MULTIPLE CRITERIA SORTING METHODS BASED ON SUPPORT VECTOR MACHINES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ASLI DUMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

DECEMBER 2010

Approval of the thesis:

MULTIPLE CRITERIA SORTING METHODS BASED ON SUPPORT VECTOR MACHINES

submitted by **ASLI DUMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Sinan Kayalıgil _____
Head of Department, **Industrial Engineering**

Assoc. Prof. Dr. Esra Karasakal _____
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Meral Azizoğlu _____
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Esra Karasakal _____
Supervisor, Industrial Engineering Dept., METU

Assoc. Prof. Dr. Yasemin Serin _____
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Cem İyigün _____
Industrial Engineering Dept., METU

Dr. Yaşar Nuri Sevgen _____
Manager, Türk Telekominikasyon A.Ş.

Date: 17.12.2010

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

MULTIPLE CRITERIA SORTING METHODS BASED ON SUPPORT VECTOR MACHINES

Duman, Aslı

MSc., Department of Industrial Engineering

Supervisor : Assoc. Prof. Dr. Esra Karasakal

December 2010, 76 pages

This study addresses sorting problems with predefined ordinal classes. We develop a new method based on Support Vector Machine (SVM) model, which is mainly used for nominal binary or multi-class classification processes. In the proposed method, the SVM model is extended to include the preferences of the decision maker and the ordinal relationship between classes in sorting problems. New sets of constraints are added to the SVM model. We demonstrate the performance of the proposed method through several data sets. We compare the results with those of classical SVM model and UTADIS method, a well-known multiple criteria sorting method. We also analyze the effect of feature space mapping by Kernel Trick utilization on the results.

Keywords: Multiple Criteria, Sorting, Support Vector Machine, Kernel Trick

ÖZ

DESTEK VEKTÖR MAKİNELARI TEMELLİ ÇOK KRİTERLİ SINIFLANDIRMA YÖNTEMLERİ

Duman, Aslı

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Esra Karasakal

December 2010, 76 sayfa

Bu çalışma, önceden tanımlanmış sınıflara sahip sıralama (sorting) problemi üzerine yapılmıştır. Tezde, esas olarak iki ya da daha çok sınıflı nominal (tercih bağımsız) sınıflandırma problemlerinde kullanılan Destek Vektör Makineleri temelli yeni bir yöntem önerilmektedir. Önerilen yöntemde karar vericinin tercih önceliklerinin ve tanımlanmış olan sınıfların sıralı (ordinal) ilişkilerin dikkate alınması amacıyla DVM modeline ek kısıt kümeleri eklenmiştir. Önerilen yöntemin performansının gözlenmesi amacıyla, seçilmiş olan veri kümelerine önerilen yöntem uygulanmıştır. Sonuçlar mevcut model ve tanınmış çok kriterli karar alma yöntemlerinden biri olan UTADIS yöntemi uygulamalarından elde edilen sonuçlar ile karşılaştırılmıştır. Ayrıca Çekirdek Fonksiyon uygulamasının etkisi de incelenmiştir.

Anahtar Kelimeler: Çoklu kriter, Sınıflandırma (Sorting), Destek Vektör Makinesi, Çekirdek Fonksiyon

To

My lovely family

and

Cihat Güner

ACKNOWLEDGEMENTS

Many thanks to my colleagues at my company, especially to my Manager and to my Director.

Many thanks to my supervisor, for her understanding and support.

Special thanks to my family, for their support and understanding, not only for my thesis process but also for all my education life.

And finally never enough thanks to my fiancée, for his added value to my whole life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

PROBLEM DEFINITION

An objective can be described as something that is tried to be achieved or obtained by a decision maker. A decision maker may have more than one objective. A criterion is a measure of effectiveness of performance according to the objectives of the decision maker. Usually due to scarce resources, the improvement in one criterion may lead to a worse score in another. This concept of conflicting criteria is the basis of the Multi Criteria Decision Making (MCDM) problems.

The MCDM problems can be classified into three main classes;

Choice (Selection) problems are the problems where the decision maker selects the best alternative from a given set of alternatives (e.g. choosing only one book to buy),

Ranking problems try to rank the given set of alternatives from best to worst (e.g. MBA program rankings),

Classification problems deal with the assignment of a set of alternatives into predefined prioritized or non-prioritized classes. These classes may be *nominal* (e.g. cancerous and non-cancerous cells) or may be *ordinal* (e.g. grading students from AA to FF). Ordinal classification problems are also named as *sorting* problems.

The main interest of this study is *sorting* problems with predefined ordinal classes. In many real life cases, in personal or in business life sorting is required. As an

example in business application area, sorting is utilized in Turk Telekom with predefined ordinal customer classes. Corporate customers are assigned to classes according to some criteria defined by Corporate Marketing Department. Since the criteria used for sorting the alternatives are conflicting in addition with large customer size and volume, it is difficult to proceed manually. Moreover, obtained information is planned to be utilized for Customer Experience Improvement actions. These actions will differentiate according to the ordinal class that the customer assigned to. So it is critical to realize subjective separation as much as possible.

In order to be utilized in such ordinal classification problems, we develop a new method based on Support Vector Machine (SVM), which is mainly used for nominal classification processes. After first proposed in Boser et al. (1992), SVM's are being widely used in decision making problems. These problems mainly consist of visual categorization purposes, as text categorization (Joachims, 1998), medical cell categorization (Furey, 2000), image categorization (Tong E. C., 2001), etc. As seen from the publications, binary classification is the main area of applications. Moreover, the classes do not have any prioritizations. The main focus is to decide wheater the alternative is belong to a class or not.

When the problems consist more than two nominal classes, Multi-class SVMs are utilized as combinations of binary SVMs. After performance shown in learning ranking functions, SVM's are extended to be utilized in ranking problems, (Yu, 2005).

In the proposed method, SVMs are extended to handle ordinal classification problems with two or more classes and to incorporate the preferences of the decision maker. To our knowledge, there is no study that utilizes SVMs in ordinal class problem. New sets of constraints are added to the SVM model and both versions of model are applied to the selected data sets. Effects of feature space mapping by Kernel Trick utilization are also studied. Moreover, UTADIS method,

a method utilized for multi class sorting problems, is applied to the data sets and all of the results are compared with each other.

Organization of the thesis is as follows. In Chapter 2, we present Literature Review with detailed information on SVM and UTADIS technique. We develop our approach in Chapter 3. We report our computational experiments in Chapter 4 and present concluding remarks and potential directions for further research in the last chapter.

CHAPTER 2

LITERATURE REVIEW ON SORTING PROBLEM

There are a few literature review papers on Multicriteria Decision Aid (MCDA) methods developed for classification/sorting problems (Zopounidis and Doumpos, 2002), (Zopounidis and Doumpos, 2003), (Yevseyeva, 2007). There also exists a study of Wallenius, et al. (2008) which covers Multiple Criteria Decision Making (MCDM) and Multiattribute Utility Theory (MAUT) studies from 1992 to 2008, and explains possible future research questions.

Multidimensional classification models are first studied by the mid 1930s, followed by many extensions until late 1940s (Zopounidis and Doumpos, 2002). It is figured out that, in classification literature, methods are classified into groups according to: (1) the form of the criteria aggregation model and (2) the methodology utilized for parameter definition (Zopounidis and Doumpos, 2002).

Zopounidis and Doumpos (2002) divides MCDA sorting methods into three classes.

To begin with, Utility Theory Framework is considered. Majority of the MCDM/MCDA methods operate under the assumption of the *utility function* existence. Utility theory assumes that during the decision making process, DM tries to maximize, consciously or unconsciously, some utility function. The preferences of the decision maker are introduced to the model by utilization of the criteria weights and by determination of the shape of the utility function. As mentioned in Yevseyeva (2007), since Utility Theory is first developed in 1947, it is also utilized

in many methods. UTADIS is also one of the most well-known sorting methods worked under utility theory framework. In the following sections, UTADIS will be explained in detail.

Utility Theory is also utilized in methods not only for sorting. Multi Attribute Utility Theory (MAUT) (Keeny, 1976) is developed with the aim of dealing with the tradeoffs among multiple objectives and Analytical Hierarchy Process (AHP) (Saaty, 1980) is introduced in order to handle the complex decision, based on finding the alternative best suits the needs of the decision maker.

In the second class, outranking relations concept is visited. Outranking relations are based on the comparison of each pair of alternatives in order to deal with the decision making process. First developed for Choice (Selection) Problems, ELECTRE (ELimination Et Chox Traduisant la REalité) method evolved in ranking and sorting types of decision making problems. ELECTRE TRI (Zopounidis and Doumpos, 2002) is an example of evolvement for sorting problems. In this method, information regarding the weights, preferences, thresholds, etc. need to be obtained from the decision maker in order to proceed. Some methods as DIVAPIME (Determination d'Intervalles de Variation pour les Parametres d'Importance des Methodes ELECTRE) and ELECTRE TRI Assistant are developed with the aim of simplification of this procedure (Yevseyeva, 2007).

Dias, et al. (2008) developed a method called Electre Tri-C, based on ELECTRE framework. In this method the classes are defined regarding central reference actions instead of boundary profiles. Two distinct assignment rules are applied. One of the rules can be thought as optimistic while the other one pessimistic.

One of the other methods that use indirect preference elicitation is Stochastic Multicriteria Acceptability Analysis (SMAA) (Yevseyeva et al., 2007) althoug it is for stochastic problems. In this method, each class is required to have some pre-assigned alternatives. This information is necessary in order to proceed for the determination of the characteristics related with each class.

Preference Ranking Organization METHod for Enrichment Evaluations (PROMETHEE), introduced by Brans (1982), is another method based on outranking relations. In this method, relative importance of criteria is required to proceed. Moreover pre-assigned alternatives are evaluated based on pairwise comparisons in order to find the deviation between the evaluations. This information is also utilized during the utilization of the method.

Nemery and Lamboray (2007) developed FlowSort method, for the assignment of the actions to ordinal categories. It is based on PROMETHEE and utilizes the relative position of an alternative concerning the set of reference profiles.

Third class, especially used for nominal classification problems is the simple discriminant function. In this method, score for each of the alternatives is produced by multiplication of the vector of the alternative with a vector of coefficients.

Although above mentioned methods require information from the decision maker in advance, some studies interact with the decision maker not only at the beginning but also during the decision making process. Chen, Hipel and Kilgour (2008) proposed an interactive method to improve the efficiency of sorting for the decision maker. In this method decision maker can decide on the number of groups and other sorting characteristics and also interfere and adjust the information given during the procedure.

In the study of Greco, Mousseau and Slowinski, (2008), the information obtained from the decision maker by the assignment of the alternatives into classes are used to construct a model representing his preferences. The model derived from a set of general additive compatible value functions also takes the thresholds associated into account.

As stated in Wallenius, Dyer, Fishburn, Steuer, Zionts and Deb (2008), a variety of studies have been carried out on Multicriteria Decision Making field . But this does

not mean that all problems are solved. There is still lot to do by the research community, which also promises other subfields to cooperate and collaborate in.

One of the promising methods is Support Vector Machines (SVM). Based on the studies of Vapnik in 1979 (Vapnik, 1995), SVM's are first proposed in Boser et al (1992). In this method, the only required information is the training data set where alternatives are assigned in predefined classes. By the utilization of this input, without any need of additive information from the decision maker, model creates separating hyperplanes between the classes.

To our knowledge, there is no attempt to incorporate ordinal relationship between classes into SVMs and extend SVMs to sorting problems. Classical SVMs mainly used for binary classification without any prioritizations between the classes. The main focus is to decide whether the alternative is belong to a class or not.

When the problems consist more than two nominal classes, Multi-class SVMs are utilized as combinations of binary SVMs. Basically, model is compiled for each class of alternatives and the results are combined to decide the actual class of the alternative. However again there is no prioritization between the classes.

After performance shown in learning ranking functions, SVMs are extended to be utilized in ranking problems, (Yu, 2005).

2.1. Support Vector Machines

SVM is a machine learning method used mainly for classification and regression based problems.

An alternative (data point) is represented by an n-dimensional vector where n denotes the number of the criteria. Set of this kind of alternatives are tried to be separated into classes geometrically with an (n-1)-dimensional hyperplane. These separative hyperplanes are also called as linear classifiers.

When alternatives of different classes are located in space, there exist more than one hyperplane that can be used for geometric separation between each consecutive class. A binary classification representation can be seen in Figure 1.

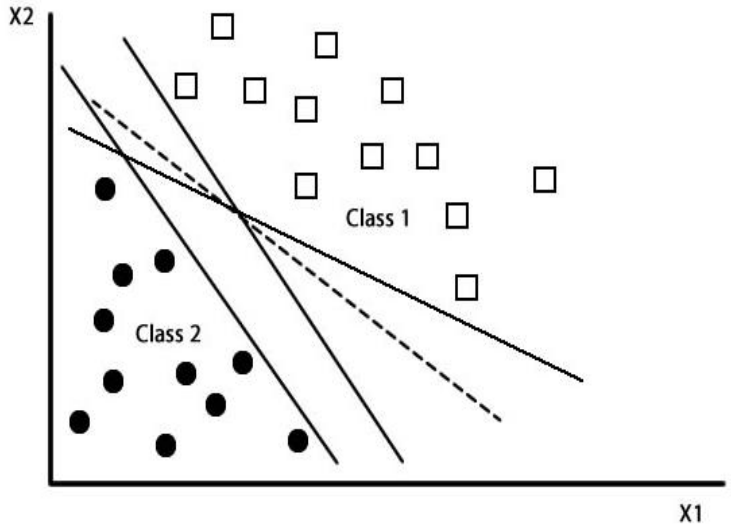

Figure 1 - Multiple hyperplanes separating points belong to two different classes

A good separation will be achieved with a hyperplane providing largest separation (i.e. largest margin (M) between two classes). Margin can be defined as the width that the boundary can be enlarged as much as possible before hitting any point representing alternatives. This hyperplane is called as the maximum-margin hyperplane and the linear classifier defined by this hyperplane is called as a maximum margin classifier.

With the aim of finding the maximum margin hyperplane, support vectors are created. Any point in n-dimensional space can be represented as an n-dimensional vector (pattern vector), which connects the origin with the point. A support vector is the vector perpendicular to the pattern vector of a data point, as shown in Figure 2.

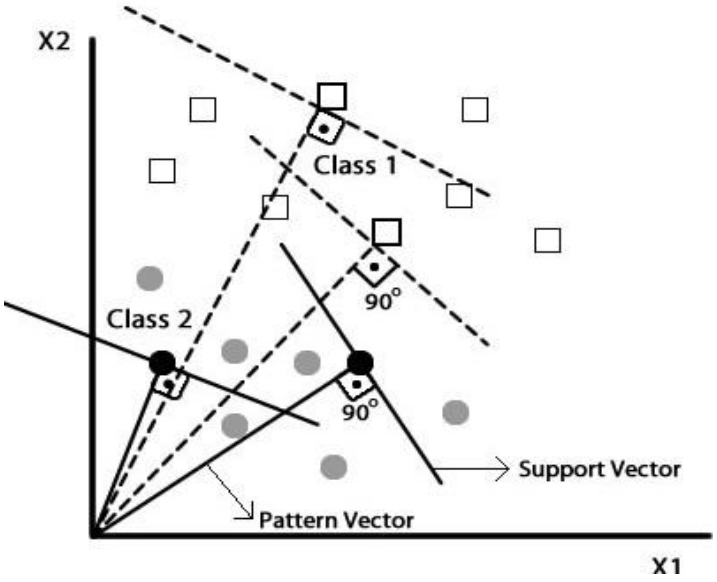

Figure 2 – Support vectors according to the vectors representing the alternative points

Based on the Statistical Learning Theory (Vapnik, 1979), SVM is a classifier that utilizes a given set of training alternatives assigned to classes initially. According to support vectors created, SVM finds the two closest points belonging to consecutive classes and creates the maximum-margin hyperplane. The representation in binary ordinal classification can be seen in Figure 3.

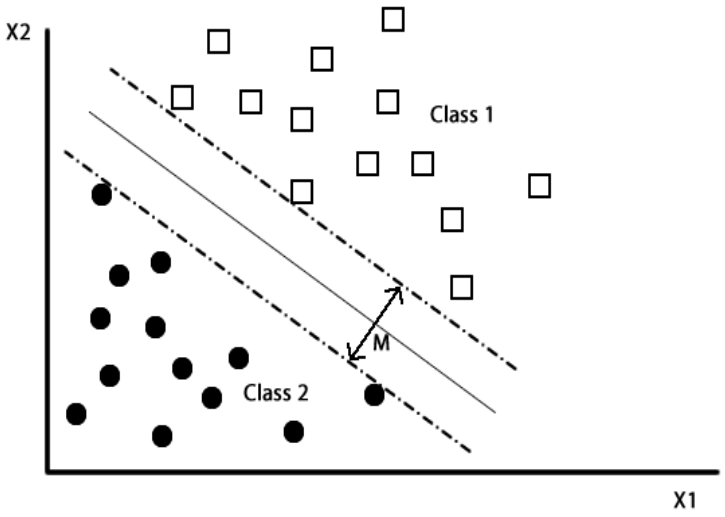

Figure 3 – Maximum margin hyperplane created by SVM

In the following sections, three cases for a binary classification problem, namely; Linearly Separable Case, Linearly Non-Separable Case and Non-Linear SVM Case are explained. Moreover Kernel trick used for Non-Linear SVM Case is described. Extensions for multi class separation are also visited at the end of this section in Multi-class SVM clause.

2.1.1. Linearly Separable Case

Linearly separable case covers the basic binary SVM model, where the alternative points belonging to distinct classes can be separated without any violation. This means that a hyperplane can be created without any points located on the wrong side of the hyperplane.

Given a dot product space S, where all of the training data $x_1,\ldots,x_m \in x$ and $x \in S$, any hyperplane in S can be written as (Schölkopf and Smola, 2002);

$$\{x \in S | w^T x + b = 0\}, w \in S, b \in R \quad (1)$$

Similarly, hyperplane created by SVM model has the following form (Cristianini and Shawe-Taylor, 2000);

$$f(x) = w^T x + b = 0 \quad (2)$$

where w a vector that defines the normal of the separating hyperplane, b is the threshold.

Let N be the number of alternatives to be classified and n be the number of criteria. Let $x_1, \ldots, x_N \in R^n$ denote the training data.

The model uses signum function (sign) as the classification rule with labels $y_i \in \{-1, +1\}$. y_i takes the value {-1} or {+1} according to the class which alternative i

belongs to. These labels show the position of the alternative compared to the hyperplane to be created.

$$w^T x_i + b \geq +1, \;\; y_i = +1 \,, \;\; \text{x}_\text{i} \text{ of the first class} \tag{3}$$

$$w^T x_i + b \leq -1, \;\; y_i = -1, \;\; \text{x}_\text{i} \text{ of the second class} \tag{4}$$

Equations (2) and (3) can be written in a more compact form by introducing y_i as follows:

$$y_i(w^T x_i + b) \geq +1, i = 1, \ldots, N \tag{5}$$

Graphical representation of the linearly separable case is shown in Figure 4 below.

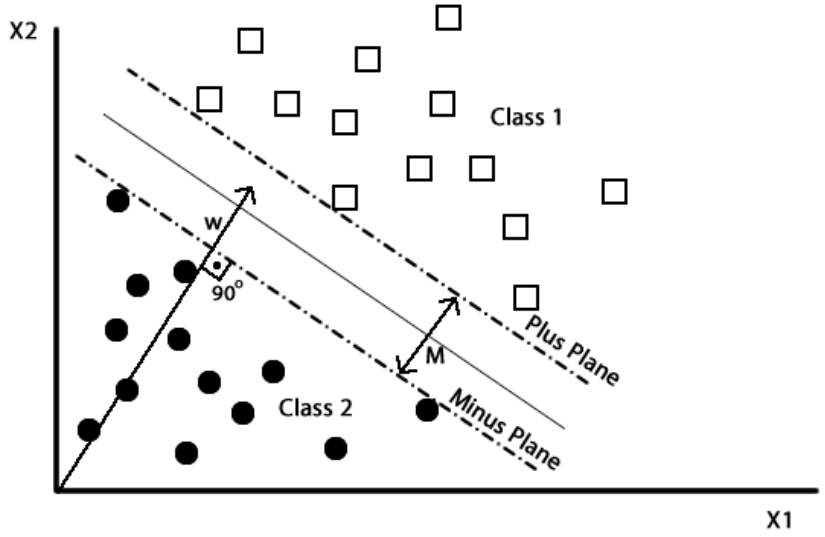

Figure 4 – Linearly separable case

In this case, the margin represented by M is the distance between the closest points belonging to separate classes. M is also equal to the distance between plus and minus planes. Vector w is perpendicular to both plus and minus planes.

A good separation will be achieved by a maximum margin hyperplane. In order to calculate the margin following equations are used:

$$w^T x_1 + b = +1 \tag{6}$$

$$w^T x_2 + b = -1 \tag{7}$$

where x_1 is any point on plus plane and x_2 is any point on minus plane. The "1" in the equations results from fixing the functional margin to be equal to 1. So the following equation can be created under the knowledge of w is a perpendicular vector to the both planes:

$$x_1 = x_2 + \lambda w \tag{8}$$

where λ is a coefficient. And the margin is the distance between the planes so the following equation can be written:

$$M = |x_1 - x_2| \tag{9}$$

When x_1 in Eq. (6) is replaced by the x_1 in Eq. (8), following equation is obtained:

$$w^T(x_2 + \lambda w) + b = +1 \tag{10}$$

By expanding the expressions in the brackets:

$$w^T x_2 + b + \lambda w^T w = +1 \tag{11}$$

When Eq. (7) is introduced into Eq. (11), following equation is obtained:

$$\lambda w^T w = +2 \tag{12}$$

By leaving the coefficient λ on the left-hand-side, λ is found to be equal to the following expression:

$$\lambda = \frac{2}{w^T w} \tag{13}$$

By subtracting x_2 from both sides in Eq. (8), following equation is obtained:

$$x_1 - x_2 = \lambda w \tag{14}$$

When Eq.(12) is introduced in Eq. (7), following equation is obtained:

$$M = |\lambda w| \tag{15}$$

When λ in Eq. (15) is replaced with λ in Eq. (13), margin is found to be equal to the following expression:

$$M = \frac{2}{\sqrt{w^T w}} \tag{16}$$

Since maximizing the margin is the main objective, maximizing Eq. (16) corresponds to minimization of the following expression:

$$\frac{1}{2} w^T w \tag{17}$$

A maximizing-margin hyperplane can be created by solving the following mathematical model, as shown in Cristianini and Shawe-Taylor (2000).

$$\text{Min z} = \frac{1}{2} \mathrm{w^T w} \tag{18}$$

subject to

$$y_i(w^T x_i + b) \geq +1, \; i = 1, \ldots, N \tag{19}$$

where $x_1, \ldots, x_N$ is the training data set, w is a vector of decision variables that define the normal of the separating hyperplane, y_i is the class label of x_i and b is the threshold.

2.1.2. Linearly Non-Separable Case

In cases where the alternatives assigned to different classes cannot be separated linearly, misclassifications need to be tolerated but also penalized. When real life situations are taken into account, perfectly separable case may not be covering all cases. Consequently, covering the misclassifications within the model expands the area of the application.

A two-class linearly non-separable example is depicted in Figure 5 – *Linearly non-separable*

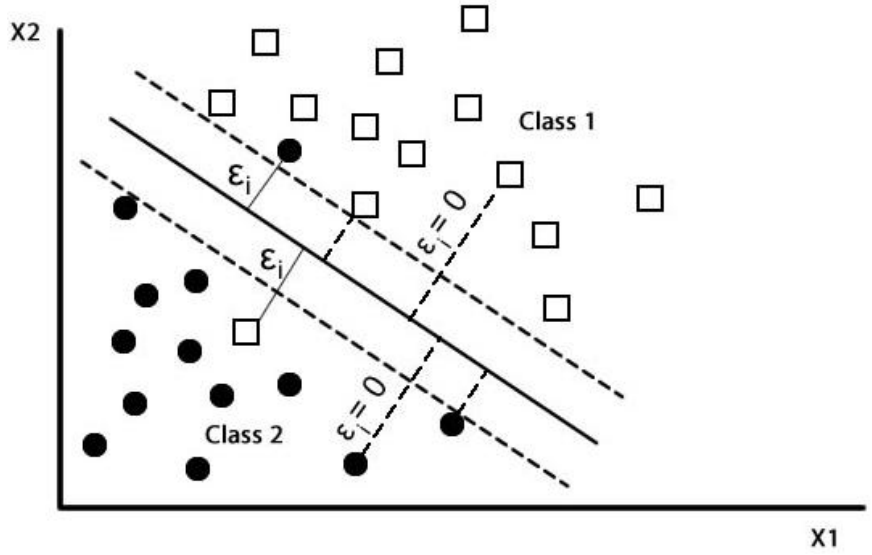

Figure 5 – Linearly non-separable case

With the aim of introducing the misclassifications into the model, the slack variables are added to Equations (20) and (21) as follows (Cristianini and Shawe-Taylor, 2000):

$$\text{Min } z = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \varepsilon_i \quad (20)$$

subject to

$$y_i(w^T x + b) \geq +1 - \varepsilon_i, \; i = 1, \ldots, N \quad (21)$$

$$\varepsilon_i \geq 0, \; i = 1, \ldots, N \quad (22)$$

where C is a positive constant denoting the trade-off between margin maximization and error minimization and ε_i is the slack variable measuring misclassification error of alternative i.

ε_i is the distance between alternative point i, in training data set x, and the hyperplane created by SVM model. ε_i takes a value greater than zero if and only if alternative point i is located on the wrong side of the hyperplane. Otherwise, ε_i is equal to zero which makes the related constraint equal to the one in the linearly separable case.

2.1.3. Kernel Function and Kernel Trick

As stated in Sewell (2007); word *kern* is first mentioned in a study in 1904 and for the linear space representation. First studies were conducted in 1940s.

In Cristianini and Shawe-Taylor (2004, p.48), a kernel function is described as "a function that returns the inner product between the images of two inputs in some feature space". A kernel function can be defined as a function K, where:

$$K(x, x') = \langle \emptyset(x), \emptyset(x') \rangle \text{ for all } x, x' \in X \subseteq R^n \quad (23)$$

where $\emptyset$ is a mapping from X to another space *F*, which is called as feature space.

$$\emptyset : x \rightarrow \emptyset(x) \in F \subseteq R^n \quad (24)$$

Feature space, F, is a strict inner product space, which means there is a real-valued map $\langle x, x \rangle$ which has the following properties:

- $\langle x, x \rangle$ is real-valued, symmetric and bilinear (when the elements of the two vector spaces (input and feature spaces) considered, they are all linear in all arguments)
- $\langle x, x \rangle \geq 0$,
- $\langle x, x \rangle = 0$ if and only if $x = 0$.

Moreover, feature space, F, also has additional properties of separability and completeness. It is also shown that, any kernel can be utilized to create feature space, F (Cristianini and Shawe-Taylor, 2004).

The main aspects of the Kernel approach are listed in Cristianini and Shawe-Taylor (2004) as follows:

- The vector space where input data points are embedded into is called a feature space,
- In the feature space, linear relations between the input points are also directed to the embedded ones,
- Coordinates are not introduced separately, only the dot products of the embedded points are utilized,
- By the utilization of the kernel function, dot products can be calculated with the input data points.

i. Kernel Trick

As cited in Sewell (2007), kernel trick was first proposed by Mercer (1909) and was first utilized with the aim of diverting kernels for usually higher dimensional future space by Aizerman, et al. (1964). Combination of kernel function with separating hyperplanes, is first introduced by Boser, et al. (1992).

The kernel trick transforms any input space form that solely depends on the dot product between two vectors to feature space form. Wherever a dot product is used, it is replaced by the kernel function. Thus, a linear algorithm can easily be transformed into a non-linear algorithm. This non-linear algorithm is equivalent to the linear algorithm operating in the range space of $\emptyset(x)$.

Some of the most widely used kernel functions are;

Linear: $K(x_i, x_j) = x_i x_j$

Polynomial: $(x_i, x_j) = (1 + x_i x_j)^p$, p will be selected by the user.

Gaussian: $K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$, σ will be selected by the user.

In Figure 6, an example of kernel function application is shown. The data in two-dimensional space is mapped into three-dimensional space with polynomial kernel with $p = 2$.

$$\emptyset : x = (x_1, x_2) \rightarrow \emptyset(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in F \subseteq R^3$$

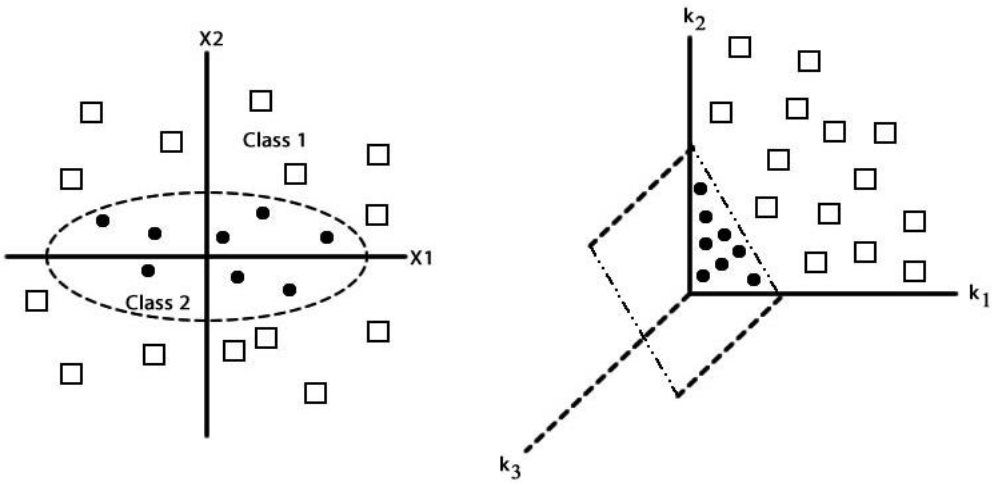

Figure 6 – An example of higher dimensional feature space

Kernel functions utilized in this thesis are;

Polynomial:$K(x_i, x_j) = (1 + x_i x_j)^p$, $p = 2$.

Gaussian:$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$, $\sigma = 2$.

ii. Non-Linear SVM Case

The method applicable to linear cases is extended by Vapnik (1995) with the aim of covering non-linear cases. The main idea behind this extension is to map the data into a higher dimensional space where the classes will be linearly separable.

This is achieved by utilizing kernel trick to map the training data x into feature space as seen below;

$$\text{Min } z = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \varepsilon_i \tag{25}$$

subject to

$$y_i(w^T \emptyset(x) + b) \geq +1 - \varepsilon_i, \ i = 1, \dots, N \tag{26}$$

$$\varepsilon_i \geq 0, \ i = 1, \dots, N \tag{27}$$

Geometric explanation can be seen in Figure 5. In this figure by the utilization of a kernel function, linearly non-separable data are mapped into a feature space where they can be separated linearly.

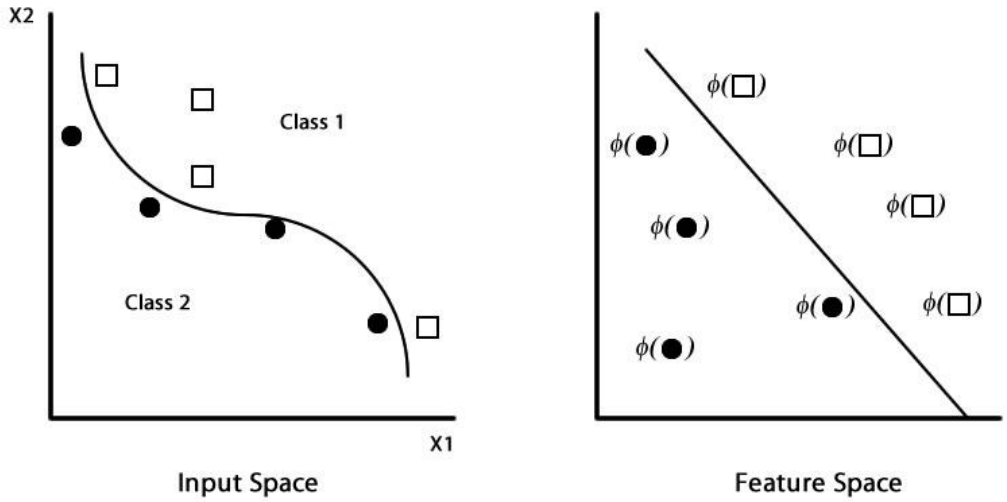

Figure 7 - Geometric interpretation of kernel utilization from input space to feature space.

2.1.4. Multi Class SVM

As mentioned before SVMs are mainly created for binary classification. But there are also methods for extending the method for multi-class applications. Here it should be noticed again that these applications are for nominal multi-class separations, there is no prioritization or preference order between the classes. Actually these methods could be considered as combination of two-class SVMs.

In Schölkopf and Smola (2002, pp. 211-214) the SVM methods used for solving classification problems, having more than two classes, are classified into four groups.

i. One versus the Rest

In this method, a set of binary classifiers are constructed by solving the model for each class for separating it from the rest of the classes. This means in order to complete the nominal-multiclass separation, the model will be compiled for each class individually.

The classifiers constructed and resulting classification labels can be represented as follows;

$$f^j(x) = sgn(\langle w^j, x_i \rangle + b^j), j = 1, \dots, m \quad (28)$$

where f^j is the binary classifier, in iteration for class of alternatives x_i in class j and m is the number of classes.

After compilations for all of the classes are completed, alternative x_i is classified into the class, according to the maximal output as a result of the combination as follows;

$$\text{argmax}(\langle w^j, x_i \rangle + b^j) \quad (29)$$

where argmax (f^j (x_i)) is the set of values of x_i for which f^j (x) has the largest value.

The main weakness of this approach is called "winner-takes-all approach". According to this approach, the real valued outputs of the binary classifiers, obtained by solving the model each time, may not be on comparable scales. This means, when each class is separated from the rest, a different problem is trained, and when this process is repeated for all of the classes, several solutions that may be on different scales are obtained and compared according to the training data used.

Moreover when all of the hyperplanes are located on the space of the alternatives, there are regions remaining unclassified, as seen in Figure 8.

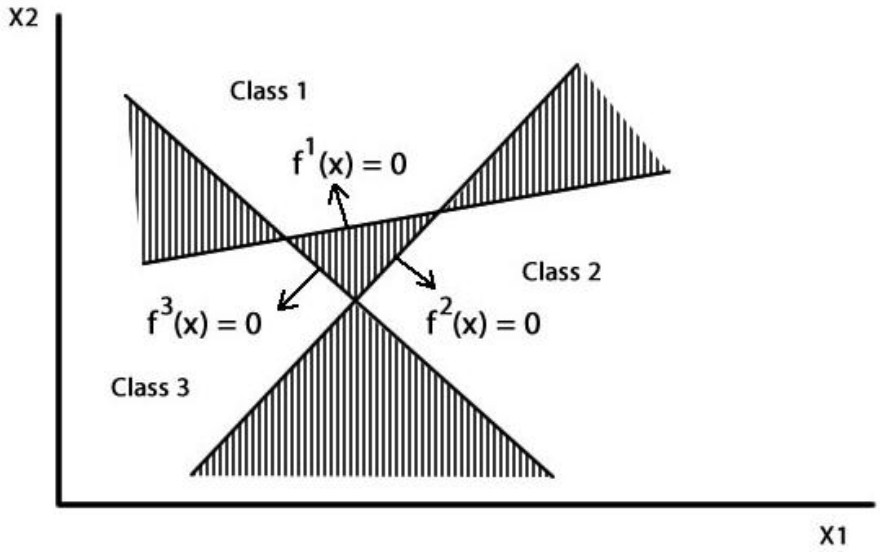

Figure 8 – Regions remain unclassified after one versus all method is applied.

ii. Pairwise Classification (One versus one)

In pairwise classification, classifiers are constructed for each possible pair of m classes, resulting in (m-1)m/2 binary classifiers.

The constructed classifiers and resulting classification labels can be represented as follows:

$$f^{kj}(x) = (\langle w^{kj}, x_i \rangle + b^{kj}), \qquad k, j = 1, \dots, M \tag{30}$$

where f^{kj} is the binary classifier, for the alternatives in class k with respect to class j and training data set is represented by x_i.

Datum x is classified into the class, according to the maximal output as a result of the combination as follows;

$$\text{argmax} \ (i) f^{i}(x), \qquad i = 1, \dots, M \tag{31}$$

where argmax f(x) is the set of values of x for which f(x) has the largest value and,

$$f^{\,i}(x) = \sum_{j \neq i, j=1}^{M} sign(f^{\,ij}(x)), \qquad i, j = 1, \dots, M \tag{32}$$

Although in pairwise classification, the number of the model compilations is larger than that in one versus the rest method, the size of each individual problem to be solved is relatively smaller. This results in shorter computation time per problem.

Graphical representation of the class regions constructed by pairwise classification can be seen in Figure 7.

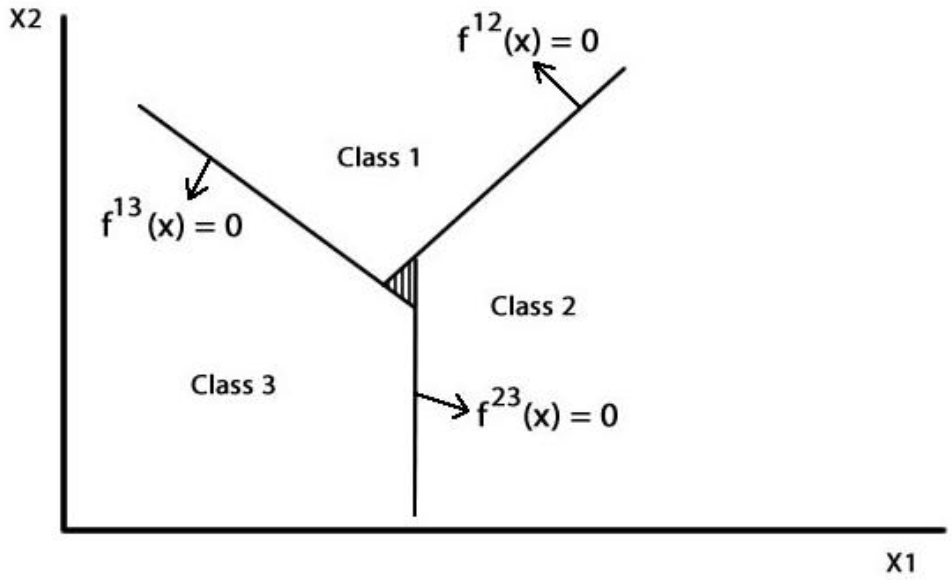

Figure 9 – Class regions constructed after one versus one method is applied.

As seen in the figure above, between the classes, a region is left unclassifiable, where equation (31) holds for more than one i. In order to get rid of this kind of regions, following methods are developed.

- Decision Directed Acyclic Graph (DDAG) (Platt, Cristianini, Shawe-Taylor, 2000): decision-tree-based pairwise classification,

- Adaptive Directed Acyclic Graph (ADAG) (Pontil and Verri, 1998; Kijsirikul and Ussivakul, 2002): pairwise classification based on rules of a tennis tournament.

iii. Error-Correcting Output Coding

In this method, there exist two stages. At first stage a set of binary classifiers is created with the aim of test pattern class determination. This set of classifiers is represented as follows:

$$f^1, \dots, f^L \tag{33}$$

This will lead to a vector of responses for each alternative in training data set x, represented as follows:

$$f^1(x), \dots, f^L(x) \tag{34}$$

where each vector consists of $\{\pm 1\}^L$.

When all of the *M* classes are considered, in the second stage, a matrix, so-called Decoding matrix M is created, and this matrix is represented as follows:

$$\text{Decoding matrix M} \in \{\pm 1\}^{M\ x\ L} \tag{35}$$

This matrix is used as a voting scheme. In order to obtain robustness, the matrix is created with the Hamming Distance to be at least three, where, Hamming Distance is the difference, calculated by counting the different entries of the two vectors of equal length.

iv. Multi-Class Objective Functions

This method allows simultaneous computation of a multi-class classifier by the utilization of the following quadratic program (Weston and Watkins, 1993):

$$\text{Min } z = \frac{1}{2}\sum_{r=1}^{M}\|\mathrm{w_r}\|^2 + \frac{\mathrm{C}}{\mathrm{m}}\sum_{\mathrm{i=1}}^{\mathrm{m}}\sum_{\mathrm{r}\neq \mathrm{y_i}}\varepsilon_{\mathrm{i}}^{\mathrm{r}} \tag{36}$$

Subject to $$w_{y_i}^T x_i + b_{y_i} \geq w_r^T x_i + b_r + 2 - \varepsilon_i^r, \tag{37}$$

$$\varepsilon_i^r \geq 0 \tag{38}$$

where, M is the number of classes, $m \in \{1, \dots, M\} \setminus y_i$ and $y_i \in \{1, \dots, M\}$.

The size of the problem to be solved is larger compared to the previous methods while all of the classification is done at the same time.

Although this mathematical model deals with all the classification under single problem, again there is no prioritization between the classes. It works as combination of several binary SVMs, where constraints are compiled together for each class of alternatives.

Within the Experimental results section, this model is used as "classical SVM" method and results are compared with the proposed method.

For instance, for a three classed data set, the model is constracted as follows:

$$\text{Min } z = \frac{1}{2}\sum_{r=1}^{3}\|\mathrm{w_r}\|^2 + 1\sum_{\mathrm{i=1}}^{3}\sum_{\mathrm{r}\neq \mathrm{y_i}}\varepsilon_{\mathrm{i}}^{\mathrm{r}} \tag{39}$$

Subject to $$w_1^T x_i + b_1 \geq w_r^T x_i + b_r + 2 - \varepsilon_i^r, \quad \text{r=2, 3 and i} \in \text{Class 1} \tag{40}$$

$$w_2^T x_i + b_2 \geq w_r^T x_i + b_r + 2 - \varepsilon_i^r, \quad \text{r=1, 3 and i} \in \text{Class 2} \tag{41}$$

$$w_3^T x_i + b_3 \geq w_r^T x_i + b_r + 2 - \varepsilon_i^r, \quad \text{r=1, 2 and i} \in \text{Class 3} \tag{42}$$

$$\varepsilon_i^r \geq 0 \tag{43}$$

As seen from the Multi Class SVM methods above, they cannot be utilized for ranking or sorting problems as they are not constructed for that purpose. Moreover as stated in Schölkopf and Smola (2002, pp. 211-214), it is hard to say that one of the methods always outperforms the others for classification problems. Factors as, number of classes on hand, time available, etc. will affect the choice between the methods. But if the problem to be solved includes prioritization of the decision maker, these methods would not work properly.

2.1.5. Areas of Application

Since SVM's are easy to utilize and no assumptions or approximations needed for compilation, there are studies on a wide range of areas.

Shawe-Taylor and Cristianini (2004) states that the main advantages of the SVM utilization is the utilization of kernels, not producing local minima, the sparseness of the solution and capacity control obtained by margin optimization.

In literature, SVMs are used for the following purposes;

Financial analysis purposes: (Pang, 2002), (Gastel and July, 2001), (Huang, 2004), (Boser, 1992), (Tay, 2001)

Text categorization purposes: (Joachims, 1998), (Joachims, 2002), (S. Tong D. K., 2002)

Medical diagnosis purposes: (Furey, 2000), (Guyon, 2002)

Image classification/recognition purpose: (Tong E. C., 2001), (Pontil, 1998)

Face detection purposes: (Osuna, 1997), (Heisele, 2001), etc.

2.2. UTADIS

Although UTADIS method was developed in 1980, not until mid 1990s, it is utilized effectively. Afterwards, the method used for evaluation and classification/sorting problems. While being a variant of UTA (UTilités Additives) method, the main aim of UTADIS is to classify the alternatives by developing a criteria aggregation model. There is a preference relation between the classes where alternatives are assigned. While the group of alternatives in the first class has the highest scores, the scores get worse and worse as moving to the last class of alternatives.

The model can be described as follows:

$$\text{Min } \left\{\sum_{k=1}^{q}\left[\frac{\sum_{\forall x_j \in C_k}\left(\sigma_j^{+}+\sigma_j^{-}\right)}{m_k}\right]\right\} \tag{44}$$

subject to

$$\sum_{i=1}^{n}\left(\sum_{p=1}^{r_{ji}-1} w_{ip} + \frac{g_{ji}-g_i^{r_{ji}}}{g_i^{r_{ji}+1}-g_i^{r_{ji}}} w_{ir_{ji}}\right) - u_1 + \sigma_j^{+} \geq \delta_1, \qquad \forall x_j \in C_1 \tag{45}$$

$$\left.\begin{aligned}\sum_{i=1}^{n}\left(\sum_{p=1}^{r_{ji}-1} w_{ip} + \frac{g_{ji}-g_i^{r_{ji}}}{g_i^{r_{ji}+1}-g_i^{r_{ji}}} w_{ir_{ji}}\right) - u_k + \sigma_j^{+} \geq \delta_1 \\ \sum_{i=1}^{n}\left(\sum_{p=1}^{r_{ji}-1} w_{ip} + \frac{g_{ji}-g_i^{r_{ji}}}{g_i^{r_{ji}+1}-g_i^{r_{ji}}} w_{ir_{ji}}\right) - u_{k-1} - \sigma_j^{-} \geq -\delta_2\end{aligned}\right\}$$

$$\forall x_j \in C_k (2 \leq k \leq q-1) \tag{46}$$

$$\sum_{i=1}^{n}\left(\sum_{p=1}^{r_{ji}-1} w_{ip} + \frac{g_{ji}-g_i^{r_{ji}}}{g_i^{r_{ji}+1}-g_i^{r_{ji}}} w_{ir_{ji}}\right) - u_{k-1} - \sigma_j^{-} \geq -\delta_2, \qquad \forall x_j \in C_q \tag{47}$$

$$\sum_{i=1}^{n}\left(\sum_{p=1}^{a_i-1} w_{ip}\right) = 1, \tag{48}$$

$$u_k - u_{k-1} \geq s, \ \forall k = 1, 2, \ldots, q-2 \tag{49}$$

$$\sigma_j^+ \geq 0,\ \sigma_j^- \geq 0,\ \forall j = 1, 2, \dots, m \tag{50}$$

$$w_{ip} \geq 0,\ \forall i = 1, 2, \dots, n, \forall p = 1, 2, \dots, a_i - 1 \tag{51}$$

where n is the number of criteria, m is the number of alternatives, q is number of classes and $a_i - 1$ is number of the subintervals created. Moreover σ_j^+ is the violation of the lower bound and σ_j^- is the violation of the upper bound.

$$g_{ji} \in \left[g_i^{r_{ji}}, g_i^{r_{ji}+1}\right] \tag{52}$$

where $g_i^{r_{ji}}$and $g_i^{r_{ji}+1}$ stands for consecutive break points of a subinterval and r_{ji} denotes the related subinterval which the score g_{ji} of alternative x_j on criterion $g_i$belongs to. Global utility of alternative x_j is;

$$U(g_j) = \sum_{i=1}^{n}\left(\sum_{p=1}^{r_{ji}-1} w_{ip} + \frac{g_{ji} - g_i^{r_{ji}}}{g_i^{r_{ji}+1} - g_i^{r_{ji}}} w_{ir_{ji}}\right) \tag{53}$$

δ_1 and δ_2 , very small positive values, are used to avoid the cases like $U(g_j) = u_k$ when $x_j \in C_k$.

Eq. (44) represents the objective that is to minimize the overall sum of the total violations of the thresholds calculated for each class of alternatives.

Eq. (45) – Eq. (47) are used in order to locate the alternatives into the subintervals created by the user.

Eq. (48) and Eq. (51) are used in order to normalize the utility function defined on criteria's scale. This means, least preferred value of the criterion is equal to 0, while most preferred value of the criterion is equal to 1. Moreover the criteria in between will take values between 0 and 1 according to their location on scale.

Eq. (50) ensures the differentiation of consecutive thresholds.

Eq. (50) is used with the aim of representing the lower and upper bounds. When all error variables, $\sigma_j^+$and σ_j^-, are zero, this means there exist multiple optimal solution, known as degeneracy.

In addition to degeneracy issue, stability of the solution needs to be studied. Both of these issues covered as post-optimality analysis after the model is solved (Doumpos and Zopounidis, 2002, pp. 78-95).

CHAPTER 3

PROPOSED APPROACH

3.1. A Sorting Method Based on SVMs

As stated before, in literature SVMs are mainly used for nominal binary or multi-class classification processes. Even the model is constructed with the aim of solving multi-class classification problems; it performs as combination of binary SVMs. Basically, model is compiled for each class of alternatives. This can be done by compiling a model for seperating each class of alernatives from the rest of the classes than cobining the results (i.e. one versus the rest) or by compiling a model for seperating each pair of classes seperately than combining the results (i.e. pairwise classification, error-correcting output coding) or pairwise classification can be performed in a single model (i.e. multi-class objective functions). However again there is no prioritization between the classes.

To our knowledge, there are some studies conducted with the aim of extending the SVM's to be utilized in ranking problems. Their performance is shown in learning ranking functions (Yu, 2005). However there are no studies that address the utilization of SVM's on sorting types of problems.

In this proposed method, called as SVM – Sorting (SVM-S-v0), the model is constructed so preferences of the decision maker and the ordinal relationship between classes will also be taken into account. Unlike the current Multi-class SVM's, constraints included in the model are adjusted so an alternative is located according to its location to all of the hyperplanes simultaneously.

In addition to SVM-S, with the aim of forcing the alternatives to be located correctly more new sets of constraints are added to the SVM model (SVM-S-v1).

SVM-S is explained in detail as follows:

$$\text{Min } z = \sum_j \frac{1}{2} w_j^T w_j + C \sum_j \sum_i \varepsilon_{ij} \tag{54}$$

subject to

$$\left(w_j^T x_i + b_j\right) \geq +1 - \varepsilon_{ij}, \ \forall i \in \text{Class 1 and } \forall j \tag{55}$$

$$\left.\begin{array}{l} \left(w_j^T x_i + b_j\right) \leq -1 + \varepsilon_{ij}, \ \forall j \leq k-1 \\ \left(w_j^T x_i + b_j\right) \geq +1 - \varepsilon_{ij}, \ \forall j \geq k \end{array}\right\} \forall i \in \text{Class k } (2 \leq \text{k} \leq \text{m-1}) \tag{56}$$

$$\left(w_j^T x_i + b_j\right) \leq -1 + \varepsilon_{ij}, \ \forall i \in \text{Class m and } \forall \text{j} \tag{57}$$

$$\varepsilon_{ij} \geq 0, \ \forall i \in \text{Class m and } \forall \text{j} \tag{58}$$

where m is the number of classes where Class 1 has the most-preferred alternatives and Class m has the least-preferred ones, x_i represents the alternative i in Class k. j represents the number of hyperplanes created ($1 \leq j \leq m$-1). The numbering of the hyperplanes starts from the rightmost hyperplane to the leftmost one. ε_{ij} represents the error relevant to alternative i according to hyperplane j.The model is created under the assumption of more is better for all of the criteria included.

Eq. (54) represents the objective function, used for maximizing the margin between the two closest points between two consecutive classes in addition to the objective of minimizing the sum of the errors corresponding to the misclassified points.

Eq. (55) – Eq. (57) ensure that all of the alternatives in Class k will be forced to be located on the *appropriate* side of the hyperplane created. Since the classes are ordinal, the hyperplanes are located accordingly between the consecutive classes. Positions of the alternatives are decided due to the class they belong to. These set of

constraints are also included in the classical SVM but are adjusted so an alternative is located according to its location to all of the hyperplanes simultaneously.

In SVM-S-v1, error related constraints are introduced to the model. The main aim is to investigate the potential ways of achieving less misclassified points. Additional constraints are explained below:

$\varepsilon_{ij} \geq \varepsilon_{it} \; \forall \, i \in$ Class k

where k ≤ m-2 , , j $\in \{k, \dots, m-2\}, t \in \{j+1, \dots, m-1\}$ (59)

$\varepsilon_{ij} \geq \varepsilon_{in} \; \forall i \in$ Class k

where k ≥ 3, j $\in \{2, \dots, k-1\}, n \in \{1, \dots, j-1\}$ (60)

This set of constraints force the errors to assure better geometric separation of the alternatives belonging to different classes. These equations are not included in the classical version of the SVM's. With these equations, in addition to previous set, the errors are also tried to be controlled again according to the ordinal classes that alternatives belong to. Due to the proposed positions of the alternatives, errors representing the distances between the misclassified points and the hyperplanes are tried to be organized geometrically.

In order to explain in detail, following example is presented. First of all, consider a set of alternatives grouped into three classes (m = 3), with two criteria as shown in Figure 10. Class 1 represents the best class where Class 3 has the least preferred set of alternatives. For each criterion, scores are assigned under the assumption of "more is better".

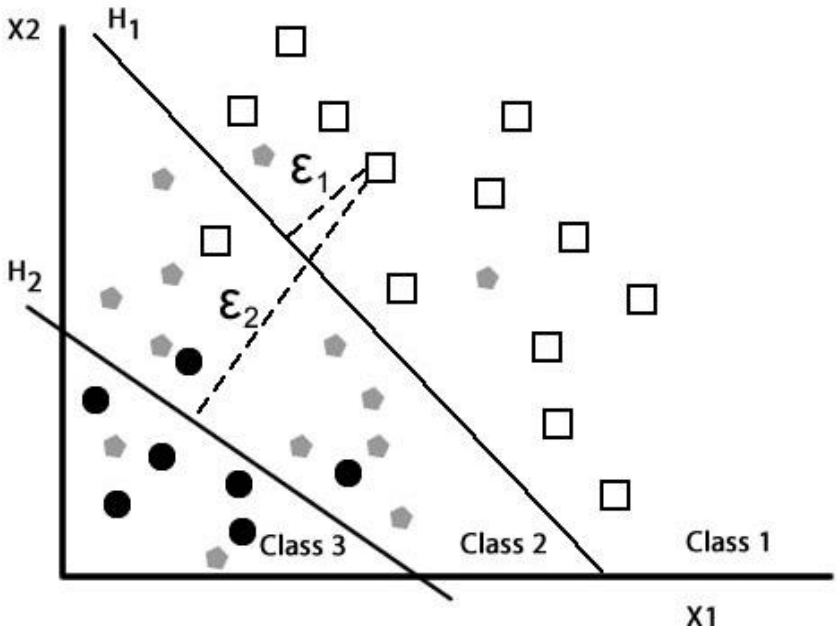

Figure 10 – Graphical representation of bicriteria problem with three classes

As being known, y_i in Eq. (19) represents the location of the point according to the hyperplane created. In first set of constraints added, Eq.(55) – Eq.(57), this property is used. This means;

- for Class 1, $y_i = +1$ for H1 and H2, $\forall$ i $\in$ Class 1.
- for Class 2, $y_i = -1$ for H1 and $y_i = +1$ for H2, $\forall$ i $\in$ Class 2
- for Class 3, $y_i = -1$ for H1 and H2, $\forall$ i $\in$ Class 3.

As stated above, all of the alternatives in Class q will be forced to stay on the *appropriate* side of the hyperplane by the utilization of equations (54)-(56).

Second set of constraints, equations Eq. (59) – Eq. (60) include the error based constraints. According to the example presented, "ε_{i1}" is the distance between a point representing an alternative and H1, which is positive if and only if corresponding i is a misclassified point. Within this point of view, according to the

preference based locations of the alternatives in the input space, following constraints are added to the model.

- for *Class 1*,
    - $\varepsilon_{i1} \geq \varepsilon_{i2}$, $\forall\, i \in$ Class 1. If ith alternative is on the correct side of the H1, $\varepsilon_{i1} = 0$, which also will mean $\varepsilon_{i2} = 0$. This set of constraints will force this condition unless $\varepsilon_{i1} > 0$, then the constraint will limit ε_{i2}. So ith alternative will be closer to H1 than H2.
- for *Class 2*, constraints related with errors are not valid. When the number of the classes and thus the number of the hyperplanes constructed is more than three, positioning of the alternatives are considered relatively. But the logic of the application of the constraints is the same.
- for *Class 3*,
    - $\varepsilon_{i2} \geq \varepsilon_{i1}$, $\forall\, i \in$ Class 3. This set of constraints forces Class 3 points to be closer to H2 than H1. If ith alternative is on the correct side of the H2, $\varepsilon_{i2} = 0$, which also will mean $\varepsilon_{i1} = 0$. This set of constraints will force this condition unless $\varepsilon_{i2} > 0$, then the constraint will limit ε_{i1}. So ith alternative will be closer to H2 than H1.

As seen, in the second set of constraints, positioning according to the hyperplanes are used with the aim of improving the effectiveness of the model. In order to utilize these constraints, more than one hyperplane is required. Consequently when a problem with two classes is considered, two imaginary hyperplanes will be added directly to the model.

Furthermore, with the aim of inspecting the effect of the imaginary hyperplanes, they are also introduced to the problems with three or more classes as an extension of SVM-S (SVM-S-v1-IH). For any number of classes, imaginary hyperplanes (IH), as shown in Figure 11, are used and results are also analyzed. One of these hyperplanes is located on the right hand side of Class 1 alternatives (i.e. all of the

alternatives in *m* classes are located on the negative side of this hyperplane) and the other one is located on the left hand side of Class *m* alternatives (i.e. all of the alternatives in m classes are located on the positive side of this hyperplane).

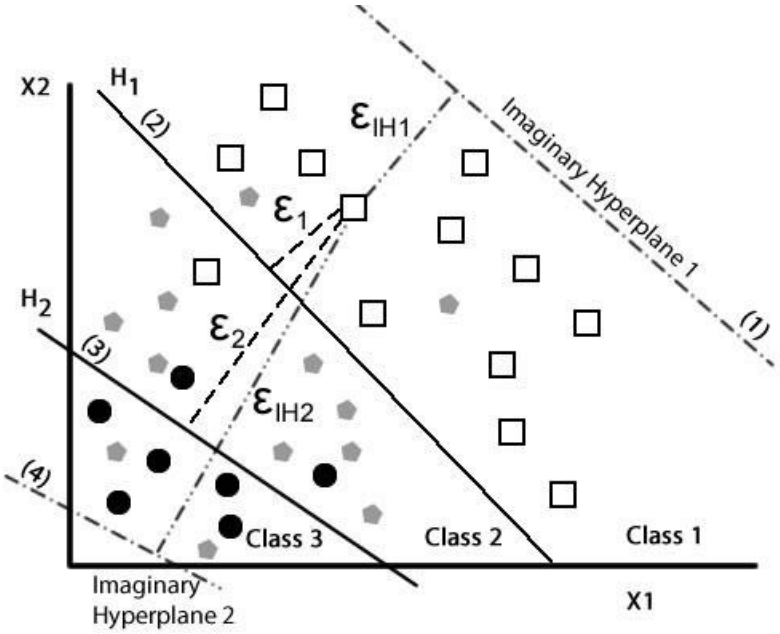

Figure 11 – Graphical representation of imaginary hyperplanes.

According to the locations of the imaginary hyperplanes,

- for IH1, y_i = - 1 for ∀ i ∈ Class 1, Class 2 and Class 3.
- for IH2, y_i = + 1 for ∀ i ∈ Class 1, Class 2 and Class 3.

Moreover, error related constraints are arranged according to the additional hyperplanes, as follows:

- for *Class 1*,

$\varepsilon_{i2} \geq \varepsilon_{i3}$, ∀ i ∈ Class 1.

$\varepsilon_{i2} \geq \varepsilon_{i4}$, ∀ i ∈ Class 1.

$\varepsilon_{i3} \geq \varepsilon_{i4}$, $\forall\, i \in$ Class 1.

- for *Class 2*,

$\varepsilon_{i2} \geq \varepsilon_{i1}$, $\forall\, i \in$ Class 2.

$\varepsilon_{i3} \geq \varepsilon_{i4}$, $\forall\, i \in$ Class 2.

- for *Class 3*,

$\varepsilon_{i3} \geq \varepsilon_{i2}$, $\forall\, i \in$ Class 3.

$\varepsilon_{i3} \geq \varepsilon_{i1}$, $\forall\, i \in$ Class 3.

$\varepsilon_{i2} \geq \varepsilon_{i1}$, $\forall\, i \in$ Class 3.

where hyperplanes, IH1, H_1, H_2 and IH2, are renumbered as 1,2,3and 4.

3.2.Training Data Selection

Training data is used to construct the hyperplanes separating the classes. Since the representativeness of the training data set directly affects the outcome of the mathematical model, in many applications training data set construction is one of the most time-consuming tasks.

With the aim of improving data selection performance, Wang. et al.(2005) study namely confidence based-data selection, Housdorff distance based data selection and random data selection. After some numerical experiments conducted on classification problems, it is concluded that, number of the samples taken must be in accordance with the class size. Sample size taken from each class often affects the performance of the outcome of the SVM model. Moreover random selection is also turned out to be performing consistently well when compared to other techniques considered.

Active learning or selective sampling is an approach that aims to decrease the number of training data set. It is achieved by selecting the samples that would bring more representative information. This approach is mainly used for the classification problems (Schohn, G., Cohn, D., 2000). There are studies for extending this method for ranking problems (Yu, H., 2005). But up to our knowledge, this approach has not been applied to sorting problems yet.

CHAPTER 4

COMPUTATIONAL EXPERIMENTS

4.1. Tools Used

Computer used for solving the models is HP 6530b with Intel (R) Core (TM) 2 Duo CPU T9600 @ 2.80 GHz processor, 4.00 GB RAM, 32 Bit OS and Windows 7 Operating system.

Software used is the General Algebraic Modeling System (GAMS), v. 23.3.3, with BARON solver, v. 9.0.2.

4.2. Data Sets

Data sets used within this study are retrieved from UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/contact.html).

During data set selection stage, there are some characteristics that the data sets must have. Data set is needed to have ordinal classes, where Class 1 represents the best class where Class n has the least preferred set of alternatives. Criteria are needed to be ordinal. Three data sets are found to be appropriate to be utilized during the computational experiments phase. Data sets used are Teaching Assistant Evaluation Data Set (named as "Assistant" hereafter), Car Evaluation Data Set (named as "Car" hereafter), and Statlog German Credit Data (named as "Credit" hereafter). These data sets will be briefly explained in the following sections.

In Table 1, total number of data, number of criteria, number of classes and number of pre-assigned data into the classes information regarding the utilized data sets is provided.

Table 1 – Data set summary

Data set name	**# of criteria**	**# of data in class**				**Total # of data**
		1	**2**	**3**	**4**	
Assistant	3	49	50	52		**151**
Car	6	65	69	384	1210	**1728**
Credit	20	300	700			**1000**

4.2.1. "Assistant" data set

In this data set, three criteria are defined. These criteria and the assigned scores respectively are as follows;

- Whether or not the assistant is a native English speaker;
  - It takes "1" if assistant is a non-native English speaker,
  - It takes "2" if assistant is a native English speaker,
- Whether the assistant is teaching in Summer or Regular semester;
  - It takes "1" if assistant is teaching in Summer semester,
  - It takes "2" if assistant is teaching in Regular semester,
- Class size(number of students registered to the class);
  - Normalized value of the class size between 0 and 1.

In each criterion "*more is better"* assumption holds.

Moreover, 151 alternative points are assigned to three ordinal classes according to the overall scores they get from each criterion;

- Class 1: High – 49 alternatives
- Class 2: Medium – 50 alternatives
- Class 3: Low – 52 alternatives.

4.2.2. “Car” data set

In this data set, six criteria are defined. These criteria and the assigned scores respectively are as follows are;

- Price of the alternative;
    - Assigned score is “1” when the alternative has “very high” price,
    - Assigned score is “2” when the alternative has “high” price,
    - Assigned score is “3” when the alternative has “medium” price,
    - Assigned score is “4” when the alternative has “low” price.
- Maintenance cost of the alternative;
    - Assigned score is “1” when the alternative has “very high” maintenance cost,
    - Assigned score is “2” when the alternative has “high” maintenance cost,
    - Assigned score is “3” when the alternative has “medium” maintenance cost,
    - Assigned score is “4” when the alternative has “low” maintenance cost.
- Number of doors of the alternative;
    - Assigned score is “1” when the alternative has 2 doors,
    - Assigned score is “2” when the alternative has 3 doors,
    - Assigned score is “3” when the alternative has 4 doors,
    - Assigned score is “4” when the alternative has more than doors,
- Number of person can be carried by the alternative;
    - Assigned score is “1” when the alternative can carry 2 persons,
    - Assigned score is “2” when the alternative can carry 4 persons,
    - Assigned score is “3” when the alternative can carry more than 4 persons,
- Luggage boot capacity of the alternative;

    - Assigned score is “1” when the luggage boot of the alternative is “small”,
    - Assigned score is “2” when the luggage boot of the alternative is “medium”,
    - Assigned score is “3” when the luggage boot of the alternative is “big”,
- Safety of the alternative;
    - Assigned score is “1” when the safety of the alternative is “low”,
    - Assigned score is “2” when the safety of the alternative is “medium”,
    - Assigned score is “3” when the safety of the alternative is “high”,

In each criterion “*more is better”* assumption holds

Moreover, 1728 alternative points are assigned to four ordinal classes according to the overall scores they get from each criterion;

- Class 1: Very good – 65 alternatives
- Class 2: Good – 69 alternatives
- Class 3: Acceptable – 384 alternatives
- Class 4: Unacceptable – 1210 alternatives.

4.2.3. “Credit” data set

In this data set, twenty criteria are defined. These criteria and the assigned scores respectively are as follows;

- Status of existing check account;
    - Assigned score is “1” when there exists no checking account,
    - Assigned score is “2” when there exists a checking account but no money put in,

    - Assigned score is “3” when there exists a checking account with at most 200 monetary value put in,
    - Assigned score is “4” when there exists a checking account with more than 200 monetary value put in,
- Duration of credit application in month;
    - Normalized value of the duration between 0 and 1,
- Credit history of the applicant;
    - Assigned score is “1” when applicant has critical account or owns other credits at other banks,
    - Assigned score is “2” when applicant has delay in paying off in the past,
    - Assigned score is “3” when applicant has paid back the existing credits duly till now,
    - Assigned score is “4” when applicant has paid back all credits at this bank duly,
    - Assigned score is “5” when applicant has paid back all credits duly or has taken no credits till now,
- Purpose of the credit application;
    - Assigned score is “1” when the purpose is a new car,
    - Assigned score is “2” when the purpose is a used car,
    - Assigned score is “3” when the purpose is furniture or is equipment,
    - Assigned score is “4” when the purpose is radio or is television,
    - Assigned score is “5” when the purpose is domestic appliances,
    - Assigned score is “6” when the purpose is repair,
    - Assigned score is “7” when the purpose is education,
    - Assigned score is “8” when the purpose is vacation,
    - Assigned score is “9” when the purpose is retraining,
    - Assigned score is “10” when the purpose is a business,
- Amount of the credit application;
    - Normalized value of the amount between 0 and 1,

- Amount of saving accounts/bonds of the applicant;
    - Assigned score is “1” when there is no information or there exists no savings account of the applicant,
    - Assigned score is “2” when the applicant’s saving account is below 100 monetary value,
    - Assigned score is “3” when the applicant’s saving account is between 100 and 500 monetary value,
    - Assigned score is “4” when the applicant’s saving account is between 500 and 1000 monetary value,
    - Assigned score is “5” when the applicant’s saving account is above 1000 monetary value,
- Employment status of the applicant;
    - Assigned score “1” if the applicant is unemployed,
    - Assigned score “2” if the applicant is employed with less than 1 year duration of working,
    - Assigned score “3” if the applicant is employed with more than 1 year and less than 4 year duration of working,
    - Assigned score “4” if the applicant is employed with more than 4 year and less than 7 year duration of working,
    - Assigned score “5” if the applicant is employed with more than 7 year duration of working,
- Installment rate in percentage of disposable income
    - Normalized value of the rate between 0 and 1.
- Marital status and sex of the applicant;
    - Assigned score is “1” if the applicant is male and is divorced or separated,
    - Assigned score is “2” if the applicant is female and is divorced or separated or married,
    - Assigned score is “3” if the applicant is male and is single,

    - Assigned score is “4” if the applicant is male and is married or widowed,
    - Assigned score is “5” if the applicant is female and is single,
- Whether or not there exist other debtors or guarantors;
    - Assigned score is “1” if there exists none,
    - Assigned score is “2” if there exists co-applicant,
    - Assigned score is “3” if there exists guarantor,
- Duration of the residence of the applicant;
    - Normalized value of the duration between 0 and 1,
- Properties belong to the applicant;
    - Assigned score is “1” if there is no information or there exists no property belong to the applicant,
    - Assigned score is “2” if the applicant has a car, etc. not mentioned in criteria,
    - Assigned score is “3” if the applicant has building society savings agreement or has life insurance,
    - Assigned score is “4” if the applicant has real estate,
- Age of the applicant;
    - Normalized value of age between 0 and 1.
- Whether or not there exist other installment plans of the applicant;
    - Assigned score is “1” if the applicant has installment plans to bank,
    - Assigned score is “2” if the applicant has installment plans to stores,
    - Assigned score is “3” if the applicant has no installment plans,
- Housing information of the applicant;
    - Assigned score is “1” if the applicant’s house is rent,
    - Assigned score is “2” if the applicant owns the house,
    - Assigned score is “3” if the applicant’s house is for free,
- Number of existing credits of the applicant at this bank;
    - Normalized value of number between 0 and 1,
- Job of the applicant;

    - Assigned score is “1” if the applicant is unemployed or is unskilled-non-resident,
    - Assigned score is “2” if the applicant is unskilled-resident,
    - Assigned score is “3” if the applicant is skilled employee or is official,
    - Assigned score is “4” if the applicant is a manager or is self-employed or is a highly qualified employee or is an officer,
- Number of people being liable to provide maintenance for the applicant;
    - Assigned score is “1” if the number is one,
    - Assigned score is “2” if the number is two,
- Whether or not the applicant has telephone;
    - Assigned score is “1” if the applicant does not own a telephone,
    - Assigned score is “2” if the applicant owns a telephone,
- Whether or not the applicant is a foreign worker;
    - Assigned score is “1” if the applicant is a foreign worker,
    - Assigned score is “2” if the applicant is not a foreign worker.

In each criterion “*more is better”* assumption holds

Moreover, 1000 alternative points are assigned to two ordinal classes according to the overall scores they get from each criterion;

- Class 1: Approved – 700 alternatives
- Class 2: Not Approved – 300 alternatives

See Appendix-A, Appendix-B and Appendix-C for the detailed information about the data sets.

4.3. Results

4.3.1. Classical and SVM-S Applications

Each data set is solved with classical SVM and then with SVM–S-v0, SVM-S-v1 and SVM-S-v1-IH. In the following phases, data sets are mapped by utilization of Polynomial Kernel and Gaussian Kernel.

First, constant "C" in the objective function, i.e., Equation (49), is set to "1"to give the priority to the first objective. This provides the problem to minimize the first objective first and select the solution that minimizes the second objective among alternative solutions.

Moreover with the aim of increasing the effectiveness of the second set of constraints, two imaginary hyperplanes are introduced to SVM – S applications and results are reported separately.

At each run, number of misclassified points is stated as the primary performance measure. An alternative is decided to be misclassified if the related error value is found to be non-zero. In addition, total sum of the errors are reported as the secondary performance measure of the models. Computation times are also reported as a supplementary comparison measure of value.

When the alternative points have inconsistencies according to the proposed position, it is stated as "unclassified". It is decided due to the errors assigned according to the hyperplanes created. Number of unclassified points is also presented with the aim of providing additional performance measure. These points can be classified by interacting with the Decision Maker using an interactive approach. This will be a subject of a future study.

The number of the misclassified points, total sum of the errors and the computation times are compared for each data set. For each data set, results of each run are summarized in the following tables. Detailed information can be found in Appendices – E, F, G and H.

In the tables, under the "model" column it is stated if the model is SVM or SVM – S. In the "explanation" column, the versions of the models are stated. Applied kernel is stated in the third column followed by the number of misclassified points observed after the models are run in the fourth column. In the fifth column, percentage of the misclassified points according to the sample size of the data set is presented. In the sixth and seventh column, number and percentage of the unclassified points are stated respectively while reporting the computation time in the last column.

Table 2 – Comparisons of the results of "Assistant" Data Set

Model		**Kernel**	**# of misclssfd.**	**% of misclssfd.**	**# of unclssfd.**	**Sum of errors**	**Computation time**
Classical		-	102	67.55%	38	255.748	24.607 sec
		poly	91	60.26%	0	181.261	17 min 32.527 sec
		Gaus	103	68.21%	45	253.524	3 min 53.924 sec
SVM_S	v0	-	115	76.16%	0	181.000	1 min 20.870 sec
		poly	**91**	**60.26%**	**0**	**90.442**	**3 min 5.315 sec**
		Gaus	111	73.51%	5	185.152	48.382 sec
SVM_S	v1	-	119	78.81%	1	181.000	18.137 sec
		poly	91	60.26%	0	90.441	7 min 37.466 sec
		Gaus	112	74.17%	0	188.387	26.134 sec
SVM_S	IH	-	106	70.20%	0	75.000	57.131 sec
		poly	91	60.26%	0	90.442	24 min 34.872 sec
		Gaus	119	78.81%	0	188.387	1 min 0.211 sec

In "Assistant" data set, all of the SVM–S models have better results than the classical model in each of three runs. The best result is found with SVM-S-v0 with polynomial kernel applied. Although number of misclassified points is equal to each other with polynomial kernel applied models, when total sum of errors and computation time are also taken into account, SVM-S-v0 with polynomial kernel applied has less summation of errors when compared to classical model. Within the

SVM-S models with polynomial kernel applied although the total sums of errors are equal to each other, SVM-S-v0 with polynomial kernel applied has less computational time. In all of the models with polynomial kernel applied, there exists no unclassified point.

Table 3 – Comparisons of the results of "Car" Data Set

Model		**Kernel**	**# of misclssfd.**	**% of misclssfd.**	**# of unclssfd.**	**Sum of errors**	**Computation time**
Classical			601	34.78%	72	1236.580	2 hrs 19 min 44.926 sec
		poly	215	12.44%	5	368.008	6 hrs 50 min 14.000 sec
		Gaus	444	25.69%	59	667.759	2 hrs 51 min 30.000 sec
SVM_S	v0		664	38.43%	8	682.620	1 hrs 40 min 52.664 sec
		poly	**201**	**11.63%**	**6**	**188.189**	**2 hrs 44 min 30.000 sec**
		Gaus	451	26.10%	9	332.327	2 hrs 38 min 42.506 sec
SVM_S	v1		671	38.83%	6	683.290	5 hrs 48 min 24.383 sec
		poly	204	11.81%	1	189.044	2 hrs 43 min 50.000 sec
		Gaus	467	27.03%	4	336.043	5 hrs 58 min 13.810 sec
SVM_S	IH		672	38.89%	5	685.173	16 hrs 19 min 31.759 sec
		poly	202	11.69%	0	189.898	6 hrs 17 min 49.000 sec
		Gaus	470	27.20%	3	337.522	12 hrs 19 min 35.210 sec

In “Car” data set experiments, the SVM–S models have better results again. Especially there exist significantly less unclassified points with the models with no kernel applied and with Gaussian kernel applied. Best result is found with SVM-S-v0 with polynomial kernel applied. It has significant improvement in performance when number of misclassified points and total sum of errors are taken into account, while computation time is less than SVM classical model with polynomial kernel applied, which has the best result among SVM classical applications. Models with polynomial kernel applied seem to have better results when both number of misclassified and number of unclassified points are considered.

Table 4 – Comparisons of the results of “Credit” Data Set

Model		**Kernel**	**# of misclssfd.**	**% of misclssfd.**	**# of unclssfd.**	**Sum of errors**	**Computation time**
Classical			579	57.90%	-	1169.286	9 min 42.000 sec
		poly	0	0.00%	-	0.000	55 min 0.000 sec
		Gaus	590	59.00%	-	1187.638	1 hrs 11 min 0.000 sec
SVM_S	v0		579	57.90%	-	584.643	16 min 20.125 sec
		poly	0	0.00%	-	0.000	26 hrs 58 min 26.000 sec
		Gaus	591	59.10%	-	593.824	32 min 19.000 sec
SVM_S	IH		584	58.40%	-	589.037	3 hrs 12 min 03.129 sec
		poly	0	0.00%	-	0.000	16 hrs 47 min 39.000 sec
		Gaus	590	59.00%	-	593.831	8 hrs 4 min 39.640 sec

For “Credit” data set experiments, it is seen that, all of the models result with no unclassified point. Moreover again models with polynomial kernel applied seem to have significantly better results, which is no misclassified point. Although no misclassified point is reached by all of the three models with polynomial kernel applied, computation time of the classical model is less than those of the other models.

To summarize, SVM – S models end up with promising results for the data sets used in the computational experiments. More specifically SVM – S models with polynomial kernel are found to be promising in the experiments. Note that proper kernel selection according to the characteristics of the data set has positive effect on separation in theory.

In addition, due to the expansion in the dimensions of the models by introducing new sets of constraints and new hyperplanes, it is expected to have longer computation times. This difference does not become more significant as the volume of the data set increases. The results support these expectations in most of the cases.

4.3.2. UTADIS Applications

The results of the classical SVM method and the results of the SVM-S are compared with the results of the UTADIS method.

For the “Assistant” data set, $g_i^{r_{ji}}$, i.e. break points of subinterval related to r_{ji} is as follows:

Table 5 – Breakpoints of subintervals of “Assistant” data set

$g_i^{r_{ji}}$	**1**	**2**	**3**	**4**
1	0	0.25	0.5	0.75
2	0	0.25	0.5	0.75
3	1	2	3	4

For the “Car” data set, $g_i^{r_{ji}}$, i.e. break points of subinterval related to r_{ji} table is as follows:

Table 6 – Breakpoints of subintervals of “Car” data set

$g_i^{r_{ji}}$	**1**	**2**	**3**	**4**
1	1	2	3	4
2	1	2	3	4
3	1	2	3	4
4	1	2	3	
5	1	2	3	
6	1	2	3	

For the “Credit” data set, $g_i^{r_{ji}}$, i.e. break points of subinterval related to r_{ji} table is as follows:

Table 7 – Breakpoints of subintervals of “Credit” data set

$g_i^{r_{ji}}$	**1**	**2**	**3**	**4**	**5**
1	1	2	3	4	5
2	0	0.2	0.4	0.6	0.8
3	1	2	3	4	5
4	0	2	4	6	8
5	0	0.2	0.4	0.6	0.8
6	1	2	3	4	5
7	1	2	3	4	5
8	1	2	3	4	5
9	1	2	3	4	5
10	1	2	3	4	5
11	0	0.2	0.4	0.6	0.8
12	1	2	3	4	5
13	0	0.2	0.4	0.6	0.8
14	1	2	3	4	5
15	1	2	3	4	5
16	1	2	3	4	5
17	1	2	3	4	5
18	1	2	3	4	5
19	1	2	3	4	5
20	1	2	3	4	5

The results of the UTADIS method applications presented in Table 8, below.

Table 8 – Results of the UTADIS Method application.

Data set	# of data points	# of criteria	# of classes	# of misclssfd. point	Total sum of errors	Computation time
Assistant	151	3	3	5	0.05	0.304 sec
Car	1728	6	4	47	0.47	1.844 sec
Credit	1000	20	2	0	0	2.230 sec

As seen from the results, UTADIS applications for these three data sets have better results. Only for "Credit" data set both methods achieve no misclassified points. UTADIS model has significantly less computation times.

4.4. Real Life Application

In this section, the method is applied to data set retrieved from a telecommunications company. Data set is assigned to five pre - defined classes at first. Afterwards, data set is divided into two groups randomly. 50% of the data from each class is used to train the models (i.e. training data set). Within this phase, models are solved and hyperplanes are created respectively. Number of misclassified points is defined as a primary performance measure as in the experiments realized on the previous data sets.

Then in the second phase, rest of the alternatives (i.e. test data set) is assigned to the "proposed" classes according to the positions of the alternatives with respect to the hyperplanes created in the first phase. Then pre-assigned class and proposed class of the testing data set are compared and misclassified points of testing data set are also defined as an additional performance measure.

As done in the previous experiments. Classical SVM, SVM – S - v0, SVM – S – v1 and SVM – S – v1 - IH models are implemented on the training data set. Moreover, data set is mapped by utilization of Polynomial Kernel and Gaussian Kernel. The results are compared within each other and also with the results obtained from UTADIS method application.

4.4.1. "Telecom" data set

In Table 9, total number of data, number of criteria and number of class information regarding the "Telecom" data set is provided.

Table 9 - Data set summary of "Telecom"

Data set name	**# of criteria**	**# of data in class**					**Total # of total data**
		1	**2**	**3**	**4**	**5**	
Telcom	7	188	238	670	1652	3385	**6133**

In this data set, seven criteria and scores to be assigned are defined by the telecommunications company. These criteria and the assigned scores are as follows;

- Total bandwidth of the products corporate customer is using;
  - Normalized value of number between 0 and 1,
- Whether or not the corporate customer has data product;
  - Assigned score is "0" if the corporate customer has not data product,
  - Assigned score is "1" if the corporate customer has data product,
- Whether or not the corporate customer has voice product;
  - Assigned score is "0" if the corporate customer has not voice product,
  - Assigned score is "1" if the corporate customer has voice product,
- Total number of the products corporate customer is using;
  - Normalized value of number between 0 and 1,

- Segment of the corporate customer;
    - Assigned score is “1” if the customer is from “government” segment,
    - Assigned score is “2” if the customer is from “large business” segment.
- Sector of the corporate customer;
    - Assigned score is “1” if the customer is from “embassy/consulate or non-profit organization” sector,
    - Assigned score is “2” if the customer is from “sports” sector,
    - Assigned score is “3” if the customer is from “construction” sector,
    - Assigned score is “4” if the customer is from “automotive” sector,
    - Assigned score is “5” if the customer is from “technology or media” sector,
    - Assigned score is “6” if the customer is from “logistics or retail” sector,
    - Assigned score is “7” if the customer is from “municipality, travel/tourism or energy” sector,
    - Assigned score is “8” if the customer is from “health, service or manufacturing” sector,
    - Assigned score is “9” if the customer is from “education, insurance or holding” sector,
    - Assigned score is “10” if the customer is from “bank or strategy” sector,
- Total amount of the bill accrued in June 2010 belong to corporate customer;
    - Normalized value of number between 0 and 1,

In each criterion “*more is better”* assumption holds

Moreover, 6133 alternative points are assigned to five ordinal classes according to the overall scores they get from each criterion;

- Class 1: Very high
- Class 2: High
- Class 3: Medium
- Class 4: Moderate
- Class 5: Low

4.4.2. Results of the real life application

Results of the first phase are presented in Table 9 below. Table has the design similar to the results tables belong to the previous experiments. Model names are stated under the “model” column. In the “explanation” column, the versions of the models are stated. Applied kernel is stated in the third column followed by the number of misclassified points observed after the models are run in the fourth column.

In the fifth column, percentage of the misclassified points according to the sample size of the data set is presented. In the sixth and seventh column, number and percentage of the unclassified points are reported respectively and in the last column the computation times are presented.

Table 9 – Results of the applications on "Telecom" data set

Model		Kernel	# of misclssfd. points in training data set	% msclassfd point of training data	# of unclssfd. points in training data set	% unclssfd. point of training data	Sum of errors	Computation time
Classical		-	1791	58.38%	1264	41.20%	1262343.492	28 hrs 20 min 9.955 sec
		poly	747	24.35%	1974	64.34%	1985.824	55 hrs 48 min 57.342 sec
		Gaus	2197	71.61%	795	25.91%	1139934.756	49 hrs 35 min 9.723 sec
SVM-S	v0	-	1790	58.34%	100	3.26%	3704.424	15 hrs 59 min 29.363 sec
		poly	1176	38.33%	1	0.03%	3439.896	16 hrs 23 min 28.605 sec
		Gaus	1995	65.03%	191	6.23%	4135.408	15 hrs 55 min 26.550 sec
SVM-S	v1	-	916	29.86%	103	3.36%	587.354	15 hrs 26 min 51.103 sec
		poly	**73**	**2.38%**	**0**	**0.00%**	**14.985**	**13 hrs 11 min 10.000 sec**
		Gaus	1475	48.08%	238	7.76%	1366.428	26 hrs 59 min 25.000 sec
SVM-S	IH	-	1147	37.39%	104	3.39%	585.365	12 hrs 54 min 33.000 sec
		poly	-	-	-	-	-	could not be solved
		Gaus	-	-	-	-	-	could not be solved
UTADIS			1298				18.330	6.190 sec

The best result is obtained with SVM-S-v1 with polynomial kernel applied. The result is significantly better when compared with other SVM models and UTADIS method. There exists no unclassified point while total sum of error is much less than the other models in addition to the successive computation time.

For the UTADIS application, break points of subinterval related to r_{ji} table is constructed as follows:

Table 10 – Breakpoints of subintervals of “Telecom” data set

	1	**2**	**3**	**4**
1	0	0.25	0.5	0.75
2	0	0.25	0.5	0.75
3	0	0.25	0.5	0.75
4	0	0.25	0.5	0.75
5	0	0.5	1	1.5
6	0	2.5	5	7.5
7	0	0.25	0.5	0.75

Furthermore, as mentioned in the introduction of this section, 50% of the “Telecom” data from each class is used to train the models (i.e. training data set). According to the results of the models, hyperplanes are created. Alternatives in the “test data set” are assigned to the “proposed” classes according to the positions of the alternatives with respect to the hyperplanes created. Then pre-assigned class and proposed class of the testing data set are compared and misclassified points of testing data set are also counted as an additional performance measure. Results are reported in Table 11.

Table 11 – Results of the allocations of the testing data

Model		Kernel	# of misclssfd. points of testing data	% misclassfd. point of testing data	# of unclssfd. points in testing data set	% unclssfd. point of testing data
Classical		-	2287	74.57%	587	19.14%
		Poly	0	0.00%	3059	99.74%
		Gaus	0	0.00%	3068	100.03%
SVM-S	v0	-	0	0.00%	3068	100.00%
		Poly	826	26.93%	552	17.99%
		Gaus	1374	44.80%	0	0.00%
SVM-S	v1	-	1981	64.59%	238	7.76%
		Poly	2068	67.43%	0	0.00%
		Gaus	0	0.00%	1374	44.78%
SVM-S	IH	-	2974	96.97%	0	0.00%
		Poly	-	-	-	-
		Gaus	-	-	-	-

As seen from the results, the best outcome is achieved from the hyperplanes created by SVM – S – v1 with Gaussian kernel, when number of misclassified points is thought as the primary performance measure and number of unclassified points as the secondary.

CHAPTER 5

CONCLUSION AND DISCUSSIONS

The main interest of this study is sorting problems with predefined ordinal classes. A new method is proposed based on Support Vector Machine (SVM) model, which is mainly used for nominal binary classification processes. Even some SVM models are costructed for multi-class purposes, they cannot be utilized for ranking or sorting problems as they are not constructed as combination of several binary SVMs. Basically, model is solved for each class of alternatives and the results are combined to decide the actual class of the alternative. However again there is no prioritization between the classes. Moreover as stated in Schölkopf and Smola (2002, pp. 211-214), it is hard to say that one of the multi-class SVM methods always outperforms the others for classification problems.

Although there are some studies conducted with the aim of extending the SVM's to be utilized in ranking problems, there are no studies that address the utilization of SVMs on sorting problems.

In this study, the model, called as SVM – Sorting (SVM-S-v0), is developed to take into account the preferences of the decision maker and the ordinal relationship between classes. , Constraints in the the classical Multi-class SVM's are modified so that an alternative is located according to its location to all of the hyperplanes simultaneously.

In addition to SVM-S, with the aim of forcing the alternatives to be located correctly new sets of constraints are added to the SVM model (SVM-S-v1).

Furthermore, with the aim of analysing the effect of the imaginary hyperplanes, they are also introduced into the models for problems with three or more classes as an extension of SVM-S (SVM-S-v1-IH).

Computational experiments show that proposed method performs equivalent or better than the classical SVM model applications even when the kernel tricks are applied in terms of number misclassified points and unclassified points.

Moreover, it is observed that every kernel is not appropriate to every data set. We also observe that proper kernel selection according to the characteristics of the data set has positive effect on separation.

In addition, due to the expansion in the dimensions of the models by introducing new sets of constraints and new hyperplanes, it is expected to have longer computation times. But this difference does not always become significant as the volume of the data set increases.

Although proposed SVM models produce promising results in the experiments, UTADIS model performs significantly better on the data sets used.

In order to provide additional experiments, a real life case is studied. It is observed that the proposed SVM model, especially SVM-S-v0 and SVM-S-v1 with polynomial kernel applied, achieves significantly better outcome. Moreover, UTADIS method produces better results in data sets other than real life application.

SVM – S is found promising in the computational experiments and the real life application. To obtain more concrete results, the proposed models may be applied to more data sets of different characteristics, i.e. different number of classes, different number of criteria, etc as a future work.

REFERENCES

Boser, B. E., Guyon, I. M., Vapnik, V. N., 1992. A Training Algorithm for Optimal Margin Classifiers. *Annual Workshop on Computational Learning Theory,* pp. 144 -152. Pittsburgh, Pennsylvania, USA.

Brans, J.P., 1982. L'ingéniérie de la décision. Elaboration d'instruments d'aide à la décision. Méthode PROMETHEE", In: R. Nadeau and M. Landry, eds, *L'aide à la décision: Nature, Instruments et Perspectives d'Avenir*, Québec, Canada: Presses de l'Université Laval. pp. 183–213

Center for Machine Learning and Intelligent Systems, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/contact.html, last visited on September 2010.

Chen, Y., Hipel, K. W., Kilgour, D. M., 2008. A Multiple Criteria Sequential Sorting Procedure. *Journal of Industrial and Management Optimization,* Vol. 4 (Number 3), pp. 407–423

Cristianini, N., Shawe-Taylor, J., 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Methods.* United Kingdom: Cambridge University Press.

Cristianini, N., Shawe-Taylor, J., 2004. *Kernel Methods for Pattern Analysis.* United Kingdom: Cambridge University Press.

Dias, J. A., Figueira, J. R., Roy, B. *Electre Tri-C: A Multiple Criteria Sorting Method Based on Central Reference Actions*. Universite Paris Dauphine, Centre National de la Recherche Scientifique, 2008. 33 p. Report no: hal-00281307, Ver. 1.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer

tissue samples using microarray expression data. *Bioinformatics* Vol. 16 no. 10, pp. 906-914.

Greco, S., Mousseau, V., Slowinski, R. *Multiple criteria sorting with a set of additive value functions*. Universite Paris Dauphine, Centre National de la Recherche Scientifique, 2008. 35 p.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning Journal,* Volume 46 (Numbers 1-3), pp. 389-422.

Heisele, B., Ho, P., Poggio, T., 2001. Face Recognition with Support Vector Machines: Global versus Component-based Approach. In: *Eighth International Conference on Computer Vision (ICCV'01)*. July 7-14, 2001, Vancouver, British Columbia, Canada.

Joachims, T., 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In: *Lecture Notes in Computer Science - Machine Learning: ECML-98.* Heidelberg: Springer Berlin. pp. 137-142.

Joachims, T., 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Dordrecht: Kluwer Academic Publishers.

Keeny, R.L. , Raiffa, H., 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* United Kingdom: Cambridge University Press.

Kijsirikul, B., Ussivakul, N., Meknavin, S., 2002. Adaptive Directed Acyclic Graphs for Multiclass Classification. In: *Lecture Notes in Computer Science,* Volume 2417/2002, Springer, pp. 423-430.

Osuna, E., Freund, R., Girosi, F., 1997. Training Support Vector Machines: an Application to Face Detection. In: *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97).* June 17-19, 1997, Puerto Rico.

Pang, B., Lee, L., Vaithyanathan, S. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In: SIGDAT and Association for Computational Linguistics, *EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* - Volume 10, (pp. 79 - 86). University of Pennsylvania, July 6-7, 2002. Philadelphia, PA, USA.

Platt, J. C., Cristianini, N., Shawe-Taylor, J., 2000. Large Margin DAGs for Multiclass Classification. In: S.A. Solla, T.K. Leen, and K. R. Müller, eds., *Advances in Neural Information Processing Systems 12*. USA: MIT Press. pp. 526-532.

Pontil, M., Verri, A., 1998. Support Vector Machines for 3D Object Recognition. *IEEE Trabsactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, pp. 637 - 646.

Saaty, T., 1980. *The Analytic Hierarchy Process*. New York: McGraw-Hill.

Schohn, G., Cohn, D., 2000. Less is more: Active learning with support vector machines. In: *Proc. Int. Conf. Machine Learning (ICML'00)*, pp. 839–846.

Schölkopf, B., Smola, A. J., 2002. *Learning with Kernels - Support Vector Machinesi Regularization, Optimization, and Beyond*. London: The MIT press.

Sewell, M., 2007. *Kernel Methods*. Department of Computer Science, University College London, unpublished.

Tay, F. E.H., Cao, L. C., 2001. Application of support vector machines in financial time series forecasting. *Omega - International Journal of Management Science* , pp. 309-317.

Tong, S., Chang, E., 2001. Support vector machine active learning for image retrieval. In: *International Multimedia Conference*; Vol. 9 (pp. 107 - 118). New York: ACM.

Tong, S., Koller, D., 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, Volume 2, pp. 45-66.

Van Gestel, Tony, et al., 2001. Financial Time Series Prediction Using Least Squares Support Vector Machines Within the Evidence Framework. *IEEE Transactions on Neural Networks*, Volume 12 (Number 4), pp. 809-821.

Vapnik, V. N., 1995. *The Nature of Statistical learning Theory*. NY, USA: Springer-Verlag Inc.

Wallenius, J., Dyer, J. S., Fishburn, P. C., Steuer, R. E., Zionts, S., Deb, K., 2008. Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead. *Institute for Operations Research and the Management Sciences (INFORMS), Management Science Archive* Volume 54 (Issue 7), pp. 1336-1349.

Wang, J., Predrag, N., Cooper, L. N., 2005. Training Data Selection for Support Vector Machines. In: Lecture Notes in Computer Science (LNCS). Heidelberg: Springer Berlin.Yevseyeva, I. 2007. *Solving Classification Problems with MultiCriteria Decision Aiding Approaches.* Jyvaskyla: Jyvaskyla University Printing House.

Weston, J., Watkins, C., 1999. Multi-class support vector machines. In: M. Verleysen, ed, Proceedinga ESANN, Brussels, D Facto.

Yevseyeva, I., Miettinen, K., Salminen, P., Lahdelma, R., 2007. SMAA-Classification – A New Method for Nominal Classification. In: *Helsinki School of Economics – Working Papers W-422.* HSE Print: Finland. p. 22.

Yu, H., 2005. SVM selective sampling for ranking with application to data retrieval. In: *Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD'05)*

Zopounidis, C., Doumpos, M., 2002. Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research,* Vol. 138, pp. 229-246.

Zopounidis, C., Doumpos, M., 2002. *Multicriteria decision aid classification methods.* The Netherlands: Kluwer Academic Publishers.

Zopounidis, C., Doumpos, M., 2003. Multi-criteria decision aid in financial decision making: methodologies and literature review. *Journal of Multi-Criteria Decision Analysis,* Vol. 11 (Issue 4-5), pp. 167 – 186.

APPENDIX A

DETAILED INFORMATION FOR “ASSISTANT” DATA SET

Table A - Assistant Evaluation Data set "Assistant"

Criteria	**Scores assigned to the criteria**	
native english speaker	no / 1	yes / 2
summer/regular semester	summer /1	regular /2
class size	numerical values (normalized)	

APPENDIX B

DETAILED INFORMATION FOR “CAR” DATA SET

Table B - Car Evaluation Data set "Car"

Criteria	**Scores assigned to the criteria**			
buying	Very high	high	med	low
maintenance	Very high	high	med	low
doors	2	3	4	more
persons	2	4	more	-
lug_boot	small	med	big	-
safety	low	med	high	-
Scores	1	2	3	4

APPENDIX C

DETAILED INFORMATION FOR “CREDIT” DATA SET

Table C - Credit Evaluation Data set "Credit"

	Criteria	**Scores assigned to the criteria**				
1	Status of existing checking account	no checking account /1	(... < 0 MV)/2	(0 <= ... < 200 MV)/3	(... >= 200 MV)/4	
2	Duration in month	numerical values *				
3	Credit history	Critical account/other credits existing (not at this bank)/1	delay in paying off in the past/2	existing credits paid back duly till now/3	all credits at this bank paid back duly/4	no credits taken/all credits paid back duly/5
4	Purpose	car (new)/1	car (used)/2	furniture/equipment/3	radio/television/4	domestic appliances/5
		repairs/6	education/7	vacation/8	retraining/9	business/10
5	Credit amount	numerical values				
6	saving account/bonds	unknown/ no savings account/1	... < 100 MV/2	100 <= ... < 500 MV/3	500 <= ... < 1000 MV/4	.. >= 1000 MV/5

Table C – Continued

7	Present employment since	unemployed/1	... < 1 year/2	1 <= ... < 4 years/3	4 <= ... < 7 years/4	.. >= 7 years/5
8	Installment rate in percentage of disposable income	numerical values*				
9	Personal status and sex	male: divorced/ separated/1	female: divorced/ separated/married/2	male: single/3	male: married/ widowed/4	female: single/5
10	Other debtors / guarantors	none/1	co-applicant/2	guarantor/3		
11	Present residence since	numerical values				
12	Property	unknown / no property/1	car or other, not in attribute 6/2	building society savings agreement/life insurance/3	real estate/4	
13	Age in years	numerical values				
14	Other installment plans	bank/1	stores/2	none/3		
15	Housing	rent/1	own/2	for free/3		
16	Number of existing credits at this bank	numerical values *				
17	Job	unemployed/ unskilled - non-resident/1	unskilled - resident/2	skilled employee / official/3	management/ self-employed/highly qualified employee/ officer/4	

Table C – Continued

18	Number of people being liable to provide maintenance for	1	2			
19	Telephone	none/1	yes, registered under the customer's name/2			
20	foreign worker	yes/1	no/2			

* These numerical values have "less is better" characteristic. There are normalized to have "more is better" characteristic, in order to be eligible for the model to be applied on.

APPENDIX D

DETAILED INFORMATION FOR "TELECOM" DATA SET

Table D – Corporate Customer Data Set of a Telecommunication Company "Telecom"

	Criteria	**Scores assigned to the criteria**	
1	Total bandwidth	Normalized numerical values *	
2	Whether or not the corporate customer has data product	"1" if customer has	"2" if customer does not have
3	Whether or not the corporate customer has voice product	"1" if customer has	"2" if customer does not have
4	Total number of the products	Normalized numerical values *	

Table D – Continued

5	Segment of the corporate customer	"1" for embassy/consulate or non-profit organization	"2" for sports	"3" for construction	"4" for automotive	"5" for technology or media
		"6" for logistics or retail	"7" for municipality, travel/tourism or energy	"8" for health, service or manufacturing	"9" for education, insurance or holding	"10" for bank or strategy
6	Sector of the corporate customer	"2" if "large business"	"1" if "government"			
7	Total amount of the bill accrued in June 2010 belong to corporate customer	Normalized numerical values *				

APPENDIX E

DETAILED INFORMATION FOR RESULTS OF "ASSISTANT" DATA SET

Table E – Results of the experiments on "Assistant"

Model	Explanation	Kernel	# of misclassified	% of misclassified	# of unclassified	Sum of errors	wTw	Computation time
SVM	Classical		102	67.55%	38	255.748	18.20	24.607 sec
		poli	91	60.26%	0	181.261	12.29	17 min 32.527 sec
		gaus	103	68.21%	45	253.524	70.89	3 min 53.924 sec
SVM_S	v0		115	76.16%	0	181.000	2.00	1 min 20.870 sec
		poli	91	60.26%	0	90.442	3.42	3 min 5.315 sec
		gaus	111	73.51%	5	185.152	9.07	48.382 sec
SVM_S	v1		119	78.81%	1	181.000	2.00	18.137 sec
		poli	91	60.26%	0	90.441	3.42	7 min 37.466 sec
		gaus	112	74.17%	0	188.387	9.07	26.134 sec
SVM_S	IH		106	70.20%	0	75.000	2.00	57.131 sec
		poli	91	60.26%	0	90.442	3.42	24 min 34.872 sec
		gaus	119	78.81%	0	188.387	9.07	1 min 0.211 sec

APPENDIX F

DETAILED INFORMATION FOR RESULTS OF "CAR" DATA SET

Table F – Results of the experiments on "Car"

Model	Explanation	Kernel	# of misclassified	% of misclassified	# of unclassified	Sum of errors	wTw	Computation time
SVM	Classical		601	34.78%	72	1236.580	88.796	2 hrs 19 min 44.926 sec
		poli	215	12.44%	5	368.008	109.129	6 hrs 50 min 14.000 sec
		gaus	444	25.69%	59	667.759	366.343	2 hrs 51 min 30.000 sec
SVM_S	v0		664	38.43%	8	682.620	32.569	1 hrs 40 min 52.664 sec
		poli	201	11.63%	6	188.189	43.042	2 hrs 44 min 30.000 sec
		gaus	451	26.10%	9	332.327	183.926	2 hrs 38 min 42.506 sec
SVM_S	v1		671	38.83%	6	683.290	31.969	5 hrs 48 min 24.383 sec
		poli	204	11.81%	1	189.044	43.859	2 hrs 43 min 50.000 sec
		gaus	467	27.03%	4	336.043	186.019	5 hrs 58 min 13.810 sec
SVM_S	IH		672	38.89%	5	685.173	31.969	16 hrs 19 min 31.759 sec
		poli	202	11.69%	0	189.898	43.859	6 hrs 17 min 49.000 sec
		gaus	470	27.20%	3	337.522	186.020	12 hrs 19 min 35.210 sec

APPENDIX G

DETAILED INFORMATION FOR RESULTS OF "CREDIT" DATA SET

Table G – Results of the experiments on "Credit"

model		kernel	# of misclassified points	% of misclassified points	# of unclassified points	sum of errors	wTw	Computation time
SVM	Classical		579	57.90%	-	1169.286	6.865	9 min 42.000 sec
		poly	0	0.00%	-	0.000	0.006	55 min 0.000 sec
		gaus	590	59.00%	-	1187.638	12.354	1 hrs 11 min 0.000 sec
SVM_S	v0		579	57.90%	-	584.643	3.432	16 min 20.125 sec
		poly	0	0.00%	-	0.000	0.003	26 hrs 58 min 26.000 sec
		gaus	591	59.10%	-	593.824	6.177	32 min 19.000 sec
SVM_S	IH		584	58.40%	-	589.037	3.358	3 hrs 12 min 03.129 sec
		poly	0	0.00%	-	0.000	0.008	16 hrs 47 min 39.000 sec
		gaus	590	59.00%	-	593.831	6.177	8 hrs 4 min 39.640 sec

APPENDIX H

DETAILED INFORMATION FOR RESULTS OF "TELECOM" DATA SET

Table H – Results of the experiments on "Telecom"

model		kernel	# of missclassified points in training data set	% misclassified point of training data	# of inconsistent points in training data set	% unclassified point of training data	sum of errors	wTw	# of missclassified points of testing data	% misclassified point of testing data	# of inconsistent points in testing data set	% unclassified point of testing data	computational time
classical		-	1791	58.38%	1264	41.20%	1262343.492	172.743	2287	74.57%	587	19.14%	28 hrs 20 min 9.955 sec
		poli	747	24.35%	1974	64.34%	1985.824	14081701.983	0	0.00%	3059	99.74%	55 hrs 48 min 57.342 sec
		gaus	2197	71.61%	795	25.91%	1139934.756	273.003	0	0.00%	3068	100.03%	49 hrs 35 min 9.723 sec
SVM-S	v0	-	1790	58.34%	100	3.26%	3704.424	490.111	0	0.00%	3068	100.00%	15 hrs 59 min 29.363 sec
		poli	1176	38.33%	1	0.03%	3439.896	64.869	826	26.93%	552	17.99%	16 hrs 23 min 28.605 sec
		gaus	1995	65.03%	191	6.23%	4135.408	374.638	1374	44.80%	0	0.00%	15 hrs 55 min 26.550 sec

Table H – Continued

SVM-S	v1	-	916	29.86%	103	3.36%	587.354	9.971E+21	1981	64.59%	238	7.76%	15 hrs 26 min 51.103 sec
		poli	73	2.38%	0	0.00%	14.985	92.920	2068	67.43%	0	0.00%	13 hrs 11 min 10.000 sec
		gaus	1475	48.08%	238	7.76%	1366.428	564.612	0	0.00%	1374	44.78%	26 hrs 59 min 25.000 sec
SVM-S	IH	-	1147	37.39%	104	3.39%	585.365	781.844	2974	96.97%	0	0.00%	12 hrs 54 min 33.000 sec
		poli		0.00%		0.00%							could not be solved
		gaus		0.00%		0.00%							could not be solved
UTADIS			1298				18.33						6.190 sec