ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Barış DİNÇ

A HYBRID APPROACH FOR FEATURE REDUCTION

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

A HYBRID APPROACH FOR FEATURE REDUCTION

Barış DİNÇ

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was received and approved the award of degree of the Master of Science by the board of jury on 04/07/2019.

……..….....	……..…	……..……..……
Asst. Prof. Dr. Buse Melis ÖZYILDIRIM	Prof. Dr. Selma Ayşe ÖZEL	Asst. Prof. Dr. Onur ÜLGEN
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Computer Engineering Department of Institute of Natural and Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

This work is supported by the Çukurova University Academic Research ProjectsUnit. Project Number: FYL-2018-11162

ABSTRACT

MSc THESIS

A HYBRID APPROACH FOR FEATURE REDUCTION

Barış DİNÇ

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
Year: 2019, Pages: 127
Juries : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
: Prof. Dr. Selma Ayşe ÖZEL
: Asst. Prof. Dr. Onur ÜLGEN

In this thesis, currently available feature selection and dimension reduction methods are analyzed on three different datasets. Particular emphasis is given to filtering-based feature selection methods, which are popular in machine learning area due to their speed. In addition, a linear generalization of the features is obtained by applying dimension reduction in case the scores of the features are close to each other. The feature selection methods that are examined in this thesis evaluate features individually by ignoring correlation between them. At this stage, features marked as redundant are accepted completely irrelevant. Hence, in this thesis, a hybrid feature reduction approach is proposed. In this approach, unchosen features have also opportunity to be involved in classification and clustering by providing a linear projection. Hence, with this approach both the most relevant features selected by feature selection algorithm and projection of unchosen features are utilized. The results obtained in terms of f-measurement show that, besides selecting the best feature subset consisting of top-n features, giving a perfect-fit chance to unchosen features may lead to satisfactory classification results.

Keywords: Feature Selection, Dimension Reduction, Classification, Clustering, Curse of Dimensionality

ÖZ

YÜKSEK LİSANS TEZİ

ÖZNİTELİK AZALTMA İÇİN HİBRİT BİR YAKLAŞIM

Barış DİNÇ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM
Yıl: 2019, Sayfa: 127
Jüri : Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM
: Prof. Dr. Selma Ayşe ÖZEL
: Dr. Öğr. Üyesi Onur ÜLGEN

Bu tezde, şu anda mevcut olan öznitelik seçimi ve boyut azaltma yöntemleri üç farklı veri kümesi üzerinde analiz edilmiştir. Makine öğrenim alanında hızları nedeniyle popüler olan filtreleme temelli öznitelik seçim yöntemlerinin üzerinde durulmuştur. Ek olarak, özniteliklerin puanlarının birbirine yakın olması durumunda boyut azaltma uygulanarak özniteliklerin doğrusal bir genellemesi elde edilmiştir. Bu tezde incelenen öznitelik seçme yöntemleri, aralarındaki ilişkiyi göz ardı ederek özniteilkleri ayrı ayrı değerlendirmektedir. Bu aşamada, gereksiz olarak işaretlenen öznitelikler tamamen ilgisiz kabul edilir. Bu nedenle, bu tezde, bir hibrid öznitelik azaltma yaklaşımı önerilmiştir. Bu yaklaşımda, seçilmemiş öznitelikler, doğrusal izdüşümleri sağlanarak sınıflandırma ve kümelemeye katılma fırsatına da sahiptir. Dolayısıyla, bu yaklaşımla hem öznitelik seçimi algoritması tarafından seçilen en alakalı öznitelikler hem de seçilmemiş özniteliklerin projeksiyonu kullanılmaktadır. F-ölçümü açısından elde edilen sonuçlar, en yüksek skora sahip özniteliklerden oluşan en iyi öznitelik altkümesinin seçilmesinin yanı sıra, seçilmemiş özniteliklere mükemmel uyum şansı vermenin, tatmin edici sınıflandırma sonuçlarına neden olabileceğini göstermektedir.

Anahtar kelimeler: Öznitelik Seçimi, Boyut Azaltma, Sınıflama, Kümeleme, Boyutsallığın Laneti

EXTENDED ABSTRACT

In recent years, studies in field of bioinformatics have accelerated thanks to developments in genetic technology. Especially after completion of human genome project, an important step was taken to understand the knowledge behind the genetic codes. The project was carried out by transforming of parts belonging to base pairs of human genome to artificial chromosome and reproducing in bacteria. After sequence analysis, these fragments were located in the human genome. Hence, it is aimed to extract the DNA map can be called the identity card of human. In this way, the first studies on functioning of genetic code and meaning of series have begun. The need for analysis of the data obtained as a result of the studies has led to the use of machine learning techniques in this field. The accurate analysis of obtained gene information is important in terms of diagnosis of normal and diseased individuals and detection of disease-related genes. Obtaining reliable results from complex datasets is one of the major problems for machine learning techniques due to their high dimensional structure. Moreover, number of samples on these datasets tends to be small. Small sample size is another important problem for machine learning researchers. It is difficult to generalize on high dimensional datasets with a small amount of data for the learning algorithm. The curse of dimensionality affects the success of classifiers and causes scalability problem. Moreover, increasing the number of features causes the search space to grow exponentially.

In this study, the aim is to investigate the effect of preprossesing approaches such as feature selection and dimension reduction on classification and clustering f-measure of high-dimensional bioinformatics datasets. Therefore 10000, 7000 and 20581 dimensional datasets named Arcene, Gisette and Gene Expression Cancer RNA-Seq are selected respectively to determine the effect of the 11 feature selection and 3 dimension reduction methods on the classification and clustering.

In the first experiment, 11 well known feature selection and 3 dimension reduction methods on the 3 datasets mentioned above are tested. The 11 feature selection methods handle features individually and mark all the features that are below a certain threshold as completely irrelevant. If the scores of the features are close to each other, it is more logical to move the dataset to a new space by applying dimension reduction instead of returning a subset of the original feature set. Then, a hybrid approach is proposed to improve feature selection approach efficiency. The idea behind the proposed approach is using features marked as irrelevant by feature selection algorithm in the class approach. For this aim, after applying feature selection algorithm linear projection of irrelevant features is utilized.

In the second experiment, proposed method is tested on Colon, Leukemia, Prostate_GE, ALLAML bioinformatic datasets where the number of data tends to be small compared to the number of features and a different Madelon dataset besides Arcene, Gisette and Gene Expression Cancer RNA-Seq datasets.

While the first experiments are done on MLP, Naive Bayes, LDA and SVM classifiers besides K-Means and AGNES clustering methods. Second experiments are done on only the classifiers due to the nature of proposed algorithm. Test results that are obtained by combining the feature selection method ReliefF and dimension reduction method Principal Component Analysis by after concatenating 25 and 50-dimensional projection of irrelevant features in PCA and top-50 features selected by ReliefF are compared with the classification results that is gotten with the top-100 features selected by ReliefF. Thus, it is understood the effect of low and high dimensional representation of irrelevant features in classification.

According to the test results, we see that with 50-dimensional projection of irrelevant features, provide the most successful results on some datasets in all 4 classifiers. This is due to the fact that instead of completely ignoring irrelevant

features, including a projection in the classification gives a chance for improvement.

Furthermore, it is understood that performing 25-dimensional projection instead of representing irrelevant features in 50 dimensions leads to higher f-measure results on some datasets. While the proposed method causes better results than original ReliefF on 6 of 8 datasets in MLP classifier, this number is 4 in Naive Bayes, 3 in LDA classifier and 3 in SVM. In Naive Bayes classifier, the proposed method has the same classification success as ReliefF on 2 datasets. Similarly, performance of the proposed method in classifying on a dataset in the SVM classifier achieves the same success with ReliefF. Due to the excessive dimensionality and memory problems, Gene Expression Cancer RNA-Seq dataset cannot be used with in the LDA classifier.

In the literature, Canedo, Marono, Betanzos, Benitez and Herrera's studies on microarray data sets, 7 classical feature selection methods were tested on 9 binary class datasets with 3 classifiers such as C4.5, SVM and Naive Bayes. In these studies, the most successful classification results were obtained by SVM and Naive Bayes, and the superiority of SVM over other classifiers was mentioned (Canedo, Marono, Betanzos, Benitez and Herrera, 2014).

Considering the feature selection methods used in these studies, on average, superior success was achieved with CFS and Interact for all datasets. In addition, SVM-RFE performed poorly with the SVM classifier, which was expected to yield good results.

The results obtained with feature selection methods are multivariate except Information Gain. However, Information Gain, on average, led to similar results for all datasets.

Although the methods used in the studies are based on knowledge theory (Information Gain, MRMR, FCBF and Interact) or correlation coefficients (CFS, ReliefF and SVM-RFE), the behavior of the methods of the same category according to the classifier and data tested may vary (Canedo et al, 2014).

In the study conducted by Canedo, Marono, Betanzos, Benitez and Herrera, it was emphasized that the best classification success was obtained by SVM classifier and DOB-SVC validation method, but it was emphasized that the dataset and classifier should be analyzed well besides the feature selection method used for successful classification.

GENİŞLETİLMİŞ ÖZET

Son yıllarda, genetik teknolojideki gelişmeler sayesinde biyoinformatik alanındaki çalışmalar hızlanmıştır. Özellikle insan genom projesinin tamamlanmasından sonra, genetik kodların arkasındaki bilgiyi anlamak için önemli bir adım atıldı. Proje, insan genomunun baz çiftlerine ait parçaların yapay kromozomlara dönüştürülmesi ve bakterilerde çoğaltılmasıyla gerçekleştirildi. Dizi analizinden sonra, bu parçalar insan genomunda yerleştirildi. Böylece insanın kimlik kartı da denilebilen DNA haritasının çıkarılması hedeflendi. Bu şekilde, genetik kodun işleyişi ve dizilerin anlamları üzerine ilk çalışmalar başlamıştır. Çalışmalar sonucunda elde edilen verilerin analizine duyulan ihtiyaç bu alanda makine öğrenme tekniklerinin kullanılmasına yol açmıştır. Elde edilen gen bilgilerinin doğru analizi, normal ve hastalıklı bireylerin teşhisi ve hastalıklarla ilişkili genlerin tespiti açısından önemlidir. Karmaşık veri kümelerinden güvenilir sonuçlar elde etmek, yüksek boyutlu yapıları nedeniyle makine öğrenimi teknikleri için en önemli sorunlardan biridir. Ayrıca, bu veri kümelerindeki numunelerin sayısı küçük olma eğilimindedir. Küçük örneklem büyüklüğü, makine öğrenmesi araştırmacıları için bir başka önemli sorundur. Öğrenme algoritması için az sayıda veriyle yüksek boyutlu veri kümeleri üzerinde genelleme yapmak zordur. Boyutsallığın laneti, sınıflandırıcıların başarısını etkiler ve ölçeklenebilirlik sorununa neden olur. Ayrıca, özniteliklerin sayısının arttırılması, arama alanının katlanarak büyümesine neden olur.

Bu çalışmada, öznitelik seçimi ve boyut azaltma gibi ön işleme yaklaşımlarının yüksek boyutlu biyoinformatik veri setlerinin sınıflandırma ve kümeleme f-ölçüsü üzerindeki etkisinin araştırılması amaçlanmıştır. Bundan dolayı, Arcene, Gisette ve Gene Expression Cancer RNA-Seq adlı sırasıyla 10000, 7000 ve 20581 boyutlu veri setleri, 11 öznitelik seçiminin ve 3 boyut azaltma yönteminin sınıflandırma ve kümeleme üzerindeki etkisini belirlemek için seçilmiştir.

İlk deneyde, iyi bilinen 11 öznitelik seçim ve 3 boyut azaltma yöntemi yukarıda belirtilen 3 veri setinde test edilmiştir. 11 öznitelik seçim yöntemi öznitelikleri ayrı ayrı ele alır ve belirli bir eşiğin altındaki tüm öznitelikleri tamamen alakasız olarak işaretler. Özniteliklerin puanları birbirine yakınsa, orijinal öznitelik kümesinin bir alt kümesini döndürmek yerine boyut azaltma uygulayarak veri kümesini yeni bir uzaya taşımak daha mantıklıdır. Ardından, öznitelik seçiminin yaklaşım verimliliğini artırmak amacıyla karma bir yaklaşım önerilmektedir. Önerilen yaklaşımın ardındaki fikir, sınıf yaklaşımında öznitelik seçim algoritmasıyla alakasız olarak işaretlenmiş öznitelikleri kullanmaktır. Bu amaçla, öznitelik seçim algoritması uygulandıktan sonra alakasız özniteliklerin doğrusal projeksiyonu kullanılmıştır.

İkinci deneyde, önerilen yöntem Arcene, Gisette ve Gene Expression Cancer RNA-Seq veri kümelerinin yanında, veri sayısının öznitelik sayısına göre az olma eğilimi gösterdiği Colon, Lösemi, ALLAML, Prostate_GE veri kümeleri ile farklı bir Madelon veri kümesi üzerinde test edilmiştir.

İlk deneyler K-Means ve AGNES kümeleme yöntemlerinin yanı sıra MLP, Naive Bayes, LDA ve SVM sınıflandırıcıları üzerinde yapılır. Önerilen algoritmanın yapısından dolayı ikinci deneyler sadece sınıflandırıcılar üzerinde yapılır. PCA'de ilgisiz özniteliklerin 25 ve 50 boyutlu projeksiyonları ile ReliefF tarafından seçilen en ilgili 50 öznitelik birleştirildikten sonra, öznitelik seçim yöntemi ReliefF ve boyut azaltma yöntemi Temel Bileşen Analizi bir araya getirilerek sonuçlar ReliefF tarafından seçilen en ilgili 100 öznitelik ile elde edilen sınıflama sonuçları ile karşılaştırıldı. Böylece ilgisiz özelliklerin düşük ve yüksek boyutlu temsilinin sınıflamadaki etkisi anlaşılmıştır.

Test sonuçlarına göre, 50 boyutlu ilgisiz özniteliklerin projeksiyonuyla, 4 sınıflandırıcının hepsinde bazı veri kümelerinde en başarılı sonuçları sağladığını görüyoruz. Bunun nedeni, ilgisiz özellikleri bütünüyle yok saymak yerine bir projeksiyonunu sınıflamaya dahil etmenin, bir iyileştirme şansı vermesidir.

Ayrıca, 50 boyutta ilgisiz öznitelikleir temsil etmek yerine 25 boyutlu projeksiyon yapmanın bazı veri setlerinde daha yüksek f-ölçüm sonuçlarına yol açtığı anlaşılmaktadır. Önerilen yöntem, MLP sınıflandırıcısındaki 8 veri setinin 6'sında orijinal ReliefF'ten daha iyi sonuçlara neden olurken, bu sayı Naive Bayes'de 4, LDA sınıflandırıcısında 3 ve SVM'de 3'tür. Naive Bayes sınıflandırıcısında, önerilen yöntem 2 veri setinde ReliefF ile aynı sınıflandırma başarısına sahiptir. Benzer şekilde, önerilen yöntemin SVM sınıflandırıcısın bir veri setini sınıflandırmadaki performansı, ReliefF ile aynı başarıyı sağlar. Aşırı boyutsallığı ve hafıza problemleri nedeniyle, Gene Expression Cancer RNA-Seq veri seti LDA sınıflandırıcısında kullanılamaz.

Literatürde, Canedo, Marono, Betanzos, Benitez ve Herrera'nın mikroarray veri setleri konusundaki çalışmaları, 7 klasik öznitelik seçim yöntemi, C4.5, SVM ve Naive Bayes gibi 3 sınıflandırıcı ile 9 ikili veri setinde test edilmiştir. Bu çalışmalarda en başarılı sınıflandırma sonuçları SVM ve Naive Bayes tarafından elde edilmiş ve SVM'nin diğer sınıflandırıcılara göre üstünlüğünden bahsedilmiştir(Canedo, Marono, Betanzos, Benitez ve Herrera, 2014).

Bu çalışmalarda kullanılan öznitelik seçim yöntemleri düşünüldüğünde, ortalama olarak, tüm veri setleri için CFS ve Interact ile üstün başarı sağlanmıştır. Ek olarak, SVM-RFE, iyi sonuçlar vermesi beklenen SVM sınıflandırıcı ile düşük performans göstermiştir.

Öznitelik seçme yöntemleri ile elde edilen sonuçlar, Bilgi Kazancı haricinde çok değişkendir. Yalnız, Bilgi Kazancı, ortalama olarak, tüm veri kümeleri için benzer sonuçlara yol açmıştır.

Çalışmalarda kullanılan yöntemler bilgi teorisine (Bilgi Kazancı, MRMR, FCBF ve Interact) veya korelasyon katsayılarına (CFS, ReliefF ve SVM-RFE) dayanmasına rağmen, aynı kategorideki metotların sınıflandırıcı ve test edilen verilere göre davranışı değişebilir (Canedo ve diğerleri, 2014).

Canedo, Marono, Betanzos, Benitez ve Herrera tarafından yapılan çalışmada, en iyi sınıflandırma başarısının SVM sınıflandırıcısı ve DOB-SVC

validasyon yöntemi ile elde edildiği, ancak başarılı sınıflandırma için kullanılan özellik seçimi yönteminin yanında veri kümesinin ve sınıflandırıcının da iyi analiz edilmesi gerektiği vurgulanmıştır

ACKNOWLEDGEMENT

I would like to express my endless thanks and appreciation to my supervisor, Asst. Prof. Dr. Buse Melis ÖZYILDIRIM, for all respectable knowledge, support, and patience. I would like to thank members of my defense committee Prof. Dr. Selma Ayşe Özel and Assist. Prof. Dr. Onur ÜLGEN for their time and contributions.

I would like to extend my gratitude to my parents, Mehmet Dinç and Aysel Dinç, my sister, Ezgi Dinç, for their belief, support, patience, motivation and encouragement.

I would like to thank Çukurova University Academic Research Projects Unit for their support.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

1. INTRODUCTION

In recent years thanks to developments in genetic technology, studies in the field of bioinformatics have accelerated. Especially after completion of human genome project, an important step was taken to understand the knowledge behind the genetic codes. The project was carried out by transforming of parts belonging to base pairs of human genome to artificial chromosome and reproducing in bacteria. After sequence analysis, these fragments were located in the human genome. Therefore, it is aimed to extract the DNA map that can be called the identity card of human. In this way, the first studies on functioning of genetic code and meaning of series have begun. In order to solve meaning of the obtained genetic codes, very strong computerized analysis techniques are needed.

The need for analyzing of data obtained from studies has led to the use of machine learning techniques in this field. The faultless analysis of obtained gene information is important in terms of diagnosis of normal and diseased individuals and detection of disease-related genes. Obtaining reliable results from complex datasets is one of the major problems for machine learning techniques because of their high dimensional structure. Moreover, number of samples on these datasets tends to be small. Small sample size is another major problem for machine learning researchers. It is difficult to generalize on high dimensional datasets with a small amount of data for the learning algorithm. The curse of dimensionality affects the success of classifiers and causes scalability problem. Moreover, increasing the number of features causes the search space to grow exponentially (Debie and Shafi,2017). It is possible to constitute a meaningful model for a classification problem with a successful training on a high-dimensional dataset. The aim is to maximize generalization ability of the learning algorithm to adapt to new data points. If the model is overworked to adapt itself to training samples, instead of ensuring a generalization, it becomes more complex by increasing number of parameters. This is known as “overfitting” (Dietterich, 1995). In the overfitting

problem, the learning algorithm tries to estimate the most accurate predictions for given data instead of focusing on to solve major problem. It processes the training data individually rather than producing an inclusive model.

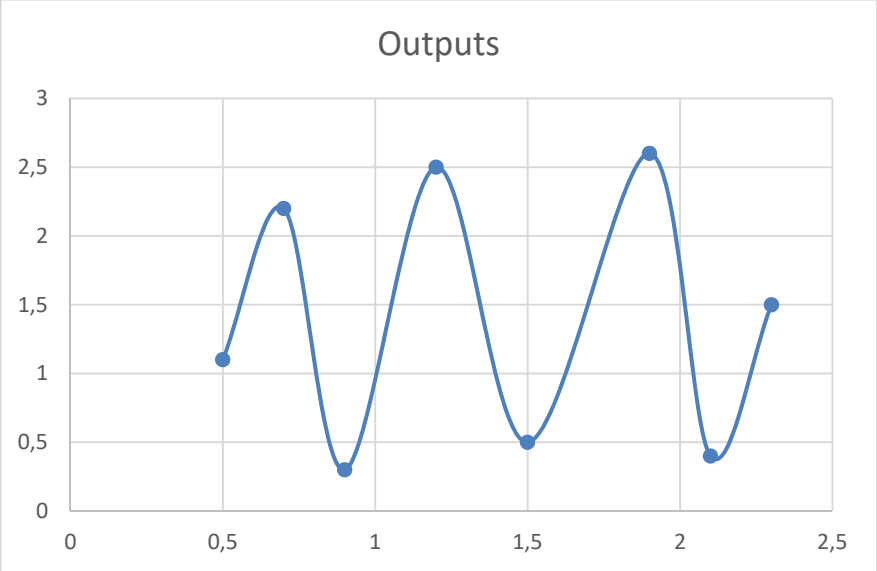


Figure 1.1 Overfitting chart

In Figure 1.1 the concept of overfitting is shown schematically. The horizontal axis represents inputs while the outputs correspond in vertical. The approach function shows an excessive fluctuation between training data which leads to weak generalization. As the learning algorithm focuses on resolving the noise, this leads to an increase in learning complexity (Narayan and Tagliarini, 2005).

On the contrary, if a statistical model is not complex enough to establish an accurate relation between inputs and target variables, the learning algorithm cannot train itself. For example, a linear model that tries to capture linear predictions on a non-linear dataset tends to make weak performance. In case the learning model is incomplete, “underfitting” occurs. An underfit model has high error in both training and testing while an overfit model has high testing error even if it produces

excellent results in training. Graphical representation of overfitting and underfitting are shown in following figure:

Figure 1.2. Schematic representation of underfitting and overfitting ("Underfitting", n.d.)

In a multi parameter machine learning structure, if the number of samples is too small in comparison with number of attributes, increasing number of features causes a weak generalization. In a similar way, although the training error decreases when learning time is increased, a successful prediction for test data may not be performed. As the complexity increases, more parameters will come into play, and the generalization will get worse. On the contrary, in the case of underfitting, a longer training period may be a solution.

In the cases where the learning algorithm is not sufficiently complex, a short-fat or long-thin architectures should be preferred according to model's need. Since a bigger model causes use of more parameters in learning, complexity will increase and underfitting may be solved.

One way to avoid overfitting is regularization. Regularization prevents overfitting by adding a penalty term to error function. If the training set contains too much noise, the trained model cannot make successful predictions for new data points. Regularization prevents the coefficients from taking excessive values by punishing them; hence, it reduces estimates towards zero or regularized. Suppose that training data consisting of independent variables $X = \{x_1, x_2, \dots, x_n\}$ and desired outputs $d = \{d_1, d_2, \dots, d_n\}$. The loss function is:

$$RSS = \sum_{i=1}^{n}(d_i - \sum_j \beta_j x_{ij} + \beta_0)^2 \quad (1.1)$$

If the function is regulated by using a penalty term:

$$\sum_{i=1}^{n}(d_i - \sum_j \beta_j x_{ij} + \beta_0)^2 + \lambda \sum_j {\beta_j}^2 \quad (1.2)$$

where λ is penalty parameter that determines the effect of regularization on learning. If $\lambda = 0$, the regularization term has no effect and the loss function will become equal to least squares. As λ grows, the penalty term further reduces the learned coefficients and variance that occurred because of noise. Regularization ensures that the model gains a generalization capability by reducing variance. However, after a point, the rise in λ causes the model to lose the parameters that it has learned and underfitting occurs. Therefore, optimal selection of λ is critical in terms of avoiding both high bias and high variance.

Another approach to avoid overfitting is dropout. Dropout refers giving up certain set of neurons that are selected randomly during training phase. In other words, all the incoming and outcoming connections are ignored by being dropped out a neuron. Therefore, many "thin" sub-neural networks derived from a neural network are formed. Each thin neural network is trained in itself. Training a dropout neural network means training a collection of various thinned neural networks. In the following figures, a fully connected and dropped out neural networks are shown:

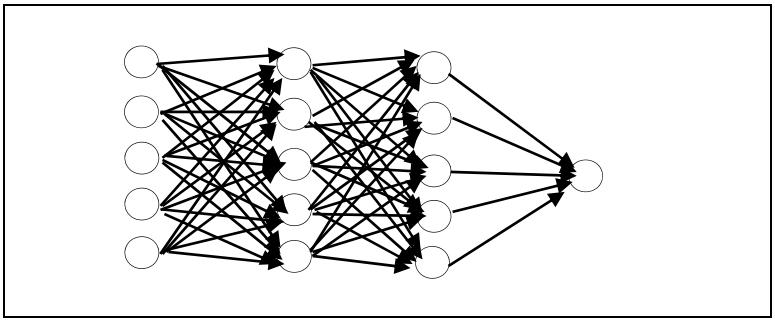


Figure 1.3 Fully connected Neural Network

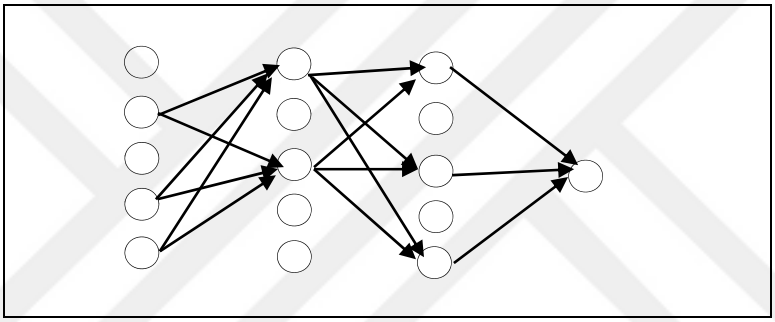


Figure 1.4 Dropped out Neural Network

In a neural network consisting of n units, it can be created 2^n dropped out sub-neural networks (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014).

Working on more training data can help the model to adapt more easily to new data points and help to prevent overfitting. However, if the training set contains too much noise, this would cause high variance. Variance is variety of model prediction for a given certain data. A model with high variance makes a lot of effort into adapting to training data, however this causes it to fail to generalize. In contrast to variance, bias is the difference between desired variable and estimated variable by the model. High biased model tends to simplify itself by giving less importance to training data, hence it generates high error in both training and testing. Therefore, the data should be balanced and clear for a

successful training and testing. The following figure shows precautions to be taken in case of underfitting, called high bias and overfitting, high variance:

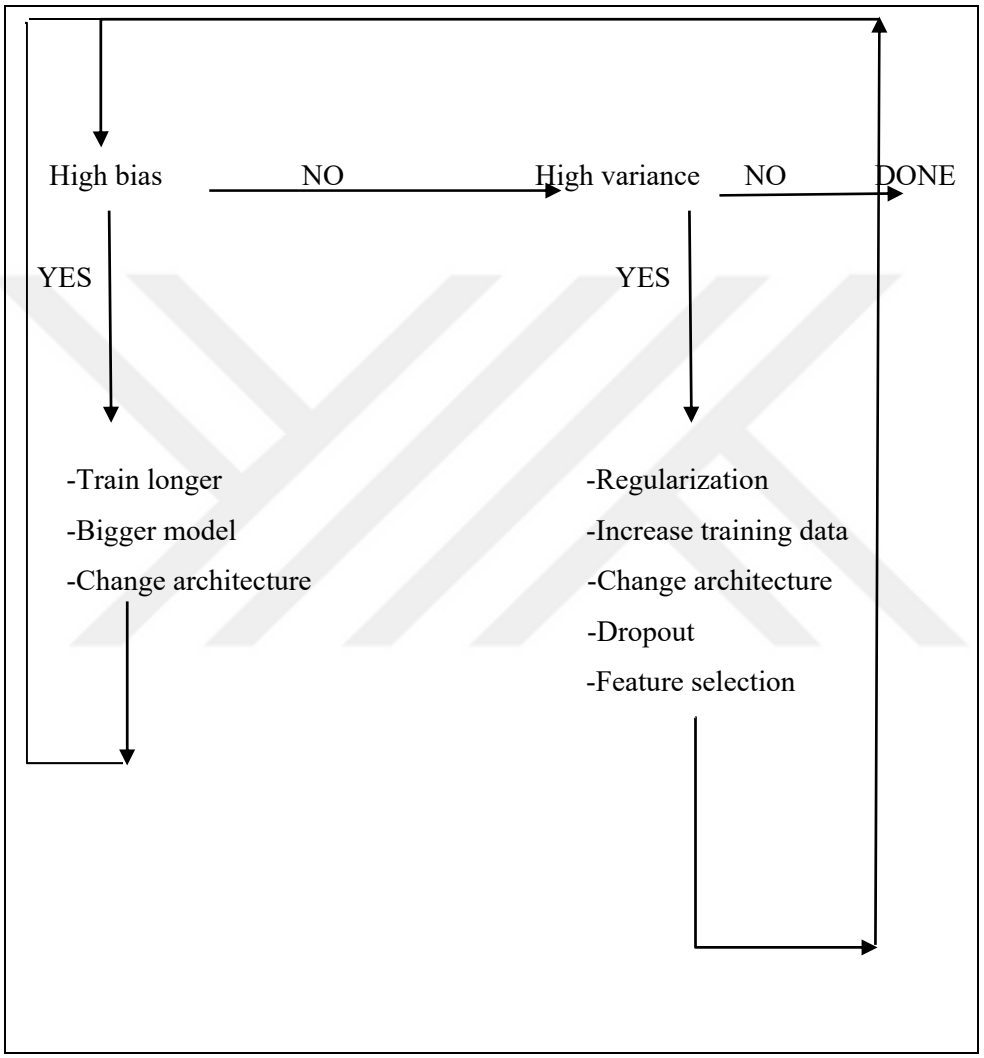


Figure 1.5 High bias-High variance trade off flowchart (Mallick, 2017)

In the Figure 1.6 a visual scenario of bias-variance trade-off is shown:

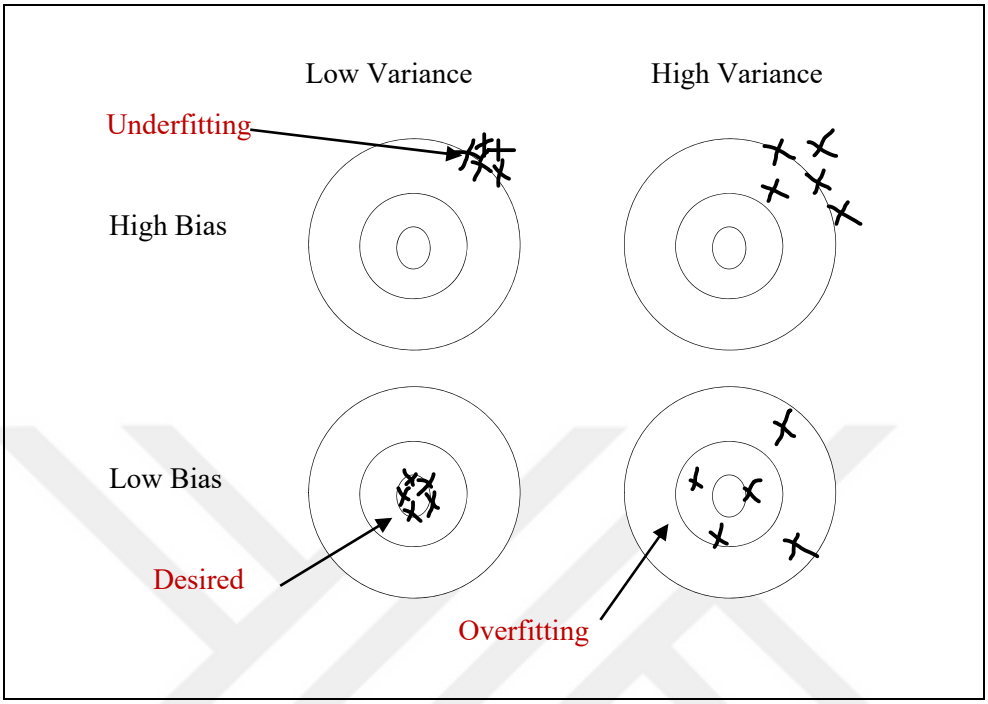


Figure 1.6 Bias and Variance Bulls-eye Diagram (Domingos, 2012)

One of major problems on high dimensional datasets is that there may be many irrelevant features in feature set. In such a case, these features have negative effects on the learning algorithm such as increasing the computational cost and time complexity. The accuracy of the learning algorithm is reduced due to presence of unnecessary features (Langley and Sage, 1994). Therefore, there exist some methodologies for reducing number of features in datasets. There are two major approaches for feature reduction: One is feature selection that returns the subset of relevant features and the other one is dimension reduction that projects the original data to a new feature space. In the feature selection methods, feature correlations are not considered. Hence, the redundant features may not be detected by the feature selection methods and although they are meaningless for the analysis, they may remain in the feature set. In such a case it is reasonable to produce new dimensions from original feature set instead of selecting them.

Selection of the appropriate feature subset has a significant impact on the classification performance (Koller and Sahami, 1996). Using feature selection, the problem of finding “false positive” arising from high dimensionalism is resolved. In addition, feature selection techniques on microarrays datasets do not only remove irrelevant features but also give researchers the chance to interpret genes that causes a certain disease (Canedo et al, 2014).). Feature selection methods can be divided into categories such as filter, wrapper and embedded methods.

Filter methods do not rely on any foresight, they only use the general characteristics of data (Marono, Betanzos and Sanroman, 2007). For this reason, they are faster than other feature selection techniques. They evaluate features individually and rank according to their scores. Since they do not use any machine learning method, their results are unbiased than wrapper methods (Wang, Wang and Chang, 2016).

Another feature selection method is wrapper models. Unlike filter methods, feature selection in wrapper models is performed using a machine learning approach (such as classifier). Undoubtedly this approximation also brings a computational cost and time complexity. Although wrapper methods tend to yield better results, filter methods for complex datasets have always been cheaper in terms of cost (Marono et al., 2007).

Contrary to wrapper models, in embedded methods features are selected after a learning phase in which all the features are taken into account (Lal, Chapelle, Weston and Elisseeff, 2006). Information is taken from the learning algorithm rather than directly from data. The basic problem is here that the information obtained from different learning algorithms may vary.

When the features show similarities in terms of the information they possess, feature selection may not distinguish them. In such a case, rather than returning a feature subset, it is necessary to represent original variables with new dimensions in a lower domain by creating the widest variance. This type of feature extraction method is known as dimension reduction. The most known and used

dimension reduction techniques in literature are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). While PCA is unsupervised, LDA is a supervised technique.

In this thesis, feature selection and dimension reduction techniques such as PCA, LDA and DBN are tested and analyzed on low, medium and large-scale datasets. As a result of analysis, it is seen that existing feature selection methods evaluate features individually and do not give the chance them to demonstrate perfect-fit by getting together. As a consequence of these disadvantages that arise from approaches of filter-based feature selection methods, a new hybrid feature selection technique is proposed by considering the projection of features that are marked as completely irrelevant in this thesis. Therefore, ReliefF that is one of features selection methods and PCA, one of the rotation processes of original feature set is used. In literature, the similar approaches exist. In the experiment that was performed by Zeng, Wang, Zhang and Cai for the classification of underwater sound by applying ReliefF and PCA, firstly PCA was used to maximize the variance of the features then, ReliefF was applied to select the most scored features (Zeng, Wang, Zhang, Cai, 2013). Although the variance of features were tried to be maximized by applying PCA, the classification was made only with the top-n features which were selected after rotation process by ignoring the features that had low variance. Unlike this study, in this thesis linear projection of all irrelevant features is calculated by using PCA besides selecting the most relevant feature set by ReliefF. Thus, after selecting the most relevant features by ReliefF, we give the chance to show perfect-fit to irrelevant features. Moreover, the relevant features are selected by staying loyal to original feature set instead of carrying them to another feature space by maximizing the variance. The validation of the proposed hybrid method was tested on the 8 high-dimensional datasets to prove its ability a general characteristic.

2. REVIEW OF RELATED WORKS

The most commonly used feature selection algorithms belong to filter category as they are not based on any foresight and their calculation speed. Because of their computational cost, it is not a good idea to use wrapper methods on high dimensional datasets. Although embedded methods have begun to attract attention in recent years they are not as popular as filter-based techniques.

The following mostly used feature selection methods in the literature are divided into subsections as information theoretical, statistical and similarity-based feature selection.

2.1. Information Theoretical Based Feature Selection

2.1.1. Information Gain

Suppose that the class labels are represented in $L_1 \dots L_c$. Let the probability of each part be: $P(D) = \frac{D}{L}$ for any subset D. Suppose that a feature F_i has $A_1 \dots A_k$ values in training set. Let $P(L_c|A_k) = \frac{P(L_c \cap A_k)}{P(A_k)}$.

In this case information gain is defined as follows:

$$I_{Gain} = H\big(P(L_1), \dots, P(L_c)\big) - \sum_{k=1}^{K} P(A_k) H(P(L_1|A_k), \dots, P(L_c|A_k)) \quad (2.1)$$

where H is the entropy function (Xing, Jordan and Karp, 2001). In fact, information gain measures the expected reduction in entropy (Mitchell,1997). A feature F_i which has high gain value is expected to separate and represent classes effectively.

2.1.2. Minimal Redundancy-Maximal Relevance (MRMR)

MRMR is a filter-based feature selection algorithm that selects the features having the highest relevance in terms of classes and removes the most distant features by using mutual information. The main idea of MRMR is to penalize the feature at the rate of redundancy which it has.

For class C, the relevance level of an attribute set F is equal to mean of mutual information between individual feature F_i and class C.

$$r(F,C) = \frac{1}{|F|}\sum_{F_i \in F} I(F_i;C) \tag{2.2}$$

The redundancy of F_i is average of mutual information between F_i and all the features F_j.

$$w(F) = \frac{1}{|F^2|}\sum_{F_i,F_j \in F} I(F_i;F_j) \tag{2.3}$$

MRMR is calculated by combining redundancy and relevance information:

$$mrmr = max[\frac{1}{|F|}\sum_{F_i \in F} I(F_i;C) - \frac{1}{|F^2|}\sum_{F_i,F_j \in F} I(F_i;F_j)] \tag{2.4}$$

2.1.3. Interact

Interact uses a goodness measure based on the symmetric uncertainity (SU)(Canedo et al., 2014). The algorithm consists of two parts: In the first phase, the features are ranked in descending order in terms of their SU values. Secondly, they are evaluated one by one starting from end of the sorted list. If the coherence contribution of a feature is lower than a specified threshold, corresponding attribute is removed, otherwise it is selected. Let $H(Y)$ and $H(Y,C)$ be entropy and joint

entropy and $I(Y,C) = H(Y) + H(C) - H(Y,C)$ mutual information between two variables (Zhao and Liu, 2007).

SU between class label C and attribute A_i is:

$$SU(A_i, C) = 2[\frac{I(A_i,C)}{H(A_i)+H(C)}] \quad (2.5)$$

2.2. Statistical Based Feature Selection

2.2.1. Gini Index

Gini Index evaluates the uncleanliness of a data set which are the part of training tuples divided into D. Supposing that X is a discrete-valued attribute having v distinct values, $\{x_1, x_2, x_3, \dots x_v\}$, seen in training set. Let $P(x_j)$ and $P(Y_c)$ be probability of the feature X having value x_j and probability of a random sample belonging to class Y_c. $P(Y_c|x_j)$ is the probability of a random sample whose feature X has the value x_j and belongs to class Y_c. The Gini Index of an attribute X is defined as

$$Gini(X) = \sum_j P(x_j) \sum_c P(Y_c|x_j)^2 - \sum_c P(Y_c)^2 \quad (2.6)$$

2.3. Similarity Based Feature Selection

2.3.1. Relief

Relief is an average feature weight-based feature selection algorithm developed to solve binary classification problems by Kira and Rendell in 1992(Kira and Rendell, 1992). Supposing m training data and n number of features, for a random sample X, Relief starts by determining the similarity between each sample pairs (X_i, X_j) and selects nearest neighbors in the same class, called nearest-hit, and alternate class, named nearest-miss. Relief applies Euclidean Similarity Measure when choosing relevant pairs. Suppose that the averaged

weight vector w has n dimensions. Relief updates w for each random sample X iteratively by calculating relevance as

$$w_j = w_j - \left(X_{ij} - nearhit_{ij}\right)^2 + \left(X_{ij} - nearmiss_{ij}\right)^2 \quad (2.7)$$

The steps of algorithm:

1. Take m tuples that has n attributes belonging to two classes,
2. Calculate similarity between each sample pairs and pick nearest hit and miss,
3. Compose a n-long zeros weight vector,
4. Update the average weight vector according to formula (2.7)
5. Repeat step 4 for each sample.

2.3.2. Fisher Scoring

The key idea of Fisher Scoring is to find the best feature subset by regarding the differences between alternate classes as large as possible whereas the similarity between samples belonging to same class as close as possible. (Gu, Li and Han,2011). Supposing that D, a set of training tuples, has n samples and m_a^i and σ_a^i be mean and standart deviation of $a-th$ class on $i-th$ feature f^i. Supposing that m^i and σ^i correspond global mean and standart deviation in D for f^i. Fisher Score for $i-th$ feature f^i is computed as

$$fisher\left(f^i\right) = \frac{\sum_{a=1}^{c} n_a\left(m_a^i - m^i\right)^2}{(\sigma^i)^2} \quad (2.8)$$

Where $(\sigma^i)^2$ is calculated as

$$(\sigma^i)^2 = \sum_{a=1}^{c} n_a (\sigma_a^i)^2 \quad (2.9)$$

Since Fisher Scoring regards global mean besides standart deviation within classes, it prevents to drop off the map the effect of a feature f^i that has an importance on the whole dataset.

2.3.3. T Test

T Test is a statistical approach used for determining whether there is a significant difference between average of two classes or not. This type of T Test is also known as Student T Test. T test measures the distinctness ratio between two groups and within groups. Let u_{i1} and u_{i2} be average of two groups, D_1 and D_2, and $S_{iD_1D_2}$ be common standart deviation of groups for $i-th$ feature. T statistic for $i-th$ feature is obtained as

$$t_i = \frac{u_{i1} - u_{i2}}{S_{iD_1D_2} * \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (2.10)$$

$$S_{iD_1D_2} = \sqrt{\frac{(n_1 - 1) * {S_{iD_1}}^2 + (n_2 - 1) * {S_{iD_2}}^2}{n_1 + n_2 - 2}} \quad (2.11)$$

where n_1 and n_2 are number of tuples in two groups (Liu, Peng, Hsieh, Yeh, Lin, Chen, Hou, Shih and Liang, 2013).

2.3.4. Fisher Ratio

Fisher Ratio is employed to compute discriminative level between two classes. Suppose that m_1 and m_2 are mean vectors of two classes, C_1 and C_2 are covariance matrix belonging to features. Fisher ratio is obtained by dividing variance of inter classes into variance of intra classes. Accordingly,

$$f = \frac{\sigma^2{}_{inter}}{\sigma^2{}_{intra}} = \frac{(d.m_1 - d.m_2)^2}{(d^T C_1 d + d^T C_2 d)} = \frac{[d.(m_1 - m_2)]^2}{d^T (C_1 + C_2) d}$$ Type equation here. (2.12)

where d is discriminating power which is acquired as

$$d=(C_1 + C_2)^{-1}(m_1 - m_2) \quad (2.13)$$

High scored features have strongly separation efficiency classes (Dat and Guan, 2007).

2.3.5. Significant Analysis of Microarrays

In the last two decades, with the developments in modern biotechnology, studies on DNA microarrays have been improved thus; bioinformatic datasets consisting of thousands of dimensional gene expressions have begun to be produced. It is important to examine the gene sequences to find out the functioning behind the microarrays produced. Significant Analysis of Microarrays (SAM) has been proposed to measure changes in gene sequences statistically.

SAM calculates meaningful genes by computing a d_j parameter that statistically shows the strength of correlation between a variable j and gene sequence. To understand the effect of related genes on whole expression, SAM permits to check the impact of genes in terms of different classes. d_j statistic is computed as

$$d_j = \frac{\mu_{1j} - \mu_{2j}}{S_j + S_0} \quad (2.14)$$

where μ_{1j} and μ_{2j} are average of $j - th$ feature in two groups and S_j is common standart deviation. S_0 parameter is also added to prevent small variation of gene expression (Liu et al., 2013). S_j is calculated by following formula:

$$S_j = \sqrt{\tau(\sum_{k=1}^{m}(D_{kj} - \mu_{1j})^2 + \sum_{l=1}^{n}(D_{lj} - \mu_{2j})^2)} \quad (2.15)$$

where D_{kj} and D_{lj} are samples belonging to different classes and momentum parameter τ is computed as:

$$\tau = \frac{\frac{1}{m}+\frac{1}{n}}{m+n-2} \quad (2.16)$$

where m and n are number of samples belonging to each class. Since the distribution of data is unknown, the relation between a feature and gene sequence is computed by permutation. The d_j statistic for a specific gene is calculated in each permutation and incremented cumulatively by adding to d_j^p parameter that is calculated by equation 2.14 and d_j^f is calculated by d_j^p is averaged eventually.

$$d_j^f = \frac{\sum_p d_j^p}{epoch(i)} \quad (2.17)$$

The difference between average scattering d_j^f and the d_j gives us the impact of related gene on expression.

2.3.6. ReliefF

ReliefF is an extended model of original Relief. While original Relief scores features according to distance between nearest sample pairs, ReliefF solves multi-classes problems searching k-nearest neighbors from the same class, known as nearest hit H_j and from other classes, called nearest miss M_j. For $n\ x\ k$ dataset, assuming n is number of samples on whole dataset, k is dimension size, S_i is $i-th$ sample, a is number of classes, $w[m]$ is attribute vector that is updated

according to similarities between S_i, H_j and M_j. ReliefF generates a score for each feature by computing the contribution of similarities between closest neighborhoods.

The representation of the ReliefF algorithm:

1. initialize weight vector w[:] :=0.0;
2. **for** i:=1 **to** n **do begin**
3. take i-th sample S_i;
4. find k nearest hits H_j;
5. **for** each class C≠class(S_i) **do**
6. get k nearest misses M_j in class C;
7. **for** m:=1 **to** k **do**
8. $$w[m] := w[m] - \sum_{j=1}^{a} \frac{diff(m,S_i,H_j)}{(n.a)} + \sum_{c \neq class(S_i)} \frac{P(c)}{1-P(class(S_i))} \sum_{j=1}^{a} \frac{diff(m,S_i,M_j(C))}{(n.a)}$$
9. **end**

2.3.7. Rank Product

Rank Product is a biologically justified method for detection of differentially expressed genes in microarray experiments. Contrary to other parametric feature selection methods, rank product is a non-parametric statistical model based on ranks of fold changes (Liu et al., 2013).

The rank product is a statistical method used for two-classes microarray data analysis. Given m genes for c_1 and c_2 independent replicates in sample 1 and sample 2, expression level of gene j in $i-th$ replicate of $n-th$ sample is X_{jin}, where $1 \leq j \leq m$, $1 \leq i \leq c_n$, $1 \leq n \leq 2$. Rank level for each replicate i is $R_{jin} = rank(X_{jin})$,where $1 \leq R_{jin} \leq m$. Rank Product for $j-th$ gene is computed by:

$$RP_j = \frac{(\prod_{i=1}^{c_1} R_{ji1})^{1/c_1}}{(\prod_{i=1}^{c_2} R_{ji2})^{1/c_2}} \quad (2.18)$$

RP_j is ratio of geometric means of ranks of sample 1 and sample 2 for $j-th$ gene (Koziol, 2010).

2.4. Dimension Reduction Methods

2.4.1. Unsupervised Dimension Reduction

2.4.1.1. Principal Component Analysis

PCA is the rotating process of data in $n-$ dimensional space into a new $k < n$ dimensional domain creating the widest variance. Since class labels are not utilized during rotation, PCA is an unsupervised dimension reduction technique. The mathematical background of PCA is based on linear algebra. Let C be an $m\ x\ n$ symmetric, orthogonally diagonalizable matrix, which has merely real eigenvalues and C^T be transpose. There exist real numbers $\alpha_1 \dots \alpha_n$(eigenvalues) and orthogonal, non-zero real vectors $\overrightarrow{w_1} \dots \overrightarrow{w_n}$(eigenvectors) such that for each $i = 1,2, \dots n$ (Jauregui,2012).

$$C\overrightarrow{w_i} = \alpha_i \overrightarrow{w_i} \quad (2.19)$$

If C is a matrix $m\ x\ n$ of real values, $m\ x\ m$ matrix CC^T and $n\ x\ n$ matrix C^TC are both symmetric. CC^T and C^TC have the same non-zero eigenvalues. Suppose that $\vec{w}$ is an eigenvector of C^TC and $\alpha \neq 0$. In this case:

$$(C^TC)\ \vec{w} = \alpha\vec{w}. \quad (2.20)$$

According to following formula $C\vec{w}$ is eigenvector of CC^T.

$$CC^T(C\vec{w}) = \alpha(C\vec{w}). \quad (2.21)$$

Here, $C\vec{w}$ should not be zero vector. For a dataset that has n samples, the mean vectors of all features are computed as

$$\vec{m} = \frac{1}{n}(\overrightarrow{X_1} + \cdots + \overrightarrow{X_n}). \quad (2.22)$$

Supposing that Z is an $m\,x\,m$ matrix. The values belonging to $i-th$ column are $\overrightarrow{X_i} - \vec{m}$ and

$$Z = \left[\overrightarrow{X_1} - \vec{m}\right] \ldots [\overrightarrow{X_n} - \vec{m}]. \quad (2.23)$$

$m\,x\,m$ covariance matrix P is defined as

$$P = \frac{1}{n-1} ZZ^T. \quad (2.24)$$

2.4.1.2. Deep Belief Network

Deep Belief Network (DBN) is a deep architecture constructed by combining a stack of Restricted Boltzmann Machines (RBF). Thus, the more complex features are learned by using features obtained from training of previous layers (Salama, Hassanien and Fahmy, 2011). In other words, inputs are fed from the bottom layer called as visible layer and respectively hidden layers are connected as stacked multiple RBMs.

RBM is an energy-based generative model that defines probability distribution over visible variables with the help of hidden units. Generally, data is not enough to construct desired model therefore, it is hard to observe all desired probability distributions. Thus, it is tried to overcome this problem by both increasing power of the model and defining new unobserved variables with the

help of hidden units. Since the probability distribution over the variables are computed by an energy function, RBM is known as an energy-based model. It is constituted from a set of visible units $V = \{v_i\}$ and hidden units $H = \{h_j\}$, i and j specify nodes respectively in visible and hidden layers. In Figure (2.1) an RBM structure consisting of 3 visible and 2 hidden units is shown:

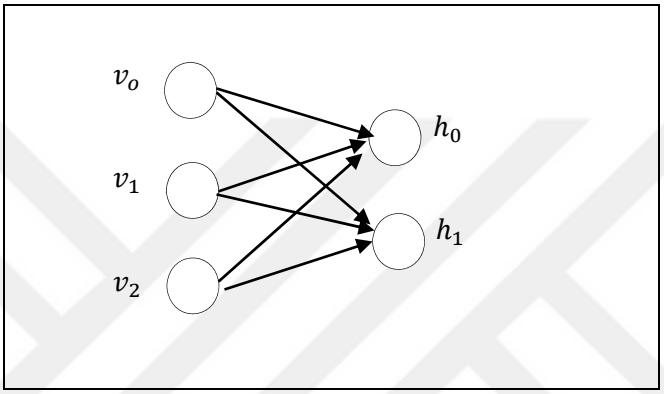


Figure 2.1. Graphical representation of RBM

RBM is a special architecture that restricts connections between hidden-hidden and visible-visible units. The learning steps of RBM are defined as follows:

- By the reason of conditional independence within the same layer, the conditional distributions are:

$$P(H|V) = \prod_j p(h_j|v)$$

$$p(h_j = 1|v) = f(a_i + \sum_i w_{ij} v_i)$$

$$p(h_j = 0|v) = 1 - p(h_j = 1|v) \tag{2.25}$$

Besides,

$$P(V|H) = \prod_i p(v_i|h)$$

$$p(v_i = 1|h) = f(b_j + \sum_j w_{ij} h_j)$$

$$p(v_i = 0|h) = 1 - p(v_i = 1|h) \quad (2.26)$$

The activation function for binary data is sigmoid such that $\frac{1}{1+e^{-z}}$.

- The distribution between visible and hidden units is:

$$P(v,h) = e^{-E(v,h)} / \sum_i e^{-E(v_i,h)}$$

$$E(x,h) = -h^T wv - b^T v - c^T h \quad (2.27)$$

- The average of log probability according to parameters is:

$$\Delta w_{ij} = \varepsilon(\delta log\ p(v)/\delta w_{ij}) = \varepsilon(< x_i h_j >_{data} - < v_i h_j >_{model}) \quad (2.28)$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 >_{model}) \quad (2.29)$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 >_{model}) \quad (2.30)$$

where ε is epsilon parameter having a very small value.

- Since computing $<>_{model}$ term takes exponential time to acquire gradient, Contrastive Divergence (CD) is applied as learning approach. In Contrastive Divergence, sampling is easier by using two basic approaches:

1. Since it is expected that the probability distribution of the model is close to training set, Markov chain can be started from a sample. Thus, it uses a closer distribution to target distribution and sampling chain converges it.
2. CD does not wait to converge Markov chain. Instead, samples that are obtained after only k-steps Gibbs sampling are used. In practice, even 1-step Gibbs sampling may be enough to converge. In this case $<>_\iota$ represents the expectation according to distribution of samples from Gibbs sampler initiated at the data. The new update rules are:

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{data} - < v_i h_j >_\iota) \quad (2.31)$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 >_\iota) \quad (2.32)$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 >_\iota) \quad (2.33)$$

The Harmonium is an RBM with continuous hidden nodes. Where the f is Normal distribution function that takes the form in following formula:

$$p(h_j = h|x) = N(c_j + w_j x, I) \quad (2.34)$$

The basic idea of training DBN is to train a stack of RBMs. The parameters of the model, θ, identify both $p(v|h,\theta)$ and prior distribution on hidden vectors, $p(h|\theta)$ therefore, the probability of visible vector v is:

$$p(v) = \sum_h p(h|\theta)p(v|h,\theta) \quad (2.35)$$

Thereafter learning θ, while $p(v|h,\theta)$ is kept, $p(h|\theta)$ can be changed by a better model that is learned from hidden vector h. (Salama et al., 2011). As it is seen in McAfee's study (McAfee, 2008), when the number of hidden units in the top level exceeds a threshold, the performance of DBN flattens at about certain accuracy. Besides, there against increasing number of layers, performance of network tends to decrease. As the number of iterations increases, it is observed that performance of DBN increases.

2.4.2. Supervised Dimension Reduction

2.4.2.1. Linear Discriminant Analysis

LDA is a statistical method used to find a linear combination of features that characterize two or more classes. The obtained combination can be used for dimension reduction, computational efficiency enhancement and classification problems. This method increases the separability of classes by maximizing the ratio between inter-classes variance to intra-classes variance. Although LDA is closely related to PCA, the main difference between them is that since PCA provides a projection of features by rotating them into a new domain, it does not take differences between classes into account, while LDA considers separability of classes.

Let D be a set of tuples consisting of n samples and m attributes,

$$D = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1m} \\ a_{21} & a_{22} \dots & a_{2m} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} \dots & a_{nm} \end{bmatrix} \tag{2.36}$$

μ_i is mean vector of features belonging to $i-th$ class,

$$\mu_i = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{im} \end{bmatrix} \tag{2.37}$$

S_W, within-class, and S_B, between-class, scatter matrices are calculated to formulate class separability:

$$S_W = \sum_i S_i \tag{2.38}$$

$$S_i = \sum_{a \in D_i} (a - \mu_i)(a - \mu_i)^T \tag{2.39}$$

$$S_B = \sum_i n_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{2.40}$$

where μ is global mean and n_i is number of samples belonging to $i - th$ class. Next step of PCA is solving the generalized eigenvalue problem by using $S_W^{-1} S_B$ to get linear discriminants. An eigenvector of a conversion is a representation of the vector space in which the conversion is applied. Each eigenvalue that is non-equal to zero is linearly independent. If there is a linear dependence between features, this is represented by zero eigenvalue and the eigenvectors corresponding non-zero eigenvalues are selected to obtain a nonrecurring feature set, the ones corresponding to zero eigenvalues are ignored (Balakrishnama and Ganapathiraju, 1995). Supposing that $\alpha_1 \dots \alpha_m$ are eigenvalues and $\overrightarrow{w_1} \dots \overrightarrow{w_m}$ are eigenvectors. In order to decide lower dimensional space, the corresponding eigenvalues of eigenvectors are examined. The eigenvectors with the lowest eigenvalues have less information and they are dropped. The idea is that to create a new $k-$dimensional feature space where $k < m$ by selecting top k eigenvectors that have the highest eigenvalues. Let B be mean corrected data produced by subtracting global mean from original data:

$$B = \begin{bmatrix} a_{11} - \mu & a_{12} - \mu \dots & a_{1m} - \mu \\ a_{21} - \mu & a_{22} - \mu \dots & a_{2m} - \mu \\ \vdots & \vdots & \vdots \\ a_{n1} - \mu & a_{n2} - \mu \dots & a_{nm} - \mu \end{bmatrix} \tag{2.41}$$

By multiplying the vector B with the first k eigenvectors, the original data is transformed in new k-dimensional subspace:

$$D_2 = B \times \vec{w}_{[m-k:m]} \tag{2.42}$$

2.5. Major Contributions

- The proposed method gives unchosen features the chance to show perfect-fit.
- It provides a representation of all features just in case the high-scored features that are marked as relevant in the training phase cannot separate data in testing.
- In addition to selecting the top$-n$ features in datasets where the features cannot be separated sufficiently, it increases the variance by providing m-dimensional linear transformation.

2.6. Studies on Tested Datasets

In this thesis Arcene, Gisette, Gene Expression Cancer RNA-Seq, Colon, ALLAML, Leukemia, Prostate_GE and Madelon datasets are used. Leukemia and Prostate_GE microarray datasets were used in a study conducted by Canedo, Marono and Betanzos, Benitez and Herrera (Canedo et al, 2014). In these studies, dataset shifting problems have been emphasized. On the datasets where training and testing data show different distributions, a feature marked as important in the training phase is not informative in the testing stage. In addition, while the class distributions of training data are balanced on these datasets, there is inequality in testing.

For example, on the Prostate_GE dataset, training and testing data were obtained from different experiments. While the class distributions of data show a balance in training, there is a big difference in the testing. Since Prostate_GE dataset suffers from dataset shifting problem, classifiers such as C4.5 and Naive

Bayes could not solve this problem and found the data belong to major class. In microarray classification, it is emphasized that small samples are affected by error prediction and therefore the importance of validation technique in these studies.

Arcene, Gisette and Madelon datasets were used in a feature selection challenge in 2003(Guyon, Gunn, Hur, Dror, 2003). Using Gaussian kernel on these datasets is said to cause more successful results than linear kernel in these experiments.

3. MATERIALS AND METHODS

This section describes utilized datasets named Arcene, Gisette and Gene Expression Cancer RNA-Seq, classification methods such as Multi Layer Perceptron, Naïve Bayes, Linear Discriminant Analysis, Support Vector Machine, clustering approaches AGNES, one of hierarchical clustering methods and K-Means that belongs to partitioning clustering, and proposed dimension reduction-based hybrid feature selection algorithm.

3.1. Materials

The datasets used in this thesis comprise Arcene, Gisette and Gene Expression Cancer RNA-Seq that have large-scale feature sets. Utilized datasets were chosen to represent low, medium and large scales. All the datasets were provided from web-based dataset repositories. While selecting materials, curse of dimensionality problem caused by large scale feature set size was the focused issue.

3.1.1. Arcene Dataset

Arcene dataset was formed from two origins: The National Cancer Institute (NCI) and the Eastern Virginia Medical School (EVMS). This version of dataset was arranged by combining different sources and used as one of five materials for NIPS 2003 feature selection challenge (Guyon et al., 2003). The data consist of normal patterns versus patients who have cancer. Although proposed version of the dataset has 900 instances, due to data loss caused from dataset repository we used 200 of them. Compared to the number of samples, feature set includes 10000 attributes which considered as large dimension space.

3.1.2. Gisette Dataset

Gisette is binary handwriting discernment problem for separating 4 and 9 digits. This dataset was also used for NIPS 2003 feature selection challenge

(Guyon et al., 2003). Original data were produced from MNIST data which was created by Yann LeCunn and Corinna Cortes; and this version of the dataset was prepared for the NIPS 2003 for feature selection. While constructing feature set, randomly selected feature subsets were added to original normalized pixels and all the zero valued features were removed. Obtained dataset consists of 5000 features and 13500 samples that are separated as training, testing and validating sets. Due to the absence of testing labels, we used 7000 of them.

3.1.3. Gene Expression Cancer RNA-Seq

This dataset was produced from RNA-Seq PANCAN dataset for multiclass classification problems (The Cancer Genome Atlas Research Network, Weinstein, Collisson, Mills, Shaw, Ozenberger, Ellrott, Shmulevich, Sander, Stuart, 2013). It contains the information about five different tumor types: BRCA, KIRC, COAD, LUAD and PRAD. The dataset covers 801 instances and 20531 features.

3.2. Methods

3.2.1. Classifiers

3.2.1.1. Multi Layer Perceptron

Artificial neural networks are mathematical simulation methods, modelling human brain behaviors and information processing techniques. The purpose of artificial neural networks is to understand the working principle of brain and create self-learning, adaptable decision machines by simulating it. In comparison with the computers, human brain has some superiorities in terms of speech, recognition and understanding. Understanding the mechanism behind these capabilities enables us to perform efficient engineering works by imitating them in non-organic environment.

In many aspects human brain is different from computers. Human brain has about 10^{11} neurons and each of them has approximate 10^{4}

connections(synapses). This is the high interconnection between neurons that make the brain powerful.

In an artificial neural network, all the axon branches (output of a neuron) are produced sequentially while human brain is distributed. A feedforward neural network uses an output of a neuron as input of next and updates weights(dendrites) by backpropagating total error from the output to input units. Organic neural networks are scattered to work in a parallel and non-synchronous manner.

In biological neurons, the firing duration and conduction speed of signals may vary depending on age, gender, disease and environmental conditions. Synaptic transmissions are occurred by some neurotransmitter materials. Instead, artificial neural networks use floating point weight value and signal speed depends on optimization of feedforward and backpropagation with hardware capacity of computer.

Biological neurons have fault tolerance and although they may store many unnecessary information, they are not affected from this. Artificial neural networks are not successful as organic neurons in terms of self-renewal and fault tolerance since not to be a matter of fatigue.

Brain yards can expand, change, associate a neuron to another neuron; function of synapses may change. An information stored in brain may be sharpen by learning. All the mechanism behind of these behaviors is not still known properly. Learning is only occurred by changing weight values between neurons. Weights are updated until the differences between desired and produced outputs by the network are minimized. However, it is not correct to assume that updating process always minimizes total error. We can imagine it as a hill when we climb a mountain. A downhill slope may cause us to perceive it as peak of the mountain. It means stucking weights to local minimum. The step size of the climb is known as learning rate. It is important to find optimum learning rate here. A very large learning rate causes jump whereas small one leads longer training duration.

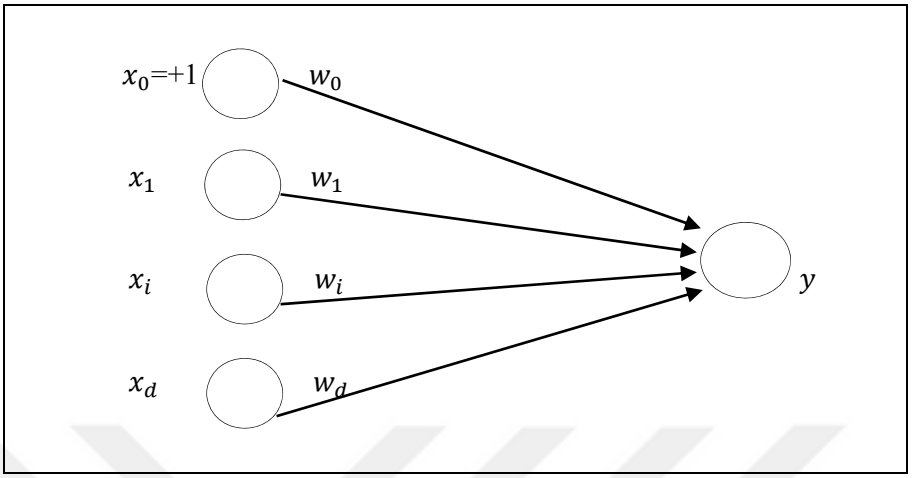


Figure 3.1 Perceptron

In Figure 3.1 x_i, where $i = 1, \dots d$, represents d −dimensional input units and x_0 is extra input taken 1 value. $w_i \dots where\ i = 1 \dots d$, is weight carrying the information of input x_i.

Perceptron is the smallest part of an artificial neural network. It takes information from external world or output of other neurons as input. For each x_i , for $i = 1, \dots d$ a w_i connection parameter is defined. In the simplest way, output y is equal to weighted sum of inputs.

$$y = \sum_{i=1}^{d} w_i\, x_i + w_0 \tag{3.1}$$

w_0 is bias used to improve output value and specifies the importance of x_0 extra input unit. The perceptron defined in equation 3.1 indicates a multi-input plane and can be used to divide the space into two parts. When used for a linear separation operation, a threshold $e(.)$ is defined as follows:

$$e(a) = \begin{cases} 1, & a > 0 \\ 0, & otherwise \end{cases} \tag{3.2}$$

$$select \begin{cases} L_1, & e(w^T x) > 0 \\ L_2, & otherwise \end{cases} \quad (3.3)$$

where L_1 and L_2 specify class labels. If a finite probability is calculated, we can use sigmoid activation function $\delta(o)$ as:

$$o = W^T x \quad (3.4)$$

$$y = \delta(o) = \frac{1}{1+\exp(-w^T x)} \quad (3.5)$$

W is weight vector representing all the weights between x and y.

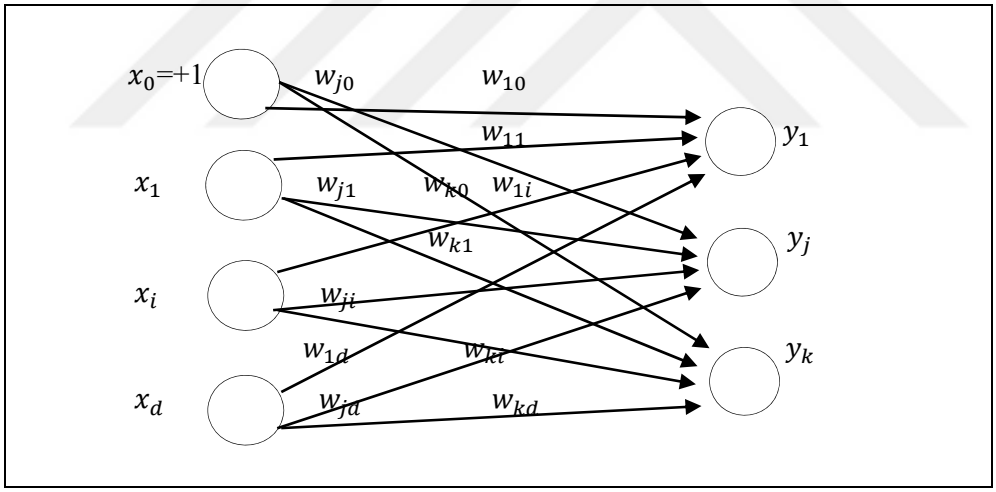


Figure 3.2 Multi-Output Perceptron

In neural networks having more than 2 outputs, each perceptron has its own w_j weight parameter.

$$y_j = \sum_{i=1}^{d} w_{ji} x_i + w_{j0} \quad (3.6)$$

where w_{ji} is weight parameter between x_i and y_j. For the multiclass classification problems, neural network structure has more than one perceptron, the output is determined by selecting the highest output value. In an artificial neural network that we need finite probabilities, outputs can be calculated as:

$$o_j = W_j{}^T x \tag{3.7}$$

$$y_j = \frac{\exp(o_j)}{\sum_{c=1}^{L} \exp(o_c)} \tag{3.8}$$

In the training of neural networks, generally online learning is preferred. Samples are sequentially given to neural network and parameters are updated in order to fit data. Starting with randomly generated weights, the error for each sample is tried to be minimized. For a sample $x^a, a = 1 \dots n, n$ is number of samples, the error is calculated as

$$E^a(W|x^a, o^a) = \frac{1}{2}(o^a - y^a)^2 = \frac{1}{2}(o^a - (W^T x^a))^2 \tag{3.9}$$

where o^a is desired output for sample x^a. The error on sample pair (x^a, o^a) is decreased by using online gradient descent :

$$\Delta w_i = \sigma(o^a - y^a)x_i^a \; i = 0, \dots, d. \tag{3.10}$$

where σ is learning rate. In a two-class separation, a single output can be computed by an δ function:

$$y^a = \delta(W^T x^a) \tag{3.11}$$

In that case, cost function is calculated as

$$E^a(W|x^a, o^a) = -o^a \log y^a - (1 - o^a)\log(1 - y^a) \quad (3.12)$$

If the number of classes is more than 2, for the sample (x^a, o^a), if $x^a \in L_j$ $o_j^a = 1$, else $o_j^a = 0$ is computed as:

$$y_j^a = \frac{\exp\left(w_j^T x^a\right)}{\sum_c \exp(w_c^T x^a)} \quad (3.13)$$

Cost function is defined as

$$E^a\left(\{W_j\}_j \middle| x^a, o^a\right) = -\sum_j o_j^a \log y_j^a \quad (3.14)$$

Update rule calculated by gradient descent is

$$\Delta w_{ji}^a = \sigma\left(o_j^a - y_j^a\right) x_i^a, i = 0, \dots, d, j = 1, \dots, k \quad (3.15)$$

If the output of neural network is smaller than desired output and input value is positive, the update magnitude will be positive, however if the input value is negative, the update magnitude will be negative. Therefore, output value is increased, and it approaches desired output. In the exact opposite situation, the error is minimized by decreasing actual output.

When the input is very small, its effect on updating will be low. As the input increases, its effect also increases.

Another parameter affecting the update magnitude is learning rate (σ). If the σ is selected very large, the training is shortened, and impact of the last samples increases. Otherwise, training takes long.

Logic operations are two-class applications to distinguish between true and false. Table 3.1 and 3.2 shows samplings for AND, OR gates.

Table 3.1 AND Gate

x_1	x_2	o
0	0	0
0	1	0
1	0	0
1	1	1

We define discriminant y as

$$y = e(x_1 + x_2 - 1.5) \tag{3.16}$$

where $x = [1, x_1, x_2]$ and $w = [-1.5,1,1]^T$ are extended input and weights. When we apply y to Table 3.1, it actualizes all the conditions. Likewise, $y = e(x_1 + x_2 - 0.5)$ equation is valid for OR gate in Table 3.2.

Table 3.2. OR Gate

x_1	x_2	o
0	0	0
0	1	1
1	0	1
1	1	1

Despite the fact that AND, OR logical operations are linearly separable, any w parameter for XOR problem cannot be defined.

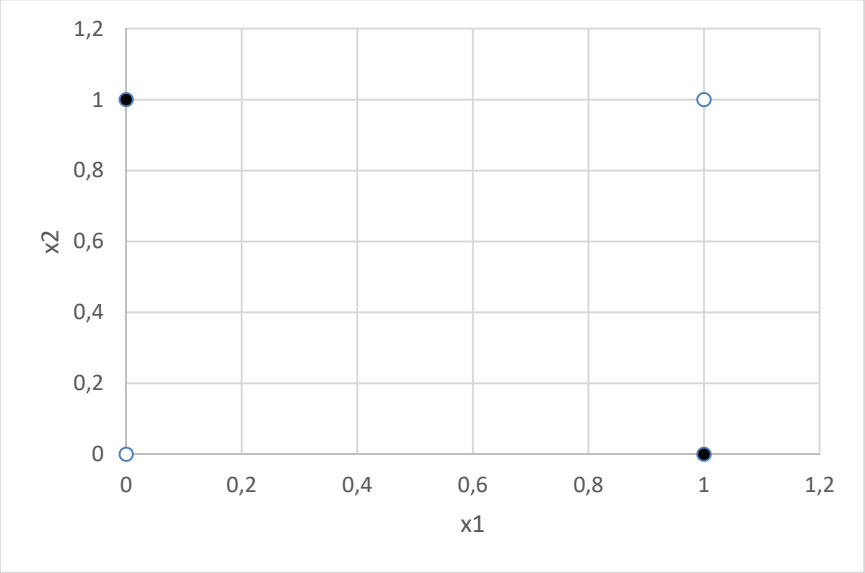


Figure 3.3 XOR Gate in Coordinate System

The single-layer perceptron can only learn linear functions however cannot perform operations such as XOR. A multi-layer architecture is required to solve non-linear problems. Input x is given to input layer by adding additional input unit $x_o = +1$ and hidden units are calculated showing by h_z. Each hidden unit is also a perceptron and applies δ operation to weighted sums.

$$h_z = \delta(W_z^T x) = \frac{1}{1+\exp[-(\sum_{i=1}^{d} w_{zi}x_i + w_{i0})]}, z = 1,,,Z \quad (3.17)$$

$$y_j = v_j^T h = \sum_{z=1}^{Z} v_{jz}h_z + v_{j0} \quad (3.18)$$

The hidden layer also has an additional unit that takes +1 and v_{j0} shows the weights leading to outputs. Since input units evaluate external information, no calculation is made here. Hence, this type of structure is known as a two-layered perceptron. In a two-classes structure, there is only one output that uses δ operation, whereas in the networks that have more than two classes, the number of output units equals to the number of classes. Using δ function in hidden layers is important in order to propagate derivative of the error. Otherwise, it would pass incoming information to output layer linearly. Instead of using $sigmoid$ function $tanh$ can also be used. This means that output range will be -1 and +1 instead of 0 and 1.

In the more complex architectures, multiple hidden layers can be used. However, they have some difficulties in terms of computational cost.

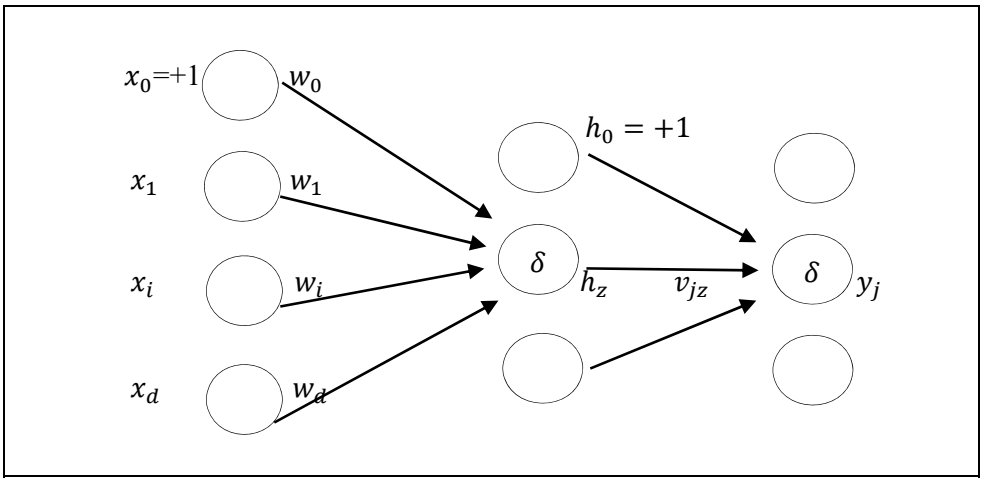


Figure 3.4 Multi Layer Perceptron

Training of multi-layer perceptron is like training of perceptron, the only difference is that output is non-linear function of input due to used activation function in hidden units. Since hidden units are inputs of output layer, we can backpropagate the error by using an chain rule over v_{jz} and w_{zi}:

$$\frac{\partial E}{\partial w_{zi}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial h_z}\frac{\partial h_z}{\partial w_{zi}} \tag{3.19}$$

Since the error is propagated through layers, this algorithm is called error backpropogation.

To give an example of a single-output non-linear regression:

$$y^a = \sum_{z=1}^{Z} v_z\, h_z^a + v_0 \tag{3.20}$$

h_z is calculated by using equation (3.17). In this regard, total error for all the samples is computed as:

$$E(W, v|X) = \frac{1}{2}\sum_a (o^a - y^a)^2 \tag{3.21}$$

When the hidden units are accepted input, the weights between hidden layer and output are updated similar to equation (3.15).

$$\Delta v_z = \sigma \sum_a (o^a - y^a)\, h_z^a \tag{3.22}$$

The outputs of the first layer consist of perceptron, thus their values are not known. Therefore, weights are updated by the chain rule instead of using the least squares:

$$\Delta w_{zi} = -\sigma \frac{\partial E}{\partial w_{zi}} \tag{3.23}$$

$$= -\sigma \sum_a \frac{\partial E^a}{\partial y^a} \frac{\partial y^a}{\partial h_z^a} \frac{\partial h_z^a}{\partial w_{zi}} \tag{3.24}$$

$$= -\sigma \sum_a -(o^a - y^a)\ v_z\ h_z^a(1 - h_z^a) x_i^a \tag{3.25}$$

where $(o^a - y^a)\ v_z$ term is amount of error belonging to $z - th$ hidden unit. $(o^a - y^a)$ shows output error and v_z carries responsibility of $z - th$ hidden unit on total error. $h_z^a(1 - h_z^a)$ expression is derivative of activation function δ and x_i^a is $i - th$ input unit belonging to $a - th$ sample. The most sensitive spot here is the effect of v_z values on the weights leading to hidden layer. The weights of both layers are updated but the previous value of v_z are used to calculate the current value of w_{zi}. This is because the calculated error is under the responsibility of previous weight.

In an online learning, the parameters are updated after each sample. Once all the samples are over, an epoch is completed. As an update will be made for each sample, it is reasonable to select learning rate as small and give the samples in order.

When multiple regressions are required by using same samples, more than one output is needed:

$$y_j^a = \sum_{z=1}^{Z} v_{jz} h_z^a + v_{j0} \tag{3.26}$$

Square errors of all the outputs are picked up:

$$E(W, V|X) = \frac{1}{2} \sum_a \sum_j (o_j^a - y_j^a)^2 \tag{3.27}$$

Updating rules are also derived by gradient descent:

$$\Delta v_{jz} = \sigma \sum_a \left(o_j^a - y_j^a\right) h_z^a \tag{3.28}$$

$$\Delta w_{zi} = \sigma \sum_a [\sum_j \left(o_j^a - y_j^a\right) v_{jz}] h_z^a (1 - h_z^a) x_i^a \tag{3.29}$$

where $\sum_j \left(o_j^a - y_j^a\right) v_{jz}$ is the total error that is carried to $z - th$ hidden unit via all the weights.

We have said that the only one output is sufficient to separate two classes:

$$y^a = \delta(\sum_z^Z v_z h_z^a + v_0) \tag{3.30}$$

Backpropagation algorithm for a multi layer perceptron that has L output(Alpaydın,2017) :

Begin w_{zi} and v_{jz} randomly

Repeat

For all $(x^a, o^a) \in X$ randomly order

For all $z = 1, \dots, Z$

$h_z = \delta(w_z^T x^a)$

For all $j = 1, \dots, L$

$y_j = v_j^T h$

For all $j = 1, \dots, L$

$\Delta v_j = \sigma\left(o_j^a - y_j^a\right) h$

For all $z = 1, \dots, Z$

$\Delta w_z = \sigma(\sum_j \left(o_j^a - y_j^a\right) v_{jz}) h_z (1 - h_z) x^a$

For all $j = 1, \dots, L$

$v_j = v_j + v_j$

For all $z = 1, \dots, Z$

$$w_z = w_z + \Delta w_z$$

End

The y^a value for a classification of two classes is a prediction for the probability $P(L_1|X)$; in a similar way, $P(L_2|X)$ corresponds to $1 - y^a$. Cross entropy of error used for this situation is:

$$E(W, v|X) = -\sum_a o^a \log y^a + (1 - o^a) \log(1 - y^a) \quad (3.31)$$

Derived updating rules are as follows:

$$\Delta v_z = \sigma \sum_a (o^a - y^a)\, h_z^a \quad (3.32)$$

$$\Delta w_{zi} = \sigma \sum_a (o^a - y^a)\, v_z h_z^a (1 - h_z^a) x_i^a \quad (3.33)$$

Like in a single layer perceptron, for a classification and regression updating rules are the same.

In case of $L > 2$ classes L outputs are defined :

$$u_j^a = \sum_z^Z v_{jz} h_z^a + v_{j0} \quad (3.34)$$

It can be defined dependency between classes by using softmax function:

$$y_j^a = \frac{\exp(u_j^a)}{\sum_{c=1}^{L} exp(u_c^a)} \quad (3.35)$$

Let us mention that y_j indicates probability $P(L_j|x^a)$. Via the error function:

$$E(W, V|X) = -\sum_a \sum_j o_j^a log y_j^a \quad (3.36)$$

We can derivate updating rules by gradient descent:

$$\Delta v_{jz} = \sigma \sum_a (o_j^a - y_j^a)\, h_z^a \tag{3.37}$$

$$\Delta w_{zi} = \sigma \sum_a \left[\sum_j \left(o_j^a - y_j^a\right) v_{jz}\right] h_z^a (1 - h_z^a) x_i^a \tag{3.38}$$

We can also create a multi-layer perceptron that has multi hidden layers which have their own weights and use δ activation function. Below is a multi layer perceptron consisting of two hidden layers:

$$h_{1z} = \delta(w_{1z}^T x)$$

$$= \delta\left(\sum_{i=1}^{d} w_{1zi} x_i + w_{1z0}\right), z = 1, \dots, Z_1 \tag{3.39}$$

$$h_{2m} = sigmoid(w_{2m}^T h_1)$$

$$= \delta\left(\sum_{z=0}^{Z_1} w_{2mz} h_{mz} + w_{2m0}\right), m = 1, \dots, Z_2 \tag{3.40}$$

$$y = v^T h_2 = \sum_{m=1}^{Z_2} v_m h_{2m} + v_0 \tag{3.41}$$

Here, w_{1z} and w_{2m}, respectively specify weights of first and second layer, h_{1z} and h_{2m} show hidden units belonging to first and second hidden layers and v is weight vector of third layer. We can use chain rule mentioned above to train network that has multi hidden layers.

Gradient descent has some advantages such as simplicity, easy applicability. An updating weight depends on only previous and next units. In an online learning, since samples are not required to store, when the model changes, if enough time is given, it can adapt itself to new model. There are two simple

method to improve success of gradient descent: momentum and adaptable learning rate.

Let w_j be a weight of multi-layer perceptron. The successive Δw_j^a value at time a may be very different and lead to oscillation in training. To prevent this, a momentum parameter that takes into account previous update values is added to learning:

$$\Delta w_j^a = -\sigma \frac{\partial E^a}{\partial w_j} + \alpha \Delta w_j^{a-1} \tag{3.42}$$

It is logical to select momentum parameter between 0.5 and 1.0. In an online learning, oscillation is prevented by averaging.

Learning rate determines the amount of slope in gradient descent. We can update this parameter for a quick convergence:

$$\Delta\tau = \begin{cases} +q & if\ E^{a+\rho} < E^a \\ -r\sigma & Otherwise \end{cases} \tag{3.43}$$

A multi-layer perceptron that has $d-$inputs, $z-$hidden units and $L-$outputs has $Z(d+1)$ and $L(z+1)$ weights respectively in the first and second layers thus, the time complexity is $O(z.(L+d))$. Supposing that epoch number is e, in that case learning time complexity becomes $O(e.z.(L+d))$. Since d and L are static, learning and time complexity is adapted by changing number of hidden units (Alpaydın, 2017).

3.2.1.2. Support Vector Machine

SVM is a discriminator-based machine learning technique that uses edge margin as a criterion for separating linear and non-linear classes in practice (Çoban, Özyıldırım and Özel, 2018). The main principle of SVM is that while

solving a problem trying to avoid more complex trouble as a fore-step (Vapnik, 2000). To find out the discriminant in a classification problem rather than estimating $p(x|C_t)$ or $p(C_t|x)$ probabilities sensitively, finding x values that provides $p(C_t|x) = p(C_r|x)$ boundary conditions is sufficient. After learning is finished, the weigth vector that is parameter of linear model, is written using a subset of learning set called support vectors. For classification, these are samples that are closest to bound since they provide information about the boundary that seperates two classes. In linear problems, the objective is to find the hyperplane that has maximum margin. For non-linear problems, this is done by transforming data to higher dimensional space by a kernel function.

We start assuming linear decision rules. Let $-1/+1$ be class labels: In the sample $X = \{x^i, o^i\}$, if $x^i \in C_1$, $o^i = +1$ and if $x^i \in C_2$, $o^i = -1$. We define the separating hyperplane as following:

$$w^T x^i + w_0 \geq +1 \quad if\ o^i = +1 \tag{3.44}$$

$$w^T x^i + w_0 \leq -1 \quad if\ o^i = -1 \tag{3.45}$$

If we write a unified equation for these unequlities:

$$o^i(w^T x^i + w_0) \geq +1 \tag{3.46}$$

It is called a hyperplane:

$$w^T x^i + w_0 = 0 \tag{3.47}$$

For a good generalization, samples should not only remain on the right side of hyperplane but also be a certain distance. The distance between nearest samples

on both sides of the border is known as margin and the aim is to make them as large as possible. The distance of the samples on both side to hyperplane is:

$$\frac{o^i(w^T x^i + w_0)}{||w||} \tag{3.48}$$

Supposing that this value is at least ρ:

$$\frac{o^i(w^T x^i + w_0)}{||w||} \geq \rho, \forall i \tag{3.49}$$

To determine this hyperplane, it has to be solved following quadratic problem (Vapnik,2000):

$$min \frac{1}{2}||w||^2, constraint: w^T x^i + w_0 \geq +1, \forall i \tag{3.50}$$

The solution to this optimization problem can be written in unrestricted form using Lagrange functional:

$$L_p = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha^i \left[o^i\left(w^T x^i + w_o\right) - 1\right]$$

$$= \frac{1}{2}||w||^2 - \sum_i \alpha^i \, o^i\left(w^T x^i + w_o\right) + \sum_i \alpha^i \tag{3.51}$$

where α^i are Lagrange multipliers. We try to minimize point in reference to w and w_0, and maximize in reference to $\alpha^i \geq 0$. This convex is a quadratic optimization problem since both base term and linear constraints are convex. Thus, this dual problem can be solved by using Karush-Kuhn Tucker conditions:

$$\frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_i \alpha^i o^i x^i \tag{3.52}$$

$$\frac{\partial L_p}{\partial w_0} = 0 \rightarrow \sum_i a^i o^i = 0 \tag{3.53}$$

Replacing them to equation (3.51):

$$L_d = \frac{1}{2}(w^T w) - w^T \sum_i \alpha^i o^i x^i - w_0 \sum_i \alpha^i o^i + \sum_i \alpha^i$$

$$= -\frac{1}{2}(w^T w) + \sum_i \alpha^i$$

$$= -\frac{1}{2}\sum_i \sum_s \alpha^i \alpha^s o^i o^s (x^i)^T x^s + \sum_i \alpha^i \tag{3.54}$$

While expanding this function with respect to a^i:

$$\sum_i \alpha^i o^i = 0, and\ a^i \geq 0, \forall i \tag{3.55}$$

When it is solved by quadratic optimization it is seen that most of the a^i values are 0. x^i samples that provide $a^i \geq 0$ are called support vectors. As equation (3.52), w can be written the sum of support vectors:

$$w = \sum_{support\ vectors} \alpha^i o^i x^i, a^i \geq 0 \tag{3.56}$$

x^i data points that have $\alpha^i = 0$ the condition $o^i(w^T x^i + w_0)$ is provided and as these are far enough to hyperplane, they have no effect on learning. Even any subset of these samples is removed from learning set, they do not affect the result.

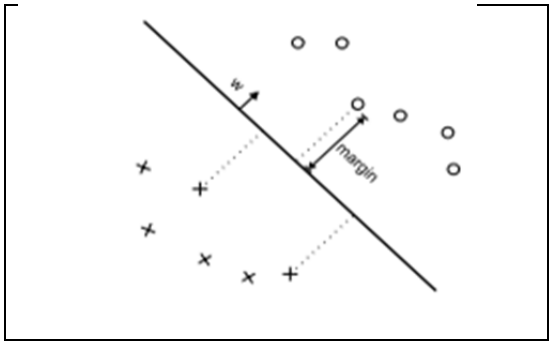


Figure 3.5 Simple Linear Support Vector Machine (Tong, Koller, 2001)

In case of the data cannot be seperable linearly, there is no hyperplane to seperate perfectly. Then, we call the last error maker. To that end, the residual variables $\varepsilon^i \geq 0$ is defined. There may be two types of deflections: since the sample is on wrong side, it is classified wrong or even it is on right side, but as it is not far enough, stays in margin. Let us recompose Equation (3.46) by adding ε^i:

$$o^i\left(w^T x^i + w_0\right) \geq 1 - \varepsilon^i \tag{3.57}$$

If $\varepsilon^i = 0$ there is no problem with x^i. If $0 < \varepsilon^i < 1$, x^i is classified correctly but it is in margin. If $\varepsilon^i \geq 1$, x^i is neither classified correctly and on the right side. In the case of $\{\varepsilon^i > 1\}$, number of misclassified data:

$$\sum_i \varepsilon^i \tag{3.58}$$

Soft margin hyperplane is specified by vector w which minimizes the functional :

$$L_p = \frac{1}{2}||w||^2 + C \sum_i \varepsilon^i \tag{3.59}$$

where C is constraint parameter that spicifies the effect of fault in complexity. Here, the fault is misclassified data points. Let us rewrite equation (3.51) by adding new constraints:

$$L_p = \frac{1}{2}||w||^2 + C\sum_i \varepsilon^i$$

$$-\sum_i \alpha^i\left[o^i\left(w^T x^i + w_0\right) - 1 + \varepsilon^i\right] - \sum_i \mu^i \varepsilon^i \tag{3.60}$$

where μ^i is new Lagrange parameter that is added to be provide ε^i is nonnegative. Let us differentiate:

$$\frac{\partial L_p}{\partial w} = w - \sum_i \alpha^i o^i x^i = 0 \rightarrow w = \sum_i \alpha^i o^i x^i \tag{3.61}$$

$$\frac{\partial L_p}{\partial w_0} = \sum_i \alpha^i o^i = 0 \tag{3.62}$$

$$\frac{\partial L_p}{\partial \varepsilon^i} = C - \alpha^i - \mu^i = 0 \tag{3.63}$$

Since $\mu^i > 0$ the last row is required $0 \leq \alpha^i \leq C$ condition. The parameters that maximize the same quadratic form accourding to α^i :

$$L_d = \sum_i \alpha^i - \frac{1}{2}\sum_i \sum_s \alpha^i \alpha^s o^i o^s (x^i)^T x^s \tag{3.64}$$

under the constraints:

$$\sum_i \alpha^i o^i = 0 \; and \; 0 \leq \alpha^i \leq C, \forall i \tag{3.65}$$

When the equation is solved, $a^i = 0$ for the samples that are on the right side of boundary and far enough such in case linear seperable. Samples where at $a^i > 0$ are support vectors and w defines weight vector. Through them, ones that are at where $a^i < C$ are on the margin and they are used to compute w_0. They also provide $o^i(w^T x^i + w_0) = 1$ and $\varepsilon^i = 0$. $\alpha^i = C$ for the samples that stay in margin or misclassified.

If support vectors were not be in learning set, they would either be misclassified or it would no be a safe classification even if they were classified correctly since they would be so close to border. It can be said number of support vectors can define a upper bound for expected error:

$$E_{N[P(error)] \leq \frac{E_{N[\#\ suport\ vectors]}}{N}} \tag{3.66}$$

where $E_{N[.]}$ shows expected error for N-sized learning set. Here, the most remarkable point is that error depends on number of support vectors rather than input size.

There is one more definition that uses $v \in [0,1]$ parameter instead of C. The aim function:

$$min \frac{1}{2}||w||^2 - v\rho + \frac{1}{N}\sum_i \varepsilon^i \tag{3.67}$$

under the constraints:

$$o^i(w^T x^i + w_0) \geq \rho - \varepsilon^i, \varepsilon^i \geq 0, \rho \geq 0 \tag{3.68}$$

The new parameter ρ scales the margin: margin equals to $2\rho/||w||$. It is shown that v is a lower bound for the ratio of support vector; and is a upper bound for the ratio of samples that generate soft margin error ($\sum_i \#\{\varepsilon^i > 0\}$). The dual:

$$L_d = -\frac{1}{2}\sum_i \sum_s \alpha^i \alpha^s o^i o^s (x^i)^T x^s \quad (3.69)$$

under the constraints:

$$\sum_i \alpha^i o^i = 0,\ 0 \le \alpha^i \le \frac{1}{N}, \sum_i a^i \le v \quad (3.70)$$

As stated above, if the problem is not linear, a linear model can be learned by moving to a new space by means of non-linear basis functions. The linear model in the new space corresponds to non-linear model in the original space(Alpaydın,2017). Supposing that new dimensions are defined by basis functions:

$$z = \varphi(x)\ such\ that\ z_j = \varphi_j(x), j = 1 \dots, k \quad (3.71)$$

So that it is passed from d −dimensional x space to a new k −dimensional z space. The discriminant is defined in new space as follows:

$$g(z) = w^T z$$

$$g(x) = w^T \varphi(x)$$

$$= \sum_{j=1}^{k} w_j \varphi_j(x) \quad (3.72)$$

Generally, k value is bigger than d or N and thus, it is preferred to use a dual definition that depends its complexity on N. The optimization problem is the same with equation (3.59). The only difference is that it is defined in the new space:

$$o^i w^T \varphi(x^i) \geq 1 - \varepsilon^i \tag{3.73}$$

Let's add the Lagrange terms:

$$L_p = \frac{1}{2}||w||^2 + C\sum_i \varepsilon^i - \sum_i \alpha^i \left[o^i w^T \varphi(x^i) - 1 + \varepsilon^i\right] - \sum_i \mu^i \varepsilon^i \tag{3.74}$$

If we take derivative according to parameters and equal to zero:

$$\frac{\partial L_p}{\partial w} = w = \sum_i \alpha^i o^i \varphi(x^i) \tag{3.75}$$

$$\frac{\partial L_p}{\partial \varepsilon^i} = C - \alpha^i - \mu^i = 0 \tag{3.76}$$

In this case the dual is:

$$L_d = \sum_i \alpha^i - \frac{1}{2}\sum_i \sum_s \alpha^i \alpha^s o^i o^s \varphi(x^i)^T \varphi(x^s) \tag{3.77}$$

under the constraints:

$$\sum_i \alpha^i o^i = 0 \; and \; 0 \leq \alpha^i \leq C, \forall i \tag{3.78}$$

The main idea in support vector machine is to read $\varphi(x^i)^T \varphi(x^s)$ dot product as a kernel function, $K(x^i, x^s)$, on the first inputs. In other words, instead of firstly mapping x^i, x^s inputs in z space and then applying dot product, kernel function is applied directly to first form in the original space:

$$L_d = \sum_i \alpha^i - \frac{1}{2}\sum_i \sum_s \alpha^i \alpha^s o^i o^s K(x^i, x^s) \tag{3.79}$$

Most frequently used general-purpose kernel functions are:

Polynomial kernel: for degree q,

$$K(x^i, x) = (x^T x^i + 1)^q \tag{3.80}$$

Here, q must be defined priorly, for instance in the case of $q = 2$ and $d = 2$, following kernel

$$K(x, y) = (x^T y + 1)^2$$

$$= (x_1 y_1 + x_2 y_2 + 1)^2$$

$$= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 +$$

$$x_1{}^2 y_1{}^2 + x_2{}^2 y_2{}^2 \tag{3.81}$$

where $y = w^T x + w_0$, corresponds the dot product of following basis function:

$$\varphi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1{}^2, x_2{}^2]^T \tag{3.82}$$

Gaussian kernel:

$$K(x^i, x) = \exp(-\frac{||x^i - x||^2}{2s^2}) \tag{3.83}$$

It defines a spheric kernel; x^i centre specifies radius in predetermined s.

Sigmoidal kernel:

$$K(x^i, x) = \tanh(2x^T x^i + 1) \quad (3.84)$$

tanh(.), is sigmoidal like sigmoid(.) but its boundary is between -1 and +1.

If there are $K > 2$ classes the way is to define K binary class problems and train $g_t(x), t = 1, \dots, K$ support vector machines that seperate each class from others. In other words, during training of $g_t(x)$, samples belonging to C_t are labelled as +1 and all the other samples $C_k, k \neq t$ are labelled as -1. After all $g_t(x)$ values are calculated the highest one is selected. Platt suggested to transform of output of two classes SVM to posterior probability by sigmoidal model (Platt,1999). In a similar way, posterior probabilities can be estimated after $K > 2$ outputs of SVM are trained to reduce cross entropy by a linear softmax model (Mayoraz and Alpaydın, 1999):

$$y_t = \sum_{j=1}^{K} v_{tj} f_j(x) + v_{t0} \quad (3.85)$$

Here, $f_j(x)$ specifies output of SVM and y_t is posterior probability. v_{tj} weights are trained to reduce cross entropy. Instead of training K binary class SVM, it may also be trained $K(K-1)/2$ binary classifiers. Ech $g_{tj}(x)$ discriminant is trained to give +1 for data belonging to C_t class and -1 for data belonging to C_j class.

3.2.1.3. Naïve Bayes

The Naïve Bayes classifier is a probabilistic based algorithm that is used to calculate a set of probabilities by numbering the combinations and frequency of values in a dataset (Patil and Sherekar, 2013). The algorithm is on Bayes theorem basis thus, assumes all the features are independent for given value of a class variable.

Supposing that T is an $n-$dimensional training set and each data point is represented a vector, $X = \{x_1, x_2, \dots, x_n\}$ taking $A_1, A_2, \dots, A_n$ values. There are k classes C_1, C_2, C_k. For a given sample X, the classifier predicts the class that has highest probability on sample X:

$$\mathrm{P(Ci\ |X)\ >\ P(Cj\ |X)\ for\ 1\ \leq\ j\ \leq\ m, j \neq i}$$

Thus, the aim is to find the class which maximizes $\mathrm{P(Ci\ |X)}$. The maximum posteriori probability $\mathrm{P(Ci\ |X)}$ is computed as:

$$P(C_i\ |X)\ = \frac{P(X|C_i)\ P(C_i)}{P(X)} \tag{3.86}$$

Since $P(X)$ does not change for any classes, only $P(X|C_i)\ P(C_i)$ can be maximized for class C_i. If the probability of C_i, $P(C_i)$, is not known, it is supposed that all the classes have equal probability, i.e., $P(C_1) = P(C_2) = \cdots P(C_k)$. It should not be forgotten that class probabilities are also predicted $P(C_i) = freq(C_i, T)/|T|$.

Given a data set consisting of many features, computing $P(X|C_i)$ would be very expensive in terms of computational cost. With the aim of reducing computation in evaluating, $P(X|C_i)\ P(C_i)$, the naïve hypothesis of class independence is made. Mathematically,

$$P(X|C_i) \approx \prod_{k=1}^{n} P(x_k|C_i) \tag{3.87}$$

where x_k indicates the value of feature A_k on sample X. The probabilities $P(x_1|C_i), P(x_2|C_i), \dots P(x_n|C_i)$ is predicted easily in training set.

(a) In case of A_k is categorical, $P(x_k|C_i)$ is the number of samples of

C_i class that have x_k value for feature A_k in T divided by frequency (C_i, T), the number of samples belonging to C_i in T.

(b) If A_k is continuous valued, then it is assumed that the values posses a Gaussian distribution with a standart deviation σ and mean μ defined by:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp -\frac{(x-\mu)^2}{2\sigma^2} \tag{3.88}$$

Thus,

$$P(x_k|C_i) = g(x_k, \mu C_i, \sigma C_i) \tag{3.89}$$

where $\mu C_i, \sigma C_i$ mean and standart deviation of feature A_k for training samples belonging to clas C_i. The classifier evaluates $P(X|C_i)\ P(C_i)$ for each class C_i and predicts the maximum one for sample X(Leung,2007).

3.2.1.4. Linear Discriminant Analysis

As it is said in section (2.4.2.1), Linear Discriminant Analysis is interested separability of classes, thus it can be used for classification as well as dimension reduction. Let x_i be a training set belonging to class C_i; μ_i is mean vector of features belonging to $i-th$ class and μ is global mean. x_i^0 is mean corrected data of class C_i that is $x_i - \mu$. The covariance matrix of C_i:

$$cov_i = \frac{(x_i^0)^T x_i^0}{n_i} \tag{3.90}$$

where n_i is number of samples belonging to class C_i. Pooled within covariance matrix is computed for each entry (r,s) as:

$$COV(r,s) = \frac{1}{n}\sum_i n_i cov_i(r,s) \quad (3.91)$$

Supposing that p_i is prior probability of C_i in training set. After generating covariance matrix and prior probabilities of each classes, the seperability of data is examined on a test set by a discriminant function for each class. The discriminant function is computed as:

$$f_i = \mu_i COV^{-1} x_k^T - \frac{1}{2}\mu_i COV^{-1}\mu_i^T + \ln(p_i) \quad (3.92)$$

where COV^{-1} is Moore-Penrose pseudo inverse of covariance matrix. For sample x_k, LDA predicts class that has maximum discriminant function.

3.2.2. Clustering Methods

3.2.2.1. K-Means

Supposing that D is data set that has n samples in Euclidian space. D is divided into k clusters, $C_1, C_2, \dots C_k$, where $C_i \cap C_j = 0$ for $1 < i,j \leq k$ by partitioning algorihm. While an object is placed in the same group as the objects most similar to it, the similarty between different groups is tried to be minimized.

A centroid based partitioning clustering method k-means uses cendroid c_i to represent a cluster. Cendroid is the center of a cluster. A centroid can be computed in different ways such as the medoid or mean of objects for a cluster. The quality of a cluster is computed by the within cluster variation that is sum square error between all the data points and centroid c_i in cluster C_i:

$$E = \sum_{i=1}^{k}\sum_{p\in C_i} dist(p,c_i)^2 \quad (3.93)$$

where E is sum squared error, p is a data point in cluster C_i. The aim is to increase seperability of clusters. It is hard to compute within cluster variation. In

the worst case, we need to enumerate all the partitions that depend on the number of clusters and to check within cluster variation. If k number of clusters and d dimension of features are fixed, the problem can be solved in $O(n^{dk+1}logn)$ times where n is number of data points. To overcome computational cost problem, generally gready search is preferred in practice. One of the most commonly used methods in greedy search is k-means.

The k-means firstly selects k data points in data set D as centre of clusters randomly. All the data points are assigned to each cluster to compute Euclidian based similarty between centroid c_i. In each iteration, k-means recomputes new centroids by averaging each cluster. All the objects are assigned to new clusters according to new centroids. The iterations continue until clusters do not change for all data points.

In the following steps k-means is summarized:

1. select k randomly object as initial centroids in data set D
2. **repeat**
3. (re)assign the object to the cluster that is most similar, according to average of object in the cluster
4. Update average of clusters, that is means of object in for each cluster
5. **until** no change

As with the other greedy searching algorithms, k-means has no guarantee about finding global optimum and generally terminates at local minimum. The result depends on the number of iterations and starting point that means first centroids.

The time complexity of k-means is $O(nkt)$ where n is number of samples, k is number of clusters and t is number of iterations (Han, Kamber and Pei, 2012).

3.2.2.2. AGNES

While partitioning methods require a certain number of clusters to group a set of objects, in some case it may be needed to partition data into clusters at different levels like a hierarchy. Hierarchical clustering method tries to group objects with respect to a hierarchy tree of clusters.

AGNES is one of hierarchical clustering method that uses distance matrix as clustering criteria. This method needs a termination condition rather than requiring the number of k clusters. Typically, AGNES accepts each data point as a cluster and iteratively merges the most similar clusters into bigger and bigger clusters until all the data points are collected into same cluster. The single cluster is known as root of hierarchy tree. In each merging step AGNES finds the closest cluster pairs to merge them.

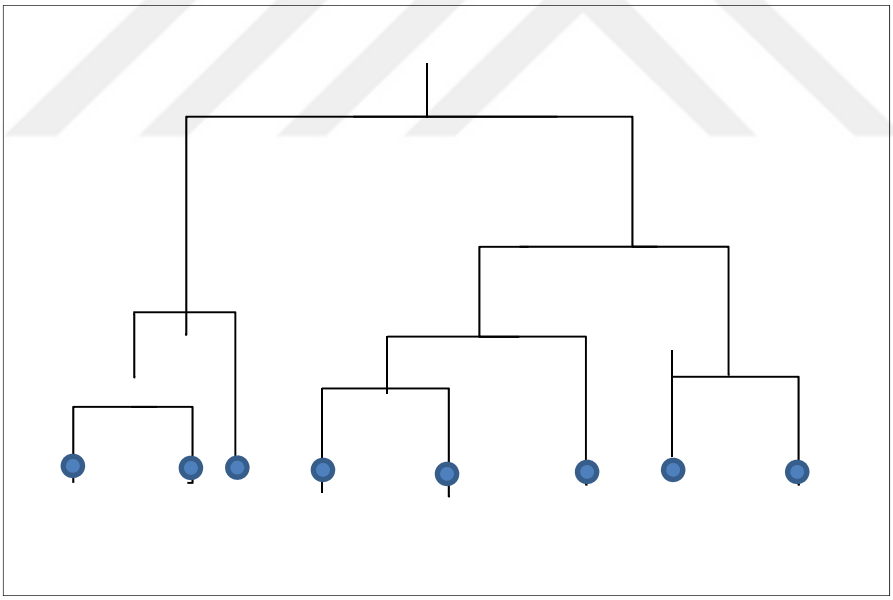


Figure 3.6 Dendogram of AGNES

3.2.3. Proposed Method

Filter-based techniques ignore the correlations between features and evaluate them individually. This means that the features that are considered irrelevant are completely excluded from the calculation and prevented the probability of perfect fit between them.

In order to prevent removal of unchosen features in proposed method, the most relevant top-n features are selected by applying feature selection and unchosen features are passed to dimension reduction algorithm in order to ensure m-dimensional projection of them. In this way, new $m+n$ dimensional data is created by combining features that are selected from original feature set and derived as a result of the projection of irrelevant features by dimension reduction. Hence, irrelevant features gain chance to be used in classification.

For this purpose, ReliefF is selected among feature selection methods and PCA, one of the unsupervised dimension techniques to apply a projection of irrelevant feature set. Although LDA performed better results, in this thesis PCA is chosen for dimension reduction because of computational cost and biased results of LDA. DBN is also not preferred because of its excessive time complexity and local minimum problem caused by the initial weights.

In the following diagram, the proposed hybrid feature selection method is shown:

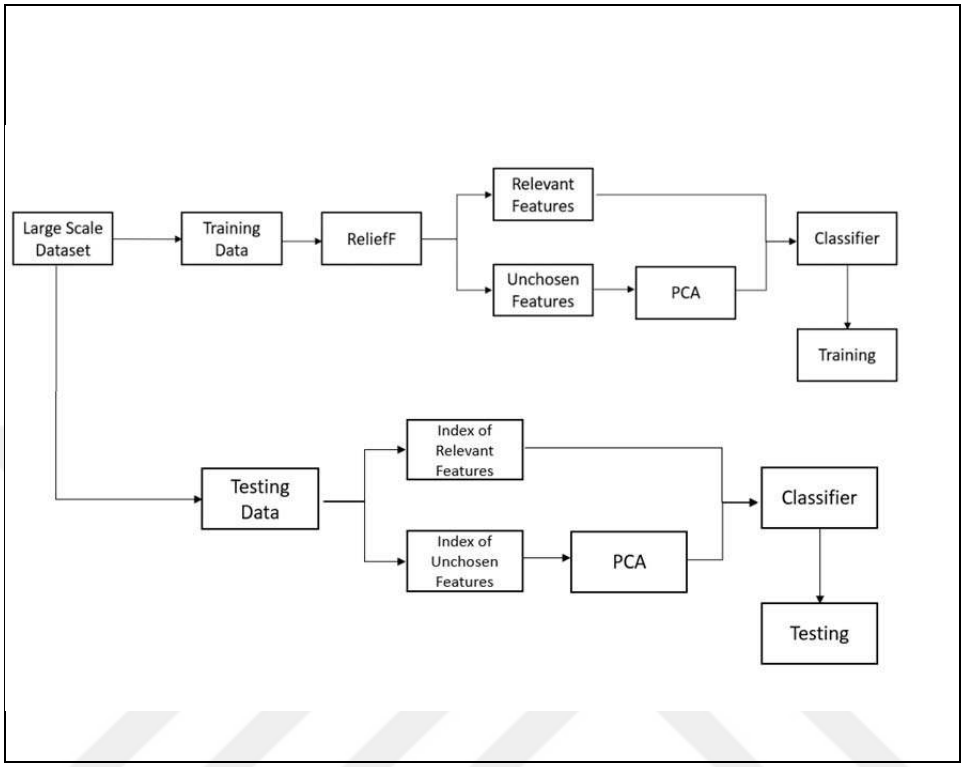


Figure 3.7 Diagram of Proposed Hybrid Feature Selection Method

4. RESULTS AND DISCUSSIONS

In this thesis, the feature selection methods, PCA, LDA and DBN were applied on three datasets which can be considered as small, medium and large scales. For all tests, 70% of dataset was used as train data and 30% of dataset was used as test data. F-measure metric was utilized to evaluate performances of the approaches. In Table 4.1, 4.2 and 4.3, statistical information about the datasets utilized in experimental studies and the distribution of training and test data for each dataset are given. In addition, the number of common features that are selected by feature selection methods explained in the previous section are shared in the Table 4.4, 4.5, 4.6 for each dataset.

Table 4.1 Dataset Descriptions

Dataset	# of samples	# of features	# of classes
Arcene	200	10000	2
Gisette	7000	5000	2
Gene Expression Cancer RNA-Seq	801	20531	5

Table 4.2 Distribution of Training Data According to Classes

	Class 0	Class 1	Class 2	Class 3	Class 4
Arcene	79	61	-	-	-
Gisette	2466	2434	-	-	-
Gene Expression	208	104	54	99	96

Table 4.3 Distribution of Test Data According to Classes

	Class 0	Class 1	Class 2	Class 3	Class 4
Arcene	33	27	-	-	-
Gisette	1034	1066	-	-	-
Gene Expression	92	42	24	42	40

Table 4.4 Selected Common Features Between Feature Selection Algorithms for top-100 Features on Arcene Dataset

	InfoGain	MRMR	Interact	Relief	ReliefF	SAM	Fisher Score	Fisher Ratio	T-Test	Rank Product	Gini Index
InfoGain	-	7	8	5	4	2	1	0	0	0	96
MRMR	7	-	0	8	2	1	1	0	0	0	6
Interact	8	0	-	4	1	2	1	1	2	1	8
Relief	5	8	4	-	10	0	0	7	0	0	5
ReliefF	4	2	1	10	-	0	0	4	0	2	4
SAM	2	1	2	0	0	-	98	0	93	8	2
Fisher Score	1	1	1	0	0	98	-	0	94	7	0
Fisher Ratio	0	0	1	7	4	0	0	-	0	0	0
T-Test	0	0	1	0	0	93	94	0	-	9	0
Rank Product	0	0	1	0	2	8	7	0	9	-	0
Gini Index	96	6	8	5	4	2	0	0	0	0	-

Table 4.4 shows the number of common features for the top 100 features selected by feature selection methods on the Arcene dataset. There are very few similarities in the 100 features selected by entropy-based methods such as Information Gain, MRMR and Interact. In addition, Gini Index, which has a similar approach to Information Gain, has 96 common features. Similarly, the number of common features between Relief and ReliefF is only 10. T-Test, Fisher Score and SAM, which show similar approaches, are very close to each other in terms of selected common features. Fisher Ratio does not seem to have a significant similarity with almost any other feature selection method.

Table 4.5. Selected Common Features Between Feature Selection Algorithms for top-100 Features on Gisette Dataset

	InfoGain	MRMR	Interact	Relief	ReliefF	SAM	Fisher Score	Fisher Ratio	T-Test	Rank Product	Gini Index
InfoGain	-	3	0	62	29	88	93	0	93	32	98
MRMR	3	-	0	6	5	3	4	0	4	6	3
Interact	0	0	-	0	0	0	0	4	0	1	0
Relief	62	6	0	-	48	62	61	0	61	30	62
ReliefF	29	5	0	48	-	30	29	0	29	17	29
SAM	88	3	0	62	30	-	92	0	92	32	87
Fisher Score	93	4	0	61	29	92	-	0	100	33	92
Fisher Ratio	0	0	4	0	0	0	0	-	0	0	0
T-Test	93	4	0	61	29	92	100	0	-	33	92
Rank Product	32	6	1	30	17	32	33	0	33	-	32
Gini Index	98	3	0	62	29	87	92	0	92	32	-

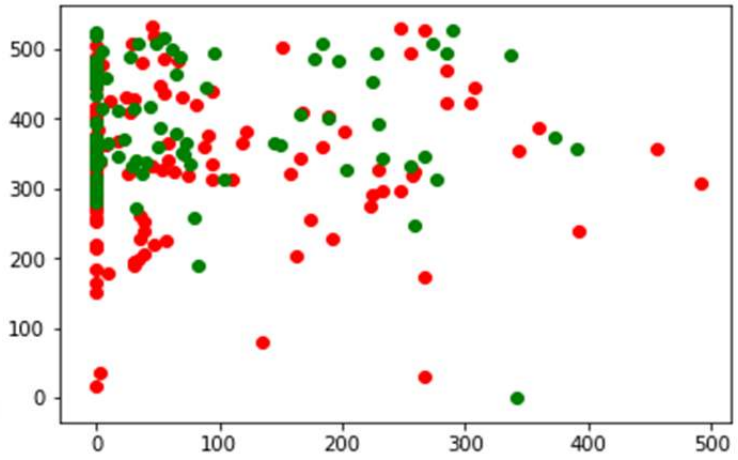
Just as on Arcene dataset, the 100 features selected by Information Gain, MRMR and Interact are not similar on the Gisette dataset. Similarly, Information Gain and Gini Index have 98 common features. It is seen a increase in terms of number of common features selected by Relief and ReliefF compared to Arcene on Gisette. Fisher Score and T-Test find all the top-100 features the same. This means that the results they produce in a classifier that finds global optimum will be the exactly same. In the same way, SAM finds 92 features to be the same with Fisher Score and T-Test. Just as on the Arcene dataset, Fisher Ratio is not similar to almost any method.

Table 4.6. Selected Common Features Between Feature Selection Algorithms for top-100 Features on Gene Expression Cancer RNA-Seq Dataset

	InfoGain	MRMR	Interact	ReliefF	Fisher Score	Gini Index
InfoGain	-	3	100	0	0	99
MRMR	3	-	3	0	0	3
Interact	100	3	-	0	0	99
ReliefF	0	0	0	-	17	0
Fisher Score	0	0	0	17	-	0
Gini Index	99	3	99	0	0	-

On the Gene Expression dataset, since the remaining 5 feature selection methods are used to solve binary classification problems, 6 feature selection methods are compared. Information Gain and Interact, which do not show any similarity in terms of number of common features in the previous two datasets, find all the 100 features common on Gene Expression Cancer RNA-Seq dataset. MRMR differs from both as in other datasets. Similarly, the high similarity between Gini Index and Information Gain is becoming prominent.

Due to the high dimensionality of the Gene Expression Cancer RNA-Seq data, in the LDA classification and dimension reduction processes, memory error was encountered in extracting the covariance matrix. Therefore, Gene Expression Cancer RNA-Seq dataset was exempted from the LDA classifier and dimension reduction method. In addition to the tables, distribution of all datasets were illustrated in two-dimensional space to understand the complexity of the problem on whole datasets.



4.1 Distribution of Arcene Dataset

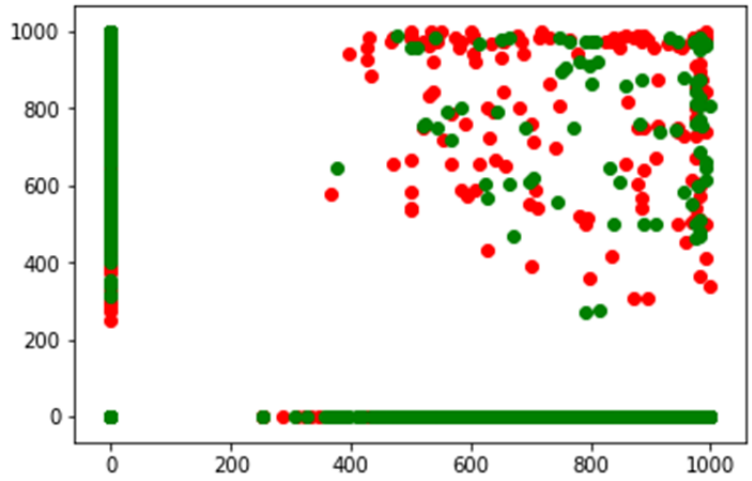


Figure 4.2 Distribution of Gisette Dataset

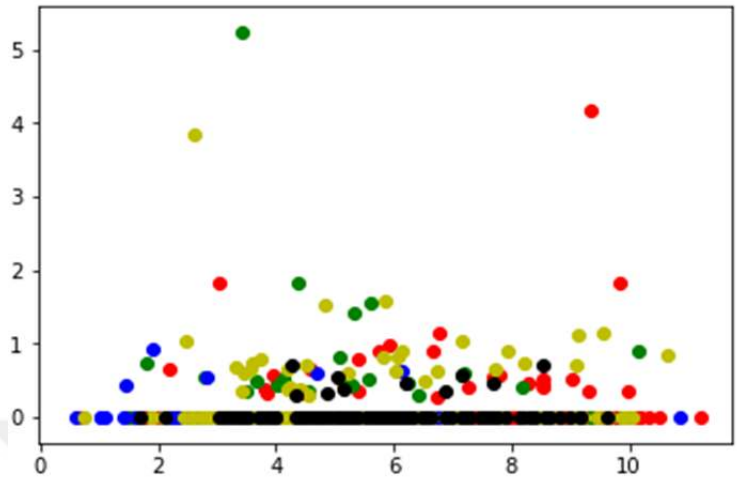


Figure 4.3 Distribution of Gene Expression Cancer RNA-Seq Dataset

In Figures 4.1, 4.2 and 4.3, the distribution of each dataset is plotted with 2 features. According to the figures, a linear separation does not seem possible on all three data sets using these two features.

4.1. Tests and Results of Classifiers with Existing Methods

In this thesis, four different classifiers were utilized to discuss efficiency of the feature selection methods, PCA, LDA, and DBN. The chosen classifiers are MLP, Naïve Bayes, SVM, and LDA. The following subsections demonstrate the test results of these classifiers for each tested methods.

4.1.1. Multi-Layer Perceptron

Since different initial weight values directly affect the results, tests were repeated 10 times and average results were given in Table 4.7.

Table 4.7 F-Measure Results of MLP

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		0.69	0.88	0.11
InfoGain	#50	0.70	**0.89**	0.93
	#100	0.62	**0.92**	0.90
MRMR	#50	0.69	0.69	0.90
	#100	0.62	0.86	0.77
Interact	#50	0.52	0.57	0.93
	#100	0.51	0.64	0.73
Relief	#50	0.63	0.74	-
	#100	0.64	0.78	
ReliefF	#50	0.61	**0.89**	0.72
	#100	0.64	0.80	0.67
SAM	#50	0.64	**0.89**	-
	#100	0.68	0.88	
Fisher Score	#50	0.65	0.82	0.77
	#100	0.66	0.81	0.51
Fisher Ratio	#50	0.49	0.54	-
	#100	0.53	0.27	
T-Test	#50	0.58	0.73	-
	#100	0.62	0.79	
Rank Product	#50	0.66	0.68	-
	#100	**0.72**	0.74	
Gini Index	#50	0.68	0.85	0.78
	#100	0.68	0.74	**0.95**
PCA	#50	0.49	0.73	0.50
	#100	0.60	0.89	0.81
LDA	#50	**0.73**	0.82	-
	#100	0.65	0.84	
DBN	#50	0.65	0.83	0.21
	#100	0.68	0.86	0.23

The highest f-measure results for each dataset are shown in bold. As seen in Table 4.7, for Arcene dataset, the best f-measure result, 0,73, is achieved by LDA for 50-dimensions. The worst results are obtained by Fisher Ratio and PCA for Arcene dataset in 50-dimensional. In comparison to no feature selection, LDA increases the success of classification by 4% in 50 dimensional space. In top-100 dimensions, it is seen that the best classification result is achieved by Rank Product on Arcene dataset. Except Information Gain, MRMR, Interact and LDA, it is

shown that the success is increased in 100-dimensional dataset in comparison with 50-dimensional. As seen in Table 4.7 the result doesn't change in Gini Index. For Information Gain and MRMR that are similar algorithms, the results are only 1% diverged for 50 dimensions and the same for top 100 dimensions. Arcene dataset is balanced dataset in terms of number of samples in each class. However, when the Table 4.4 is reviewed, although the number of common features that are selected by different feature selection algorithms are low, the results of classification are so close to each other. Even Information Gain and MRMR that are similar approaches, they produce the same result for top-100 features, only 7 of selected features by them are common. Similarly, ReliefF and Relief algorithms give the same result in 100 dimensions, but only 10 features are common. Moreover, Information Gain and T-Test which have no common features in top 100, have the same results in 100-dimensional space. This is the proof that each feature provides meaningful information to classifier, so no feature selection method produces acceptable result. While the algorithms that have non-common features produce similar results, the methods which have almost all the features are common produces slightly different f-measure such as SAM-Fisher Score, SAM-T-Test and Fisher Score-T-Test. Information Gain and Gini Index, one of which uses entropy and other probability values, have 96 common features, but they do not give the same results in both dimensions. In the figure below, we show the distribution of data according to the two methods that produce the most successful results, Rank Product and LDA, on Arcene dataset:

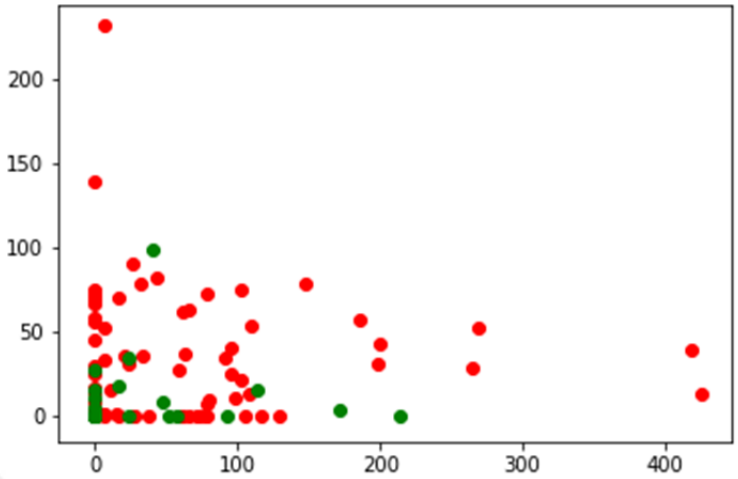


Figure 4.4 Distribution of Arcene Dataset with The Features Selected by Rank Product

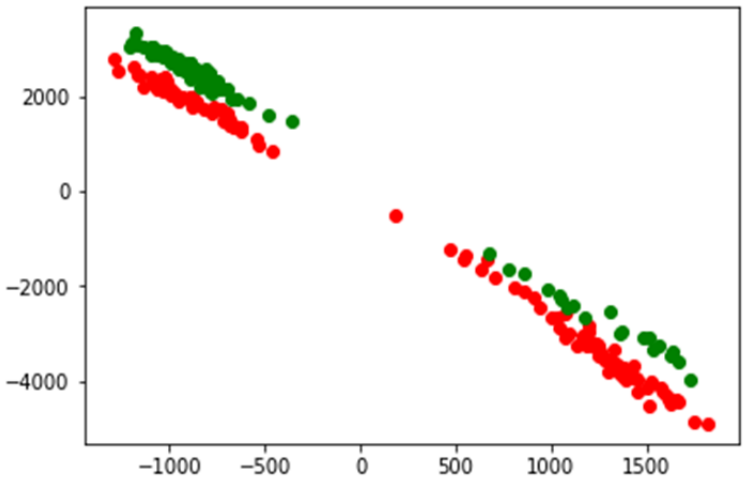


Figure 4.5 Distribution of Arcene Dataset Rotated by LDA

When the distribution of the data according to Rank Product is reviewed on Arcene dataset in Figure 4.4, it is seen that the data cannot be separated even with the best feature selection method. When the data distribution drawn according to LDA is examined in Figure 4.5, it is seen that the data are separated more

successfully than Rank Product. The success of 0.73 f-measure is due to the local minimum problem of MLP.

For Gisette dataset, the highest f-measure results that is 0.89 belong to Information Gain, SAM and ReliefF for 50 dimensions. When it is compared to Information Gain, it is found that the success of classification by MRMR falls to 0.69 in 50 dimensions. This circumstance shows that irrelevant features have more effect on classification by MRMR as against on Arcene dataset. The fact that there are only 3 common features between Information Gain and MRMR in top 100 features also refers this. For top 100 features the best f-measure result, 0.92, is obtained by Information Gain. Despite the fact that selected top 100 features by Gini Index are almost the same with Information Gain, except 2 features, the classification which is obtained with Gini Index's features is worse than success of Information Gain's features. SAM and Fisher Scoring which have 92 common features, show 7% differences in 50 and 100 dimensions. SAM and ReliefF which cause the same classification result in MLP have 18 common features in top 50 features while this number is 30 for top 100. There are 88 common features between SAM and Information Gain and this similarity reflect the result positively, that is 0.89 in 50 dimensions. The worst results for both 50 and 100 dimensions belong to Fisher Ratio with respectively 0.54 and 0.27 and it is clearly observed that Fisher Ratio does not provide common features with any other feature selection algorithm. Although the distribution of data to classes is balanced in Gisette as in Arcene, it is seen that the success of some similar methods becomes distant such as Relief and ReliefF; Information Gain and MRMR; Fisher Score, Fisher Ratio and T-Test. The reason for this may be that even the number of data shows balance, the values of features may differ in training and test. Distributions of data according to the features selected by the most successful feature selection and dimension reduction methods are shown below.

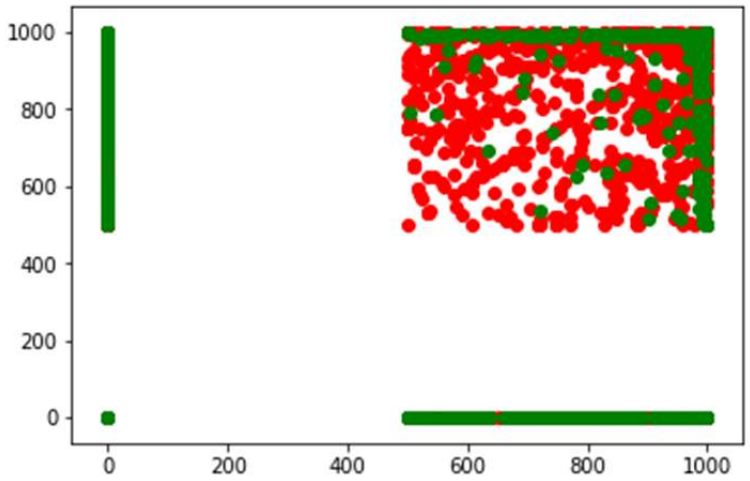


Figure 4.6 Distribution of Gisette Dataset with The Features Selected by Information Gain

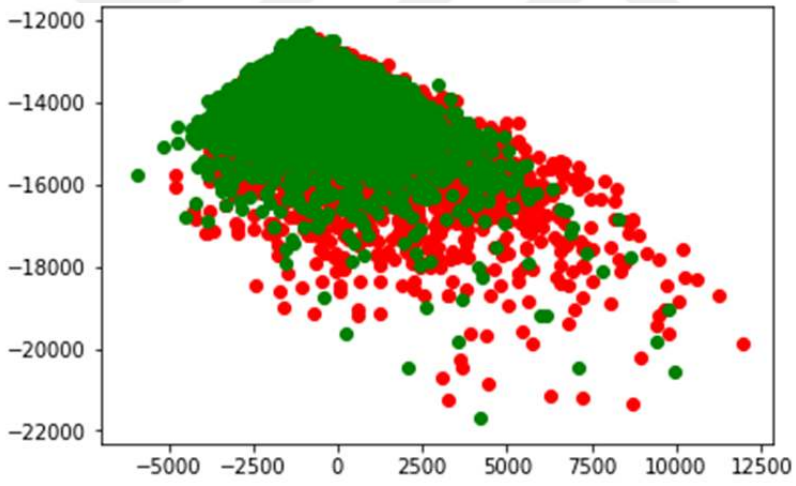


Figure 4.7 Distribution of Gisette Dataset Rotated by PCA

When the Figure 4.6 and 4.7 are examined, although they do not appear to be separable in two-dimensional space, it can be said that a hyperplane passes between the data of the two classes at higher dimensions. However, there is no loss-free seperation in both figures.

In the Gene Expression Cancer RNA-Seq dataset, all feature selection methods in the MLP classification provide better performances than that of the no feature selection case. When the results obtained with Information Gain and Interact, where all of the top 100 features are common, are analyzed, it is seen that they produce the same result for the top 50 features, however the success of Interact with 100 features decreases. Moreover, the rank of the selected features by the two methods is exactly the same. This is due to the fact that MLP may stuck to the local minimum depending on the initial weight even if the same input units are used. In another example, Gini Index, which has 99 features in the first 100 features are common with Information Gain and Interact, provides worse results than that of both Information Gain and Interact performance in 50 dimensions, but in 100 dimensions it is vice versa. There are only 3 common features between Information Gain and MRMR therefore, it can be said that the farthermost features of the set change their order in MRMR. Since Relief, Sam, Fisher Ratio, T-Test and Rank Product are suitable for binary datasets, they are not used on Gene Expression Cancer RNA-Seq dataset. It is understood that in all feature selection methods except the Gini Index, the top second 50-dimensional feature set has negative effect on the success achieved with the first 50-dimensional feature set.

In Figure 4.8 and 4.9, the distribution of Gene Expression Cancer RNA-Seq data according to top-2 features selected by Gini Index that has the highest f-measure result and rotated features which have the highest eigenvalues by PCA in MLP classifier are shown, respectively.

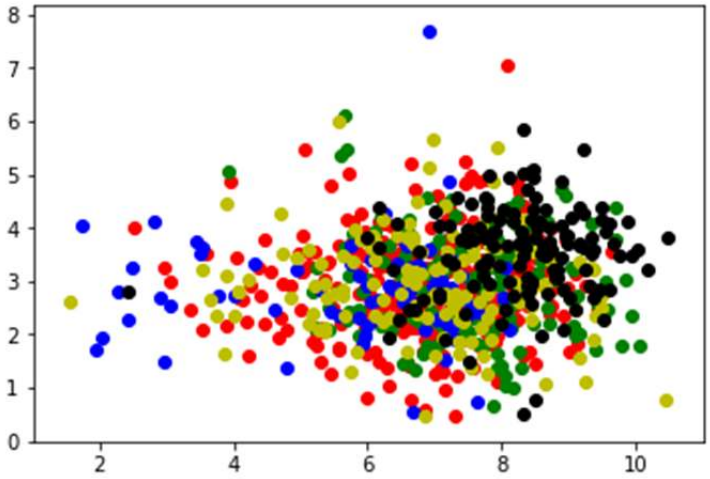


Figure 4.8 Distribution of Gene Expression Cancer RNA-Seq Dataset with The Features Selected by Gini Index

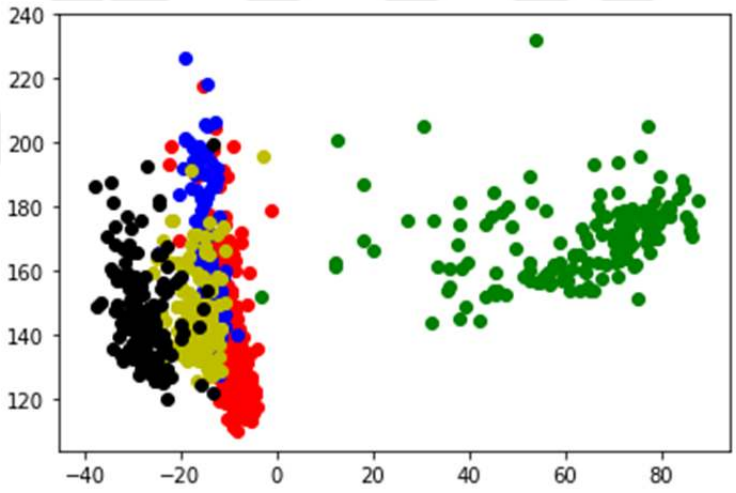


Figure 4.9 Distribution of Gene Expression Cancer RNA-Seq Dataset rotated by PCA

According to Figures 4.8 and 4.9, although the separability of Gene Expression Cancer RNA-Seq dataset with PCA is greater, a more successful classification is achieved by Gini Index. This is due to the local minimum problem of MLP.

4.1.2. Naïve Bayes

Naïve Bayes was utilized as the second classifier in this thesis. The following table shows the Naïve Bayes classifier F-measure performances with the selected features.

Table 4.8 F-Measure Results of Naïve Bayes

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		0	0.07	0.03
InfoGain	#50	0.43	0.11	0.04
	#100	0.20	0.10	0.05
MRMR	#50	**0.47**	0.31	0.04
	#100	0.24	0.19	0.05
Interact	#50	0.19	**0.67**	0.04
	#100	0	**0.67**	0.05
Relief	#50	0.45	0.14	-
	#100	0.40	0.11	
ReliefF	#50	0.45	0.14	0.04
	#100	0.20	0.13	0.05
SAM	#50	0.14	0.11	-
	#100	0.05	0.09	
Fisher Score	#50	0.14	0.11	0
	#100	0.05	0.09	0
Fisher Ratio	#50	0.07	**0.67**	-
	#100	0	**0.67**	
T-Test	#50	0.15	0.11	-
	#100	0.05	0.09	
Rank Product	#50	0.12	0.12	-
	#100	0	0.11	
Gini Index	#50	0.45	0.11	0.04
	#100	0.26	0.11	0.05
PCA	#50	0.25	0.49	0.11
	#100	0.30	0.47	0.12
LDA	#50	0.30	0.09	-
	#100	**0.44**	0.02	
DBN	#50	0.21	0.15	0.11
	#100	0.18	0.13	0.01

According to Table 4.8, the best classification result in 50-dimensional Arcene dataset by Naïve Bayes classifier is obtained with MRMR algorithm, that is 0.47. Surprisingly, neither no feature selection nor Interact, Rank Product and Fisher Ratio show any success on Arcene for 100 features. For the top 100 features

the best f-measure result that is 0.44 belongs to LDA. Two of feature selection methods that have 98 common features, SAM and Fisher Score, provide the same results in both dimensions. In addition to this, Information Gain - Gini Index, SAM-T-Test and T-Test-Fisher Score pairs that have more than 90 common features lead to similar classification results. As in MLP, Fisher Ratio which has almost no common features with other approaches shows unsuccessful result. The features selected by Rank Product does not provide any success and for Arcene dataset, the most similar method to Rank Product in terms of number of common features, T-Test, also induces bad classification results. Since Naïve Bayes directly assigns a probability to each unique value that is seen in a feature, beside distribution of number of data both training and test, it is also important how many times a variable appears in training. In Arcene dataset, using all the features causes Naïve Bayes finds all the data belongs to class 0. When the number of unique features in training data is examined, it is seen that fewer unique values appear in the class 1 data for almost all the features. This leads to zero-f-measure in Naïve Bayes that has lower generalization capability than MLP for repeated data. While the number of repeated features in no feature selection is two or three times as much in class 0, this clutter has become more collective in 50-dimensions selected by MRMR. For the top-50 features selected by ReliefF it is also observed a balanced distribution of unique features on each feature for both classes. To illustrate this situation, the number of unique values for 10 features selected by MRMR, Rank Product and ReliefF in training data for both classes and how many of them also appears in test are given in following table:

Table 4.9 Number of unique values in training and common values for 10-features selected by MRMR in training and test data on Arcene dataset for each class

Class 0		Class 1	
# of common values	# of unique values	# of common values	#of unique values
6	74	2	58
7	68	1	56
7	66	3	50
12	69	3	54
4	73	2	55
7	70	4	53
8	74	4	56
7	69	9	50
5	65	4	51
10	66	6	55

When the number of common unique values between training and test data in the 10 features selected by MRMR is reviewed, both number of common values and number of unique values that are appear in training resemble for class 0 and class 1. Nevertheless, class 0 dominates class 1 for some features and this causes Naïve Bayes predicts some data as class 0 belonging to class 1. However, the number of false negatives is very low compared to other feature selection methods. The classification that is made with the features selected by MRMR produces one of the best f-measure results due to this balance between training and test data for each class.

Table 4.10 Number of unique values in training and common values for 10-features selected by Rank Product in training and test data on Arcene dataset for each class

Class 0		Class 1	
# of common Values	# of unique values	# of common Values	#of unique values
11	31	4	20
1	4	1	3
5	45	1	20
6	49	1	11
3	28	1	4
6	21	1	3
1	4	1	3
9	41	1	9
1	1	1	2
1	18	3	26

As seen in Table 4.10, the number of common values for class 0 dominates to class 1 in 6 features. Although unique values appear less in class 1, stucking most of common values at 1 causes weak classification. For this reason, Naïve Bayes predicts almost all the test data belong to class 0 except one. Unfortunately, it is seen that class 1 cannot be obtained in this classification because selected features by Rank Product are biased.

Table 4.11 Number of unique values in training and common values for 10-features selected by ReliefF in training and test data on Arcene dataset for each class

Class 0		Class 1	
# of common Values	# of unique values	# of common Values	#of unique values
9	44	4	33
6	45	2	35
3	22	4	24
11	64	3	47
10	55	8	32
2	20	3	17
9	68	12	45
11	62	8	56
9	36	4	27
10	52	7	43

When Table 4.9 and 4.11 are reviewed, it is seen that number of common values in both training and test sets and number of appeared unique values in training data show similar distribution for MRMR and ReliefF although number of their common features is just 2. In both tables, there is no dominant distribution biased any class as seen Table 4.11.

The distribution of the data with the features selected by MRMR, which gives the most successful result in Naive Bayes, is shown in the figure below. The distribution obtained with LDA, the most successful dimension reduction method, is given in Figure 4.5.

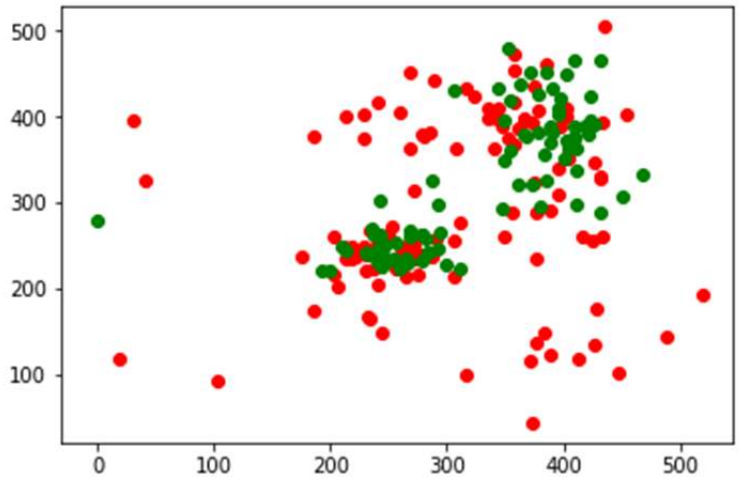


Figure 4.10. Distribution of Arcene Dataset with The Features Selected by MRMR

According to Figure 4.10, even with the top-2 features selected by MRMR, the Arcene dataset is not seen to be linearly separable. Although MRMR is the most successful feature selection method on the Naive Bayes classifier, it produced only a value of 0.47 f-measur due to this inseparability.

On Gisette dataset, the highest f-measure results, 0.67, are obtained with the features that are selected by Interact and Fisher Ratio for both dimensions. None of the remaining algorithms can exceed this success and provide worse performances than them. The best classification results that are 49% and 47% on 50 and 100-dimensional Gisette dataset are acquired with PCA among dimension reduction methods. Although the classification performance of PCA is 49%, in comparison to classification results that are obtained with other feature selection and dimension reduction methods, it can be considered successful. The classification results obtained using the features selected by Information Gain, SAM, T-Test, Gini Index and Fisher Scoring whose binary combinations have 87 to 98 common features, are the similar varying in the range of 0.09 and 0.11 in both dimensions. On the contrary, the approaches of Interact and Fisher Ratio having no common feature with these 5 methods, in top 100 features work on

Gisette dataset against Arcene. Gisette dataset consists of most of zero values such that only value that is observed in some selected features is zero. When selected features by Interact and Fisher Ratio are examined, most of them have only one unique value, that is zero. The number of common values for both training and test data sets and number of unique values in 10 features by selected Interact, Fisher Ratio and SAM on Gisette dataset are given.

Table 4.12 Number of unique values in training and common values for 10-features selected by Interact in training and test data on Gisette dataset for each class

Class 0		Class 1	
# of common Values	# of unique values	# of common Values	#of unique values
2	12	3	15
1	13	1	15
1	13	1	15
1	10	1	18
2	13	1	16
3	19	1	1
1	2	3	27
1	1	1	26
1	1	6	22
1	3	2	25

As seen in Table 4.12, all the values belonging to 10 features selected by Interact show a balanced range in each class. The fact that there are many zeros in both training and test data, that causes some features to be composed by only zero-values. When the number of unique values is examined, a narrow distribution is observed for both classes. However, these zero values have the same effects on both classes and prevents the biased results in favor of any class.

Table 4.13 Number of unique values in training and common values for 10-features selected by Fisher Ratio in training and test data on Gisette dataset for each class

Class 0		Class 1	
# of common Values	# of unique values	# of common Values	#of unique values
1	1	1	3
1	1	1	2
1	1	1	2
1	1	1	5
1	2	1	1
1	1	1	6
1	1	1	4
1	1	1	3
1	5	1	18
1	1	1	16

According to Table 4.13, it is clear that almost all of the features selected by Fisher Ratio consist of zero values, which are more than the features selected by Interact. In both training and test sets, the number of unique and common values are noteworthy.

Table 4.14 Number of unique values in training and common values for 10-features selected by SAM in training and test data on Gisette dataset for each class

Class 0		Class 1	
# of common Values	# of unique values	# of common Values	#of unique values
98	271	7	51
133	305	10	61
141	335	34	154
132	312	42	144
114	294	4	46
87	262	3	33
106	296	2	43
163	343	77	236
129	329	32	144
81	240	3	33

When it is compared with Table 4.12 and 4.13, the values belonging to features that are selected by SAM are too messy as seen in Table 4.14. The unbalances of common values between class 0 and class 1 cause results to be biased to class 0. Most of data belonging to class 1 are predicted as class 0 as the result of classification because of this messy. In the test data, 971 of 1066 data belonging to class 1 were predicted as class 0 and only 95 were predicted correctly. When the values taken by the features selected by SAM are examined, there is imbalance class problem for the class 1 and it causes biased classification. This disproportion is observed in other methods that give similar results to SAM. The worst results in both 50 and 100 dimensional are greatest proof of this argument.

In Figure 4.11, distribution of dataset according to Interact is shown. The distribution obtained with Fisher Ratio is also completely the same with Interact.

The distribution with PCA, the most successful dimension reduction method on Gisette, is shown in Figure 4.7.

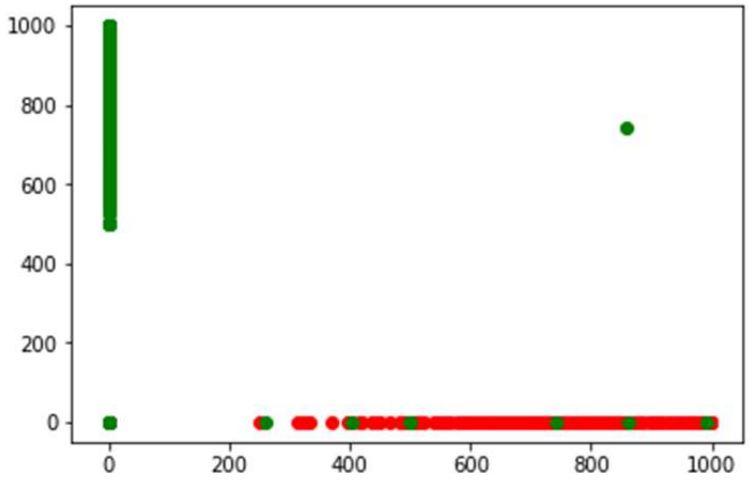


Figure 4.11 Distribution of Gisette Dataset with The Features Selected by Interact

Looking at the distribution of the data with the features selected by Interact on the Gisette dataset, it is seen that even though more successful f-measure value is produced compared to Arcene dataset, a perfect classification is still not possible according to Figure 4.11.

For Gene Expression Cancer RNA-Seq, it can be said that no feature selection method could show any success. Since none of them can exceed 0.05 f-measure value, illustration of the distribution of dataset according to top-2 features selected by Gini Index is given in Figure 4.8. Besides, both the information of the common values seen in both training and testing sets and the number of unique values seen in training set belonging to 10 features that are selected by Information Gain for each class are given in Table 4.15. The feature selection methods, such as MRMR and Interact, providing the best f-measure classification results on Arcene and Gisette datasets cannot provide success for Gene Expression Cancer RNA-Seq dataset in Naive Bayes classifier. Fisher Score that does not provide any success

onboth Arcene and Gisette datasets, also provides zero f-measure on Gene Expression Cancer RNA-Seq dataset. Although dimension reduction methods provided better f-measure results in comparison to feature selection techniques relatively, they are not also satisfactory for a classification.

On the Gene Expression Cancer RNA-Seq dataset, feature selection and dimension reduction methods could not provide satisfactory results with Naïve Bayes classifier. Therefore, feature analysis was performed on Information Gain method. The number of common features in training and test sets and the number of unique values in the training set are given in the table below. In addition, the data distribution obtained by PCA is given in Figure 4.9.

Table 4.15 Number of unique values in training and common values for 10-features selected by Information Gain in training and test data on Gene Expression Cancer RNA-Seq dataset for each class

Class 0		Class 1		Class 2		Class 3		Class 4	
#of common values	#of unique values	#of common values	#of unique values	#of common values	#of unique values	#of common values	#of unique values	#of common values	#of unique values
1	5	1	9	1	4	1	9	1	5
1	195	1	99	0	54	1	98	0	95
0	207	0	104	0	54	0	98	0	96
0	207	0	104	0	54	0	99	0	96
0	207	0	104	0	54	0	99	0	96
1	1	1	1	1	1	1	1	1	1
0	207	0	104	0	54	0	99	0	96
2	135	1	70	1	36	1	57	2	56
1	7	1	2	1	1	1	4	1	2
1	2	1	1	1	1	1	1	1	1

4.1.3. Linear Discriminant Analysis

The third classification method was chosen as LDA. The following table represents the test results obtained with LDA.

Table 4.16 F-Measure Results of LDA

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		**0.85**	0.65	Memory error
InfoGain	#50	0.69	0.89	0.93
	#100	0.58	0.53	0.97
MRMR	#50	0.72	0.74	0.93
	#100	0.70	0.86	0.96
Interact	#50	0.60	0.57	0.93
	#100	0.54	0.62	0.97
Relief	#50	0.61	0.89	-
	#100	0.74	0.93	
ReliefF	#50	0.35	0.90	0.98
	#100	0.60	0.93	**0.99**
SAM	#50	0.74	0.89	-
	#100	0.57	0.92	
Fisher Score	#50	0.75	0.89	**0.99**
	#100	0.64	0.91	**0.99**
Fisher Ratio	#50	0.50	0.13	-
	#100	0.56	0.21	
T-Test	#50	0.66	0.89	-
	#100	0.61	0.91	
Rank Product	#50	0.71	0.91	-
	#100	0.61	0.91	
Gini Index	#50	0.66	0.89	0.94
	#100	0.50	0.92	0.97
PCA	#50	0.82	**0.95**	**0.99**
	#100	0.76	**0.96**	**0.99**
LDA	#50	-	-	-
	#100			
DBN	#50	0.74	0.90	0.05
	#100	0.58	0.93	0.03

When the results of LDA classifier is analyzed, it is seen that no method reaches to success of no feature selection, 0.85, on both 50 and 100 dimensional Arcene dataset. Besides that, the success obtained by PCA is the highest among

that 14 methods. ReliefF provided worse result lower than the average results in 50-dimension, which is 0.35. Relief algorithm has higher f-measure result than ReliefF for both dimensions. While the difference between Relief and ReliefF is 0.61-0.35 in 50-dimension, for 100 dimensional it is 0.74-0.60. However, the number of common features Relief and ReliefF is just 10. Among the three methods that are entropy based, Information Gain, MRMR and Interact, the best classification results are obtained with the features selected by MRMR, which are 0.72 for 50-dimensions and 0.70 for 100. Here, it must be pointed out that while there are 7 common features between Information Gain and MRMR, this is 8 between Information Gain and Interact and 0 between MRMR and Interact. Since the results will be biased, we didn't include dimension reduction results of LDA to this test. When the results of Fisher Score, Fisher Ratio, SAM and T-Test that have similar approaches algorithms, are reviewed, Fisher Score and SAM having 98 common features provide similar f-measure result in 50 dimensions. However, Fisher Ratio provided worse than the other 3 methods whereas the most of features of Fisher Score-T-Test, T-Test-SAM and Fisher Score-SAM are common, Fisher Ratio shows difference from both of them.

When the scores of features obtained by a feature selection method such as T-Test, are examined it is seen that the score range of top 100 features is just between 4 and 5 points. This shows the distinction between features is not very clear and more features are needed for good classification in LDA. Undoubtedly, this inseparability between features has effect on achieving more successful results with no feature selection in LDA. We already show that features cannot be separated clearly by the feature selection methods on Arcene dataset; therefore, the discriminant function applied on the selected features cannot work properly in LDA.

Although none of feature selection methods does not exceed the success of the no feature selection, according to the features obtained by Fisher Score, as the most successful feature selection method for LDA, data distribution of Arcene

dataset is given in Figure 4.12. In Figure 4.13, the data obtained by PCA are also plotted.

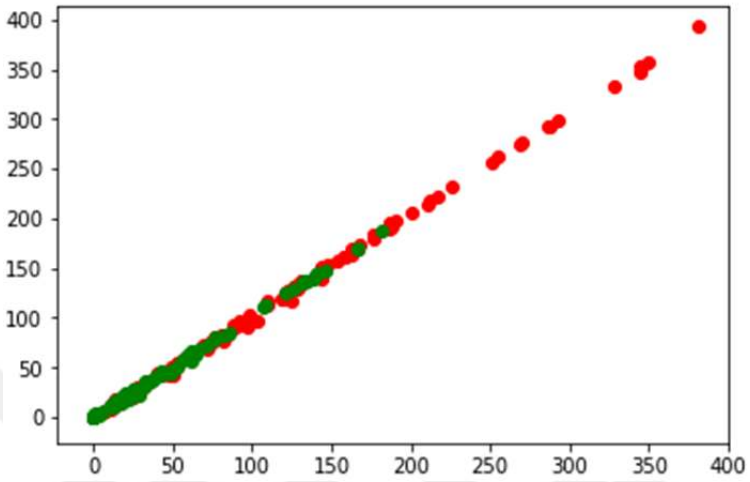


Figure 4.12 Distribution of Arcene Dataset with The Features Selected by Fisher Score

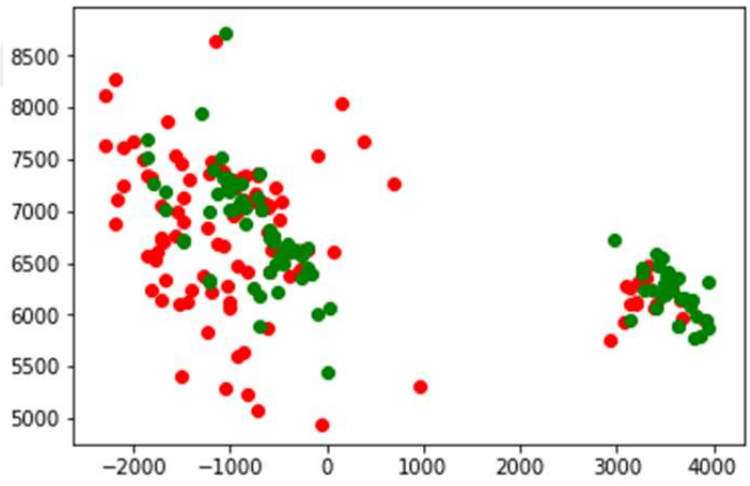


Figure 4.13 Distribution of Arcene Dataset Rotated by PCA

When Figure 4.12 and 4.13 are examined, it is seen that dimension reduction on Arcene dataset has more contribution to the classification compared to

feature selection. However, a perfect classification of LDA is not possible with either approach.

The experiments that are made on Gisette dataset show that except Interact and Fisher Ratio, all the feature selection and dimension reduction methods achieve more success than no feature selection. PCA is seen as the most contributing of classification with the 0.95 and 0.96 f-measure values for 50 and 100 dimensions. When the most similar method pairs are compared, while the f-measure results of Information Gain and MRMR are 0.80 and 0.74 in 50 dimensions, the results become reverse with 0.53 and 0.86 for 100 dimensional. Surprisingly, despite the fact that they have 98 common features, Gini Index and Information Gain show 39% differences in 100 dimensional. Here, when we regard the results obtained with MRMR and Interact, using probability rather than entropy leads to better results in LDA. In Relief and ReliefF pairs, consistent results are achieved for both dimensions; these are 0.89-0.90 and 0.93-0.93. The classification successes of all methods are either unchanged or increased except Information Gain in 100-dimension in comparison with 50. Unlike Arcene dataset, the features that are selected by T-Test show a clear difference on Gisette dataset. The top 100 features that have highest score take between 27 and 240 points. This wide range indicates the distinction between features is made easier on Gisette in comparison with Arcene. Moreover, the top 100 features selected by T-Test and Fisher Score are completely the same. Thus, the success of selected features by T-Test is the same with Fisher Score. Although the score of the features selected by Fisher Ratio scattered, the classification performed with the features selected by Fisher Ratio produces the worst f-measure results on Gisette dataset. When we compare Fisher Ratio's features with that of Fisher Score and T-Test, none of them is common. This shows that Fisher Ratio is a weak feature selection method, although it does not stuck in the narrow range. It should be noted that having broad score ranges does not always mean choosing successful features. For this, it is necessary to analyze attitude of the method on other datasets. For example, Fisher Score

compresses the scores to narrow intervals in both Arcene and Gisette datasets. The T-Test, which selects the same features with Fisher Score finds the top features in a wider range. In spite the fact that Fisher Ratio assigns the points to the features in the similar range with T-Test on Gisette, its features show weak performance.

The distribution of Gisette data according to the ReliefF method is given in Figure 4.14. The most successful method for LDA, is PCA and shown in Figure 4.7.

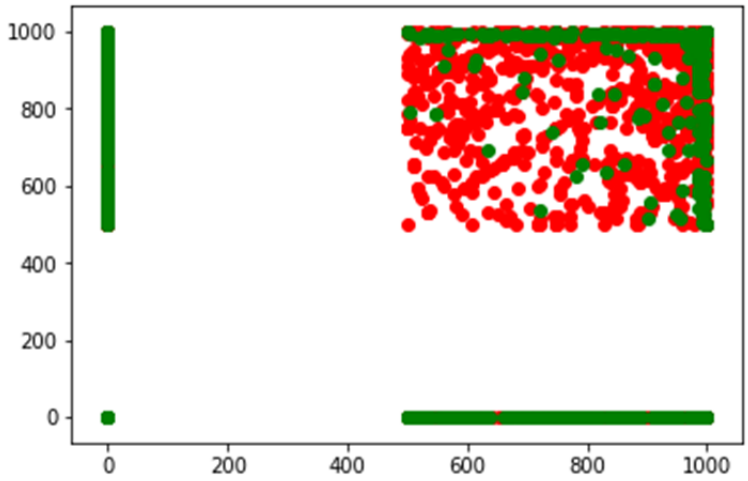


Figure 4.14 Distribution of Gisette Dataset with The Features Selected by ReliefF

According to the distribution obtained with the top-2 features selected by ReliefF on the Gisette dataset in Figure 4.14, a very high f-measure value is obtained even if complete classification is not seen possible. In particular, the fact that the selected features consist of 0 values for almost every data point affects the distribution dataset as is seen.

In LDA classifier, all the feature selection and dimension reduction methods produce successful f-measure results except DBN on Gene Expression Cancer RNA-Seq dataset. As well as the all the f-measure results so close to each other, the highest values are produced by Fisher Score in 50-dimensions and both

ReliefF and Fisher Score in 100-dimensions. Although number of common features selected by entropy-based methods such as Information Gain, MRMR and Interact is too low, all three cause almost the same f-measure results in LDA, that are 0.93 in 50-dimensional and 0.96-97 in 100 dimensional. Whereas one of unsupervised dimension reduction methods, PCA leads to very high f-measure results, DBN doesn't attain any success. Since giving a new feature set rotated by LDA to LDA classifier causes biased results, the results of new dimensions generated by LDA as were not applied to LDA classifier on Arcene and Gisette datasets.

The distribution of the Gene Expressin Cancer RNA-Seq data obtained from Fisher Score, which gives the one of highest results in both dimensions, is shown in the figure below. The distribution of Gene Expression Cancer RNA-Seq dataset obtained after rotation by PCA is given in Figure 4.9.

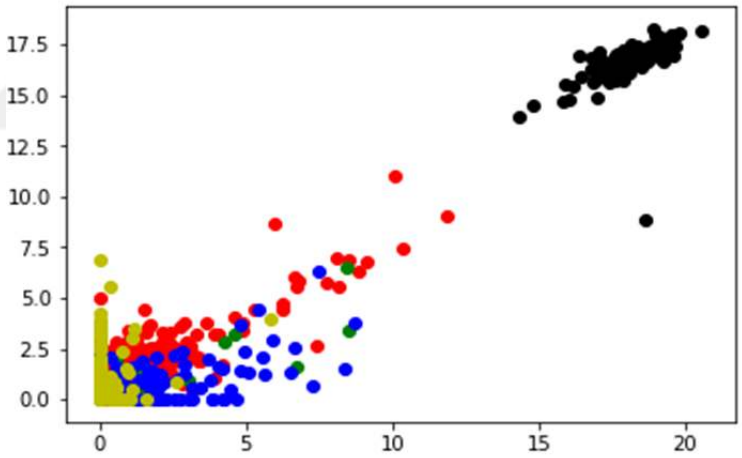


Figure 4.15 Distribution of Gene Expression Cancer RNA-Seq Dataset with The Features Selected by Fisher Score

As can be seen from Figure 4.15, applying feature selection to the Gene Expression Cancer RNA-Seq dataset improves the distribution of data compared to the original dataset. As a matter of fact, f-measure value of 0.99 is obtained by Fisher Score according to distribution in Figure 4.15.

4.1.4. Support Vector Machine

Since pretests on SVM showed that the linear kernel function gives better results than the others, linear kernel function was chosen.

Table 4.17 F-Measure Results of SVM

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		0.90	**0.97**	**1**
InfoGain	#50	0.57	0.89	0.94
	#100	0.71	0.93	0.97
MRMR	#50	0.70	0.74	0.93
	#100	0.81	0.85	0.95
Interact	#50	0.64	0.58	0.94
	#100	0.61	0.64	0.97
Relief	#50	0.66	0.87	-
	#100	0.72	0.92	
ReliefF	#50	0.36	0.91	0.98
	#100	0.49	0.93	**1**
SAM	#50	0.71	0.89	-
	#100	0.66	0.93	
Fisher Score	#50	0.76	0.89	**1**
	#100	0.70	0.92	0.99
Fisher Ratio	#50	0.47	0.13	-
	#100	0.56	0.21	
T-Test	#50	0.76	0.89	-
	#100	0.67	0.92	
Rank Product	#50	0.71	0.91	-
	#100	0.57	0.92	
Gini Index	#50	0.51	0.13	0.95
	#100	0.68	0.92	0.96
PCA	#50	0.74	0.96	0.99
	#100	0.86	**0.98**	0.99
LDA	#50	**1**	0.92	-
	#100	**1**	0.94	
DBN	#50	0.76	0.93	1
	#100	0.75	0.92	0.98

As shown in Table 4.17, it is found that LDA gives the best classification result with 100% success for both 50 and 100-dimensional space for Arcene dataset. In the remaining 13 methods, none of them can achieve the success of classification that is obtained by no feature selection in either dimension. This may have to do with the fact that the support vectors consisting of the features selected on Arcene are not good enough to separate data. We said that the selected features

in Arcene does not improve clearly the classification performance of classifiers. As against no feature selection, the success for 50 dimension decreases from 0.90 to 0.57 for Information Gain, 0.36 for ReliefF, 0.57 for Fisher Ratio and 0.51 for Gini Index. In 100-dimensions the worst result belongs to ReliefF again with 0.49. Studies on SVM show that as the dimensions are increased, the success of SVM gets higher in dimension reduction compared to feature selection methods (Zhu, Zhu and Chen, 2005). Decreasing the success of SVM in all the methods except LDA is due to the fact that the support vectors that is found in new 50 and 100-dimensional space are not sufficient enough to separate data. As discussed in section 3.1.1.1, for a good generalization the data should be at a certain distance to margin as well as being on the correct side of the hyperplane.

The data drawn according to MRMR and LDA, which give the highest results in the classification in SVM on the Arcene dataset, are shown in figures 4.5 and 4.10.

For Gisette dataset, the only method that exceed the 0.97 classification success of no feature selection is PCA with 0.98 in 100-dimensions. Studies show that feature selection and dimension reduction are necessary to reduce the time complexity of SVM. Among them, it is emphasized that using feature selection in SVM that is used linear kernel function preserves the success of baseline. However, in SVM that is used non-linear kernel functions, feature selection causes more information loss and the performance of classifier decreases sharply. On the contrary, use of dimension reduction such as PCA or LDA, leads to more successful results in case of both linear and non-linear kernel functions in SVM. Studies show that the use of dimension reduction for different kernel functions is more suitable than the feature selection. This is proved in this thesis with the results obtained in Gisette dataset (Zhu et al., 2005). Beside that the worst f-measure values belong to Fisher Ratio and Gini Index with 0.13 for 50-dimension, the success of Gini Index surprisingly increases to 0.92 in 100 dimensions. When the results that are obtained in 50 and 100 dimensional space are compared the

success increases for Information Gain from 0.89 to 0.93;increases for MRMR from 0.74 to 0.85;increases for Interact from 0.58 to 0.64;increases for Relief from 0.87 to 0.92;increases for ReliefF from 0.91 to 0.93;increases for SAM from 0.89 to 0.93; increases for Fisher Scoring from 0.89 to 0.92; increases for Fisher Ratio from 0.13 to 0.21; increases for T-Test from 0.89 to 0.92; increases for Rank Product from 0.91 to 0.92; increases for Gini Index from 0.13 to 0.92; increases for PCA from 0.96 to 0.98; increases for LDA from 0.92 to 0.94 and decreases for DBN from 0.93 to 0.92. SAM, Fisher Score and T-Test, which have similar characteristics in terms of selected features, show close results.

The data drawn from Information Gain, PCA and ReliefF, which give the highest results in the classification with SVM on the Gisette dataset, are shown in Figures 4.6, 4.7 and 4.14.

Experimental studies reveal that SVM is the most successful classifier on Gene Expression Cancer RNA-Seq dataset. In case of all the features are regarded in classification SVM produces 100% success. On the other hand, both top 100 features selected by ReliefF and top 50 selected by Fisher Score also cause 100% classification performance. Besides that, classifications that are made with the features selected by other feature selection algorithms also have successful performance with over 93% f-measure results.

4.1.5. Summary

In the experiments, it is seen that no method exceeds a certain threshold on Arcene dataset, except that LDA achieves great success in SVM. The results that are obtain in MLP show that the achievements of these 14 methods stuck 50-60% band including no feature selection. It is explained above that these are related to the distribution of data beside approach of the algorithms. As the classification which is made with the feature selection algorithms that select many common features may produce the similar results, the feature selection methods that have no common feature can also access these scores on Arcene. When the results on

Gisette obtained by MLP are reviewed, the performance of algorithms that can't exceed 0.70 f-measure results such as Information Gain, SAM and ReliefF on Arcene increase to 0.89 even 0.92. When the common features in Gisette are examined, it is seen that the features are more separable than the Arcene dataset, hence feature selection methods produce better results in comparison with Arcene dataset.

When Naive Bayes classifier is analyzed, it is seen that the method that gives the highest success for Arcene data gives the worst results in Gisette, or vice versa. This indicates that not only the approach of a method but also compliance with tested classifier and dataset have effect in the success. The reason why dimension reduction methods give bad results in Naive Bayes is that they produce new values from scratch and they are independent of the values in the original dataset.

In the LDA classifier, similar to the MLP, while the results obtained on Arcene show no difference with no feature selection, and they appear to be stuck on a certain bound, on Gisette dataset their successes generally increase.

In SVM, there is a clear success of LDA on the Arcene that no other method can demonstrate. It is also seen that there is a certain success of PCA on Gisette on 100 dimensions in comparison with other algorithms.

In general, on the Arcene dataset, the projections obtained by dimension reduction methods give better or non-absorptive results in the classifiers except Naïve Bayes due to the inability to separate the features. This is important in terms of showing dimension reduction can be a solution if the feature selection does not work well.

4.2. Tests and Results of Clustering with Existing Methods

4.2.1. K-Means

Table 4.18 F-Measure Results of K-Means

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		0.47	0.73	0.12
InfoGain	#50	0.46	0.84	0.11
	#100	0.49	0.83	0.15
MRMR	#50	0.56	0.45	**0.29**
	#100	0.60	0.56	0.23
Interact	#50	0.47	0.21	**0.29**
	#100	0.47	0.16	**0.25**
Relief	#50	0.62	0.85	-
	#100	0.61	**0.85**	
ReliefF	#50	0.64	0.83	0.12
	#100	0.63	0.82	0.20
SAM	#50	0.68	0.84	-
	#100	**0.68**	0.83	
Fisher Score	#50	**0.69**	0.84	0.05
	#100	**0.68**	0.83	0.04
Fisher Ratio	#50	0.47	0.03	-
	#100	0.44	0.06	
T-Test	#50	0.68	0.84	-
	#100	0.67	0.83	
Rank Product	#50	0.67	0.82	-
	#100	0.67	0.82	
Gini Index	#50	0.46	0.84	0.16
	#100	0.46	0.83	0.15
PCA	#50	0.53	0.68	0.15
	#100	0.45	0.68	0.12
LDA	#50	0.66	**0.86**	-
	#100	0.66	**0.85**	
DBN	#50	0.57	0.61	0.18
	#100	0.60	0.62	0.11

In clustering with K-Means, we approach the issue as a classification problem. In each cluster, we extracted a confusion matrix of data for different classes and rate them with each other. For a certain class, we give the label of that class to cluster that has the most rate of the class in all clusters.

In Table 4.18, it is seen that the best clustering result belongs to Fisher Score for top-50 features with 0.69 on Arcene dataset. Information Gain, Interact, Fisher Ratio and Gini Index cannot exceed the success of no feature selection, that

is 0.47 in top-50 feature space. When the most distant features are removed, in increase of 10% is observed in favor of MRMR between MRMR and Information Gain, 0.56. The closest results, 0.68, to Fisher Score belongs to SAM and T-Test in 50-dimensions. Again, as we see in classification results, the feature selection methods that have many common features provide similar result in K-Means such as Fisher Score, T-Test, SAM; Information Gain and Gini Index. The most successful results are LDA in both dimension among the dimension reduction methods, that is 0.66. SAM and Fisher Score stand out for the top 100 features. The worst results are PCA and Fisher Ratio, that are 0.45 and 0.44.

The data drawn with the top 2 dimensions that have the highest eigenvalues by LDA and top 2 features selected by Fisher Score on Arcene dataset are shown in Figure 4.5 and 4.12.

The best result on Gisette is LDA's 0.86 f-measure, which is 13% higher than no feature selection in 50-dimensions. The difference between PCA, unsupervised generalization of LDA, and LDA is 18%. Another unsupervised dimension reduction method, DBN, is far behind both. The wide difference between Information Gain and MRMR shows that the removal of residual values is negatively reflected in the result. If we look at the number of common features they choose, this is clearly seen. Interact gives the worst result by a huge gap in both dimensions. In general, there is no big gap the algorithms that produce more successful results than no feature selection. Nevertheless, LDA and Relief outshine by the little difference in 100-dimensions.

The data drawn with the top 2 features selected by Relief and top 2 dimensions that have the highest eigenvalues by LDA on Gisette dataset are shown in Figure 4.16 and 4.17.

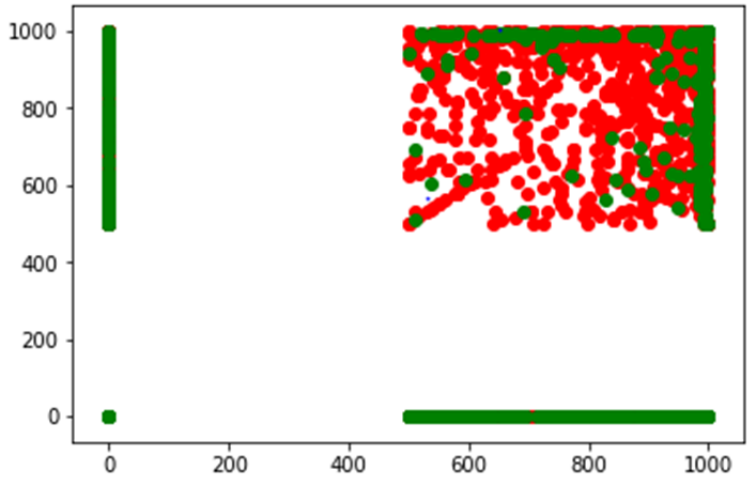


Figure 4.16 Distribution of Gisette Dataset with The Features Selected by Relief

Comparing Figures 4.14 and 4.16, the distribution obtained with the top-2 features selected by Relief and ReliefF is very similar. However, while the f-measure value of LDA classifier according to the features selected by ReliefF is 0.93, the result is 0.85 with Relief in K-Means. This is due to the fact that local minimum problem of K-Means.

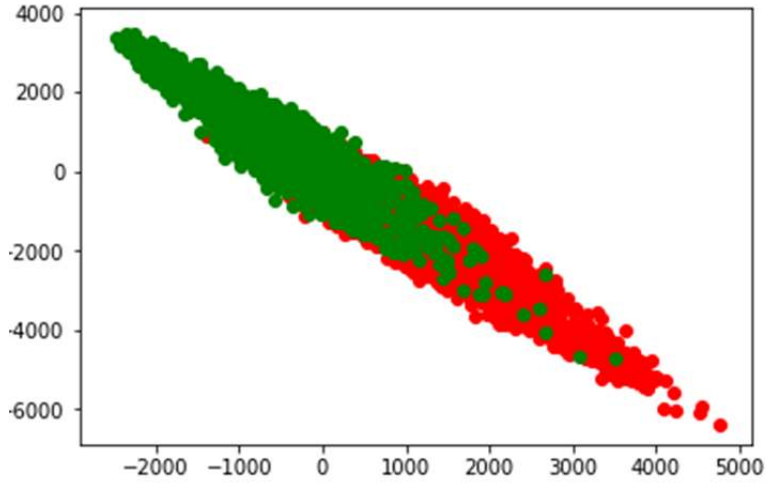


Figure 4.17 Distribution of Gisette Dataset Rotated by LDA

Although there is no perfect classification in K-Means clustering method with the top-2 features that have highest variance and rotated by LDA, in a higher dimensional distribution, classes can be separated by a hyperplane.

The studies on Gene Expression Cancer RNA-Seq dataset reveal that using supervised feature selection methods to an unsupervised machine learning technique causes weak clustering performance as seen in Table 4.18. None of existing feature selection methods can achieve a satisfactory f-measure result for clustering. Nevertheless, the highest f-measure score is shown with the features selected by MRMR and Interact with 0.29. It is seen that it is not considered the redundant features in the selected feature set in favor of MRMR against Information Gain. Again, it can be seen from the results of Gini Index that instead of entropy, a probabilistic selection method has a positive effect of 5% in 50 dimensions against Information Gain. In 100 dimensions, it is understood that their approach does not cause a difference.

As with supervised feature selection methods, it is understood that unsupervised dimension reduction methods such as PCA and DBN have not been successful on the Gene Expression Cancer RNA-Seq dataset. Whereas PCA has maximum 0.15 f-measure success in 100 dimensional, the performance of DBN is 0.18 in 50 dimensions.

The data drawn according to top 2 features selected by Interact is shown in Figure 4.18.

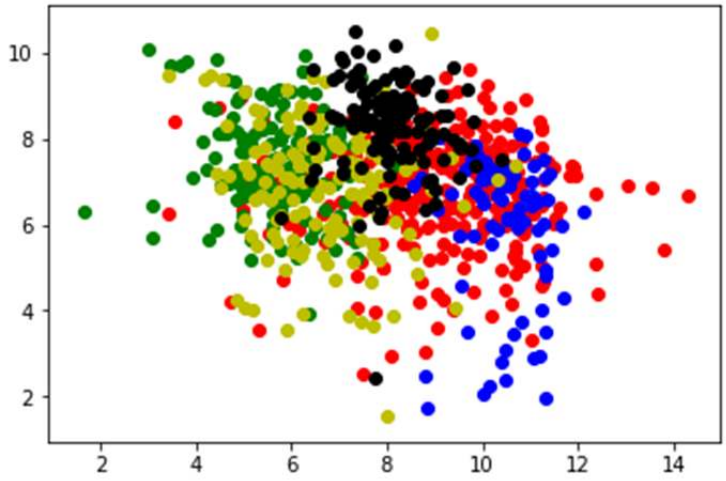


Figure 4.18 Distribution of Gene Expression Cancer RNA-Seq Dataset with The Features Selected by Interact

On Gene Expression Cancer RNA-Seq dataset, no method produces satisfactory results with K-Means. Looking at Figure 4.18, it can be seen that the top-2 features selected by Interact cannot make a successful distinction.

4.2.2. AGNES

Table 4.19 Distance-Based Results of AGNES

		ARCENE	GISETTE	GENE EXPRESSION
NoFS		10560.04	27194.84	266.67
InfoGain	#50	1373.02	2388.95	9.62
	#100	1938.12	3856.38	14.91
MRMR	#50	1148.11	3321.19	6.75
	#100	1655.99	5009.04	11.10
Interact	#50	816.93	2248.37	9.62
	#100	1037.15	2709.53	14.91
Relief	#50	1084.07	2791.08	-
	#100	1491.10	4015.32	
ReliefF	#50	570.64	2964.01	23.87
	#100	749.61	4149.44	38.48
SAM	#50	484.08	2496.18	-
	#100	811.52	3824.06	
Fisher Score	#50	522.34	2306.39	25.08
	#100	813.14	3802.09	34.81
Fisher Ratio	#50	722.11	1461.25	-
	#100	1001.37	1804.07	
T-Test	#50	532.01	2306.39	-
	#100	832.12	3802.09	
Rank Product	#50	476.27	2973.48	-
	#100	701.83	4010.67	
Gini Index	#50	1391.67	2443.76	9.59
	#100	1895	3838.32	14.89
PCA	#50	5434.44	10089.10	183.01
	#100	5758.79	12083.74	188.21
LDA	#50	2107.65	2235.61	-
	#100	2428.40	3773.82	
DBN	#50	**3.66**	**2.62**	**0.17**
	#100	**4.73**	**4.68**	**1.42**

Our success criterion for AGNES is to measure the distance between the last two clusters. The similarities of clusters were measured by singlelink method. Since the least similar clusters are merged last, we think that how small distance between them, the success of method is high. Since the feature selection methods returns the original value of features, they can be compared in terms of distance but to apply this approach to dimension reduction methods causes unreasonable results

because they produce new dimensions from original values. As it is shown in Table 4.19, the most succesfull results for all datasets and dimensions are belong to DBN since it uses sigmoid activation function. Nevertheless, we add these biased results to our study to indicate why such a comparison cannot be made.

If we compare 11 feature selection methods among themselves, the most successful result for Arcene is produced by Rank Product in both 50 and 100 dimensions. In 50-dimensional space, SAM, Fisher Score, T-Test and ReliefF follow Rank Product. It is understood that the order changes as ReliefF, SAM, Fisher Score and T-Test in 100-dimenson.

The highest similarity success on Gisette dataset relates to Fisher Ratio among feature selection algorithms in both dimensions. The results are followed by Interact, Fisher Score and T-Test, Information Gain respectively in 50-dimension. Besides that, this order for 100-dimension is Interact, Fisher Score and T-Test, SAM.

For Gene Expression Cancer RNA-Seq dataset, MRMR gives the best results for both dimensions. Gini Index is superior to Information Gain and Interact with very small differences in both 50 and 100-dimensional space. Entropy-based methods, Information Gain and Interact show the completely same success in both dimensions. The worst results belong to ReliefF and Fisher Score which are far behind of other methods. The feature selection methods that have the same approach by calculating entropy and probability values, Information Gain and Gini Index, lead to be produced very close results in both dimensions. In addition, removing the features that are most distant from the relevant features has a positive effect on the result as seen in MRMR. Another entropy-based feature selection method, Interact also remarks with the similarity between Gini Index in terms of results obtained.

4.3. Results and Discussions with Proposed Hybrid Method

We shared the test results of the most used 11 feature selection methods on Arcene, Gisette and Gene Expression Cancer RNA-Seq datasets in section 4.1. The reason for the most use of these methods is the fact that they are filter based and the calculation speed they promise. In addition, there were other tests on data with 3-dimension reduction methods in the event that feature selection methods cannot separate features. The drawbacks of feature selection approaches and dimension reductions methodologies were discussed, and a hybrid approach is proposed to analyze the effect of unchosen features. 25-dimensional and 50-dimensional PCA projections were combined with ReliefF. In order to prove the efficiency of proposed method, 5 more bioinformatics datasets were utilized during tests. Dataset descriptions are given in Table 4.20.

Table 4.20 Dataset Descriptions

Dataset	# of samples	# of features	# of classes
Arcene	200	10000	2
Gisette	7000	5000	2
Gene Expression	801	20531	5
Colon	62	2000	2
ALLAML	72	7129	2
Leukemia	72	7070	2
Prostate_GE	102	5966	2
Madelon	2600	500	2

Table 4.21 Distribution of Training Data According to Classes

	Class 0	Class 1	Class 2	Class 3	Class 4
Arcene	79	61	-	-	-
Gisette	2466	2434	-	-	-
Gene Expression	208	104	54	99	96
Colon	28	15	-	-	-
ALLAML	32	18	-	-	-
Leukemia	36	14	-	-	-
Prostate_GE	36	35	-	-	-
Madelon	910	910	-	-	-

Table 4.22 Distribution of Test Data According to Classes

	Class 0	Class 1	Class 2	Class 3	Class 4
Arcene	33	27	-	-	-
Gisette	1034	1066	-	-	-
Gene Expression	92	42	24	42	40
Colon	12	7	-	-	-
ALLAML	15	7	-	-	-
Leukemia	10	12	-	-	-
Prostate_GE	14	17	-	-	-
Madelon	390	390	-	-	-

There were some problems encountered with some of the datasets. Prostate_GE dataset is a major challenge for machine learning techniques since the test data are extracted from different experiments, they show about 10-fold difference from the training data. Besides, while the distribution of data to classes is 49-51% in training, this distribution is 26-74% in the test. Therefore, some classifiers simply assign samples to one of the classes. A similar problem exists in

Leukemia and ALLAML data (Canedo et al., 2014). In order to overcome this problem, shuffling was used in all three datasets and the data were tried to be distributed in a balanced manner to the classes during the test and training stages. The distributions of data in the training and test stages are given in Tables 4.21 and 4.22.

As in the previous section, the distribution of data for each bioinformatics dataset are given before the analysis of test results:

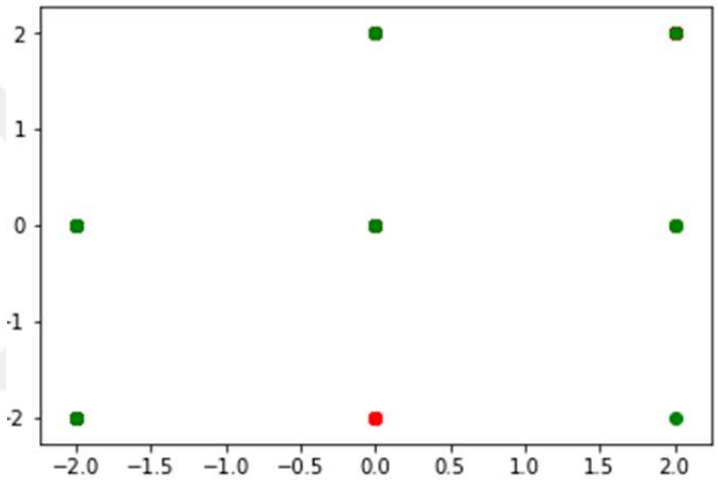


Figure 4.19 Distribution of Colon Dataset

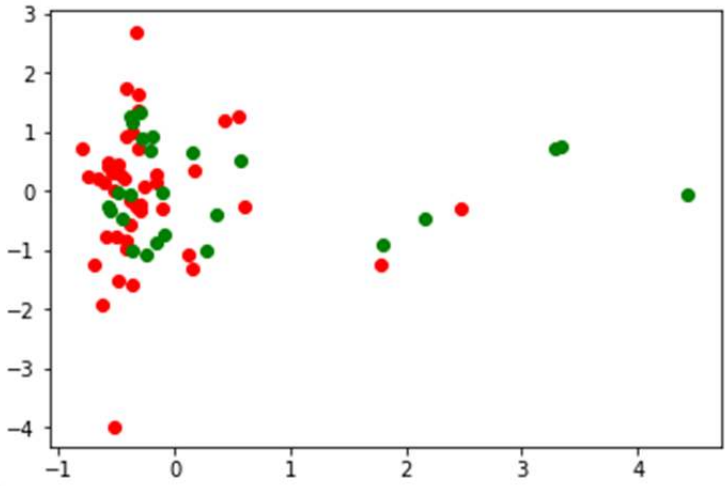


Figure 4.20 Distribution of ALLAML Dataset

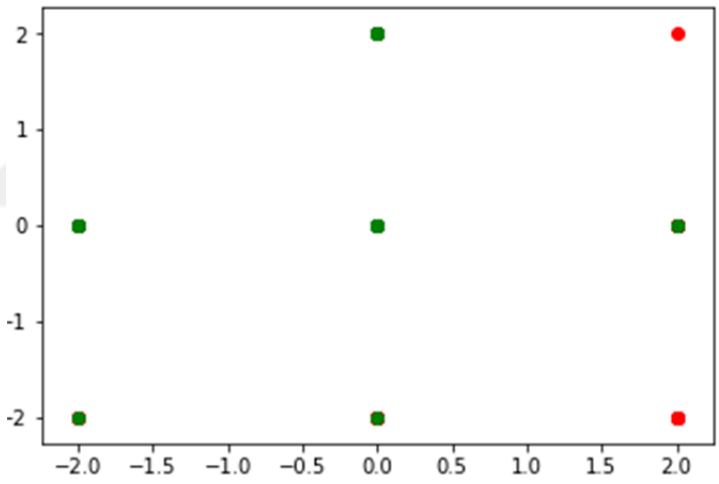


Figure 4.21 Distribution of Leukemia Dataset

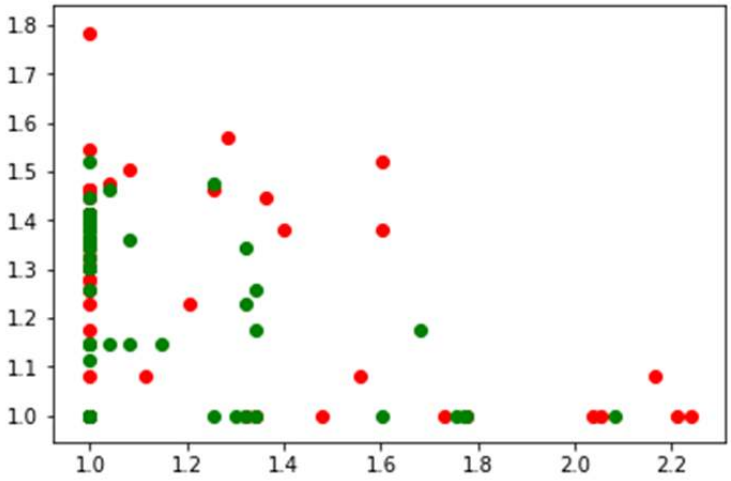


Figure 4.22 Distribution of Prostate_GE Dataset

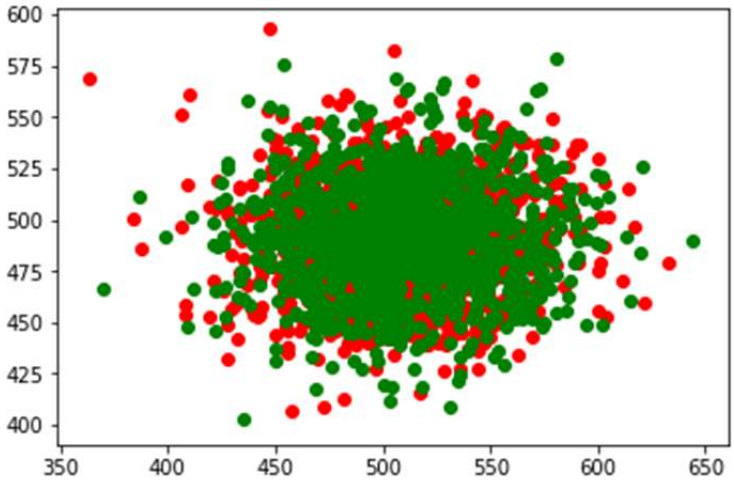


Figure 4.23 Distribution of Madelon Dataset

4.3.1. Tests and Results of Classifiers

4.3.1.1. Multi-Layer Perceptron

Since initial weights affect performances directly, each test was repeated 10 times and the average score was calculated. Used learning rate is 0.01.

Table 4.23 F-Measure Results of MLP

Dataset	NoFS	$ReliefF_{100}$	$ReliefF_{50} + PCA_{25}$	$ReliefF_{50} + PCA_{50}$
Arcene	0.69	0.49	**0.64**	**0.64**
Gisette	0.88	0.80	**0.86**	0.80
Gene Expression	0.11	0.67	**0.99**	0.66
Colon	0.53	0.59	**0.60**	0.54
ALLAML	0.68	0.74	**0.86**	0.73
Leukemia	0.53	**0.75**	0.73	**0.75**
Prostate_GE	0.87	**0.90**	0.87	0.82
Madelon	0.05	0.54	0.63	**0.64**

As a result of our experimental studies, f-measure results obtained in Multi Layer Perceptron neural network are shown in Table 4.23. On Arcene dataset, the f-measure results of no feature selection and ReliefF are 0.69 and 0.49 whereas with the proposed method, the success increases 15% in comparison with ReliefF for both dimensions. It is very promising to provide 15% increase by adding irrelevant features. It is promising to achieve a 64% success with the 25 and 50 redundancy, however the low classification success which is seen in other methods because of the inability to separate the features is also in question here.

It is seen that proposed method achieved the success of ReliefF by using 50 irrelevant features for 100 dimensions on Gisette dataset. When the number of residual features is dropped to 25 the f-measure results of the proposed method increases 6%. Although the success of no feature selection is not captured by neither ReliefF-$ReliefF_{100}$ nor $ReliefF_{50}$+ PCA_{25} and $ReliefF_{50}$+ PCA_{50}, achievement of hybrid method doesn't fall behind of original ReliefF. When it is compared with results of existing methods, our hybrid method is better than 11 of them in the tests that are made with 25 residual features and better than 6 with 50 residual features on Gisette.

The most successful result in the Gene Expression Cancer RNA-Seq dataset is obtained by using the 25-dimensional combination of irrelevant features in the hybrid method, that is 0.99. This is also a result that none of the existing

methods shared in the previous section, can reach. While the success of ReliefF and PCA in 50 dimensions is 0.72 and 0.5, there is a great fit in a ReliefF combination where we send irrelevant features to PCA. When we compare the results of whole feature set and our recommend method beside ReliefF, we see that irrelevant features adversely affect the distribution of related features. When the entire feature set is used, the result is just 0.11, while the ReliefF and 50-dimensional projection of irrelevant features are 0.66.

Tests on Colon dataset show that the features cannot be separated by the feature selection methods as on Arcene. The highest f-measure is obtained by $ReliefF_{50}$+ PCA_{25}, that is 0.60. This value is 7% and 1% higher than no feature selection and $ReliefF_{100}$. Although increasing the number of irrelevant features to 50 reduces the success to 0.54, nevertheless it is still not behind the no feature selection.

The highest result on ALLAML dataset belongs to $ReliefF_{50}$+ PCA_{25} with 0.86 f-measure value. Respectively the success of no feature selection, $ReliefF_{100}$ and $ReliefF_{50}$+ PCA_{50} are 0.68, 0.74 and 0.73. It is seen that when the number of irrelevant features is no longer 25, the success increases as 13%.

Leukemia dataset suffers from dataset shifting and no method can overcome this problem on original dataset. Therefore, shuffling was applied on the dataset. When the original dataset is used, it is seen that no method exceeded 0.19 success on Leukemia. When the dataset was shuffled 1 time, the number of data for the 1st class was 24 in the training and 2 in the test. The results in this case were 0.45 for no feature selection, 0.26 for ReliefF, 0.40 for $ReliefF_{50}$+ PCA_{25} and 0.30 for $ReliefF_{50}$+ PCA_{50}. When the data were shuffled once again, a more balanced dataset was obtained, 14 of the data belonging to 1st class in the training and 12 data in the test as seen Table 4.21 and 4.22. It is seen that the balanced distribution is reflected to results positively. As a matter of fact, the maximum success of 0.19 in the original data is increased to 75% in $ReliefF_{100}$

and$ReliefF_{50}$+ PCA_{50} after shuffling. Although we have doubled the number of irrelevant features now, achieving the success of $ReliefF_{100}$ is promising in terms of the results that we aim at our study.

It has been reported that the training and test data are extracted from the different experiments on the Prostate_GE dataset. This means that a feature marked as important for training is not distinguishable for testing. For this reason, this problem was solved by shuffling the dataset. When evaluating the classification results of MLP, it is observed that f-measure values of Prostate_GE don't fall behind in $ReliefF_{100}$ and $ReliefF_{50}$+ PCA_{25} in comparison with no feature selection even increases by 3% in ReliefF. As expected, 25 extra irrelevant features are found to adversely affect the classification success as against the first 25, that is 0.82. $ReliefF_{100}$ is superior than other two with 0.90 f-measure success.

When the results are analyzed on no feature selection on Madelon, it is understood that feature selection or dimension reduction are required. The success of no feature selection is only 5% and this is too behind of ReliefF and recommended hybrid method. Although the results in 50, 75 and 100-dimensional space are not good enough for a classifier, they are very good in terms of reached point. $ReliefF_{50}$+ PCA_{50} has the highest performance with 0.64 f-measure by increasing the success 59%.

4.3.1.2. Naïve Bayes

Table 4.24 F-Measure Results of Naïve Bayes

Dataset	NoFS	$ReliefF_{100}$	$ReliefF_{50} + PCA_{25}$	$ReliefF_{50} + PCA_{50}$
Arcene	0	0.20	0.46	**0.47**
Gisette	0.07	0.13	**0.14**	**0.14**
Gene Expression	0.03	**0.05**	0.004	0
Colon	0.23	**0.33**	**0.33**	**0.33**
ALLAML	0	**0.28**	0.22	0.11
Leukemia	0.25	**0.44**	0.33	0.26
Prostate_GE	0.68	**0.68**	**0.68**	**0.68**
Madelon	0.52	0.53	**0.54**	0.53

For 100 dimensions, our hybrid feature selection method achieved the highest success with 47% f-measure result in 11 feature selection techniques that we tested previous section on Arcene dataset in Naïve Bayes. This achievement was obtained with only MRMR in 50-dimensions. To increase the success rate from 0 to 47% by adding 50-dimensional projection of residual features is a significant improvement. It is seen that when we add second 50 to ReliefF's top-50 features, f-measure value drops to 20%. Besides, the success we achieve by using 25-dimensional residual features information is better than most of existing methods that we explain in previous section.

Although the success of our method has increased in comparison with no feature selection, ReliefF and most of other methods, that is only 14% for both dimensions on Gisette. Our method doesn't come close to Fisher Ratio and Interact, but it is understood that more than 25 features no longer affect the success of our method on Gisette.

Gene Expression dataset cannot achieve satisfactory f-measure values by any method. Even the most successful method, ReliefF, is able to produce only 0.05 f-measure in Naïve Bayes classifier. It can be said that the unique values of the new 100-dimensional dataset produced result a trouble since Naive Bayes is a probability-based classifier. When the number of common values seen in training

and testing are analyzed, it is seen that almost none of the features have common values. This problem is contrary to Naive Bayes' approach, so it can be attributed to the fact that our method causes to be produced 0 successes.

The results of studies that is done on Colon dataset with the Naïve Bayes classifier suggest that the feature selection affects performance of classifier positively. The success rate of 23% in 2000-dimensional original dataset reaches to 33% with ReliefF, $ReliefF_{50}$+ PCA_{25} and $ReliefF_{50}$+ PCA_{50}. The 50-dimensional representation of residual features doesn't drop the success of Naïve Bayes just like on Gisette.

On ALLAML dataset, best f-measure value belongs to ReliefF for 100 dimensions, 0.28. Although our method falls behind success of ReliefF, it can overcome the faithful of no feature selection in both dimensions, however the results are not successful, that are 0.22 and 0.11. The result of the f-measure obtained by the no feature selection is 0.

The highest f-measure result is obtained by $ReliefF_{100}$ on shuffled Leukemia dataset, 0.44. Unfortunately, no satisfactory classification success has been achieved with any of the methods. While the f-measure result that is acquired by $ReliefF_{50}$+ PCA_{25} is 0.33, $ReliefF_{50}$+ PCA_{50} shows 0.26 f-measure success. As we said that Leukemia suffers from dataset shifting. The experiments on original Leukemia dataset shows that the classification that are made by top 100 features selected by ReliefF makes the f-measure drop to 17% whereas it is 0.48 in no feature selection. This proves that a distinctive feature for the training data does not work in the test. The achievement in original dataset also drops to 25% and 17% in tests that is made by $ReliefF_{50}$+ PCA_{25} and $ReliefF_{50}$+ PCA_{50}. It can be said that changing the order of data positively reflects the classification here.

On Prostate_GE dataset, it is seen that all the approaches produce the same f-measure result, that is 0.68. To add 50 dimensional residual features causes no change in 100 dimensional in comparison with 25 irrelevant features. When no shuffling is done on Prostae_GE, the f-measure result for no feature selection is

0.78, whereas for $ReliefF_{100}$, $ReliefF_{50}$+ PCA_{25} and $ReliefF_{50}$+ PCA_{50} are 0.17. As can be seen from this point, changing the order of samples on the Prostate_GE dataset that suffers from dataset shifting positively affected the classification. As a matter of fact, because the features are selected only through training data, the 10-fold difference between training-test data leads the success of feature selection and dimension reduction to drop 0.17.

On Madelon dataset, the best classification success belongs to $ReliefF_{50}$+ PCA_{25} by a small difference, that is just 1%. It is seen that on each low-dimensional dataset, f-measure value is produced better than the result on the original dataset. Although we increase f-measure result by 2% in our model against 52% success of no feature selection, it cannot exceed 54% on Madelon.

4.3.1.3. Linear Discriminant Analysis

Table 4.25 F-Measure Results of LDA

Dataset	NoFS	$ReliefF_{100}$	$ReliefF_{50} + PCA_{25}$	$ReliefF_{50} + PCA_{50}$
Arcene	0.85	0.60	0.62	**0.65**
Gisette	0.65	0.93	**0.95**	**0.95**
Gene Expression	-	-	-	-
Colon	0.71	**0.66**	0.62	0.53
ALLAML	0.60	**0.83**	0	0.35
Leukemia	0.58	0.47	0.47	**0.52**
Prostate_GE	0.97	**0.80**	0.66	0.74
Madelon	0.57	0.54	**0.57**	**0.57**

As it can be seen in Table 4.16. and 4.25, the dimension reduction on Arcene dataset does not yield better result than the f-measure test results of original dataset in LDA classifier. Our proposed method cannot surpass the 65% threshold in both dimensions but when we compare with ReliefF's 100-dimensional feature subset, it is now satisfactory to reach 65% with irrelevant 50 features.

As it is presented in Table 4.16, the studies that is done on Gisette dataset show that none of feature selection methods could achieve the success of our combined method, that is 0.95 for both dimensions. PCA has 0.95 and 0.96 f-measure values in 50 and 100-dimensional space but this success is achieved by using irrelevant features. It is seen that the dimension reduction that is done by both pure ReliefF and recommended hybrid method reflect in the classification results positively in Table 4.25.

When the Colon dataset is presented to the LDA classifier with all its features, the highest f-measure value is achieved, that is 0.71. Using the best second- 50-dimensional feature set now appears to be better than the 50-dimensional projection of irrelevant features in the LDA classifier on Colon dataset. It is also seen that irrelevant features cause worse classification when the f-measure value obtained with 50-dimensional residual feature set is compared with their 25-dimensional projection in LDA.

On the ALLAML dataset, the 100-dimensional feature set selected by ReliefF achieves 83% f-measure. Unfortunately, with our proposed method, all of the 7 data belonging to the 1st class are found as belonging to the class 0 in the 25-dimensional residual feature projection and therefore the result of f-measure is produced as 0. In the classification that is made with 50-dimensional irrelevant projection, the success has increased to 35% but this is behind the success of both no feature selection and ReliefF.

As in other classifiers in the experiments on the Leukemia dataset, the classification results on the shuffled dataset are more successful than the results obtained using the original data. The most successful result on the original dataset is taken by ReliefF with 42%, no feature selection and the hybrid method we recommend show 0.33 success in both dimensions. The highest f-measure value except no feature selection belongs to $ReliefF_{50}$+ PCA_{50} on shuffeled dataset, that is 0.52. Besides, the success of both $ReliefF_{100}$ and $ReliefF_{50}$+ PCA_{25} is 0.47. The ultimate goal in our study is to achieve the highest success with projection of

the features by regarding their perfect fit. The results that we obtain are hopeful for the future.

The most successful result on Prostate_GE data is obtained by no feature selection, 0.97. While the test results are 0.80 for ReliefF, when the 25-dimensional projection of irrelevant features is added, it becomes 0.66 and with the 50-dimensional residual features are 0.74. Increasing success with more residual features is due to the fact that we now have a perfect fit chance to irrelevant features, such as on the Leukemia dataset. When all the features of the original Prostate_GE data are used, the classification success of the LDA is only 41%. The fact that after shuffling success are doubled is important because it shows the importance of the distribution of data in both training and test.

Although feature selection and dimension reduction on the Madelon dataset generally don't adversely affect the classification, they are not sufficient for successful classification. As seen in other classifiers, the f-measure value obtained in the LDA is fitted at a certain threshold. Nevertheless, with a difference of 3%, the proposed method overrides the original ReliefF, that is 0.57.

4.3.1.4. Support Vector Machine

Table 4.26 F-Measure Results of SVM (linear)

Dataset	NoFS	$ReliefF_{100}$	$ReliefF_{50} + PCA_{25}$	$ReliefF_{50} + PCA_{50}$
Arcene	0.90	0.49	0.73	**0.78**
Gisette	0.97	0.93	0.95	**0.96**
Gene Expression	1	**1**	**1**	0.99
Colon	0.66	**0.66**	0.57	0.16
ALLAML	0.92	**0.80**	0.62	0.40
Leukemia	0.90	**0.90**	0.66	0.57
Prostate_GE	0.97	0.87	**0.88**	0.80
Madelon	0.56	0.53	0.55	**0.56**

The most successful results in the SVM classifier are obtained when the entire feature set is used for each dataset. This is because SVM tends to separate dataset by moving into a high dimensional space. Therefore, the most successful f-measure result on the Arcene dataset is obtained with the original high-dimensional featureset that is 0.90. On Arcene dataset, we already said that the features are not distinctive thus, feature selection methods cannot produce brilliant results in SVM. As a matter of fact, it is seen that the most unsuccessful result is the 100-dimensional data selected by ReliefF with 0.49. However, when the proposed method is compared with ReliefF, success is very high in both dimensions. Now that we have doubled the number of features, but the 78% f-measure success is a proof that SVM works better in high dimensionality. It is also said that turning the irrelevant feature set into a n-dimensional space by applying dimension reduction will cause to be produced more successful results than feature selection (Zhu et al., 2005).

Considering a satisfactory success measure for a very brilliant results are obtained by all methods on Gisette dataset. We approve to compare the method we propose with ReliefF, and in the comparison where 25 residual features are used, our method achieves 2% more success and 3% in 50-dimensional projection that

are 0.95 and 0.96. As in the Arcene dataset, it is seen that ReliefF lags behind the proposed method on Gisette dataset.

It has already seen SVM classify very successfully on the Gene Expression Cancer RNA-Seq dataset in our tests with existing methods. We understand that this success also continues in our proposed method in both dimensions. When the data dimension exceeds twenty thousand, SVM classifies with 100% success as in the 100-dimensional space of ReliefF and 25-dimensional projection of irrelevant features. Regarding the 50-dimensional generalization of residual properties, the success drops to 99%, however when it is compared with success of existing methods and the success obtained with ReliefF's 50-dimensional feature set, our method is still equal or superior than most of them.

On Colon dataset, the highest f-measure value belongs to both no feature selection and pure ReliefF with 0.66. Adding residual features to the Colon dataset has been shown to reduce SVM's classification force. While the success in the classification made with 25-dimensional irrelevant features is 57%, adding a second 25-dimensional residual feature set information leads to a very bad classification, that is 0.16.

While on the original ALLAML dataset, the success of no feature selection is 0.60 and on the shuffled data, it increases to 92%. Similarly, in the 100-dimensional feature set selected by ReliefF from the original dataset, the f-measure value increases from 0.62 to 0.80 as a result of shuffling. The shuffle procedure is found to have a negative effect in SVM while using our proposed method since success in the 25-dimensional residual combination decreases from 78% to 62% in comparison with the original dataset and likewise, in the 50-dimensional projection, whereas the success on the original data is 0.66, after shuffling it drops to 0.40. These results mean that the support vectors chosen by SVM after dimension reduction and shuffling are not the best support vectors for testing on ALLAML dataset.

In contrast to the ALLAML dataset, the shuffling process positively reflects both the existing ReliefF method and the method we propose besides no feature selection on Leukemia dataset. While the success on the original data is only 63% in no feature selection, it increases to 90% as a result of shuffling. Again, on the original dataset, the success of ReliefF increases from 80% to 90% with shuffling in SVM. Although the addition of the projection of irrelevant features increases the success in our proposed method compared with the results obtained from original Leukemia dataset, it lags behind the original ReliefF. Before the shuffling, whereas success is 50% for the 25-dimensional residual combination and 31% for the 50-dimensions, after shuffling, it increases to 62% and 40% for both dimensions, respectively. This is because when ReliefF uses the original values of dataset, we add new dimensions to them in our hybrid method, causing a change in the support vectors to be selected.

As on the Leukemia dataset, the shuffling on the Prostate_GE dataset yields positive results in the SVM classifier for each approach. Given that Prostate_GE dataset suffers from dataset shifting, its success in SVM is increased by shuffling, which is one of our goals. When the entire feature set is used, the f-measure value obtained on the original dataset is 0.78. After shuffling, it inreases to 0.97. In the 100-dimensional feature set chosen by ReliefF, it has a positive effect of 2%, from 80% to 82%. While the success of the new data that have 25-dimensional projection of irrelevant features obtained from the original dataset, is far behind RelieF, after shuffling it becomes reversed. On the original dataset, the success of our method is 0.70, due to shuffling it becomes to 0.88. When we add 50-dimensional residual feature domain, the success of our method increases from 78% to 80% compared to the before shuffling.

On Madelon data, as in other classifiers, all approaches are fitted at a certain threshold. Compared to ReliefF, the success of the proposed method is now 2% higher in 25-dimensional projection of irrelevant features, and 3% higher in 50-

dimensional projection. Moreover, the success of no feature selection is achieved in $ReliefF_{100} + PCA_{50}$ by using both fewer parameters and residual features.

5. CONCLUSION AND FUTURE WORK

In this study, a new hybrid feature reduction technique is proposed by combining dimension reduction and feature selection for bioinformatics datasets as well as other high-dimensional datasets. Arcene, Gisette and Gene Expression Cancer RNA-Seq datasets are used for testing the existing methods. In order to test the validity of the tested methods, they are tested with 2 clustering methods and 4 classifiers that have different approaches. The success of the tested methods is evaluated by using f-measure criteria.

In the first stage of this thesis, the results of totally 15 techniques, including no feature selection results are compared according to the features on 6 different approaches. Correspondingly, it is analyzed that which feature selection and dimension reduction methods with various approaches are more effective for solving the problem.

The comparative analysis shows that as well as the feature selection and dimension reduction methods, the characteristics of datasets also affect the classification and clustering results. Dimension reduction is found to be more effective in classification and clustering according to the plotting of the distribution obtained on datasets that cannot be separated by feature selection.

It is also seen that different classifiers may cause various classification results with the same feature set. While classifiers such as SVM, LDA and Naive Bayes tend to reflect the characteristics of the data, the results of MLP classifier depends on qualification of initial weights. The local minimum concern is the major problem of the MLP, although selected feature set have high separation power of classses. As a matter of fact, while the SVM classifies the distribution obtained by LDA dimension reduction method on Arcene dataset with 100% success, the f-measure result with the same feature set is is only 0.73 in MLP.

The disadvantage of all feature selection methods we test is that they ignore the correlation between the features. Therefore, they evaluate the features

individually. However, the possibility that low-scored features show perfect-fit by combining is not taken into account. As a matter of fact, the results obtained with the best 100 features are found to be more successful than the results that are obtained with the top-50 features.

In this study, an improvement on feature selection algorithm is aimed to utilize features marked as irrelevant. For this purpose, we combine ReliefF and PCA methods. It is seen that the results obtained with irrelevant feaures are more successful in some classifiers than the results of original feature selection method ReliefF. Further studies can be done on developing a new feature selection method based on artificial neural network or machine learning.

REFERENCES

Alpaydın, E. (2017). Yapay Öğrenme. İstanbul, Turkey: Boğaziçi Üniversitesi Yayınevi.

Canedo, V.B., Marono, N.S., Betanzos, A.A., Benitez, J.M., Herrera, F. (2014). A Review of Microaarray Datasets and Applied Feature Selection Methods, Elsevier, 292, 111-135.

Çoban, Ö., Özyıldırım, B.M., Özel, S.A. (2018). An Empirical Study of the Extreme Learning Machine for Twitter Sentiment Analysis. International Journal of Intelligent Systems and Applications in Engineering, 6(3), 178-184.

Dat, T.H., Guan, C. (2007). Feature Selection Based on Fisher Ratio and Mutual Information Analyses for Robust Brain Computer Interface. IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP-07', pp. 337-340.

Debie, E. and Shafi, K., (2017). Implications off the Curse of Dimensionality for Supervised Learning Classifier Theoretical and Emprical Analyses. SINGH, S.(Ed.), Pattern Analysis and Applications(p.1-18) in. New York, USA.

Dietterich, T. (1995). Overfitting and Undercomputing in Machine Learning. ACM Computing Surveys, 27(3), 326-327.

Domingos, P. (2012). A few Useful Things to Know About Machine Learning. Communications of the ACM, 55(10), 78-87.

Gu, Q., Li, Z., Han, J. (2011). Generalized Fisher Score for Feature Selection. Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, pp. 266-273.

Guyon, I., Gunn, S., Hur, A.B., Dror, G. (2003). Result Analysis of the NIPS 2003 Feature Selection Challenge. NIPS'04 Proceedings of the 17th International Conference on Neural Information Processing Systems, pp. 545-552.

Han, J., Kamber, M., Pei, J. (2012). Data Mining. Concept and techniques. Waltham: Morgan Kaufmann.

Jauregui, J. (2012). Principal Component Analysis with Linear Algebra. Acces Adress: http://www.math.union.edu/~jaureguj/PCA.pdf

Kira, K., Rendell, L.A. (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm. AAA'92 Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 129-134.

Koller, D., Sahami, M. (1996). Toward Optimal Feature Selection. Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, pp. 284-292.

Koziol, J.A. (2010). The Rank Product Method with Two Samples. Elsevier, 584(21), 4481-4484.

Lal, T.N., Chapelle, O., Weston, J., Elisseeff, A. (2006). Embedded Methods. Guyon, I., Gunn, S., Nijravesh, M.(Ed.), Feature Extraction Foundations and Applications(p.137-167) in. Netherlands: Springer.

Langley, P., Sage, S. (1994). Induction of Selective Bayesian Classifiers. UAI'94 Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence, pp. 399-406.

Leung, K. M. (2007). Naive Bayesian Classifier. Polytechnic University Department of Computer Science / Finance and Risk Engineering. Lecture Notes.

Liu, H.C, Peng, P.C, Hsieh, T.C, Yeh, T.C, Lin, C.J, Chen, C.Y, Hou, J.Y, Shih, L.Y, Liang, D.C (2013). Comparison of Feature Selection Methods for Cross-Laboratory Microarray Analysis. IEE/ACM, Transactions on Computational Biology and Bioinformatics, 10(3), 593-604.

Mallick S., (2017). Bias-Variance Tradeoff in Machine Learning. Access address: https://www.learnopencv.com/bias-variance-tradeoff-in-machine-learning/

Marono, N.S., Betanzos, A.A., Sanroman, M.T. (2007). Filter Methods for Feature Selection: A Comperative Study. Proceedings of the 8th International

Conference on Intelligent Data Engineering and Automated Learning, pp. 178-187.

Mayoraz, E., E. Alpaydın. 1999. “Support Vector Machines for Multiclass Classification.” *Foundations and Tools for Neural Modeling, Proceedings of IWANN’99, LNCS 1606, ed. J. Mira, J.V. Sanchez-Andres, 833-842. Berlin-Springer.*

McAfee, L. (2008). Document Classification Using Deep Belief Nets.

Mitchell, T.M. (1997). Decision Tree Learning. Mitchell, M.(Ed.), Machine learning(p.52-75) in. New York, USA:McGraw-Hill.

Narayan, S., Tagliarini, G. (2005-July). An Analysis of Underfitting in Mlp Neural Networks. Proceedings of International Joint Conference on Neural Networks, pp. 984-988.

Patil, T.R, Sherekar, S.S. (2013). Performance Analysis of Naïve Bayes and J48 Classification Algorithm for Data Classification. International Journal of Computer Science and Applications, 6(2), 256-261.

Platt, J. 1999. “Probabilities for Support Vector Machines.” Advances in Large Margin Classifiers, ed. A.J. Smola, P. Bartlett, B. Schölkopf, D. schuurmans, 61-74. Cambridge, MA: MIT Press.

Salama, M.A., Hassanien, A.E., Fahmy, A.A. (2010). Deep Belief Network for Clustering and Classification of a Continuous Data. ISSPIT’10 Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology, pp. 473-477.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929-1958.

The Cancer Genome Atlas Research Network, Weinstein, J.N, Collisson, E.A., Mills, G.B., Shaw, K.R., Ozenberger, B.A., Ellrott, K., Shmulevich, I, Sander, C., Stuart, J.M. (2013). The Cancer Genome Atlas Pan-Cancer Analysis Project. Nature Genetics, 45(10), 1113-1120.

Tong, S., Koller, D. (2001). Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 2,45-66.

Underfitting, https://www.datarobot.com/wiki/underfitting/

Vapnik, V. (1998). Statistical Learning Theory. New York: Wiley.

Xing, E.P., Jordan, M.I, Karp, R.M. (2001). Feature Selection for High-Dimensional Genomic microarray Data. Proceedings of Eighteenth International Conference on Machine Learning, pp. 601-608.

Zeng, X., Wang, Q., Zhang, C., Cai, H., (2013). Feature Selection Based on RelieF and PCA for Underwater Sound Classification. 3rd International Conference on Computer Science and Network Technology, pp. 442-445.

Zhu, M., Zhu, J., Chen, W. (2005). Effect Analysis of Dimension Reduction on Support Vector Machines. International Conference on Natural Language Processing and Knowledge Engineering, pp. 592-596.

CURRICULUM VITAE

Barış DİNÇ was born in Adana, in 1993. He went to elementary school at Yunus Emre Primary Education School in Adana. He completed his high school education at Ramazan Atıl High School. He graduated from the Department of Computer Engineering, Çukurova University, Adana, in 2017. He started working at Adana Alparslan Turkes Science and Technology University as Research Assistant in 2019.