbrace{p(Y)}_{\text{Evidence}}}$$

Component	**Description**
Posterior	Which parameters are most likely to produce the data we observed.
Likelihood	How likely is the observed data given these parameters *N.B.* $p(X\|\theta)=L(\theta\|X)$
Prior	Our prior assumptions about the distribution of the parameters
Evidence	How likely is the observed data across all models *N.B.* This normalises the distribution of the posterior

N.B. The *Evidence* is normally the most challenging component of this relationship to calculate.

Theorem 2.5 - *Central Limit Theorem*
Let $\{X_n\}_{n\in\mathbb{N}}$ be a sequence of independent & indetically distributed with $\mathbb{E}(X_i)=\mu<\infty$ and $\text{Var}(X_i)=\sigma^2<\infty$. Then

$$\sqrt{\frac{n}{\sigma^2}}(Z_n-\mu)\to_{\mathcal{D}} Z\sim\text{Normal}(0,1)$$

2.3 Likelihood

Definition 2.14 - *Likelihood Function*
Let $\mathbf{X}\sim f_n(\cdot;\theta^*)$ for some $\theta^*\in\Theta$ and $\mathbf{x}$ be a realisation of $\mathbf{X}$.
*Likelihood Function*s are the family of functions which measure the probability of observing $\mathbf{x}$ given a value of the parameter θ.

$$L(\theta;\mathbf{x})\propto f_n(\mathbf{x};\theta)$$

N.B. AKA *Obsevered Likelihood Function.*

Definition 2.15 - *Log-Likelihood Function*
Let $L(\cdot)$ be a likelihood function.
The *Log-Likelihoof Function* of $L(\cdot)$ is the natural log of $L(\cdot)$

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

We use the *Log-Likelihood* since it removes exponentials and makes products into summations, making many problems more tractable.
N.B. $\ell(\cdot)$ is increasing with $L(\cdot)$.

Definition 2.16 - *Maximum Likelihood Estimate*
Let $\mathbf{X} \sim f_n(\cdot;\theta^*)$ for some $\theta^* \in \Theta$ and $\hat{\theta}$ be an esimate of the value of θ^*.
$\hat{\theta}$ is the *Maximum Likelihood Estimate* of θ^* if

$$\begin{aligned}\hat{\theta} &= \text{argmax}_{\theta\in\Theta} L(\theta;\mathbf{x})\\ &= \text{argmax}_{\theta\in\Theta} \ell(\theta;\mathbf{x})\end{aligned}$$

N.B. The *Maximum Likelihood Estimate* of θ is often denoted as $\hat{\theta}_{\text{MLE}}$.

Proposition 2.1 - *Finding the Maximum Likelihood Estimate*

i) Define a *Likelihood Function*, $L(\theta;\mathbf{x})$.

ii) Get the *Log-Likelihood Function*, $\ell(\theta;\mathbf{x}) := \ln L(\theta;\mathbf{x})$.

iii) Take the derivative wrt each parameter, $\frac{\partial}{\partial\theta}\ell(\theta;\mathbf{x})$.

iv) Set each derivative to 0 and solve to find stationary points for the parameters.

v) If a stationary point is a maximum then it is a *Maximum Likelihood Estimate*

Definition 2.17 - *Type II Maximum Likelihood*
In some cases we are only able to marginialise some of the variables in a problem.
Type II Maximum Likelihood is the process of marginialising out & intergrating over all the variables we can and then maximising the remaining.
N.B. This is used in Gaussian Processses.

2.4 Evidence

Remark 2.5 - *Also known as the Marginal Likelihood.*

Definition 2.18 - *Evidence*
The *Evidence* is the probability distribution that is left when we have integrated out everything except for the data.
Let Y be a set of observed data & θ be the set of parameters for a model we are theorising. Then the evidence is

$$\begin{aligned}p(Y) &= \int_\theta p(Y|\theta)p(\theta)d\theta\\ p(Y) &= \sum_\theta p(Y|\theta)p(\theta)\end{aligned}$$

N.B. The likelihood for observing the data we have, given all possible hypotheses about θ.

Proposition 2.2 - *Evidence of a Model*
Let $\mathcal{D}$ be the observable-space, M be a theorised model & θ be the parameters that M depends upon. Then the *Evidence* for M in $\mathcal{D}$ is the distribution

$$p(\mathcal{D}|M) = \int_\theta p(\mathcal{D}|M,\theta)p(\theta)d\theta$$

This requires us to define the *Prior*, $p(\theta|M)$. As we do not know much about the possible parameters, we allow for a large range of possible values of θ. One possibility is a zero-meaned

Gaussian with high variance.

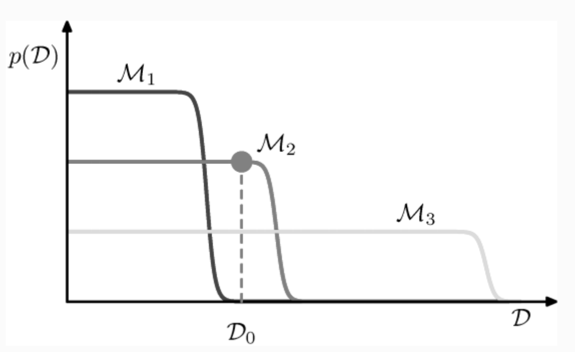
$$p(\theta|M) = \text{Normal}(\mathbf{0}, \sigma^2 I) \text{ for large } \sigma^2$$

Definition 2.19 - *Naïve Monte Carlo Approach*
Performing the integration defined in **Proposition 2.2** is generally tricky so we use the *Naïve Monte Carlo Approach* to approximate $p(\mathcal{D}|M)$.
Consider making samples, $\{\theta_1, \dots, \theta_S\}$, from the prior of the parameters.

$$p(\mathcal{D}|M) \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D}|M, \theta_s) \text{ where } \theta_s \sim p(\theta|M)$$

Remark 2.6 - *Comparing Models*
Suppose we observe some data Y, we can evaluate the probability of observing this under our *Evidence*. The greater this probability is the greater the probabilitiy that the model we have chosen is the *right* one.
This provides a metric for comparing multiple models and suggests choosing the model with the greatest likelihood, under our evidence, is better.

Remark 2.7 - *Choosing Models*
Suppose we can sort the observable-space in order of increasing complexity & can work out the evidence for each model, at each point in the observation-space. Then we should pick the simplest model which is able to explain our data.
Let $\mathcal{D}$ be the known observable-space, $\mathcal{D}_0$ be a reasonable point in the observation space where we may make our observations and $\mathcal{M}_1, \mathcal{M}_2$ & $\mathcal{M}_3$ be models which explain $\mathcal{D}$.



We see $\mathcal{M}_1$ is too simple to explain $\mathcal{D}_0$ & $\mathcal{M}_3$ is unnecessarily complex (*i.e.* explains too many factors). Thus we choose $\mathcal{M}_2$ in this case.
N.B. This is an application of *Occam's Razor*

2.5 Stochastic Processes

Definition 2.20 - *Stochastic Process*
A *Stochastic Process* is a collection of random variables, all defined in the same probabilty space & indexed by some set. Often the random variables are indexed by time or space.

2.5.1 Gaussian Processes

Definition 2.21 - *Gaussian Process*
Let S be a set.
A *Gaussian Process* on S is a set of random variables $\{Z_t\}_{t\in S}$ such that $\forall\, n \in \mathbb{N}$, $\forall\, t_1, \dots, t_n \in S$ the vector $\{Z_{t_1}, \dots, Z_{t_n}\}$ is a *Multivariate Gaussian.*
The distribution of a *Gaussian Process* is the continuous joint distribution of the random

variables which make it up.
For a *Gaussian Process* we define functions for calculating the mean, $\mu(\cdot)$, & covariance, $k(\cdot,\cdot)$. These functions only depend on the training data & a few hard-coded parameters thus *Gaussian Processes* are a non-parametric.
N.B. Covariance Functions are *Kernels.*

Remark 2.8 - *Verbose Definition of Gaussian Processes*
Gaussian Processes are the family of *Stochastic Processes* where each finite collection of the random variables has a *Multivariate Normal Distribution.*

Remark 2.9 - *Gaussian Processes is the infinite dimensional generalisation of a Gaussian Distribution*

Remark 2.10 - *Sampling from a Gaussian Process*
When we sample a *Gaussian Process* we are given a number of points from which we interpolate a function.

Proposition 2.3 - *Covariance Functions,* $k(\cdot,\cdot)$
Below are some common co-variance used for *Gaussian Processess*

Name	**General Form**
Constant	$k(x,y)=c$
Linear	$k(x,y)=x^Ty$
White Gaussian Nose	$k(x,y)=\sigma^2\delta_{x,y}$
Squard Exponential	$k(x,y)=\sigma^2e^{-\frac{1}{2\ell^2}(x-y)^T(x-y)}$
Periodic	$k(x,y)=e^{-\frac{2}{\ell^2}\sin^2\left(\frac{1}{2}(x-y)\right)}$

where σ is a standard deviation parmater, ℓ is a length-scale parameter for the function & $\delta_{x,y}$ is the *Kronecker Delta.*

Proposition 2.4 - *Mean Function,* $\mu(\cdot)$
We typically assume that the *Mean Function* is zero for all inputs as this simplifies a lot of calculations.

$$\mu(\mathbf{x})=0\ \forall\ \mathbf{x}$$

To ensure this is a valid assumption we centre our data around 0 by subtracting the sample mean from each data point.

$$\mathbf{x}'=\mathbf{x}-\bar{\mathbf{x}}$$

2.5.2 Dirichlet Processes

Definition 2.22 - *Dirichlet Processes*
Dirichlet Processes are the family of *Stochastic Processes* whose realisations are probability distributions over a set S.
The *Dirichlet Process* is a prior over countable infinite sets and used to generate a partitioning of a, possibly, infinite number of data points (Clusters). *Dirichlet Processes* are hard to define explicitly & here shall be defined constructively.
N.B. Dirichlet Processes are distributions over distributions over a set S.

Remark 2.11 - *Dirichlet Processes Parameters*
Dirichlet Processes are parameterised by a *base distribution*, $f(\cdot)$, and a *concentration parameter*, α.

Remark 2.12 - *Dirichlet Processes are the infinite dimensional generalisation of a Dirichlet Distribution*

Remark 2.13 - *Application of Dirichlet Processes*
Dirichlet Processes are used in scenarios where we do not know how many clusters our data has & we do not wish to hard code a specific number for this. *Dirichlet Processes* are able to grow the number of clusters as it recieves more data.

Proposition 2.5 - *Formulation of Dirichlet Processes - Chinese Restaurant Process*
Consider having an infinite number of tables (clusters) & dishes (labels for clusters) and a finite number of customers (data points).
We allow any number of customers to sit at a table, but only ever allow one dish per table.
We consider how to distribute customers between tables.
When a new customer arrives, the probability that they start a new table is

$$\frac{\alpha}{N+\alpha-1}$$

where α is a senstivity parameter that we set & N is the number of customers already in the restaurant.
If the customer chooses to not start a new table then they sit at table i with probabiltiy

$$\frac{n_i}{N}$$

where n_i is the number of customers already at table i & N is the number of customers already in the restaurant.
By varying α we can perform simulations to determine how any tables are used & how many people sit at each one.
Joining a new table is applied as sampling from the given distribution $f(\cdot)$ and joining an exisiting table is applied as sampling the value which formed the table.

Proposition 2.6 - *Formulation of Dirichlet Processes - Stick Breaking*
Consider taking a sample $\{x_1, x_2 \dots\}$ from Beta$(1,\alpha)$ ($x_i \in [0,1]\ \forall\ i$).
Imagine having a stick. Each of these sampled values tells you what proporition of the remaining stick to break off.
Thus after n samples $C_n := x_n \prod_{i=1}^{n-1} x_i$ of the stick remains. Each portion can be considered a cluster.
Take a sample $\{z_1, z_2, \dots\}$ from the given distribution $f(\cdot)$.
Merge the sample to form a new distribution $g(z) = \sum_{k\in\mathbb{N}} C_k \delta_{z_k,z}$.

Definition 2.23 - *Latent Dirichlet Allocation*
Latent Dirichlet Allocation, LDA, is a *Generative Model* used in natural language processing to find topics within a document. To do so, it makes the following assumptions

i) Documents contain a variety of topics.

ii) For each topic their is a distribution of words which characterises it.

iii) The order of words is irrelevant.

N.B. Based on the premise that certain data types can be well explained by atomic units with them.

Proposition 2.7 - *LDA Process*

1) Assign each word in each document to a random topic.

2) For each document, d

   (a) For each word, w, with current topic assignment t.

      i. Assume all topic assignments are correct, except for the word currently being considered.

      ii. Calculate two propotions

         - Proporition of words in d that are currentlty assign to t.

$$p(\text{Topic} = t|\text{Document} = d)$$

         - Proportion of occurences of w (across all documents) which are assigned to t.

$$p(\text{Word} = w|\text{Topic} = t)$$

      iii. Multiply the two proportions and assign w to a new topic based on the resulting probability.

$$p(\text{Topic} = t|\text{Document} = d) \times p(\text{Word} = w|\text{Topic} = t)$$

3) Repeat until convergence.

Proposition 2.8 - *Latent Dirichlet Allocation - Topic Model*
Let $\{\beta_1, \dots, \beta_T\}$ be the characteristic distribution of words for each possible topic and $w_{d,i}$ be the i^{th} word in the d^{th} document.
We want to find a distribution for the topics in each document, θ_d.
Define $z_{d,i}$ to be the topic we assign to the i^{th} word in the d^{th} document.
We have the following dependencies

$$p(w_{d,i}|\boldsymbol{\beta}, z_{d,i}) \quad p(z_{d,i}|\theta_d)$$

Since we assume each word is independent we have a distribution for all words in a document

$$p(\mathbf{w}_d|\boldsymbol{\beta}, \mathbf{z}_d) = \prod_{i=1}^{N} p(w_{d,i}|\boldsymbol{\beta}, z_{d,i})$$

We assume that the distribution of topics in documents is independent & the characteristic distribution for each topic is independent

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{D} p(\theta_d) \text{ and } p(\boldsymbol{\theta}) = \prod_{i=1}^{T} p(\beta_i)$$

Given a final joint distribution

$$p(w, z, \boldsymbol{\theta}, \boldsymbol{\beta}) = \underbrace{\prod_{t=1}^{T} p(\beta_t) \underbrace{\prod_{d=1}^{D} p(\theta_d) \underbrace{\prod_{i=1}^{N} p(w_{d,i}|\boldsymbol{\beta}, z_{d,i}) p(z_{d,i}|\theta_d)}_{\text{word}}}_{\text{document}}}_{\text{corpus}}$$

We still have to choose the form of each distribution. A good set of choices is

$$\begin{array}{rcl} \beta_k & \sim & \text{Dirichlet}(\eta) \\ \theta_d & \sim & \text{Dirichlet}(\alpha) \\ z_{d,i} & \sim & \text{Multi-Normal}(\theta_d) \\ w_{d,i} & \sim & \text{Multi-Normal}(\beta_{z_{d,i}}) \end{array}$$

where η & α are sensitivity parameters that we set

2.6 Distributions

Definition 2.24 - *β-Distribution*
Let $X \sim \text{Beta}(\alpha, \beta)$.
A *continuous* random variable with shape parameters $\alpha, \beta > 0$. Then

$$\begin{array}{rcl} f_X(x) & \propto & x^{\alpha-1}(1-x)^{\beta-1}\mathbb{1}\{x \in [0,1]\} \\ \mathbb{E}(X) & = & \dfrac{\alpha}{\alpha+\beta} \\ \text{Var}(X) & = & \dfrac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \\ \mathcal{M}_X(t) & = & 1 + \displaystyle\sum_{k=1}^{\infty}\left(\prod_{r=0}^{k-1}\frac{\alpha+r}{\alpha+\beta+r}\right)\frac{t^k}{k!} \end{array}$$

N.B. Beta-Distributions are used to encode our prior beliefs about the distribution of the parameters.

Definition 2.25 - *Bernoulli Distribution*
Let $X \sim \text{Bernoulli}(p)$.
A *discrete* random variable which takes 1 with probability p & 0 with probability $(1-p)$. Then

$$\begin{array}{rcl} p_X(k) & = & \begin{cases} 1-p & \text{if } k=0 \\ p & \text{if } k=1 \\ 0 & \text{otherwise} \end{cases} \\ P_X(k) & = & \begin{cases} 0 & \text{if } k<0 \\ 1-p & \text{if } k\in[0,1) \\ 1 & \text{otherwise} \end{cases} \\ \mathbb{E}(X) & = & p \\ \text{Var}(X) & = & p(1-p) \\ \mathcal{M}_X(t) & = & (1-p)+pe^t \end{array}$$

N.B. Often we define $q := 1-p$ for simplicity.

Definition 2.26 - *Binomial Distribution*
Let $X \sim \text{Binomial}(n, p)$.
A *discrete* random variable modelled by a *Binomial Distribution* on n independent events and rate of success p.

$$\begin{array}{rcl} p_X(k) & = & \binom{n}{k}p^k(1-p)^{n-k} \\ P_X(k) & = & \sum_{i=1}^k \binom{n}{i}p^i(1-p)^{n-i} \\ \mathbb{E}(X) & = & np \\ \text{Var}(X) & = & np(1-p) \\ \mathcal{M}_X(t) & = & [(1-p)+pe^t]^n \end{array}$$

N.B. If $Y := \sum_{i=1}^n X_i$ where $\mathbf{X} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$ then $Y \sim \text{Binomial}(n, p)$.

Definition 2.27 - *Dirichlet Distribution*
Let $\mathbf{X} \sim \text{Dir}(\boldsymbol{\alpha})$ with $\mathbf{X}, \boldsymbol{\alpha} \in \mathbb{R}^N$.

A *continuous* random vector with concentration parameters $\boldsymbol{\theta}$ with $\alpha_i > 0$.

$$\begin{aligned} p_X(k) &= \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\times\cdots\times\Gamma(\alpha_N)}\prod_{i=1}^N x_i^{\alpha_i-1} \\ \mathbb{E}(X_i) &= \frac{\alpha_i}{\sum_{j=1}^N \alpha_j} \\ \text{Var}(X_i) &= \frac{\mathbb{E}(X_i)-1}{(\sum_{j=1}^N \alpha_i)-N} \end{aligned}$$

N.B. The *Dirichlet Distribution* is a multivariate generalisation of a *Beta Distribution.*

Definition 2.28 - *Exponential Distribution*
Let $X\sim\text{Exponential}(\lambda)$.
A *continuous* random variable modelled by a *Exponential Distribution* with rate-parameter λ. Then

$$\begin{aligned} f_X(x) &= \mathbb{1}\{t\geq 0\}.\lambda e^{-\lambda x} \\ F_X(x) &= \mathbb{1}\{t\geq 0\}.\left(1-e^{-\lambda x}\right) \\ \mathbb{E}(X) &= \frac{1}{\lambda} \\ \text{Var}(X) &= \frac{1}{\lambda^2} \\ \mathcal{M}_X(t) &= \mathbb{1}\{t<\lambda\}\frac{\lambda}{\lambda-t} \end{aligned}$$

Definition 2.29 - *Normal Distribution*
Let $X\sim\text{Normal}(\mu,\sigma^2)$.
A *continuous* random variable with mean μ & variance σ^2.

$$\begin{aligned} f_X(x) &= \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ F_X(x) &= \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{x} e^{-\frac{(y-\mu)^2}{2\sigma^2}}dy \\ \mathbb{E}(X) &= \mu \\ \text{Var}(X) &= \sigma^2 \\ \mathcal{M}_X(\theta) &= e^{\mu\theta+\sigma^2\theta^2(1/2)} \end{aligned}$$

Definition 2.30 - *Multivariate Normal Distribution*
Let $\mathbf{X}\sim\text{Normal}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ for $\boldsymbol{\mu}\in\mathbb{R}^N$ & $\boldsymbol{\Sigma}\in\mathbb{R}^{N\times N}$.
A *continuous* random vector with mean values $\boldsymbol{\mu}$ and co-variance matrix $\boldsymbol{\Sigma}$.

$$\begin{aligned} f_{\mathbf{X}}(\mathbf{x}) &= \frac{1}{(\sqrt{2\pi})^k}\det(\boldsymbol{\Sigma})^{-\frac{1}{2}}e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \\ \mathbb{E}(\mathbf{X}) &= \boldsymbol{\mu} \\ \text{Var}(\mathbf{X}) &= \boldsymbol{\Sigma} \\ \mathcal{M}_{\mathbf{X}}(\mathbf{t}) &= e^{\boldsymbol{\mu}^T\mathbf{t}+\frac{1}{2}\mathbf{t}^T\boldsymbol{\Sigma}\mathbf{t}} \end{aligned}$$

2.7 Gaussian Indentities

Remark 2.14 - *Motivation*
Lots of scenarios can be modelled by a Normal distribution, particular due to the *Central Limit Theorem.* Thus it is useful to know common results involving *Normal* Distributions.

Remark 2.15 - *Decomposition of Covariance Matrix*
Let $\mathbf{X}\sim\text{Normal}(\boldsymbol{\mu},\boldsymbol{\Sigma})$. Then

$$\boldsymbol{\Sigma}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \implies \boldsymbol{\Sigma}^{-1}=\boldsymbol{U}^{-T}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{-1}=\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T$$

where $\boldsymbol{U}$ is an orthonormal matrix (*i.e.* $U^T = U^{-1}$) & $\boldsymbol{\Lambda}$ is a diagonal matrix.
By further derivations of the exponential of the PDF of $\mathbf{X}$ we find that

$$(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = (\boldsymbol{U}^T\boldsymbol{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}))^T(\boldsymbol{U}^T\boldsymbol{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}))$$

This means we only need to consider $(\boldsymbol{U}^T\boldsymbol{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}))$ which is a general linear mapping and can be interpretted as a rotation of the basis.

Remark 2.16 - *Independent Multivariate Gaussians*
Let $\mathbf{X} \in \mathbb{R}^{N\times D}$ represent N independent data points, each of D dimensions.
Suppose $\mathbf{X}_i \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then

$$\begin{aligned}\mathbb{P}(\mathbf{X}) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{X}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{X}_i-\boldsymbol{\mu})\right)\\ &= \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}\text{tr}[(\mathbf{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})]\right)\end{aligned}$$

3 Regression Analysis

Definition 3.1 - *Regression Analysis*
Regression Analysis is the set of techniques for estimating the relationship, f, between independent, $\mathbf{X}$, and dependent, $\mathbf{y}$, variables.

$$\mathbf{y} = f(\mathbf{X})$$

Proposition 3.1 - *Using a Learned Regression Model*
Suppose we have learned a distribution for the possible functions f which approximates the relationship between the dependent & indepenet variables, $\mathbf{y} = f(\mathbf{X})$, and let $\mathbf{x}$ be a realisation of $\mathbf{X}$ that we want to predict the value of $\mathbf{y}$ for.
Then we want to find the *most likely* value of $\mathbf{y}$, under $\mathbf{x}$, for our learned function.
This is just the expected value weighted by the likelihood of each possibility of f

$$\hat{\mathbf{y}} = \mathbb{E}_{p(f)}[\mathbf{y}|f,\mathbf{x}] = \int f(\mathbf{x})p(f)df$$

3.1 Linear Regression

Definition 3.2 - *Linear Regression Analysis*
Linear Regression Analysis is a subset of *Regression Analysis* which concerns just scenarios where we assume there is a linear relationship between the independent & dependent variables.

$$\mathbf{y} = \mathbf{a}^T\boldsymbol{\phi}(\mathbf{X}) + \varepsilon$$

where $\mathbf{X}$ is a D-dimensional random vector and $\varepsilon \sim \text{Normal}(0,\sigma^2)$ for some $\sigma > 0$ represents additive noise.
Here $\boldsymbol{\phi}(\cdot) : \mathbb{R}^D \to \mathbb{R}^E$ is a vector of *Basis Functions* & $\mathbf{a} \in \mathbb{R}^{E+1}$ is a set of parameters for us to learn.

Proposition 3.2 - *Linear Regression Analysis - Posterior*
Given our assumption of the relationship between $\mathbf{y}$ and $\mathbf{X}$ we have the following posterior

$$\mathbb{P}(y|\mathbf{a},\boldsymbol{\phi},\mathbf{x}) \sim \text{Normal}(\mathbf{a}^T\boldsymbol{\phi}(\mathbf{x}),\sigma^2)$$

By Bayes' Theorem our task is simplified to specifying a *Prior* distribution, $\mathbb{P}(\mathbf{a})$.

Proposition 3.3 - *Linear Regression Analysis - Prior*
We use *conjugacy* when choosing our *Prior*. Since the *Posterior* is normally distributed, we chooes a normal distribution for the *Prior.*

Theorem 3.1 - *Linear Least-Squares Method*
Linear Least-Squares Method is a method for finding the linear line which minimises the *Residual Sum of Squares* (*i.e.* the line which minimises $\sum_i(\hat{y}_i - y_i)^2$).

i) Let $(\mathbf{X}, \mathbf{y})$ be a set of training data and $\boldsymbol{\phi}(\cdot) : \mathbb{R}^D \to \mathbb{R}^E$ be a vector of *Basis Functions.*

ii) Define matrix Z to be $\boldsymbol{\phi}$ applied to X st $Z_{ij} = \phi_j(X_i)$.

iii) The General *Least-Squares Estimate* for the parameter values is given as

$$\hat{\mathbf{a}}_{\text{LS}} = (Z^T Z)^{-1} Z^T \mathbf{y}$$

N.B. Typically we add a column of 1s to Z which represents a constant (*i.e.* bias).

Proposition 3.4 - *Linear Regression in 2 Dimensions*
Suppose there is a single independent & a single dependent variable

$$\hat{a}_{\text{LS}} = \bar{\mathbf{y}} - \hat{b}_{\text{LS}}\bar{\mathbf{x}} \text{ and } \hat{b}_{\text{LS}} = \frac{(\sum_i x_i y_i) - n\bar{\mathbf{x}}\bar{\mathbf{y}}}{(\sum_i x_i^2) - n\bar{\mathbf{x}}^2}$$

3.2 Kernel Methods

Definition 3.3 - *Duals*
Duals a general idea of problems which can be reformulated using a different variable basis, kernels can be used to reformulate data.

Definition 3.4 - *Kernel*
A *Kernel* is a function which takes in two vectors, applies the same transformation to both of them & then returns the inner-product of these transformations.

$$\begin{aligned}&k(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}\\ &k(\mathbf{x}, \mathbf{y}) := \phi(\mathbf{x})^T \phi(\mathbf{y}) \text{ for some } \phi : \mathbb{R}^n \to \mathbb{R}^m\end{aligned}$$

*Kernel*s are symmetric (*i.e.* $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$) by the symmetry of the inner-product.
*N.B. Kernel*s allow us to never realise the whole space & thus can work in infinite dimensional spaces.

Definition 3.5 - *Common Kernels*
Below are some common traits of *Kernel*s which have particular names

Name	**Trait**
Linear Kernel	Identity function is used as the transform *i.e.* $\phi(\mathbf{x}) = \mathbf{x} \implies k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$.
Stationary Kernel	Value only depends on the difference of the arguments. *i.e.* $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$.
Homogeneous (AKA Radial Basis Functions)	Value only depends on the *magnitude* of the difference of the arguments. *i.e.* $k(\mathbf{x}, \mathbf{y}) = k(\|\mathbf{x} - \mathbf{y}\|)$.

Definition 3.6 - *Kernel Regression*
Kernel Regression is the process of performing linear regression in an induced space.

This allows us to learn non-linear functions by reparametising the data.

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

where

- $\mathbf{x}$ is the unseen data-point we wish to estimate $\mathbf{y}$ for.
- $\mathbf{X}$ is the training data.
- $\mathbf{y}$ is the observations associated to the training data.
- $k(\cdot, \cdot)$ is a pre-defined kernel function.
- λ is a parameter we set according to the noise we assume the data to have.

N.B. This formulation is the least-squares solution.

Remark 3.1 - *Kernel Regression*
Kernel regression is a non-parameteric method as it does require the training of a parameter vector, $\mathbf{a}$, instead it uses the *training data* directly.

3.3 Gaussian Processes

Remark 3.2 - *Motivation*
Gaussian Processes offer a non-parameteric approach to *Linear Regression* as it finds a distribution over the possible functions of f, where we believe $y = f(x) + \varepsilon$ for some linear function f, by first considering every possible function & then putting weight towards those that are consistent with the observed data.

Proposition 3.5 - *Gaussian Processes Regression*
We need to specify a *prior*, over functions (our domain space), in order to constrain the functions we consider. Generally we constrain covariance function in order to get functions which are somewhat smooth.

Remark 3.3 - *We are not talking about the joint probability of two variables but the joint probability of the values of $f(x)\ \forall\ x \in \mathcal{X}$*

Proposition 3.6 - *Gaussian Process Prior*
Let $\mathbf{f} := (f_1, \dots, f_N)$ be possble values of our function, evaluated at N different points, and $\mathbf{x} := (x_1, \dots, x_n)$ be N different points in the domain space.
Since f_i is unknown we assume the following form & distribution of f_i

$$f_i = g(x_i) \text{ for some } g(\cdot) \text{ and } f_i \sim \text{Normal}(\mu(x_i), k(x_i))$$

where $\mu(\cdot)$ evaluates the mean & $k(\cdot)$ evaluates the variance at a given point.
We assume each function value is jointly normally distributed. Thus

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} \sim \text{Normal}\left(\begin{pmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_N) \end{pmatrix}, \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{pmatrix} \right)$$

where $\mu(\cdot)$ evaluates the mean & $k(\cdot, \cdot)$ evaluates the covariance at given points.
To produce a *Gaussian Process* we consider evaluating this as an infinite collection of random

variables, *i.e.* $N \to \infty$.

Proposition 3.7 - *Gaussian Predicitve Posterior*
Let $\mathbf{x} := (x_1, \dots, x_N)$ be a set of points for which we have observations, $\mathbf{f} := (f_1, \dots, f_N)$.
Consider wishing to predict the values of the functon, $\mathbf{f}^* := (f_1^*, \dots, f_M^*)$ at points $\mathbf{x}^* := \{x_1^*, \dots, x_M^*\}$.
If we apply the product rule & factorise the joint distribution we get

$$\begin{array}{rrcl} & p(\mathbf{f}, \mathbf{f}^*|\mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta}) & = & p(\mathbf{f}^*|\mathbf{y}, \mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta})p(\mathbf{f}|\mathbf{x}, \boldsymbol{\theta}) \\ \implies & p(\mathbf{f}^*|\mathbf{f}, \mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta}) & = & \dfrac{p(\mathbf{f}|\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{f}, \mathbf{f}^*|\mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta})} \\ \implies & p(\mathbf{f}^*|\mathbf{f}, \mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta}) & = & \text{Normal}\Big(k(\mathbf{x}^*, \mathbf{x})k(\mathbf{x}, \mathbf{x})^{-1}\mathbf{f},\ k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{x})k(\mathbf{x}, \mathbf{x})^{-1}k(\mathbf{x}, \mathbf{x}^*)\Big) \end{array}$$

where we assume $\mu(\mathbf{x}) = 0\ \forall\ \mathbf{x}$ and $k(\cdot, \cdot)$ is some *Covariance Function.*
More generally we get that

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \text{Normal}\left(\begin{pmatrix} \mu(\mathbf{x}) \\ \mu(\mathbf{x}^*) \end{pmatrix}, \begin{pmatrix} K(\mathbf{x}, \mathbf{x}) & K(\mathbf{x}, \mathbf{x}^*) \\ K(\mathbf{x}^*, \mathbf{x}) & K(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix}\right)$$

where $K(\cdot, \cdot)$ is the matrix formed by evaluating the covariance function between each point in the two given vectors.

Proposition 3.8 - *Gaussian Predictive Posterior with Additive Noise*
The definition in **Proposition 3.7** does not allow for additive noise around the training data, $\{\mathbf{x}, \mathbf{f}\}$.
We can implement additive noise by altering the joint distribution to have variance at each training point

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} \sim \text{Normal}\left(\begin{pmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_N) \end{pmatrix}, \begin{pmatrix} k(x_1, x_1) + \frac{1}{\beta} & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) + \frac{1}{\beta} & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) + \frac{1}{\beta} \end{pmatrix}\right)$$

where β is a precision parameter.
This gives us a predicitve posterioir

$$p(\mathbf{f}^*|\mathbf{y}, \mathbf{x}, \mathbf{x}^*, \boldsymbol{\theta}) = \text{Normal}\left(k(\mathbf{x}^*, \mathbf{x})\left(k(\mathbf{x}, \mathbf{x}) + \frac{1}{\beta}I\right)^{-1}\mathbf{y},\quad k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{x})\left(k(\mathbf{x}, \mathbf{x}) + \frac{1}{\beta}I\right)^{-1}k(\mathbf{x}, \mathbf{x}^*)\right)$$

4 Optimisation

Definition 4.1 - *Black-Box Function*
A *Black-Box Function* is a function which we can supply inputs to and can observe outputs from but have no knowledge of the internal workings. Often this is due to the function being intractable and is common in real life scenarios.
N.B. Black-Box Functions are said to be *"explicity unknown"*.

Definition 4.2 - *Optimisation Problem*
Given a scenario, find the inputs wwhich produce the most optimal outcome (Typically we take the value which *minimises* the scenario to be optimal). This problem can be broken down into two situations

i) *Classical Optimisation* - Given an explicit function, $f(\cdot)$, find $\hat{x} := \text{argmin}_x f(x)$.

ii) *Black Box Optimisation* - Given a *Black-Box Function* find a global minimum.

N.B. Finding a global minimum is analogous to finding a global maximum (consider multiplying all outcomes by -1).

Remark 4.1 - *Justification*
It is possible to solve the *Black Box Optimisation Problem* by evaluating the function at many different points. IRL this would be done by evaluating (*e.g.* Taking blood tests) under different scenarios, however it is likely that each evaluation is expensive to undergo and thus we wish to minimise the number of evaluations required to get a good answer.
N.B. We need to balance our exploration & explotation of the domain space.

4.1 Bayesian Optimisation

Definition 4.3 - *Bayesian Optimisation*
Bayesian Optimisation is a sequential algorithm for solving the *Black-Box Optimisation Problem* & seeks to minimise the number of evaluations required to do so.
Each iteration of *Bayesian Optimisation* updates two functions:

i) A *Surrogate Model* - Our current belief for the shape of the black-box function.

ii) An *Acquisition Funtion* - Decides where to evaluate next, given our *surrogate function.*

Proposition 4.1 - *Suggested Surrogate Models*
Typically we define a *Prior* which forms the initial *Surrogate Model* and then update this model using the *Posterior* produced by each iteration.
Often *Gaussian Processes* are used for the *surrogate model* as they give access to the predicitve posterior of f, $p(f|x,\mathcal{D})$.

Definition 4.4 - *Acquistion Functions*
An *Acquisition Function* encodes our strategy for utilising our current knowldege of the function.
Typically we define a *Utility function*, $u(x)$, which depends on a probabilistic model (usually the surrogate model) and define the *Acquisition Function* to be the expcted value of the utiltiiy function, $\alpha := \mathbb{E}[u(x)|x,\mathcal{D}]$.
We then seek to find the value of $\hat{x} := \text{argmin}_{x\in\mathcal{X}}\alpha(x)$.

Proposition 4.2 - *Suggested Acquisition Functions*
When designing an *Acquistion Function* we should consider how we balance exploration & explotation of the domain space.

i) *Expected Improvement* - The utility of exploring an given point, x, in the domain space increase with the level of expected improvement of the function.

$$u(x) := \max\{0, f(\hat{x}) - f(x)\}$$

where $\hat{x}$ is our current best guess for the global minimum & $f(\hat{x})$ is the value of $f(\cdot)$ at $\hat{x}$ (which is known explicitly) & $f(x)$ is the expected value of $f(\cdot)$ at x under our *surrogate model.*
Then

$$EI(x|\mathcal{D}) := \alpha(x) = \mathbb{E}[u(x)|x,\mathcal{D}] = \int_{-\infty}^{f(\hat{x})}[f(\hat{x}) - f(x)]p(f|x,\mathcal{D})df$$

We should then evaluate at the point x where $EI(x) \geq EI(x')\ \forall\ x' \in \mathcal{X}$.

ii) *Thomson Sampling*

$$\alpha(x) = g(x) \sim p(f(x)|x,\theta,\mathcal{D})$$

iii) *Upper Confidence Bound*

$$\alpha(x)=\mu(x)-\beta\sigma(x)$$

where $\beta>0$ and $\mu(\cdot)$ & $\sigma(x)=\sqrt{k(x,x)}$ are the mean & standard devatiation functions of $f(x)$ defined by the surrogate model.

Remark 4.2 - *Expected Improvement \w Gaussian Processes Surrogate Model*
If we use *Gaussian Processes* for the *Surrogate Model* we can further evaluate the *Expected Improvement*

$$EI(x|\mathcal{D})=[f(\hat{x}-\mu(x)]\Psi(\hat{f}|\mu(x),k(x,x))+k(x,x)\mathcal{N}(\hat{f}|\mu(x),k(x,x))$$

where $\Psi(\cdot)$ is the cummulative defensity function for a normal distribution.
From this definition we see we can increase its values by picking points, x, where the difference $f(\hat{x})-f(x)$ is large (exploiting knowledge) or $k(x,x)$ is large (exploration).

Proposition 4.3 - *Bayesian Optimisation Process*
Let $f(\cdot)$ be a black-box function we wish to evaluate.
Define $\mathcal{D}:=\{\mathbf{x},\mathbf{y}\}$ be the points we have evaluated f at, along with the outputs they produced, and $\hat{x}$ to be our current best guess for the maximum of f.

i) Pick a random set of start-points.

ii) Evaluate f at each of the start points.

iii) Find the minimum value of these evaluations, $f(\hat{x})$.

iv) For a set number of iterations

   (a) Calculate the *Predictive Posterior* of the *Surrogate Model.*
   (b) Evalue the *Acquisition Function*, $\alpha(\cdot)$, wrt the *Predictive Posterior.*
   (c) Pick the point which maximises the *Acquisition Function*, x'.
   (d) Evaluate f at this point, $f(x')$.
   (e) If $f(x')<f(\hat{x})$ then set $\hat{x}=x'$.

v) return $f(\hat{x})$.

5 Approximative Inference

Definition 5.1 - *Approximative Inference*
Approximative Inference is a group of techniques used to approximately learn models when exact learning is intractable. Exact learning is typically intractable due to the *Evidence* being intractable.
Approximative Inference can be split into two categories

i) Stochastic - Typically would produce an exact result with infinite computational resources but only generate an approximate result due to the use of finite processor time.
e.g. Markov Chain Monte Carlo, Sampling

ii) Deterministic - Based on making analytical approximations to the posterior distribution (*e.g.* assuming that it factorises in a particular way) and as such will could never produce an exact result.
e.g. Variational Inference, Variational Bayes, Laplace Approximation

N.B. The strengths & weaknesses of *Stochastic* and *Deterministic* techniques are complementary.

Definition 5.2 - *Monte Carlo Methods*
Monte Carlo Methods are a set of methods which use sampling to make numerical estimations of parameters in a distribution.
Monte Carlo Methods follow the following steps

i) Determine how to sample inputs.

ii) Generate many sets of possible inputs

iii) Perform a deterministic calculation with these sets.

iv) Analyse the result statistically.

N.B. The error of the result typically decreases as $\frac{1}{\sqrt{N}}$.

5.1 Stochastic Techniques

Definition 5.3 - *Stochastic Techniques for Approximative Inference*
Stochastic Technqiues for *Approximative Inference* rely on using probabilistic distributions. It is hard to know how well our approximation is doing during computation and these techniques are typically very slow, but in theory will reach an exact definition eventually.

5.2 Laplace Approximation

Definition 5.4 - *Laplace Approximation*
Laplace Approximation is a *Stochastic Approximative Inference* techniques which aims to find a *Normal* approximation for probability densities defined over a set of continuous variables. The motivation for this is that under most conditions a posterior is asymptotically normally distributed as the number of data points goes to ∞ (Central Limit Theorem).

Proposition 5.1 - *Laplace Approximation Process*

i) Find *mode* of *posterior.*

ii) Set the *mode* to be the *mean* of the *normal* distribution we are trying to fit.

iii) Find the *variance* of *posterior* around this *mode.*

iv) Set the *variance* around the *mode* to be the *variance* of the normal distribution.

N.B. This can easily be extended to the *multivariate* case by performing on each dimension independently.

Proposition 5.2 - *Finding the Mode*
Let $p(\cdot)$ be a probability distribution that we wish to find the *mode* of. This is analogous to find the global maximum.
Consider the function

$$f(x) = e^{Mp(x)}$$

Suppose x' is the *mode* of $p(\cdot)$ and consider the ratio of $f(\cdot)$.

$$\frac{f(x')}{f(x)} = e^{M[f(x')-f(x)]}$$

We notice that this ratio grows significantly *only* when x is in the neighbourhood of x'.
Thus we can estimate the *mode*, x', by finding the points of greatest growth.

Remark 5.1 - *Limitations of Laplace Approximation*
No good for distributions with multiple global maxima or are assymetric.

5.3 Sampling Techniques

Definition 5.5 - *Sampling Techniques*
Sampling Techniques are *Stochastic Approximative Inference* techniques which aim to find the expectation of some function $f(\cdot)$ wrt a proability distribution $p(\cdot)$ (where one or both are intractable). $f(\cdot)$ & $p(\cdot)$ can be either continuous or discrete, (or mixed in the multivariate case).
N.B. Finding soley the expectation is useful as it is analogous to making a prediction.
N.B. Also know as *Monte Carlo Techniques.*

Remark 5.2 - *Central idea for Sampling Techniques*
The general idea begind sampling methods is to obtain a set of independent samples $\{\mathbf{x}_1,\dots,\mathbf{x}_N\}$ from $p(\mathbf{x})$ and use these to calculate a sample mean. By the *Weak Law of Large Numbers* this value will converge on the population mean as $N\to\infty$.

$$\mathbb{E}(\hat{f}) := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)$$

We can thus estimate the variance of estimator, $\hat{f}$ as

$$\text{Var}(\hat{f}) := \frac{1}{N}\mathbb{E}[(f-\mathbb{E}(f))^2]$$

N.B. Often having $N\in[10,20]$ produces sufficient accuracy.
N.B. Problems arise when samples cannot be drawn independently as this means the effective sample size is much smaller than the apparent sample size.

Remark 5.3 - *Computers are not good at sampling.*
Computers cannot understand most distributions and thus we must convert them into uniform distributions in order to samples from them.
Let $f(\cdot)$ be a probabilitiy distribution, then $F(x) := \displaystyle\int_{-\infty}^{x} f(t)dt \sim \text{Uniform}[0,1]$.
Thus we can take a sample, u, from Uniform$[0,1]$ and turn it into a sample from f as $x := F^{-1}(u)$.
N.B. This process is called *Change-of-Variables.*
N.B. If $F^{-1}(\cdot)$ is intractable then this technique is not useful and we must use alternative sampling techniques.

Definition 5.6 - *Rejection Sampling*
Rejection Sampling is a technique for generating observations from a distribution.
Let $p(\cdot)$ be the distribution which we wish to sample from.

i) Pick an approximate distribution, $q(x)$, which is tractable.

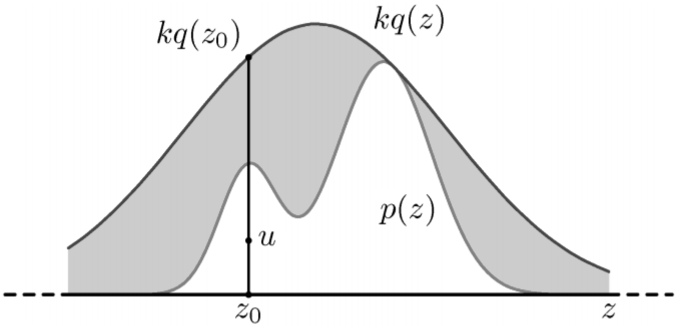
ii) Choose a $k\in\mathbb{R}^{\geq 0}$ st $k.q(x)\geq p(x)\ \forall\ x\in\mathcal{X}$.

iii) Sample a point, x', from the approximate distribution, $q(\cdot)$.

iv) Sample a point, u, from Uniform$[0,k.q(x')]$.

v) If $u > f(x')$ then reject x', otherwise sample it.

N.B. AKA *Acceptance-Rejection Method*



Remark 5.4 - *Comments on Rejection Sampling*

- If the bound between $k.q(\cdot)$ & $p(\cdot)$ is tight then the sample is efficient. Otherwise it can be wanting.
- Often teh bound between $k.q(cdot)$ & $p(\cdot)$ is not tight as we have to set k very high.
- Technique does not scale well to multiple dimensions.
- Lots of samples get rejected.

Definition 5.7 - *Importance Sampling*
Importance Sampling is a technique for approximating the *Expected Value* of a distribution.
Suppose we wish to calcualte an expectation of $f(\cdot)$ wrt probability distribution $p(\cdot)$.
Chose a tractable distribution $q(\cdot)$ st $q(x) = 0 \implies p(x) = 0$. Then we make the following approximation

$$\mathbb{E}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)\frac{p(x_i)}{q(x_i)}$$

where $\{x_1, \dots, x_N\}$ is a sample from $q(x)$.

Proof 5.1 - *Definition 5.7*

$$\mathbb{E}[f(x)] := \int_{\mathcal{X}} f(x)p(x)dx = \int_{\mathcal{X}} f(x)\frac{q(x)}{q(x)}p(x)dx = \int_{\mathcal{X}} f(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_{q(\cdot)}\left[f(x)\frac{p(x)}{q(x)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)\frac{p(x_i)}{q(x_i)}$$

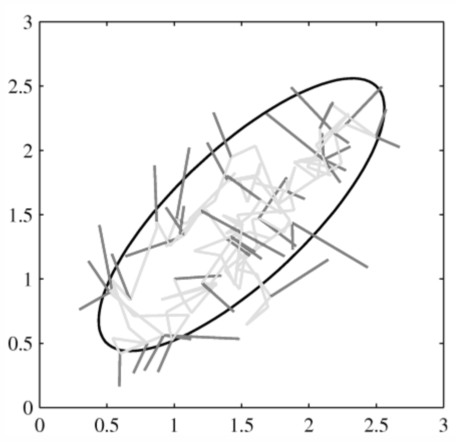
Remark 5.5 - *Comments on Importance Sampling*

- Accepts all samples.
- $\frac{p(\cdot)}{q(\cdot)}$ is called the *Sampling Ratio* and acts to correct the bias caused by sample from the wrong distribution.
- We cannot have $q(x_i) = 0$ for any $i \in [1, N]$.
- We may not always be to evaluate $p(\cdot)$.

Definition 5.8 - *Markov Chain Monte Carlo*
Markov Chain Monte Carlo, MCMC, methods are techniques for generating observations from a distribution and they work well with multi-dimensional continous random variables.
Let $p(\cdot)$ be the distribution we wish to sample from, $\mathbf{x}_t$ be a state at time t and $q(\cdot)$ be a proposed distribution which we can sample from.

```
i) Initalise a state, $\mathbf{x}_0$. Set $t=0$.
ii) While time limit has not been reached
    1) Sample a point, $\mathbf{x}'$, from the proposed distribution, $q(\cdot)$.
    2) Compute an acceptance probability
           $A(\mathbf{x}',\mathbf{x}_t):=\min\left\{1,\frac{p(\mathbf{x}')}{p(\mathbf{x}_t)}\right\}$
    3) Sample a point, $u$, from Uniform$[0,1]$.
        i. If $A(\mathbf{x}',\mathbf{x}_0)>u$ then set $\mathbf{x}_{t+1}=\mathbf{x}'$.
        ii. Otherwise, set $\mathbf{x}_{t+1}=\mathbf{x}_t$.
    4) Increment $t$
```

This can be visualised as a chain which works its way through the domain space, but backtracks every time it does not make a sample.



Remark 5.6 - *Commnets on Markov Chain Monte Carlo*

- Has the *Markov Property.* (*i.e.* Remembers its state from the last sample & uses it).
- Good at exploring.

Definition 5.9 - *Gibbs Sampling*
Gibbs Sampling is an application of the *Markov Chain Monte Carlo* algorithn. *Gibbs Sampling* exploits the fact that 1-Dimensional samples are often easier to generate by only every sampling from a single dimension. Let $p(\cdot)$ be the distribution we wish to sample from, $\mathbf{x}_t$ be a state at time t and $q(\cdot)$ be a proposed distribution which we can sample from.

```
i) Initalise a state $\mathbf{x}_0$. Set $t=0$.
ii) While time limit has not been reached.
    (a) For each dimension $i$:
        1) Isolate the dimension, $x_{t,i}$, from $\mathbf{x}_t$.
        2) Define $\mathbf{x}_{t,\neg i}:=\mathbf{x}_t/x_{t,i}$ to be $\mathbf{x}_t$ without its $i^{\text{th}}$ dimension.
        3) Calculate the posterior for this dimension, conditional on $\mathbf{x}_{t,\neg i}$.
               $p(x_{t,i}|\mathbf{x}_{t,\neg i})$
        4) Sample a point, $x_{t+1,i}$, from $p(x_{t,i}|\mathbf{x}_{t,\neg i})$.
    (b) Increment $t$
```

5.4 Deterministic Techniques

Definition 5.10 - *Deterministic Techniques for Approximative Inference*
Deterministic Techniques for *Approximative Inference* rely on making analytical approximations about the *posterior* distribution in order to make it tractable.
Often these assumptions are about the particular distribution the *posterior* takes or how it factorises.

Definition 5.11 - *Variational Bayes*
Variational Bayesian Methods are techniques used for approximating intractable distributions which arise in *Bayesian Inference.*
Let $\{X, Y\}$ be a set of training data. From *Bayes' Theorem* we have that the *evidence* is

$$p(Y)=\frac{p(Y|X)p(X)}{p(X|Y)}$$

Both $p(Y|X)$ and $p(X)$ are known, thus only $p(X|Y)$ is intractable.
Thus in order to approximate the *evidence*, $p(Y)$, we need to approximate the *posterior*, $p(X|Y)$.

Theorem 5.1 - *Jensen's Inequality* The function of the expected value is a lower bound to the expected value of the function.

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

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

Definition 5.12 - *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.

Remark 5.7 - *Features of Kullback-Leibler Divergence*

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

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

Proposition 5.3 - *Variational Bayes*
Here we shall reformulate the inference problem as an *optimisation* problem.
Let $p(\mathbf{Y})$ be an intractable distribution we are trying to approximate & choose $q(\mathbf{X})$ to be a tractable distribution.
The idea is to choose $q(\mathbf{X})$ st it is approximately equal to $p(\mathbf{X})$.

$$\begin{array}{rcl}\ln p(\mathbf{Y}) & = & \ln\left(\displaystyle\int p(\mathbf{Y},\mathbf{X})d\mathbf{X}\right) \text{ by definition of evidence} \\ & \vdots & \\ & \geq & \displaystyle\int q(\mathbf{X})\ln\left(\frac{p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y})}{q(\mathbf{X})}\right)d\mathbf{X} \text{ by Jensen's Inequality} \\ & \vdots & \\ & = & -KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y}))+\ln p(\mathbf{Y})\end{array}$$

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).

$$\begin{array}{rcl} KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y})) & := & \displaystyle\int q(\mathbf{X})\ln\left(\frac{q(\mathbf{X})}{p(\mathbf{X}|\mathbf{Y})}\right)d\mathbf{X} \\ & \vdots & \\ & = & H(q(\mathbf{X}))-\mathbb{E}_{q(\mathbf{x})}[\ln p(\mathbf{X},\mathbf{Y})]+\ln p(\mathbf{Y}) \\ \implies \ln p(\mathbf{Y}) & = & KL(q(\mathbf{X})||p(\mathbf{X},\mathbf{Y}))+\mathbb{E}_{q(\mathbf{X})}[\ln p(\mathbf{X},\mathbf{Y})-H(q(\mathbf{X})) \\ & \geq & \mathbb{E}_{q(\mathbf{X})}[\ln p(\mathbf{X},\mathbf{Y})]-H(q(\mathbf{X}))\text{ since }KL(\cdot||\cdot)\geq 0 \\ \text{Define } L(q(\mathbf{X})) & := & \mathbb{E}_{q(\mathbf{X})}[\ln p(\mathbf{X},\mathbf{Y})]-H(q(\mathbf{X}))\text{ called Evidence Lower BOund (ELBO)} \end{array}$$

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.

Proposition 5.4 - *Using result of Variational Bayes*
Remembering that $q(\mathbf{X})$ is an approximation of the true posterior $p(\mathbf{X}|\mathbf{Y})$.
Using the likelihood function, $L(\cdot)$, derived in **Proposition 5.3** we want to find $q(\mathbf{X})$ which maximises its value since all other terms are known.
Note that if we can not formulate the joint distribution, $p(\mathbf{X},\mathbf{Y})$, we are no better off using this derivation.

5.4.1 Mean Field Approximation

Remark 5.8 - *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* is a specific family of approximations, $q(\mathbf{X})$, where we assume 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 5.5 - *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

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

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

$$\begin{aligned} L(q) &= \int \prod_i q_i(X_i) \left(\ln p(\mathbf{Y}, \mathbf{X}) - \sum_k \ln q_k(X_k) \right) d\mathbf{X} \\ &\vdots \\ &= -KL(q_j(X_j)||f_j(X_j)) + c \end{aligned}$$

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 5.6 - *Mean Field Approximation - Ising Model*
THIS IS A VERY LONG DERIVATION WHICH IS IN FULL NOTES.
LOOK AT LATER.

6 Reinforcement Learning

Definition 6.1 - *Reinforcement Learning*
Reinforcement Learning aims to learn a solution to a task without specifying the task explicitly.
Instead we define a system which rewards the computer for doing good actions.
We wish to learn a *policy*, π, which decides what action will produce the greatest reward in any given state.
We note that the model is not stationary as it should evolve over time, as it recieves rewards (in effect learning). This means we have to collect new data each iteration.

Proposition 6.1 - *Reinforcement Learning Process*

1) Until time limit is reached
    i) Agent takes the action it believes will give the greatest reward.
    ii) The environment changes accordingly.
    iii) The agent recieves some reward (dependent upon how good the action was).
2) Backpropogate results (learning).

Proposition 6.2 - *Assessing Optimal Behaviour*
Depending on the task we may wish to adjust how we measure *Optimal Behaviour* wrt time

i) *Finite Time Horizon* - Sum of scores up to a specific time point.

$$\mathbb{E}\left(\sum_{t=0}^{T} r_t\right)$$

ii) *Infnite Time Horizon* - Sum of scores over all time, with decaying relevance.

$$\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \text{ for } \gamma \in [0,1]$$

iii) *Average Reward over Infinite Time* - Average score per time period.

$$\lim_{T\to\infty}\mathbb{E}\left(\frac{1}{T}\sum_{t=0}^{H}r_t\right)$$

0 Reference

Remark 0.1 - *Useful Links*
https://katbailey.github.io/post/gaussian-processes-for-dummies/
https://www.quora.com/What-is-the-difference-between-backpropagation-and-gradient-descent-when-training-a-deep-learning-neural-network-Which-of-the-two-is-Tensorflow-using
https://www.youtube.com/watch?v=0NMC2NfJGqo

Remark 0.2 - *Interpretting a data set*
Let $\mathbf{X}$ be a matrix which represents a set of data.
Rows of $\mathbf{X}$ will represent a single observation and columns will reprensent all observations of a given variable.

Definition 0.1 - *Basis Functions*
Basis Functions are sets of functions from which all functions in their space can be decomposed into a linear combination of them. *Basis Functions* cannot themselves be decomposed any further.

Polynomial Basis	$\{f_i(x) = x^i : i \in \mathbb{N}_0\}$
Fourier Bais	$\{f(x) = \sin(x), f(x) = \cos(x)\}$

Theorem 0.1 - *Schur Complement*
Let $M \in \mathbb{R}^{p \times q}$ be a general matrix with block decomposition $\begin{pmatrix} E & F \\ G & H \end{pmatrix}$.
The *Schur Complement* is a tool used for inverting matrices as it isolates the problem to only needing to know the inverse of block H.

$$M^{-1} = \begin{pmatrix} (E - FH^{-1}G)^{-1} & -(E - FH^{-1}G)FH^{-1} \\ -H^{-1}G(E - FH^{-1}G)^{-1} & H^{-1} + H^{-1}G(E - FH^{-1}G)FH^{-1} \end{pmatrix}$$

Here $E - FH^{-1}G$ is called the *Schur Complement* of M wrt H.
N.B. This is denoted as M/H.

Definition 0.2 - *Exlpaining Away*
Explaining Away is the process of breaking down a feature into multiple features in such a way that you isolate a particular variable of that feature. This is useful as we may not know much about the original feature, but a lot about the sub-features.

Definition 0.3 - *Hierarchical Knowledge*
Hierarchical Knowledge is an application of *Explaining Away* where we apply it to the sub-features in order to produce a series of linear sub-...-sub features until we reach features which we have sufficient knowledge about.

Definition 0.4 - *Kernel*
The *Kernel* of a function is all the elements in teh input that maps to a defined point in the output space, x'.

$$\text{Kern}(f) := \{x' : f(x') = x'\}$$

N.B. Often $x' := 0$.
Note that $\text{Kern}(f_1) \subseteq \text{Kern}(f_2 \circ f_1) \subseteq \cdots \subseteq \text{Kern}(f_N \circ \cdots \circ f_1)$.

Definition 0.5 - *Image*
The *Image* of a function is the set of all possible output values a function can produce

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

where $f : X \to Y$.
Note that $\text{Im}(f_N \circ \cdots \circ f_1) \subseteq \cdots \subseteq \text{Im}(f_N \circ f_{N-1}) \subseteq \text{Im}(f_N)$.

Definition 0.6 - *Sigmoid Function*

$$\sigma(z) := \frac{1}{1+e^{-z}}$$

Definition 0.7 - *Ising Model*
The *Ising Model* is a simple example of an undirected graphical model (Markov Random Fields).
The *Ising Model* has a prior of the form

$$p(\mathbf{x}) = \frac{1}{Z_0}e^{E_0(\mathbf{x})}$$

where Z_0 is a normalising factor & $E_0(\cdot)$ is a function that is learge if $\mathbf{x}$ is somethign we believe is likely, otherwise it is small.
In scenarios where $x_i \in \{-1, 1\}\ \forall\ i$ a good definition for $E_0(\cdot)$ is

$$E_0(\mathbf{x}) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} w_{ij}x_ix_j$$

where $\mathcal{N}_i$ is the neighbourhood of x_i and w_{ij} is the weight that i gives the values of x_j, as it is only positive when $x_i = x_j$.

Definition 0.8 - *Gradient Descent*
Gradient Descent is an iterative *optimisaiton algorithm* which aims to find the local minimum of a function, and thus the parameters which produce it.
Gradient Descent can be used in scenarios where a function is intractable, which *Least Squares* cannot.

i) Set $t = 0$.

ii) Randomly initalise parameter values, θ_t.

iii) Repeat until convergence of θ_t.

   (a) Evaluate the performance of this state using a *Loss Function*, $L(\mathbf{x}, \theta_t)$.
   (b) Find the derivative of the *Loss Function* wrt the parameters, $\frac{d}{d\theta}L(\mathbf{x}, \theta)$.
   (c) Evaluate the derivative for the parmater values we just tested, z.
   (d) Calculate *Step Size* $s := z \times \alpha$, where α is the *Learning Rate* (≈ 0.1).
   (e) Set $\theta_{t+1} = \theta_t - s$.

N.B. Sum of the Squared Residuals is a good loss function, $L(\mathbf{x}, \theta) := \sum_i [o_i - e_i]^2$ where o_i is the observed value & e_i is the predict value at a given point.

Definition 0.9 - *Eigenvalues & Eigenvectors*
Eigenvalues & *Eigenvectors* are properties of square matrices. For a square matrix $A \in M(\mathbb{R})_n$, $\lambda \in \mathbb{R}$ & $\boldsymbol{v} \in \mathbb{R}^n$ is an *Eigenvalue* & *Eigenvector* pair of A if

$$A\boldsymbol{v} = \lambda\boldsymbol{v}$$

N.B. Solving $|A - \lambda I| = 0$ produces in the eigenvalues of A, from which the eigenvectors can be found.

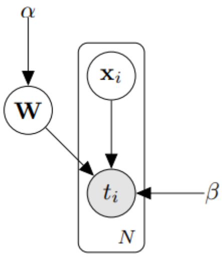
0.1 Graphical Models

Definition 0.10 - *Graphical Models*
Graphical Models are graphs which shows the minimal factorisation for the joint distribution of all variables in a model. *Graphical Models* are made up of a few core elements

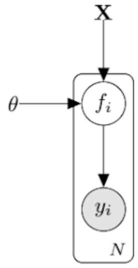
- *Nodes* - Variables (Random & realisations).
  N.B. We typically draw circles around random variables, with observed data being shaded, but not around constants.
- *Edges* - A stochastic relationship between varaibles.
- *Plates* (*i.e.* groups of Nodes) - A product(*i.e.* repeated variables).
  i.e. Suppose we have independent random variables $\{X_1, \dots, X_N\}$ which all have the same dependencies, we can simplfy the graphical model by simply defining a generic random variable X_i with the dependencies & drawing a plate around X_i which denotes the number of occurences.
  This can easily be extended for multiple variables with dependence.

The *Edges* can be *directed*, known as *Bayesian Networks*, or *un*directed, known as *Markoc Random Fields.*

Example 0.1 - *Plate Notation*
Below are examples of *Graphical Models*, using Plate Notation, for common machine learning model



Linear Regression



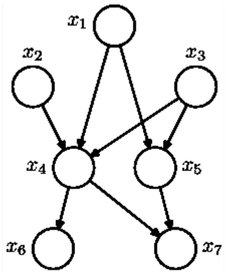
Gaussian Process Regression

Remark 0.3 - *Interpretting Factorisation from Bayesian Network*
Let A be a random variable & $X_1, \dots, X_N$ be the parents of A.
Then we have the following factorisation of the joint distribution

$$p(A, X_1, \dots, X_N) = p(A|X_1, \dots, X_N)p(X_1)\dots p(X_N)$$

Note that A may have other ancestors but we only consider its direct parents as all other dependencies will be encoded in the distribution of A's parents.

Example 0.2 - *Interpretting Factorisation from Bayesian Network*



$$p(x_1, \dots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$