ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Dissimilarity based Multiple Instance Learning Using Dictionary Ensembles

M.Sc. THESIS

Nazanin Moarref

Department of Computer Engineering

Computer Engineering Programme

JANUARY 2017

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Dissimilarity based Multiple Instance Learning Using Dictionary Ensembles

M.Sc. THESIS

Nazanin Moarref
(504131535)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Asst. Prof. Dr. Yusuf Yaslan

JANUARY 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Sözlük Toplulukları Kullanılarak Farklılık Tabanlı Çoklu Örnek Öğrenme

YÜKSEK LİSANS TEZİ

Nazanin Moarref
(504131535)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Asst. Prof. Dr. Yusuf Yaslan

OCAK 2017

Nazanin Moarref, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504131535 successfully defended the thesis entitled “Dissimilarity based Multiple Instance Learning Using Dictionary Ensembles ”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Dr. Yusuf Yaslan** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Mustafa Ersel Kamaşak** Istanbul Technical University	.....
	Asst. Prof. Dr. Serap Kırbız Şimşek MEF University	.....
		.....

Date of Submission : **21 November 2016**
Date of Defense : **13 January 2017**

To my family,

FOREWORD

I would like to thank to my supervisor Asst. Prof. Dr. Yusuf Yaslan for his support, effort, patience and valuable advices either in study years and thesis research. Another important people considering my thesis are Göksu Tüysüzoğlu and Emrullah Gazioğlu who assist me during research process. I would also thank to my family for their care and support during whole my life.

January 2017

Nazanin Moarref
(Computer Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

MIL	**:** Multi Instance Learning
APR	**:** Axis-Parallel Rectangle
DL	**:** Dictionary Learning
SVM	**:** Support Vector Machine
DRS	**:** Dissimilarity Random Subspace
DBS	**:** Dissimilarity Bag Subspace
DRSSVM	**:** Dissimilarity Random Subspace using SVM
DBSSVM	**:** Dissimilarity Bagging using SVM
DRSDL	**:** Dissimilarity Random Subspace using DL
DBSDL	**:** Dissimilarity Bag Subspace using DL
EM	**:** Expectation-Maximization
NP-Hard	**:** Non-deterministic Polynomial-time Hard
ACC	**:** Accuracy
AUC	**:** Area under curve

SYMBOLS

$\mathbf{B}$, $\mathbf{B_i}$	: Bag and the *ith* bag in training set
$\mathbf{N}$	: Number of instances
$\mathbf{x_i}$	: The *ith* instance in the bag
$\mathbf{y_i}$	: The label of *ith* bag
$\mathbf{m}$	: Number of classes in the dataset
$\mathbf{M}$	: Number of training bags
$\mathbf{x_{ij}}$	: The *jth* instance in the *ith* bag
$\mathbf{T}$	: Tarining set
$\mathbf{W}$	: Dimension size of each training sample in Ensemble Learning methods
$\mathbf{d()}$	: Dissimilarity function
$\mathbf{d_i}$	: Basis vector of Dictionary
$\mathbf{R}$	: Real number
$\mathbf{r}$	: Prototype set
$\mathbf{r_i}$	: The *ith* prototype
$\mathbf{z}$	: Number of prototypes
$\mathbf{o_i}$	: The *ith* object (bag or instance)
$\mathbf{v_i}$	: The *ith* object dissimilarity vector
$\mathbf{S}$	: Number of instances/features/bag in a subspace
$\mathbf{X}$	: Input signal
$\mathbf{n}$	: Number of signals
$\mathbf{D, D_i}$	: Dictionary and the Dictionary related to the class *i*
$\mathbf{D_{ij}}$	: The dictionary classifier of *ith* subspace related to the jth category
$\mathbf{A}$	: Number of atoms
$\mathbf{K}$	: Number of subspaces
$\mathbf{t}$	: Number of sample in test set in Ensemble Learning
$\mathbf{P}$, $\mathbf{P_i}$	: Base ensemble classifier and the ensemble classifier of *ith* subspace
$\mathbf{P_{ij}}$	: The ensemble classifier of *ith* subspace related to the jth category
$\mathbf{L}$, $\mathbf{L_i}$	: Label and the label of *ith* sample
$\mathbf{T^{RS}}$ (i)	: The *ith* subspace in Random Subspace Ensemble Learning
$\mathbf{T^{BS}}$ $(\mathbf{i})$	: The *ith* subspace in Bagging Ensemble Learning
$\mathbf{T_{instance}}$	: The instances of entire training set

LIST OF TABLES

Page

LIST OF FIGURES

Page

Dissimilarity based Multiple Instance Learning Using Dictionary Ensembles

SUMMARY

Multiple Instance Learning (MIL) is one of important topics in the pattern recognition research field. It differs from many traditional machine learning problems in terms of real-world object representations. In MIL problems, samples are represented by multi-sets which are commonly named as bags where each bag include a set of feature vectors called instances. Many real world problems such as image, text or document classification, drug activity prediction and etc., can be formalized as MIL problems. Mostly, MIL problem, deals with supervised learning paradigm, which aims to learn models from the on hand samples and use it to solve regression or classification problems. Most of the cases, MIL problem is referred to binary classification where each bag has to be classified into one of the two categories; "positive" or "negative". Many MIL algorithms are developed considering the assumption that a bag is positive if at least it has one positive instance. Although this assumption has good results in some domains such as drug activity predictions, it may be restrictive for other domains of MIL problems such as computer vision MIL problems can be solved using instance based, bag based or embedded space algorithms. In this thesis, embedded-based strategy has been applied which converts the MIL problem to a standard supervised learning. In this approach, instead of using the mentioned assumption which relates the instance labels to the bags' labels, the dissimilarity of bags to the selected training instances which are called prototypes are taken into consideration. This mapping makes the bags be represented by standard fixed sized feature vectors. Most of the time, this feature mapping may include lots of redundant and irrelevant features. In this work we use Dictionary Learning as a classifier, which generates the sparse representation of each signal and classify them simultaneously. Using dictionaries would result in more efficient, fewer noises, simple and sparse manner, which make signals have more global look. As a third Approach, Ensemble Learning technique is implemented. This algorithm, apply two mentioned methods in its strategy and combine their strength where leads to higher and more reliable classification performances. Random Subspace and Bagging are two strategy implemented as Ensemble Learning approach. The proposed algorithm is evaluated on 11 different MIL datasets and compared with a recently proposed dissimilarity based ensemble MIL algorithm that uses Support Vector Machines (SVM). Experimental results show that the proposed algorithm outperforms the counterpart algorithm that uses SVM.

Sözlük Toplulukları Kullanılarak Farklılık Tabanlı Çoklu Örnek Öğrenme

ÖZET

Çoklu Örnek Öğrenme, örüntü tanıma problemleri içerisinde karşılaşılan önemli problemlerden birisidir. Geleneksel makine öğrenmesi algoritmalarından ayrılan en önemli yanlarından birisi, geleneksel yöntemlerin öznitelik vektörü gösteriminden farklı bir gösterime sahip olmasıdır. Geleneksel makine öğrenmesi yöntemlerinde sınıflandırılacak veya demetlenecek nesnelere karşı bir öznitelik vektörü hesaplanmakta ve bu vektörler kullanılarak modeller oluşturulmaktadır. Bu modeller kullanılarak yeni veriler sınıflandırılmakta veya demetleme gerçekleştirilmektedir. Buna karşılık Çoklu Örnek Öğrenme problemlerinde her bir veri birden fazla örnek içeren bir torba ile gösterilmektedir. Problemi zorlaştıran durumlardan birisi de her bir torba içindeki örneklerin sayısının farklı olabilmesidir. Bu tür makine öğrenmesi problemleri teknolojinin gelişimi ile birlikte metin sınıflandırma, molekül aktivitelerinin belirlenmesi, görüntü kategorizasyonu, ses ve müzik türü sınıflandırması gibi birçok alanda karşımıza çıkmaktadır. Fakat bu alanların bazılarında verilerin gösterimi Çoklu Örnek Öğrenme problemine karşılık gelebilmektedir. Örneğin, görüntü sınıflandırma probleminde bir görüntü birden fazla örnek içerebilmektedir. Çoklu Örnek Öğrenme, ilk olarak molekül aktivitelerinin yapısını ögrenmek için önerilmiş ve metin sınıflandırma, proteinlerin bağlanma bölgelerinin tahmini gibi diğer farklı makine ögrenmesi problemlerinde de kullanılmaya başlanmıştır. Çoklu Örnek sınıflandırma problemlerinde çoğunlukla torbaların etiket bilgisi elimizde olabilmekte, buna karşı örnekelerin etiket bilgisi bulunmayabilmektedir. Torbaların sınıflandırılması örnekler ile ilişkilendirilmiştir. Bir torbanın pozitif olarak sınıflandırılabilmesi için içinde en az bir pozitif örnek olması yeterlidir. Diğer yandan torbanın negatif olarak sınıflandırılabilmesi için torba içindeki tüm örneklerin negatif olması gerekmektedir. Şimdiye kadar Çoklu Örnek Öğrenme için önerilen algoritmalar genel olarak torba ve örnek uzayında ayrı ayrı çalışacak şekilde tasarlanmışlardır. Birinci tipteki sınıflandırıcılar örnek uzayında çalışan sınıflandırıcılar olup torbaların sadece örneklerini göz önünde bulundurmaktadırlar. Bu sayede sınıflandırıcı, pozitif etiketli örnekleri ve negatif etiketli örnekleri kullanarak model oluşturmaktadır. Yeni gelecek verilerin örnekleri dikkate alınıp, yeni verilerin torbalarının etiket bilgisi belirlenmektedir. Bu yöntemin en önemli dezavantajı torbalar içerisindeki tüm örneklerin etiket bilgisini gerektirmesidir. Öte yandan bu yöntemde torbayı bir bütün olarak kullanıp öğrenme yapılmadıgı için torbanın genel yapısı öğrenilemez. Bu nedenle, torba uzayında sınıflandırıcı öğrenen yöntemler önerilmiştir. Bu yöntem, örnekleri teker teker değerlendirilmek yerine, torbayi örneklerle birlikte bir bütün olarak değerlendirmektedir. Bu yöntemde, torbalar örnekler kullanılarak tek bir öznitelik vektörü ile gösterilmekte veya yine örnekler kullanılarak torbalar arası benzerlik değerleri hesaplanabilmektedir. Bu yöntemde torbalar ikili olarak birbirleriyle karşılaştırılmakta ve sonuç olarak bir benzerlik değeri hesaplanabilmektedir.

Bu çalışmada örnek ve torba uzayını birlikte göz önünde bulunduran, sözlük öğrenme yöntemlerini de kullanan farklılık tabanlı bir algoritma önerilmiştir. Bu yöntem, modeli oluşturulacak verilerin torbalarını ve torbaların içinde olan tüm örnekleri dikkate almaktadır. Yöntemde öncelikli olarak tüm eğitim kümesindeki torbalardan rastgele örnekler seçilmektedir. Bu seçilen örnekler prototip örnek olarak kabul edilmektedir. Bu prototip örnekler ile torbaların içindeki örnekler karşılaştırılmakta ve farklılık değerleri hesaplanmaktadır. Her bir prototip için en düşük farklılık o torbanın ilgili prototipe karşı düşen vektör değeri olarak yazılmaktadır. Bu sayede torbalar için sabit uzunluklu birer öznitelik vektörü çıkarılabilmektedir. Böylece, çalışmanın ilk aşamasında Çoklu Örnek Öğrenme problemi, eğiticili makine öğrenmesi problemine dönüştürülmüştür.

Çoklu Örnek Öğrenme probleminde, veriyi uygun öznitelik vektörü gösterimine çevirmeye ek olarak başarılı sonuçlar elde edebilmek için iyi sınıflandırıcılara da ihtiyaç duyulmaktadır. Seyrek kodlama ve sözlük öğrenimi çoğunlukla sinyal işleme ve görüntü işleme alanlarında kullanılmış ve ayrıca sınıflandırıcı olarak da kullanılan başarılı bir yöntemdir. Bu yöntemde, veriler bir sözlüğün temel elemanlarının ayrık doğrusal birleşimi olarak temsil edilmektedir. Bu gösterim, verinin örüntüsünü ortaya çıkarıp veriyi daha yalın bir şekilde kullanmaktadır. Seyrek kodlama ve sözlük öğrenimi iki aşamada gerçekleştirilmektedir. Birinci adımda, seyrek kodlama gerçekleştirilmekte ve ikinci aşamada ise sözlük elemanları güncellenmektedir. Bu adımlar belirli bir hata değerine kadar iteratif olarak devam etmektedir. Bu aşamalarda seyrek kodlama ve sözlük matrisi yenilenmektedir ve her tekrarlamada, yeninden yapılandırma hatası azaltılmaktadır. Böylece en çok ayrık ve yalın gösteremine neden olan sözlük matrisi elde edilmektedir.

Bu tez çalışmalasının ikinci aşaması olarak seyrek kodlama kullanılarak sözlük öğrenimi sınıflandırıcı yöntemi olarak kullanılmıştır. Seyrek kodlama ve sözlük öğrenimi aynı anda verinin üzerinde en bilgilendirici öznitelikleri seçip çıkartmaktadır. Her bir sınıf için ayrı birer sözlük matrisi oluşturulmaktadır. Bu sayede pozitif ve negatif torbalar için iki ayrı sözlük matrisi elde edilmiştir. Sınıflandırma aşamasında, etiketlenecek verinin her iki matris kullanılarak ayrık kodlaması gerçekleştirilmekte ve geri çatım hatası en küçük matris sınıf etiketini vermektedir.

Bir çok makine öğrenmesi uygulamalarınında sınıflandırıcı toplulukları tek sınıflandırıcılara göre daha yüksek başarımlı sonuçlar üretebilmektedir. Sınıflandırıcı toplulukları birden fazla model oluşturup bu modellerin verdiği kararları birleştirmeye yarayan makine öğrenmesi yöntemleridir. Bu yöntemlere topluluk öğrenme algoritmaları da denilmektedir. Topluluk öğrenme algoritmaları hem örnek hem de öznitelik alt uzayları üzerinde uygulanabilir. Random subspace ve Bagging algoritmaları literatürde sıkça kullanılan topluluk öğrenme yöntemlerindendir. Random subspace algoritması öznitelik uzayında ve bagging algoritmalası da örnek uzayında veri alt kümeleri seçmektedir. Bu tez çalışmasının üçüncü aşaması olarak topluluk öğrenme özelliklerinden yaralanarak Random subspace ve bagging algoritmaları kullanılmıştır.

Random subspace yöntemi, öznitelik uzayında çalışmakta olup sınıflandırıcı topluluğundaki her bir sınıflandırıcının kullanacağı öznitelik vektörleri rastgele seçilmektedir. Dolayısıyla topluluktaki her sınıflandırıcı için farklı öznitelik alt uzayları kullanılmaktadır. Karar aşamasında çoğunlukla çoğunluk kararı veya sonsal olasılık değeri kullanılabilmektedir.

Bagging yöntemindeyse tüm öznitelik değerleri kullanılmaktadır. Buna karşın, topluluk içindeki her bir sınıflandırıcının kullanacağı veriler training kümesinden rastgele seçilen verilerden elde edilmektedir. Sınıflandırma aşamasında ise Random subspace yönteminde olduğu gibi tüm sınıflandırıcıların verdiği kararlar çoğunluk değerine veya sonsal olasılık değerine göre birleştirilmektedir. Bu tez çalışması kapsamında tüm topluluk yöntemlerinde sınıflandırıcı kararlarının birleştirilmesi için sonsal olasılık değerleri kullanılmıştır. Tez çalışması kapsamında önerilen yöntemler 11 farklı Çoklu Örnek Öğrenme verileri üzerinde test edilmiştir. Verilerin üçü görüntü kategorizasyonu, ikisi ilaç molekül aktivitelerinin sınıflandırılması ve kalan veriler ise metin sınıflandırma problemlerine aittir. Elde edilen sonuçlarsa, literatürde yakın zamanda önerilmiş Çoklu Örnek Öğrenme Yöntemiyle (DRS) karşılaştırılmıştır. Bu yöntemin en önemli farkı sınıflandırıcı olarak literatürde yüksek başarım verdiği gösterilmiş olan Destek Vektör Makinası algoritmasını kullanmasıdır. Elde edilen test sonuçlarına göre tez çalışması kapsamında önerilen yöntem sınıflandırma başarımını arttırmakta ve DRS yöntemine göre çok daha iyi sonuçlar vermektedir.

1. INTRODUCTION

As time goes on, technology grows faster, the collected datasets become more complex and the need for more impressive approaches to analyze these datasets increases. Many research areas have been arisen to deal with datasets. Machine Learning is a domain that discovers algorithms which can learn the complicated pattern of the data and make decisions according to the obtained information. Supervised learning is one of the classical problems of machine learning. In this area the data samples have class labels which indicate each sample belongs to a category or a class. These labeled samples can be used as a training set to learn a model. The label of unseen samples named test set, could be predicted using the learned model.

Classification problem is applied in many fields such as spam detections, pattern recognition, speech recognition, bioinformatics, handwriting recognition [1]. Acquiring a good accuracy in label predictions is one of the challenging field for researchers. Many algorithms have been developed to construct good classifiers like support vector machines, linear classifiers such as fisher‘s linear discriminant, quadric classifiers, neural networks, kernel estimation, k-nearest neighbors and many other algorithms will be developed to solve day-to-day problems of classification.

Recently, Multi Instance Learning (MIL) has been one of the interesting fields in supervised learning problems. There are a lot of works on MIL that have been applied in real-world problems such as drug activity prediction [2], content-based image retrieval, classification [3] and text or document classification [4], mostly. In an MIL problem, the classifiers are dealing with set of instances which are gathered in different kinds of bags. MIL is a type of supervised learning but it differs from general type of mentioned classification problems. In general form, each example to be learned has fixed length of feature vector where in MIL problem each example is a bag consisting of multi feature vectors [5]. Most of the time, MIL problem deals with binary classification problems where each bag is labeled as a positive bag if at least one of the instances in related bag has positive label. This means that the bag consists

of the type of instances that we are looking for. If the bag doesn't include any positive instance, then it will be labeled as a negative bag. Classifiers have to train the model using bags (bag-based, embedded-based) or instances(instance-based) and their labels to predict the labels using obtained model for unseen bags [6, 7].

In order to achieve good classification performance, the classifier has to be well-informed by the data. Most of the time the data have a lot of noises which mislead classifiers and lead to decreased performance of classifiers. Therefore, feature representation plays a crucial role in classification performance.

In literature many methods have been proposed such as; wavelet transformation [8], Fourier transformation [9], kernels [10] and etc., to transform feature vectors into different dimensional space in order to obtain more informative feature vectors with lower noise.

In pursuit of these variant methods the sparsity of the wavelet coefficients is observed to be proper approach in obtaining reasonable models. It has been applied largely in the past decade and in search of improved versions of these techniques, sparse and redundant representation modeling has been achieved [11]. The obtained experiment results in many research areas, regarding this issue, show that sparse and redundant representations conduce to state-of-the-art results. Accordingly, sparse coding and dictionary learning has recently attracted researcher‘s interests by representing each instance as sparse as possible and presenting them as linear combinations of basic elements which are called atoms and create dictionaries. Many research fields have been benefited from sparse and dictionary learning techniques such as signal denoising, image compression, feature extraction [12], texture synthesis [13], supervised learning [14] and unsupervised clustering [15].

In order to improve the classification performance, ensemble learning also could be used. In this method, more than one diverse classifiers are created to learn and make decisions. The final decisions are made by taking the decisions of each classifier into consideration. The main purpose of applying this approach is to obtain better predictive performance in comparison to performance which could be obtained from any of the constituent classifiers alone. Classifier level, feature space, instance space are different ways of generating ensemble classifiers. Adaboost, bucket of models, bootstrap

aggregating (bagging), random subspace feature selection, random forests are the frequent applied ensemble learning methods in most research area [16]. Random subspace feature selection is a feature-based ensemble learning technique. Subspace are created by using randomly selected features from feature space. Bagging is an instance-based ensemble learning where each subspace is formed by randomly selected instances. Each ensemble classifiers uses relative subspace to learn a model and make a decision. The final decision is a product of combination of all classifier‘s outcomes. In this study MIL is considered as a classification problem and dictionary learning classifier is used as a base classifier. To enhance the classifier performance ensemble methods have been applied.

1.1 Contribution of the Thesis

In this thesis ensemble based dictionary learning algorithms that use dissimilarity based features are implemented to solve MIL problem.

In order to represent the data, dissimilarity of bags with all the instances are calculated and used as feature vector for each bag and the dissimilarities between instances are obtained using eucleadian distance as used in [17]. Therefore, the MIL problem is converted into supervised learning problem. Random subspace feature selection and bagging are two selected ensemble learning techniques. Sparse coding and dictionary learning is used as a base classifier in ensemble learning. Experimental results are obtained on 11 MIL datasets and the performance of the proposed ensemble dictionary learning is compared with the results that are obtained on ensemble SVMs on the same feature sets. The remarkable results of these two ensemble methods can show how these two approaches could boost the prediction ability of dictionary learning model. Ensemble SVMs indicate the results of a recently proposed solution to MIL problem namely “dissimilarity based ensemble for multi instance learning” [17] where the method outperforms most of the successful MIL solutions. The comparison with this prominent method shows the importance of selecting appropriate base classifier and the strength of sparse coding as feature representation and dictionary learning as classifier.

1.2 Literature Review

To exemplify MI representation in real world problems, drug activity prediction can be referred [18]. In this problem, each drug is a molecule as a bag consist of its conformations as instances. The classifiers should decide whether the molecule is active or not or if it binds to another molecule. In addition to recognizing which molecules are active, the researchers intend to determine which conformers of the related molecules are responsible for the activity. The difficulty is that each molecule can adopt multiple conformations, and only few of them are responsible for desired observations. As a result, the complete molecule is depicted as a bag of N possible conformations set, where each conformation instance is represented with a feature vector and the number of conformations (N) can differ for different molecules.

Image classification is another example for real-world problems. Each image consists of some regions and the given image should be classified according to its visual content. In order to determine whether the image include the desired part or not, it is necessary to extract the regions of the image. At the end of this process, each region is depicted with a vector of features. As a result, the image is defined as a bag included N feature vectors which are named instances describing the different regions of the image. Depending on the methods that are going to be used for extracting different regions of the image, N might vary for different images.

Document categorization is another field which can be formulated as a multi instance problem. A document (such as articles and conference paper, product or movie reviews, news items) consists of several paragraphs which are described by sentences. Each document can be considered as a bag and each paragraph as an instance.

Information retrieval, audio processing, economic predictions and etc. are other domains of real-world problems that require MIL formulations for categorization [19]. According to the information existent in the on hand data, classifiers could use the characteristics of individual instances, without considering the global characteristics of the whole bag or the learner could apply the global characteristic of the bags considering that the discriminative information lie at the bag-levels. In all of the

mentioned problems, it is necessary to use appropriate strategy to obtain good classification accuracies.

In literature according to the used classification strategies, MIL problem research could be grouped into three categories: bag-based and instance-based space or embeded-based. In bag-based and embeded-based the bags are directly classified while in instance-based, instance classifiers are built and combined to classify bags.

In bag-based methods, the classifier is built directly using the bags under the assumption that the bags of the same classes are similar to each other. In this way the bags can be categorized implementing distances, kernels or single instance techniques which are used as a representation for relative bags. Wangs et al, presented an approach based on Hausdorff distance of bags which are then used to label the bags according to KNN (K Nearest Neighbor) classification method [6].

Embeded-based method is similar to bag-based method in which each bag is represented by a fixed sized vector. Chen et al, uses similarity values as a simple vector to represent each bag. [7] Following this work Cheplygina et al apply dissimilarity space to represent each bag using prototypes can be instances or bags [17].

Instance-based method classifier uses instances to label bags. The assumption is that the label of the bag is positive if and only if it has at least one positive instance, otherwise the bag will be labeled as negative bag. Multiple-instance learning problem was proposed by Dietterich et al. originally [20]. In this method, the instance-based classifier used axis-parallel hyper-rectangle (APR) method to find a region which has at least one instance from positive bags but there is not any instance from negative bag. Hence, the algorithm begins the optimization by selecting an initial positive instance. Then the greedy process extend the APR using greedy steps, where in each step, it finds the positive instance of a not yet covered positive bag where by adding it to the APR, make the least increasing in APR size. By this way, the APR is then expanded adding that positive instance. These steps are proceeded until APR include at least one positive instance of each bag. Similarly Andrews et al. also used the instance labels to construct classifiers [21]. In this work instance labels are initialized hypothetically, and the classifier is built using these labels, then by considering the bag labels and the constraint they put on the instance labels, the labels are updated. Thereby, the classifier

is also updated iteratively. SVM is used as a supervised classifier. Similarly, in this way, Zhang et al. apply boosting method [22] and Leistner et al. uses Random forest technique as a supervised classifier [23]. Besides Ramon and De Raedt used neural networks [24] and Blockeel et al. implemented decision trees [25] and Raykar et al. applied bayesian aprroachs to obtain the instance labels to predict bag labels [26].

Analogously, a lot of interest has been grown in dictionary learning and sparse coding field recently. Dealing with high dimensional signals, brings us necessity to reduce their dimension while their main properties remain intacked. Hence we can efficiently, use them in process or store them. Image compression is a good example for this concept. Bryt and Elad used dictionary learning and sparse coding as a dimensionality reduction method in image compression field [27].

Tosic and Frossard used dictionary learning and sparse coding for Audio and visual data not only for representing the data in reduced dimensions which are adaptive to the basic structure underlying the signals, but also used this technique in face recognition application to show the discriminative power of sparse representation in supervised problems [28]. In addition to classification, Sprechmann et al. implemented dictionary learning and sparse coding technique as a clustering approach [29]. In this approach, a set of dictionaries one for each cluster are constructed. Each signal is associated to a cluster in which the signal has minimum reconstruction error using relative dictionary. By this way, the signals that use the same dictionaries for their sparse representations, belong to the same clusters. According to the results obtained using the extended standard datasets and texture images, dictionary learning not only has remarkable performance in classification and discriminative aspects but also it is suitable approach to manage large datasets.

Recently, Ensemble methods have been applied in many classification problems. Making decisions according to more than one classifiers, result in more reliable decisions and increase the classification accuracy. Hence, using ensembles instead of using one classifier, reduce the risk of making mistakes in prediction and many researches have been presented in this field [16]. Additionally, classifier ensembles give ability to manage large amount of data. In problems with complex decision boundaries, this method, helps to apply divide and conquer strategy easier. A detailed description about ensemble based techniques such as bagging, Adaboost,

stack generalization, and general combination rules such as voting based techniques, decision templates, algebraic combination of outputs and etc. is given in [16].

Random Subspace Method (RSM) is an ensemble learning method which tries to produce divers classifiers and reduce correlations between classifiers trained with randomly selected features instead of whole features [16]. Ho used the RSM to prevent overfitting while decision trees as classifiers attempt to learn the model and preserve the maximum accuracy simultaneously [30]. RSM has been applied in different areas such as Chawla et al. applied in 2-D face recognition task [31], Kuncheva et al. applied it for classification of brain images [32] and Xia et al. used RSM for hyperspectral image classifications [33]. RSM has also different variations. Lai et al. proposed a new ensemble method by integrating the informativeness of features as a qualification for selection in each subspace with RSM. In this approach, random subspace method, initially, select features from the entire features randomly and then, Liknon or Recursive Feature Elimination as multivariate search methods, is used to extract the informative features to obtain the reduced feature space. This strategy proceeds iteratively to cover large amount of entire features [34].

Bagging is another ensemble learning technique. In this method, instead of selecting features, in order to construct subspaces, instances are selected randomly. In literature there is a vast amount of researches that apply bagging algorithm. For example, Zhang used bagging for inferential estimation of polymer quality [35], Dettling and Marcel for tumor classification using gene expression data [36], West et al. for financial decision applications [37] and Hsieh et al. for credit scoring [38].

Recently Ensemble dictionary learning model is implemented to extract the outstanding region namely saliency region of an image [39]. In this approach, each subspace, is consisting of image samples which are selected randomly and dictionaries are trained on these generated subspaces. Therefore, for each image patch, multiple sparse representations are attained. In addition, reconstruction residual based model for reducing atoms of learned dictionaries applied to increase the distinctness of salient patches from background. As a result, multiple probabilistic saliency acquired for each patch. The final decision is made by combining these multiple probabilistic results and calculating posterior probabilities to predict the performance of each patch as a saliency region. In this thesis, a dictionary based ensemble method for MIL problem is

proposed by using dissimilarity values. The proposed algorithm not only can be used for image classification but also can be applied to any MIL problem.

1.3 Thesis Structure

The remaining chapters of the thesis are organized as follows. In Chapter 2, we cover the methods and techniques that are used in this thesis. In first phase, Dissimilarity based Multi Instance learning is explained, then the combination of this method with Ensemble based Random Subspace technique is described. Dictionary learning and sparse coding is clarified as a next step and in last phase Random Subspace and Bagging Ensemble methods are discussed in details. In Chapter 3, the proposed method "Dissimilarity Based Multi Instance using dictionary Ensembles " is explained in detail. In Chapter 4, The experiment results are illustrated, And in Chapter 5, we will conclude the thesis by emphasizing on the important results achieved.

2. METHODOLOGY

2.1 Dissimilarity Based Multi Instance Learning

In multi instance learning, a bag is a set $B = \{x_i | i = 1,2,\ldots,N\}$, where the $x_i \in R^d$ are instances called feature vectors. N is the number of instances in a bag and it can vary from bag to bag. All the instances belong to the d dimensional feature space called instance space. The aim of this learning method is to learn a model which can predict targets of unseen bags where the training set $T = \{B_i, y_i | i = 1,2,\ldots,N\}$ including positive bags $y_i = +1$ and negative bags $y_i = -1$.

In large amount of work on MI learning, make the standard assumption which is about the relationship between the instances within a bag and the class label of the bag and state that the bag is positive if it contains at least one positive instance. In this work, instead of representing a bag by its instances, it is represented by relative dissimilarities to reference objects called prototypes. Let $d()$ be any dissimilarity function where $d(o_i, o_j) \in R$, $x_i \in R^d$ and $r = \{r_1, r_2 \ldots, .., r_z\}$ be the set of prototype objects. Each object o_i is represented as a simple vector of dissimilarities $v_i = [d(o_i, r_1), ..., d(o_i, r_z)]$ which has z dimension and the j-th feature is related to the dissimilarity of the object to j-th prototype. Hence, the bags classification can be considered as a simple supervised learning. In this work, Euclidean distance is used as a dissimilarity function $d()$ (equation 2.1 and equation 2.2).

$$d(x_i, x_j) = \left\| x_i - x_j \right\|_2 \tag{2.1}$$

$$\left\| x_i - x_j \right\|_2 = \sqrt{(x_i - x_j) \cdot (x_i - x_j)^T} \tag{2.2}$$

Dissimilarity based multi instance learning method is initilized by Cheplygina et al. [17]. In this approach two type of prototypes are selected: bags or instaces. Firstly, bags can be selected as reference prototypes. In this way, each bag is represented using the dissimilarity values to prototypes. Suppose that we have z number of bags as a prototypes. Therefore, we get: $v_{bag}(bag, r) = [d(bag, r_1) \quad d(bag, r_2) \quad \ldots \quad d(bag, r_z)]$

where dissimilarity of a bag using prototype set is:

$$v_{bag}(bag_i, bag_j) = 1/N_j \sum_{c=1}^{N_j} min_l(x_{il}, x_{jc}) \tag{2.3}$$

The equation 2.3 average out the minimum dissimilarity value of the instances of bag_i to each instance of bag_j. The result is a scalar and if we determine z prototype bags $d_{bag}(bag, r)$ gives us z dimensional vector. Under the condition when the instances of the bag are informative we expect a good performance but if a few instances of the bags are informative, d_{bag} average out the dissimilarities where it leads to lower performance. Secondly, Instances of bags can be chosen as prototypes. In this case $r = bag_j = \{x_{j1}, x_{j2}, ...x_{jN}\}$.

$$\begin{aligned} v_{instance}(bag_i, r) = [&d_{instance}(bag, r_1), d_{instance}(bag, r_2), ..., \\ &d_{instance}(bag, r_z)] \end{aligned} \tag{2.4}$$

As a result, distance will have $\sum_{j=1}^{z} N_j$ dimension where N_j is the number of instances in j-th prototype bag. In contradiction to d_{bag} which average out the dissimilarities, $d_{instance}$ preserve all the dissimilarities but it could suffer from high dimensionality and a lot of redundant and uninformative dissimilarities which makes the classifiers have troubles to select relevant dissimilarities [17]. Following this approach, as a next step these two methods combined with ensemble learning technique and proposed two separated methods. DBS (Dissimilarity Bag Subspace) and DRS (Dissimilarity Random subspace).

In DBS, each subspace is created by selecting one of the training bags as a prototype. Then, the dissimilarity of remaining training bags to the instances of the prototype bag are used to represent each training bag in prototype space. Therefore, the dimensionality of subspace varies according to the number of instance in selected prototype bag.

In DRS, prototype bags are created from all available instances in the training set. As a result, we would have M prototype bags where each of them consist of S number of instances. Similar to the DRS method, in DBS method, the subspaces are generated using training bags, where at each subspace one of the training bags is selected as prototype and the dissimilarity vector of training bags is calculated using the instance

of the prototype bag. One can choose any value for M and S, but according to the work in [17], if there is no idea about the number of redundant features in data, it is better to choose M, relatively small and S relatively large.

The classifier can be built using these dissimilarity vectors as a feature vector for training bags and predict labels for unseen samples. Note that the test set would be represented in dissimilarity space using the same prototype set applied in training set. Cheplygina et al. used linear SVM as a classifier in the ensembles [17]. According to the results, DRS has better performance in comparison to DBS. One reason might be because of the subspace dimensionality. In DBS, the subspaces have different and small dimensionality size whereas in DRS the subspaces have large size and supply more informative instances as prototypes which yield better performance for classifiers. Another reason could be that the classifiers generated by DRS are more diverse than the classifiers created by DBS.

DRS as a better approach is compared to other prominent algorithms in multi instance problem field such as MILES [7], minimax-SVM [40], MILBoost [22], MI-SVM [21], EM-DD [41]. It is indicated that in most cases DRS brings higher accuracy compared to the other mentioned methods. In pursuit of this approach, we tried to implement DRS with another classifier and compare the results with another Ensemble method called Bagging. We illustrate these algorithms in more details in the next chapter.

2.2 Dictionary Learning and Sparse Coding

In this section we will give details about general dictionary learning and sparse coding methods. Consider $X \in R^n$ as an input instance. The dictionary is a matrix, consisting of normalized, basis vectors d_i where $d_i.d_i^T = 1$. We symbolize dictionary with $D = [d_1, d_2, \ldots, d_k]$ where $D \in R^{n \times k}$. Most of the times, the number of instances are greater than the number of atoms$(n > k)$ in this case the dictionary is called an over complete dictionary. $\alpha \in R^k$ is called sparse code, which is the coefficient vector. Using the dictionary, the input signal can be represented as linear combination of atoms. It can be formulated as:

$$min\|\alpha\|_0 \quad s.t \quad X = D\alpha \tag{2.5}$$

$\|\alpha\|_0$ is the L_0 norm of the coefficient vector α which indicate the number of non-zero elements of α. In order to represent the signal as sparse as possible, we try to find the minimum number of non-zero elements. In the case that the dictionary is an over complete dictionary, finding the sparsest representation would be difficult. Because the computational problem would arise which needs combinatorial search which encounters us with NP-Hard problems. To find best solution, instead of using L_0, L_1 norm can be used which make the non-convex problem to the convex problem, and ensures the existence of a unique global minimum result for the mentioned formulation. As a result, sparse representation will be formulated as:

$$min\|\alpha\|_1 \quad s.t \quad X = D\alpha \tag{2.6}$$

considering the noises: X=$D\alpha + \varepsilon$ Therefore, in general form the sparse representation is formulated as:

$$\alpha^* = argmin_\alpha \|\alpha\|_1 \quad s.t \quad \|D\alpha - X\|_2 \leq \varepsilon \tag{2.7}$$

Dictionary learning, is a learning method by constructing dictionary from the input signals directly. To address this purpose, dictionaries could be attained by solving this minimization problem:

$$min_{D,\{\alpha_{i=1,\ldots,N}\}} \sum_{i=1}^{N} \|X_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \tag{2.8}$$

Where X_i is representative of input signals, N is the number of signals, $\|X_i - D\alpha_i\|_2^2$ is data fitting term which defines reconstruction error, $\|\alpha_i\|_1$ is the regularization term which specifies the decomposition sparsity and λ is penalty parameter that hold the trade-off between the data fitting and regularization term in balance. Considering D and X as variables of the mentioned problem, make this optimization problem as non-convex problem. To address this issue, the solution is considering one of the variable fixed, so that the minimization function turns into convex optimization problem concerning to the other variable. As a result, to solve this objective function, two optimization steps related to these two variables are needed. In order to achieve a predetermined convergence, it is required to apply these two steps iteratively until achieving the desired convergence.

- Sparse Approximation step: in this way, Dictionary D is considered as a fixed variable, then coefficients α of signal X using the dictionary D are calculated by minimizing the objective function. (Note that at the first step the dictionary is initialized randomly)

- Dictionary Update step: New dictionaries are computed and become updated using the calculated sparse coding matrix X. Updating dictionaries leads to reducing the approximation error.

2.2.1 Discriminative dictionary learning and sparse coding

Dictionary learning not only is applied for reconstructive purpose but also can be used for discriminative (classification) purposes. Discriminative dictionaries are appropriate strategy to classify input data. In pursuit of this aim, class labels of the input data would get involved in learning dictionaries which result in different signal representation for separate classes. Hence, both sparse representation and reconstruction are taken into consideration for categorization purpose. Thereby, for each separate class of the input data, a unique dictionary is trained using the instances which are involved in their related class. If we consider an input data with m class labels the base dictionary D is the result of m sub-dictionaries $D = [D_1, D_2, ..., D_m]$ where $D_i \in R^{n \times k}$. To classify the test set we use D dictionary to encode the signal. This means that the signal used all sub-dictionaries to be encoded. Which sub-dictionary leads signal to have least reconstruction error and the sparsest representation, the signal would be assigned to the class label of considering dictionary. To explain in more details, the steps are:

- obtain the sparse code $(\alpha_1, \alpha_2, ..., \alpha_m)$ of the signal X for each sub-dictionary D_i which are trained for each class of input data.

- compare the cost of representation $\delta(i)$ for each sub-dictionary and its related sparse code.

- Assign the class labels of a dictionary which leads to minimum representation cost:

$$class \quad i^* = argmin_{i \in 1,\ldots,m} \delta_i(X) \tag{2.9}$$

$$where \quad \delta_i(X) = min \|X_i - D_i \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \tag{2.10}$$

2.3 Ensemble Methods

Ensemble is a strategy using multiple classifiers to train the model and make a decision by combining the results of all classifiers considering some decision criteria. There is always possibility that the on hand data, can be full of noise which result in misleading the classifier. Thereby, single classifiers cannot make a certain accurate decisions or give reliable accuracies. Among the ensemble methods Random subspace and bagging has received a lot of attention. These two methods are explained in more details in next two parts.

2.3.1 Random subspace ensemble learning

Random subspace ensemble learning is one of the popular ensemble learning method which attempts to generate subspaces from randomly selected features [42]. This technique can be used in any algorithm and most of the times, increase the performance of classification methods. Let W be the number of features and S be the number of features to be selected $(S < W)$. We create K subspace of these features where S features are selected randomly at each subspace. Each of the subspace classifiers is trained using the relative selected features, and predict the labels for unseen test data using the learned model. Since we have K subspaces, therefore we have K separate classifiers and their predictions on the unseen test data sets. The advantage of this method is that in high dimensional data, it may reduce the problems that may arise from curse of dimensionality problem. Pseudocode for Random Subspace Ensemble Learning is shown in Table 2.1

2.3.2 Bagging ensemble learning

This ensemble learning technique is the most straightforward approach for manipulating the instances of training set. Contradict with random subspace method which select the features randomly, in bagging instances are selected randomly (with replacement) from the entire training set samples. Such a training set has S samples of entire samples. Each subspace, consists of a bootstrap replicate of the entire training set. In this way, this method is called bootstrap aggregation which the

Table 2.1 : Pseudocode for Random Subspace Ensemble Learning.

Algorithm: Random subspace Ensemble learning
Input: : training set T, number of features in the subspace S, number of ensemble classifiers K, base classifier P.
Output: a set of ensemble classifiers $P = [P_1, P_2, ..., P_K]$.
For i = 1: K
Create a subspace using sample data; $T^{RS}(i)$,
with S randomly selected instances from T.
Train the model of subspace $T^{RS}(i)$ with classifierP_i.
End for
Return obtained ensemble model $P = [P_1, P_2, ..., P_K]$

bagging term is originated from. Each bootstrap replicate or in other words, each subspace has the same size (number of samples) as entire samples size of training set, and approximately, contain 63.2% [43] of instances in the main training set with several samples which appear multiple times. Bagging instance selection strategy help creating diverse classifiers which increase the performance of learners for instance label predictions. Pseudocode for Bagging Ensemble Learning is shown in Table 2.2

Table 2.2 : Pseudocode for Bagging Ensemble Learning.

Algorithm: Bagging Ensemble Learning Learning
Input: training set T, the number of samples in training set N, number of ensemble classifiers K, base classifier P.
Output: a set of ensemble classifiers $P = [P_1, P_2, ..., P_K]$.
For i = 1: K
Create a bootstrap sample data ; $T^{BS}(i)$, with N randomly selected instances
with replacement from T. (Omit the repetitive selected samples)
Train the model of $T^{BS}(i)$ with classifier P_i.
End for
Return obtained ensemble model $P = [P_1, P_2, ..., P_K]$.

3. PROPOSED DICTIONARY LEARNING BASED ENSEMBLE MULTIPLE INSTANCE

In this study, dissimilarity based MIL has been addressed. In this case we have MIL problem datasets which consist of bags and their associated feature vectors (instances). To convert the MIL problem into standard supervised learning problem, dissimilarity of bags to the prototypes is calculated according to the equation 2.4. Thereby, each bag is represented by the dissimilarity values as features. Increasing the number of features in dissimilarity vector can result in more complex calculations for learning the models. Sparse coding can help to decrease the dimension, therefore the classifiers deal with lower complex problems, so their performance can increase [28]. In this work, supervised Dictionary Learning and Sparse Coding are proposed as a classifier, which generate the sparse representation of each signal and classify them simultaneously. As a third approach, ensemble learning methods Random Subspace and Bagging are taken into consideration. Random Subspace is applied in dissimilarity feature space while bagging in sample (bags) set. In Random subspace, some instances from all training instances are randomly selected as prototypes. You can consider the selected prototypes as a bag in each subspace. The instance-based dissimilarity is used to calculate the dissimilarity of each bag to the prototype instances (considered as a prototype bag). In bagging method, all training instances are selected as prototypes. Thereby, our feature vector consists of instance-based dissimilarity values of each bag to all existent instances. Then, each subspace is generating by randomly selected bags. The pseudocode of proposed methods with Random Subspace ensemble(DRSDL) and bagging ensemble learning(DBSDL) technique are given in Table 3.1 and Table 3.2 respectively. The framework figures for DRSDL and DBSDL are indicated in Figure 3.1 and 3.2 respectively.

Table 3.1 : Pseudocode for Dissimilarity based MIL using Random Subspace Dictionary Learning.

Algorithm: Dissimilarity based Multiple Instance Learning using Random subspace Dictionary learning

Input: training set T, the number of samples in training set N, number of ensemble classifiers K, subspace size S, base classifier $D(P \leftarrow D)$, number of atoms A, number of samples in test set t, prototypes $r = \{r_1, r_2, ..., r_z\}$

Output: labels of test set $L = [L_1, L_2, ..., L_t]$

Training:

For i = 1: K

Create a subspace using sample data; $T^{RS}(i)$, with S randomly selected instances as prototypes from T; $r = \{x_1, x_2, .., x_z\}$.

Calculate dissimilarity of each bag in $T^{RS}(i)$ to the each instance selected as prototype in r; $d_{instance}(bag, R)$.

Use dissimilarities to represent related bags(equation2.4).

With A atoms, Train two different dictionaries for positive and negative bags $(D_{i1}, D_{i,2})$ in T, using dissimilarity vectors.$(P_{i1} \leftarrow D_{i1}, P_{i2} \leftarrow D_{i,2})$

End for

Test:

Calculate dissimilarity of test bags to the prototype instance as well.

For i=1: K

Predict the test bag labels using P_{i1}, P_{i2}.

End for

Output:

Classify using posterior probabilities for each test sample; L

Table 3.2 : Pseudocode for Dissimilarity based MIL using Bagging Dictionary Learning.

Algorithm: Dissimilarity based Multiple Instance Learning using Bagging Dictionary Learning

Input:training set T, the number of samples in training set N, number of ensemble classifiers K, base classifier $D(P \leftarrow D)$, the number of atoms A, number of samples in test set t, prototypes $r = \{r_1, r_2, ..., r_z\}$.

Output: labels of test set $L = [L_1, L_2, ..., L_t]$

Training:

For i = 1: K

Calculate dissimilarity of each bag in T, to all instances in training set; $d_{instance}(bag, T)$. and represent each bag with related dissimilarity vector(equation2.4).

Create a subspace using training bags; $T^{BS}(i)$, with randomly selected Bags from T with replacement.

With A atoms train two different dictionaries for positive and negative bags $(D_{i1}, D_{i,2})$ in $T^{BS}(i)$, using dissimilarity vectors.$(P_{i1} \leftarrow D_{i1}, P_{i2} \leftarrow D_{i,2})$

End for

Test:

Calculate dissimilarity of test bags to the all training instance as well.

For i=1: K

Predict the test bag labels using P_{i1}, P_{i2}.

End for

Output:

Classify using posterior probabilities for each test sample; L

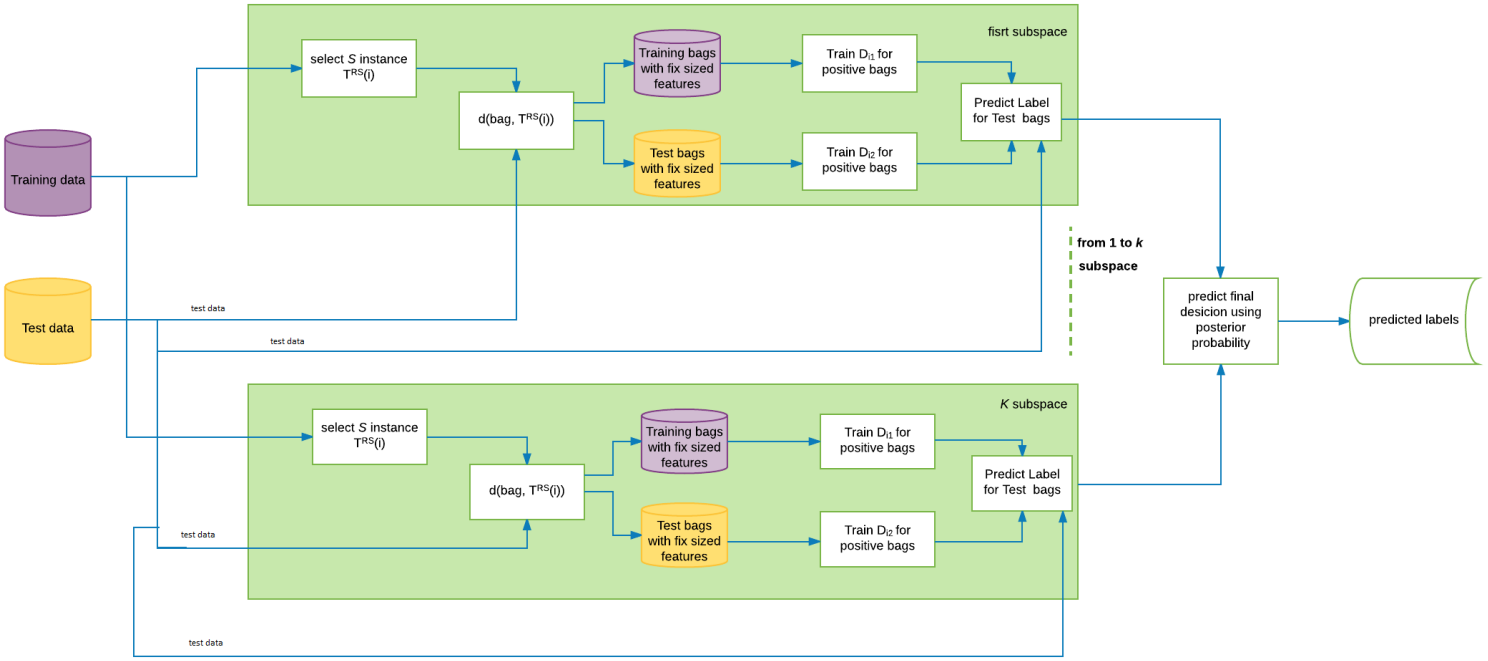


Figure 3.1 : DRSDL framework.

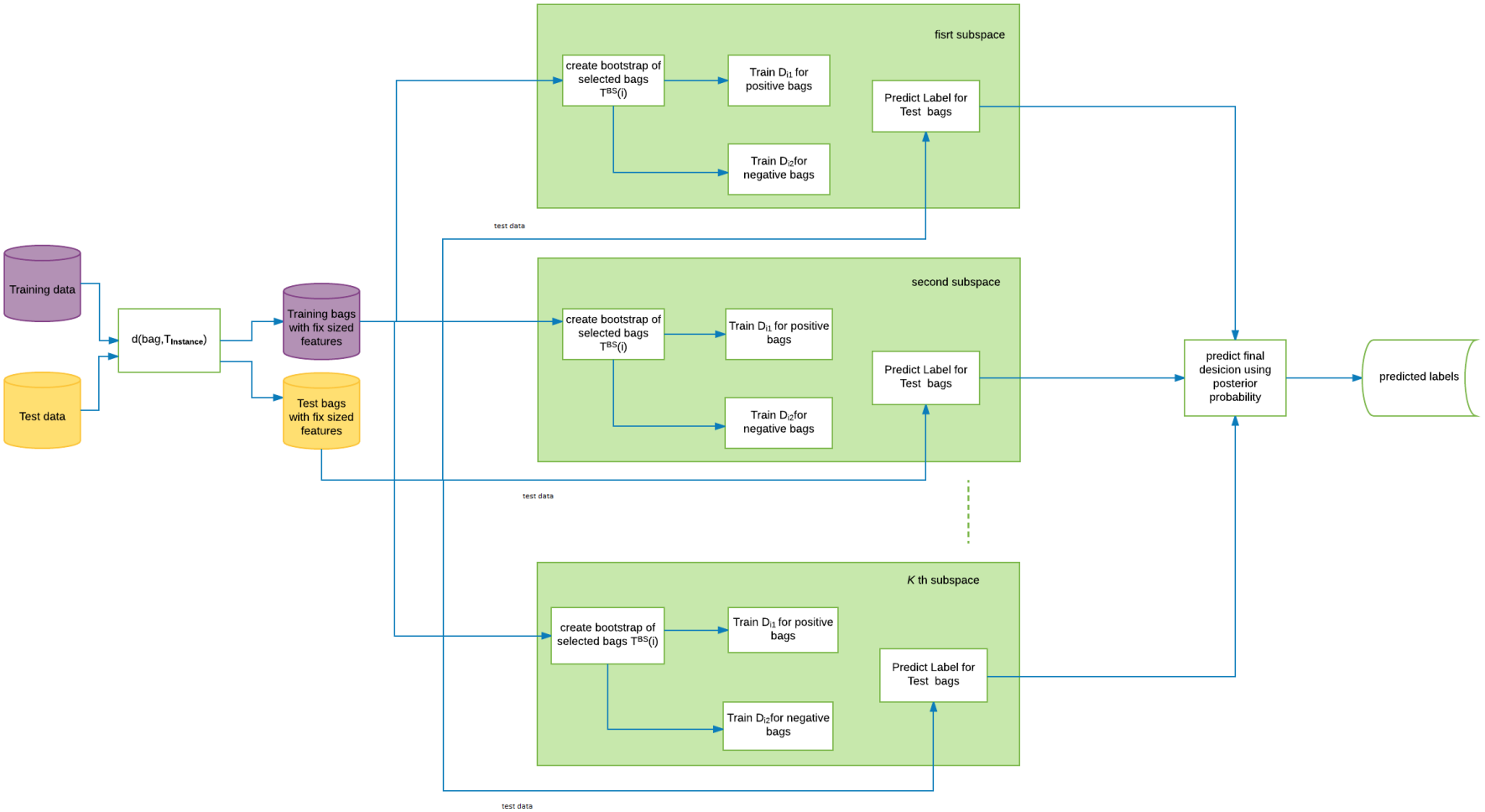


Figure 3.2 : DBSDL framework.

4. EXPERIMENTAL RESULTS

The proposed methods (DRSDL and DBSDL) performances are examined on 11 different MIL datasets [44]. Table 4.1 indicates the number of bags including number of positive and negative bags and the total number of instances that each dataset contains. Tiger, Fox, and Elephant dataset are most frequently used benchmarks in

Table 4.1 : MIL Datasets used for experimental results.

dataset	total bags	positive	negative	total instances
Tiger	200	100	100	1220
Fox	200	100	100	1320
Elephant	200	100	100	1391
Musk2	102	63	39	6598
Musk1	92	47	45	476
alt.atheism	100	51	49	5443
comp.graphics	100	52	48	3094
rec.autos	100	51	49	3458
sci.crypt	100	51	49	4284
sci-med	100	51	49	3045
talk.politics.guns	100	51	49	3558

MIL problems related to image categorization fields. The bags are images and the instances are segments of images. Positive bags are considered the images which contain the relevant animal and the negatives are considered the ones which don't contain the relative animal. Musk1 and Musk2 are about molecule activity prediction problems. In this problem, the classifiers try to make decisions whether a molecule has a musky smell or not. As mentioned in previous chapters a molecule can have different shapes which are fold into conformers. Hence, each bag is considered as a molecule and each instance is one of its conformers. In this case if at least one of the conformers can make the molecule smell musky, then the bag would have positive label. The remaining datasets are from 20Newsgroups dataset. Newsgroups is about a text categorization problem and generated from 20 different categories. A bag consists of different posts (as instances) from different categories. The positive bags contain 3% posts from relevant category and 97% from other categories.

Tenfold cross validation is used to obtain results for all the methods. All the dissimilarity values for training set and test sets are normalized with zero mean and unit variance. As the number of features in each dataset is data dependent, in Random Subspace Ensemble methods (DRS and DRSDL) the number of selected features are different. Since there is no information about the number of redundant features, the number of features selected for each subspace is examined for different sizes. In the experiments 5%, 10%, 20%, 30%, 40% and 50% of training instances are evaluated and the one which gives the best accuracy among these, are selected for being the random subspace size. For example, the Elephant dataset has the best accuracy with 5%, whereas Fox has the best accuracy with 50%. By this way, we tried to obtain better results using fewer than 50% of all instances in training set.

In Bagging Subspace Ensemble methods (DBSDL and DBSSVM) all the training bags are selected with replacement, Hence, at the end approximately, 63.2% of different training bags are chosen to generate subspaces. In both of the Ensemble methods, the number of subspaces is chosen 20. Because there is not any information about irrelevant bags and instances, we tried to choose the number of classifiers averagely small, in comparison to the subspaces size. The reason why we use the same number of subspaces for Bagging and Random subspace methods is that we try to fix all possible parameters so we have approximately accurate perception about the performance of each Ensemble method. For example, we tried to compare whether the instance selection gives us better accuracies or bag selection. On the other hand, the performance of two different classifiers (SVM and DL) are tried to be analyzed. As a result, in this study, we did not focus on analyzing the number of subspace or subspace size. Table 4.2 indicates the number of atoms used for each relevant dictionary based method and the percentage of selected instances (S) from entire training instances (Random subspace dimension size). Table 4.3 shows the Accuracy and Standard Error (SE) results for all data sets. The AUC and SE performance of these methods are shown in Table 4.4.

As it is clear in these two tables, The DRSDL out-perform not only the DRSSVM method, which shows the high performance of dictionary learning comparison to SVM but also it out-performs the Bagging Ensemble method for each DBSDL and DBSSVM.

Table 4.2 : Number of atoms and percentage of selected instances.

dataset	DDL	DBSDL	DRSDL	percentage
Tiger	100	50	200	20%
Fox	25	50	200	50%
Elephant	150	25	200	5%
Musk2	25	25	25	10%
Musk1	50	25	100	30%
alt.atheism	25	25	50	20%
comp.graphics	200	50	50	50%
rec.autos	50	50	150	50%
sci.crypt	25	25	100	30%
sci-med	25	25	200	50%
talk.politics.guns	100	25	50	50%

Table 4.3 : Accuracy (%) and SE results.

DATA	DSVM		DDL		DBSSVM		DBSDL		DRSSVM		DRSDL	
	ACC	SE	ACC	SE	ACC	SE	ACC	SE	ACC	SE	ACC	SE
Tiger	83	2.71	80	3.16	82.5	2.5	83.89	1.93	81	2.56	**87**	2
Fox	63	2.81	56	2.21	**74**	2.96	73.5	3.17	64.5	2.17	65.5	1.57
Elephant	71.5	2.59	80	4.08	75	2.98	83	3.67	83.5	3.08	**87.5**	3.27
Musk2	85.09	3.1	76.45	4.16	**89**	3.48	80.27	2.63	**89**	3.48	82.27	3.28
Musk1	87.8	2.61	77.44	4.15	90.02	2.51	**90.28**	3.11	88.94	3.32	81.66	3.48
alt.atheism	69.8	3.07	73.07	3.34	68.92	2.86	81.07	2.29	69.05	1.69	**90.18**	2.02
comp.graphics	54.02	1.95	51.01	0.67	55.13	2.32	76.32	3.54	54.02	1.95	**81.94**	2.51
rec.autos	59.23	2.87	65.25	3.9	60.12	3.11	73.12	4.02	61.23	2.43	**79.36**	4.43
sci.crypt	68.94	3.81	73.16	4.17	69.93	2.16	78.07	2.88	70.14	2.79	**90.07**	2.02
sci-med	64.92	3.1	70.74	3.72	68.03	2.39	79.85	2.75	71.94	3.62	**83.98**	3.71
talk.politics.guns	66.14	2.89	57.01	1.99	72.94	3.04	83.05	2.5	76.05	2.98	**84.16**	2.49

Table 4.4 : AUC (%) and SE results.

DATA	DSVM		DDL		DBSSVM		DBSDL		DRSSVM		DRSDL	
	AUC	SE	AUC	SE	AUC	SE	AUC	SE	AUC	SE	AUC	SE
Tiger	83	2.71	80	3.16	88.75	2.66	89.2	2.14	84.5	2.54	**92.05**	2.05
Fox	63	2.81	57	2	**85.70**	2.30	84.7	2.67	68.85	2.36	71.5	2.16
Elephant	71.50	2.59	80	4.08	89.35	2.62	91.55	2.75	89.95	2.19	**93.8**	2.21
Musk2	85.65	3.29	74.11	4.83	**96.46**	2.06	92.77	2.17	90.33	3.21	79.73	4.12
Musk1	88	2.32	78.	3.94	**98.40**	1.06	96.60	2.0	95.75	1.75	90.89	3.3
alt.atheism	70	2.98	73.17	3.39	80.42	2.51	91.73	2.27	76.70	3.76	**97.60**	1.36
comp.graphics	52	2.0	50	0	53.25	2.24	90.23	3.29	52	2.0	**92.9**	2.45
rec.autos	58.50	2.79	64.92	3.82	61.05	3.43	88.83	3.74	61.90	2.81	**97.27**	1.08
sci.crypt	68.67	3.81	72.92	4.11	82.60	3.3	86.87	2.84	78.28	3.36	**97.87**	0.95
sci-med	64.92	3.10	70.92	3.63	75.27	4.16	92.38	1.27	78.05	4.46	**96.27**	1.75
talk.politics.guns	65.67	2.80	58	2.13	88.30	3.23	**93.65**	2.3	89.30	2.83	92.23	2.44

As it is shown in Figure 4.1, comparing DRSSVM to DBSSVM, these two methods in most case show approximately similar results but it seems that the DRSSVM slightly performs better. Comparison to the DBSDL in Figure 4.2, it is clear that in most cases DBSDL has higher performance. The performance of DL in Random subspace is given

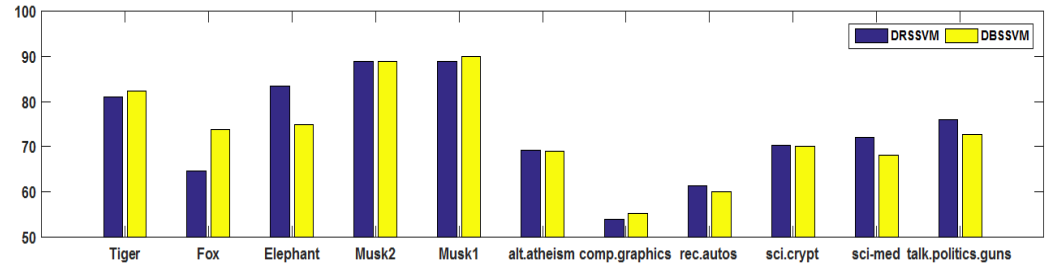


Figure 4.1 : Comparing the accuracy performance of DRSSVM to DBSSVM.

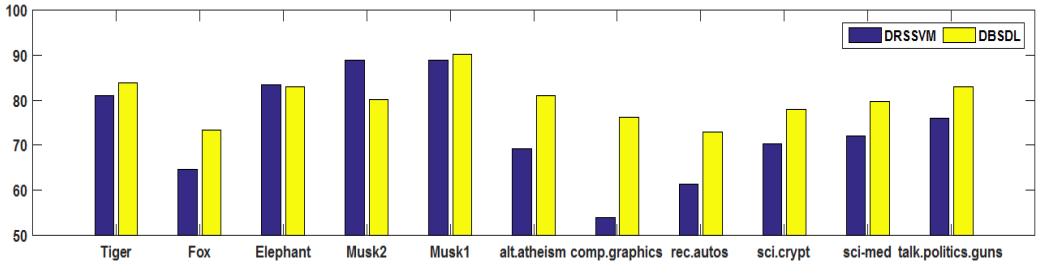


Figure 4.2 : Comparing the accuracy performance of DRSSVM to DBSDL.

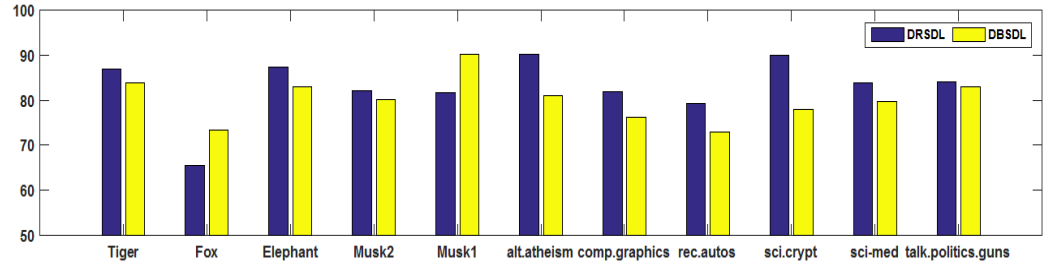


Figure 4.3 : Comparing the accuracy performance of DRSDL to DBSDL.

at the pervious paragraph and here we see it in bagging Ensemble method. Note that the DL increases the classification accuracy of each Ensemble method.

Comparing DRSDL to DBSDL in Figure 4.3, and DRSSVM to DBSSVM in Figure 4.1, it is clear that Random Subspace Ensemble method has better performance comparison to the Bagging Ensemble method. It means that instance-based analyzing provides us better information than bag-based.

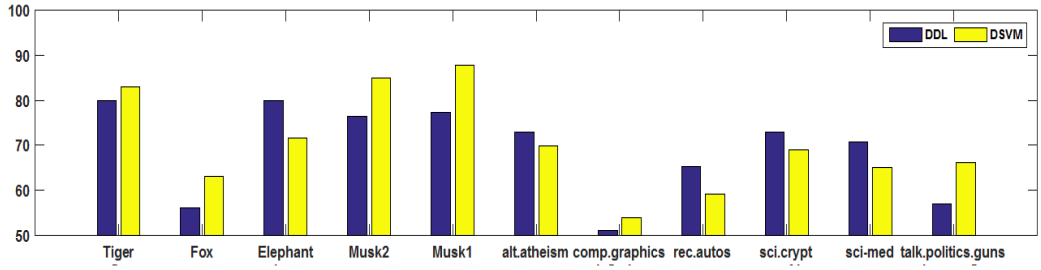
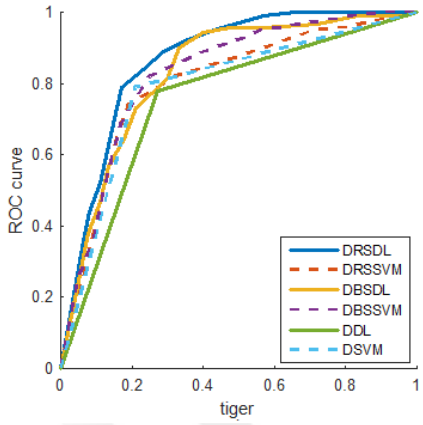


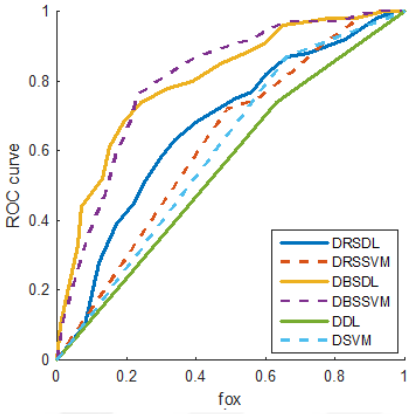
Figure 4.4 : Comparing the accuracy performance of DDL to DSVM.

To illustrate the constructive performance of Ensembles, one can compare the results of DSVM and DDL methods in Figure 4.4, where only single classifiers are trained on all the dissimilarity values. It is clear that the performance is decreased significantly. Comparing DDL and DSVM, it seems that SVM slightly performs better than DL. Contradict with SVM which has better performance as a single classifier, combining DL with Ensembles result in high performance classification accuracy. It seems that DL Ensemble methods generate more diverse classifiers comparison to the SVM. It is notable that in Ensemble methods what matters is not about using high performance classifiers as a base classifier, it is about applying methods which generate more diverse classifiers at each subspace.

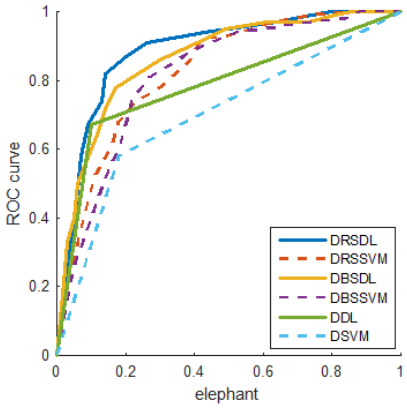
The area under curve performance of all methods for each dataset can be seen in Figure 4.5. As it is clear, DRSDL has the best AUC performance in most cases. Following this method, DBSDL is the second method that has better AUC performance compared to the remained approaches(DRSSVM, DBSSVM, DSVM, DDL).



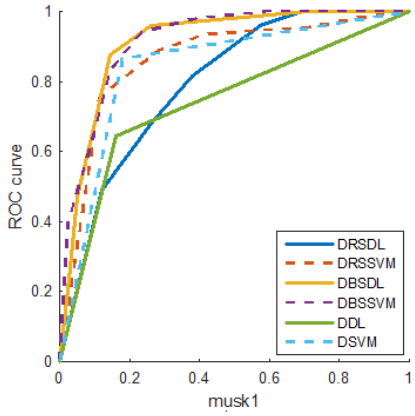
(a) Tiger



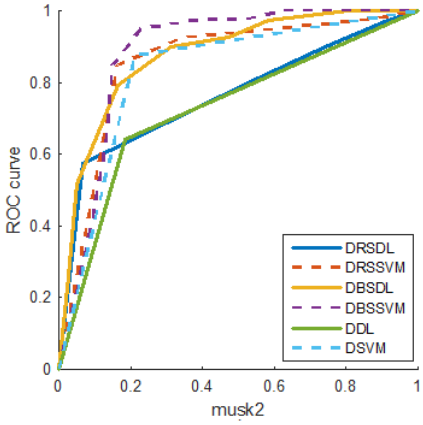
(b) Fox



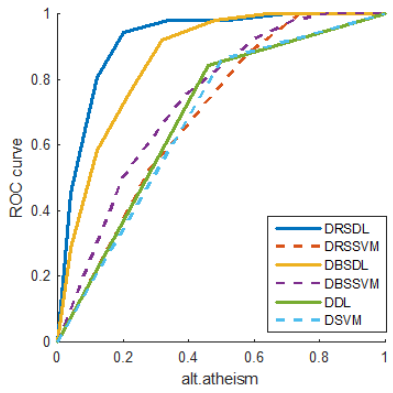
(c) Elephant



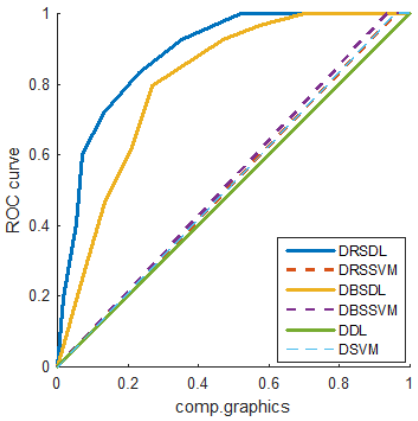
(d) Musk1



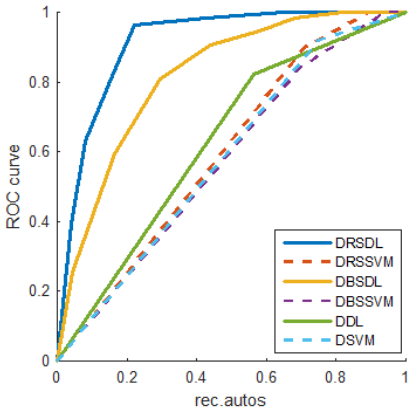
(e) Musk2



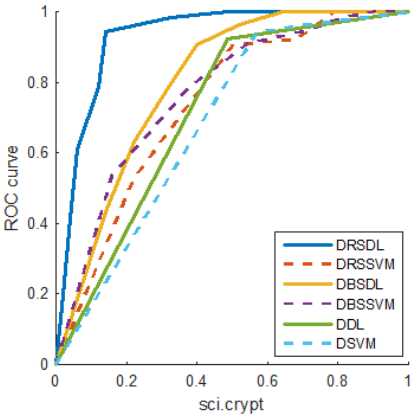
(f) alt.atheism



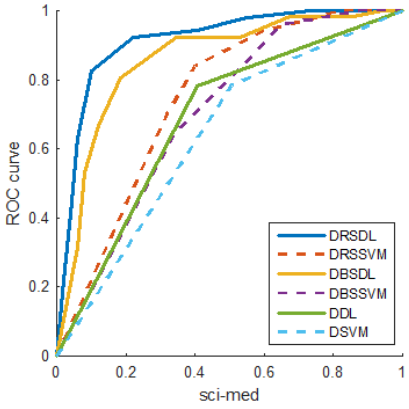
(g) comp.graphics



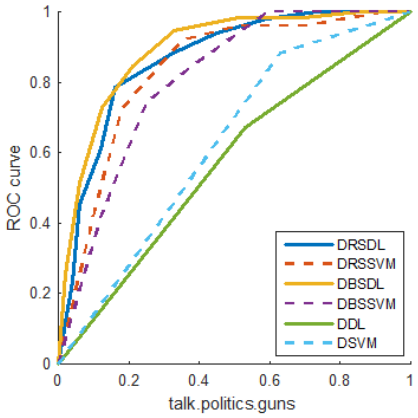
(h) rec-autos



(i) sci-crypt



(j) sci-med



(k) talk.politics.guns

Figure 4.5 : Area under curve results.

5. CONCLUSION

In this work, Dissimilarity based Dictionary Ensemble Learning methods are implemented for Multi Instance Learning problems. We evaluated the proposed method on three different Multi Instance Learning problem fields; Image categorization, drug activity prediction and text classification on 11 datasets. Multi instance datasets are demonstrated with bags containing vectors of instances. To convert the multi instance problem data sets into simple vectorial forms, dissimilarity technique is used. In this case dissimilarity of each bag to the instances are calculated and the calculated values, are used as a features to represent the bags. As result, each bag is described with a simple vector of dissimilarity values. As a second contribution, the advantages of dictionary learning and sparse coding are taken into consideration. In this phase, two separate dictionaries are constructed for negative and positive bags. Using these dictionaries, the bags are represented with their relative sparse code as simple as possible and classified simultaneously. As a third contribution, Random Subspace and Bagging are combined with these techniques. By this way, two method DRSDL and DBSDL are proposed. in DRSDL the instances are selected randomly where in DBSDL the bags are selected randomly. These two methods are compared to one of the new successful technique DRSSVM in MIL field which outperforms most of the previously proposed MIL algorithms like MILES, EM-DD, MI-SVM and etc. The experimental results show that our proposed methods outperform DRS-SVM. To have an accurate perception of DL role in Ensembles, either DBS-SVM which is bagging method with SVM classifier, DDL and DSVM which cope with a dataset using a single classifier, are taken into consideration. Analyzing the results show that single SVM may slightly perform better than DL. Whereas combining with Ensemble methods, DL learning gives us higher performances either in Random Subspace or Bagging methods in most cases. Another information you can get from these results is that, Random subspace is more powerful than Bagging in dissimilarity based MIL problems which means that instance selection analysis give more information about the structure of the dataset and result in higher classification accuracies. The importance of selecting

appropriate classifier has to be noted. As it is shown in results, the SVM has better performance compared to DL, but when they are combined with Ensembles, the DL Ensembles out-perform SVM Ensembles. This can be explained by considering the diversity of classifiers. One can obtain better classifiers when diverse enough classifiers are combined. As a future work we will investigate the diversity of classifiers in ensembles and their effects to overall performance.

REFERENCES

[1] **Bahlmann, C.**, **Haasdonk, B. and Burkhardt, H.** (2002). Online handwriting recognition with support vector machines-a kernel approach, *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*, IEEE, pp.49–54.

[2] **Ben-Hur, A.** *et al.* (2012). Multiple instance learning of Calmodulin binding sites, *Bioinformatics*, *28*(18), i416–i422.

[3] **Zha, Z.J.**, **Hua, X.S.**, **Mei, T.**, **Wang, J.**, **Qi, G.J. and Wang, Z.** (2008). Joint multi-label multi-instance learning for image classification, *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, pp.1–8.

[4] **Kotzias, D.**, **Denil, M.**, **De Freitas, N. and Smyth, P.** (2015). From group to individual labels using deep features, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.597–606.

[5] **Foulds, J. and Frank, E.** (2010). A review of multi-instance learning assumptions, *The Knowledge Engineering Review*, *25*(01), 1–25.

[6] **Wang, J. and Zucker, J.D.** (2000). Solving multiple-instance problem: A lazy learning approach.

[7] **Chen, Y.**, **Bi, J. and Wang, J.Z.** (2006). MILES: Multiple-instance learning via embedded instance selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 1931–1947.

[8] **Meyer, Y.**, **Sellan, F. and Taqqu, M.S.** (1999). Wavelets, generalized white noise and fractional integration: the synthesis of fractional Brownian motion, *Journal of Fourier Analysis and Applications*, *5*(5), 465–494.

[9] **Bracewell, R. and Kahn, P.B.** (1966). The Fourier transform and its applications, *American Journal of Physics*, *34*(8), 712–712.

[10] **Lu, J.**, **Plataniotis, K.N. and Venetsanopoulos, A.N.** (2003). Regularized discriminant analysis for the small sample size problem in face recognition, *Pattern Recognition Letters*, *24*(16), 3079–3087.

[11] **Rubinstein, R.**, **Bruckstein, A.M. and Elad, M.** (2010). Dictionaries for sparse representation modeling, *Proceedings of the IEEE*, *98*(6), 1045–1057.

[12] **Aharon, M.**, **Elad, M. and Bruckstein, A.** (2006). The K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, *IEEE Transactions on signal processing*, *54*(11), 4311–4322.

[13] **Peyré, G.** (2009). Sparse modeling of textures, *Journal of Mathematical Imaging and Vision*, *34*(1), 17–31.

[14] **Yeh, C.C.M. and Yang, Y.H.** (2012). Supervised dictionary learning for music genre classification, *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ACM, p. 55.

[15] **Ramirez, I.**, **Sprechmann, P. and Sapiro, G.** (2010). Classification and clustering via dictionary learning with structured incoherence and shared features, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, pp.3501–3508.

[16] **Polikar, R.** (2006). Ensemble based systems in decision making, *IEEE Circuits and systems magazine*, *6*(3), 21–45.

[17] **Cheplygina, V.**, **Tax, D.M. and Loog, M.** (2016). Dissimilarity-based ensembles for multiple instance learning, *IEEE transactions on neural networks and learning systems*, *27*(6), 1379–1391.

[18] **Zhao, Z.**, **Fu, G.**, **Liu, S.**, **Elokely, K.M.**, **Doerksen, R.J.**, **Chen, Y. and Wilkins, D.E.** (2013). Drug activity prediction using multiple-instance learning via joint instance and feature selection, *BMC bioinformatics*, *14*(Suppl 14), S16.

[19] **Amores, J.** (2013). Multiple instance classification: Review, taxonomy and comparative study, *Artificial Intelligence*, *201*, 81–105.

[20] **Dietterich, T.G.**, **Lathrop, R.H. and Lozano-Pérez, T.** (1997). Solving the multiple instance problem with axis-parallel rectangles, *Artificial intelligence*, *89*(1), 31–71.

[21] **Andrews, S.**, **Tsochantaridis, I. and Hofmann, T.** (2002). Support vector machines for multiple-instance learning, *Advances in neural information processing systems*, pp.561–568.

[22] **Zhang, C.**, **Platt, J.C. and Viola, P.A.** (2005). Multiple instance boosting for object detection, *Advances in neural information processing systems*, pp.1417–1424.

[23] **Leistner, C.**, **Saffari, A. and Bischof, H.** (2010). MIForests: Multiple-instance learning with randomized trees, *European Conference on Computer Vision*, Springer, pp.29–42.

[24] **Ramon, J. and Raedt, L.D.**, (2000), Multi Instance Neural Networks.

[25] **Blockeel, H.**, **Page, D. and Srinivasan, A.** (2005). Multi-instance tree learning, *Proceedings of the 22nd international conference on Machine learning*, ACM, pp.57–64.

[26] **Raykar, V.C.**, **Krishnapuram, B.**, **Bi, J.**, **Dundar, M. and Rao, R.B.** (2008). Bayesian multiple instance learning: automatic feature selection and inductive transfer, *Proceedings of the 25th international conference on Machine learning*, ACM, pp.808–815.

[27] **Bryt, O. and Elad, M.** (2008). Compression of facial images using the K-SVD algorithm, *Journal of Visual Communication and Image Representation*, *19*(4), 270–282.

[28] **Tosic, I. and Frossard, P.** (2011). Dictionary learning, *IEEE Signal Processing Magazine*, *28*(2), 27–38.

[29] **Sprechmann, P. and Sapiro, G.** (2010). Dictionary learning and sparse coding for unsupervised clustering, *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp.2042–2045.

[30] **Ho, T.K.** (1995). Random decision forests, *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, IEEE, pp.278–282.

[31] **Chawla, N.V. and Bowyer, K.W.** (2005). Random subspaces and subsampling for 2-d face recognition, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, IEEE, pp.582–589.

[32] **Kuncheva, L.I.**, **Rodríguez, J.J.**, **Plumpton, C.O.**, **Linden, D.E. and Johnston, S.J.** (2010). Random subspace ensembles for fMRI classification, *IEEE transactions on medical imaging*, *29*(2), 531–542.

[33] **Xia, J.**, **Dalla Mura, M.**, **Chanussot, J.**, **Du, P. and He, X.** (2015). Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles, *IEEE Transactions on Geoscience and Remote Sensing*, *53*(9), 4768–4786.

[34] **Lai, C.**, **Reinders, M.J. and Wessels, L.** (2006). Random subspace method for multivariate feature selection, *Pattern recognition letters*, *27*(10), 1067–1076.

[35] **Zhang, J.** (1999). Inferential estimation of polymer quality using bootstrap aggregated neural networks, *Neural networks*, *12*(6), 927–938.

[36] **Dettling, M.** (2004). BagBoosting for tumor classification with gene expression data, *Bioinformatics*, *20*(18), 3583–3593.

[37] **West, D.**, **Dellana, S. and Qian, J.** (2005). Neural network ensemble strategies for financial decision applications, *Computers & operations research*, *32*(10), 2543–2559.

[38] **Hsieh, N.C. and Hung, L.P.** (2010). A data driven ensemble classifier for credit scoring analysis, *Expert systems with Applications*, *37*(1), 534–545.

[39] **Zhu, Z.**, **Chen, Q. and Zhao, Y.** (2014). Ensemble dictionary learning for saliency detection, *Image and Vision Computing*, *32*(3), 180–188.

[40] **Gärtner, T.**, **Flach, P.A.**, **Kowalczyk, A. and Smola, A.J.** (2002). Multi-Instance Kernels., *ICML*, volume 2, pp.179–186.

[41] **Zhang, Q. and Goldman, S.A.** (2001). EM-DD: An improved multiple-instance learning technique, *Advances in neural information processing systems*, pp.1073–1080.

[42] **Bousmina, A.**, **Jlassi, C. and Arous, N.** (2016). Combining ensemble methods of Bagging, Subagging and Random Subspace for phoneme recognition, *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, IEEE, pp.677–682.

[43] **Efron, B. and Tibshirani, R.J.** (1994). *An introduction to the bootstrap*, CRC press.

[44] **Cheplygina, V.**, Multi instance learning data sets, `http://www.miproblems.org/datasets/`, accessed: 2016-08-12.

CURRICULUM VITAE

Name Surname: Nazanin Moarref

Place and Date of Birth: Iran, 07.09.1990

E-Mail: moarref@itu.edu.tr

EDUCATION:

- **B.Sc.:** 20013, University of Tabriz, Faculty of Electrical and Computer engineering, Faculty of Electrical and Computer Engineering

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Tuysuzoglu G., Moarref N., and Yaslan Y., 2016: Ensemble Based Classifiers Using Dictionary Learning. The 23rd International Conference on Systems, Signals and Image Processing, May 23-25, 2016 Bratislava, Slovakia.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Tuysuzoglu G., Moarref N., Cataltepe Z., Misirli A. T., and Yaslan Y., 2015: Analysing Graduation Project Rubrics Using Machine Learning Techniques. In Computer Science& Education (ICCSE), 2015 10th International Conference on (pp. 19-24). IEEE.
