A HYBRID APPROACH OF DEA AND SUPPORT VECTOR MACHINE FOR DECISION MAKING IN OPTIMISTIC& PESSIMISTIC ENVIRONMENT

Thesis Submitted in partial fulfilment of the requirement for
the award of the degree of
Master of Science
In
Mathematics and Computing

Submitted by
Sarbjeet Kaur
Reg.No.301703025

Under the guidance of
Dr. Jolly Puri

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

School of Mathematics
Thapar Institute of Engineering & Technology (Deemed to be University)
Patiala-147001 (PUNJAB)
INDIA
July, 2019

Certificate

I hereby certify that the dissertation entitled, **"A Hybrid Approach of DEA and Support Vector Machine for Decision Making in Optimistic & Pessimistic Environment"** in the partial fulfillment of the requirement for the award of degree of Master of Science in the School of Mathematics, Thapar Institute of Engineering and Technology(Deemed to be University),comprises of my own research work which is carried out under the supervision and the guidance of Dr. Jolly Puri, Assistant Professor, School of Mathematics Patiala from the period january 2019 to July 2019.The part of the work presented in this dissertation has not been submitted either in part or in full to this or to any other University/Institute for the award of degree.

Sarbjeet Kaur

Sarbjeet Kaur

(301703025)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Jolly Puri.

Dr. Jolly Puri

Assistant Professor,
School of Mathematics,
Thapar Institute of Engineering and Technology (Deemed to be University), Patiala.

ACKNOWLEDGEMENTS

I would like to thank a number of people who have contributed to this thesis in different ways. Firstly, I pay my obeisance to the **GOD**, the almighty who awarded me good health, courage, inspiration, zeal and light etcetera without which this work would never be completed.

After GOD, I would like to express my sincere and deepest gratitude to my supervisor, **Dr. Jolly Puri, Assistant Professor, School of Mathematics, Thapar Institute of Engineering and Technology, Patiala** for her invaluable guidance, patience, constant encouragement, enthusiasm and great knowledge. Her simplicity and affectionate attitude always inspired and encouraged me throughout my research work and without this inspiration, it would have been impossible for me to complete this study.

I owe my special thanks to **Dr. Satish Kumar Sharma**, Head of Department, School of Mathematics, Thapar Institute of Engineering and Technology, Patiala for his warm approach, healthy environment, timely motivation, sympathetic attitude and unfailing help during my study. I am highly thankful to **Dr. Paramjeet Singh**, Assistant Professor and PG Coordinator, School of Mathematics, Thapar Institute of Engineering and Technology, Patiala for his valuable suggestions and continual inspiration that have been very helpful for my thesis work. I would like to thank all the professors in the **School of Mathematics** for their constant support.

I owe my most gratitude to **Dr. Prashant Rana**, Assistant Professor, Computer Science Engineering Department for sharing his knowledge and valuable time which helps me a lot during my thesis work.

I sincerely express my gratitude from the core of my heart to Mr. L.S. Munday for being a soulmate and making me free from the worries all the time. His valuable suggestions, extreme support, selfless affection and care always helps me to remain happy and march towards the goals.

Words prove a meagre media to write down my feelings for my father Mr. Sucha Singh, mother Mrs. Manjit Kaur, brother Gurpreet Singh and all the relatives for providing me constant inspiration, divine presence, co-operation and supporting me spiritually throughout.

(Sarbjeet Kaur)

ABSTRACT

Data envelopment analysis (DEA) developed by Charnes, Cooper and Rhodes in 1978, is a powerful non- parametric technique used to measure the relative performance of similar organizations. This technique is used to compare the relative performance of several homogenous units called decision making units (DMUs). DEA is based on linear programming approach. The presence of multiple inputs and outputs is the great advantage of DEA over other techniques which measure the performance of organizations. This is because, generally, multiple inputs and outputs are not comparable, but it also makes the technique a little bit difficult. The CCR model (Charnes et al., 1978) and the BCC model (Banker et al., 1984) are some standard DEA models which are based upon the assumption of input minimization and output maximization. However, DEA technique requires accurate input and output data for its successful implementation. But data of real- life problems is not always accurate or crisp. For instance, data in manufacturing sector, banking sector and health care sector is usually complex or imprecise due to lack of some information or some other reasons. So, to solve these types of problems, input and output data is represented by fuzzy numbers or interval numbers.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning is of three types: Supervised Machine Learning, Semi-Supervised Machine Learning and Unsupervised Machine Learning. SVM is a supervised machine learning algorithm that is used for the classification, regression or other purposes like outlier detection. It is a fast and dependable classification algorithm that performs very well with a limited amount of data. The goal of the SVM is to train a model that assigns new unseen objects into a particular category.

Over the last thirty years, the organization failure prediction is an important issue that has attracted the intention of wide academic studies. Organization failure refers to situation when bill is overdrawn, the company is not able to pay the wages etc. So, the objective of our present study is to use the optimistic and pessimistic efficiencies of several DMUs calculated by using proposed DEA models and SVM technique to predict the accuracy of organization failure. In the proposed study (IDEA-SVM), DEA technique use the interval data to calculate the optimistic and pessimistic efficiencies. Further, using SVM approach, prediction of efficiencies of organization is estimated.

The summary of thesis is given below.

Chapter 1 is introductory type. This chapter gives the brief introduction of data envelopment analysis and its various approaches. It also represents the introduction of machine learning along with its different categories. It gives the brief review of Support vector Machine which is a widely used algorithm of supervised machine learning.

Chapter 2 includes the basic definitions and important theorems related to convex optimization problem. It also determines the dual of convex optimization problem and presents the brief review of CCR model developed by Charnes, Cooper and Rhodes in 1978. It presents the Charnes-Cooper transformation which transform the Fractional Programming Problem (LFP) into Linear Programming Problem (LPP).

Chapter 3 includes the working of SVM and basic definitions related to it. It represents the properties of maximal margin method and various metrics to compare the hyperplanes. Finally, it derives the SVM optimization problem and presents the relationship between primal and dual of SVM optimization problem.

In **Chapter 4** DEA models from optimistic as well as pessimistic point of view are presented. In these models, interval data is used. Then on these models SVM technique is applied and finally present IDEA-SVM approach in optimistic environment and IDEA-SVM approach in pessimistic environment.

TABLE OF CONTENTS

CHAPTER 1

Introduction

1.1 Data envelopment analysis (DEA)

Data envelopment analysis (DEA) also called frontier analysis is a widely used technique and is based upon linear programming. It was developed by Charnes, Cooper and Rhodes in 1978. This technique is used to measure or compare the relative performance of several homogenous units called decision making units (DMU). It is basically a non-parametric technique. DEA has many applications in industrial areas like bank branches, shops, hospitals, educational institutions and many other where units do similar jobs. The presence of multiple inputs and outputs is the great advantage of this technique because multiple inputs and outputs are generally not comparable, but it also makes the technique a little bit difficult. Some standard DEA models are: The CCR model (Charnes et al., 1978) and the BCC model (Banker et al., 1984). These models are based upon the assumption of input minimization and output maximization and used only desirable, exact valued inputs and outputs initially. In real applications, it is also possible that the production process may produce undesirable outputs along with the desirable. There are many approaches that have been reported to use the undesirable outputs in data envelopment analysis. According to Sheel (2001), these approaches are of two types: direct and indirect approach. In indirect approach, these undesirable outputs are converted to desirable ones by a function f which is basically monotonically decreasing function (Scheel, 2001; Seiford and Zhu, 2002; Liu et al., 2010). On the other hand, in direct approach, no data transformation is done, and the undesirable outputs are used directly in DEA model (See Korhonen and Luptacik, 2004; Puri and Yadav, 2014a). Generally, conventional DEA requires crisp data for both inputs and outputs. But the input and output data which is accessible from the real-life problems is sometimes not crisply defined due to incomplete information and other reasons. So, to deal with imprecision, the data can be represented in the form of intervals or fuzzy numbers. Initially, Cooper et al. (1999; 2000) worked on the problem of DEA with imprecise data. The DEA model so formed was a linear programming model which produce the efficiency as a deterministic numerical value less than or equal to one.

In this thesis, the objective is to analyse the performance in different environments, so the data of inputs and outputs is taken in the form of intervals. Since the data is in the form of intervals, so the final efficiency of each DMU will also be of the form of intervals. All the DEA models use following interval arithmetic for the interval data. If any $x\epsilon[x^L, x^U]$ where $x^L \leq x^U$; then x is called an interval number. Let x and y be two interval numbers such that $x = [x^L, x^U]\ and\ y = [y^L, y^U]$.Then arithmetic operations defined on x and y are given by

1. Addition: $x + y = [x^L + y^L, x^U + y^U]$
2. Subtraction: $x - y = [x^L - y^U, x^U - y^L]$
3. Multiplication: $x * y = [\min(a), \max(a)], where\ \ a = \{x^L x^L, x^L x^U, x^U x^L, x^U x^U\}$

4.Scalar Multiplication: $kx = [kx^L, kx^U]\ for\ k > 0\ and\ k \in R$ and

$$kx = [kx^U, kx^L]\ for\ k < 0\ and\ k \in R$$

5. Division: $\frac{x}{y} = [\frac{x^L}{y^U}, \frac{x^U}{y^L}]$

Entani et al. (2002) measure the performance of each DMU from optimistic and pessimistic prospective. The efficiency so calculated for each DMU, was in the form of interval. The efficiencies calculated from optimistic prospective are called best relative efficiencies or optimistic efficiencies which gives the favourable weights for the DMUs. On the other hand, the efficiencies evaluated from pessimistic prospective are called worst relative efficiencies or pessimistic efficiencies which provides the unfavourable weights for the DMUs. If the best relative efficiency or optimistic efficiency of the DMU under evaluation come out to be equal to one, then it is called optimistic efficient; otherwise it is called optimistic non-efficient. The performance of optimistic efficient DMU is better than optimistic non-efficient DMU. Similarly, if the worst relative efficiency or pessimistic efficiency of the DMU under evaluation come out to be equal to one, then it is called pessimistic inefficient; otherwise it is called pessimistic non- inefficient. The performance of pessimistic inefficient DMU is worse than pessimistic non-inefficient DMU. (Wang et. al., 2007).

As Entani et al. (2002) consider both optimistic and pessimistic efficiencies to calculate the efficiency interval of DMU, there were some drawbacks of this model. Their model can measure the efficiency interval of DMU only from pessimistic point of view and this pessimistic approach uses only one input and one output and ignore rest of the input and output data. Furthermore, their model fails to identify pessimistic inefficient DMUs adequately.

Wang et al. (2006) calculated the efficiency interval of each DMU from different point of view. They use two another DEA models namely ideal DMU (IDEA) and anti-ideal DMU (ADEA) in addition to existing two virtual DEA models. These four models are then used to measure the efficiency interval of DMU from both optimistic and pessimistic point of view. The two efficiencies are then integrated to calculate the relative closeness index (RC). This index is used to assign rank to each DMU. The use of fixed weights for all the DMUs is the main drawback of their models.

Azizi (2011) study the DEA model from output oriented prospective. Their model calculated the lower bound of efficiency interval of each DMU from optimistic point of view. But their model had two drawbacks. Firstly, their model used only one input and one output to calculate the lower bound of efficiency interval and ignore rest of the input and output data. Secondly, their model fails to identify optimistic efficient DMUs correctly.

1.2 Support Vector Machine (SVM) technique in machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow computers learn automatically with human intervention or assistance and adjust actions accordingly. Machine learning algorithms find natural patterns in data that generate insight and help you make better decisions and predictions. It is used when you have a complex task or problem involving a large amount of data and lots of variables, but no existing formula or equation. With the rise in big data, machine learning has become a key technique for solving problems in areas, such as:

Image processing and computer vision, Computational biology, Energy production, Natural language processing, Classification of decision-making units (DMUs) in terms of efficient or inefficient DMUs. Machine learning is of three types: Supervised Machine Learning, Semi-Supervised Machine Learning and Unsupervised Machine Learning.

1.2.1 Types of Machine Learning

I. Supervised Learning

A supervised learning algorithm takes a known set of input and output data and it tries to build a model. The goal is to approximate the mapping function to make best guess predictions for the unlabelled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data. Learning stops, when the algorithm achieves an acceptable level of performance. Supervised machine learning consists of two stages: (a) Training stage and (b) Testing stage. In the training stage, the supervised machine learning builds a model while in the testing stage it is trying to predict output.

Supervised learning uses two types of techniques to develop predictive models:

1) **Classification techniques predict discrete responses**: Classification models classify input data into categories. Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for hand-writing recognition use classification to recognize letters and numbers, predicting gender of person by his/her writing style, predicting whether monsoon will be normal next year, predicting whether an email is genuine or spam. Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, k-nearest neighbour, Naïve Bayes, discriminant analysis, logistic regression, and neural networks.
2) **Regression techniques predict continuous responses**: Regression techniques are used if the nature of the response is a real number. For example, predicting house price based on area, predicting age of person, predicting number of copies of music albums will be sold next month. Typical applications include electricity load forecasting and algorithmic trading. Common regression algorithms include linear model, nonlinear model, regularization, stepwise regression, boosted and bagged decision trees.

II. Unsupervised Learning

Unsupervised machine learning algorithms have a known set of input data and no corresponding output data. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. Unsupervised learning uses clustering technique to develop predictive models.

Clustering: A clustering problem is to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour. Applications for cluster analysis include gene sequence analysis, market research, and object recognition. For example, if a cell phone company wants to optimize the locations where they build cell phone towers, the company uses clustering algorithms to design the best placement of cell towers to optimize signal reception for groups, or clusters, of their customers. Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, fuzzy c-means clustering, and subtractive clustering.

III. Semi-Supervised Machine Learning

Problems where you have a large amount of input data and only some of the data has known output are called semi-supervised learning problems. This type of machine learning sits in between both supervised and unsupervised learning.

IV. Reinforcement Learning

Reinforcement learning (RL) is one approach that aims to produce intelligent programs, often called agents, through a process of learning and evolving. RL setup which is composed of two components, an agent and an environment. The environment starts by sending a statement to the agent, which then based on its knowledge to take an action in response to that state. After that, the environment sends a pair of next state and reward back to the agent. The agent will update its knowledge by the environment to evaluate its last action. The loop keeps going on until the environment sends a terminal state. Some of commonly used RL algorithms are: Q-Learning, SARSA, Deep Q Network etc. The applications of Reinforcement Learning involve Robotics and Industrial Automation, Data Science and Machine Learning etc.

1.2.2 Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that is used for the classification, regression or other purposes like outlier detection. It is a fast and dependable classification algorithm that performs very well with a limited amount of data. The goal of the SVM is to train a model that assigns new unseen objects into a particular category. It achieves this by creating a linear partition of the feature space into two categories. Based on the features in the new unseen objects, it places an object "above" or "below" the separation plane, leading to a categorization. This makes it an example of a non-probabilistic linear classifier. It is non-probabilistic, because the features in the new objects fully determine its location in feature space and there is no stochastic element involved.

CHAPTER 2

Preliminaries

2.1 An overview of convex programming problem

The mathematical model of a convex programming problem (Deng et. al.,2013) which is an optimization problem is given by

Model 2.1

$$\begin{array}{ll} \text{minimize} & f_0\,(x) \\ \text{subject to} & f_i(x) \leq 0 \quad where\ i = 1,2,\ldots\ldots\ldots\ldots, m, \\ & h_i\,(x) = 0 \quad where\ j = 1,2,\ldots\ldots\ldots\ldots., p, \end{array}$$

where x is optimization variable of this problem, $f_0\,(x)$ & $f_i(x),\ i = 1,2,\ldots\ldots\ldots\ldots, m$ are continuously differentiable & convex functions on R^n and $h_i\,(x),\ j = 1,2,\ldots\ldots\ldots\ldots., p$ are linear functions. If $m + p = 0$, i.e., if the problem does not have any constraints, then this optimization problem is called unconstrained optimization problem.

Definition 1 A point satisfying all the constraints is called feasible point. The set of all such points constitute the feasible region D where $D = \{x\,|f_i(x) \leq 0, i = 1,2,\ldots.., m; h_i(x) = 0, i = 1,2,\ldots\ldots.., p; x \in R^n\}$.

Definition 2 The optimal value $p^*$of convex optimization problem is defined as the greatest lower bound of the objective function $f_0\,(x)$ in the feasible region D when D is not empty and is equal to infinity if D is empty. Mathematically, p^*= $\inf\{f_0(x)|x \in D\}, when\ D \neq \emptyset$ otherwise equal to ∞ .

Definition 3 The point x^* is called global solution of the convex optimization problem, if x^* is the feasible point and $f_0\,(x^*)$ = $\inf\{f_0(x)|x \in \text{D}\} = p^*$ where D is the feasible region. The point x^* is called local solution of convex optimization problem, if x^* is the feasible point and there exist $\in > 0$ such that $f_0\,(x^*)$ = $\inf\{f_0(x)\big|x \in D; \big||x - x^*|\big| \leq \in\}$. The set of all local and global solutions are called corresponding solution set.

Definition 4 Let R^n be a set and let S be its subset. Then the set S is said to be convex set if the line segment joining any two points of S lies entirely in S. In other words, for any $x, y \in S\ and\ \lambda \in$ [0,1], then we have $\lambda x + (1 - \lambda)y \in S$.

Definition 5 A function f defined on R^n is said to be convex function if we take any two points $x, y \in R^n$, then the graph of function f lie below the line joining the points $\big(x, f(x)\big) and \big(y, f(y)\big)$, i.e., for $\lambda \in [0,1]$, we have

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

If strict inequality holds in above, then the function is called strict convex function on R^n.

Theorem 1 Let f be a function which is continuously differentiable on R^n. Then f is called convex function iff for all $x, y \in R^n$; $f(x) \geq f(y) + \nabla f(y)^T (x-y)$ holds. If strict inequality holds in this inequality, then the function is called strict convex function on R^n.

Theorem 2 Let us consider the Quadratic function $f(x) = \frac{1}{2} x^T H x + r^T x + \delta$ where $H \in R^{n*n}, r \in R^n, \delta \in R$..Then $f(x)$ is a convex function on R^n if H is positive semi definite and $f(x)$ is strictly convex function on R^n if H is positive definite.

Proof The second order Taylor Polynomial of $f(x)$ at point $x = a$ is given by

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2!} f"(a)(x-a)^2$$

or

$$f(x) = f(a) + \nabla f(a)(x-a) + \frac{1}{2!} \nabla^2 f(a)(x-a)(x-a)^T$$

Let H=$\nabla^2 f(a)$, Then

$$f(x) = f(a) + \nabla f(a)(x-a) + \frac{1}{2!} (x-a) H (x-a)^T$$

As H is Semi definite, so this implies

$$(x-a) H (x-a)^T \geq 0$$

Hence,

$$f(x) \geq f(a) + \nabla f(a)(x-a)$$

Using Theorem 1, we proved that $f(x)$ is convex function.

2.1.1 Dual of convex programming problem

Consider a convex programming problem defined in Model 2.1. The optimal value of this problem is given by $p^* = \inf\{f_0(x) | x \in D\},\ when\ D \neq \emptyset$ where D is the feasible region.

Now, the Lagrangian function is given by

$$L(x, \lambda, \mu) = f_0(x) + \sum_{i=1}^{m} f_i(x) \lambda_i + \sum_{i=1}^{p} h_i(x) \mu_i$$

where λ_i and $\mu_i$are Lagrangian vectors.

Clearly, $L(x, \lambda, \mu) \leq f_0(x)$, when $x \in D$

Hence,

$$\underset{x \in R^n}{Inf} L(x,\lambda,\mu) = \underset{x \in D}{Inf} L(x,\lambda,\mu) \leq \underset{x \in D}{Inf} f_0(x) = p^*$$

Using Lagrangian function,

$$g(\lambda,\mu) = \underset{x \in R^n}{Inf} L(x,\lambda,\mu)$$

$$g(\lambda,\mu) \leq p^*$$

This expression shows that $g(\lambda,\mu)$ is the greatest lower bound (g.l.b.) of p^***.** Choosing the best out of these lower bounds leads to following optimization problem:

Model 2.2

$$max g(\lambda,\mu) = \underset{x \in R^n}{Inf} L(x,\lambda,\mu)$$

$$\lambda \geq 0$$

This is the corresponding dual problem.

Theorem 3 (**Weak Duality Theorem**) Let p^* and d^* be the optimal value of primal problem and corresponding dual problem respectively. Then

$$p^* = inf\{f_0(x) | f_i(x) \leq 0, i = 1,2,\ldots\ldots, m;$$

$$h_i(x) = 0, i = 1,2,\ldots\ldots\ldots, p; x \in R^n\} \geq \sup\{g(\lambda,\mu) | \lambda \geq 0\} = d^*$$

Note: (i) When equality occur in the above expression then the theorem is called **Strong Duality Theorem**. (ii) If strict inequality occurs in non- linear constraints, then these conditions are called **Slater's conditions**. (iii) If the primal problem is unbounded below then the dual problem is infeasible and if the dual problem is unbounded above then primal problem is infeasible.

2.2 An overview of DEA

2.2.1 Basic CCR Model

This model was put forward by Charnes, Cooper and Rhodes in 1978. In this model, we form virtual input and virtual output for each DMU with the help of m input and s output weight vectors v_i (i=1, 2……, m) & $u_r (r = 1,2 \ldots\ldots, s)$ respectively and then try to calculate the weight vectors using LP so that the efficiency is maximum. Hence,

Virtual input = $\sum_{i=1}^{m} v_i x_{ij}$ where x_{ij}= input i^{th} consumed by j^{th} DMU, (i=1, 2……, m),

Virtual output = $\sum_{r=1}^{s} u_r y_{rj}$ where y_{rj}= output r^{th} produced by j^{th} DMU, $(r = 1,2 \ldots\ldots, s)$, and

Efficiency of DMU_j in DEA is defined as

$$E_j = \mathit{Efficiency} = \frac{Virtual\ Output}{Virtual\ Input} = \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}},\ \forall j = 1,2 \ldots\ldots\ldots, n$$

Let there are n DMUs. We calculate the efficiency of each of DMU_j (j=1, 2, …., n) w.r.t. all the (n-1) DMUs and identify the most efficient and inefficient DMU. Let's calculate the efficiency of DMU_0 (0=1, 2 ,...n) w.r.t. all DMU_j (j=1,2,….,n).For this we solve the fractional programming problem to get the values of m input and s output weight vectors v_i and u_r respectively that maximize the efficiency of DMU_0.

Model 2.3

$$max\ E_0 = \frac{\sum_{r=1}^{s} u_r y_{r0}}{\sum_{i=1}^{m} v_i x_{i0}}$$

$$\text{subject to } E_j = \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1\ \forall j = 1,2 \ldots\ldots.., n$$

$$u_r \geq 0\ \ \forall r = 1,2, \ldots\ldots\ldots., s$$

$$v_i \geq 0\ \ \forall i = 1,2, \ldots\ldots\ldots., m$$

2.2.2 Charnes-Cooper Transformation

This transformation was introduced by Charnes and Cooper in 1967. Using the transformation, we convert the linear fractional programming problem into LP problem which is further solved by simplex method. Model 2.3 is a fractional programming problem which can be converted to linear programming problem by using Charnes-Cooper transformation.

Relation between LFP and LP Problem

The mathematical model of LP problem is given by

Model 2.4

$$\text{max (min) } Z = c_1x_1 + c_2x_2 + c_3x_3 + \ldots\ldots\ldots + c_nx_n$$

Subject to

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots\ldots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots\ldots + a_{2n}x_n = b_2$$

$$\vdots$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots\ldots\ldots + a_{mn}x_n = b_m$$

$$x_1, x_2, x_3, \ldots\ldots\ldots, x_n \geq 0$$

Similarly, the mathematical model of LFP is given by

Model 2.5

$$max\ Z = \frac{cx+\alpha}{dx+\beta}$$

Subject to

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots\ldots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots\ldots + a_{2n}x_n \leq b_2$$

$$\vdots$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots\ldots\ldots + a_{mn}x_n 1 \leq b_m$$

$$x_1, x_2, x_3, \ldots\ldots\ldots, x_n \geq 0$$

Where x, c, d$\in R^n$ and $\alpha, \beta \in$ R

Let us assume that

1. The region $X= \{x \in R^n : Ax \leq b, x \geq 0\}$ is non-empty and bounded feasible region.
2. Let the denominator of Z be strictly positive.

Under these assumptions, Charnes and Cooper put the following transformation

$$y = tx \quad where\ \ t = \frac{1}{dx+\beta}; \qquad t \geq 0$$

This transformation converts the above linear fractional problem into the following linear problem.

To get linear form, multiply the by t,

$$max\ Z = \frac{cxt+\alpha t}{t(dx+\beta)} \gg max\ Z = \frac{cy+\alpha t}{1}$$

Subject to

$$Axt - bt \leq 0 \gg Ay - bt \leq 0$$

$$dy + \beta t = 1$$

$$t \geq 0$$

The optimal solution as well as the optimal objective value of both LFP and LP problems is same as the above transformation is of reversible nature under the assumption as given above. This problem can be solved by simplex method. The optimal solution obtained from this method is relatively slower than the optimal solution obtained from its dual.

Hence, using Charnes-Cooper transformation, the Model 2.3 can be written as

Model 2.6

$$\begin{aligned} \max\ & E_0 = \textstyle\sum_{r=1}^{s} u_r y_{r0} \\ \text{subject to}\ & \textstyle\sum_{i=1}^{m} v_i x_{i0} = 1, \\ & \textstyle\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \le 0\ \forall\, j = 1,2,\dots\dots,n \\ & u_r \ge 0\ \ \forall\, r = 1,2,\dots\dots\dots,s \\ & v_i \ge 0\ \ \forall\, i = 1,2,\dots\dots\dots,m \end{aligned}$$

Theorem 4 (**Unit Invariance Theorem**) The optimal objective value of both LFP and LP problem is same and does not depend upon the units of inputs and outputs provided the units are same for each DMU.

Proof See Cooper et al. (2007).

CHAPTER 3

Support Vector Machine Technique

3.1 Support vector machine (SVM)

Support vector machine is a well-known supervised learning algorithm supervised machine learning algorithm. The basics and working of SVM can be best understood with the help of the following example: Let us suppose that we have two colored balls: orange and green, and let our data set has two features say X and Y.

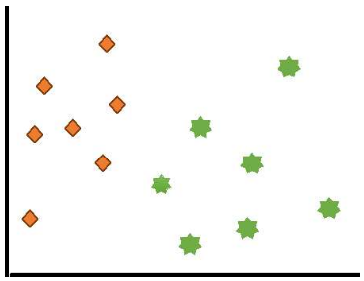

Figure 1

Now, what does SVM do is that it takes these data points and build a hyperplane that segregate both balls. The hyperplane in this case is nothing but the decision boundary. Anything that will fall one side of it will be classified as orange; and anything that will fall on other side will be classified as green.

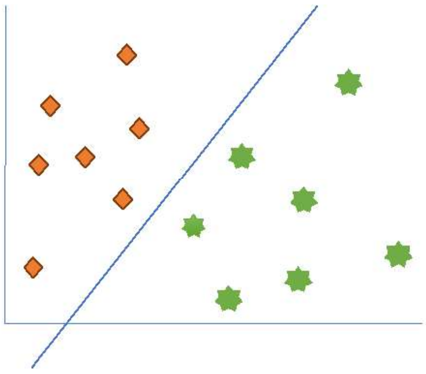

Figure 2

3.1.1 SVM terminology

Definition 6 Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

Definition 7 In geometry, a Hyperplane is a subspace whose dimension is one less than that of its ambient space. The Hyperplane is just a line for two-dimensional space and is actually a hyperplane in *n*-dimensional space having dimension *n*-1. Fig.2 presents the two-dimensional case. The two-dimensional linearly separable data can be separated by a line. The function of the line is $y=ax+b$. We rename x with x_1 and y with x_2 and we get: $ax_1-x_2+b=0$

If we define $x=(x_1,x_2)$ and $w=(a,-1)$, we get: w.x+b=0

This equation is derived from two-dimensional vectors. However, it also works for any number of dimensions. This is the equation of the hyperplane.

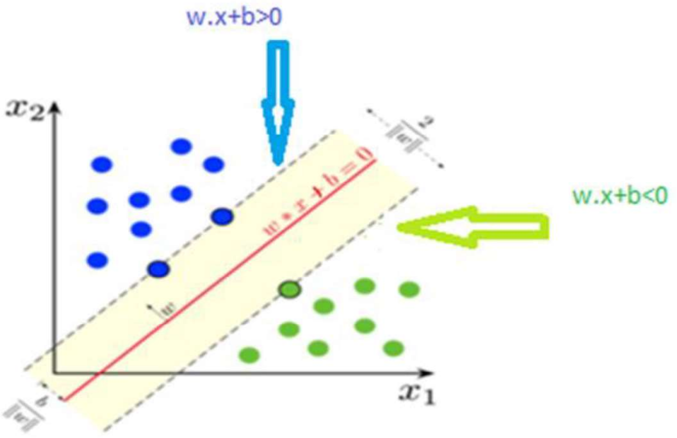

Figure 3 (Source: https://en.wikipedia.org/wiki/File:SVM_margin.png)

Definition 8 Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as n-dimensional vector (a list of n numbers), and we want to know whether we can separate such points with a (n-1)-dimensional hyperplane. This is called a linear classifier.

Define the hypothesis function h as**:**

$$h(x_i) = \begin{cases} +1, if w.x + b \geq 0 \\ -1, if w.x + b \leq 0 \end{cases}$$

The point above or on the hyperplane will be classified as class +1, and the point below the hyperplane will be classified as class -1.

The aim of SVM learning algorithm is to find a hyperplane which could separate the data accurately. There might be many such hyperplanes. But the objective is to find the best one, which is often referred as the optimal hyperplane.

There are two thumb rules to identify the right hyperplane: -

1) Select that hyperplane which segregates the two classes better.

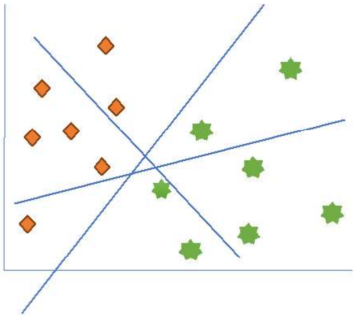

Figure 4

Here we have three hyperplanes. Now, identify which is the right hyperplane. In this case, hyperplane "B" is the best hyperplane.

2) Choose a hyperplane with the greatest possible margin between the hyperplanes. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; since in general the larger the margin the lower the generalization error of the classifier.

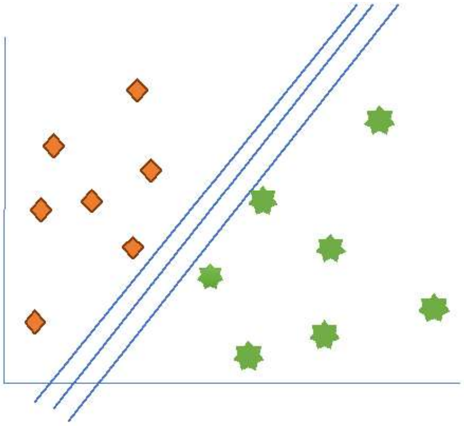

Figure 5

Here the word margin means the distance between nearest data point and hyperplane.

In this case, margin for hyperplane B is high as compared to A and C. If we select a hyperplane having low margin, then there is a chance of misclassification.

3.1.2 PROPERTIES OF MAXIMAL MARGIN METHOD

Theorem 5 The solution (ω^*, b^*) of linearly separable optimization problem exists and the solution should satisfy

1. $\omega^* \neq 0$.

2. there exists some j such that $(\omega^*.x_j) + b^* = 1$ where $j \in \{i|y_i = 1\}$.

3.there exists some k such that $(\omega^*.x_k) + b^* = -1$ where $k \in \{i|y_i = -1\}$.

Proof Firstly we will prove that the solution of convex optimization problem exists. Let $(\omega^{\circ}, b^{\circ})$ be the feasible solution of linearly separable convex optimization problem. So, our problem is equivalent to

Model 3.1

$$\min \frac{1}{2}(\omega.\omega) \; or \; \frac{1}{2}||\omega||^2$$

Subject to

$$(y_i((\omega.x_i) + b)) \geq 1 \quad where\ i = 1,2,\dots,m$$

$$\frac{1}{2}||\omega||^2 \leq \frac{1}{2}||\omega^{\circ}||^2$$

Noted that the feasible region of above problem is the non-empty closed bounded set. Also, as we know that we can get the minimum value of $\frac{1}{2}||\omega||^2$ in non-empty closed bounded set. Hence the solution of above optimization problem exists.

Now we prove

1. $\omega^* \neq 0$

To prove this, we will prove that $(0, b^*)$ is not the solution of above convex optimization problem. If possible, let $(0, b^*)$ is the solution of above convex optimization problem. Then this solution should satisfy $(y_i((\omega.x_i) + b)) \geq 1$ constraint.

Substitute $y_i = 1$ and $(\omega, b) = (0, b^*)$, we get $b^* \geq 1$ and

Substitute $y_i = -1$ and $(\omega, b) = (0, b^*)$, we get $b^* \leq -1$

This implies $1 \leq b^* \leq -1$ which is impossible. Hence our contradiction is wrong. So, $(0, b^*)$ is not the solution of above convex optimization problem.

2. there exists some i such that

$(\omega^*.x_i) + b^* = 1$ where $i \in \{i|y_i = 1\}$

On the contrary, let

$(\omega^*.x_i) + b^* > 1$ where $i \in \{i|y_i = 1\}$

Since (ω^*, b^*) is the feasible solution, so solution (ω^*, b^*) must satisfy the above constraint, that is

$(\omega^*.x_i) + b^* \leq -1$ where $i \in \{i|y_i = -1\}$ (1)

Now let (ω', b') is some point such that

$\omega' = a\omega^*$ and $b' = a(b^* + 1) - 1$ where $a \in (0,1)$

For $a \in (0,1)$, the equation (1) becomes

$(\omega'.x_i) + b' \leq -1$ where $i \in \{i|y_i = -1\}$ (2)

Now, for $i \in \{i|y_i = 1\}$, we have

$$\lim_{a \to 1}[(\omega'.x_i) + b'] = \lim_{a \to 1}[(a\omega^*.x_i) + a(b^* + 1) - 1] = (\omega^*.x_i) + b^* > 1$$

So, there exist $a \in (0,1)$ such that

$(\omega'.x_i) + b' > 1$ where $i \in \{i|y_i = 1\}$ (3)

From equations (2) and (3), it is clear that (ω', b') is the feasible point of optimization problem and the value of objective function is

$\frac{1}{2}||\omega'||^2 = a^2\frac{1}{2}||\omega^*||^2 < \frac{1}{2}||\omega^*||^2$

This implies that (ω^*, b^*) is not a feasible solution. Hence contradiction occur.

So, $(\omega^*.x_i) + b^* = 1$ where $i \in \{i|y_i = 1\}$

3. The part 3 can be proved similarly as part 2.

Theorem 6 The solution of linearly separable optimization problem is unique.

Proof On the contrary, let there are two solutions namely $(\omega_1^*, b_1^*)\ and\ (\omega_2^*, b_2^*)$ of linearly separable optimization problem. Since the solution of optimization problem is unique w.r.t ω^*, so

$$\omega_1^* = \omega_2^* = \omega^*$$

Hence, the two solutions become $(\omega^*, b_1^*)\ and\ (\omega^*, b_2^*)$

$(\omega^*.x_i) + b_1^* \geq 1 \forall i \in \{i|y_i = 1\}$ (4)

$(\omega^*.x_i) + b_1^* = 1 \forall i \in \{i|y_i = 1\}$ (5)

$(\omega^*.x_j) + b_2^* \geq 1 \forall j \in \{j|y_j = 1\}$ (6)

$(\omega^*.x_j) + b_2^* = 1 \forall j \in \{j|y_j = 1\}$ (7)

From (4) and (7), we get

$b_1^* \geq b_2^*$ (8)

From (5) and (6), we get

$b_1^* \leq b_2^*$ (9)

Now, from (8) and (9), we get

$$b_1^* = b_2^*$$

Hence, the solution of linearly separable optimization problem is unique.

3.1.3 Metrics to compare hyperplanes

(I) First version

Consider the equation of the hyperplane $w \cdot x+b=0$. If point $(\mathbf{x}, y)$ is on the hyperplane, then $w \cdot x+b=0$. If the point $(\mathbf{x}, y)$ is not on the hyperplane, the value of $w \cdot x+b$ could be positive or negative. For all the training example points, obtain the point which is closest to the hyperplane. Given a dataset $D=\{(x_i, y_i)|x_i \in R^n, y_i \in \{-1,1\}\}_{i=1}^m$, compute $\beta_i=|w \cdot x_i+b|$ for each training example, and $B=\min |w \cdot x_i+b|$ $for\ i=1, 2\dots.m$

If there are s hyperplanes, each of them will have a $B_i$value and select the hyperplane with the largest $B_i$value.

$H=max\{h_i\ |B_i\}$ $for\ i=1, 2\dots\dots s$

The problem with this metric is that it could fail to distinguish between a good hyperplane and a bad one. It is due to the fact that the absolute value of $w \cdot x+b$ may result into the same value for a correct and an incorrect hyperplane. So, there is a need to adjust this metric.

(II) Second version

As there is an information of the label y. Let's define $f=y(w \cdot x+b)$, and the sign of f will always be positive if the point is correctly classified and will be negative if incorrectly classified. To make it formal, given a dataset D, we compute f for each training example, and F is the smallest f. In literature, F is called the *functional margin* of the dataset.

$F=\min y_i(w \cdot x_i+b)$ for $i=1\dots m$

When comparing hyperplanes, the hyperplane with the *l*argest F will be favourably selected.

This metric also suffers from a problem of scale variant. For example, consider two vectors w_1= *(3,4)* and w_2= *(30,40*). Since they have the same unit vector *u= (0.6,0.8)*, the two vectors w_1 and w_2 represent the same hyperplane. However, when we compute *F*, the one with w_2 will return a larger number than the one with w_1. Thus, there is a need to find a metric which is scale invariant.

(III) Third version

Divide *f* by the length of the vector *w* and define $\gamma = y\ (\frac{w}{||w||} \cdot x + \frac{b}{||w||})$

To make it formal, given a dataset *D*, compute *γ* for each training example, and *M* is the smallest *γ*. In literature, M is called the *geometric margin* of the dataset.

$M = \min y_i \left(\frac{w}{||w||} . x + \frac{b}{||w||}\right)\ i = 1, 2 \ldots \ldots m$

When comparing hyperplanes, the hyperplane with the largest *M* will be favourably selected.

Now objective is to find an optimal hyperplane, i.e., to find the values of w and b of the optimal hyperplane. The problem of finding the values of *w* and *b* is called an optimization problem.

3.1.4 Derivation of SVM optimization problem

To find the values of *w* and *b* of the optimal hyperplane, solve the following optimization problem, with the constraint that the geometric margin of each example should be greater than or equal to *M*:

$$\max_{w,b} M \quad \text{subject to} \quad \gamma_i \geq M \quad , i = 1 \ldots m \tag{10}$$

Also, $M = \frac{F}{||w||}$, so the above problem can be rewritten as:

$$\max_{w,b} M \quad \text{subject to} \quad f_i \geq F\,, \qquad i = 1 \ldots m \tag{11}$$

If *w* and *b* are rescaled, *M* is still maximizing, and the optimization result will not change. Let's rescale *w* and *b* and make *F=1*, the above problem can be rewritten as:

$$\max_{w,b} \frac{1}{||w||} \qquad \text{subject to} \qquad f_i \geq 1, \quad i = 1 \ldots m \tag{12}$$

This maximization problem is equivalent to the following minimization problem:

$$\min_{w,b} ||w|| \quad \text{subject to} \quad f_i \geq 1, \qquad i = 1 \ldots m \tag{13}$$

This minimization problem is equivalent to the following minimization problem:

Model 3.2

$$\min_{w,b} \frac{||w||^2}{2}$$

subject to $y_i(w \cdot x_i + b) - 1 \geq 0$ $, i=1 \dots m$

Note: For a linearly separable problem, the solution to this optimization problem is unique.

The above statement is the SVM optimization problem. It is called a convex quadratic optimization problem.

Note: For a linearly separable problem, the solution to this optimization problem is unique.

The above statement is the SVM optimization problem. It is called a convex quadratic optimization problem.

In case of non-linear data, the minimization problem becomes

Model 3.3

$$\min_{w,b,\xi} \frac{||w||^2}{2} + C\sum \xi_i$$

subject to $y_i((w \cdot \Phi(x_i)) + b) \geq 1 - \xi_i$, $\xi_i \geq 0, i = 1, \cdots, l$,

where $\xi_i \geq 0$, $i = 1, \cdots, l$ are slack variables and $C > 0$ is a penalty parameter.

3.1.5 Relationship between Primal and Dual problem

Let us consider the training data $(x_1, x_2, x_3, \dots\dots\dots, x_n) \in R^n$ with desired output $y_i \in \{-1,1\}$. Let $(\omega . x + b)$ be the equation of hyperplane. Now we want our margin to be maximum. So, this idea leads to following optimization problem.

$$min\ \emptyset(\omega) = \frac{1}{2}(\omega . \omega)\ or\ \frac{1}{2}||\omega||^2$$

subject to

$$f_i(\omega, b) = (y_i(\omega . x_i) + b) - 1 \geq 0 \quad \forall\, i = 1, \dots, n$$

The above problem can be converted into dual in order to get the solution easily. To derive its dual, we introduce the Lagrangian function given by

$$L(\omega, b, \alpha) = \frac{1}{2}||\omega||^2 - [\alpha_1 f_1(\omega, b) + \alpha_2 f_2(\omega, b) + \cdots \dots \dots \dots \alpha_n f_n(\omega, b)]$$

Using the fact that $g(\lambda, \mu)$ is the g.l.b. of p^*, we get $g(\alpha) \leq p^*$

OR $\max\ g(\alpha) = inf_{\omega,b}\ L(\omega, b, \alpha)$

subject to

$$\alpha_i \geq 0$$

As $L(\omega,b,\alpha)$ is a strictly convex quadratic function of ω, so its minimal value is achieved at ω satisfying

$$0 = \frac{\partial L}{\partial \omega} = \omega - (y_1x_1\alpha_1 + y_2x_2\alpha_2 + y_3x_3\alpha_3 + \cdots \ldots \ldots \ldots + y_nx_n\alpha_n)$$

This implies $\omega = y_1x_1\alpha_1 + y_2x_2\alpha_2 + y_3x_3\alpha_3 + \cdots \ldots \ldots \ldots + y_nx_n\alpha_n.$

Substituting the value of ω in expression of Lagrangian function, we get

$inf_{\omega}L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j(x_i^T.x_j)\alpha_i\alpha_j - b\sum_{j=1}^{n}\alpha_jy_j + \sum_{i=1}^{n}\alpha_j$

Model 3.4

$$max\ g(\alpha) = inf_{\omega,b}\ L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j(x_i^T.x_j)\alpha_i\alpha_j + \sum_{i=1}^{n}\alpha_j$$

subject to

$$\sum_{j=1}^{n}\alpha_jy_j = 0\ ;\ \alpha_j \geq 0$$

3.1.6 Inseparable Data

Till now it was very easy to build a hyperplane. But if the dataset is non-linear as plotted below:

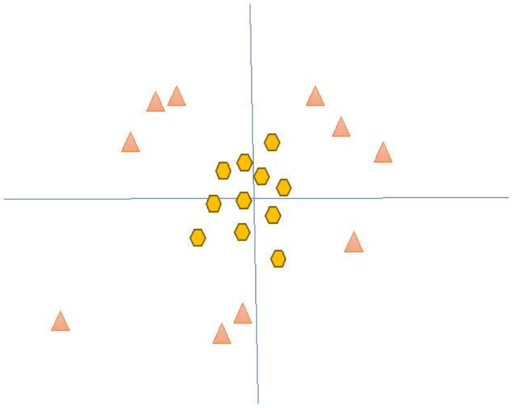

Figure 6

In this case, it is quite clear that there is not a linear decision boundary, i.e., it is difficult to draw a straight line to separate both the classes. In order to classify these two classes, create a new *z* dimension. Let us explain it with an example. Suppose that our two sets of colored balls above are sitting on a sheet and this sheet is lifted suddenly, launching the balls into the air. While the balls are up in the air, use the sheet to separate them. This 'lifting' of the balls represents the mapping of data into a higher dimension. This is known as kernelling.

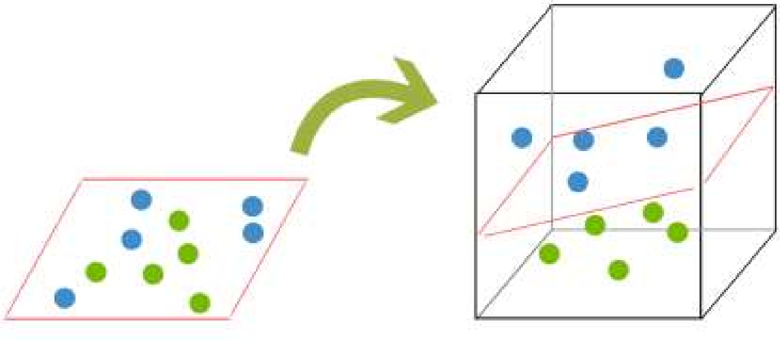

Figure7

(Source: http://blog.aylien.com/wpcontent/uploads/2016/06/tumblr_inline_o9aabehtqP1u37g00_540.png)

Boser, Guyon and Vapnik (1992) suggested a way to create non-linear classifier by applying the kernel trick to maximum margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be non - linear and the transformed space high dimensional; although the classifier is a hyperplane in the transformed feature space, it may be non - linear in the original input space.

It is noted that working in a higher-dimensional feature space increases the generalization error of SVM, although given enough samples the algorithm still performs well.

3.1.6.1 Kernel Function

Kernel is a mathematical function wherein we transfer two-dimensional points into n-dimensional space. It depends upon what the data looks like. It entails transforming linearly inseparable data like (Figure 3) to linearly separable ones (Figure 2). The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

So, for a general n-dimensional problem, the kernel is just a map, say Φ that transform an n-dimensional vector x into another m-dimensional vector x in Euclidian space R^m. Thus, the map can be expressed as

Φ:$R^n \rightarrow R^m$

i.e., $x = \left([x]_1, [x]_{2,\ldots\ldots\ldots\ldots,}[x]_n\right)^T \rightarrow \left([x]_1, [x]_{2,\ldots\ldots\ldots\ldots,}\right)^T = \Phi(x)$

Suppose the original training set is given by

$$T = \{(x_1, y_1), (x_2, y_2), \dots\dots\dots.. (x_l, y_l)\},$$

where $x_i \in R^n$, $y_i \in Y = \{-1,1\}$ $, i = 1,\cdots ,l$. Under the above map, the training set T is transformed to $T_\Phi = \{(x_1, y_1), (x_2, y_2), \dots\dots\dots.. (x_l, y_l)\}$

where $x_i = \Phi\ (x_i) \in R^m$, $y_i \in Y = \{-1,1\}$, $i = 1,\cdots ,l$. Next step is to compute the linear separating hyperplane $(w^* \cdot x) + b^* = 0$ in this space, thus deduce the separating hypersurface $(w^* \cdot \Phi(x)) + b^* = 0$ and the decision function $f(x) = sgn((w^* \cdot x) + b^*) = sgn((w^* \cdot \Phi(x)) + b^*)$ in the original space R^n. Note that the distance between the two hyperplanes

$$(w \cdot x) + b = 1 \text{ and } (w \cdot x) + b = -1$$

can still be represented by $\frac{2}{||w||}$.

In order to relax the requirement to separate all of the inputs correctly, allow the existence of training points that violate the constraints $y_i((w \cdot x_i)+b) \geq 1$ by introducing slack variables $\xi_i \geq 0$, $i = 1,\cdots ,l$, then yielding loose constraints $y_i((w \cdot x_i)+b) \geq 1-\xi_i$, $i = 1,\cdots ,l$.

On the other hand, in order to make the above violation as little as possible, avoid making ξ_i too large by superimposing a penalty upon them in the objective function. For instance, we can add a term $\sum \xi_i$ to the objective function resulting in changing the primal problem (5) into in the following problem

$$\min_{w,b,\xi} \frac{||w||^2}{2} + C\sum \xi_i \quad \text{subject to } y_i((w \cdot x_i)+b) \geq 1-\xi_i, \quad i = 1,\cdots ,l,\ \xi_i \geq 0, i = 1,\cdots ,l \qquad (14)$$

where $\xi = (\xi_1, \xi_2, \dots\dots\dots. \xi_l)^T$ and $C > 0$ is a penalty parameter. The two terms in the objective function indicate that not only $||w||^2$ minimizes (margin maximizes), but also $\sum_{i=1}^{l} \xi_i$ minimizes simultaneously which is a measurement of violation of the constraints $y_i((w \cdot x_i)+b) \geq 1$, $i = 1,\cdots ,l$. The parameter C determines the weighting between the two terms.

Therefore, we can also construct the primal problem corresponding to the problem (14)

$$\min_{w,b,\xi} \frac{||w||^2}{2} + C\sum \xi_i \quad \text{subject to } y_i((w \cdot \Phi(x_i)) + b) \geq 1-\xi_i, \quad i = 1,\cdots ,l \xi_i \geq 0, i = 1,\cdots ,l$$

Mathematical definition: ***K*** *(x, y) = <f(x), f(y)>. Here K is the kernel function, x, y are n dimensional inputs. f is a map from n-dimension to m-dimension space. <x, y> denotes the dot product. Usually m is much larger than n.*

For example, take the kernel as $x^2 + y^2 = z^2$, which is an equation of circle.

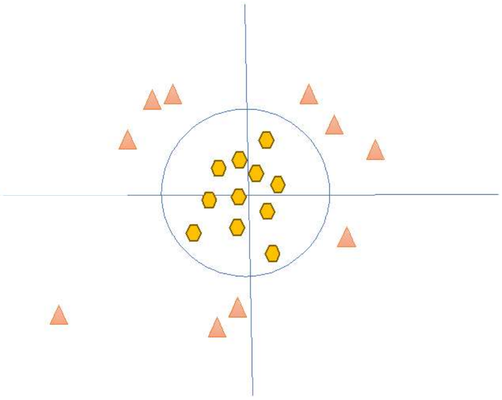

Figure 8

Here, decision boundary is a circumference of a particular radius, which separates both classes using SVM. A support vector machine only takes care of finding the decision boundary.

3.1.6.2 Examples of SVM kernels:

Some common kernels used with SVMs and their uses are given below:

1. **Polynomial kernel**

$K(x_i, x_j) = (x_i . x_j + 1)^d$

where d is the degree of the polynomial.

It is used in image processing.

2. **Gaussian kernel**

$$k(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$$

where $||x-y||^2$ is the squared Euclidean distance between two feature vectors and $\sigma$is a free parameter.

It is a general-purpose kernel; used when there is no prior knowledge about the data.

3. **Gaussian radial basis function (RBF)**

$k(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$ for $\gamma > 0$

where $\gamma = \frac{1}{2\sigma^2}$. It is a general-purpose kernel; used when there is no prior knowledge about the data.

4. **Laplace RBF kernel**

$$k(x,y) = \exp\left(-\frac{||x-y||}{\sigma}\right)$$

It is general-purpose kernel; used when there is no prior knowledge about the data.

5. **Hyperbolic tangent kernel**

$k(x_i, x_j) = \tanh(kx_i . x_j + c)$ for some (not every) $k>0$ and $c<0$.

It is used in neural networks.

6. **Sigmoid kernel**

$$k(x,y) = \tanh(\alpha x^T y + c)$$

It is used as the proxy for neural networks

7. **Bessel function of the first kind Kernel**

$$k(x,y) = \frac{\mathrm{J}_{\mathrm{v}+1}(\sigma||\mathrm{x}-\mathrm{y}||)}{||\mathrm{x}-\mathrm{y}||^{-\mathrm{n}(\mathrm{v}+1)}}$$

where J is the Bessel function of first kind.

It is used to remove the cross term in mathematical functions.

8. **ANOVA radial basis kernel**

$$k(x,y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$$

It is used in regression problems.

9. **Linear splines kernel in one-dimension**

$$k(x,y) = 1 + xy + xy\min(x,y) - \frac{x+y}{2}\min(x,y)^2 + \frac{1}{3}\min(x,y)^3$$

It is useful when dealing with large sparse data vectors. It is often used in text categorization. It also performs well in regression problems.

CHAPTER 4

Proposed hybrid approach of Interval DEA and SVM in optimistic & pessimistic environment

Consider a performance evaluation problem to measure the performance of *n* DMU's. Each DMU use m inputs and give s outputs out of which s_1 are desirable(good) and s_2 are undesirable(bad). Let us suppose that the input and output data is in the form of closed intervals. Let $\hat{X} \in R^{n\times m}$, $\hat{Y}^g \in R^{n\times s_1}$ and $\hat{Y}^b \in R^{n\times s_2}$ be the matrices having observed input, desirable output and undesirable output data for each DMU respectively. Let the i^{th}interval input of DMU_k is $[x_{ik}^L, x_{ik}^U]$. Further suppose the r^{th} desirable(good) and p^{th}undesirable(bad) interval output of DMU_k is $[y_{rk}^{gL}, y_{rk}^{gU}]$ and $[y_{pk}^{bL}, y_{pk}^{bU}]$ respectively. Use y_{rk}^g as the r^{th} desirable output; y_{pk}^b as the p^{th} undesirable output and x_{ik} as the i^{th} input for the *k*th DMU. So, the efficiency of kth DMU is given by:

$$\hat{E}_k = \frac{\left(\sum_{r=1}^{s_1} u_{rk}^g \hat{y}_{rk}^g - \sum_{p=1}^{s_2} u_{pk}^b \hat{y}_{pk}^b\right)}{\sum_{i=1}^{m} v_{ik} \hat{x}_{ik}} \tag{15}$$

where u_{rk}^g,u_{pk}^b, v_{ik} are the weights of r^{th} desirable(good) interval output, p^{th}undesirable(bad) interval output and i^{th} interval input respectively.

Interval arithmetic (See Section 1.1) is used to calculate the interval efficiency which is given by:

$$\hat{E}_k = \frac{\sum_{r=1}^{s_1} u_{rk}^g \left[y_{rk}^{gL}, y_{rk}^{gU}\right] - \sum_{p=1}^{s_2} u_{pk}^b \left[y_{pk}^{bL}, y_{pk}^{bU}\right]}{\sum_{i=1}^{m} v_{ik}\left[x_{ik}^L, x_{ik}^U\right]}$$

$$= \left[\frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bU}}{\sum_{i=1}^{m} v_{ik} x_{ik}^U}, \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bL}}{\sum_{i=1}^{m} v_{ik} x_{ik}^L}\right]$$

$$= [E_k^L, E_k^U]$$

Suppose that $E_k^L \geq 0$ and $E_k^U \leq 1 \; \forall \; k = 1,2, \dots, K, \dots, n$. Then $\hat{E}_k = [E_k^L, E_k^U] \subseteq [0,1]$.

4.1 OPTIMISTIC APPROACH-Best Relative Efficiency

The upper bound efficiency E_k^U of DMU_k in interval efficiency $[E_k^L, E_k^U]$ is known as the optimistic efficiency of the DMU_k in optimistic environment and it is measured by the following mathematical model:

Model 4.1

$$maxE_k^U = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bL}}{\sum_{i=1}^{m} v_{ik} x_{ik}^L}$$

subject to
$$E_j^U = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bL}}{\sum_{i=1}^{m} v_{ik} x_{ij}^L} \leq 1 \ \forall j,$$

$$E_j^L = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bU}}{\sum_{i=1}^{m} v_{ik} x_{ij}^U} \geq 0 \ \forall j,$$

$$u_{rk}^g \geq \epsilon \ \forall\, r = 1,2,\dots,s_1; \quad u_{pk}^b \geq \epsilon \ \forall\, p = 1,2,\dots,s_2; \quad v_{ik} \geq \epsilon \ \forall\, i = 1,2,\dots,m, \epsilon > 0$$

where u_{rk}^g,u_{pk}^b, v_{ik} are the weights of r^{th} desirable(good) output, p^{th}undesirable(bad) output and i^{th} input respectively.

Model 4.1 can be transformed to the following linear programming problem by using Charnes-Cooper transformation (Cooper et al., 2007).

Model 4.2

$$maxE_k^U = \sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bL}$$

subject to
$$\sum_{i=1}^{m} v_{ik} x_{ik}^L = 1,$$

$$\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bL} - \sum_{i=1}^{m} v_{ik} x_{ij}^L \leq 0 \ \forall j,$$

$$\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bU} \geq 0 \ \ \forall j,$$

$$u_{rk}^g \geq \epsilon \ \forall\, r = 1,2,\dots,s_1; \quad u_{pk}^b \geq \epsilon \ \forall\, p = 1,2,\dots,s_2; \quad v_{ik} \geq \epsilon \ \forall\, i = 1,2,\dots,m, \epsilon > 0$$

Model 4.2 represents the optimistic point of view and is used to calculate the upper bound efficiency E_k^U of interval efficiency $[E_k^L, E_k^U]$ of DMU_k.

Definition 9 A DMU_k is said to be efficient in optimistic environment if ${E_k^U}^* = 1$, where ${E_k^U}^*$ is the optimal objective function value of Model 4.2.

4.2 PESSIMISTIC APPROACH-Worst Relative Efficiency

The lower bound efficiency E_k^L of DMU_k in interval efficiency $[E_k^L, E_k^U]$ is known as the pessimistic efficiency of the DMU_k in pessimistic environment and it is measured by the following mathematical model:

Model 4.3

$$maxE_k^L = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bU}}{\sum_{i=1}^{m} v_{ik} x_{ik}^U}$$

subject to $$E_j^U = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bL}}{\sum_{i=1}^{m} v_{ik} x_{ij}^L} \leq 1 \; \forall j,$$

$$E_j^L = \frac{\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bU}}{\sum_{i=1}^{m} v_{ik} x_{ij}^U} \geq 0 \; \forall j,$$

$$u_{rk}^g \geq \epsilon \;\; \forall \, r = 1,2,\ldots,s_1; \;\; u_{pk}^b \geq \epsilon \; \forall \, p = 1,2,\ldots,s_2; \;\; v_{ik} \geq \epsilon \; \forall \, i = 1,2,\ldots,m, \epsilon > 0$$

where u_{rk}^g,u_{pk}^b, v_{ik} are the weights of r^{th} desirable(good) output, p^{th}undesirable(bad) output and i^{th} input respectively.

Model 4.3 can be transformed to the following linear programming problem by using Charnes-Cooper transformation (Cooper et al., 2007).

Model 4.4

$$maxE_k^L = \sum_{r=1}^{s_1} u_{rk}^g y_{rk}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pk}^{bU}$$

subject to $$\sum_{i=1}^{m} v_{ik} x_{ik}^U = 1,$$

$$\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gU} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bL} - \sum_{i=1}^{m} v_{ik} x_{ij}^L \leq 0 \; \forall j,$$

$$\sum_{r=1}^{s_1} u_{rk}^g y_{rj}^{gL} - \sum_{p=1}^{s_2} u_{pk}^b y_{pj}^{bU} \geq 0 \; \forall j,$$

$$u_{rk}^g \geq \epsilon \;\; \forall \, r = 1,2,\ldots,s_1; \;\; u_{pk}^b \geq \epsilon \; \forall \, p = 1,2,\ldots,s_2; \;\; v_{ik} \geq \epsilon \; \forall \, i = 1,2,\ldots,m, \epsilon > 0$$

Model 4.4 represents the optimistic point of view and is used to calculate the lower bound efficiency E_k^L of interval efficiency $[E_k^L, E_k^U]$ of DMU_k.

Definition 10 A DMU_k is said to be efficient in pessimistic environment if ${E_k^L}^* = 1$, where ${E_k^L}^*$ is the optimal objective function value of Model 4.4.

Theorem 7 (Puri and Yadav, 2017) If $E_k^{U^*}$ and $E_k^{L^*}$ are the optimum objective function values of Model 4.2 and Model 4.4 respectively, then $E_k^{L^*} \leq E_k^{U^*}$. The equality occurs only if all the input data and desirable and undesirable output data degenerate from imprecise data to crisp data.

Proof Suppose that $(u_{1k}^{g^*}, u_{2k}^{g^*}, \ldots, u_{s_1k}^{g^*}; u_{1k}^{b^*}, u_{2k}^{b^*}, \ldots, u_{s_2k}^{b^*}; v_{1k}^*, v_{2k}^*, \ldots, v_{mk}^*)$ be the optimal solution of Model 4.4.

consider $\delta_k = \sum_{i=1}^{m} v_{ik}^* x_{ik}^L$, $U_{rk}^g = \frac{u_{rk}^{g^*}}{\delta_k} \forall r$, $U_{rk}^b = \frac{u_{pk}^{b^*}}{\delta_k} \; \forall p$ and $V_{ik} = \frac{v_{ik}^*}{\delta_k} \; \forall i.$

Now, $\delta_k = \sum_{i=1}^{m} v_{ik}^* x_{ik}^L \;\; \leq \;\; \sum_{i=1}^{m} v_{ik}^* x_{ik}^U \; = \; 1,$

$$\sum_{i=1}^{m} V_{ik} x_{ik}^{L} = \sum_{i=1}^{m} \frac{v_{ik}^{*}}{\delta_k} x_{ik}^{L} = \frac{1}{\delta_k} \sum_{i=1}^{m} v_{ik}^{*} x_{ik}^{L} = 1, \tag{16}$$

$$\begin{aligned}
&\sum_{r=1}^{s_1} U_{rk}^{g} y_{rj}^{gU} - \sum_{p=1}^{s_2} U_{pk}^{b} y_{pj}^{bL} - \sum_{i=1}^{m} V_{ik} x_{ij}^{L} \\
&= \frac{1}{\delta_k} \left(\sum_{r=1}^{s_1} u_{rk}^{g*} y_{rj}^{gU} - \sum_{p=1}^{s_2} u_{pk}^{b*} y_{pj}^{bL} - \sum_{i=1}^{m} v_{ik}^{*} x_{ij}^{L} \right) \le 0, \forall j
\end{aligned} \tag{17}$$

$$\begin{aligned}
&\sum_{r=1}^{s_1} U_{rk}^{g} y_{rj}^{gL} - \sum_{p=1}^{s_2} U_{pk}^{b} y_{pj}^{bU} \\
&= \frac{1}{\delta_k} \left(\sum_{r=1}^{s_1} u_{rk}^{g*} y_{rj}^{gL} - \sum_{p=1}^{s_2} u_{pk}^{b*} y_{pj}^{bU} \right) \ge 0, \forall j
\end{aligned} \tag{18}$$

$$\left.\begin{aligned}
U_{rk}^{g} &= \frac{u_{rk}^{g*}}{\delta_k} \ge \frac{\varepsilon}{\delta_k} \ge \varepsilon \quad \forall r, \\
U_{pk}^{b} &= \frac{u_{pk}^{b*}}{\delta_k} \ge \frac{\varepsilon}{\delta_k} \ge \varepsilon \quad \forall p, \\
V_{ik} &= \frac{v_{ik}^{*}}{\delta_k} \ge \frac{\varepsilon}{\delta_k} \ge \varepsilon \quad \forall i,
\end{aligned}\right\} \tag{19}$$

Equations (17), (18), (19) and (20) implies $(U_{1k}^{g}, U_{2k}^{g}, \ldots, U_{s_1 k}^{g}; U_{1k}^{b}, U_{2k}^{b}, \ldots, U_{s_2 k}^{b}; V_{1k}, V_{2k}, \ldots, V_{mk})$ is a feasible solution of Model 4.2.

Therefore, $\sum_{r=1}^{s_1} U_{rk}^{g} y_{rk}^{gU} - \sum_{p=1}^{s_2} U_{pk}^{b} y_{pk}^{bL} \le E_k^{U*}$.

Now,
$$\begin{aligned}
E_k^{L*} &= \sum_{r=1}^{s_1} u_{rk}^{g*} y_{rk}^{gL} - \sum_{p=1}^{s_2} u_{pk}^{b*} y_{pk}^{bU} \\
&\le \sum_{r=1}^{s_1} u_{rk}^{g*} y_{rk}^{gU} - \sum_{p=1}^{s_2} u_{pk}^{b*} y_{pk}^{bL} \\
&= \delta_k \left(\sum_{r=1}^{s_1} U_{rk}^{g} y_{rk}^{gU} - \sum_{p=1}^{s_2} U_{pk}^{b} y_{pk}^{bL} \right) \\
&\le \delta_k \, E_k^{U*} \le E_k^{U*}
\end{aligned}$$

Hence, $E_k^{L*} \le E_k^{U*}$.

The above equality holds only when $y_{rk}^{g\,L} = y_{rk}^{g\,U}$ $\forall r$, $y_{pk}^{b\,L} = y_{pk}^{b\,U}$ $\forall p$ and $x_{ik}^{L} = x_{ik}^{U}$ $\forall i$.

Note: Ordinal data can be converted into the interval data (Wang et. al., 2005; Azizi, 2014) and can easily be use into Models-4.2 and 4.4.

4.3 IDEA- SVM approach in OPTIMISTIC ENVIRONMENT

Suppose that data observed is linearly separable. This input and output data and the efficiency data obtained from the linear optimistic Model 4.2 is used to calculate the optimal hyperplane. Since in optimistic approach, we use y_{rj}^{gL} as the r^{th} desirable output; y_{pj}^{bU} as the p^{th} undesirable output and x_{ij}^{U}as the i^{th} input of another j DMU's. So, we form of a vector $X_j^o = (x_{1j}^L, x_{2j}^L, \dots, x_{mj}^L, y_{1j}^{gL}, y_{2j}^{gL}, \dots, y_{s_2j}^{gL}, y_{1j}^{gU}, y_{2j}^{gU}, \dots, y_{s_1j}^{gU})$ $\forall j = 1,2,\dots,n$.The desired efficiency $y_j^o \in \{-1,1\} \, \forall j = 1,2,\dots,n.$ Let $\omega.x_i + b = 0$ is the equation of linear hyperplane which segregate these two classes where ω is weight vector and b represents the bias. The point above or on the hyperplane will be classified as class +1 or efficient, and the point below the hyperplane will be classified as class -1 or inefficient. Now we use following optimization problem to calculate the optimal, linear hyperplane for linearly separable data.

Model 4.5

$$min\, f(\omega) = \frac{1}{2}||\omega||^2$$

subject to $y_j^o(X_j^o.\omega + b) \geq 1 \quad \forall j = 1,2,\dots,n.$

The above optimization problem can be converted into dual problem to get the required solution. Using Model 3.4, we get;

Model 4.6

$$max\, g(\alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} \alpha_j \alpha_i y_j^o y_i^o {X_j^o}^T.X_i^o$$

$$\text{subject to} \qquad \sum_{j=1}^{n} \alpha_j y_j^o = 0\, ;\; \alpha_j \geq 0 \;\; \forall j = 1,2,\dots,n$$

On solving this model, we get the optimal value of Lagrange multipliers $\alpha_j \, \forall i = 1,2,\dots\dots,n$ denoted as $\alpha_j^* \, \forall i = 1,2,\dots\dots,n.$ So, using these values of multipliers, we get the optimal weight vector ω and bias value b as follows.

$$\omega = \sum_{j=1}^{n} \alpha_j^* y_j^o X_j^o \qquad \forall j = 1,2,\dots,n$$

$$b = y_i^o - y_i^o \alpha_j^* {X_j^o}^T.X_i^o \qquad \forall j = 1,2,\dots,n$$

For non-linear data, use slack variables $\epsilon_i \geq 0, i = 1,2,\dots,n$ in the constraints. In this case, the following models are used calculate optimal hyperplane from optimistic point of view;

Model 4.7

$$min\, f(\omega) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\epsilon_i$$

subject to $y_j^o\left(X_j^o.\omega + b\right) \geq 1 - \epsilon_i, \quad \forall\, j = 1,2, \dots, n\ and\ \epsilon_i \geq 0, i = 1,2, \dots, n$

where *C* is user choice non-negative constant.

The above optimization problem can be converted into dual problem to get the required solution. Since the data is linearly inseparable, so RBF kernel is used to transform the linearly inseparable data to separable one. Using Model 3.4, we get;

Model 4.8

$$max\, g(\alpha) = \sum_{j=1}^{n}\alpha_j - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_j\alpha_i y_j^o y_i^o K(X_j^o, X_i^o)$$

subject to $\sum_{j=1}^{n}\alpha_j y_j^o = 0\ ;\ 0 \leq \alpha_j \leq C\ \ \forall\, j = 1,2, \dots, n$

On solving this model, we get the optimal value of Lagrange multipliers $\alpha_j\ \forall\, i = 1,2, \dots\dots, n$ denoted as $\alpha_j^*\ \forall\, i = 1,2, \dots\dots, n$. So, using these values of multipliers, we get the optimal weight vector ω and bias value *b* as follows.

$\omega = \sum_{j=1}^{n}\alpha_j^* y_j^o X_j^o \qquad \forall\, j = 1,2, \dots, n$

$b = y_i^o - y_i^o\alpha_j^* X_j^{o^T}.X_i^o \qquad \forall\, j = 1,2, \dots, n$

4.4 IDEA- SVM approach in PESSIMISTIC ENVIRONMENT

The input and output data and the efficiency data obtained from the linear pessimistic Model 4.4 is used to calculate the optimal hyperplane. Since in pessimistic approach, we use y_{rj}^{gU} as the r^{th} desirable output; y_{pj}^{bL} as the p^{th} undesirable output and x_{ij}^Las the i^{th} input of another j DMU's. So, we form of a vector $X_j^p = (x_{1j}^U, x_{2j}^U, \dots, x_{mj}^U, y_{1j}^{gL}, y_{2j}^{gL}, \dots, y_{s_1j}^{gL}, y_{1j}^{gU}, y_{2j}^{gU}, \dots, y_{s_2j}^{gU}) \quad \forall\, j = 1,2, \dots, n$.The desired efficiency $y_j^p \in \{-1,1\}\ \forall\, j = 1,2, \dots, n$. Let $\omega.x_i + b = 0$ is the equation of linear hyperplane which segregate these two classes where ω is weight vector and *b* represents the bias. The point above or on the hyperplane will be classified as class +1 or inefficient, and the point below the hyperplane will be classified as class -1 or non-inefficient. Now we use following optimization problem to calculate the optimal, linear hyperplane.

Model 4.9

$$min\, f(\omega) = \frac{1}{2}||\omega||^2$$

subject to

$$y_j^p\left(X_j^p.\omega + b\right) \geq 1 \quad \forall\, j = 1,2,\ldots,n.$$

The above optimization problem can be converted into dual problem to get the required solution. Using Model 3.4, we get

Model 4.10

$$max\, g(\alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} \alpha_j \alpha_i y_j^p y_i^p X_j^{p^T}.X_i^p$$

subject to $\quad \sum_{j=1}^{n} \alpha_j y_j^p = 0\,;\ \alpha_j \geq 0\ \ \forall\, j = 1,2,\ldots,n$

On solving this model, we get the optimal value of Lagrange multipliers $\alpha_j\ \forall\, i = 1,2,\ldots\ldots,n$ denoted as $\alpha_j^*\ \forall\, i = 1,2,\ldots\ldots,n$. So, using these values of multipliers, we get the optimal weight vector ω and bias value *b* as follows.

$\omega = \sum_{j=1}^{n} \alpha_j^* y_j^p X_j^p \qquad \forall\, j = 1,2,\ldots,n$

$b = y_i^o - y_i^p \alpha_j^* X_j^{p^T}.X_i^p \qquad \forall\, j = 1,2,\ldots,n$

For non-linear data, the following models are used to calculate optimal hyperplanes from pessimistic point of view;

Model 4.11

$$min\, f(\omega) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n} \epsilon_i$$

subject to

$$y_j^p\left(X_j^p.\omega + b\right) \geq 1 - \epsilon_i, \quad \forall\, j = 1,2,\ldots,n\ and\ \epsilon_i \geq 0, i = 1,2,\ldots,n$$

where *C* is user choice non-negative constant.

The above optimization problem can be converted into dual problem to get the required solution. Since the data is linearly inseparable, so RBF kernel is used to transform the linearly inseparable data to separable one. Using Model 3.4, we get

Model 4.12

$$max\, g(\alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} \alpha_j \alpha_i y_j^p y_i^p K(X_j^p, X_i^p)$$

subject to $\quad \sum_{j=1}^{n} \alpha_j y_j^p = 0\,;\ 0 \leq \alpha_j \leq C\ \ \forall\, j = 1,2,\ldots,n$

On solving this model, we get the optimal value of Lagrange multipliers $\alpha_j\ \forall\, i = 1,2,\ldots\ldots,n$ denoted as $\alpha_j^*\ \forall\, i = 1,2,\ldots\ldots,n$. So, using these values of multipliers, we get the optimal weight vector ω and bias value *b* as follows.

$\omega = \sum_{j=1}^{n} \alpha_j^* y_j^p X_j^p \quad \forall\, j = 1,2,\dots,n$ and $b = y_i^o - y_i^p \alpha_j^* X_j^{p^T} . X_i^p \quad \forall\, j = 1,2,\dots,n.$

4.5 Numerical Illustration

To illustrate the IDEA-SVM approach in both optimistic and pessimistic environment, consider the example from Puri and Yadav (2017) which contains 18 DMUs having two inputs, two desirable outputs and one undesirable output. The input & output data and the interval efficiency calculated from optimistic and pessimistic prospective is given below in the form of a table. Further, by using the proposed IDEA-SVM approach, we calculated the parameters using Models 4.8 and 4.12 to generate the hyperplane in optimistic and pessimistic environments by using the data listed in Tables 2 and 3 respectively. The results of the proposed approach are presented in Table 4.

Table 1 Input and output data for 18 DMUs with DEA efficiency results

DMU	Input 1	Input 2	Input 2 (interval estimation)	Desirable output	Undesirable output	Efficiency interval (Puri and Yadav, 2017 approach)
1	253	5	[0.01574, 0.22917]	[50, 65]	1	[0.653, 1]
2	268	10	[0.02773, 0.40388]	[60, 70]	5.3	[0.197, 0.255]
3	259	3	[0.01254, 0.1827]	[40, 50]	4.6	[0.136, 0.253]
4	180	6	[0.01762, 0.25668]	[100, 160]	30	[0.488, 0.888]
5	257	4	[0.01405, 0.20462]	[45, 55]	30	[0.152, 0.267]
6	248	2	[0.0112, 0.16312]	[85, 115]	30	[0.3, 0.618]
7	272	8	[0.02211, 0.32197]	[70, 95]	30	[0.225, 0.378]
8	330	11	[0.03106, 0.45235]	[100, 180]	13.8	[0.267, 0.556]
9	327	9	[0.02476, 0.36061]	[90, 120]	4	[0.334, 0.586]
10	330	7	[0.01974, 0.28748]	[50, 80]	30	[0.131, 0.294]
11	321	16	[0.05474, 0.79719]	[250, 300]	26.4	[0.686, 0.824]
12	329	14	[0.04363, 0.63552]	[100, 150]	25.8	[0.266, 0.401]
13	281	15	[0.04887, 0.71178]	[80, 120]	25.8	[0.249, 0.375]
14	309	13	[0.03896, 0.56743]	[200, 350]	21.9	[0.57, 1]
15	291	12	[0.03479, 0.50663]	[40, 55]	9	[0.121, 0.171]
16	334	17	[0.0613, 0.89286]	[75, 85]	7	[0.198, 0.225]
17	249	1	[0.01, 0.14564]	[90, 180]	6.3	[0.319, 1]
18	216	18	[0.06866, 1]	[90, 150]	28.8	[0.366, 0.611]

Source: Puri and Yadav (2017)

Table 2 Data used in Optimistic approach

DMUs	Data used to generate hyperplane in Optimistic approach $(x_{1j}^{L}, x_{2j}^{L}, \dots, x_{mj}^{L}, y_{1j}^{gL}, y_{2j}^{gL}, \dots, y_{s_2j}^{gL}, y_{1j}^{gU}, y_{2j}^{gU}, \dots, y_{s_1j}^{gU})$	Efficiency in the form of +1 or -1
1.	(253, 0.01574, 1, 65)	1
2.	(268, 0.02773, 5.3, 70)	-1
3.	(259, 0.01254, 4.6, 50)	-1
4.	(180, 0.01762, 30, 160)	-1
5.	(257, 0.01405, 30, 55)	-1
6.	(248, 0.0112, 30, 115)	-1
7.	272, 0.02211, 30, 95)	-1
8.	(330, 0.03106, 13.8, 180)	-1
9.	(327, 0.02476, 4, 120)	-1
10.	(330, 0.01974, 30, 80)	-1
11.	(321, 0.05474, 26.4, 300)	-1
12.	(329, 0.04363, 25.8,150)	-1
13.	(281, 0.04887, 25.8, 120)	-1
14.	(309, 0.03896, 21.9, 350)	1
15.	(291, 0.03479, 9, 55)	-1
16.	(334, 0.0613, 7, 85)	-1
17.	(249, 0.01, 6.3, 80)	1
18.	(216, 0.06866, 28.8, 150)	-1

Table 3 Data used in Pessimistic approach

DMUs	Data used to generate hyperplane in Pessimistic approach $(x_{1j}^{U}, x_{2j}^{U}, \dots, x_{mj}^{U}, y_{1j}^{gU}, y_{2j}^{gU}, \dots, y_{s_2j}^{gU}, y_{1j}^{gL}, y_{2j}^{gL}, \dots, y_{s_1j}^{gL})$	Efficiency in the form of +1 or -1
1.	(253, 0.22917, 50, 1)	1
2.	(268, 0.40388, 60, 5.3)	-1
3.	(259, 0.1827, 40, 4.6)	-1
4.	(180, 0.25668, 100, 30)	-1
5.	(257, 0.20462,45, 30)	-1
6.	(248, 0.16312, 85, 30)	-1
7.	(272, 0.32197, 70, 30)	-1
8.	(330, 0.45235, 100, 13.8)	-1
9.	(327, 0.36061, 90, 4)	-1
10.	(330, 0.28748, 50, 30)	-1
11.	(321, 0.79719, 250, 26.4)	-1
12.	(329, 0.63552, 100, 25.8)	-1
13.	(281, 0.71178, 80, 25.8)	-1
14.	(309, 0.56743, 200, 21.9)	1
15.	(291, 0.50663, 40, 9)	-1
16.	(334, 0.89286, 75, 7)	-1
17.	(249, 0.14564, 90, 6.3)	1
18.	(216, 1, 90, 28.8)	-1

Table 4 Results of the proposed IDEA-SVM approach in Optimistic and Pessimistic Environments

Parameters		Optimistic	Pessimistic
Lagrange Multipliers	α_1	0.023	0.59
	α_2	0.061	0.039
	α_3	0.118	0.244
	α_4	0.406	0.028
	α_5	0.008	0.165
	α_6	0.737	0.675
	α_7	0.136	0.008
	α_8	0.386	0.355
	α_9	0.723	0.414
	α_{10}	0.257	0.347
	α_{11}	0.079	0.953
	α_{12}	0.381	0.012
	α_{13}	0.17	0.524
	α_{14}	0.58	0.315
	α_{15}	0.075	0.012
	α_{16}	0.137	0.091
	α_{17}	0.12	0.186
	α_{18}	0.716	0.045
Weight vector	ω	$\begin{pmatrix} -1.0632.813 \\ -1.39432649 \\ -721.099800 \\ -5551.52000 \end{pmatrix}^T$	$\begin{pmatrix} -7755.725 \\ -16.14387515 \\ -2867.655 \\ -513.6618 \end{pmatrix}^T$
Bias value	b_1	-1386678543.5940723	-1228109572.6855645
	b_2	+1386678543.5940723	+1228109572.6855645

4.6 Conclusion

DEA is based on linear programming approach and is used to measure the relative performance of similar organizations. However, DEA technique is restricted to crisp input and output data for its successful implementation. However, to handle imprecision in the data like interval data, DEA has been extended by various researchers. DEA can help the decision maker to rank the set of similar DMUs and further to estimate input and output targets to improve the inefficient DMUs. However, in order to predict failures of the organizations using the past data has not been discussed using DEA. Moreover, machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. SVM is a supervised machine learning algorithm that is used for the classification, regression or other purposes like outlier detection. It is a fast and dependable classification algorithm that performs very well with a limited amount of data. In this work, we have proposed the integrated/hybrid approach using interval DEA (IDEA) and SVM technique to generate hyperplanes in optimistic and pessimistic environments. The proposed approach is named as IDEA-SVM approach. It can be used by the decision makers to predict the accuracy of organization failure. For validation, a numerical illustration has been presented using 18 DMUs.

Appendix A

Python code

```
import math as m
import numpy as np
import random as ran

iteration=1000
sigma=2
C=1
x=np.array([[253, 0.01574, 1, 65],[268, 0.02773, 5.3, 70],[259, 0.01254, 4.6, 50],[180, 0.01762, 30, 160],[257, 0.01405, 30, 55],[248, 0.0112, 30, 115],[272, 0.02211, 30, 95],[330, 0.03106, 13.8, 180],[327, 0.02476, 4, 120],[330, 0.01974, 30, 80],[321, 0.05474, 26.4, 300],[329, 0.04363, 25.8, 150],[281, 0.04887, 25.8, 120],[309, 0.03896, 21.9, 350],[291, 0.03479, 9, 55],[334, 0.0613, 7, 85],[249, 0.01, 6.3, 180],[216, 0.06866, 28.8, 150]])
n=18
eudistance=0
bestY=[ ]
bestAlpha=[ ]
bestFitnessValue=9999999999


def k(x1,x2):
   x1=np.array(x1)
   x2=np.array(x2)
   eudistance=np.sqrt(np.sum((x1-x2)**2))
   y=m.exp(-((eudistance**2)/(2*sigma*sigma)))
   return y

def s(a):
   w=0
```

```
    for i in range(n):
        w=w+a[i]
    return w

def r(a,y):
    w=0
    for i in range(n):
        for j in range(n):
            z=a[i]*a[j]*y[i]*y[j] * k(x[i],x[j])
            w=w+z
    return w

def FitnessFunction(a,y):
    return round(s(a) - 0.5 * r(a,y),3)



# Saving Result
fp=open("result.csv","w");
fp.write("Iteration,FitnessValue,y,Alpha\n")

i=0
while i <= iteration:
    #y=[ int("".join(ran.sample(['-1','1'],1))) for j in range(n)]
    y=[1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1,-1,-1,1,-1]
    alpha = [float("".join(str(round(ran.random(),3)))) for j in range(n)]



    w=0
    for j in range(n):
        w=w+alpha[j]*y[j]



    if w==0:
```

```
        continue

    fitnessValue = FitnessFunction(alpha,y)

    if fitnessValue < bestFitnessValue:
        bestFitnessValue = fitnessValue
        bestY=y
        bestAlpha=alpha

    if i%10==0:
        print (i,bestFitnessValue)
        fp.write(str(i) + "," + str(bestFitnessValue) + "," \
                + str(bestY)+"," + str(bestAlpha)+ "\n")
    i=i+1

fp.write(str(i) + "," + str(bestFitnessValue) + "," + str(bestY)+ "," + str(bestAlpha)+ "\n")
fp.close()
print ("bestFitnessValue --> ",bestFitnessValue)
print ("BestY --> ", bestY)
print ("BestAlpha --> ",bestAlpha)
W=0
H=0
for i in range(n):
    G = bestAlpha[i] *y[i]
    H=H+G
    P = np.dot(H,x[i])
    W=W+P
print ("W =",W)

b1=0
L=0
for j in range(n):
    for i in range(n):
```

```
        S = np.dot(x[i],x[j])
        L=L+S
        b1 = b1 + bestAlpha[i] *L

b1=1-b1
print("b1=", b1)
R=0
b2=0
for j in range(n):
    for i in range(n):
        J = np.dot(x[i],x[j])
        R=R+J
        b2 = b2 + (-1)*bestAlpha[i] *R

b2=-1-b2
print("b2=", b2)
```

BIBLIOGRAPHY

Azizi H. (2010). *The interval efficiency based on the optimistic and pessimistic points of view,* Applied Mathematical Model. 35 (2011) 2384–2393.

Azizi H. (2013). *DEA efficiency analysis: A DEA approach with double frontiers,* International Journal of Systems Science 35 (2011) 2384–2393.

Azizi H. (2014). *A note on ''Supplier selection by the new AR-IDEA model''* Int J Adv Manuf Technol 71:711–716

Cooper WW, Seiford LM & Tone K (2007). *Data envelopment analysis: a comprehensive text with models, applications, references and DEA-solver software,* 2nd edition. Springer, New York.

Deng N., Tian Y. & Zhang C. (2013). *Support Vector Machines: Optimization Based Theory, Algorithms, and Extension.* Taylor & Francis Group: Boca Raton.

Korhonen, P.J., & Luptacik, M. (2004). *Eco-efficiency analysis of power plants: An extension of data envelopment analysis*. European Journal Operation Research, 154, 437-446

Liu, W.B., Meng, W., Li, X.X., & Zhang, D.Q. (2010). *DEA models with undesirable inputs and outputs*. Ann. Oper. Res., 173(1), 177-194.

Puri, J., & Yadav, S.P. (2014a). *A fuzzy DEA model with undesirable fuzzy outputs and its application to the banking sector in India*. Expert Syst. Appl., 41(14), 6419-6432.

Puri J, Yadav SP (2017). *Improved DEA models in the presence of undesirable outputs and imprecise data: an application to banking industry in India*. International Journal of System Assurance Engineering and Management, 8, 1608-1629.

Scheel, H. (2001). *Undesirable outputs in efficiency valuations*. European Journal Operation Research, 132(2), 400-410.

Seiford, L., & Zhu, J. (2002). *Modeling undesirable factors in efficiency evaluation*. European Journal Operation Research, 142(1), 16-20

Wang YM, Greatbanks R & Yang JB (2005). *Interval efficiency assessment using data envelopment analysis*. Fuzzy Sets Syst 153:347–370

Wang YM, Chin KS & Yang JB (2007). *Measuring the performances of decision-making units using geometric average efficiency.* Journal Operational Research Society 58: 929–937.

Yeh C.C., Chi D.J., Hsu M.F. (2010) *A hybrid approach of DEA, rough set and support vector machines for business failure prediction*, Expert Systems with Applications, 37(2010) 1535-1541.

https://en.wikipedia.org/wiki/Support_vector_machine

https://www.quantstart.com/articles/Support-Vector-Machines-A-Guide-for-Beginners

https://shuzhanfan.github.io/2018/05/understanding-mathematics-behind-support-vector-machines/

http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/