AN EXACT APPROACH TO MAXIMIZE AREA UNDER RECEIVER OPERATOR CHARACTERISTIC CURVE FOR MULTI-INSTANCE LEARNING

A Thesis

by

Gizem Atasoy

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
August 2018

AN EXACT APPROACH TO MAXIMIZE AREA UNDER RECEIVER OPERATOR CHARACTERISTIC CURVE FOR MULTI-INSTANCE LEARNING

Approved by:

Associate Professor O. Erhun Kundakçıoğlu, Advisor
Department of Industrial Engineering
Özyeğin University

Assistant Professor Mustafa Gökçe Baydoğan
Department of Industrial Engineering
Boğaziçi University

Assistant Professor Ihsan Yanıkoğlu
Department of Industrial Engineering
Özyeğin University

Date Approved: 1 August 2018

To my beloved Mom, Dad, and Brothers...

ABSTRACT

The purpose of this study is to solve the multi-instance classification problem by directly maximizing the area under Receiver Operating Characteristic (ROC) curve (i.e., AUC). We derive a mixed integer linear programming model that produces the best possible hyperplane-based classifier for multi-instance classification. Our study sheds a light on the potential of hyperplane-based approaches, reflecting cross validation (CV) results for benchmark instances. As we maximize AUC directly, a hyperplane-based classifier can only *coincidentally* provide a better CV accuracy than those presented in this paper. Finally, we present how Kernel trick can be applied to produce nonlinear classifiers that maximize AUC.

ÖZETÇE

Bu çalışmanın amacı çoklu örnek sınıflandırma problemini Alıcı İşletim Karakteristiği (ROC) eğrisi altındaki alanı (yani, AUC) doğrudan maksimize ederek çözmektir. Çok örnekli sınıflandırma için mümkün olan en iyi hiperdüzlem tabanlı sınıflandırıcıyı üreten karma bir tamsayı doğrusal programlama modeli türetilmiştir. Çalşmamız, kıyaslama örnekleri için çapraz doğrulama (CV) sonuçlarını yansıtan hiperdüze tabanlı yaklaşımların potansiyeline ışık tutmaktadır. Doğrudan AUC'yi en üst düzeye çıkardıkça, hiperdüze tabanlı bir sınıflandırıcı sadece şans eseri bu yazıda sunulanlardan daha iyi bir CV doğruluğu sağlayabilir. Son olarak, AUC'yi maksimize eden doğrusal olmayan sınıflandırıcılar üretmek için çekirdek püf noktasının nasıl uygulanabileceğini sunuyoruz.

ACKNOWLEDGEMENTS

In each step of the thesis my supervisor Dr. Erhun Kundakçıoğlu supported me both motivationally and intellectually positively. I would like to thank him for his continuous guidance. In addition, I would like to thank to Özyeğin University Industrial Engineering Department and Academics for their support and trust.

I would like to express my heartfelt gratitude to my mother, my angel star, for her continuous motivation, courage and unconditional love.

I also want to take this opportunity to express my appreciation to my friends who believe in me in M.Sc. degree and the members of my research group, especially Cem Bozkır, Tonguç Yavuz, and Şeyma Gözüyılmaz.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Supervised learning studies pairs of (x_i,y_i) where x_i introduces set of features (attributes) for instances and y_i are corresponding responses or labels [2]. The intend is learning a mapping from input x to output y with approximations. In the literature of multi-instance classification, instances are grouped as *bags* and labels only refer to bags those behave on similar patterns [3]. This makes it challenging to reach the label of an individual instance a bag. Well-known multi-instance problems arise from drug activity prediction, image classification, economic predictions, audio processing, and information retrieval [4].

Classification methods and algorithms have the most attention in machine learning. However in today's world, needs are shifting beyond classification where ordering applications are becoming popular and solely classification does not meet the needs. Related to ordering , in the area of ranking , the goal is ordering a set of instances with their possibility to have desired attributes while showing a corresponding ranking to the end-user.

For example, in e-mail filtering, where unread messages appear above read messages; the ranking of unread messages would be in an order based on their probability being "urgent" [5]. Note that, learning whether a message is urgent or not is more valuable than knowing one's preference list of reading unread e-mails. Getting related information that an urgent message should appear on non-urgent ones is more natural and might be easier than obtaining general classification information. In addition, there are crucial topics such as ranking electrical grid components and drug screening, where a small improvement has a huge impact. Spotify challenge is another

similar setting that focuses on music recommendation to generate automatic play-list continuation after a play-list ends [6]. For that reason, a ranking information of the candidate songs for a specific list would be more valuable than only having the information of which songs can be added to this list. Due to aforementioned necessities, a classification approach which considers directly the performance of the learning would be a solution.

This paper focuses maximizing area under ROC curve (AUC) by developing a mixed integer model for Multi-Instance Learning (MIL). As our approach perform in the light of AUC, our model's structure resembles supervised bipartite ranking problem as well. Our approach provides a guarantee for optimality in small cases and a bound on the optimal objective value for large-scale problems. On the development of our model, first we provide an optimal hyperplane that maximize AUC. Next, a comparative study of hinge loss, 2-norm soft margin loss, hard margin loss, and ramp loss are presented to find offset parameters. Furthermore, our study sheds a light on the potential of hyperplane-based approaches, reflecting cross validation (CV) results for benchmark instances. Our findings indicate a hyperplane-based classifier providing better CV accuracy than those presented in this paper can be explained by luck.

The rest of this paper is organized as follows. Chapter 2, describes definitions, notations and background information related to the concepts. Chapter 3 presents our motivation behind this study and reviews the related work. Chapter 4 contains our mixed integer formulation for MIL. In Section 5, we demonstrate our computational results comparing them with a linear programming approach. In Chapter 6, we show our primary future work and conclude in Chapter 7.

CHAPTER II

BACKGROUND

2.0.1 Supervised bipartite ranking and Notation

Supervised bipartite ranking problems are generally favored in the community of machine learning [7]. The problem consists of a set of training examples/bags as inputs, which these bags contains a related binary labeled instance and the labels are known. For the latter reason, this type of ranking problems called *"Supervised"*. Typically in binary classification, classes are described as '+1' or '-1', which denotes positivity or negativity to show some characteristic of the instance. However, in bipartite ranking problem, the setting is different than classification problems, since there is a scorer. Scorer which is a function attributes real numbers to each instance, hence while positive instances have some greater scores, negative ones have scores always less than positive instances. In order to make this classification secured, violations are minimized through the bipartite ranking risk of scorer and some loss.

In the notation we refer to Bertsimas et al. [1] where a *supervised bipartite ranking* problem is studied for single instance classification. Our approach in this notation is transforming the single instance classification problem to multi-instance classification problem, where labels are defined over bags of instances. Hence, formulation of MI bipartite ranking problem adapts these constraints:

1. The rank of an instance should be greater than or equal to its minimum rank.

2. Each probable rank for every instance, can be only attained by a single example.

3. If there are repeating scores within a positive and a negative instance, in other words if they share the same score, negative instance always attain the higher

rank.

However, there is no singularity between two positively labeled or between two negatively labeled instances. This means, from these two examples, one of the positive examples might take a higher rank even if they share the same score. Otherwise, if there is no repeating scores, minimum rank is assigned to the rank of corresponding instance. Minimum rank is calculated as the number of instances that strictly score less than itself. If any negative instance has a higher rank than a positive instance or scores greater than equal to a positive instance, then we consider that as a *misrank*. This is why, in case of a tie between scores of a positive and a negative instance, positive instance stated as misranked. For this study, a linear scoring function is used in the formulations. $f(x_i) = w^T x_i$ where $w \in R^d$. Hence, scoring coefficients are optimal for the ranking quality measure.

Label y_i	+	+	+	-	-	+	-	+	-
Score $f(x_i)$	6	6	5	4	3	3	2	2	1
MinRank	7	7	6	5	3	3	1	1	0
Rank	8	7	6	5	4	3	2	1	0

Table 1: Example of rank definitions[1]

2.0.2 Mixed Integer Optimization (MIO)

Mixed integer problems involve both discrete and continuous decisions with restricting constraints less than equal to, grater than equal to or to be equal to some numerical limits [8]. Fundamentals of mixed integer linear programming are a non-empty integer variable set, linear constraints and an objective function. Besides, having integer variables participating to continuous variables separably and linearly is crucial [9]. The structure of mixed integer optimization problem commonly follows two type of model based on availability of quadratic features in the objective function. Mixed integer linear problems does not contain any quadratic terms neither in objective nor in the constraints, however; mixed integer quadratic programming problems contain quadratic terms only in objective function. In our solution approach to multi-instance

learning, we propose one mixed integer problem and one mixed integer quadratic problem for linear and non-linear datasets.

The following form of mixed integer optimization model refers to the maximization of a ranking problem introduced in this work,

$$\max \quad \sum_{j \in I} c_j x_j + \sum_{j \in C} c_j x_j \tag{1a}$$

$$\text{s.t.} \quad \sum_{j \in I} a_{ij} x_j + \sum_{j \in C} a_{ij} x_j \begin{cases} \geq \\ = \\ \leq \end{cases} b_i \qquad \forall i \tag{1b}$$

$$x_j \in \mathbb{Z}_+ \qquad \forall j \in I \tag{1c}$$

$$x_j \in \mathbb{R}_+ \qquad \forall j \in C \tag{1d}$$

Set C in formulation (1) contains the continuous variables, while set I includes only integral values. If the set of I is empty, problem becomes a linear optimization; if set C is empty then problem is called an integer optimization. In addition to integer optimization problem, if each variable is restricted to be 1 or 0, problem becomes a binary integer problem.

2.0.3 Maximize Area Under Receiver Operator Characteristic Curve (AUC)

After testing a binary classification problem, a confusion matrix containing true positive, true negative, false positive and false negative demonstrates the correct and incorrect classifications for a set of labeled instances. Based on these values one can calculate the accuracy, which can be shown as 1-Error as well, is calculated as the ratio of summation of truly predicted positives and negatives to the number of instances in the dataset. In general this probability of error is tried to minimize, but it

actually does not minimize the misclassification rate [10].

As in our case, in machine learning, if there is no information related to misclassification costs for your learning, or the data has imbalanced classes AUC has advantage over accuracy [11]. Accuracy implies, 1 minus probability of misclassification error which exceeds zero threshold. However, AUC calculates the probability of ranking error exceeds zero threshold. In addition, AUC offers successful results in differentiating distinct classes and this is more appropriate than using accuracy as a performance metric in MIL. As a result, AUC gets popularity in machine learning community.

2.0.4 Start Based Model

The initial mixed integer optimization model for single instance learning AUC maximization has the form:

$$\max \sum_{i \in I_+} \sum_{k \in I_-} z_{ik} \tag{2a}$$

$$\text{subject to } z_{ik} \leq v_i - v_k + 1 - \epsilon \qquad \forall i \in I_+ \, , \, \forall k \in I_- \tag{2b}$$

$$v_i = \mathbf{w}^T \mathbf{x_i} \qquad \forall i \in I_+ \tag{2c}$$

$$v_k = \mathbf{w}^T \mathbf{x_k} \qquad \forall k \in I_- \tag{2d}$$

$$-1 \leq w_j \leq 1 \qquad \forall j \in 1, ..., d \tag{2e}$$

$$z_{ik} \in \{0, 1\} \qquad \forall i \in I_+ \, , \, \forall k \in I_- \tag{2f}$$

Based on above model [1], v_i and v_k are the scorer for positive instance x_i and negative instance x_k. The score functions are designed in the form of constraint (2c) and (2d). Binary variable z_{ik} on the objective keeps a record if a positive instance gets strictly higher score than a negative instance as a pair. Namely, objective function

maximize the true classification of instances in a pair of (x_i,x_k) where $i \in I_+$ and $k \in I_-$. Consequently, z_{ik} takes value 1 in when true classification appears, but misclassifications are ignored and takes value 0. At the end, the aim is to catch all true classifications by using the constraint (2b). In this constraint ϵ is a user-specified small number greater than 0 because of the reason that v_i has to be forced to be strictly greater than v_k. Constraint (2e) restricts the solution space of w_j in the interval of [-1,1] to provide a shorter run-time.

2.0.5 Support Vector Machines

The idea of support vector machine (SVM) is to find the hyperplane that separates the classes in feature space. As the nature of classification methods try to separate two classes, since it is not always possible, there exist two common methodology to enable this classification. First one is that softening the definition of separation, and the latter is enlarging the future space, thus separation becomes possible in higher dimensions. For binary classification, one side of the hyperplane is labeled negative, where the other side is positive. It drives a conclusion of all points lying in one side have either positive or negative distance with the hyperplane where all points on the hyperplane are having distance of 0. On the other hand, selecting an optimal hyperplane is another subject, but having the largest gap (margin) from the points is what is aimed fundamentally.

However, not in all of the cases defining a hyperplane is that easy. In general when the number of sample points is less than the dimensions, separation with a hyperplane is easier. However, in most of the cases the number of points is large compared to dimension, which causes overlap of points. Furthermore, noises in data leads to dramatic shifts on hyperplanes. To deal with that problem, soft margin could be a solution where bias can be tolerated. Another way to eliminate overlaps on data points is what we mentioned above feature expansion. It is a standard

trick of adding features by including polynomial transformations. Therefore, the problem goes to a higher dimensional space. As the number of transform variables increases, the possibility to get a separation in this higher dimensional space increases as well. Nevertheless, turning back to 2 dimensional picture of the original variables concludes with non-linear hyperplanes. Even in regression, cubic polynomials are not preferred just because they grow fast in higher dimensions. A more controlled way of introducing non-linearity to support vector classifies is kernels. A kernel function is a function of two arguments, generally vectors. Kernels compute the inner products in a higher dimensional space and then most of the spaces squashed down so that we can fit support vector classifier in a high dimensional space.

When it comes to measuring the performance of SVM, Receiver Operator Curve (ROC) is tracing out true positive rates and false positive rates given the threshold, as there would be errors on classification after training and testing. This is a way of comparing classifiers with the Area Under Curve (AUC) a measure of curve showing that how much it closer to the one hundred percent of true positive rates corner. Finding best support vector classifier requires parameter tuning, as it is optimizing the decision boundary.

CHAPTER III

RELATED WORK

3.0.1 Motivation

Many algorithms appears in supervised learning systems containing logistic regression, decision trees, decision forest, neural networks, support vector machines and Bayesian classifiers aim to learn the mapping from *f(x)* to an output y [12]. These learning procedures mostly arise from numerical analysis or optimization theory depending on the type of machine learning problems. Indeed, diversity among these algorithms demonstrate the need of various applications with a different settings in this area. As each method address a particular level of sufficiency on computational complexity and performance offerings, there is no single beating method for any type of machine learning problem. This leads our research as our primary reason to observe the performance of an exact method. Additionally, MIL (Multi-Instance Learning) gets high attention in machine learning community in nowadays either because of the need to label large instances or because of the problem nature itself has a bag pattern. Thus, we drive our studies on the necessity of an exact approach in MIL.

This work circumvents the search for algorithms in multi-instance learning problem with the improving ability of solvers on MIO problems. Based on the paper of Bertsimas et al. called " Integer Optimization Methods for Supervised Ranking " we developed the offered mixed integer optimization model for multi-instance learning [1]. Their work offers a not commonly attempted mixed integer optimization method for machine learning problems to solve ranking tasks. As integer optimization have been explored firstly in ranking problems by them, it serves many rank statistics and objectives if it is looked for. On the upside, problem is solvable with exact solution

without any approximation or heuristic loss functions and yields an objective which is also a measure for ranking quality. Plus, model objective can be altered based on different rank statistics for different cases or specialized problems. This adds flexibility to study on various topics as well.

3.0.2 Literature Review

Multi-instance learning is a type of supervised machine learning problem [13]. Namely, it is a general form of supervised classification in that training class labels are correlated with sets of bags, or patterns, rather than singular patterns [14]. Therefore, individual labels of instances are not available in this setting. MIL handles the uncertainties in labels of bag. A classification is accepted as a true, if a bag only contains one positive instance labeled positive and a bag full of negative instances labeled as negative. The other cases than these conditions refer to misclassification [15].

[16] provides a broad range of survey about MIL problems branching the characteristics and types of it. Major characteristics of the MIL problem arises from bag composition, prediction level, label ambiguity and data distributions. Predictions might be done for instance level or bag level. However, while instance level predictions ensures the bag level predictions, the opposite is not true. Since instance labels are not available in real, classifying all bags correctly still result in misclassification of positive instances. Another issue with that is training. Even it is likely to train one hundred percent, it is not an assurance for testing with higher performance and hard to compute with the same reason of non-availability of instance labels.

Moreover, bag compositions can done by witness rate (WR) or relations between instances. WR is the rate of positive instances in a positive bag. When the WR rate is low it causes an imbalance problem and affects the performance. On the other side, relations between instances is another approach which focuses on similarities and dissimilarities of instances intra-bags. For picture subjects instance co-occurrences is

more popular because of the correlation of picture with another environment.

Beside machine learning algorithms, mathematical programming approaches are studied in the literature. They are generally derived as an extension of SVM model, and witness selection takes place in maximization of instance level margins. SVM models for MIL setting are formulated for two soft margin maximization (MIHLSVM), hard margin maximization (MIHMSVM) and ramp loss margin maximization (MIRLSVM) [15].

In this paper, we take into account the maximum margined witness (a single instance) from each bag. In order to deal with imbalance issues, we take witnesses from negative bags as well. We offer a wide range comparison of four loss models which are hinge, soft, hard and ramp margin maximization models.

CHAPTER IV

MATHEMATICAL MODEL

In this section, we introduce two comparative models; a MILP (Mixed Integer Linear Programming) model for linearly separable datasets finds optimal hyperplane. After, a hinge loss, a soft margin loss, a hard margin loss and a ramp loss linear programming models proposed to optimize offset parameter. 2-norm soft margin formulation objected to minimize error square (loss). On the other hand, hard margin formulation minimizes the misclassified selected instances among each bags. Ramp loss model both try to minimize misclassification and error in some extent [17]. These selection of instances from positive bags is determined with a decision variable, in other words a witness s selected from each positive bags, (θ_i) according to the maximum margin from the hyperplane. For negative bags, most prominent (the one has maximum margin from the hyperplane) negative instances selected after finding the margin coefficients. Later, these selected instances constitutes the optimal offset parameter(b).

The data originate in instances (vector patterns) $x_i \in R^d$, $i \in I_+ = \{1, .., N\}$ for positive instances and $x_k \in R^d$, $k \in I_- = \{1, .., N\}$ for negative instances. As bags also consists of 2 sets; positive S_+ , $p \in S_+ = \{1, .., M\}$ and negative S_-, $n \in S_- = \{1, .., M\}$ the responses (+1,-1) are representing those associated bags not instances. Vector coefficient variable $w_j \in R^d$, $j \in d = \{1, .., D\}$.

4.0.1 Parameters

ϵ A small user-specified constant

R A user-specified limit for ramp loss

4.0.2 Decision Variables

z_{ik}	Binary variable to keep track of whether x_i is scored higher than x_k
V_i	The score instance for x_i
V_k	The score instance for x_k
Ω_{pn}	A variable to keep track of whether pair of bag p and bag n correctly labeled
γ_{in}	Binary variable compares the correctness of selected positive x_i instance with the negatively labeled bag n
ζ_{in}	Binary threshold for linearization
θ_i	Weights on each instance i in a positive bag p
w_j	Hyperplane coefficients
b	Offset parameter
ξ_j	Slack variables for hinge loss
z_j	Binary variable takes 1 if there is a misclassification, 0 otherwise

4.0.3 MILP Model

The proposed model for multi-instance learning Roc maximization finds $\mathbf{w}$ the separating hyperplane. Then, with the given hyperplane coefficients $\mathbf{w}$ we find the offset parameter (threshold) b that maximize training accuracy.

$$\max \sum_{p \in S_+} \sum_{n \in S_-} \Omega_{pn} \tag{3a}$$

$$\text{subject to } z_{ik} \leq v_i - v_k + 1 - \epsilon \qquad \forall i \in I_+ \,,\ \forall k \in I_- \tag{3b}$$

$$v_i = \mathbf{w}^T \mathbf{x_i} \qquad \forall i \in I_+ \tag{3c}$$

$$v_k = \mathbf{w}^T \mathbf{x_k} \qquad \forall k \in I_- \tag{3d}$$

$$-1 \leq w_j \leq 1 \qquad \forall j \in 1..d \tag{3e}$$

$$\sum_{k \in I_n} z_{ik} \geq |I_n| \gamma_{in} \qquad \forall i \in I_+ \,,\ \forall n \in S_- \tag{3f}$$

$$\theta_i + \gamma_{in} \geq 2\zeta_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad (3g)$$

$$\sum_{i \in I_p} \zeta_{in} = \Omega_{pn} \qquad \forall p \in S_+ \ , \ \forall n \in S_- \qquad (3h)$$

$$\sum_{i \in I_p} \theta_i = 1 \qquad \forall p \in S_+ \qquad (3i)$$

$$\gamma_{in} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad (3j)$$

$$\zeta_{in} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad (3k)$$

$$z_{ik} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall k \in I_- \qquad (3l)$$

$$\theta_i \in \{0, 1\} \qquad \forall i \in I_+ \qquad (3m)$$

$$0 \leq \Omega_{pn} \leq 1 \qquad \forall p \in S_+ \ , \ \forall n \in S_- \qquad (3n)$$

Our objective is to maximize the true labeled bag numbers among instances. Constraint (3a) ensures the true rankings among instances. Constraint (3c) and (3d) presents the scores of positively and negatively labeled instances. Constraint (3e), ensures the bound of w_j in the feasible region to find optimal solution. Constraint (3f), keep track of correctly labeled bags. Ensures the comparison of positive instances with each negatively labeled bag's instances. Finally constraints (3g) and (3h) guarantees and selects the instance v_i as a bag's representative, then if bags labeled true Ω_{pn} takes 1 in case of there is at least one positive instance in each positive labeled bag(3i).

In this setting, optimal w_j values are obtained, then after picking the highest scored instances from each bag we continue to the following models to optimize treshold b value. Loss functions for training classifier that we formulated accordingly are shown in the next sections.

4.0.4 Hinge Loss

Hinge loss formulation is the traditional way of minimizing the continuous error for observations.

$$\min_{\mathbf{x_i},\mathbf{b}} \ \sum \xi_j \tag{4a}$$

$$\text{subject to } \ y_j(\mathbf{w}^T\mathbf{x}_j + \mathbf{b}) \geq -\xi_j \ , \qquad \forall j = \arg\max_{i \in I_p} \mathbf{w}^T\mathbf{x}_i, p \in S_- \cup S_+ \tag{4b}$$

$$\xi_j \geq 0 \ , \ \forall j \tag{4c}$$

$$\tag{4d}$$

4.0.5 2 Norm - Soft Margin Loss

Soft margin loss is enlarging the distance of observation from the margin by taking the square of loss. Therefore, it is the least robust method compared the other loss formulations presented in this paper.

$$\min_{\mathbf{x_i},\mathbf{b}} \ \sum \xi_j^2 \tag{5a}$$

$$\text{subject to } \ y_j(\mathbf{w}^T\mathbf{x}_j + \mathbf{b}) \geq -\xi_j \ , \qquad \forall j = \arg\max_{i \in I_p} \mathbf{w}^T\mathbf{x}_i, p \in S_- \cup S_+ \tag{5b}$$

4.0.6 Hard Margin Loss

The number of observations misclassified on the wrong side of the margin is minimized with this formulation. This method increases the robustness of classifier compared to 2-norm soft margin loss.

$$\min_{\mathbf{x_i},\mathbf{b}} \sum z_j \tag{6a}$$

$$\text{subject to } y_j(\mathbf{w}^T\mathbf{x}_j + \mathbf{b}) \geq -Mz_j\ , \qquad \forall j = \arg\max_{i \in I_p} \mathbf{w}^T\mathbf{x}_i, p \in S_- \cup S_+ \tag{6b}$$

$$z_j \in \{0,1\}\ ,\ \forall j \tag{6c}$$

4.0.7 Ramp Loss

This formulation is also known as *robust hinge loss*. The difference with the Hinge loss model is the misclassification errors. Ramp Loss puts greater error to misclassified observations. Restricts the impact of hinge loss in a specified ramp amount and punish them in the objective.

$$\min_{\mathbf{x_i},\mathbf{b}} \sum \xi_j + R\sum z_j \tag{7a}$$

$$\text{subject to } y_j(\mathbf{w}^T\mathbf{x}_j + \mathbf{b}) \geq -\xi_j - Mz_j\ , \quad \forall j = \arg\max_{i \in I_p} \mathbf{w}^T\mathbf{x}_i, p \in S_- \cup S_+ \tag{7b}$$

$$z_j \in \{0,1\}\ ,\ \forall j \tag{7c}$$

$$0 \leq \xi_j \leq R\ ,\ \forall j \tag{7d}$$

CHAPTER V

COMPUTATIONAL RESULTS

In this section we compare the performance of our MILP method with another study in the literature which proposes a linear programming approach to multi-instance learning[18]. For a performance metric, we compare the accuracy of their linear programming method, hence corresponding results of their study are listed as benchmark to ours.

Referring to their ten-fold cross validation indices, constituted by bag id numbers, we tested our linear programming method for five replication. Table 2 provides the related information for our experiment datasets and their corresponding number of instances, the min and max values of instances, number of bags and number of features. Additionally, the number of positive and negative bags are listed which is very significant for multi-instance learning. R instance and R cluster in Table 3 refers to their proposed linear programming methods. By taking their results as a benchmark to our study, along with the LP methods comparison, there are also two baseline methods results called miFV (MI based on Fisher vector representation) and MInD with $D_{meanmin}$ (uses bag-to-bag dissimilarity measures) representation presented in Table 3.

As our model objective is designed for ROC maximization, we reported our performance metric accuracy results for 5 times 10-fold cross validation. The LP solutions were generated using the Gurobi 8.0 solver. All computations are performed on a 3.6 GHz Intel Core i7-7700 computer with 16 GB DDR3 ECC (1866 MHz) RAM and the Ubuntu Linux operating system.

Based on our results it can be said that among four loss function hard margin loss

Name	Instances	Min	Max	Features	Bags	+ bags	- bags
Musk 1	476	2	40	166	92	47	45
Elephant	1391	2	13	230	200	100	100
Tiger	1220	2	13	230	200	100	100

Table 2: Description of common MIL Datasets

formulation, on average, yields the better accuracy performance with the CV among our datasets. Table 3 shows the results of their LP model. Table 4 and 5 presents the our results of MILP with hinge, soft, hard and ramp loss formulation. We tested ramp loss with 6 different parameters where R is set to 0.01, 0.5, 1, 2, 5, and 10. Hard margin loss for Elephant and Tiger datasets and soft margin loss for Musk1 dataset yield our best performance, which constitutes a lower bound for optimal hyperplane. Ramp loss with parameter 10 gives the second best bound for Musk1 among 6 different parameter. Soft margin loss is not performed well in Elephant and Tiger dataset, however performed best in Musk1 dataset.

Dataset	$R^{instance}$	$R^{cluster}$	$MInD_{(D_{meanmin})}$	$miFV$
Musk 1	**87.1**	86.2	84.1	85.2
Elephant	**87.4**	84.1	86.2	82.9
Tiger	**82.8**	80.3	77.7	80.4

Table 3: Accuracy results of other MIL methods with 10 fold cross-validation repeated 5 times.

Dataset	$MILP_{hinge}$	$MILP_{soft}$	$MILP_{hard}$	$MILP_{ramp(.01)}$
Musk 1	63.1	57.2	**70.5**	60.5
Elephant	66.1	58.6	**70.9**	67.3
Tiger	69.3	**71.1**	69.1	69.1

Table 4: Comparison of accuracy results of MIL methods with 10 fold cross-validation repeated 5 times.

Dataset	$MILP_{ramp(0.5)}$	$MILP_{ramp(1)}$	$MILP_{ramp(2)}$	$MILP_{ramp(5)}$	$MILP_{ramp(10)}$
Musk 1	**59.7**	59.6	59.5	59.4	59.1
Elephant	**66.1**	65.8	65.7	65.5	65.5
Tiger	69.1	69.1	69.3	69.3	**70.0**

Table 5: Comparison of accuracy results of MIL methods with 10 fold cross-validation repeated 5 times.

The following Figure 1 and 2 belongs to Tiger and Elephant datasets. We demonstrate the decreasing trend of average gap in time for 5 times 10-fold cross validations. For the Tiger and Elephant datasets the optimal solutions are obtained after 125 seconds and 1100 seconds on average respectively. Musk 1 dataset find optimal solution directly in one step on average 2 seconds, thus there is no progression in the gap for it.

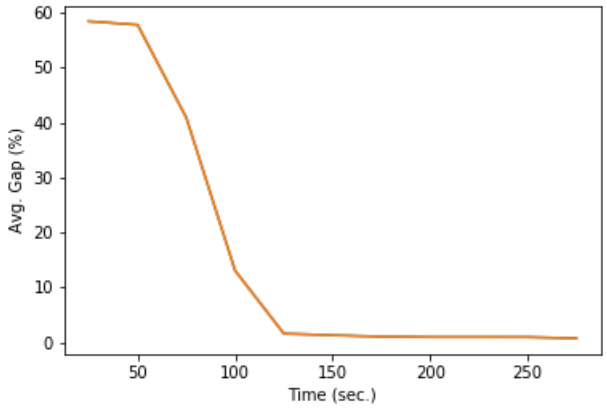

Figure 1: Average time to solve w that maximize AUC for Tiger dataset

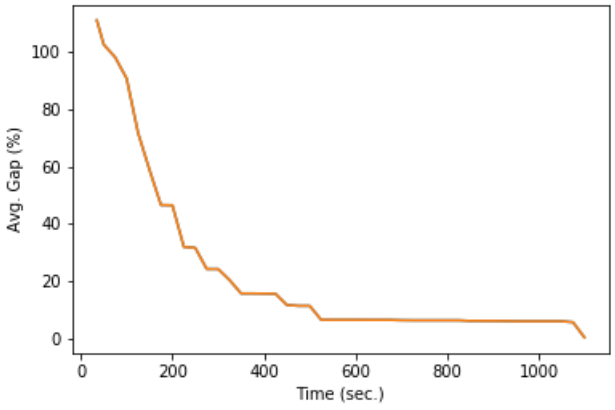

Figure 2: Average time to solve w that maximize AUC for Elephant dataset

CHAPTER VI

FUTURE WORK

MILP model is not yielding solutions for hard datasets which are not solvable with a linear hyperplanes. Therefore, we decided to improve MILP model for non-linear datasets. In this section we briefly describe the steps that we follow in our formulation. We used a SVM classifier to find the maximum margin hyperplane with a linear kernel function to map non-separable dataset in a separable format.

6.0.1 Nonlinear Transformation - MIQP Model

In this section, we introduce MINLP (Mixed Integer Non-Linear Programming) model for nonlinear datasets, which are not separable with linear hyperplanes. To address arrangements in the model, we developed objective function of the MILP formulation in a form that involves $\mathbf{w}$ with a small coefficient value ρ for regularization instead of bounding each component of $\mathbf{w}$.

$$\max \quad \sum_{p\in S_+}\sum_{n\in S_-} \Omega_{pn} - \frac{\rho}{2}\|\mathbf{w}\|^2 \tag{8a}$$

$$\text{subject to} \quad z_{ik} \leq \mathbf{w}^T\mathbf{x_i} - \mathbf{w}^T\mathbf{x_k} + 1 - \epsilon \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{8b}$$

$$\sum_{k\in I_n} z_{ik} \geq |I_n|\gamma_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{8c}$$

$$\theta_i + \gamma_{in} \geq 2\zeta_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{8d}$$

$$\sum_{i\in I_p} \zeta_{in} = \Omega_{pn} \qquad \forall p \in S_+ \ , \ \forall n \in S_- \tag{8e}$$

$$\sum_{i\in I_p} \theta_i = 1 \qquad \forall p \in S_+ \tag{8f}$$

$$\gamma_{in} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{8g}$$

$$\zeta_{in} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad \text{(8h)}$$

$$z_{ik} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall k \in I_- \qquad \text{(8i)}$$

$$\theta_i \in \{0,1\} \qquad \forall i \in I_+ \qquad \text{(8j)}$$

$$0 \leq \Omega_{pn} \leq 1 \qquad \forall p \in S_+ \ , \ \forall n \in S_- \qquad \text{(8k)}$$

This problem can be decomposed as the following:

$$\max_{\boldsymbol{\gamma},\boldsymbol{\zeta},\boldsymbol{\theta},\boldsymbol{z},\boldsymbol{\Omega}} \sum_{p \in S_+} \sum_{n \in S_-} \Omega_{pn} + \max_{\mathbf{w}:} \frac{-\rho}{2}\|\mathbf{w}\|^2 \qquad \text{(9a)}$$

$$z_{ik} \leq \mathbf{w}^T\mathbf{x_i} - \mathbf{w}^T\mathbf{x_k} + 1 - \epsilon \ \forall i \in I_+, \forall k \in I_-$$

$$\text{subject to} \quad \sum_{k \in I_n} z_{ik} \geq |I_n|\gamma_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad \text{(9b)}$$

$$\theta_i + \gamma_{in} \geq 2\zeta_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad \text{(9c)}$$

$$\sum_{i \in I_p} \zeta_{in} = \Omega_{pn} \qquad \forall p \in S_+ \ , \ \forall n \in S_- \qquad \text{(9d)}$$

$$\sum_{i \in I_p} \theta_i = 1 \qquad \forall p \in S_+ \qquad \text{(9e)}$$

$$\gamma_{in} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad \text{(9f)}$$

$$\zeta_{in} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \qquad \text{(9g)}$$

$$z_{ik} \in \{0,1\} \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{9h}$$

$$\theta_i \in \{0,1\} \qquad \forall i \in I_+ \tag{9i}$$

$$0 \leq \Omega_{pn} \leq 1 \qquad \forall p \in S_+ \ , \ \forall n \in S_- \tag{9j}$$

In this decomposition, binaries that denote misclassification are set at the outer level. Inner problem only checks for the existence of such a classifier. The inner problem can be restated as

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{10a}$$

$$\text{subject to} \quad z_{ik} \leq w^T(\mathbf{x}_i - \mathbf{x}_k) + 1 - \varepsilon \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{10b}$$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{w}} \quad \frac{1}{2}\|w\|^2 + \sum_i \sum_k \lambda_{ik}(z_{ik} - w^T(x_i - x_k) - 1 + \varepsilon) \tag{11}$$

Considering the transformation of the optimization problem into its corresponding dual problem, we found the primal Lagrangian as above, where all λ_{ik} are Lagrange multipliers and greater than zero.

After differentiating the primal Lagrangian with respect to $\mathbf{w}$ and imposing stationary,

$$\frac{\partial \mathcal{L}(w,\lambda)}{\partial w} = \mathbf{w} - \sum_i \sum_k \lambda_{ik}(x_i - x_k) = 0 \tag{12}$$

and re-substituting the relations obtained,

$$\mathbf{w} = \sum_i \sum_k \lambda_{ik}(\mathbf{x}_i - \mathbf{x}_k) \tag{13}$$

into the primal to obtain

$$\max_{\boldsymbol{\lambda}\geq 0} -\frac{1}{2}\sum_i\sum_k\sum_j\sum_n \lambda_{ik}\lambda_{jn}(\mathbf{x}_i-\mathbf{x}_k)\cdot(\mathbf{x}_j-\mathbf{x}_n)+ \sum_i\sum_k \lambda_{ik}(z_{ik}-1+\varepsilon) \tag{14}$$

Therefore, the formulation for nonlinear classification becomes with the Gaussian kernel function,

$$\max \sum_{p\in S_+}\sum_{n\in S_-}\Omega_{pn}-\frac{\rho}{2}\sum_i\sum_k\lambda_{ik}(z_{ik}-1+\varepsilon) - \frac{\rho}{2}\sum_i\sum_k\sum_j\sum_n\lambda_{ik}\lambda_{jn}K(\mathbf{x}_i-\mathbf{x}_k,\mathbf{x}_j-\mathbf{x}_n) \tag{15a}$$

$$\text{subject to} \sum_{k\in I_n} z_{ik}\geq |I_n|\gamma_{in} \qquad \forall i\in I_+\ ,\ \forall n\in S_- \tag{15b}$$

$$\theta_i+\gamma_{in}\geq 2\zeta_{in} \qquad \forall i\in I_+\ ,\ \forall n\in S_- \tag{15c}$$

$$\sum_{i\in I_p}\zeta_{in}=\Omega_{pn} \qquad \forall p\in S_+\ ,\ \forall n\in S_- \tag{15d}$$

$$\sum_{i\in I_p}\theta_i=1 \qquad \forall p\in S_+ \tag{15e}$$

$$\gamma_{in}\in\{0,1\} \qquad \forall i\in I_+\ ,\ \forall n\in S_- \tag{15f}$$

$$\zeta_{in}\in\{0,1\} \qquad \forall i\in I_+\ ,\ \forall n\in S_- \tag{15g}$$

$$z_{ik}\in\{0,1\} \qquad \forall i\in I_+\ ,\ \forall k\in I_- \tag{15h}$$

$$\theta_i \in \{0,1\} \qquad \forall i \in I_+ \tag{15i}$$

$$\lambda_{ik} \geq 0 \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{15j}$$

$$0 \leq \Omega_{pn} \leq 1 \qquad \forall p \in S_+ \ , \ \forall n \in S_- \tag{15k}$$

6.0.2 McCormick Relaxation

By applying McCormick Relaxation we are linearizing the product of λ_{ik} and z_{ik} in the above objective function by adding a new variable λ'_{ik}. Based on this relaxation new tie constrains added to the model.

$$\begin{aligned} \max \ & \sum_{p\in S_+}\sum_{n\in S_-}\Omega_{pn} - \frac{\rho}{2}\sum_i\sum_k(\lambda'_{ik} - \lambda_{ik} + \lambda_{ik}\varepsilon) \\ & - \frac{\rho}{2}\sum_i\sum_k\sum_j\sum_n \lambda_{ik}\lambda_{jn}K(\mathbf{x}_i - \mathbf{x}_k, \mathbf{x}_j - \mathbf{x}_n) \end{aligned} \tag{16a}$$

$$\text{subject to} \ \ \lambda'_{ik} \geq \lambda_{ik} - M(1 - z_{ik}) \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{16b}$$

$$\lambda'_{ik} \leq M z_{ik} \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{16c}$$

$$\sum_{k\in I_n} z_{ik} \geq |I_n|\gamma_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{16d}$$

$$\theta_i + \gamma_{in} \geq 2\zeta_{in} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{16e}$$

$$\sum_{i \in I_p} \zeta_{in} = \Omega_{pn} \qquad \forall p \in S_+ \ , \ \forall n \in S_- \tag{16f}$$

$$\sum_{i \in I_p} \theta_i = 1 \qquad \forall p \in S_+ \tag{16g}$$

$$\gamma_{in} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{16h}$$

$$\zeta_{in} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall n \in S_- \tag{16i}$$

$$z_{ik} \in \{0, 1\} \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{16j}$$

$$\theta_i \in \{0, 1\} \qquad \forall i \in I_+ \tag{16k}$$

$$0 \leq \Omega_{pn} \leq 1 \qquad \forall p \in S_+ \ , \ \forall n \in S_- \tag{16l}$$

$$\lambda_{ik} \geq 0 \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{16m}$$

$$\lambda'_{ik} \geq 0 \qquad \forall i \in I_+ \ , \ \forall k \in I_- \tag{16n}$$

CHAPTER VII

CONCLUSION

In this study we developed a multi-instance classification model by directly maximizing the area under ROC curve (AUC). Our mixed integer linear programming model produces the best possible hyperplane-based classifier. With this study, we bring light on the potential of hyperplane-based approaches providing cross validation (CV) results for benchmark instances. In addition, we demonstrate a Kernel trick application for nonlinear classifiers that maximize AUC.

It is clearly observed that having the optimal hyperplane for given datasets is not a guarantee for predicting the best accuracy as different bounds obtained in the different datasets. It is clear that in time our model generates optimal hyperplane with a decreasing optimality gap with 100% training. However, even though the accuracy results of MILP model are low compared to other LP study, generating different witness selection strategy might increase the accuracy results. In our case, we followed a strategy of selecting highest scored instance from each bag. In this study, witness selection is the most significant part in testing, which directly effects the cross validation results. Therefore, there are still potential to increase performance of this method. On the other hand, reporting AUC performance would be another strategy, but it will not be effective as changing the witness selection strategy from the bags.

To conclude, there exists potential research directions with the light of quadratic programming model as well. We believe that, by going further with the presented in our MIQP model, sophisticated optimization approaches would minimize the computation time and complexity.

Bibliography

[1] D. Bertsimas, A. Chang, and C. Rudin, "Integer optimization methods for supervised ranking," 2011.

[2] O. Chapelle, B. Scholkopf, and A. Z. Eds., "Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, pp. 542–542, March 2009.

[3] E. Alpaydn, V. Cheplygina, M. Loog, and D. M. Tax, "Single- vs. multiple-instance classification," *Pattern Recognition*, vol. 48, no. 9, pp. 2831 – 2838, 2015.

[4] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81 – 105, 2013.

[5] W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," *CoRR*, vol. abs/1105.5464, 2011.

[6] Spotify, "Spotify RecSys challenge 2018." `https://recsys-challenge.spotify.com/`, 2018.

[7] A. K. Menon and R. C. Williamson, "Bipartite ranking: A risk-theoretic perspective," *J. Mach. Learn. Res.*, vol. 17, pp. 6766–6867, Jan. 2016.

[8] J. C. Smith and Z. C. Taskn, "A tutorial guide to mixed-integer programming models and solution techniques," 2007.

[9] C. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Topics in Chemical Engineering, Oxford University Press, 1995.

[10] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 – 1159, 1997.

[11] M. Norton and S. Uryasev, "Maximization of AUC and buffered auc in binary classification," *Mathematical Programming*, Jul 2018.

[12] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[13] O. E. Kundakcioglu, O. Seref, and P. M. Pardalos, "Multiple instance learning via margin maximization," *Applied Numerical Mathematics*, vol. 60, no. 4, pp. 358 – 369, 2010. Special Issue: NUMAN 2008.

[14] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems 15* (S. Becker, S. Thrun, and K. Obermayer, eds.), pp. 577–584, MIT Press, 2003.

[15] M. H. Poursaeidi and O. E. Kundakcioglu, "Robust support vector machines for multiple instance learning," *Annals of Operations Research*, vol. 216, pp. 205–227, May 2014.

[16] M. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *CoRR*, vol. abs/1612.03365, 2016.

[17] J. P. Brooks, "Support vector machines with the ramp loss and the hard margin loss," *Operations Research*, 2011.

[18] E. S. Kucukasci, M. G. Baydogan, and Z. C. Taskin, "A linear programming approach to multiple instance learning," *Working Paper*, 2018.

VITA

Gizem Atasoy graduated from Beşiktaş Anatolian High School in 2010. She received her B.S. degree in Industrial Engineering from Özyeğin University in January 2016. After graduation, she joined Master of Science program in Industrial Engineering at Özyeğin University, and has been working under supervision of Assoc. Prof. Erhun Kundakçıoğlu. Her research focuses optimization in machine learning and data mining.