American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-7, Issue-9, pp-214-229
www.ajer.org
Research Paper Open Access

# Primer on Computational Techniques for Machine Learning

Udaychandra A Nayak*, Joel Braganza, S Mahalaxmi
*Department of Information Technology, Don Bosco Institute of Technology, Kurla(W), Mumbai-400070, India*
*Corresponding Author: Udaychandra A Nayak*

***ABSTRACT :****A brief review of the mathematical and statistical techniques involved in Machine Learning isgiven. The algebraic techniques include non-square matrices, determination of theirsingularvalues and singularvectors and quadraticformsuseful in representation of data matrices. The standard techniques in optimization like steepestdescent, Newton'smethod and ConjugateGradientalgorithms for unconstrained and constrained extremum problems are discussed. In the probabilisticapproach, Bayes, Naive Bayes and Bayes Belief Networks techniques for classification arepresented. The randomprocesses like Markov models, Hidden Markov Models, for observed and hidden variables are givenwith a brief mention of EM algorithms.*
***KEYWORDS:*** *Matrices, Singular Values, Singular Vectors, Bayes Techniques, Random Processes, Machine Learning.*

---

Date of Submission: 03-08-2018 Date of acceptance: 17-09-2018

---

**CONTENTS:**



## I. INTRODUCTION

Machine Learning (ML) is a set of tools that, broadly speaking, allows us to "teach" computers how to perform tasks by providing examples of how they should be done. For any task, writing rules to accurately distinguish genuine from non-genuine (spam emails) can be very difficult to do with. A machine learning algorithm is an algorithm that can learn from data. Mitchell (1999) provides the definition of what do we mean

by learning: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T as measured by P improves with experience E."
ML tasks are usually described in terms of how the ML system should process an example. We typically represent examples or data as a vector $\vec{x} \in R^n$ where the components $x_i$ of data vector $\vec{x}$ are attributes or features. In nearly all real-world situations our data and knowledge about the world is incomplete, indirect and noisy; hence, uncertainty must be a fundamental part of a decision-making process. In this context we have to use Bayesian probability theory which is distinguished by defining probabilities as degrees of belief in contrast to frequentist statistics, where the probability of an event is defined as the frequency in the limit of infinite number of repeated trials.

A well-defined learning problem requires a well-specified task, performance metric, and source of training experience. Designing an ML approach involves a number of design choices, including choosing a type of training experience, the target function to be learnt, a representation for this target function and an algorithm for learning the target function examples.
Learning involves search: Searching through a space of possible hypotheses to find the hypothesis that best fits the available training examples and other prior constraints or knowledge. The different hypotheses spaces which can be searched include:
i) Spaces containing numerical functions
ii) Neural Networks
iii) Decision Trees
iv) Symbolic Rules,

And using theoretical results that characterize conditions under which these search methods converge towards an optimal hypothesis.

Although it is still not possible to make competitive learning nearly as well as humans learn, many algorithms have been invented that are effective for certain types of learning tasks such as speech recognition and data mining. In data mining, ML algorithms are being used routinely to discover valuable knowledge from large commercial databases containing equipment maintenance records, loan applications, financial transactions, medical records, and the like.

## II. REVIEW OF LINEAR ALGEBRA

### 2.1 Vector Spaces & Matrices

Definition: Vector Space: A set V is said to be a vector space over a field F if V is an Abelian group under addition "+", and for each $a \in F$ and $v \in V$, $\exists$ an element $av \in V$, $\ni$, the following conditions hold:
$\forall a, b \in F$ and $\forall u, v \in V$
(i) $a(v + u) = av + au$
(ii) $(a + b)v = av + bv$
(iii) $a(bv) = (ab)v$
(iv) $1v = v$
Subspace: Let V be a vector space over a field F and let U be a subset of V. If U is also a vector space over F under the operations of V, we say that U is a subspace of V.
Linear Independence $(l.i)$
A set of vectors is said to be linearly dependent $(l.d)$ over the field F, if there are vectors $v_1, v_2, \dots, v_n$ from V and elements $a_1, a_2, \dots, a_n$ from F, such that

$$\sum_{i=1}^{n} a_i v_i = 0$$

A set of vectors that is not l.d. over F is called linearly independent $(l.i)$ over F.

Basis
Let V be a vector space over F. A subset $B \subset V$ is called a basis for V if B is $(l.i)$ over F and every element $v \in V$ is a unique linear combination $(l.c)$ of elements of B, i.e., $v = \sum a_i v_i$. All bases of V contain the same number of vectors. This number is called the dimension of V, $(\dim V)$.

Example 1) Let R be the field &$R^n$ the set of column n-vectors with real components. We call $R^n$ an n-dimensional real vector space, and we can write $\vec{v}$ in component form,

$$\vec{v} = \begin{bmatrix} a_1 \\ . \\ a_n \end{bmatrix}$$

Example 2) In $R^3$, the vectors$v_1 = (1, 0, 0), v_2 = (1, 0, 1)$&$v_3 = (1, 1, 1)$ are $(l.i)$over R
To verify this, assume $\sum a_i v_i = 0 \Longrightarrow a_1 v_1 + a_2 v_2 + a_3 v_3 = 0$

Or $a_1(1,0,0) + a_2(1,0,1) + a_3(1,1,1) = (0,0,0)$
$\Longrightarrow a_1 + a_2 + a_3 = 0$
$a_3 = 0$
$a_2 + a_3 = 0$
$\Longrightarrow a_1 = a_2 = a_3 = 0$
$\Longrightarrow v_1, v_2, v_3$ are linearly independent.

**2.2Inner Product $< , >$**

Consider a vector space $R^n$ over field R. We define the inner product <,>: $R^n \times R^n \rightarrow R$ by

$$< \vec{x}, \vec{y} > = \sum_{i=1}^{n} x_i y_i = \vec{x}^T \vec{y}$$

having the following properties:
(i) $< \vec{x}, \vec{x} > \geq 0, \ < \vec{x}, \vec{x} > = 0$ iff$\vec{x} = 0$
(ii) $< \vec{x}, \vec{y} > = < \vec{y}, \vec{x} >$
(iii) $< \vec{x} + \vec{y}, \vec{z} > = < \vec{x}, \vec{z} > \ + < \vec{y}, \vec{z} >$
(iv) $< r\vec{x}, \vec{y} > = r < \vec{x}, \vec{y} >$
The vectors $\vec{x}$and$\vec{y}$ are orthogonal if $< \vec{x}, \vec{y} > = 0$
The Euclidean norm of a vector $\vec{x}$ is $\|\vec{x}\| = \sqrt{< \vec{x}, \vec{x} >} = \sqrt{\vec{x}^T \vec{x}}$
Orthogonality in $R^3$
We can write $< \vec{a}, \vec{b} > = \ \vec{a}.\vec{b} = |\vec{a}||\vec{b}|\cos\theta$ . $\vec{a}$&$\vec{b}$ are orthogonal if $\vec{a}.\vec{b} = 0 \ \Longrightarrow \theta = 90$
Using the process known as Gram-Schmidt orthogonalization, we can show that every finite dimensional Euclidean space has an orthonormal basis.

**2.3 Eigenvalues & Eigenvectors of a Matrix**

A matrix is a rectangular array of numbers A. An mxn matrix can be written as

$$A = \begin{bmatrix} a_{11} & a_{12} & . & a_{1n} \\ a_{21} & a_{22} & . & a_{2n} \\ . & . & . & . \\ a_{m1} & a_{m2} & . & a_{mn} \end{bmatrix}$$

We can denote A by $A = [\vec{a}_1 \ldots \vec{a}_n]$ where $\vec{a}_i = \begin{bmatrix} a_{1i} \\ a_{2i} \\ . \\ a_{mi} \end{bmatrix}$

The maximal number of l.i columns of A is called the rank of the matrix A, denoted by rank(A). Note that rank(A) is the dimension of the span $[\vec{a}_1 \ldots \vec{a}_n]$
A $p^{th}$ order minor of an mxn matrix A, with $p \leq \min(m, n)$ is the determinant of a pxp matrix obtained from A by deleting m – p rows and n – p columns. We have the following theorem:
Theorem: If an mxn$(m \geq n)$ matrix A has a nonzero $n^{th}$ order minor then the columns of A are l.ithat is, rankA = n.
Lemma: Let $A \in R^{m \times n}$, m > n, then, rankA = n iff rank$A^T A$ = n (i.e., the square matrix $A^T A$ is non-regular).
Eigenvalues & Eigenvectors (Square matrices only)
Let A be a nxn square matrix. $A \in R^{n \times n}$.
The equation $A\vec{v} = \lambda\vec{v}$ where $\lambda$ is a scalar (possibly complex) and $\vec{v}$ a non-zero vector is called an eigenvalue problem with $\vec{v}$ as eigenvector and $\lambda$ as eigenvalue.
$\Longrightarrow [A - \lambda I]\ \vec{v} = 0$
Or $A - \lambda I$ is a singular matrix &$\det(A - \lambda I) = 0$. Thus $\lambda_i$ are solutions of the characteristic polynomial equation

$$\det(A - \lambda I) = 0$$

If the characteristic equation has n distinct roots $\lambda_1 \ldots \lambda_n$, then$\exists$n l.i set of vectors $\vec{v}_1 \ldots \vec{v}_n$ such that $A\vec{v_i} = \lambda_i \vec{v_i}$i=1…n
W.r.t the basis B formed by the l.i set of eigenvectors $\{\vec{v_i}\}$, the matrix A can be written as a diagonal matrix $D = \begin{bmatrix} \lambda_1 & . & 0 \\ . & . & . \\ 0 & . & \lambda_n \end{bmatrix}$
Writing $M = [\vec{v}_1 \ldots \vec{v}_n]$ where each $\vec{v_i}$is an eigenvector of A, we can transform A as $M^{-1}AM = D$ or $A = MDM^{-1}$
Consider symmetric matrices: We have the following theorems:
(i) All $\lambda_i$ of a symmetric matrix A are real i.e., $\lambda = \bar{\lambda}$
(ii) Any real symmetric matrix A has a set of n eigenvectors $\{\vec{v_i}; i = 1,2, \ldots n\}$ that are mutually orthogonal.

**2.4 Singular Values and Singular Vectors**

Let $A_{n \times m}$ be of rank r. Then, $\exists U_{n \times r}, V_{m \times r}$and$\Sigma_{r \times r} \ni$

$U^H U = I = V^H V$ and $A = U\Sigma V^H$

In $\Sigma = \text{diag}(\sigma_1 \sigma_2 \ldots \sigma_n)$, the r diagonal elements of $\Sigma$ are strictly positive and are called singular values of the matrix A. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$

The Singular Value Decomposition (SVD) of matrix A may be expressed as

$$A = \sum_{i=1}^{r} \sigma_i \overrightarrow{u_i} \overrightarrow{v_i}^H$$

(i) $A = U\Sigma V^H \Longrightarrow AV = U\Sigma \Longrightarrow A\overrightarrow{v_i} = \sigma_i \overrightarrow{u_i}$ i=1,2,…, r

$A^H = V\Sigma U^H \Longrightarrow A^H U = V\Sigma \Longrightarrow A^H \overrightarrow{u_j} = \sigma_j \overrightarrow{v_j}$ j=1,2,…, r

Or $A^H A\overrightarrow{v_i} = \sigma_i^2 \overrightarrow{v_i}$ i=1,2,…,r

Thus the r nonzero eigenvalues of $A^H A$ are the squares of the singular values of A

(ii) $A = U\Sigma V^H \Longrightarrow AA^H \overrightarrow{u_i} = \sigma_i^2 \overrightarrow{u_i}$ i=1,2,…,r

Hence$AA^H$ and $A^H A$ have the same set of unique eigenvalues.

Theorem: A rectangular matrix $A_{mn}$ can be decomposed using the following method as

$$A_{mn} = U_{mm} D_{mn} V_{nn}^T$$

where $U_{mn}$ and $V_{nn}$ are orthogonal matrices ($UU^T = 1 = VV^T$) with columns of U being orthonormal eigenvectors of $AA^T$ and columns of V being orthonormal eigenvectors of $A^T A$.

$D_{mn}$ is a diagonal matrix whose elements are $\sqrt{\lambda_i}$ , $\lambda_i$ are eigenvalues of U or V in descending order.

Using SVD to simplify data

We use SVD to represent our original data set with a much smaller dataset (we are removing noise and redundant information), and extract knowledge from data.

Matrix factorization: There are many techniques for decomposing matrices (Cf.-factoring in algebra). These various factorization techniques have different properties that are more suited for one application or another.

$M_{mn}(\text{Data}_{mn}) =$ (i) $U_{m\times m}\Sigma_{m\times n}V_{n\times n}^T$ ; (ii) $U_{m\times 3}\Sigma_{3\times 3}V_{3\times n}^T$; etc.

In the 2nd decomposition $\Sigma$ is a diagonal matrix. Its diagonal elements are called singular values. The singular values $\sigma_i$ are $\sqrt{\lambda_i}$ of M.

$\Sigma$ has only diagonal elements sorted from largest to smallest. After a certain number of singular values (call this r) of M, the other values will drop to 0. This implies the data set has only r important features, and the rest of the features are noise or repeats.

SVD (Application)

Example 3: Analyzing a document data. Consider 3204 newspaper articles from 6 different sections: entertainment, financial, foreign, metro, national and sports. The data was processed using standard techniques to remove common words, to adjust for the different frequencies with which terms appear, and to adjust for the different lengths of documents.

The data matrix A is a document-term matrix, where each row represents a document and each column a term (word): $a_{ij}$ is the $j^{th}$ term in the $i^{th}$ document.

An SVD analysis of A was performed to find the first 100 ($\sigma_i$) singular values ($\sigma_1, \ldots, \sigma_{100}$) and singular vectors. (It is too expensive to find a full SVD or PCA decomposition and often pointless since relatively few of the $\sigma_i, \overrightarrow{u_c}, \overrightarrow{v_i}$ are required to capture the structure of the matrix A).

Observations:

(i) The largest singular value $\sigma_1$ is associated with common terms that are frequent, but not eliminated by the pre-processing.

(ii) Associated with the second right singular vector $\overrightarrow{v_2}$ are the following top 10 terms (words) (all associated with sports)

"game, score, lead, team, play, rebound, season, coach, league, goal"

(iii) Associated with the third right singular vector $\overrightarrow{v_3}$ are the following top 10 terms (words)

"earn, million, quarter, bank, rose, billion, stack, company, corporation, revenue" (all financial terms)

Using $\overrightarrow{v_2}$ and $\overrightarrow{v_3}$ , we reduced the dimensionality of the data (D' is the new data matrix with two attributes)

$$D' = D \,.\, [\overrightarrow{v_2}, \overrightarrow{v_3}]$$

In other words, all documents were expressed in terms of two attributes: sports & finance.

**2.5 Quadratic forms**

A quadratic form f: $R^n \rightarrow R$ is a function

$$f(\vec{x}) = \vec{x}^T Q \vec{x}$$

where Q is a n x n real matrix. Without loss of generality, we can assume Q to be symmetric i.e., $Q = Q^T$. (Otherwise we can obtain a symmetric matrix Q' from Q)

A quadratic form $\vec{x}^T Q\vec{x}$ is said to be positive definite if $\vec{x}^T Q\vec{x} > 0$ , $\forall \vec{x} \neq 0$ i.e., $f(\vec{x}) > 0$

and positive semi-definite if $\vec{x}^T Q\vec{x} \geq 0, \forall \vec{x} \neq 0$ i.e.,
$f(\vec{x}) \geq 0$
Similarly,
A quadratic form $\vec{x}^T Q\vec{x}$ is said to be negative definite if $\vec{x}^T Q\vec{x} < 0, \forall \vec{x} \neq 0$ i.e., $f(\vec{x}) < 0$
and negative semi-definite if $\vec{x}^T Q\vec{x} \leq 0, \forall \vec{x} \neq 0$ i.e.,
$f(\vec{x}) \leq 0$
Range and Nullspace of A
Let $A \in R^{m\times n}$, ($R^{m\times n}$ denotes the set of m x n matrices with elements in the field of real numbers).
Let the image (or) range of A be denoted by
$R(A) = \{A\vec{x}: \vec{x} \in R^n\}$,
and the nullspace (or kernel) of A be denoted by

$$N(A) = \{\vec{x} \in R^n: A\vec{x} = \vec{0}\}$$

Note that R(A) and N(A) are subspaces.
Lemma: Let $A \in R^{m\times n} m \geq n$. Then, rank(A) = n iff rank$(A^T A) = n$. i.e., the square matrix $A^T A$ is non-singular.
Theorem: The unique vector $\vec{x}^*$ that minimizes $f(x) = \|A\vec{x} - \vec{b}\|^2$ is given by the solution to the equation$A^T A\vec{x} = A^T b$, i.e., $\vec{x}^* = (A^T A)^{-1} A^T \vec{b}$.
Thus, $\vec{x}^* = (A^T A)^{-1} A^T \vec{b}$ is the unique minimizer of $\|A\vec{x} - \vec{b}\|^2$

**2.6 Optimization Techniques: Unconstrained and Constrained optimization**
We consider the optimization problem
Minimize $f(\vec{x})$
Subject to$\vec{x} \epsilon \Omega$
where $f: R^n \to R$ is a real valued function, called the objective function or cost function. The set $\Omega \subset R^n$, is called the constraint set or feasible set. It normally takes the form:
$\Omega = \{\vec{x}: \vec{h}(\vec{x}) = \vec{0}, \vec{g}(\vec{x}) \leq \vec{0}\}$, functional constraints,
where $\vec{h}$ and $\vec{g}$ are given functions. The constraint may also include $\Omega = R^n$ which is usually called the unconstrained case.
Conditions for local minimizers
We derive conditions for a point $\vec{x}^*$ to be a local minimizer. For the given optimization problem with constraint set $\Omega \subset R^n$, the minimizer may be either in the interior or on the boundary of $\Omega$. A vector $\vec{d} \in R^n$, $\vec{d} \neq \vec{0}$ is called a feasible direction at $\vec{x} \epsilon \Omega$, if $\exists \alpha_0 > 0 \ni \vec{x} + \alpha\vec{d} \epsilon \Omega \forall \alpha [0, \alpha_0]$

First-Order Necessary Condition (FONC)
If we define $\vec{x}(\alpha) = x^* + \alpha\vec{d} \in \Omega$(Obviously $\vec{x}(0) = \vec{x}^*$)
Then, by Taylor's theorem,

$$f(\vec{x}^* + \alpha\vec{d}) - f(\vec{x}^*) = \alpha\vec{d}^T \nabla f(\vec{x}(0)) + o(\alpha)$$

Let $\Omega \subset R^n$ and $f \in C^1$ a real valued function on $\Omega$. If $x^*$ is a local minimizer of f over $\Omega$, then for any possible direction $\vec{d}$ at $\vec{x}^k$, we have

$$\vec{d}^T \nabla f(\vec{x}^n) \geq 0$$

Second-Order Necessary Condition (SONC)
Let $\Omega \subset R^n$ and $f \in C^2$, $x^*$ a local minimizer of f over $\Omega$, and $\vec{d}$ a feasible direction at $\vec{x}^*$
If $\vec{d}^T \nabla f(\vec{x}^*) = 0$, then, $\vec{d}^T \nabla^2 f(\vec{x}^*)\vec{d} \geq 0$. $H = \nabla^2 f(\vec{x})$ is the Hessian.

**2.6.1 Steepest Descent**
We consider a class of search methods for $f: R^n \to R$. These methods use the $\nabla f$. For the function f the set of points $\vec{x}$ satisfying $\vec{f}(\vec{x}) = c$ for some constant c is called a level set (see fig) (Cf. level curves on hill, equipotential curves or surfaces)
Note that $-\nabla f(\vec{x})$ is in the direction of negative gradient or the direction of maximum rate of decrease. Consider the point $\vec{x}^{(0)}$, the starting point, and another point $\vec{x}^{(0)} - \alpha\nabla f\vec{x}^{(0)}$

$$f\left(\vec{x}^{(0)} - \alpha\nabla f(\vec{x}^{(0)})\right) = f(\vec{x}^{(0)}) - \alpha\|\nabla f(\vec{x}^{(0)})\|^2 + O(\alpha^2)$$

For sufficiently small $\alpha > 0$, we can write when $\nabla f(\vec{x}^{(0)}) \neq 0$

$$f\left(\vec{x}^{(0)} - \alpha\nabla f(\vec{x}^{(0)})\right) < f(\vec{x}^{(0)})$$

Thus, if we start with a point $\vec{x}^k$, to find the next point we move by an amount $-\alpha_k \nabla f(\vec{x}^{(k)})$ i.e.,

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \alpha_k \nabla f(\vec{x}^{(k)})$$

In the method of steepest descent, the step size $\alpha_k$ is chosen to minimize $f\left(\vec{x}^{(k)} - \alpha \nabla f(\vec{x}^{(k)})\right)$ i.e.,

$$\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f\left(\vec{x}^{(k)} - \alpha \nabla f(\vec{x}^{(k)})\right)$$

We can prove that the algorithm possesses the descent property $f(\vec{x}^{(k+1)}) < f(\vec{x}^{(k)})$

Example 4:

Let's apply the steepest descent algorithm to the following function,

$$F(x) = {x_1}^2 + 25{x_2}^2$$

Starting from the initial guess

$$x_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

The first step is to find the gradient:

$$\nabla F(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} F(x) \\ \frac{\partial}{\partial x_2} F(x) \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 50x_2 \end{bmatrix}$$

If we evaluate the gradient at the initial guess we find

$$g_0 = \nabla F(x)|_{x=x_0} = \begin{bmatrix} 1 \\ 25 \end{bmatrix}$$

Assume that we use a fixed learning rate of $\alpha = 0.01$. The first iteration of the steepest descent algorithm would be

$$x_1 = x_0 - \alpha g_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.01 \begin{bmatrix} 1 \\ 25 \end{bmatrix} = \begin{bmatrix} 0.49 \\ 0.25 \end{bmatrix}$$

The second iteration of the steepest descent produces

$$x_2 = x_1 - \alpha g_1 = \begin{bmatrix} 0.49 \\ 0.25 \end{bmatrix} - 0.01 \begin{bmatrix} 0.98 \\ 12.5 \end{bmatrix} = \begin{bmatrix} 0.4802 \\ 0.125 \end{bmatrix}$$

**2.6.2Newton's Method**

We can write for the function $f: R^n \to R$ with $f \in C^2$, Taylor expansion of f about the point $\vec{x}^k$, neglecting terms of order three & higher as:

$f(\overrightarrow{x}) \approx f(\vec{x}^{(k)}) + (\vec{x} - \vec{x}^{(k)})^T \vec{g}^{(k)} + \frac{1}{2}(\vec{x} - \vec{x}^{(k)})^T H(\vec{x}^{(k)})(\vec{x} - \vec{x}^{(k)}) \triangleq q(\vec{x})$, say

In this $\vec{g}^{(k)} = \nabla f(\vec{x}^{(k)}$ and $H(\vec{x}^k) = \nabla^2 f(\vec{x}^{(k)})$

Setting $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{d}^{(k)}$, we can write the iterative steps as:

(i) Solve $H(\vec{x}^k)\vec{d}^{(k)} = -\vec{g}^{(k)}$ for $\vec{d}^{(k)}$

(ii) Set $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{d}^{(k)}$

Example5:

$$F(x) = {x_1}^2 + 25{x_2}^2$$

The gradient and Hessian matrices are

$$\nabla F(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} F(x) \\ \frac{\partial}{\partial x_2} F(x) \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 50x_2 \end{bmatrix}, \nabla^2 F(x) = \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}$$

If we start from the same initial guess

$x_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$,

The first step of Newton's method would be

$$x_1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 25 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This method will always find the minimum of a quadratic function in one step. This is because Newton's method is designed to approximate a function as quadratic and then locate the stationary point of the quadratic approximation.

**2.6.3 Q-Conjugate Algorithm**

Defn: Q Conjugate Directions: Let Q be a symmetric positive definite nxn matrix. If the directions $\vec{d}^{(1)}, \vec{d}^{(2)}, \ldots, \vec{d}^{(k)} \in R^n, k \leq n-1$ are non-zero and we say that they are Q-conjugate if $\vec{d}^{(i)T} Q \vec{d}^{(j)} = 0 \ \forall i \neq j$.

Theorem: The Q-Conjugate directions are l.i.

Algorithm: We now apply the conjugate direction algorithm to minimize the quadratic function

$$f(\vec{x}) = \frac{1}{2}\vec{x}^{T}Q\vec{x} - \vec{x}^{T}\vec{b}$$

(i) Given Q, and a starting point $\vec{x}^{(0)}$, we first find Q-conjugate directions $\vec{d}^{(0)}, \vec{d}^{(1)}, \dots, \vec{d}^{(n)}$.

(ii) Obtain $\vec{g}^{(k)} = \nabla f\vec{x}^{(k)} = Q\vec{x}^{(k)} - \vec{b}$, and $\alpha_k = -\frac{\vec{g}^{(k)}\vec{d}^{(k)}}{\vec{d}^{(k)T}Q\vec{d}^{(k)}}$ where $\alpha_k$ =learning rate

(iii) $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \alpha_k\vec{d}^{(k)}$

**2.6.4 Extremum problems with constraints**: (Equality constraints)

Let $h: R^2 \to R$ be the constraint function. We know that at each point x in the domain, the $\nabla h(x)$ is orthogonal to the level set that passes through that point. Let us choose a point $\vec{x}^* = (x_1^*, x_2^*)^T \ni h(\vec{x}^*) = 0$ and assume that $\nabla h(\vec{x}^*) \neq 0$. The level set through the point $\vec{x}^*$ is the set $\{\vec{x}: h(\vec{x}) = 0\}$. We then parametrize this level set in anneighborhood of $\vec{x}^*$ by a curve x(t), that is a continuously differentiable vector function $f: R \to R^2$ such that

$$\vec{x}\ (t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} t \in (a,b)\vec{x}^* = \vec{x}(t^*), x^*(t^*) \neq 0 t^* \in (a,b)$$

We now show that $\nabla h(x^*)$ is orthogonal to $x^*(t^*)$

Since h is constant on the curve $[x(t): t \in (a,b)]$ we have $\forall t \in (a,b)$

$$h\big(x(t)\big) = 0 \implies \frac{d}{dt}h\big(x(t)\big) = 0 \quad \forall t \in (a,b)$$

Therefore, $\frac{d}{dt}h\big(x(t)\big) = \nabla h[\vec{x}(t)^T.\dot{x}(t)] = 0 \implies \nabla h(x^*) \perp \dot{x}(t)$

Now suppose that $\vec{x}^*$ is a minimizer of $f: R \to R^2$ on the set $\{x: h(x) = 0\}$

We claim that $\nabla f(x^*) \perp \dot{x}(t^*)$

Consider the composite function of t given by

$$\varphi(t) = f\big(x(t)\big)$$

It achieves a minimal $t^* \implies$ the FONC for the unconstrained extremum problem

$$\implies \frac{d\varphi(t^*)}{dt} = 0$$

$$\implies 0 = \frac{d}{dt}\varphi(t^*) = \nabla[f\big(x(t^*)\big)]^T\dot{x}(t^*)$$

The fact that $\dot{x}(t^*)$ is tangent to the curve x(t) at x* means that $\nabla f(\vec{x}^*) \perp$ to the curve at $\vec{x}^*$.

Now $\nabla h(\vec{x}^*)$ is also orthogonal to $\dot{\vec{x}}(t^*)$. Therefore the vector $\nabla h(x^*)$ and $\nabla f(x^*)$ are parallel i.e., $\nabla f(\vec{x}^*)$ is a scalar multiple of $\nabla h(\vec{x}^*)$.

Lagrange's Theorem: (General case)

Let $\vec{x}^*$be a local minimizer of $f: R^n \to R$ subject to $h(\vec{x}) = 0$, $h: R^n \to R^m$, $m \leq n$. Assume that $\vec{x}^*$is a regular point. Then $\exists a\lambda^* \epsilon R^m$ such that

$$Df(\vec{x}^*) + \lambda^{*T}Dh(\vec{x}^*) = 0^T$$

Lagrange's theorem states that if $\vec{x}^*$is an extremizer then $\nabla f$ can be expressed as a l.c. of the gradients of the constraints. We refer to $\lambda^*$as a Lagrange multiplier vector and its components as Lagrange multipliers.

**2.6.5 Extremum Problems with Constraints (Equality and Inequality Constraints)**

KKT Theorem: (Karush-Kuhn-Tucker)

Let $\vec{f}, \vec{h}, \vec{g} \in C^1$. Let $\vec{x}^*$ be a regular point and a local minimizer for the problem of minimizing f subject to $\vec{h}(\vec{x}) = \vec{0}$ and $\vec{g}(\vec{x}) \leq \vec{0}$. Then there exists $\lambda^* \epsilon R^m$ and $\mu^* \epsilon R^p$ such that

(i) $\mu^* \geq 0$

(ii) $Df(\vec{x}) + \lambda^{*T}Dh(\vec{x}^*) + \mu^{*T}Dg(\vec{x}^*) = 0^T$

(iii) $\mu^{*T}g(\vec{x}^*) = 0$

In the above theorem, we refer to $\lambda^*$as a Lagrange multiplier vector and $\mu^*$as a KKT multiplier vector.

Example 6: Use the KKT conditions to solve the following NLPP (Nonlinear programming problem)

Maximize $z = 8x_1 + 10x_2 - {x_1}^2 - {x_2}^2$

Subject to: $3x_1 + 2x_2 \leq 6$

$$x_1, x_2 \geq 0$$

Here, $f(\vec{x}) = 8x_1 + 10x_2 - {x_1}^2 - {x_2}^2$

and $h(\vec{x}) = 3x_1 + 2x_2 - 6 \leq 0$

The KKT conditions are:

$$\frac{\partial f}{\partial x_i} - \lambda\frac{\partial h}{\partial x_i} = 0 \implies \begin{cases} 8 - 2x_1 - 3\lambda = 0 \\ 10 - 2x_2 - 2\lambda = 0 \end{cases}$$

$\lambda h(\vec{x}) = 0 \implies \lambda(3x_1 + 2x_2 - 6) = 0$

$h(\vec{x}) \leq 0, x_1, x_2 \geq 0, \ \lambda > 0$

$\Rightarrow \lambda(3x_1 + 2x_2 - 6) \leq 0$
Now depending upon the value of $\lambda$, the following two cases arise:
Case I) If $\lambda = 0$, then $8 - 2x_1 = 0\ and\ 10 - 2x_2 = 0$ implies $x_1 = 4, x_2 = 5$, but, this does not satisfy the condition $h(\vec{x}) = 3x_1 + 2x_2 - 6 \leq 0$
Case II) If $\lambda \neq 0, (3x_1 + 2x_2 - 6) = 0$
Eliminating $\lambda$, we get $-2x_1 + 3x_2 - 7 = 0$
Solving the above two equations we get $(x_1, x_2) = \left(\frac{4}{13}, \frac{33}{13}\right)$
Using these values, we get $\lambda = \frac{206}{33}$ implies $\lambda > 0$
The optimal solution is $(x_1, x_2) = \left(\frac{4}{13}, \frac{33}{13}\right)$ and the maximum value of Z is

$$z_{max} = z(\vec{x}_{opt}) = 21.3$$

## III. REVIEW OF PROBABILITY THEORY

### 3.1 Discrete and Continuous Univariate Random Variables

Suppose we flip a coin 10 times (or flip 10 coins once) and let the outcome be
Ω = {H, H, T, H, T, H, H, T, T, T}
In this, number of heads in 10 tosses = 5
An r.v. (random variable) X is a function $X: \Omega \rightarrow R$ (where $\Omega$= Sample space, R = Real number)
An r.v. can be discrete or continuous.
Discrete r.v. X:

a) X, Bernoulli r.v. $p(x) = \begin{cases} p \\ 1-p \end{cases}$ where p = probability of heads in a toss of a coin.

b) X, Binomial r.v.: This is the number of heads in n independent flips of a coin with heads probability p

$$P(x) = {}_{x}^{n}C p^x (1-p)^{n-x}$$

Example 7: Find the probability of 3 heads in 5 tosses where $p = {}^1/_2$ and $1 - p = {}^1/_2$

$P(3) = {}_{3}^{5}C \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2 = \frac{10}{32} = 0.3$
Continuous r.v. X:
Uniform pdf X $\sim U(a, b)$ where a < b
Example 8: $U(X) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & otherwise \end{cases}$
Example 9: $U(X) = \begin{cases} \frac{1}{b-a} if\ a \leq x \leq b \\ 0 \quad otherwise \end{cases}$
X$\sim Normal(\mu, \sigma^2)$

$$f(x) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad -\infty < x < \infty$$

Properties: (Continuous X) (pdf = probability distribution function)
i) $f(x) \geq 0$
ii) $\int_{-\infty}^{\infty} f(x)dx = 1$
iii) $\int_{x \epsilon A} f(x)dx = P(X \epsilon A)$
Properties: (Discrete X) (pmf = probability mass function)
i) $0 \leq p_X(x) \leq 1$
ii) $\sum_{x \in Val\,(X)} p_X(x) = 1$
iii) $\sum_{x \in A} p_X(x) = P(x \in A)$
Characteristics of pmf or pdf
For X, with pdf f(x), we haveE[X]→ $Expectation$
Let $f_X(x)$ be a pdf of a continuous r.v. X, and $g: R \rightarrow R$ is an arbitrary function of X, then $E[g(x)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx$ (for continuous r.v.)
$E[g(x)] = \sum_{x \in Val\,(X)} g(x)\, p_X(x)$ (for discrete r.v.)
Variance

$$Var(X) = E[(X - E(x))^2] = E[X^2] - (E[x])^2$$

Example 10:
Calculate the mean & variance of the uniform r.v. X with pdf $f_X(x) = \begin{cases} 1 & x \in [0,1] \\ 0 & otherwise \end{cases}$
Mean E[X] = $\int_{-\infty}^{\infty} x f_X(x) dx = \int_0^1 x.1 dx = \left[\frac{x^2}{2}\right]_0^1 = \frac{1}{2}$

Variance Var[X] = $E[(X-E(x))^2] = E[X^2] - (E[x])^2 = \int_0^1 x^2\,dx - \left(\frac{1}{2}\right)^2 = \left[\frac{x^3}{3}\right]_0^1 - \frac{1}{4} = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$

**3.2 Discrete and Continuous Multivariate Random Variables**

Two Random Variables (2 r.v.'s):

Joint Distribution

1.Discrete case: (X, Y)

$$Pr(X = x, Y = y) = f(x,y) = f(x \cap y)$$

$$f(x,y) \geq 0$$

$$\sum_x \sum_y f(x,y) = 1$$

2. Continuous case: (X,Y)

$$Pr(a < X < b, c < Y < d) = \int_{x=a}^{b} \int_{y=c}^{d} f(x,y)dxdy$$

$X \sim n(X; \mu, \sigma^2)$

$$P(X \leq x) = \int_{-\infty}^{x} f(x)dx$$

Joint Cumulative Distribution Function $F_{XY}(x,y) = Pr(X \leq x, Y \leq y)$

Marginal Distributions

Discrete: $P_X(x) = \sum_y P_{XY}(x,y)$

Continuous: $f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y)dy$

Conditional Distributions

Discrete

$$Pr_{Y|X}(y|x) = \frac{P_{XY}(x,y)}{P_X(x)}$$

Let A and B be two events s.t. P(A) > 0. Denote by P(B|A), the probability of B given A has occurred. A becomes the new sample space replacing the original S, thus

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

Continuous Distributions

$$f_{Y|X}(y|x) = \frac{f_{XY}(x,y)}{f_X(x)}$$

Bayes rule:

$$f_{Y|X}(y|x)f_X(x) = f_{X|Y}(x|y)f_Y(y)$$

Expectation & Covariance

$$E[g(X,Y)] = \sum_x \sum_y g(x,y)P_{XY}(x,y) = \int_0^{\infty} \int_0^{\infty} g(x,y)f_{XY}(x,y)dxdy$$

Similar to variance, we have covariance as

Cov(X,Y) = E[(X – E(X)) (Y – E(Y))]

Cov(X,Y) = E[XY] – E[X]E[Y]

Independent identically distributed rvs (iid)

If X & Y are independent, then cov(X,Y)=0 We can write E[f(X)g(Y)] = E[f(X)] E[g(Y)]

Multiple Random Variables

$\int_{\vec{X} \in A} f_{X_1 \ldots X_n}(x_1 x_2 \ldots x_n)\,dx_1 dx_2 \ldots dx_n$ = probability of the event A in $R^n$

$\vec{X} = (x_1, x_2, \ldots, x_n)$ (Random Vector)

Example 11: (n-dimensional Gaussian pdf)

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

We write this as X$\sim N(\mu, \Sigma)$

Basic properties

1) Conditional Probabilities (Chain Rule)

$f(x_1, x_2, \ldots, x_n) = f(x_n|x_1, \ldots, x_{n-1})f(x_1, x_2, \ldots, x_{n-1})$

$$= f(x_n|x_1, \dots, x_{n-1}) f(x_{n-1}|x_1, \dots, x_{n-2}) f(x_1, x_2, \dots, x_{n-2})$$

$$\dots\dots$$

$$= f(x_1) \prod_{i=2}^{n} f(x_i|x_1 \dots x_{i-1})$$

2) Independence (Mutually Independent)

$$f(x_1, x_2, \dots, x_n) = f(x_1) f(x_2) \dots f(x_n)$$

### 3.3 Bayes' Rule

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

On the surface, Bayes' rule does not seem very useful. It allows us to compute the single term P(b|a) in terms of three terms: P(a|b), P(b) and P(a). There are many cases where we do have good probability estimates for these three numbers and need to compute the fourth. Often we perceive as evidence the effect of some unknown cause and we would like to determine that cause.

$$P(cause|effect) = \frac{P(effect|cause)P(cause)}{P(effect)}$$

For example, the doctor knows P(symptoms|disease) and wants to derive a diagnosis P(disease|symptoms).

Example 12: A doctor knows that the disease Meningitis causes the patient to have a stiff neck with probability 0.7, i.e., P(s|m) = 0.7. The doctor also knows some unconditional facts: the prior probability that a patient has meningitis is 1/50000 = P(m) and the prior probability that a patient has stiff neck is P(s) = 0.01

Hence, $P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.7 \times 1/50000}{0.01} = 0.0014$

Note that $P(s) \gg P(m)$. The causal information P(s|m) is unaffected by an epidemic of meningitis, because it simply reflects the way meningitis works.

### 3.4 Naïve Bayes

Multiple Evidences

What happens when we have 2 or more evidences? (catch & toothache)

Example 13: What can a dentist conclude if her nasty probe catches in the aching tooth of a patient? Each catch and toothache is directly caused by the cavity and but neither has a direct effect on the other; toothache depends on the state of the nerves in the tooth, whereas the probe's accuracy depends on the dentist's skill, to which the toothache is irrelevant i.e., they are conditionally independent given the cavity.

Since P(X,Y|Z)=P(X|Z).P(Y|Z) when X and Y are conditionally independent given Z, we can write $P(T \wedge Ch|Cv) = P(T|Cv).P(Ch|Cv)$

When we have more evidences, we can write using chain rule:

$$P(cause, effect_1, effect_2, \dots, effect_n) = P(cause) \prod_i P(effect_i|cause)$$

Such a probability distribution is called a Naïve Bayes model.

Example14:Consider the problem of classifying days according to whether someone will play tennis. Each day is described by the 4 attributes <outlook, temperature, humidity, wind>. We have to classify the new instance x = <sunny, cool, high, strong>. The task is to predict the target value y taking <yes or no> for this new x.

From the table we see that there are n=14 training samples, each sample with 4 attributes $\vec{x} = (x_1, x_2, x_3, x_4)$ with

$x_1$ taking values "Sunny, Overcast, Rain"

$x_2$ taking values "hot, mild, cool"

$x_3$ taking values "high, normal"

$x_4$ taking values "weak, strong"

From table, we see that $P(C_1) = \frac{9}{14}$ and $P(C_2) = \frac{5}{14}$

The conditional probabilities are estimated as follows:

For example,

$$P(x_4 = strong|y = C_1) = \frac{3}{9} P(x_4 = strong|y = C_2) = \frac{3}{5}$$

$$P(x_4 = weak|y = C_1) = \frac{6}{9} P(x_4 = weak|y = C_2) = \frac{2}{5}$$

Similarly we can work out for the other variables leading to 20 values.
Consider the test data, x = <sunny, cool, high, strong>
Since we are using NB classifier, we have $P(\vec{x}|C_j) = \prod_i P(x_i|C_j)$
Hence, for $C_1$,

$$P(C_1|\vec{x}) = P(C_1).P(Sunny|C_1).P(cool|C_1).P(high|C_1)P(strong|C_1) = 0.0053$$

Similarly for $C_2$,

$$P(C_2|\vec{x}) = P(C_2).P(Sunny|C_2).P(cool|C_2).P(high|C_2)P(strong|C_2) = 0.0206$$

Normalizing to 1, we obtain
$P(C_2|\vec{x}) = 0.795$
$P(C_1|\vec{x}) = 0.205$
Hence, the test data belongs to class $C_2$

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis? |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

**Table 1 – Weather parameter values**

Conditional Independence
X is conditionally independent of Y given Z $\Rightarrow P(X|Y,Z) = P(X|Z)$
Similarly, the set of variables $\{X_1, X_2, \dots, X_l\}$ is conditionally independent of the set of variables $\{Y_1, Y_2, \dots, Y_m\}$ given the set of variables $\{Z_1, Z_2, \dots, Z_n\}$ if

$$P(X_1 \dots X_l|Y_1Y_2 \dots Y_m; Z_1Z_2 \dots Z_n) = P(X_1 \dots X_l|Z_1Z_2 \dots Z_n)$$

**3.5 Bayesian Belief Networks (BBN)**
The conditionally independent assumption made by NB classifiers may seem too rigid especially for classification problems in which the attributes are somewhat correlated.
a) If a node X does not have any parents, then the CPT contains only the prior probabilities P(X).
b) If a node X has only a single parent Y, then the CPT contains P(X|Y)
c) If a node X has multiple parents $\{Y_1Y_2 \dots Y_k\}$ then the CPT contains the conditional probabilities $P(X|Y_1Y_2 \dots Y_k)$

Normally the BBN does not give us the full CPT. We can always construct the CPTs from the individual tables, but that takes exponential time & space.

Representation and Conditional Independence
In general, a BBN represents a joint pd by specifying a set of conditional independent assumption (represented by a Directed Acyclic Graph), together with sets of local conditional probabilities. Each variable in the joint space is represented by a node in the BN. For each variable, two types of information are specified:
1st – the network arcs represent the assertion that the variable is conditionally independent of its non-descendants in the network given to immediate predecessor in the BN. We say that X is a descendant of Y if there is a directed path form Y to X.
2nd – a CPT is given for each variable, describing the pd for that variable given the value of its immediate predecessor.
The joint probability for any desired assignment of values $< y_1, y_2, \dots, y_n >$ to the tuple of network variables $< Y_1, Y_2, \dots, Y_n >$ can be computed by the formula

$$P(y_1, y_2, \dots, y_n) = \prod_{i=1}^{n} P(y_i|Parents(Y_i))$$

Where Parents $(Y_i)$ – the set of immediate predecessors of $Y_i$ in the BN
And, $P(y_i|Parents(Y_i))$ – values stored in the CPT associated with node $Y_i$.

We see that the combination of the topology of the BN (i.e., the set of nodes and links) and the conditional distribution suffices to specify (implicitly) the full joint distribution for all the variables. There are two ways in which we can understand the semantics of the BBNs:
The 1st is to see the BN as a representation of the joint pdf.
The 2nd is to view it as an encoding of a collection of conditionally independent statements.
Example 15: The BBN shown in Figure represents the joint pd over the Boolean variables:

$$\vec{X} = \langle Storm, Lightening, Thunder, ForestFire, CampFire, BusTourGroup \rangle$$

i.e., $\vec{X} = \langle S, L, T, FF, CF, BTG \rangle$
Consider the node CF: It is conditionally independent of its non-descendants L & T, given its immediate parents S and BTG. Its conditional probability given S and BTG are given in the CPT.
These together describe the full joint pd for the BBN.
For ex: P(C|S,B)=0.4
$P(\neg C|\neg S, \neg B) = 0.8$

Example:

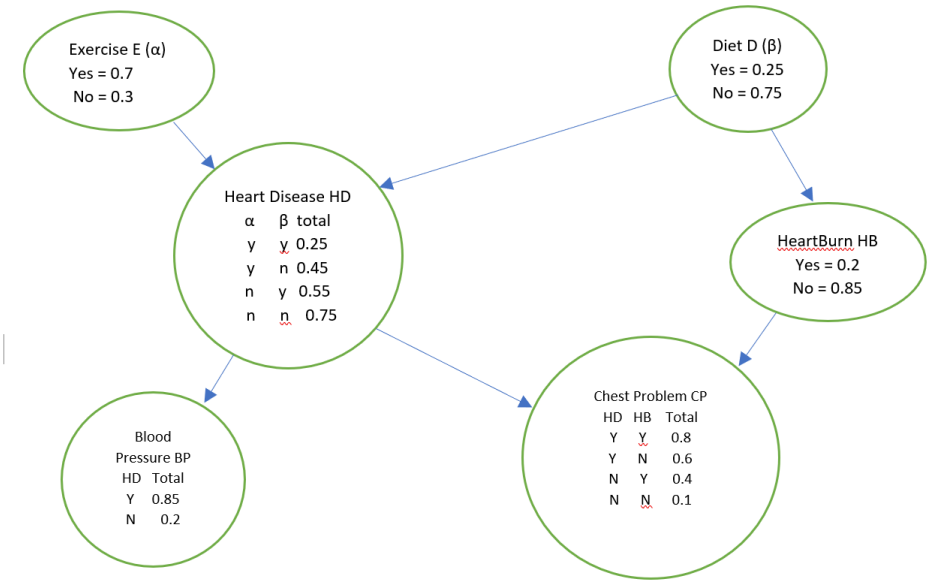


**Fig 1. Bayesian Belief Network**

Example of Inferencing Using BBN:
Suppose we are interested in using the BBN shown in above Figure to diagnose whether a person has heart disease. The following cases illustrate how the diagnosis can be made under different scenarios.

Case 1: No Prior Information
Without any prior information, we can determine whether the person is likely to have heart disease by computing the prior probabilities P(HD=Yes) and P(HD=No). To simplify the notation, let $\alpha \epsilon \{Yes, No\}$ denote the binary values of Exercise and $\beta \epsilon \{Healthy, Unhealthy\}$ denote the binary values of Diet.

P(HD=Yes) = $\sum_\alpha \sum_\beta P(HD = Yes|E = \alpha, D = \beta)P(E = \alpha, D = \beta)$

$$= \sum_{\alpha} \sum_{\beta} P(HD = Yes|E = \alpha, D = \beta)P(E = \alpha)P(D = \beta)$$

= 0.25 x 0.7 x 0.25 + 0.45 x 0.7 x 0.75 + 0.55 x 0.3 x 0.25 + 0.75 x 0.3 x 0.75
= 0.49
Since P(HD=no) = 1 – P(HD=yes)=0.51, the person has a slightly higher chance of not getting the disease.

Case 2: High Blood Pressure

If the person has high blood pressure) we can make a diagnosis about heart disease by comparing the posterior probabilities, P(HD = Yes | BP=High) against P(HD=No | BP = High). To do this, we must compute P(BP = High):
P(BP=High) = $\sum_\gamma P(BP = High|HD = \gamma)P(HD = \gamma)$

= 0.85 x 0.49 + 0.2 x 0.51 = 0.5185.
Where $\gamma \epsilon \{Yes, No\}$. Therefore, the posterior probability the person has heart disease is
P(HD = Yes | BP=High) = $\frac{P(BP=High \mid HD=Yes)\ P\ (HD=Yes)}{P(BP=High)} = \frac{0.85 \times 0.49}{0.5185} = 0.8033.$

Similarly, P(HD=no | BP=High) = 1 – 0.8033= 0.1967. Therefore, when a person has high blood pressure, it increases the risk of heart disease.

Case 3: High Blood Pressure, Healthy Diet, and Regular Exercise
Suppose we are told that the person exercises regularly and eats a healthy diet. How does the new information affect our diagnosis? With the new information, the posterior probability that the person has heart disease is
P(HD=Yes | BP=high, D=Healthy, E=Yes)

$$= \frac{P(BP=High|HD=Yes,\ D=Healthy, E=Yes)}{P(BP=High|D=Healthy, E=Yes)} \times P(HD=Yes, D=Healthy, E=Yes)$$

$$= \frac{P(BP=High|HD=Yes,\ D=Healthy, E=Yes)}{\sum_{\gamma} P(BP=High|HD=\gamma)P(HD=\gamma|D=Healthy, E=Yes)}$$

$$= \frac{0.85 \times 0.25}{0.85 \times 0.25 + 0.2 \times 0.75}$$

$$= 0.5862$$

While the probability that the person does not have heart disease is
P(HD=No|BP=High, D=Healthy, E=Yes) = 1 – 0.5862=0.4138

## IV. DISCRIMINATIVE & GENERATIVE LEARNING ALGORITHMS

For a classification problem (supervised or unsupervised) there are two different approaches:

i) In a direct approach called DLA, we use a functional form of the generalized linear model explicitly to determine its parameters directly by using maximum likelihood. (There is an efficient algorithm for finding such solutions known as Iterative Reweighted Least Squares IRLS). In this direct approach, we are maximizing a likelihood function defined – through the conditional distribution $P(C_K|\vec{x})$ which represents a form of DLA.

Example: Linear Regression
Let $\vec{x}^i$ be an i[th] sample vector of n dimensions.
When we consider m number of samples, we can construct a data matrix X of mxn dimensions, i.e., X = $[\vec{x}^1, \vec{x}^2, \ldots \vec{x}^m]$. Let $y^{(i)} = h_\theta(\vec{x}^i)$ be the target value of $\vec{x}^i$.
The linear regression hypothesis is
$y^{(i)} = h_\theta(\vec{x}^i) = \theta_0 + \theta_1 x_1{}^{(i)} + \cdots + \theta_n x_n{}^{(i)} = \sum \theta_j x_j{}^{(i)} = \vec{x}^{(i)T}\vec{\theta}$
The error for the i[th] sample is $e^{(i)} = \vec{x}^{(i)T}\vec{\theta} - y^{(i)}$
We define the cost function $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(\vec{x}^{(i)T}\vec{\theta} - y^{(i)}\right)$
And minimize J with respect to $\vec{\theta}$ by Ordinary Least Squares method.

There are two methods of minimizing J:

a) Explicit minimization by obtaining the normal equations.

To minimize J, we set the derivatives of J with respect to $\vec{\theta}$ equal to zero and obtain the normal equations $\vec{\theta} = (X^T X)^{-1} X^T \vec{y}$

b) Iterative Algorithm: Gradient Descent Algorithm: We use a search algorithm that starts with some "initial guess" for $\vec{\theta}$ and that repeatedly changes $\vec{\theta}$ to make $J(\theta)$ smaller. The update equation is: $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

ii) In the indirect approach called GLA, we fit the class-conditional densities $P(C_K|\vec{x})$ and the class priors separately (given in the training set) and then apply Bayes' theorem to find the posterior $P(C_K|\vec{x})$. This represents an example of generative modelling because we could take such a model & generate synthetic data by drawing values of $\vec{x}$ from the marginal distribution $P(\vec{x})$.

## V. RANDOM PROCESSES

### 5.1 Markov Models

Given a set of states $S = \{s_1, s_2, \dots, s_{|S|}\}$ we can observe a series over time 0 to T. For example, we might have the states from a weather system S = {sun, cloud, rain} with |S|=3 and observe the weather over a few days $\{z_1 = s_{sun}\,,\ z_2 = s_{cloud}\,, z_3 = s_{cloud}\,,\ z_4 = s_{rain}\,,\ z_5 = s_{cloud}\}$ with T = 5.

The observed states of our weather example represent the output of a random process over time. Without some further assumptions, state $s_j$ at time t could be a function of any number of variables, including all the states from times 1 to t − 1 and possibly many others that we don't even model. However, we will make two Markov assumptions that will allow us to tractably reason about time series.

The limited horizon assumption is that the probability of being in a state at time t depends only on the state at time t−1. The intuition underlying this assumption is that the state at time t represents "enough" summary of the past to reasonably predict the future. Formally:

$$P(z_t|z_{t-1}, z_{t-2}, \dots, z_1) = P(z_t|z_{t-1})$$

The stationary process assumption is that the conditional distribution over next state given current state does not change over time.

$$P(z_t|z_{t-1}) = P(z_2|z_1);\ \ t\epsilon\ 2 \dots T$$

$$A = \begin{array}{c|cccc} & s_0 & s_{sun} & s_{cloud} & s_{rain} \\ s_0 & 0 & .33 & .33 & .33 \\ s_{sun} & 0 & .8 & .1 & .1 \\ s_{cloud} & 0 & .2 & .6 & .2 \\ s_{rain} & 0 & .1 & .2 & .7 \end{array}$$

Note that these numbers represent the intuition that the weather is self-correlated: if it's sunny it will tend to stay sunny, cloudy will stay cloudy, etc. This pattern is common in many Markov models and can be observed as a strong diagonal in the transition matrix. Note that in this example, our initial state s0 shows uniform probability of transitioning to each of the three states in our weather system

Two questions of a Markov Model

1. What is the probability of a particular sequence of states $\vec{z}$ ?
2. And how do we estimate the parameters of our model A such to maximize the likelihood of an observed sequence $\vec{z}$ ?

1.Probability of a state sequence

We can compute the probability of a particular series of states $\vec{z}$ by use of the chain rule of probability:

$$P(\vec{z}) = P(z_t, z_{t-1}, \dots, z_1; A)$$
$$P(\vec{z}) = P(z_t, z_{t-1}, \dots, z_1, z_0; A)$$
$$P(\vec{z}) = P(z_t|z_{t-1}, \dots, z_1; A)P(z_{t-1}|z_{t-2}, \dots, z_1; A) \dots P(z_1|z_0; A)$$
$$P(\vec{z}) = P(z_t|z_{t-1}; A)P(z_{t-1}|z_{t-2}; A) \dots P(z_1|z_0; A)$$
$$= \prod_{t=1}^{T} P(z_t|z_{t-1}; A)$$
$$= \prod_{t=1}^{T} A_{z_{t-1}z_t}$$

For example, for the sequence $\vec{z} = (z_1, z_2, z_3, z_4, z_5) = (s, c, r, r, c)$, we have

$$P(z) = P(sun|s_0)P(cloud|sun)P(rain|cloud)P(rain|rain)P(cloud|rain)$$
$$= 0.33 \times 0.1 \times 0.2 \times 0.7 \times 0.2 = 0.000924$$

2.Estimation of transition parameters A

We determine the parameters A that maximize the log-likelihood of sequence z.

$$l(A) = log \prod_{t=1}^{T} A_{z_{t-1}z_t} = \sum_{t=1}^{T} log\, A_{z_{t-1}z_t}$$
$$= \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} \sum_{t=}^{T} 1\{z_{t-1} = s_i \wedge z_t = s_j\} log A_{ij}$$

In the last line we use an indicator function whose value is 1, when the condition holds, and 0 otherwise, to select the observed transition at each time-step.
When solving this optimization problem, we should keep in mind that A is a valid transition matrix. Thus, $max_A\, l(A)$ such that

$\sum_{j=1}^{|S|} A_{ij} = 1$ , I = 1,…,|S|
And
$A_{ij} \geq 0$ , i ,j = 1,2,…,|S|

This constrained optimization problem can be solved in closed form using the method of Lagrange Multipliers, and we can show that

$$A_{ij} = \frac{\sum_{t=}^{T} 1\{z_{t-1} = s_i \wedge z_t = s_j\}}{\sum_{t=}^{T} 1\{z_{t-1} = s_i\}}$$

The ML parameter corresponds to the fraction of time when we are in state i that we transitioned to state j.

**5.2 Hidden Markov Model (HMM)**

Now as earlier (in MM), there is a series of states $\vec{z} = (z_1, z_2, \dots, z_t)$ drawn from a state alphabet $S = (s_1, s_2, \dots, s_{|S|})$ with a state transition matrix A. Now, Z is a hidden state and we don't get to observe the actual sequence of states in Z.

However, at each time step t, the system randomly evolves from state $z_i$ to $z_{i+1}$, while emitting symbols from x governed by an emission probability matrix B. An HMM can be visualized by imagining that 2 different dice are associated with each state. Both the transitions and emissions depend on the current state only and not on the past. Only the symbols emitted by the system are observable, not the underlying random walk between states $z_j(s_k)$ to $z_{j+1}(s_l)$.

We model the probability of generating an output observation as a function of our hidden state by the output independence assumption.
$B_{jk} = P(x_t = v_k | z_t = s_j) = P(x = v_k | x_1, \dots, x_T, z_1, \dots, z_T)$
The matrix B encodes the probability of our hidden state generating output $v_k$ given that the state at the corresponding time was $s_j$.
Thus, the transition model specifies the values $P(Z_t | Z_{0,\dots,t-1}) = P(Z_t | Z_{t-1}) = A_{ij}$
And the conditional probability $P(x_t | x_{0:t}, z_{1:t}) = P(x_t | z_t) = B_{ij}$
For the sensor world, the specification is only for a particular state$z_i$. Thus, B is a diagonal matrix.
The probability of an observed sequence x can be worked out in a similar fashion of the Markov Model and can be shown to be:

$$P(\overrightarrow{z_{0:t}}, \overrightarrow{X_{1:t}}) = P(\overrightarrow{z_0}) \prod_{i=1}^{t} P(\overrightarrow{z_i} | \overrightarrow{z_{i-1}}) P(\overrightarrow{x_i} | \overrightarrow{z_i})$$

The parameters A and B can be found out as in A for MM, but involves tedious calculations using EM algorithm (Expectation Maximization)

## REFERENCES


[1]. Tom M. Mitchell: Machine Learning, McGraw Hill Education, Indian Edition (2013).
[2]. Stuart J Russell and Peter Norvig.: Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson Education (2011).
[3]. Christopher M. Bishop.: Pattern Recognition and Machine Learning, Springer (2006).
[4]. Elaine Rich, Kevin Knight, S B Nair: Artificial Intelligence, Tata McGraw Hill Education Pvt. Ltd., 3rd Edition (2010).
[5]. Andrew Ng, Lecture Notes in Machine Learning, Stanford University, USA. (2017).


## AUTHOR INFORMATION


**Mr. Uday Nayak** is an Assistant Professor inDepartment of Information Technology at Don Bosco Institute of Technology. He has an MS degree in Computer Science from University of Toledo, Ohio, USA and has 15 years of teaching experience. He is the Founder Convener of the "Artificial Intelligence Club".


**Mr. Joel Braganza** is currently in final year of BE(Bachelor of Engineering) in IT at Don Bosco Institute of technology. He is the Founder Chairperson of the "Artificial Intelligence Club" and his final year project is in the area of "Generative Adversarial Networks" in which an image is generated by a neural network when a text phrase is given.

**Ms. S. Mahalaxmi** is an Assistant Professor inDepartment of Information Technology at Don Bosco Institute of Technology. She has 20 years of teaching experience. She is the Founder Convener of the "Software Testing Club".

Udaychandra A Nayak "Primer on Computational Techniques for Machine Learning "American Journal of Engineering Research (AJER), vol. 7, no. 09, 2018, pp. 214-229