**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# SPARSE CODING BASED ENSEMBLE CLASSIFIERS COMBINED WITH ACTIVE LEARNING FRAMEWORK FOR DATA CLASSIFICATION

**M.Sc. THESIS**

**Göksu TÜYSÜZOĞLU**

**Department of Computer Engineering**

**Computer Engineering Programme**

**JUNE, 2016**

**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# SPARSE CODING BASED ENSEMBLE CLASSIFIERS COMBINED WITH ACTIVE LEARNING FRAMEWORK FOR DATA CLASSIFICATION

**M.Sc. THESIS**

**Göksu TÜYSÜZOĞLU**
**(504131520)**

**Department of Computer Engineering**

**Computer Engineering Programme**

**Thesis Advisor: Asst. Prof. Dr. Yusuf YASLAN**

**JUNE, 2016**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# VERİ SINIFLANDIRMA İÇİN AKTİF ÖĞRENME ÇERÇEVESİ İLE BİRLEŞTİRİLMİŞ AYRIK KODLAMA TABANLI SINIFLANDIRICI TOPLULUKLARI

**YÜKSEK LİSANS TEZİ**

**Göksu TÜYSÜZOĞLU**
**(504131520)**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Tez Danışmanı: Yrd. Doç. Dr. Yusuf YASLAN**

**HAZİRAN, 2016**

Göksu TÜYSÜZOĞLU, a M.Sc. student of ITU Graduate School of Science Engineering and Technology student ID 504131520, successfully defended the thesis entitled "SPARSE CODING BASED ENSEMBLE CLASSIFIERS COMBINED WITH ACTIVE LEARNING FRAMEWORK FOR DATA CLASSIFICATION", which 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**<br>İstanbul Technical University | ..... |
| **Jury Members :** | **Assoc. Prof. Dr. Sanem SARIEL**<br>İstanbul Technical University | ..... |
| | **Asst. Prof. Dr. Tolga ENSARİ**<br>İstanbul University | ..... |

**Date of Submission : 2 May 2016**
**Date of Defense : 10 June 2016**

***To my father,***

**FOREWORD**

This thesis was written for my Master degree in Computer Engineering at Istanbul Technical University. I would like to thank my supervisor Asst. Prof. Dr. Yusuf Yaslan for giving me valuable advice, guidance and support always when needed. Other important persons considering my thesis are Nazanin Moarref and all the professors in the Department of Computer Engineering at Istanbul Technical University whom I like to give my special thanks for giving me an opportunity to share their knowledge and the best available information.

Finally, I like to show my gratitude to thank my family and also my friends and colleagues, Gamze Doğan, Beyza Eken, Mine Yasemin and Kübra Cengiz for their constant support and encouragement to boost my positive energy and motivation throughout the process of writing my thesis.

May 2016

Göksu TÜYSÜZOĞLU
(Research Assistant)

**TABLE OF CONTENTS**

**Page**

## ABBREVIATIONS

**ARDL** : Active Random Subspace Dictionary Learning
**ARSVM** : Active Random Subspace Support Vector Machines
**ABDL** : Active Bagging Dictionary Learning
**ABSVM** : Active Bagging Support Vector Machines
**BDL** : Bagging Dictionary Learning
**BSVM** : Bagging Support Vector Machines
**DL** : Dictionary Learning
**RDL** : Random Subspace Dictionary Learning
**RS** : Random Subspace Feature Selection
**RSVM** : Random Subspace Support Vector Machines
**SVM** : Support Vector Machines

## LIST OF TABLES

**Page**

## LIST OF FIGURES

**Page**

## SPARSE CODING BASED ENSEMBLE CLASSIFIERS COMBINED WITH ACTIVE LEARNING FRAMEWORK FOR DATA CLASSIFICATION

## SUMMARY

Nowadays, along with the need for classification algorithms in various areas concerning machine learning such as text classification, image categorization, audio and music genre classification, new classifier models are developed and works for improving the existing ones increasingly go on. In this direction, as dictionary learning algorithm which represents signals or each problem instance at hand with sparse linear combinations of basis elements of a dictionary is also utilized in data classification and clustering, it is used in signal, image, audio and video processing applications.

In the dictionary learning model, which sparse coding and dictionary update steps are practiced and this process continues until a predetermined convergence level is attained in an iterative fashion. The main purpose is to obtain the framework of a dictionary that provides the sparsest representation while decreasing the reconstruction error.

The process where a number of classifiers are modeled and decisions from each one produce a single output by a combination rule is known as ensemble learning. In literature, ensemble learning algorithms is performed both in feature subspace and instance subspace. Random subspace feature selection and bagging are the mostly applied ensemble learning methods in feature subspace and in instance subspace respectively.

On the other hand, possibility of access to huge amount of unlabeled data has been increased along with getting easy access to data. Active learning, which is proposed for this type of problems, is a learning method in which the most informative instances from the unlabeled data are chosen, then labeled by an oracle and after then added to the training set.

At the stage of establishing the active learning framework, evaluation of the unlabelled data and how to select the most informative ones among them is an important question. One of the easiest ways is to select the signals where the classifier is least certain about their class labels in the query phase. This method is known as uncertainty sampling. One of the most popular maximal uncertainty sampling techniques is based on entropy. The more entropy in the distribution, the more uncertain the choice of class label for that data value, and the more informative that query would be.

In the first stage of this study, dictionary learning is applied in combination with random subspace feature selection and bagging ensemble models. Then, comparisons of the experimental results with support vector machine, which is one of the best classifier models, and its ensemble combinations are maintained.

According to ten-fold cross validation experimental results obtained on eleven datasets from various area of specialization taken from UCI machine learning

repository and OpenML, dictionary learning based ensemble classifiers, especially BDL algorithm, present more successful classification performance than both of SVM and its classifier ensembles. Considering the experimental results, BDL outperforms other applied methods in 4 out of 11 datasets and in 2 datasets it performs the best with the other two methods DL and RDL. As a consequence, we can infer that randomly selecting instance subspaces while constructing dictionary models has a positive effect on the classification accuracy of the established methods.

In the second stage, all the dictionary base proposed methods and support vector machine counterparts are combined with active learning framework in which the most informative unlabelled training instances are labeled and integrated into the labeled training set in each learning iteration. While predicting the class labels of the test examples, the decision is made applying majority voting. After examining the experimental results, it is evident that classification accuracy mostly increases as the number of iterations goes up by the selection of training instances intelligently. Regarding to the best results obtained for each dataset by applied models, while ARDL outperforms ARSVM's classification performance, ABSVM succeeds better results than ABDL.

After obtaining the experimental results, an important part to handle is to measure the significance of the hypotheses which put forward the equivalency of the applied methods based on classification accuracies. In this direction, Friedman and Wilcoxon signed rank test results were obtained both for the ensemble learning part and methods under active learning framework. According to outcomes from the Friedman significance tests, ARDL, ARSVM, ABDL and ABSVM do not perform equivalently regarding to the best results obtained for each dataset.

On the other hand, Friedman significance tests and Wilcoxon signed rank tests applied to the accuracy results in the last iteration of active learning models are resulted in similar classification performance in the predetermined confidence interval. In the last part, Friedman test is practiced among DL and SVM classifiers and their ensemble models. Because there is an equivalency between classification performance differences, Wilcoxon signed rank test is applied to see pairwise model differences. As a result, DL/RDL, DL/BDL and SVM/BSVM pairs have significant differences while the other model couples performs in the same manner.

# VERİ SINIFLANDIRMA İÇİN AKTİF ÖĞRENME ÇERÇEVESİ İLE BİRLEŞTİRİLMİŞ AYRIK KODLAMA TABANLI SINIFLANDIRICI TOPLULUKLARI

## ÖZET


Günümüzde metin sınıflandırma, görüntü kategorizasyonu, ses ve müzik türü sınıflandırması gibi makine öğrenmesi konusunda farklı disiplinlerden pek çok alanda sınıflandırma algoritmalarına olan ihtiyaç bir hayli artmıştır. Bu amaçla yeni sınıflandırıcı modeller geliştirilmekte ve mevcut algoritmaları da iyileştirme çalışmaları çoğalarak devam etmektedir.

Sinyalleri ya da elimizde bulunan her bir problem örneğini bir sözlüğün temel elemanlarının ayrık doğrusal kombinasyonları olarak temsil etmekte olan sözlük öğrenme algoritmasından da bu doğrultuda veri sınıflandırma ve kümeleme alanlarında çokça faydalanılmakta olup sinyal, görüntü, ses ve video işleme uygulamalarında kullanılmaktadır.

İki aşamada gerçekleştirilen sözlük öğrenmesi modelinde ayrık kodlama ve sözlük güncelleme adımları uygulanmakta ve belirli bir yakınsama elde edene kadar bu süreç iteratif olarak devam etmektedir. Ana amaç, yeniden yapılandırma hatasını azaltarak en çok ayrık gösterimi veren sözlük yapısını elde etmektir.

Birçok sınıflandırıcının modellendiği ve her birinden gelen kararların birleştirilerek tek bir çıktı ürettiği süreç topluluk öğrenme olarak bilinir. Literatürde makine öğrenmesi uygulamalarının çoğunda sınıflandırıcı topluluklar tek sınıflandırıcı yöntemlerinden daha iyi başarım gösterebilmektedir. Topluluk öğrenme algoritmaları hem örnek hem de öznitelik alt uzaylarında uygulanabilmektedir. Random subspace algoritması öznitelik uzayında ve bagging algoritması da örnek uzayında en çok uygulanan topluluk öğrenme yöntemlerindendir.

Öte yandan veriye erişimin kolaylaşması ile birlikte çok büyük miktarda etiketsiz veriye erişim imkânı doğmuştur. Bu tür problemler için sunulan aktif öğrenme, etiketi bilinmeyen veriler içerisinden en çok bilgi verici örnekleri seçip uzmanlar tarafından etiketleyerek eğitim kümesi içine katan bir öğrenme yöntemidir.

Aktif öğrenme yapısının kurulması aşamasında etiketsiz verilerin değerlendirilip içlerinden en bilgi verici olanlarının nasıl seçileceği önemli bir sorudur. En kolay yollardan biri, örnekleri sorgulayarak sınıflandırıcı modelin sınıf etiketi konusunda en az emin olduğu sinyallerin seçilmesidir ve bu yöntem belirsizlik örnekleme (uncertainty sampling) olarak bilinir. Belirsizlik örnekleme teknikleri içinde en popüler olanlarından biri düzensizlik hesabını temel alır. Bir dağılımda ne kadar fazla düzensizlik varsa, o veri için sınıf etiketi seçimi de o derecede kararsızlık içerir ve sorgulama da o kadar bilgi verici olur.

Bu çalışmanın ilk aşamasında sözlük öğrenme modeli, sınıflandırıcı topluluklarından random subspace feature selection ile öznitelik alt uzayında ve bagging ile örnek alt uzayında birleştirilerek uygulanmış ve bu sınıflandırıcılar Random Subspace Dictionary Learning (RDL) ve Bagging Dictionary Learning (BDL) olarak

adlandırılmıştır. Deneysel sonuçlarda önerilen yöntemlerin sınıflandırma başarımları en iyi sınıflandırıcı yöntemlerden biri olan destek vektör makinesi (Support Vector Machines - SVM) ve topluluk öğrenme tabanlı kombinasyonları (Random Subspace Support Vector Machines (RSVM) ve Bagging Support Vector Machines (BSVM)) ile birlikte karşılaştırılmıştır.

UCI makine öğrenmesi veri havuzundan ve OpenML' den alınan çeşitli alanlardan on bir farklı veri kümesi üzerinde elde edilen on kat çapraz sağlama deney sonuçlarına göre sözlük öğrenme tabanlı sınıflandırıcı toplulukları, özellikle de BDL algoritması, hem destek vektör makineleri hem de sınıflandırıcı topluluklarıyla birleştirilmiş modellerine göre daha başarılı sonuçlar ortaya koymuştur.

Sınıflandırma başarımlarına bakıldığında, en başarılı yöntem olan BDL 11 veri kümesinin 4 tanesinde DL, RDL SVM, BSVM ve RSVM sınıflandırıcılarından üstün gelmekte, 2 tanesinde ise DL ve RDL ile en sonuçları elde etmektedir. Bu noktada örnek altuzaylarının rastgele seçilmesiyle oluşturulan sözlük modellerinin sınıflandırma başarımına olan pozitif etkisi gözlemlenmiştir.

İkinci aşamada ise uygulanan yöntemlerin her biri aktif öğrenme yapısı içerisinde kullanılmış, elde bulunan her bir sınıf için bir sözlük öğrenilerek, her iterasyonda en bilgi verici etiketsiz örnekleri etiketleyerek eğitim kümesine ekleme işlemi uygulanmıştır. Test aşamasında her yeni örnek için sınıf etiketi sözlük topluluklarının çoğunluğuna bakılarak atanmıştır.

İlk aşamada eldeki eğitim kümesinin %20'si alınarak hem sözlük tabanlı hem de destek vektör makinesi tabanlı sınıflandırıcı toplulukları modellenmiş, sonraki altı iterasyonda geriye kalan etiketsiz veriler içerisindeki en çok bilgi verici %10 örneğin düzensizlik hesabı dikkate alınarak seçilmesiyle eğitim kümesi güncellenmiştir. Böylelikle iterasyon sayısı arttıkça sınıflandırma başarımı da çoğunlukla artışa geçmiş, örneklerin akıllıca seçilmesiyle oluşturulan eğitim kümesi bu sonuçlarda etkili olmuştur.

Test sonuçlarında her bir veri kümesi için elde edilen en başarılı sonuçlar dikkate alınırsa, rastgele öznitelik seçimiyle oluşturulan sınıflandırıcı topluluklarına bakıldığında önerilen ARDL yönteminin ARSVM yönteminden daha başarılı olduğu görülmüştür. Örneklerin rastgele seçilmesiyle oluşturulan sınıflandırıcı toplulukları kullanıldığında ise ABSVM yöntemi ABDL yönteminden daha üstün gelmiştir.

Deney sonuçlarının elde edilmesinden sonra ilgilenilmesi gereken önemli bir nokta da uygulanan yöntemlerin sınıflandırma başarımları açısından birbirine denkliğini öne süren hipotezlerin anlamlılığının ölçülmesidir. Bu doğrultuda, Friedman test ve Wilcoxon signed rank test sonuçlarına bakılmıştır. Friedman anlamlılık testinden gelen çıktılara göre aktif öğrenme altında iterasyon bazında uygulanan metotlar için en iyi sonuçlar dikkate alındığında görülen odur ki sıfır hipotezi ($H_0$) kabul edilmemelidir, başka bir deyişle uygulanan yöntemler gösterdikleri performans açısından eşdeğer değildirler.

Aktif öğrenme algoritmalarının son iterasyonlarında elde edilen başarımlar için de Friedman ve Wilcoxon signed rank testleri uygulanmıştır. Her iki test sonucunda da model çiftlerinin eş sınıflandırma performansları sundukları kanısına varılmıştır. Öte yandan pasif öğrenme kısmında uygulanan yöntemler de Friedman testiyle incelendiğinde eşdeğer oldukları görülmüştür. Bunun ardından, hangi metot çiftlerinin kendi aralarında denk performans sunup sunmadıkları sorusuna çözüm bulmak amacıyla Wilcoxon signed rank test uygulanmıştır. Sonuçlara göre DL/RDL,

DL/BDL ve SVM/BSVM metot çiftleri sınıflandırma performansı olarak eşdeğer değildirler, diğer yöntemler ise denk sayılabilir.

# 1. INTRODUCTION

Nowadays, a plenty of latent information are available in databases to be exploited for intelligent decision making. These databases generally contain data that is related to a specific category or class. When data samples come with class labels, they are called training data which can be used to train a model for predicting the labels of new unseen data samples that are called test sets. This process is called classification and it is a concept to investigate to which class a new data point should belong under favour of using the training data. Han, Kamber and Pei (2011) express the idea as "Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown" (p. 24). The process is known as supervised learning, because a model is formed using predefined training data, which in fact is used as a supervisor to classify new test examples.

There are lots of domains where classification takes place such as text categorization, optical character recognition, fraud detection, market segmentation, face detection, classification of proteins. In order to construct a classifier model, many machine learning algorithms have been developed and as new researches are presented, many others arise day-to-day. Support vector machines, decision trees, naive bayes, nearest neighbor, multilayer perceptron and logistic regression are some examples among the most popular classification algorithms. In order to obtain a good classification accuracy one also needs to have a good feature representation. In literature there is a vast amount of research to represent the features in other dimensional spaces to improve the classification performance such as kernels (Lu et al, 2003), wavelet transformation (Van de Wouwer et al, 1999), frequency representation of time domain signals (Sejdić et al, 2009).

A number of media types such as imagery, video and acoustic can be sparsely represented by applying transform-domain methods (Elad, 2010). A lot of significant tasks related to such media can be handled finding sparse solutions to underdetermined systems of linear equations. Regarding this issue, sparse coding and

dictionary learning have recently aroused much interest by representing each problem instance as linear combinations of basis elements. These elements are called atoms and they compose a dictionary.

One of the major application areas for dictionary learning is in data representation and classification. It has been applied in many problem areas such as signal processing applications (the joint analysis of correlated signals like audio-visual signals and stereo images) (Tošić and Frossard, 2011), texture segmentation (Sprechmann and Sapiro, 2010), music genre classification (Yeh and Yang, 2012) and saliency detection (Zhu, Chen and Zhao, 2014). The basic model for classification is created via generating one sub-dictionary for each category to represent the instances of the respective class and then combining them to reach a unique dictionary. The resulting dictionary base is used as a classifier that assigns a class label which has the least reconstruction error and the sparsest representation.

Let us think of a scenario in which a group of doctors diagnose a certain disease for a patient. It is clear that the diagnosis is more reliable when the majority of doctors make the same decision on the patient's disease compared to the decision taken by the minority. From data classification perspective, in order to classify new test examples in a more accurate way, more than one classifier decisions can be integrated into the system and an agreement can be made on the final decision. This learning strategy is called ensemble learning and it is constituted by the combination of predictor/classifier model outputs, which produces a final decision for an unseen data point.

Ensemble learning methods are used for classification problems as well as regression. Classifier ensembles can be obtained either in feature space, instance space or classifier level. Boosting, bootstrap aggregating (bagging), stacking, random subspace feature selection, random forests and adaboost are among the most applied ensemble learning methods (Polikar, 2006).

Bagging is an instance-based ensemble learning method which generates subspaces of instances by applying random selection method with replacement. Each ensemble classifier produces a decision and the final prediction is their combined output. On the other hand, random subspace feature selection is a feature-based counterpart of bagging model, where a sub-group of features are randomly selected with

replacement to form ensemble classifiers. Taking advantage of the strengths of these two ensemble learning methods, classification problems can be solved more accurately and the variance of the individual classifiers are reduced.

Obtaining labeled training examples for classification problems is an expensive task while a massive chunk of unlabelled data is available to process. For instance, let us think of a case where we want to predict which web pages a person can find interesting. In order to do this, we need the data of web pages which were marked as favourite by this person. The more we know about the labeling information, we can predict better and present more appropriate pages to recommend. On the other side, people are generally not willing to hand-label all the pages they like even if there are a lot. Active learning is a largely used framework for these kind of situations. It has the ability to choose the most informative unlabeled examples automatically for human annotation. Liere and Tadepalli (1997) state the concept as "Active learning in its most general sense refers to any form of learning wherein the learning algorithm has some degree of control over the examples on which it is trained" (p. 591).

Up to the present, active learning framework has been applied with many different classifiers for text classification, image retrieval, advertisement removal (Sun and Hardoon, 2010), visual object detection (Abramson and Freund, 2004), natural language processing (Olsson, 2009) etc. To the best of our knowledge, active learning has not been applied as a classifier in active learning framework. In this study, dictionary learning is used as a base classifier for active learning and active learning's intelligent selection strategy is used to enhance the training set by choosing the most informative examples.

### 1.1 Purpose of Thesis

The aim of the thesis is to introduce a number of models for data classification which are generated by sparse coding based ensemble classifiers combined with active learning framework. The proposed models are examined under three main headings: dictionary learning, ensemble learning and active learning framework.

In order to represent the input data using as few components as possible, dictionary learning is proposed as a learning model for an effective representation. Exploring

a sparse representation of the input data in the form of a linear combination of basis elements and also discovering those basis elements (i.e. atoms of a dictionary) themselves is the main purpose of dictionary learning.

Another remarkable point for this thesis is to show the effect of ensemble learning methods on the proposed classifiers. Random subspace feature selection and bagging are selected as appropriate ensemble learning methods to boost the prediction ability of dictionary learning model. On the other side, comparisons with support vector machines, which is one of the state-of-the-art algorithms, and its classifier ensembles are also presented. Toward this goal, experiments are conducted on datasets with different number of features/instances from various scopes.

Other point of purpose on the following sections is to introduce active learning framework and integrate it into ensemble dictionary learning model. It helps performing classification in cases where few number of labeled and huge number of unlabeled training instances are available. Entropy is employed as an uncertainty sampling technique for pool-based active classifier models.

As the final contribution of this paper, several significance tests are demonstrated to detect differences in treatments across multiple test attempts. For this purpose, non-parametric Friedman tests are applied to the classification accuracies of the proposed methods.

### 1.2 Literature Review

In literature, dictionary learning and sparse coding have been applied in diverse areas such as signal, image, audio and video processing applications for dimensionality reduction (Schnass and Vandergheynst, 2008; Tošić and Frossard 2011), denoising (Elad and Aharon, 2006), image restoration (Mairal et al, 2008), and image compression (Bryt and Elad, 2008).

As dictionary learning doesn’t require estimating class distributions or computing margin between classes, it is also used for data classification and clustering applications where the feature vectors are computed as linear combinations of basis elements of a dictionary. Sapiro and Sprechmann (2010) developed a clustering framework in which a set of dictionaries are built for every cluster found in a given dataset. According to the proposed approach, dictionaries are formed by choosing the

ones which provide the best representation of the signals in a cluster and giving the sparsest solution. Besides, three standard datasets, the MNIST and USPS which are composed of handwritten digits and ISOLET which includes audio features from 150 speakers were used to show the discriminative aspect of dictionary learning model. The experimental results showed that the proposed dictionary learning model provides remarkable classification performance comparable with other sophisticated classification algorithms such as SVM and k-NN in terms of reconstruction and discrimination power.

On the subject of music genre classification, Yeh and Yang (2012) developed a technique enforced by dictionary learning to summarize short-time features (codebook) of recorded music over time, where codebook represents dictionary base. Dictionary base is made up of sub-dictionaries, one for each class to represent the characteristics of the instances in these classes. Other existing codebook generation methods such as conventional VQ-based and exemplar-based methods were compared with the proposed dictionary based method. The proposed method was shown superior to others on two benchmark datasets, GTZAN composed of clips covering ten genres and ISMIR2004Genre including songs covering six genres using just the log-power spectrogram as the local feature descriptor.

Tošić and Frossard (2011) presented dictionary learning and sparse approximation as a dimensionality reduction tool to find a representation adaptive to the proper inference of causes of the observed data. In addition, supervised dictionary learning was examined in a face recognition application by using the discriminative power of the sparse representation. The incoherency between the subspaces which represent data in different classes was taken into consideration.

Recently, ensemble methods have been used to improve the classification accuracy of single classifiers. Ensemble classifiers are created using the outputs of multiple classifiers that are trained on different training datasets created by various data resampling procedures or trained on a single training dataset by selecting different classifier parameters or classifiers.

Polikar (2006) reviewed ensemble based strategies such as bagging and its variations, boosting models, stack generalization etc. by emphasizing their importance in decision making process while dealing with classification problems.

Why we tend to prefer applying ensemble learning methods instead of single classifiers is explained from various perspectives such as reducing the risk of inaccurate predictions of a single model, ability of handling large volumes of data and easily applying divide and conquer technique in problems with complex decision boundaries.

Random subspace is one of the well-known ensemble learning methods. It was firstly introduced to construct a decision tree classifier by randomly chosen subspaces of the components of the feature vector (Ho, 1998). According to the applied selection strategy to form random subspaces, a number of different feature selection techniques has been proposed such as Univariate search technique (Chow et al, 2001), Base-pair selection (Bo and Jonassen, 2002), Forward selection (Bo and Jonassen, 2002), Recursive Feature Elimination, and Liknon.

Lai, Reinders and Wessels (2006) introduced an ensemble strategy in feature space by incorporating informativeness of features as a selection strategy in the construction of each subspace. Applied multivariate feature selection technique, Random Subspace Method, initially selects features randomly from the original feature space and then, a multivariate search technique, either Liknon or Recursive Feature Elimination, takes place in this reduced feature space by retrieving the informative features. This procedure is applied iteratively by covering the large portions of the original features. According to the experimental results which were carried out in artificial datasets, ensemble based random subspace model provides robustness and a powerful classification performance especially in small sample size problems. Many other studies have been made by applying random subspace ensemble for functional magnetic resonance imaging (fMRI) classification (Kuncheva et al, 2010), the bio-molecular diagnosis of malignancies (Bertoni, 2005) and bankruptcy prediction and credit scoring (Nanni and Lumini, 2009) etc.

Bagging is another ensemble learning strategy which uses randomly selected instance subspaces. There are various studies applying bagging for solving credit scoring and bankruptcy prediction (West, 2005), optical character recognition (Mao, 1998) and day-ahead electricity price prediction (Tian and Meng, 2010). Recently, Zhu et al. (2014) combined ensemble learning with dictionary learning model in order to detect visually salient regions of an image. Instead of modeling a universal dictionary, the developed bagging based dictionary learning framework (EDL) is

constructed by applying random selection of image samples in order to train dictionaries independently for each subspace. In this way, more flexible multiple sparse representations are obtained for each of the image patches. A reconstruction residual based model for atom reduction over the learned dictionary is presented to further boost the distinctness of salient patch from the one of background. The resulting decision is made upon considering the outputs from each ensemble subspace. To the best of our knowledge there isn't any paper that applies dictionary learning as base classifier for random subspace and bagging ensembles.

The possibility of access to huge amount of data has been increased along with getting easy access to data. On the other hand, the majority of the available data is mostly unlabeled in other words we do not have enough information about its class/category label. Active learning, which is proposed for this type of problems, is a learning method in which the most informative instances from the unlabeled data are chosen, then labeled by an oracle and after then added to the training set to be used in the model construction of classification.

Active learning can be categorized by its way of synthesizing queries either by stream-based (Cohn et al, 1994), pool-based (Lewis and Gale, 1994) or query synthesis (Angluin, 1988) methods. In this work, the focus is on pool-based active learning in which a large pool of instances are sampled then the base classifier chooses the best query to be labeled. There are numerous number of studies applying pool-based active learning for different purposes such as in the application of cancer diagnosis (Liu, 2004), image classification (Zhang and Chen, 2002) and speech recognition (Tur et al, 2005).

Tong and Koller (2001) performed classification using SVM under the active learning framework in a text classification problem to determine which pre-defined topic a given text document belongs to. In the active learning part, some number of unlabeled instances are selected and added to the training set after learning its class label using one type of pool based active learning strategy. Three query strategies that split the version space into equal parts was proposed and they were shown to outperform standard passive learning counterparts.

Sun (2010) developed an active learning model which takes the correlation values between features of different views under a multi-view setting. He applied canonical

correlation analysis to select the most informative instances to integrate them into the training phase in the further iterations. According to the proposed approach, it is assumed that one example per class is labeled. The experiments were conducted on text classification, advertisement removal and content-based image retrieval and it was showed that the proposed active learning model has superiority over the general random selection approach for labeling.

Xu et al. (2014) performed active learning for dictionary construction by choosing the most informative examples using the reconstruction and classification error as the query strategy. The selected instance is only used during the dictionary update step. According to the experimental results conducted on a number of datasets from UCI Machine Learning Repository and face recognition dataset, active dictionary learning with small size dictionary can achieve comparable performance with other machine learning methods.

### 1.3 Outline of the Thesis

The rest of the thesis is organized as follows: The next chapter introduces the applied methodology. In the first step, sparse signal representation, dictionary learning and support vector machines models are explained. The following step of Chapter 2 is devoted to the ensemble learning methods in general and provides detailed knowledge on random subspace feature selection and bagging ensemble classifiers. Furthermore, active learning framework is stated by expanding different sampling scenarios used throughout literature. In the next chapter, sparse coding based ensemble classifiers combined with active learning framework is proposed. Chapter 4 discusses datasets which have been used and toolboxes managed to obtain dictionary learning model and support vector machines classifiers. In Chapter 5 experimental results achieved and significance tests applied are explained while Chapter 6 concludes the thesis with a summary of key lessons learnt.

# 2. METHODOLOGY

## 2.1 Dictionary Learning and Sparse Signal Approximation

### 2.1.1 Sparse signal approximation

Sparse representations of signals have received a great deal of attentions in recent years. Sources of data such as voice signals, images, radar images or heart signals etc. carry overwhelming amounts of data in which relevant information is often more difficult to find than a needle in a haystack. In this direction, having a sparse representation plays a fundamental role in processing signals faster and simpler as few coefficients reveal the information we are looking for.

Let's define an input signal as $x \in R^n$, $D = [d_1, d_2, \ldots, d_k] \in R^{n \times k}$ as a dictionary composed of a set of normalized ($d_j^T dj = 1$) "basis vectors", and $\alpha \in R^k$ as the coefficient vector or the representation of the signal, also known as sparse code, then the sparse representation problem can be formulated as:

$$\min_\alpha \|\alpha\|_0 \text{ s.t. } x = D\alpha \tag{2.1}$$

where $\|\alpha\|_0$ indicates $l_0$ norm of the coefficient vector $\alpha$ and it represents the number of non-zero elements in $\alpha$.

An input signal x can be represented by a linear combination of the atoms of an overcomplete dictionary in which the number of basis vectors is greater than the dimensionality of the input. However, finding the sparsest representation for a signal in an overcomplete basis is a very difficult computational problem because it needs combinatorial search and it is in the category of NP-hard problems. In order to find the best approximate solution to this problem, instead of non-convex $l_0$ norm, $l_1$ norm can take place by making it convex that ensures the existence of a unique global minimum to the above problem. Other $l_p$ norms where p is in the range [0,1] are also possible by imposing a stronger form of sparsity, but they lead to non-convex problems therefore $l_1$ norm is commonly used. Generalized formula of the $l_p$ norm can be given as:

$$\|\alpha\|_p = \left( \sum_{i=1}^{k} |\alpha_i|^p \right)^{1/p} \tag{2.2}$$

where $\alpha = [\alpha_1, \ldots, \alpha_k]^T$ and after replacing the former sparsity formulation with the $l_1$ norm, sparse representation problem can now be represented as:

$$\min_\alpha \|\alpha\|_1 \text{ s.t. } x = D\alpha \tag{2.3}$$

In general, the system under consideration can be exposed to noise, $\epsilon$, where we need an alternative solution with some proximity between $D\alpha$ and x. It can be expressed as follows:

$$\alpha^* = \text{argmin}_\alpha \|\alpha\|_1 \text{ s.t. } \|D\alpha - x\|_2 \leq \epsilon \tag{2.4}$$

### 2.1.2 Dictionary Learning

The concept of dictionary learning is about the construction of dictionary directly from a set of existing data samples so that the learned dictionary can be well adapted to the purpose of sparse representation.

Actual dictionaries can be obtained by finding a solution to the following minimization problem:

$$\min_{D,\{\alpha_i\}_{i=1\ldots m}} \sum_{i=1}^{m} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \tag{2.5}$$

where each of $\{x_i\}_{i=1\ldots m}$ represents one input signal (data sample/instance) being classified, and $\lambda$ is the penalty parameter that balances the trade-off between the data fitting term which defines the reconstruction error and the regularization term which determines the sparsity of the decomposition.

The optimization problem in equation 2.5 is usually not jointly convex concerning variables D and $\alpha$. One solution is to fix one of them, either D or $\alpha$, so that the objective function with respect to the other variable can turn into a convex function. In this direction, the optimization algorithm is made up of two convex steps which are applied in an iterative approach until a predetermined convergence criterion is met:

- *Sparse Approximation*: Dictionary D is considered fixed, then coefficients $\{\alpha_i\}_{i=1...m}$ of signal x with respect to dictionary D are calculated by minimizing equation 2.5.
- *Dictionary Update*: New dictionaries are computed using the obtained sparse coding matrix α in order to reduce the approximation error.

### 2.1.3 Supervised Dictionary Learning

Dictionary learning methods can be organized in a way that it can provide both reconstructive and discriminative purposes. Discriminative dictionary learning brings about the task of supervised classification of input signals by the inclusion of the class labels. Using the labels of training data ensures different data representations for each class by making the classification task easier. The aim of the sparse coding step is to find the sparsest representation of the data that has least reconstruction error. Both sparse representations and reconstruction error are considered for classification.

In order to realize the classification phase, actual dictionary is decomposed into sub-dictionaries each of which is trained independently with the involvement of the instances of a particular class. When we consider a training data consisting of c class labels, the corresponding dictionary base D is constructed using n sub-dictionaries as $[D_1, D_2, ..., D_c]$ and each of them is to represent one class with the same number of instances. In case of classifying a new test input which we have no idea about its class label beforehand, actual dictionary that is the combination of class-specific sub-dictionaries is used to encode the signal. The signal is then assigned to the class for which the best reconstruction is obtained and the one leading to the sparsest solution.

If we express the idea in more detailed way, classification of a signal x given a collection of dictionaries $[D_1, D_2, ..., D_c]$ where each $D_i \in R^{n \times k}$ can be fulfilled by performing the following steps iteratively and it is displayed in Figure 2.1.

- Compute the representation of the signal x in each dictionary $D_i$, which are $\alpha_1$, $\alpha_2$, ..., $\alpha_c$, using sparse coding
- Find the class membership of the signal x by comparing the cost of the representations, which are found in the previous step, and assigning it to the dictionary $D_i$ which delivers the least cost:

$$\text{class } i^* = \arg\min_{i\epsilon\{1,\ldots,c\}} \delta_i(x) \tag{2.6}$$

$$\text{where } \delta_i(x) = \min_{\alpha\epsilon R^k} \|x - D_i\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{2.7}$$

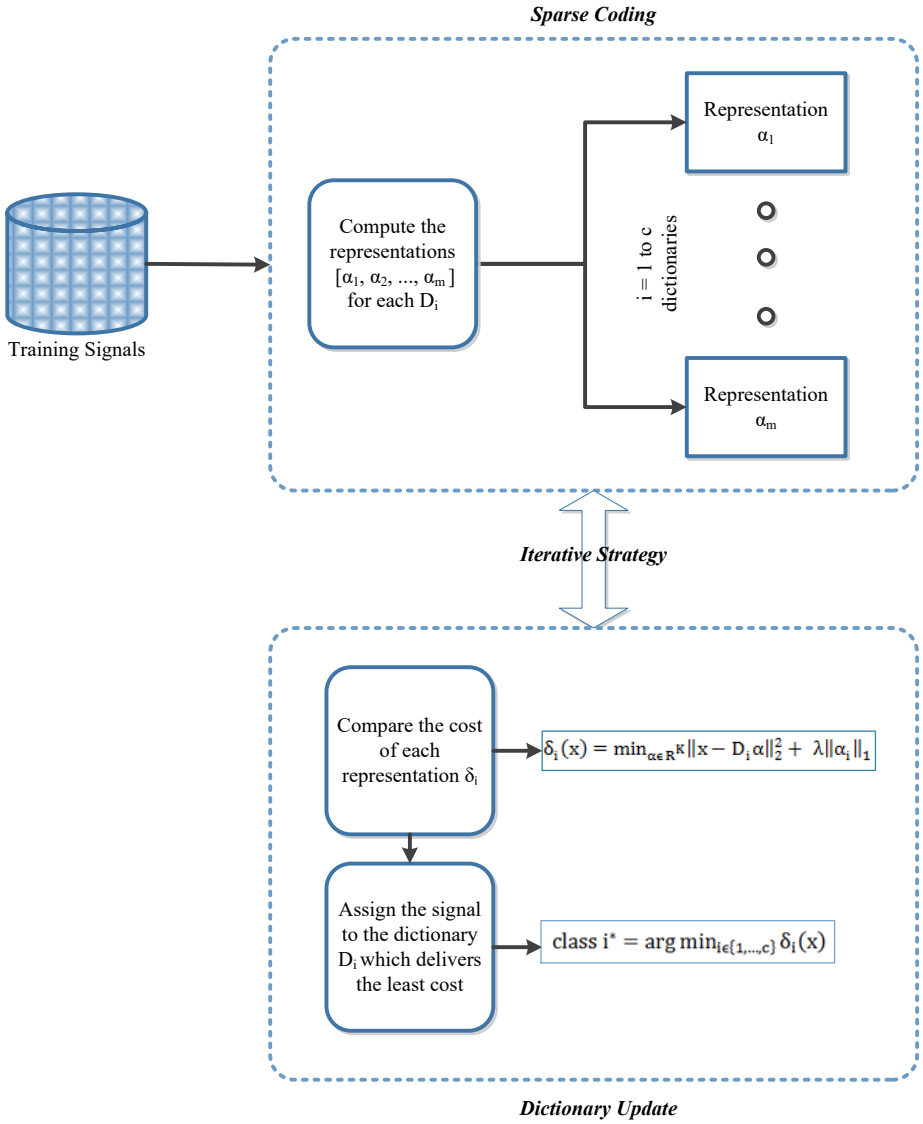


**Figure 2.1 :** Iterative dictionary learning framework.

## 2.2 Support Vector Machines

Support Vector Machines (SVM) is one of the state-of-the-art algorithms which is applied in solving classification and regression problems, feature selection and other machine learning tasks. A lot of real world problems such as bankruptcy prognosis, face detection, analysis of DNA microarrays and breast cancer diagnosis and prognosis can be dealt with by inclusion of an SVM model.

The aim of SVM is to maximize the margin between each class so that a good generalization performance on unseen test instances can be obtained. Although the subject was introduced in the late seventies (Vapnik, 1979), it has been receiving increasing attention, and so the time appears suitable for an introductory review.

In Figure 2.2, there is a classification problem for two class (+, -) dataset. The aim is to find a hyperplane so that "+" data points take place in one side and "-" ones are placed in the opposite side of this separator. SVM uses a flexible representation of the class boundaries. For each side, the data points which are located on the boundaries where the hyperplane is maximally distant from them are called *support vectors* and the gap between hyperplane and a support vector is known as *margin*. Campbell and Ying (2011) states SVM generalization error, i.e. the upper bound as "the bound is minimized by maximizing the margin ($\frac{2}{\|\mathbf{w}\|_2}$, where w is a normal vector to bounding planes) i.e., the minimal distance between the hyperplane separating the two classes and the closest data points to the hyperplane" (Chapter 1, p. 2).

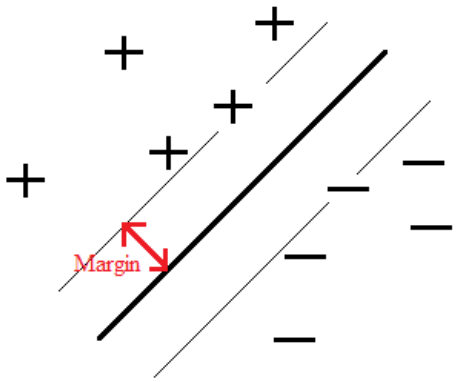


**Figure 2.2 :** Two class data points linearly separable by a hyperplane.

According to the Figure 2.2, a linear support vector machine is illustrated and the hyperplane is constructed using a simple formulation w . x + b = 0 where "." denotes inner product, b is bias from the origin in the input space, w is weight determining the orientation and x are data points taking place in the hyperplane or normal to it. "+" data points are placed in terms of the formula w . x + b ≥ 0 while "-" instances are placed by applying w . x + b < 0 so that separation makes the classification process an easy task.

Even though SVM was originally developed for binary classification, it is now applied in both binary and multi-class data classification problems providing an acceptable prediction ability.

### 2.2.1 Non-separable case

Majority of the problems are not as simple as the given scenario in Figure 2.2 because data points cannot be convenient to be separated by a linear hyperplane due to non-linear clusterings found in data. For those cases, we enhance kernel functions to transform the non-linear data into a feature space so that linear classification can be applied. The choice of a kernel depends on the problem at hand because it depends on what we are trying to model. In the literature, there are many popular kinds of kernels defined to apply in complex machine learning problems such as polynomial kernel for modeling feature conjunctions up to the order of the polynomial and radial basis function kernel where circles (or hyperspheres) are picked out.

Lee, Yeh and Pao (2012) introduce SVMs for this kind of problems by making use of support vectors in discriminating between complex data patterns by generating a highly nonlinear separating hyperplane, that is implicitly defined by a nonlinear kernel map.

For training data $x_i \in R^n$ , i = {1, ..., m} with class labels $y \in R^l$ such that $y_i$ = {-1, 1} C-SVC can be formulated as an optimization problem given in equation 2.8.

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i$$
$$\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, m \tag{2.8}$$

where w is the normal vector to the bounding planes ($x^T w + b = -1$ for class "-" and $x^T w + b = 1$ for class "+" according to Figure 2.3), b shows their position relative to the origin. ξ is a slack variable for soft margins which is defined for linearly non-separable cases ( $w^T x_i + b + \xi_i \geq +1$ for class "+" and $w^T x_i + b + \xi_i \leq -1$ for class "+") and 1-norm of ξ, $\sum_{i=1}^{l} \xi_i$ is called the penalty term. Due to the higher complexity of the separating hyperplane overfitting situation can occur by leading to poor generalization. In this direction, C>0 is used as a regularization parameter balancing the weights of the penalty term $\sum_{i=1}^{m} \xi_i$ versus the margin maximization term $\frac{1}{2}\|w\|_2^2$.

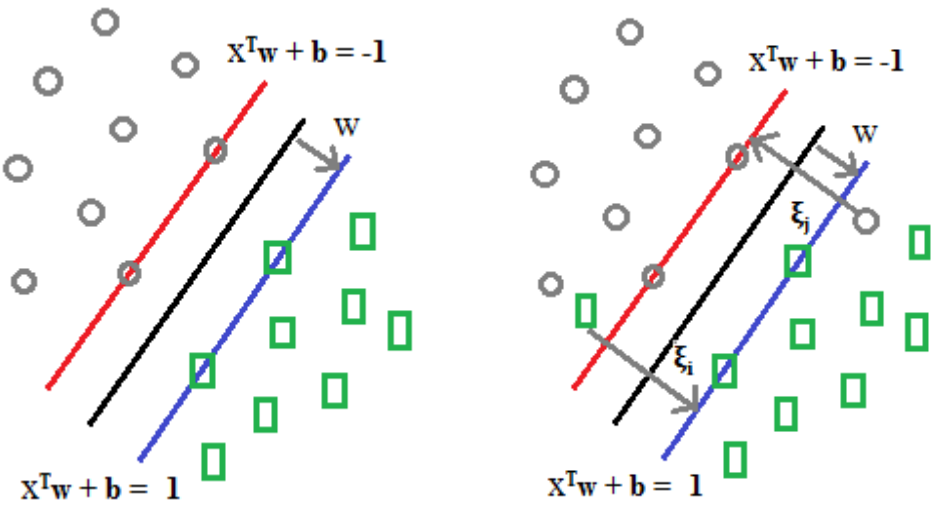


**Figure 2.3 :** An example of linearly separable (left) and non-separable (right) SVM

LibSVM (Chang and Chih-Jen, 2011) is a C/C++ based package for easily implementing support vector machines in other languages/softwares such as Matlab, Python, Java and Octave. In order to deal with binary class and multi class problems, the library provides SVC type of SVM. Except from SVC, SVR (Support vector regression) is available for regression problems and one-class SVM is also present in the package. In this study, one of the SVM types, C-SVC is used to classify both binary and multi-class datasets.

For binary classification in C-Support Vector Classification, a solution is found to the following primal optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i$$
$$\text{s.t. } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, m \tag{2.9}$$

where $\phi(x_i)$ is a function which maps data point $x_i$ into a higher dimensional space, and C, b, w and $\xi$ parameters are same as the previous ones. Because of the probable high dimensionality of the vector w, it can be formulized by a dual problem:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$\text{s.t. } y^T \alpha = 0, 0 \leq \alpha_i \leq C, i = 1, \ldots, m \tag{2.10}$$

where e is a vector which is full of ones, Q represents a m by m positive semidefinite matrix, and $Q_{ij} \equiv y_i y_j K(x_i,x_j)$ where $K(x_i,x_j)$ is the kernel function as $K(x_i,x_j) \equiv \phi(x_i)^T \phi(x_i)$. Once the problem is solved, then the optimal w satisfies

$$w = \sum_{i=1}^{m} y_i \alpha_i \, \phi(x_i) \tag{2.11}$$

and the decision function is

$$\operatorname{sgn}(w^T \phi(x) + b) = \operatorname{sgn}(\sum_{i=1}^{m} y_i \alpha_i \, K(xi, x) + b) \tag{2.12}$$

In this study, radial basis function is selected as the kernel function $K(x, x')$ for two samples x and x'. It is defined as in equation 2.13:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\alpha^2}\right) \tag{2.13}$$

where we can define a parameter $\gamma = \frac{1}{2\alpha^2}$ and result in $K(x, x') = \exp(-\gamma\|x - x'\|^2)$.

### 2.3 Ensemble Learning

Ensemble learning is a paradigm where multiple learners are trained to solve a machine learning problem and a final decision is made after combining each output of single learners according to some criteria. As No Free Lunch theorem states that there is no single model that works best for every problem the aim of the ensemble learning is to boost the accuracy of the single classifiers. Besides due to the possible noise in the data, overlapping data distributions and outliers generally single classifiers cannot achieve a certain classification accuracy. These have grown the needs to create ensemble techniques.

Establishing an ensemble model is made up of two stages. In the first part, a couple of base classifiers are generated in a parallel or sequential manner. Generally, in the sequential manner the construction of a base classifier may affect the construction of the subsequent classifiers. In the latter part, the resulting classifier outputs are combined to take a decision about the final classification of a new test instance. At this stage some type of combination schemes are applied such as majority voting for a classification problem. In the majority voting the class label is selected from the

majority of the individual classifiers' class labels. In regression weighted averaging of the base regressors' outputs gives the final prediction result. The basic framework of the ensemble modeling is depicted in Figure 2.4.

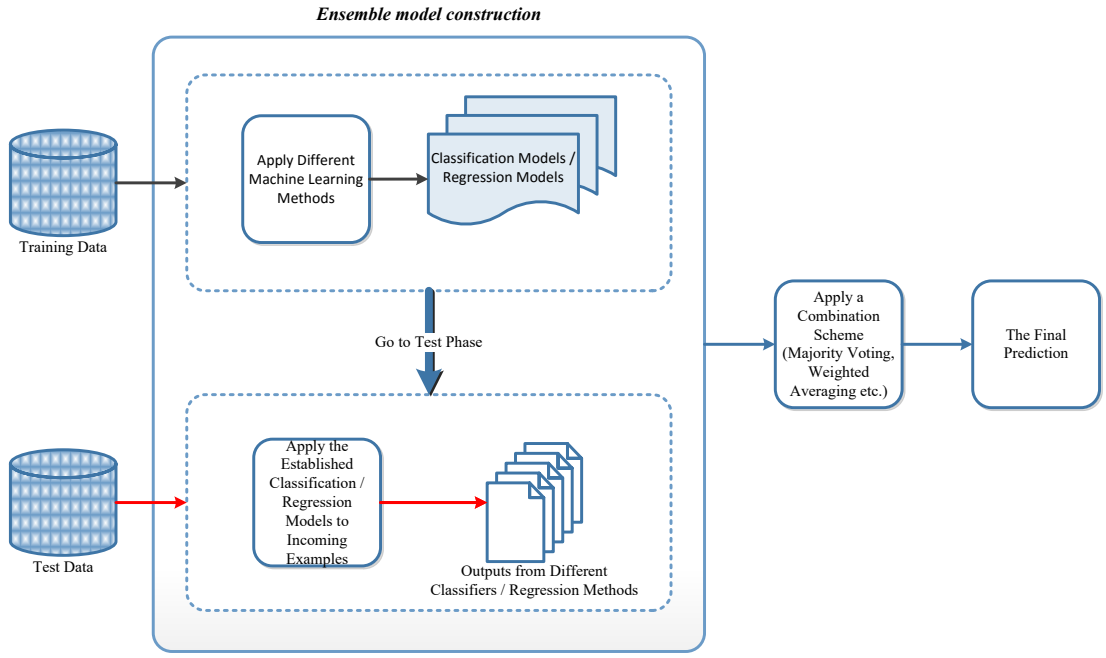


**Figure 2.4 :** Ensemble learning framework.

Model ensembles are among the highly effective techniques in machine learning and pattern recognition applications that generally outperforms other methods. Ensemble learning has already been applied in a variety of domains related to machine learning problems such as text categorization (Dong and Han, 2004), optical character recognition (Chellapilla et al, 2006), face recognition (Lu et al, 2006) and gene expression analysis (Tan and Gilbert, 2006) etc for searching a hypothesis space to reach the most accurate hypothesis by reducing the total error.

It is mostly preferred to classical single learning models because of three significant reasons. The first one is that there may be insufficient information in order to decide on which classifier performs better on the training data. A solution to this problem is therefore combining the ones which results in sufficiently well and it is a reasonable choice to apply. Another rationale is that the applied learning algorithm might practise imperfect search processes which ends up with sub-optimal hypotheses even if there exists a unique best hypothesis. The last reason is the case where ensemble strategy leads to a good approximation as one learner cannot reach a true target

function in the searching phase. On the other hand, choosing ensemble models comes at the cost of raised algorithmic and model complexity. Ensemble strategies can be obtained on either feature space, instance space or classifiers' parameter space. Next we will detail one of the most successful ensemble learning on feature space namely random subspace methods.

### 2.3.1 Random subspace ensemble learning

Searching for a feature base that leads to a considerable classification performance is another challenging task to cope with. Even if there is a single input representation, by selecting random subsets from it we can train different classifiers on selected subspace of features, which is called the random subspace method (Ho, 1998).

Random Subspace Ensemble Learning "RS", also known as Attribute Bagging, is one of the most commonly used ensemble learning methods which plays an important role in finding the subset of informative features to correctly classify given signals. It is a wrapper method that can be used with any learning algorithm. The method is applied by classifying test instances with a chosen classifier along with randomly selected subsets of all possible features iteratively and with replacement. The feature base is changed in each iteration with the same number of randomly permuted features. If we name the whole constructed feature subspace as $X_{rs}$, and the selected feature subspace at $i^{th}$ iteration as $X_{rs\_i}$, $K$ number of subspaces are created at the end of the process in $K$ iterative ensemble learning steps where $X_{rs} = \{X_{rs\_1}, ..., X_{rs\_K}\}$. Table 2.1 shows the pseudo code of the random subspace ensemble learning method.

**Table 2.1 :** The pseudo code of random subspace ensemble learning.

| Algorithm: Random subspace ensemble learning method |
|---|
| *Input*: training set X, number of features in the subspace s, number of ensemble predictors $K$, predictor h<br>*Output*: ensemble model $h = \{h_1, ..., h_K\}$ combination of whose outputs is used to predict new test instances' class labels/regression results<br>*For* i = 1:$K$<br>Create a subspace sample data $X_{rs\_i}$ with s features selected at random with replacement from X<br>Apply the predictor $h_i$ to $X_{rs\_i}$<br>*End For*<br>Return $h = \{h_1, ..., h_K\}$ ensemble model |

Note that by applying random feature subspaces, different predictors will deal with the same problem from different standpoints by resulting in more robust representation and diminishing the curse of dimensionality arising from high dimensional inputs.

### 2.3.2 Bagging

Bagging short for "bootstrap aggregating" is an ensemble learning approach which generates multiple exemplars of a predictor to lead to an aggregated learner by taking the combination of their outputs using a fixed rule. It provides a way to present variability between the different models within a committee. Creation of the multiple exemplars is done via making bootstrap replicates of the learning set.

Logic behind the bootstrap creation is treated as follows. Assume that we have a dataset $X = \{x_1, ..., x_m\}$ with m data points. If we generate a new dataset $X_{Bagged}$ whose instances are randomly drawn from the original dataset as the same number of instances with replacement, it is the case where some number of data points are repeated containing duplicates in $X_{Bagged}$ and some others in the original dataset are not included now. A particular instance which is chosen for a bootstrap sample of size m can be calculated as follows:

$$\text{probability of selection} = 1 - (1 - 1/\mathrm{m})^{\mathrm{m}} \tag{2.14}$$

which is about two-third and has limit 1-1/e = 0.632 for $\mathrm{m} \rightarrow \infty$ (Flach, 2012, Chapter 11). This means that each bootstrap sample is likely to leave out about a third of the data points. This difference between bootstrap models is exactly what we want to give rise to diversity among the models in the ensemble. An iterative process is performed by repeating this procedure *K* times and resulting in *K* randomly generated datasets. Table 2.2 displays the pseudo code of the general framework of bagging algorithm.

## 2.4 Active Learning

In passive learning, a bunch of training examples and their class labels are provided to the learning algorithm. In many machine learning problems, we need to cope with significant number of unlabelled examples whose labeling is mostly time-consuming and expensive to obtain. Active learning is a framework where abundant unlabeled data and few labeled samples are available.

**Table 2.2 :** The pseudo code of bagging.

| Algorithm: Bagging |
|---|
| *Input*: training set X, number of instances m, number of ensemble predictors *K*, predictor h<br>*Output*: ensemble models h = $\{h_1, ..., h_K\}$ combination of whose outputs is used to predict new test instances' class labels/regression results<br>*For* i = 1:*K*<br>Create a bootstrap sample data $X_{bagging\_i}$ with selecting n data points randomly with replacement from X<br>Apply the predictor $h_i$ to $X_{bagging\_i}$<br>*End For*<br>Return h = $\{h_1, ..., h_K\}$ ensemble model |

The framework resolves the labeling problem by asking the labels of some intelligently selected examples to a trained expert or an oracle. The selected data points are usually the optimal ones which boost the number of correctly classified instances upon they are labeled and incorporated into the training phase.

Active learning scenario has been applied in various machine learning real-world applications such as image classification and retrieval (Zhang and Chen, 2002), text classification (Tong and Koller, 2001), email filtering, web searching, video classification and retrieval, information extraction and speech recognition etc.

The fundamental purpose of active learning is to improve the accuracy of the initial classifier by adding new training examples from unlabelled dataset which are selected using a selection criterion i.e. informativeness measure. The learner may start the classification task with few number of labelled training data, then in each iteration one or more unlabeled instances are carefully chosen to be added to the training examples after its class label is determined by an oracle. This process is implemented iteratively by the selection of the most informative unlabeled data points which will help improve the prediction ability. Figure 2.5 points out the generalized active learning framework.

In the matter of applying active learner, sources of unlabelled instances play an important role. In literature, there are three main categories of sources of unlabelled data (Roederer, 2012; Settles, 2010):

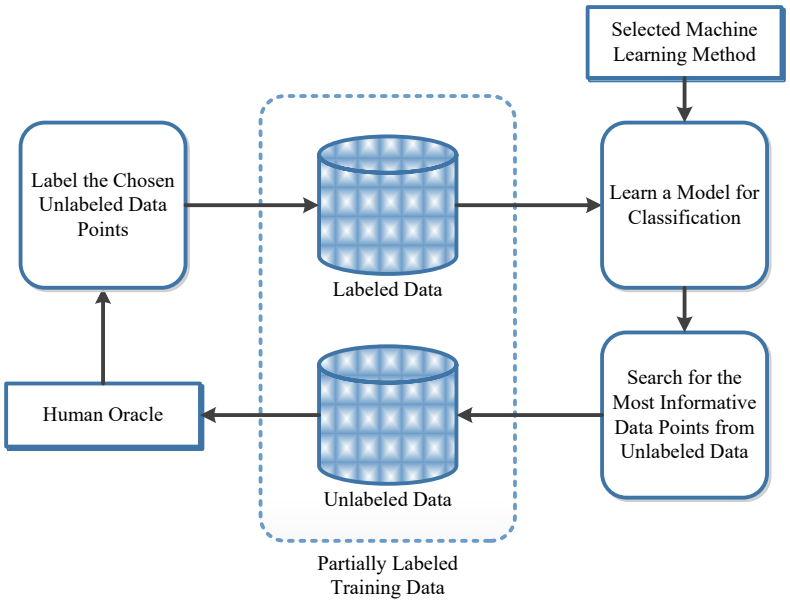


**Figure 2.5 :** Active learning framework.

- *query synthesis* in which the learner asks the labels of unlabelled data points from some underlying distribution also including the queries that the learner produces de novo without the need for a distribution.
- *stream-based sampling* where unlabelled instances are handled one by one through sampling from an actual distribution to decide whether they should be integrated into the set of labelled instances or not. It is suitable for special situation where the memory and storage capacity is limited.
- *pool-based sampling* which is the mostly applied sampling scenario that the learner chooses instances from a pool of unlabelled data to query by using a greedy approach through examining informativeness of each.

In pool-based sampling, a significant question arises in the case of how to select and assess the most informative data point among the unlabeled examples. Perhaps the simplest way is to query instances where the learner is least certain and this process is known as uncertainty sampling. One of the most popular uncertainty sampling techniques is based on entropy. We want to choose the examples which leads to the greatest reduction in entropy upon its class label is known. Calculating the entropy over the distribution of possible class labels results in a value that represents the amount of information needed. "The more entropy in the distribution, the more uncertain the choice of class label for that data value, and the more informative that

query would be" (Roederer, 2012). A generalized entropy-based query sampling strategy can be defined as in equation 2.15:

$$x_h^* = \text{argmax}_x - \sum_{i=1}^{c} P_\theta(y = i|x) \log P_\theta(y = i|x) \quad \text{where } i = \{1, ..., c\} \tag{2.15}$$

where x refers to any instance, y is the class of the instance x, c is the number of classes and θ is the parameters in the classifier model h.

According to Holub et al. (2008), "Active learning adaptively prioritizes the order in which the training examples are acquired which can significantly reduce the overall number of training examples required to reach near-optimal performance" (p. 1).

# 3. THE PROPOSED METHOD

In this study, one of the supervised classification algorithms, dictionary learning, is applied in combination with active learning framework using the strength of ensemble classifier strategies. In the first step, Bagging and Random Subspace ensemble methods are performed by using dictionary learning as the base classifier and in the latter part, Active Learning is applied by showing the effect of using the most informative unlabeled instances while modeling dictionary base.

## 3.1 Dictionary Ensembles Using Random Subspaces and Bagging

Bagging and Random Subspace methods are the most commonly used ensemble learning methods which play an important role in finding the subset of instances and features to obtain diverse classifiers given data samples. Following the idea behind dictionary learning, ensemble learning methods can be merged into the process to boost the correctly classified number of instances. Accordingly, in this study dictionary learning is used as a base classifier with Random Subspace and Bagging methods. Random Subspace Dictionary Learning (RDL) and Bagging Dictionary Learning (BDL) algorithms creates $K$ dictionaries for each class in the training set using randomly selected features for RDL and instances for BDL. Class label for a test instance is determined by picking up the majority class label among the results of all $K$ dictionaries. The pseudo-code of the algorithms is given in Table 3.1.

The framework of the proposed dictionary learning algorithms is shown in Figure 3.1. Initially, according to the choice of ensemble learning strategies, either BDL or RDL, s instance/feature subspaces are generated. In the dictionary construction phase, each subspace produces a dictionary base. Test data are classified using the ensemble dictionary learning classifiers after dictionary construction phase. As a result, ensemble strategy produces a final prediction by applying majority voting on class labels obtained in each instance/feature subspace.

**Table 3.1 :** The pseudocode of the proposed ensemble methods.

Algorithm: Ensemble dictionary learning

*Input*: training set $X \in R^{mxn}$, training class labels $Y \in R^{m}$, number of features n, number of selected feature/instances s, number of ensemble dictionaries *K*, number of classes c, test set $X' \in R^{wxn}$, number of training instances m, number of test instances w

*Output*: predicted class labels Y' for test instances

```
Training:
    For i = 1:K
        switch(ensemble_algorithm)
        case RDL:
            Select s random features from X
            Create X_r ∈ R^mxs using selected features
        case BDL:
            Select s random instances from X
            Create X_r ∈ R^sxn using selected instances
        For each class in the training set X_r:
            Train dictionaries {D_1, D_2, ..., D_c}
            D^i = {D_1, D_2, ..., D_c}
    End For
Testing: Input a test instance vector x' from test set X'
        For i = 1:K
            Calculate δ_i(x′) using (2.7) and D^i
        End For
        Classify x' using majority voting on δ_i(x′)'s
```

### 3.2 Active Learning Based Data Classification Using Dictionary Ensembles

The proposed active learning scenario is applied in combination with Random Subspace Dictionary Learning and Bagging Dictionary Learning methods as the base classifiers. Initially, 20% of the whole dataset is used as training data to construct a supervised dictionary model.

To form each random subspace ensemble, feature subspace is iteratively reduced by randomly chosen attributes with replacement. Using the dictionary model, unlabeled instances are classified into appropriate classes. At each iteration in order to select the instances to be queried, entropy query strategy is employed by computing the class label entropies using equation 2.15. Unlabeled instances are sorted in a descending order based on their entropies. At each iteration, the data samples that have the highest entropy values are asked to the oracle and 10% of the unlabelled instances are added to the training data. Figure 3.2 displays the scenario schematically.

Performing active learning ensures choosing the most informative instances to training set, by doing so, atoms can be updated or new atoms can be added to the dictionaries which may improve the sparse representations of the instances.

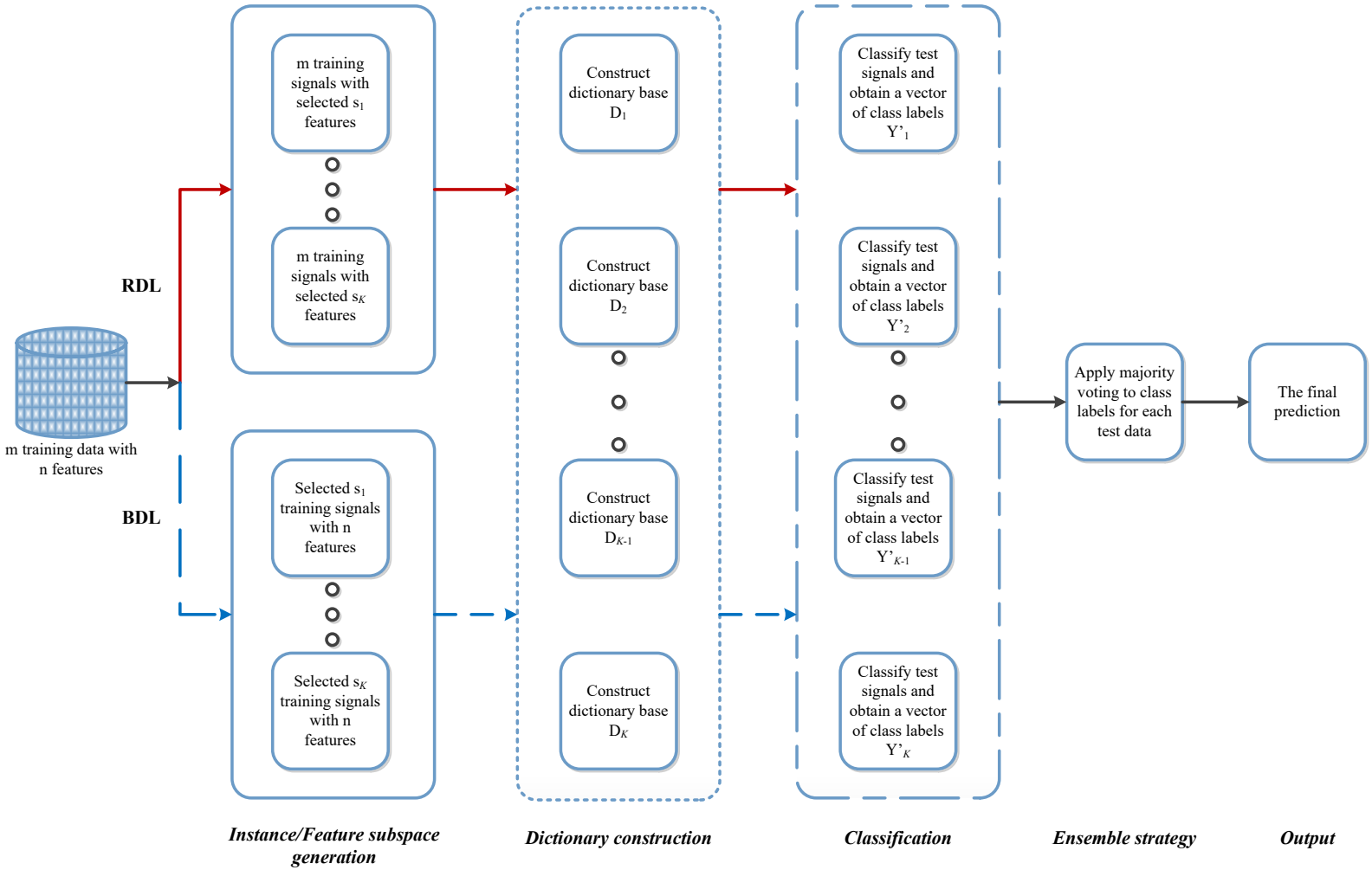


**Figure 3.1 :** Framework of the proposed ensemble dictionary learning models.

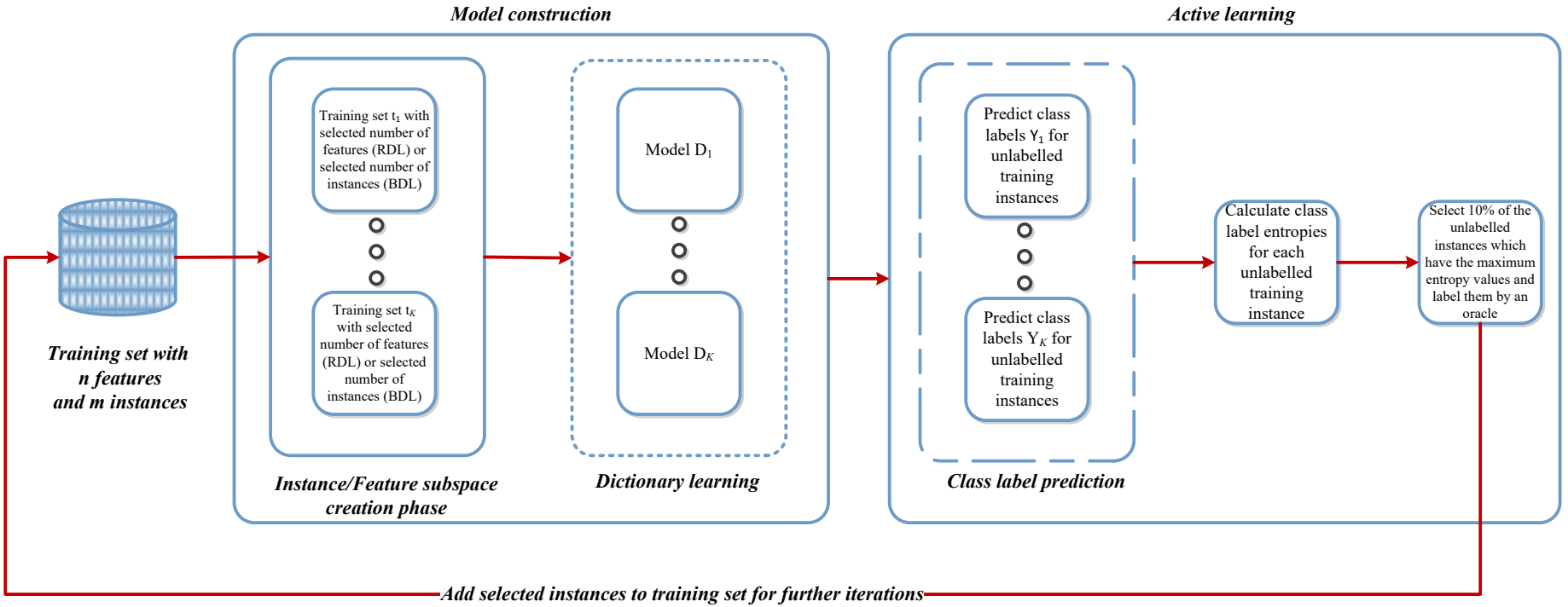


**Figure 3.2 :** Active learning framework using dictionary ensembles.

## 4. MATERIALS AND EXPERIMENTAL SETUP

In order to measure the classification performance of the proposed method, a number of datasets retrieved from UCI Machine Learning Repository (Lichman, 2013) and OpenML (Vanschoren et al, 2013) are used. Table 4.1 indicates these datasets by noting the respective number of instances, feature size including the class attribute and how many classes they cover.

**Table 4.1 :** Properties of the datasets used in the experimental results.

| Dataset | The number of instances | The number of attributes | The number of classes |
|---|---|---|---|
| cmc | 1473 | 10 | 3 |
| fri_c4_100_10 | 100 | 11 | 2 |
| ionosphere | 351 | 35 | 2 |
| pollution | 60 | 16 | 2 |
| sonar | 208 | 61 | 2 |
| spectf_train | 80 | 45 | 2 |
| statlog-german | 1000 | 25 | 2 |
| vehicle | 846 | 19 | 4 |
| waveform-5000 | 5000 | 40 | 3 |
| mfeat-karhunen | 2000 | 65 | 10 |
| optdigits | 5620 | 64 | 10 |

Datasets have been gathered from various areas. Contraceptive Method Choice, cmc, dataset holds the information of demographic and socio-economic characteristics (age, education, religion, etc.) of Indonesian married women who were not pregnant during the survey. The dataset was a part of 1987 National Indonesia Contraceptive Prevalence Survey that was to predict which contraceptive method (no-use, long-term, short-term) was chosen.

Mfeat-karhunen is among the other six subgroups of multiple features dataset which includes features of handwritten numerals 0 to 9 from a collection of Dutch utility maps which have been digitized in binary images. This subgroup is the combination of 64 Karhunen-love coefficients.

Optdigits is a dataset comprised of extracted 32x32 normalized bitmap images of printed handwritten digits, 0 to 9, from 43 people. Preprocessing step was done by NIST preprocessing tools to extract features.

Ionosphere dataset deals with the radar returns from ionosphere layer by grouping them as good, which says there is some type of structure in the ionosphere or bad

returns in which the signals just pass through the ionosphere without witnessing any structure.

Sonar dataset developed by R. Paul Gorman and Terry Sejnowski is the combination of patterns gathered from metal cylinder and rock. The objective is to classify each record into one of two classes, mine (metal cylinder) or a rock.

Spectf dataset is made up of extracted image features which are used to decide whether a patient has the signs of "normal" or "abnormal" diagnose by looking at his/her cardiac Single Proton Emission Computed Tomography (SPECT) images.

Statlog-german is one of the datasets under the database hold for European Statlog project. German credit dataset contains the attributes such as salary information, credit history, present employment etc. in order to decide whether a person has the risk of good or bad credit.

Vehicle dataset is used to distinguish between car models and characterize a given vehicle silhouette as one of four types: a double decker bus, Cheverolet van, Saab 9000 and an Opel Manta 400. Dataset features were extracted from various vehicle silhouettes which were viewed from different angles.

Waveform-5000 dataset is for the classification of three different wave classes each of which is formed using the combination of two of three base waves. Each instance is generated with added noise (mean 0, variance 1) in each attribute.

Fri_c4_100_10 is one of the datasets in the collection of 80 datasets, donated by M. Fatih Amasyalı, which were artificially produced by the Friedman function.

In order to make the experiments with SVM and SVM ensembles, Matlab platform is integrated with LibSVM library. For dictionary learning part, SPAMS (SPArse Modeling Software) toolbox is utilized (Mairal et al, 2009, 2010).

# 5. EXPERIMENTAL RESULTS

## 5.1 Performance Analysis Based on Classification Accuracies

Experimental results are obtained using tenfold cross validation for DL, RDL, BDL, SVM, RSVM and BSVM. The number of ensemble classifiers, *K*, is selected as 15 for all ensemble methods. 70% of the features are randomly selected to construct each subspace for RDL and RSVM. Similarly, for BDL and BSVM 75% of the instances are selected randomly. 10% of the instance sizes are selected as the number of atoms for initial dictionaries. The penalty parameter, $\lambda$, to constitute dictionary models is tuned to 0.05. For SVM, RSVM and BSVM models, the applied type of SVM is C-SVC in which the kernel function is radial basis function. Table 5.1 shows the optimal values for C and $\gamma$ parameters determined by grid search with respect to each dataset.

**Table 5.1 :** SVM parameters for each dataset.

| Dataset | C | $\gamma$ |
|---|---|---|
| cmc | 3 | 0.01 |
| fri_c4_100_10 | 18 | 0.02 |
| ionosphere | 19 | 0.04 |
| sonar | 4 | 0.09 |
| spectf_train | 1 | 0.01 |
| statlog-german | 1 | 0.01 |
| vehicle | 1 | 0.01 |
| waveform-5000 | 1 | 0.01 |
| mfeat-karhunen | 1 | 0.01 |
| optdigits | 1 | 0.01 |
| pollution | 1 | 0.07 |

Classification accuracies of the DL, SVM, RDL, BDL, RSVM and BSVM methods are given in Table 5.2. The best classification accuracy for each dataset is indicated by bold typing. In the experiments, BDL outperforms other methods in 6 out of 11 datasets. Note that selecting instance subspaces for dictionary learning model increases the number of correctly classified instances. On the other side, SVM follows BDL's classification performance by resulting in the best performance in 3 datasets. Each of RDL and DL algorithms produces good results in 2 datasets. Here, DL shows similar performance with RDL i.e. selecting feature subspaces does not

contribute much to the performance of default model. According to the results, BSVM cannot manage pretty good results compared to other algorithms and RSVM is the best classifier at only one dataset. It means for the given datasets applying both feature/instance subspaces of SVM do not lead improved predictions.

**Table 5.2 :** Classification accuracies of the classifiers DL and SVM along with their ensembles.

| Dataset | DL | RDL | BDL | SVM | RSVM | BSVM |
|---|---|---|---|---|---|---|
| cmc | 50.40 | 52.65 | 51.83 | **55.40** | 54.72 | 54.72 |
| fri_c4_100_10 | 59.00 | 60.00 | **64.00** | 60.42 | 58.51 | 60.42 |
| ionosphere | 92.85 | 93.42 | 92.00 | **95.16** | 94.58 | 94.58 |
| sonar | 85.71 | **87.61** | **87.61** | 83.64 | 82.21 | 83.66 |
| spectf_train | **71.25** | **71.25** | **71.25** | 58.75 | 61.25 | 50.00 |
| statlog-german | 70.30 | 72.10 | **76.20** | 73.20 | 71.10 | 72.70 |
| vehicle | 74.11 | 76.58 | **77.52** | 50.24 | 60.68 | 47.42 |
| waveform-5000 | 76.18 | 80.84 | 78.54 | **86.84** | 86.50 | 86.73 |
| mfeat-karhunen | 97.30 | 97.35 | 97.05 | 97.75 | **97.85** | 97.60 |
| optdigits | **99.07** | 99.00 | 99.05 | 86.29 | 95.96 | 86.03 |
| pollution | 71.66 | 73.33 | **78.33** | 51.71 | 51.71 | 51.71 |

In the second part, active learning has been performed on ensembles of both dictionary learning (ARDL/ABDL) and support vector machines (ARSVM/ABSVM). The number of ensemble classifiers is selected as 5 for ARDL, ABDL, ARSVM and ABSVM. Ten-fold cross validation results for ARDL and ARSVM are shown in Table 5.3 and Table 5.4 respectively. Table 5.5 and Table 5.6 correspond to the experimental results for ABDL and ABSVM. The "default" results are obtained without using active learning via taking the 20% percentage of the whole training data as the new training instances to construct the classification model. Others show the results using active learning in which the training data size (initially 20% of the whole training data) is increased with the addition of the mentioned percentage (10%) of the unlabeled data in each iteration.

70% of the features are randomly selected to construct each subspace in the generation of ARDL and ARSVM classifiers. The number of atoms for initial dictionaries for ARDL and ABDL are defined as 10% of the instance sizes. ARSVM applies radial basis function of C-SVC as the kernel type in which gamma and cost parameters are the same as the ones in Table 5.1 for each dataset. Established results show that ARDL outperforms ARSVM in 7 out of 11 of the datasets and, in majority, accuracy is improved with small fluctuations if the training size is increased using an active learner.

**Table 5.3 :** Active learning classification results based on dictionary learning using random subspace ensemble.

| Dataset | Default | Iter1 | Iter2 | Iter3 | Iter4 | Iter5 | Iter6 |
|---|---|---|---|---|---|---|---|
| cmc | 48.43 | 49.31 | **51.02** | 50.40 | 51.15 | 49.93 | 50.47 |
| fri_c4_100_10 | 56.00 | 57.00 | 57.00 | 60.00 | **69.00** | 56.00 | 59.00 |
| ionosphere | 91.14 | 91.42 | 92.28 | 91.71 | 91.14 | **92.85** | **92.85** |
| sonar | 72.85 | 79.04 | 80.95 | 82.85 | 84.28 | 86.19 | **87.14** |
| spectf_train | 60.00 | 57.50 | 63.75 | 60.00 | 63.75 | 65.00 | **66.25** |
| statlog-german | 74.00 | 74.80 | **76.80** | 75.70 | 76.30 | 76.40 | 75.30 |
| vehicle | 68.58 | 72.00 | 70.58 | 73.41 | 73.88 | 74.70 | **76.00** |
| waveform-5000 | 74.74 | 72.20 | 77.46 | 77.16 | 79.52 | **80.88** | 80.74 |
| pollution | 63.33 | 70.00 | 73.33 | 71.66 | **76.66** | 75.00 | 73.33 |
| mfeat-karhunen | 15.30 | 39.95 | 77.60 | 92.30 | 89.40 | 92.90 | **95.00** |
| optdigits | 10.30 | 9.92 | 14.19 | 43.23 | 83.07 | 96.65 | **97.97** |

**Table 5.4 :** Active learning classification results based on support vector machine using random subspace ensemble.

| Dataset | Default | Iter1 | Iter2 | Iter3 | Iter4 | Iter5 | Iter6 |
|---|---|---|---|---|---|---|---|
| cmc | 52.04 | 52.10 | 53.46 | 53.94 | **54.28** | 54.08 | 53.87 |
| fri_c4_100_10 | 51.00 | 50.00 | 56.00 | 58.00 | **60.00** | 58.00 | 59.00 |
| ionosphere | 90.00 | 93.14 | 94.57 | 94.57 | 94.57 | 95.14 | **95.42** |
| sonar | 69.52 | 75.71 | **78.09** | 75.71 | 74.76 | 77.14 | 77.61 |
| spectf_train | 47.50 | **55.00** | 52.50 | **55.00** | **55.00** | **55.00** | **55.00** |
| statlog-german | 72.00 | 72.60 | 73.00 | 73.40 | 74.20 | 74.10 | **74.30** |
| vehicle | 42.23 | 40.58 | 40.35 | 43.88 | 44.00 | 46.47 | **47.64** |
| waveform-5000 | 85.46 | 85.48 | 85.56 | 86.06 | 86.22 | 86.10 | **86.58** |
| pollution | **50.00** | **50.00** | **50.00** | **50.00** | 46.66 | 46.66 | 46.66 |
| mfeat-karhunen | 94.05 | 93.10 | **96.20** | 95.95 | 96.00 | 96.00 | 95.90 |
| optdigits | 85.71 | 89.83 | 91.93 | 93.11 | 93.73 | 93.62 | **93.98** |

**Table 5.5 :** Active learning classification results based on dictionary learning using bagging ensemble.

| Dataset | Default | Iter1 | Iter2 | Iter3 | Iter4 | Iter5 | Iter6 |
|---|---|---|---|---|---|---|---|
| cmc | 45.71 | 47.41 | 47.75 | 49.86 | 50.00 | 50.54 | **50.74** |
| fri_c4_100_10 | 47.00 | 49.00 | 55.00 | 55.00 | 55.00 | **61.00** | 57.00 |
| ionosphere | 83.14 | 92.28 | **93.42** | 92.28 | 92.28 | 92.00 | 92.00 |
| sonar | 71.90 | 79.04 | 76.66 | 80.47 | 84.28 | **86.66** | 86.19 |
| spectf_train | **66.25** | 57.50 | 57.50 | 60.00 | 63.75 | 60.00 | 63.75 |
| statlog-german | 73.40 | 76.00 | 77.00 | **77.20** | 77.00 | 76.10 | 75.80 |
| vehicle | 29.17 | 64.70 | 73.64 | **74.47** | 73.05 | 72.47 | 73.88 |
| waveform-5000 | 72.24 | 68.86 | 64.40 | 69.30 | 76.92 | 78.96 | **79.18** |
| pollution | 56.66 | 61.66 | 68.33 | 71.66 | 71.66 | 73.33 | **78.33** |
| mfeat-karhunen | 9.35 | 9.35 | 19.30 | 24.45 | 29.95 | 33.95 | **41.20** |
| optdigits | 10.30 | 10.30 | 10.30 | 10.19 | 11.97 | 26.21 | **58.16** |

**Table 5.6 :** Active learning classification results based on support vector machine using bagging ensemble.

| Dataset | Default | Iter1 | Iter2 | Iter3 | Iter4 | Iter5 | Iter6 |
|---|---|---|---|---|---|---|---|
| cmc | 52.58 | 52.51 | 51.76 | 53.06 | 52.78 | **53.26** | 52.72 |
| fri_c4_100_10 | 54.00 | 59.00 | **62.00** | 60.00 | 61.00 | 61.00 | **62.00** |
| ionosphere | 89.14 | 93.14 | 95.42 | 95.42 | **96.00** | 94.85 | 95.71 |
| sonar | 69.52 | 70.47 | 75.23 | 79.04 | 79.04 | 80.00 | **82.38** |
| spectf_train | 52.50 | 50.00 | **55.00** | **55.00** | **55.00** | **55.00** | **55.00** |
| statlog-german | 71.50 | 72.40 | 72.30 | 72.20 | 73.30 | 73.40 | **73.60** |
| vehicle | 33.17 | 33.64 | 39.41 | 40.23 | 40.23 | 43.17 | **45.05** |
| waveform-5000 | 84.82 | 85.60 | 85.82 | 86.44 | **86.64** | 86.60 | 86.32 |
| pollution | 50.00 | **55.00** | 50.00 | 50.00 | 46.66 | 46.66 | **55.00** |
| mfeat-karhunen | 90.20 | 77.70 | 89.90 | **94.85** | 93.25 | 93.50 | 93.60 |
| optdigits | 59.00 | 71.03 | 76.88 | 81.28 | 81.79 | 83.46 | **84.34** |

In order to learn the effect of applying bagging ensemble to the dictionary learning and support vector machines models under active learning framework, new experiments have also been carried out. In the same way as its random subspace counterpart, the classification accuracies for both of ABDL and ABSVM are enhanced by the selection of informative examples to training set in each iteration. According to the best results obtained for each dataset, ABDL is better in 5 out of 11 of the given datasets as a consequence. For dictionary learning model, selecting feature subspaces instead of instance subspaces is more rational under active learning framework as a result of experiments.

Considering the final iteration accuracies in Table 5.7, ARDL is the best classifier in 4 out of 11 datasets while ARSVM is good at 3 of them. Each of ABDL and ABSVM models performs the optimal accuracies with 2 datasets. In terms of the accuracies obtained by random subspace ensembles under active learning, ARDL and ARSVM give the best results in 6 and 4 out of 11 datasets respectively and for one dataset they lead to the same performance. For bagging ensembles of the applied models, ABSVM provides the best accuracy in 6 out of 11 datasets and it outperforms ABDL model.

### 5.2 Friedman Test

Each model used for a machine learning problem produces a new solution and the main purpose is to find the successful one. To determine the quality of each predictor, classification accuracy is generally used as a measurement technique. In

addition to accuracy results, we should statistically verify the performance improvement produced by the models using a hypothesis test.

**Table 5.7 :** The last iteration accuracies of active learning methods.

| Dataset | ARDL | ARSVM | ABDL | ABSVM |
|---|---|---|---|---|
| cmc | 50.47 | **53.87** | 50.74 | 52.72 |
| fri_c4_100_10 | 59.00 | 59.00 | 57.00 | **62.00** |
| ionosphere | 92.85 | 95.42 | 92.00 | **95.71** |
| sonar | **87.14** | 77.61 | 86.19 | 82.38 |
| spectf_train | **66.25** | 55.00 | 63.75 | 55.00 |
| statlog-german | 75.30 | 74.30 | **75.80** | 73.60 |
| vehicle | **76.00** | 47.64 | 73.88 | 45.05 |
| waveform-5000 | 80.74 | **86.58** | 79.18 | 86.32 |
| pollution | 73.33 | 46.66 | **78.33** | 55.00 |
| mfeat-karhunen | 95.00 | **95.90** | 41.20 | 93.60 |
| optdigits | **97.97** | 93.98 | 58.16 | 84.34 |

In the matter of comparison of c different classifier models on r different datasets, when so many pairwise tests are made, a certain proportion of the null hypotheses can be rejected due to random chance. In order to detect differences in treatments across multiple classifier models, one of the non-parametric statistical tests, Friedman test which is based on ranked rather than absolute performance has been conducted (Demšar, 2006). A null hypothesis, $H_{0,}$ is provided in which all of the applied classifiers are equivalent otherwise the alternative hypothesis, $H_{1,}$ is present that not all classifiers are equivalent.

$H_0$: Classifier models are equivalent.

$H_1$: Not all classifiers are equal.

Initially, each classifier model is rated according to their classification accuracies in each dataset. For each dataset, classifiers are put in order by assigning rank 1 to the classifier with the best classification accuracy and increasing the rank number by one until assigning rank c to the worst one for c applied classifiers (Flach, 2012, Chapter 12). If there is a case of tie, the rank value is assigned as the average rank. In the following step, rank totals per classifiers are calculated. Table 5.8 shows the Friedman test ratings obtained through classification accuracies found for each dataset by the respective classifiers.

The Friedman test statistic is calculated by equation 5.1 where r is the number of datasets, c is the number of classifiers and $R_j$ is the total of the ranks for the classifier j among all datasets.

$$F_R = \frac{12}{r * c * (c+1)} \sum_{j=1}^{c} R_j^2 - 3 * r * (c+1) \quad (5.1)$$

Friedman test statistic $F_R$ approaches chi-square distribution $\chi^2$ with c-1 degrees of freedom when the number of datasets r gets large enough (i.e. r > 10 and c > 5). In order to reach a conclusion about the proposed hypotheses, for a predetermined confidence level of α, the null hypothesis is rejected on condition that the computed value of $F_R$ is greater than the table value of $\chi^2$ in the corresponding significance α and c-1 degrees of freedom.

Reject $H_0$ if $F_R > \chi_\alpha^2$

Otherwise, do not reject $H_0$

Degrees of freedom, d, for 6 classifiers is calculated as d = c-1 = 6-1 = 5. For %95 confidence level (α = 0.05) and d = 5, the table value for $\chi^2$ statistic is 11.07. According to the found rank totals, R1 = 45, R2 = 34.5, R3 = 31.5, R4 = 34.5, R5 = 43 and R6 = 42.5. In order to check the rankings, we can use the formulation in equation 5.2.

$$\sum_{j=1}^{c} R_j = \frac{r * c * (c+1)}{2} \quad (5.2)$$

If we apply equation 5.2 to our scenario,

$$45 + 34.5 + 31.5 + 34.5 + 43 + 42.5 = \frac{11 * 6 * (6+1)}{2}$$

$$231 = 231.$$

Using equation 5.1 we calculate Friedman test statistic,

$$F_R = \frac{12}{11 * 6 * (6+1)} * (45^2 + 34.5^2 + 31.5^2 + 34.5^2 + 43^2 + 42.5^2) - 3 * 11 * (6+1)$$

$F_R = 4.1429 < \chi_{0.05}^2 = 11.07$. We cannot reject $H_0$, i.e. applied models perform equivalently on the datasets. In the next part, Wilcoxon signed rank test is applied to

see pairwise performance differences for detailed comparisons.

Let's apply Friedman test to the accuracy results of active learning scenario for ensembles of dictionary learning and SVM models. Hypotheses are initially determined same as the previous one to prove their equality in terms of their classification performance. Degrees of freedom, d, for 7 classifiers is computed as d = c-1 = 7-1 = 6. For %95 confidence level ($\alpha = 0.05$) and d = 6, the table value for $\chi^2$ statistic is 12.59.

If we look at the test results for ARDL model, according to the found rank totals in Table 5.9, R1 = 72.5, R2 = 65.5, R3 = 42.5, R4 = 44.5, R5 = 32, R6 = 28 and R7 = 23. Now we can use the formulation in equation 5.2 to check the rankings,

$$72.5 + 65.5 + 42.5 + 44.5 + 32 + 28 + 23 = \frac{11 * 7 * (7+1)}{2}$$

$$308 = 308.$$

Using equation 5.1 we compute Friedman test statistic,

$$F_R = \frac{12}{11 * 7 * (7+1)} * (72.5^2 + 65.5^2 + 42.5^2 + 44.5^2 + 32^2 + 28^2 + 23^2) - 3 * 11 * (7+1)$$

$F_R = 41.2597 > \chi^2_{0.05} = 12.59$, so reject $H_0$ in other words all classifiers constructed with additional unlabelled training data in each iteration of active learning framework do not perform equally on the datasets.

Table 5.10 is prepared for Friedman test of ARSVM model's different iterations and after applying the same steps it is found that $F_R = 27.6039 > \chi^2_{0.05} = 12.59$. Therefore, we can conclude in the inequivalency of the applied iterations on the classification performance. The same procedure is also true for ABDL and ABSVM models.

Table 5.11 shows the Friedman test applied to the accuracy values of the last iteration of the active learning models which are displayed in Table 5.7. According to the findings from 4 models, $F_R = 2.2636 < \chi^2_{0.05} = 7.81$, thus we cannot reject that the algorithms perform equivalently. Wilcoxon signed rank test is applied for this case in the next part to query for the equivalency of the methods in a pairwise comparison.

### 5.3 Wilcoxon Signed Rank Test

According to the Friedman test results, there is no acceptable significance among the classification performances of the applied DL, SVM, RDL, RSVM, BDL and BSVM methods. Instead of comparing whole algorithm space, Wilcoxon Signed Rank Test provides pairwise comparison between selected two methods over multiple datasets.

$H_0$: The performance difference between two methods is not significant.

$H_1$: The performance difference between two methods is significant.

The procedure follows the following steps to check the validity of the hypotheses. 1) calculate accuracy differences between two algorithms for each dataset, 2) transform differences into their absolute values, 3) order them in their absolute values by starting numbering with the smallest difference, 4) sum positive and negative ranks separately, 5) find Wilcoxon value, W, from table for the number of datasets at a predefined α, 6) compare W with the smallest among the sums of positive and negative ranks, 7) If W is equal or higher than the calculated value then reject the null hypothesis. In case of zero differences, the comparison is ignored and it is excluded in this way table value for Wilcoxon rank is found according to the reduced dataset size. Besides, if there is a tie among performance differences, the ranks are assigned by taking their average.

In Table 5.12, an example to show how Wilcoxon signed rank test is applied is given among the pairs of DL/RDL and DL/SVM methods. For DL/RDL, total number of positives is 2 and total number of negatives is 53. We choose the smallest sum between positives and negatives so it is 2 for this case. Table value of W is found for α=0.05and 11-1=10 datasets because we ignore zero difference. As a result, $2 < W_{\alpha=0.05,\ 10} = 8$, therefore we can reject null hypothesis in other words performance differences of DL and RDL is significant.

In the same manner, DL/SVM pairs are also tested. Total number of positives and total number of negatives are 41 and 25 respectively, we select the smallest one, 25 to continue our test. Because there is no zero difference, we look at the table value of $W_{\alpha=0.05,\ 11} = 10$. $25 > W_{\alpha=0.05,\ 11} = 10$, therefore we cannot reject $H_0$, it means there is no significant performance difference between DL and SVM algorithms.

**Table 5.8 :** Friedman test rankings of DL and SVM along with their respective ensembles.

| Dataset | DL | Rank | RDL | Rank | BDL | Rank | SVM | Rank | R-SVM | Rank | B-SVM | Rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cmc | 50.40 | 6 | 52.65 | 4 | 51.83 | 5 | 55.40 | 1 | 54.72 | 2.5 | 54.72 | 2.5 |
| fri_c4_100_10 | 59.00 | 5 | 60.00 | 4 | 64.00 | 1 | 60.42 | 2.5 | 58.51 | 6 | 60.42 | 2.5 |
| ionosphere | 92.85 | 5 | 93.42 | 4 | 92.00 | 6 | 95.16 | 1 | 94.58 | 2.5 | 94.58 | 2.5 |
| sonar | 85.71 | 3 | 87.61 | 1.5 | 87.61 | 1.5 | 83.64 | 5 | 82.21 | 6 | 83.66 | 4 |
| spectf_train | 71.25 | 2 | 71.25 | 2 | 71.25 | 2 | 58.75 | 5 | 61.25 | 4 | 50.00 | 6 |
| statlog-german | 70.30 | 6 | 72.10 | 4 | 76.20 | 1 | 73.20 | 2 | 71.10 | 5 | 72.70 | 3 |
| vehicle | 74.11 | 3 | 76.58 | 2 | 77.52 | 1 | 50.24 | 5 | 60.68 | 4 | 47.42 | 6 |
| waveform-5000 | 76.18 | 6 | 80.84 | 4 | 78.54 | 5 | 86.84 | 1 | 86.50 | 3 | 86.73 | 2 |
| mfeat-karhunen | 97.30 | 5 | 97.35 | 4 | 97.05 | 6 | 97.75 | 2 | 97.85 | 1 | 97.60 | 3 |
| optdigits | 99.07 | 1 | 99.00 | 3 | 99.05 | 2 | 86.29 | 5 | 95.96 | 4 | 86.03 | 6 |
| pollution | 71.66 | 3 | 73.33 | 2 | 78.33 | 1 | 51.71 | 5 | 51.71 | 5 | 51.71 | 5 |
| Rank Total | | 45 | | 34.5 | | 31.5 | | 34.5 | | 43 | | 42.5 |

**Table 5.9 :** Friedman test rankings of different active learning iterations for random subspace dictionary learning model.

| Dataset | Default | Rank | Iter1 | Rank | Iter2 | Rank | Iter3 | Rank | Iter4 | Rank | Iter5 | Rank | Iter6 | Rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cmc | 48.43 | 7 | 49.31 | 6 | 51.02 | 2 | 50.40 | 4 | 51.15 | 1 | 49.93 | 5 | 50.47 | 3 |
| fri_c4_100_10 | 56.00 | 6.5 | 57.00 | 4.5 | 57.00 | 4.5 | 60.00 | 2 | 69.00 | 1 | 56.00 | 6.5 | 59.00 | 3 |
| ionosphere | 91.14 | 6.5 | 91.42 | 5 | 92.28 | 3 | 91.71 | 4 | 91.14 | 6.5 | 92.85 | 1.5 | 92.85 | 1.5 |
| sonar | 72.85 | 7 | 79.04 | 6 | 80.95 | 5 | 82.85 | 4 | 84.28 | 3 | 86.19 | 2 | 87.14 | 1 |
| spectf_train | 60.00 | 5.5 | 57.50 | 7 | 63.75 | 3.5 | 60.00 | 5.5 | 63.75 | 3.5 | 65.00 | 2 | 66.25 | 1 |
| statlog-german | 74.00 | 7 | 74.80 | 6 | 76.80 | 1 | 75.70 | 4 | 76.30 | 3 | 76.40 | 2 | 75.30 | 5 |
| vehicle | 68.58 | 7 | 72.00 | 5 | 70.58 | 6 | 73.41 | 4 | 73.88 | 3 | 74.70 | 2 | 76.00 | 1 |
| waveform-5000 | 74.74 | 6 | 72.20 | 7 | 77.46 | 4 | 77.16 | 5 | 79.52 | 3 | 80.88 | 1 | 80.74 | 2 |
| pollution | 63.33 | 7 | 70.00 | 6 | 73.33 | 3.5 | 71.66 | 5 | 76.66 | 1 | 75.00 | 2 | 73.33 | 3.5 |
| mfeat-karhunen | 15.30 | 7 | 39.95 | 6 | 77.60 | 5 | 92.30 | 3 | 89.40 | 4 | 92.90 | 2 | 95.00 | 1 |
| optdigits | 10.30 | 6 | 9.92 | 7 | 14.19 | 5 | 43.23 | 4 | 83.07 | 3 | 96.65 | 2 | 97.97 | 1 |
| Rank Total | | 72.5 | | 65.5 | | 42.5 | | 44.5 | | 32 | | 28 | | 23 |

**Table 5.10 :** Friedman test rankings of different active learning iterations for random subspace support vector machines model.

| Dataset | Default | Rank | Iter1 | Rank | Iter2 | Rank | Iter3 | Rank | Iter4 | Rank | Iter5 | Rank | Iter6 | Rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cmc | 52.04 | 7 | 52.10 | 6 | 53.46 | 5 | 53.94 | 3 | 54.28 | 1 | 54.08 | 2 | 53.87 | 4 |
| fri_c4_100_10 | 51.00 | 6 | 50.00 | 7 | 56.00 | 5 | 58.00 | 3.5 | 60.00 | 1 | 58.00 | 3.5 | 59.00 | 2 |
| ionosphere | 90.00 | 7 | 93.14 | 6 | 94.57 | 4 | 94.57 | 4 | 94.57 | 4 | 95.14 | 2 | 95.42 | 1 |
| sonar | 69.52 | 7 | 75.71 | 4.5 | 78.09 | 1 | 75.71 | 4.5 | 74.76 | 6 | 77.14 | 3 | 77.61 | 2 |
| spectf_train | 47.50 | 7 | 55.00 | 3 | 52.50 | 6 | 55.00 | 3 | 55.00 | 3 | 55.00 | 3 | 55.00 | 3 |
| statlog-german | 72.00 | 7 | 72.60 | 6 | 73.00 | 5 | 73.40 | 4 | 74.20 | 2 | 74.10 | 3 | 74.30 | 1 |
| vehicle | 42.23 | 5 | 40.58 | 6 | 40.35 | 7 | 43.88 | 4 | 44.00 | 3 | 46.47 | 2 | 47.64 | 1 |
| waveform-5000 | 85.46 | 7 | 85.48 | 6 | 85.56 | 5 | 86.06 | 4 | 86.22 | 2 | 86.10 | 3 | 86.58 | 1 |
| pollution | 50.00 | 2.5 | 50.00 | 2.5 | 50.00 | 2.5 | 50.00 | 2.5 | 46.66 | 6 | 46.66 | 6 | 46.66 | 6 |
| mfeat-karhunen | 94.05 | 6 | 93.10 | 7 | 96.20 | 1 | 95.95 | 4 | 96.00 | 2.5 | 96.00 | 2.5 | 95.90 | 5 |
| optdigits | 85.71 | 7 | 89.83 | 6 | 91.93 | 5 | 93.11 | 4 | 93.73 | 2 | 93.62 | 3 | 93.98 | 1 |
| Rank Total | | 68.5 | | 60 | | 46.5 | | 40.5 | | 32.5 | | 33 | | 27 |

**Table 5.11 :** Friedman test rankings applied on the last iteration of active learning methods.

| Dataset | ARDL | Rank | ARSVM | Rank | ABDL | Rank | ABSVM | Rank |
|---|---|---|---|---|---|---|---|---|
| cmc | 50.47 | 4 | 53.87 | 1 | 50.74 | 3 | 52.72 | 2 |
| fri_c4_100_10 | 59.00 | 2.5 | 59.00 | 2.5 | 57.00 | 4 | 62.00 | 1 |
| ionosphere | 92.85 | 3 | 95.42 | 2 | 92.00 | 4 | 95.71 | 1 |
| sonar | 87.14 | 1 | 77.61 | 4 | 86.19 | 2 | 82.38 | 3 |
| spectf_train | 66.25 | 1 | 55.00 | 3.5 | 63.75 | 2 | 55.00 | 3.5 |
| statlog-german | 75.30 | 2 | 74.30 | 3 | 75.80 | 1 | 73.60 | 4 |
| vehicle | 76.00 | 1 | 47.64 | 3 | 73.88 | 2 | 45.05 | 4 |
| waveform-5000 | 80.74 | 3 | 86.58 | 1 | 79.18 | 4 | 86.32 | 2 |
| pollution | 73.33 | 2 | 46.66 | 4 | 78.33 | 1 | 55.00 | 3 |
| mfeat-karhunen | 95.00 | 2 | 95.90 | 1 | 41.20 | 4 | 93.60 | 3 |
| optdigits | 97.97 | 1 | 93.98 | 2 | 58.16 | 4 | 84.34 | 3 |
| Rank Total | | 22.5 | | 27 | | 31 | | 29.5 |

**Table 5.12 :** Application of Wilcoxon signed rank test on the pairs of DL/RDL and DL/SVM algoritms.

| Dataset | DL-RDL | \|DL-RDL\| | Rank | DL-SVM | \| DL-SVM \| | Rank |
|---|---|---|---|---|---|---|
| cmc | -2.25 | 2.25 | 8 | -5 | 5 | 6 |
| fri_c4_100_10 | -1 | 1 | 4 | -1.42 | 1.42 | 2 |
| ionosphere | -0.57 | 0.57 | 3 | -2.31 | 2.31 | 4 |
| sonar | -1.9 | 1.9 | 7 | 2.07 | 2.07 | 3 |
| spectf_train | 0 | 0 | X | 12.5 | 12.5 | 8 |
| statlog-german | -1.8 | 1.8 | 6 | -2.9 | 2.9 | 5 |
| vehicle | -2.47 | 2.47 | 9 | 23.87 | 23.87 | 11 |
| waveform-5000 | -4.66 | 4.66 | 10 | -10.66 | 10.66 | 7 |
| pollution | -0.05 | 0.05 | 1 | -0.45 | 0.45 | 1 |
| mfeat-karhunen | 0.07 | 0.07 | 2 | 12.78 | 12.78 | 9 |
| optdigits | -1.67 | 1.67 | 5 | 19.95 | 19.95 | 10 |

Wilcoxon signed rank test has been also obtained for other pairs of algorithms. According to their outcomes, while the couples of DL/BDL and SVM/BSVM algorithms have statistically significant evidence at $\alpha$=0.05 in terms of their classification performances, it is not the case for the other pairs.

In the previous part, we applied Friedman test to the accuracy results of the last iteration of the active learning methods. According to the results, classification performance of the applied methods was resulted as equivalent. In order to learn which pairs of methods have a significant performance difference, Table 5.13 is prepared for showing Wilcoxon signed rank test. The table displays the absolute differences between each couple for each dataset and their rank values. As a consequence, Friedman test and Wilcoxon signed rank test strongly agree that no pairs of models have significant performance difference, i.e. they provide approximate classification performance.

**Table 5.13 :** Application of Wilcoxon signed rank test on the pairs of the last iteration accuracies of the active learning models.

| Dataset | \|ARDL-ARSVM\| | Rank | \|ARDL-ABDL\| | Rank | \|ARDL-ABSVM\| | Rank | \|ARSVM-ABDL\| | Rank | \|ARSVM-ABSVM\| | Rank | \|ABDL-ABSVM\| | Rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cmc | 3.4 | 4 | 0.27 | 1 | 2.25 | 3 | 3.13 | 3 | 1.15 | 4 | 2 | 1 |
| fri_c4_100_10 | 0 | X | 2 | 6 | 3 | 5 | 2 | 2 | 3 | 7 | 5 | 5 |
| ionosphere | 2.57 | 3 | 0.85 | 3 | 2.86 | 4 | 3.42 | 4 | 0.29 | 1 | 3.7 | 3 |
| sonar | 9.53 | 7 | 0.95 | 4 | 4.76 | 6 | 8.58 | 6 | 4.77 | 8 | 3.8 | 4 |
| spectf_train | 11.25 | 8 | 2.5 | 8 | 11.3 | 8 | 8.75 | 7 | 0 | X | 8.8 | 7 |
| statlog-german | 1 | 2 | 0.5 | 2 | 1.7 | 2 | 1.5 | 1 | 0.7 | 3 | 2.2 | 2 |
| vehicle | 28.36 | 10 | 2.12 | 7 | 31 | 11 | 26.24 | 8 | 2.59 | 6 | 29 | 10 |
| waveform-5000 | 5.84 | 6 | 1.56 | 5 | 5.58 | 7 | 7.4 | 5 | 0.26 | 2 | 7.1 | 6 |
| pollution | 26.67 | 9 | 5 | 9 | 18.3 | 10 | 31.67 | 9 | 8.34 | 9 | 23 | 8 |
| mfeat-karhunen | 0.9 | 1 | 53.8 | 11 | 1.4 | 1 | 54.7 | 11 | 2.3 | 5 | 52 | 11 |
| optdigits | 3.99 | 5 | 39.81 | 10 | 13.6 | 9 | 35.82 | 10 | 9.64 | 10 | 26 | 9 |

# 6. CONCLUSIONS AND RECOMMENDATIONS

In this study, ensemble dictionary learning methods are proposed for passive and active learning frameworks. For data classification the aim is to minimize the error which is the combination of reconstruction error and sparsity level. Dictionary base is formed by iteratively updating instance representations (alphas) in each class and sub-dictionaries. RDL and BDL are proposed as the applied ensemble methods using randomly selected features/instances. Taking the properties of ensemble learning into consideration, the performance of dictionary learning can be significantly increased. By the use of randomly selected attributes/instances we get a smaller feature/instance space and construct effective and diverse dictionaries for classifier ensembles. Experimental results show that the combination of randomly selected features/instances provides better results than using single dictionary learning, SVM and SVM ensembles.

In the second stage of the thesis, the proposed RDL and BDL methods are considered in active learning framework and compared with SVM. Firstly, a predefined training set, which is 20% of the whole dataset is used with randomly selected attributes/features to construct the supervised dictionary learning model for the initial classification. Unlabeled data instances are classified using the established model, in this way appropriate class labels are assigned to these instances in each classifier ensemble. Using these class label information, entropy for each instance is calculated. The ones having the highest entropy results are chosen to be queried and added to the training set in the next iteration to construct the new classification model. This process is applied in an iterative manner. The same procedure is repeated for Support Vector Machine classifier as well.

According to the achieved empirical results for eleven datasets, proposed Active Random Subspace Dictionary Learning method has superiority over Active Random Subspace Support Vector Machines method. On the other hand, it is quite the opposite for its bagging ensemble classifier counterparts for DL and SVM. We can conclude that under active learning framework generating feature subspaces when modeling dictionary learning classifier provides better classification accuracy while

instance subspaces result in more accurate consequences for support vector machines model. Furthermore, using an active learner generally increases the chance of improved classification performance as the number of iterations is increased.

As a future work it is intended to select diverse and random features by using mutual information between features and class labels. It is also planned to expand the use dictionary learning with other ensemble methods such as Adaboost.

The author hopes that this research will pave the way of raising the use of active learning and sparse coding in other data classification tasks.

## REFERENCES


**Angluin, D.** (1988). Queries and concept learning. *Machine learning*, *2*, 319-342.

**Abramson, Y., & Freund, Y.** (2004). *Active learning for visual object recognition*. UCSD: Technical report.

**Bertoni, A., Folgieri, R., and Valentini, G.** (2005). Feature selection combined with random subspace ensemble for gene expression based diagnosis of malignancies. In *Biological and Artificial Intelligence Environments* (pp. 29-35). Springer Netherlands.

**Bo, T., and Jonassen, I.** (2002). New feature subset selection procedures for classification of expression profiles. *Genome biology*, *3*(4), 1-0017.

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

**Campbell, C., and Ying, Y.** (2011). Learning with support vector machines. *Synthesis lectures on artificial intelligence and machine learning*, *5*, 1-95.

**Chang, C. C., and Chih-Jen, L.** (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST),* 2, 27.

**Chellapilla, K., Shilman, M., and Simard, P.** (2006). Combining multiple classifiers for faster optical character recognition. In *Document Analysis Systems VII* (pp. 358-367). Springer Berlin Heidelberg.

**Chow, M. L., Moler, E. J., and Mian, I. S.** (2001). Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiological genomics*, *5*, 99-111.

**Cohn, D., Atlas, L., and Ladner, R.** (1994). Improving generalization with active learning. *Machine learning*, *15*, 201-221.

**Demšar, J.** (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1-30.

**Dong, Y. S., and Han, K. S.** (2004). A comparison of several ensemble methods for text categorization. In *Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on* (pp. 419-422). IEEE.

**Elad, M., and Aharon, M.** (2006). Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, *15*, 3736-3745.

**Elad, M.** (2010). *Sparse and Redundant Representations From Theory To Applications In Signal And Image Processing*. New York, USA: Springer-Verlag New York.

**Flach, P.** (2012). *Machine learning: the art and science of algorithms that make sense of data.* New York, USA: Cambridge University Press.

**Han, J., Kamber, M., & Pei, J.** (2011). *Data mining: concepts and techniques*. San Francisco, CA:Elsevier.

**Ho, T. K.** (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *20*, 832-844.

**Holub, A., Perona, P., and Burl, M. C.** (2008). Entropy-based active learning for object recognition. In *Computer Vision and Pattern Recognition Workshops on IEEE Computer Society Conference*, Anchorage, AK, 23-28 June.

**Kuncheva, L. I., Rodríguez, J. J., Plumpton, C. O., Linden, D. E., and Johnston, S. J.** (2010). Random subspace ensembles for fMRI classification. *Medical Imaging, IEEE Transactions on*, *29*, 531-542.

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

**Lee, Y. J., Yeh, Y. R., and Pao, H. K.** An introduction to support vector machines. *National Taiwan University of Science and Technology*.

**Lee, Y. J., Yeh, Y. R., & Pao, H. K.** (2012). *Introduction to support vector machines and their applications in bankruptcy prognosis*. In *Handbook of Computational Finance*. Springer Berlin Heidelberg.

**Lewis, D. D., and Gale, W. A.** (1994, August). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 3-12). Springer-Verlag New York, Inc..

**Liere, R., and Tadepalli, P.** (1997). Active learning with committees for text categorization. *In AAAI/IAAI*, 591-596.

**Lichman, M.** (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

**Liu, Y.** (2004). Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of chemical information and computer sciences*, *44*, 1936-1941.

**Lu, J., Plataniotis, K. N., Venetsanopoulos, A. N., and Li, S. Z.** (2006). Ensemble-based discriminant learning with boosting for face recognition. *Neural Networks, IEEE Transactions on*, *17*, 166-178.

**Mairal, J., Elad, M., & Sapiro, G.** (2008). Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, *17*, 53-69.

**Mairal, J., Bach, F., Ponce, J., and Sapiro, G.** (2009). Online dictionary learning for sparse coding. *International Conference on Machine Learning*, Montreal, Canada.

**Mairal, J., Bach, F., Ponce, J., and Sapiro, G.** (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research, 11*, 19-60.

**Mao, J.** (1998). A case study on bagging, boosting and basic ensembles of neural networks for OCR. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on* (Vol. 3, pp. 1828-1833).

**Nanni, L., and Lumini, A.** (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *36*(2), 3028-3033.

**Olsson, F.** (2009). *A literature survey of active machine learning in the context of natural language processing*. SICS Technical Report (T2009:06).

**Polikar, R.** (2006). Ensemble based systems in decision making, *IEEE Circuits and Systems Magazine, 6*, 21-45.

**Roederer, A.** (2012). *Active learning for classification of medical signals* (Doctoral dissertation). University of Pennsylvania, Department of Computer and Information Science, Philadelphia, Pennsylvania.

**Schnass, K., and Vandergheynst, P.** (2008). Dictionary learning based dimensionality reduction for classification. *The 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP)*, St. Julian's, Malta, 12–14 March.

**Settles, B.** (2010). Active learning literature survey. *University of Wisconsin, Madison*, *52*, 55-66.

**Sprechmann, P., and Sapiro, G.** (2010). Dictionary Learning and Sparse Coding for Unsupervised Clustering. In ICASSP ’10. *IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, USA, 14-19 March.

**Sun, S., and Hardoon, D. R.** (2010). Active learning with extremely sparse labeled examples. *Elsevier Neurocomputing, 73*, 2980-2988.

**Tan, A. C., and Gilbert, D.** (2003). Ensemble machine learning on gene expression data for cancer classification.

**Tian, H., and Meng, B.** (2010). A new modeling method based on bagging ELM for day-ahead electricity price prediction. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on* (pp. 1076-1079).

**Tong, S., and Koller, D.** (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research, 2,* 45-66.

**Tošić, I., and Frossard, P.** (2011). Dictionary learning, *IEEE Signal Processing Magazine, 28*, 27-38.

**Tur, G., Hakkani-Tür, D., and Schapire, R. E.** (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, *45*, 171-186.

**Vanschoren, J., van Rijn, J. N., Leidenuniv, L., Bischl, B., Uni, S., & Casalicchio, G.** (2013). OpenML: a networked science platform for machine learning. *SIGKDD Explorations, 15*, 49-60.

**Vapnik, V.** (1979). *Estimation of dependences based on empirical data [in Russian]*. Nauka, Moscow. (English translation (1982). New York: Springer Verlag.

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

**Xu, J., He, H., & Man, H.** (2014). Active dictionary learning in sparse representation based classification. arXiv preprint arXiv:1409.5763.

**Yeh, C. C. M., and Yang, Y. H.** (2012). Supervised Dictionary Learning for Music Genre Classification. In ICMR '12. *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, Hong Kong, China, 5-8 June.

**Zhang, C., and Chen, T.** (2002). An active learning framework for content-based information retrieval. *Multimedia, IEEE Transactions on*, *4*, 260-268.

**Zhu, Z., Chen, Q., and Zhao, Y.** (2014). Ensemble dictionary learning for saliency detection, *Elsevier Image and Vision Computing, 32*, 180-188.

**CURRICULUM VITAE**

**Name Surname:** Göksu TÜYSÜZOĞLU

**Place and Date of Birth:** Giresun, 1990

**E-Mail:** tuysuzoglug@itu.edu.tr

**EDUCATION:**

- **B.Sc.:** 2013, Dogus University, Engineering Faculty, Information Systems Engineering
- **B.Sc.:** 2013, Dogus University, Engineering Faculty, Industrial Engineering (Double Major)
- **M.Sc.:** 2016, Istanbul Technical University, Faculty of Computer and Informatics Engineering, Computer Engineering

**PROFESSIONAL EXPERIENCE AND REWARDS:**

- 2008-2013 Top Scoring Student in the Programme of Information Systems Engineering at Dogus University

**PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:**

- Tuysuzoglu G., and Yaslan Y., 2016: Data Classification Using Sparse Coding Based Active Learning. 24. Signal Processing and Communications Applications Conference, May 16-19, 2016 Zonguldak, Turkey.
- 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.