MAY 2017 M.Sc. in Electronics and Computer Engineering ABAS ISMAEL SILO ALI

UNIVERSITY OF GAZİANTEP

GRADUATE SCHOOL OF

NATURAL & APPLIED SCIENCES

BAGGING CONSTRAINT LAPLACIAN SCORE FOR EFFICIENT SEMI- SUPERVISED FEATURE SELECTION

M. Sc. THESIS

IN

ELECTRONICS AND COMPUTER ENGINEERING

BY

ABAS ISMAEL SILO ALI

MAY 2017

Bagging Constraint Laplacian Score for Efficient Semi- Supervised Feature Selection

M.Sc. Thesis

in

Electronics and Computer Engineering

University of Gaziantep

Supervisor

Assoc. Prof. Dr. Sema Koç KAYHAN

by

Abas Ismael Silo ALI

May 2017

REPUBLIC OF TURKEY

UNIVERSITY OF GAZİANTEP

GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES

ELECTRONICS AND COMPUTER ENGINEERING

Name of the Thesis: Bagging Constraint Laplacian Score for Efficient Semi Supervised Feature Selection

Name of the student: Abas Ismael Silo ALI

Exam date: May 2, 2017

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ahmet Necmeddin YAZICI
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERÇELEBİ
Head of department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Sema Koç KAYHAN
Supervisor

Examining Committee Members:	Signature
Assoc. Prof. Dr. Sema Koç KAYHAN	.....
Assoc. Prof. Dr. Kemal DELİHACIOĞLU	.....
Assist. Prof. Dr. Tolgay KARA	.....

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

Abas Ismael Silo ALI

ABSTRACT

BAGGING CONSTRAINT LAPLACIAN SCORE FOR EFFICIENT SEMI-SUPERVISED FEATURE SELECTION

ALI, ABAS ISMAEL SILO
M.Sc. in Electronics and Computer Engineering
Supervisor: Assoc. Prof. Dr. SEMA KOÇ KAYHAN
May 2017
45 Pages

In this study, we present efficient and robust approach for semi-supervised merit choices, depending on Constrained Laplacian Score (CLS). The main obstacle of this method is that you have a few options to choose supervision information, which is given by pairwise constraints. In fact, constraints are defined to calculate the noise that causes the decline of learning performance. In this study, we aim to overcome the performance-reducing effects of resource constraints. This is achieved by resampling the data (bagging) and using the random material strategy technique. For the experimental study, the high-dimensional datasets are supported to exhibit the performance of the proposed approximation, and to compare it with other representative methods.

Key words: Bagging, Constraint Score, Laplacian Score, Feature selection.

ÖZET

VERİMLİ YARI DENETİMLİ ÖZELLİK SEÇİMİ İÇİN TORBALAMA KISITLAMALI LAPLAS SKORU

ALI, ABAS ISMAEL SILO
Yüksek Lisans Tezi, Elektronik ve Bilgisayar Mühendisliği
Danışman: Doç. Dr. SEMA KOÇ KAYHAN
Mayıs 2017
45 Sayfa

Bu çalışmada, yarı denetimli seçimler için Kısıtlamalı Laplas Skoru(CLS)'na bağlı olarak, verimli bir çözüm önerisinde bulunuyoruz. Bu yaklaşımın en büyük engeli, çift yönlü kısıtlama ile verilen denetim bilgilerinde birkaç seçeneğin olmasıdır. Aslında, kısıtlamalar, öğrenme performansının çöküşüne sebep olan gürültüyü hesaplamak için tanımlanmıştır Bu çalışmada, kaynakların varyasyonuyla ortaya çıkan kısıtlamanın performans etkilerini aşmaya çalışıyoruz. Bu çalışma, verilerin yeniden örneklenmesi (torbalama) ve rastgele madde stratejisi tekniği kullanılarak yapılmıştır. Yüksek boyutlu veri kümeleri deneylerde kullanılarak önerilen yaklaşımın performansı gösterilmiş ve diğer önde gelen yöntemlerle karşılaştırılması sağlanmıştır.

Anahtar Kelimeler: Torbalama, Kısıtlama Skoru, Laplas Skoru, Özellik Seçimi.

To the “values” that were always “relevant”

And never “redundant” in my life

My parents, my sisters, my brothers

ACKNOWLEDGEMENTS

First of all, I am grateful to The Almighty God for establishing me to complete this work.

Special thanks to Assoc. Prof. Dr. Mohammed HINDAWI who help to during my study and give me a great knowledge.

In addition, I would thank the place that I work there the treasury of Duhok city for their support and giving me the time to finish this work.

Secondly, I would like to thank a person how always with me during my work and helped my establishing new dream (Zinah Salim).

Finally, I would also like to thank my parents, my sisters, brothers and my friends; Alan Nazar, Khalid Rasheed, Ali Ehsan, Raad Shareef, Qahraman Ahmed, Zaharadeen Babagana, who helped me a lot in finalizing this work within the limited period.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

MLC	Must-link Constraint
CLC	Cannot-link Constraint
CS	Constraint Score
LS	Laplacian Score
CLS	Constraint Laplacian Score
FS	Feature Selection
BCS	Bagging Constraint Score
BCLS	Bagging Constraint Laplacian Score
K-N-N	K-Nearest-Neighbor
RSM	Random Subspace Method
BAGGING	Bootstrap Aggregating

CHAPTER ONE

INTRODUCTION

1.1 Background

Feature selection is finding of the significant feature, meaning a huge dimensional data, in a task perform with the selection of features in a data, with intention of finding a small subset of great importance [1]. In considering, the numerous influences derived from the feature selection, as for example “decrease” measures, storages request and dimensionality of challenging [1]. These datasets consist of digital images expression of the gene, series of finical time [2]. In addition, the benefit is derived by removing irrelevant and redundancy in the features, in the data accurate rate in learning of data, improve the learning comprehension [2]. Ideally, the features selection is a phenomenon used to select the data with most relevant features, which is some line referred as dimensionality reduction; in a variety of Context [3]. The methodologies employed in this technique are:

- Filter method [4]
- Wrapper method [5]

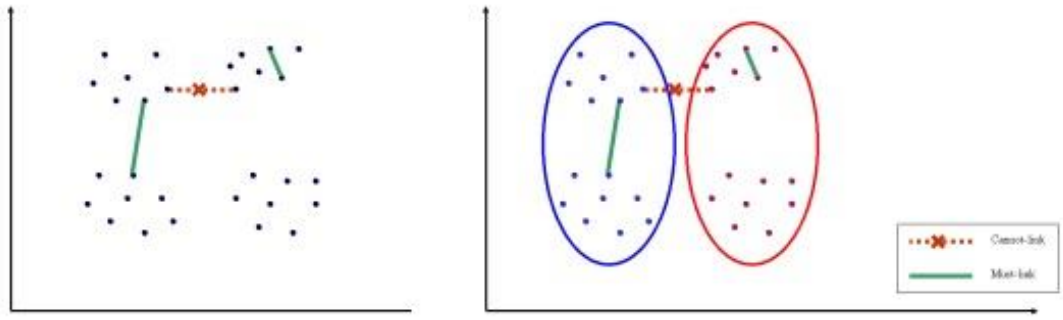
In these two methods, the filter method emphases on data analysis evaluate, in terms of relevant features using behaviors of training data independent on several learning algorithms. While on the other method, the wrapper method uses the direct pre-determine algorithms that are learned to analyses the features between the two, the wrapper method is performed very excellently compared to filter method in terms of accuracy, but computationally, costly. It is noted that when dealing with a very large number of features, the filter method is much more required to be employed [2]; because of its reliability and efficiently.

The important role of feature selection is determined by depending on the context; such as Supervised, Unsupervised, and Semi-supervised feature selection.

In Supervised feature selection, the dataset instances are considered as labeled and its relevant is measured according to its correlation with label information [3]. Unsupervised feature selection label and its relevant measured according to its ability in preserving from behaviors of learning data [6]. Lastly, the semi-supervised feature selection acquiring some information to the certain label of data that leads to getting some sort of small-labeled samples problem [3, 6].

The supervised outperform unsupervised, especially if the labeled data are enough provided, although getting a labeled data is very costly, while intervene of human professionals makes it less efficient. So to curtail these sorts of problem, semi-supervised feature selection technique is employed, which will ultimately handle both labeled and unlabeled data.

The beneficial usage of the two type data, labeled and unlabeled happens by making changes to their appearance into the pairwise constraint that is remarkably identified in two subsets (Must-links instances constraint) and (Cannot-links instances constraint). Must-links constraint; in this subset, the instance of data has the same label, while Cannot-link constraint; is the instance that does not has the same label. The technique has been employed perfectly and shows the improvement, while the labeled part of data has been included for instance; in the diagram below:



(a) (b)

Figure 1.1 (a) Input instances and constraints (b) A clustering that satisfies all constraints [7].

Notably, in filter method technique, we observe the method undertook the method as pre-processing phases, without yielding any algorithm to perform the computation

(induction algorithm) [8]. Purposely the feature selections of training datasets are learned such as the distance between the classes.

This method is quite speedy when compared with the wrapper method and make an independent algorithm for the whole phases, but in terms of subset selection, it chooses a maximum number of features [9].

A part from these, a minor training size in the high dimensionality of data, it is quite hard to establish its classes. Mostly every small size yielded a biased with huge variance in term of classifier features, are not evaluated well. This yield in very performance as a weak classifier, which lately may cause changes with the dataset. However, a poor performance; witness are caused reasons such as setting of classifier parameters, false assumption, low complexity, but with intention of improving the weak classifier, several approaches are employed which include:

- Noise injection.
- Regularization method.

The efficient one is when we construct several weak classifiers other than one and merge them as with good efficiency, and subsequently the techniques used for combining the weak classifier which called "Bagging" others are boosting, and random subspace technique, due to our consideration work on this thesis, we are interested to further enroll the Bagging, constraint the method in questioned uses the training dataset is came up with unique bootstrap, that has been generated randomly, and it is a new learning set. The influential significances situation for beneficial ensemble learning technique is to integrate the models that differ from each other by employing two important phenomena.

1. RSM: the RSM [10] is constructing numerous classifiers for a cause of high dimensionality issues by using each classifier on different subsets. The strength may effect; first is by using the random subsample of the training data and the second is by randomization of the algorithm in learning base-level classifier, which is a feature at each set of tree formation and selects the best among them. In this phase, it randomly chose the subset of the feature from the bagging for a deterministic category in the base level algorithm.

This is the same process as it is in the decision tree of random forest [11]. Nevertheless, it is even important than random forest because it is more applicable in many base level without concerning of any random in any form. It may benefit in using completely random subspace in constructing a classifier and aggregating it. In addition, if the digit is comparatively less than the size of the information then one may solve the issue as some subspace dimensionality is smaller than the original tree (feature space) and the number of training sets remains the same.

Then even the data set has many redundant features, one can obtain a better classifier in random space than the original feature space. It is quite expected the RSM to work better and efficient when there is redundantly in the data set.

2. Bootstrapping method [12]: where the varieties are ultimately maintaining.

It is so important to note that an individual learner who produce a meagless result as a learning algorithm breakdown with high dimensionality of data. However, in ensembles learning phenomena, numerous components of learners are then come together to produces a good result.

Considering a better way of boosting and RSM is actually to examine the relationship between the sample training data and small size properties of such classifier, for understanding the weak classifier. It is significantly and thoroughly investigated between the three techniques; Bagging, Boosting, and RSM by many investigations and the result show that they are good for regression and the classification trees. They also may be considered for the k-nearest neighbor classifier, and note that the boosting gives better performance, then bagging then RSM. It is categorically elaborated that both the bagging and boosting are used in unstable data, they used for perceptron with RSM may serve to function in k-nearest neighbor classification rules. The most significant linear classifier are the nearest mean classifiers 'NMC' [13] and the second significant linear classifier is fishers linear discriminants functions [14], however, when the number of training object is smaller than the dimensionality of the data, then the easy estimate of the covariance matrix is quite singular. Nevertheless, in this work, we will restrict to the significances reflected by bagging constraints and the emphasized work should be reflected.

1.2 Statement of the Problem

Feature selection is an important task in data mining and machine learning processes. This is well known in both supervised and unsupervised contexts. The semi-supervised feature selection is still under development and far from being mature. In general, machine learning has been well developed in order to deal with partially labeled data. Thus, feature selection has obtained special importance in the semi-supervised context. One of the promising methods for dealing with semi-supervised feature selection is the Constraint Laplacian Score (CLS).

The main drawback of this method is the choice of the little supervision information, represented by pairwise constraints. In fact, constraints are proven to have some noise, which may deteriorate the learning performance.

1.3 Purpose of Thesis

The goal of this thesis is to override any negative effects of constraints set by the variation of their sources. This is done by an ensemble technique using both a resampling of data (Bagging) and random subspace strategy, then removing any redundancy in selected feature.

1.4 Importance of the Thesis

The application of the feature selection help in improving the performance of learning algorithms, such algorithms can be applied in various ranges of applications starting from the economical filed to the health domain and passing by any domain where the learning of the machine or the dealing with huge amount of data become emergent.

This thesis is divided as follow:

- The second chapter of this work (Thesis) will describe the review of the related works.
- Chapter three explains the purpose of this work.
- Chapter four provides the result of work.
- Chapter five is the conclusion and recommendation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Background

This section will review some feature selection methods and semi-supervised feature selection and also bagging.

2.2 Feature Selection

Feature selection as defined by Daoqiang et. al, is an important preprocessing step in high dimensional data. It is a selection method for the supervised feature with supervision information. It is very superior to unsupervised ones without supervision information [2], it is also referred as variable selection or attributes selection [15].

Feature selection is the process of selecting a subset of relevant features for use in model construction [17], and has diversity terms (Variable selection or Attributes selection). It is the automatic selection of attributes in one’s data e.g., columns in tabular data that are most relevant to the predictive modeling problem that one's is working on [16].

However, Feature selection mostly works as a filter, muting out features that are not useful besides to your existing features [20]. Feature selection is generally different from dimensionality reduction, both methods try to reduce the number of attributes in the dataset, but in the side of dimensionality reduction method does so by creating new combinations of attributes, while in the feature selection methods side it just include and exclude attributes present in the data without doing any change to the data [18]. Principal component analysis and singular value decomposition, both of them are examples of dimensionality reduction methods [19].

2.2.1 The problem that feature selection solves

Feature selection methods can used to specify and delete a useful, irrelevant and redundant attribute from data that do not participate in the accuracy of a predictive model or to decrease the accuracy of the model [21]. They help to make an accurate predictive model by selecting features that will give better accuracy whilst requiring fewer data [22], fewer attributes are desirable since they reduce the complexity of a model, and the model is simple, however, it be easy to understand and explained [15].

The variable selection's aim is three-fold: supplying faster and more cost-effective predictors, improving the prediction performance of the predictors, and supplying a better understanding of the underlying process that generated the data [15].

2.2.2 Feature selections algorithms

Filter methods, wrapper methods and embedded methods are three general classes of feature selection algorithms.

a) Filter Method

The first method is the filter feature selection methods; it applies a statistical measure to allocate a scoring to each one of the features. Each feature is ranked by the score and either chosen to be kept or deleted from the dataset. The methods are often univariate and view the feature with regard to the dependent variable or independently [4]. Information gain, correlation coefficient scores, and the Chi-squared test are some examples of filter methods [22].

b) Wrapper Method

Wrapper methods regard the selection of a set of features as a search problem, where various combinations are prepared, assessed and compared to other combinations. A predictive model is used to assess a combination of features and allocate a score on the basis of model accuracy [5].

The search problems process can stochastic-like random hill-climbing algorithms, or it can be methodically like a best-first search, or it can use heuristics, such as

backward and forward passes to ads and deletes feature, the recursive features eliminations algorithms is an example of a wrapper method [23].

c) Embedded Method

Embedded methods learn which features better participate in the accuracy of the model while the model is being created. Regularization methods are the most common kind of embedded feature selection methods [24]. This kind of embedded feature selection method (regularization methods) are also referred to penalization methods that introduce extra constraints into the optimization of a predictive algorithm e.g. regression algorithm, that bias the model toward fewer coefficients (lower complexity) [25]. Elastic NET, the LASSO and Ridge Regression are some examples of regularization algorithms.

2.3 Semi-supervised feature selection

Supervised and unsupervised feature selection methods require quantifying feature pertinence, but in various ways. Along these lines, the key for making a powerful semi-supervised feature selection algorithm is to evolve a framework, under which the pertinence of a feature can be assessed by both labeled and unlabeled data normally [26]. The clustering assumption defined as a base assumption for most semi-supervised learning algorithms. It supposes, "If points are in the same cluster, they are likely to be of the same class" [27].

We suggest semi-supervised features selections algorithms; (s Select). We initially transfer a features vectors F_i into a cluster signal, thus each on of element F_{ij}, (j = 1, 2, 3, . . ., n) of F_i signify the affiliation of the corresponding instance X_j. the cluster signify 's fitness's can be assessed by two factors:

First factor: consistency - whether the cluster structures formed is consistent with the given label information.

Second factor: separability - whether the cluster structures formed are well separable [28], the best case is all labeled data in each cluster coming from the same class.

Assume F and F‘ are two feature vectors, G and G’ are the corresponding cluster indicators, the cluster structures made by g and G’. Those formed by g are preferred, compared with the cluster structures formed by G’. Both cluster indicators constitute clearly separable cluster structures, from the unlabeled data point of view. Nevertheless, the cluster structure formed by G appear to be more consistent, when the label information is considered because all labeled data in a cluster are from the same class, Under the clustering assumption, G suits the data best than G’. Suggesting that the feature corresponding to the feature vector F’ is less relevant to target concept than the feature corresponding to feature vector f [26].

2.4 Feature selection methods

There are many feature selection techniques in the following three of these techniques will be reviewed:

2.4.1 Constraint Score (CS)

Daoqiang et. al, formulate the pairwise constraint feature selection as follows: Given a set of data samples X= [X_1,X_2,…,X_m] and some supervision information in the form of pairwise must-link constraints:

M = {(x_i, x_j)|x_i and x_j are in the same class}

Pairwise cannot-link constraints, C = {(xi , xj)|xi and xj belong to different classes}, to find the most relevant feature subsets from the original n features of X, use the supervision information in pairwise constraints in C and M.

Let f_{ri} indicate the rth feature of the ith sample x_i ,i=, i=1,…,m; r =1,…,n. To assess the score of the r-th feature using the pairwise constraints in M and C, we identify 2 different score functions by minimizing C_r^1 and C_r^2 :

$$C_r^1 C_r^1 = \frac{\sum(xi,xj)\in M(fri-frj))^2}{\sum(xi,xj)\in C(fri-frj)2} \quad (1)$$

$$C_r^2 = \sum_{(xi,xj)\in M}(fri - fri)^{2-\gamma} \ \sum_{(xi,xj)\in M}(fri - fri)^2 \ . \quad (2)$$

The intuition of equations (1) and (2) is simple and normal. This means that we want to choose features with better constraint preserving ability. More realistic, if there is

a CLC between two data samples, a ‘good’ feature should be the one on which those two samples are far away from each other; on the other side, if there is an MLC between two data samples, a ‘good’ feature should be the one on which those two samples are close to each other; both eqs. (1) And (2) realize FS in accordance with the features constraint preserving ability. There is a regularization coefficient (λ) whose function is to make a balance between the contributions of the (2) terms in Eq. (2). We set λ <1, due to the distance between samples in the same class is typically smaller than that in different classes.

We refer to feature selection algorithms based on the score functions in eqs. (1) and eqs. (2) as CS. To be more specific, we refer to the algorithm using Eq. (1) as CS-1, and to the algorithm using Eq. (2) as CS-2. The whole procedure of the proposed CS algorithm is abstracted in the algorithm (1) as follows:

Algorithm 1: (CS)

Input: Dataset B, pairwise constraints set C and M, λ (for CS-2 only)

Output: the rated features list

Step one: for each of the N features, compute its constraint score using Eq. (1) (for CS-1) or Eq. (2) (for CS-2).

Step two: the features are rate according to their CS from small to big rate.

We can also provide an alternative clarification on the CS functions in both Eq. (1) and Eq. (2) from the spectral graph theory [29].

First, we form two graphs GC and GM both of them with m nodes, using the pairwise constraints in C and M, respectively. In these 2 graphs, the i-th sample xi corresponds to the i-th node. For graph GM, we place an edge between these 2 nodes i and j if there is a CLC between samples xi and xj in C, we place an edge between these 2 nodes i and j if there is a MLC between samples xi and xj in M. Similarly, for graph Gc, once the graphs GM and GC are formed, their weight matrices, indicated by SC and SM respectively can be defined as:

$$S_{ij}^{M} = \begin{cases} 1 & if\ (xi, xj) \in M\ or\ (xj, xi) \in M, \\ 0 & otherwise, \end{cases} \tag{3}$$

$$S_{ij}^{C} = \begin{cases} 1 \;\; if \; (xi\,, xj\,) \in C \; or \; (xj\,, xi\,) \in C, \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise, \end{cases} \tag{4}$$

fr = $[fr_1, fr_2, \ldots, fr_m]^T$ and let H^C and H^M be the diagonals matrices with

$D_{ii}^{M} = \sum_j S_{ij}^{M}$ *and* $D_{ii}^{C} = \sum_j S_{ij}^{C}$ Then, count the matrices of laplacian [2].

$L^M = D^M - S^M$, LC = DC – SC.

as indicated by Eq. (3) and Eq. (4) we get

$$\sum_{(xi,xj)\in M} (fri - fri)^2 = \sum_{i.j} (fri - fri)^{2}_{S_{ij}^{M}}$$

$$= \sum_{i.j} (f_{ri}^2 + f_{rj}^2 - 2 f_{ri} f_{rj})^2 S_{ij}^{M}$$

$$= \sum_{i.j} f_{ri}^2 S_{ij}^{M} + \sum_{i.j} f_{rj}^2 S_{ij}^{M} - 2 \sum_{i.j} fri S_{ij}^{M}\, frj$$

$$= 2 f_r^T D^M f_r - 2 f_r^T S^M f_r$$

$$= 2 f_r^T L^M f_r. \tag{5}$$

Correspondingly, we have

$$\sum_{(xi,xj)\in C} (\, fri - fri\,)^2 = \sum_{(i,j)} (\, fri - fri\,)^2\, S_{ij}^{C} = 2 f_r^T L^C f_r. \tag{6}$$

From Eq. (5) and Eq. (6), neglecting the constant (2), Eq. (1) and Eq. (2) will change to:

$$C_r^1 = \frac{f_r^T L^M fr}{f_r^T L^C fr.} \tag{7}$$

$$C_r^1 = f_r^T L^M f_r - \lambda\, f_r^T L^C f_r. \tag{8}$$

The detailed procedure of the spectral graph based Constraint Score algorithm is abstracted in Algorithm (2) as shown below:

Algorithm 2: (spectral graphs)

Input: Datasets X, constraint set C and M, λ (for CS-2 only).

Output: The rated features lists.

Step 1: Constructs the graph G^C and G^M from C and M.

Step 2: guess the weight S^C and S^M using Eq. (3) and Eq. (4), and count the matrices of Laplacian.

Step 3: count the CS of r-th using Eq. (7) (for CS-1) or Eq. (8) (for CS-2).

Step 4: rate the feature depending on their (CS) from small to big.

It is worth mentioning that although both Algorithms 1 and 2 outputs have the same result, introducing the spectral graph theory can bring some extra advantages. First, we can easily extend Algorithm 2 for semi-supervised feature selection, which makes use of unlabeled data together with pairwise constraints for feature selection, by defining appropriate graphs, and corresponding weight matrices in Algorithm (2), second, it provides us with a unified framework from which we can link CS with other feature selection methods, as example: Laplacian Score (LS) [30].

Then they analyze the time complexity of both Algorithms (1) and (2). First, we suppose that the number of pairwise constraints used in (CS) is l, which is limited by $0< 1 < O(m^2)$. Algorithm (1) comprises 2 steps:

- : assesses m feature need O (nl) operations.
- Rank m feature need O (n log n) operations, the overall time complexity of algorithm 1 is O (n max (1, logn)).

Algorithm (2) comprises four steps:

- Steps one and two build the graph matrices use constraints need O (m^2) Operations;

- Step three assesses the m feature based on the graphs, need O (nm^2) operations;

- Step four ranks the features in ascending order in terms of CS, needing O (nlogn) operations. Hence, the overall time complexity of Algorithm 2 is O (n max(m2,log n)). When the number of constraints is very big, i.e. 1=O (m2), algorithm 1 and algorithm 2 have the same time complexity. Anyhow, in experience, often only a few

constraints are efficient, i.e. 1≪O(m2), then we can conclude that Algorithm (1) will be more efficient than Algorithm (2) [29].

2.4.2 Laplacian Score (LS)

The reason of computing LS is to reflect its locality preserving power. Laplacian score depends on the observation that, two data points may relate to the same topic if they are near to each other. Indeed, in many learning problems classification as the example, the global structure of the data space is less important than the local one. In order to model the local geometric structure, we build a nearest neighbor graph. laplacian score search for those features that are related to this graph structure [30].

LS is relying on Laplacian Eigenmaps [31], and Locality Preserving Projection [31]. The main idea of laplacian score is to assess the features according to their locality preserving power [32], it is a popular feature ranking based feature selection method both unsupervised and supervised [33]. LS form the nearest neighbor graph to model the local structure and then choose those features which best to do with this graph structure. The detailed procedure of laplacian score (LS) algorithm is summarized in Algorithm (3) as shown below:

Algorithm 3: Laplacian Score (LS)

Let L_r indicates the LS of the r-th feature. Let f_{ri} indicate the i-th sample of the r-th feature, (i = 1, · · · , m) the algorithm can be stated as follows:

Step 1: Construct the nearest neighbor graph G with m nodes. The i-th node corresponds to x_i. We place an edge between nodes i and j if x_i and x_j are" close", i.e. x_i is among K-N-N of x_j or x_j is among K-N-N of x_i. When the label information is available, one can put an edge between two nodes sharing the same label.

Step 2: If nodes i and j are linked, put Sij = e −(||xi-xj||^2)/t, where t is a suitable constant. Otherwise, put Sij = 0. The weight matrix S of the graph models the local structure of the data space.

Step 3: For the r-th feature, we define:
$f_r = [f_{r1}, f_{r2}, \cdots, f_{rm}]^T$, D = diag(S1), $1 = [1, \cdots, 1]^T$, L = D − S where the matrices L is often called diagram Laplacian [28].

$$f_r^{\sim} = f_r - \frac{f_r^T D1}{1^T D1} 1$$

Step 4: Compute the Laplacian Score of the r-th feature as follows:

$$L^r = \frac{f_r^{\sim T} L f_r^{\sim}}{f_r^{\sim T} D f_r^{\sim}}$$

2.4.3 Fisher score

Fisher score chooses each feature independently according to their scores under the fisher criterion, which leads to a suboptimal subset of features; it is one of the most widely used supervised feature selection techniques, a generalized Fisher score is presented to jointly choose features. It goals to find a subset of features, which expand the lower bound of customary Fisher score. The resulting feature selection issue is a varied integer programming, which can be reformulated as a (QCLP) quadratically constrained linear programming. This issue can be solved by cutting plane algorithm, in each repetition of which a multiple kernel learning problem is solved alternatively by multivariate ridge regression and projected gradient descent [34].

2.5 Constraint Scores for Semi-Supervised Feature Selection

Feature selection scores using pairwise constraints (ML and CL) have shown best performances than the unsupervised technique as comparable to the supervised; the constraint scores use only the pairwise constraints and neglect the available information caused by the unlabeled data. Furthermore, these constraint scores strongly rely on the given must-link constraint and cannot-link constraint subsets made by the user [6].

Another score defined by Zhao *et al*. (2008) which uses both pairwise constraints and unlabeled data to retrieve both discriminating structures and locality properties in the data samples; they build a new graph (GW) which links samples having the high probability of sharing the same label:

- G^W is the within class graph: 2 nodes i and j are linked if the two samples are unlabeled but they are sufficiently close to each other (by using K-N-N), or if (xi,xj) or (xi,xj) belongs to M.

The edges in the graphs (GW) are weighted by using the similarity matrix (SW) (n x n) and are expressed as follows:

$$S_{ij}^{W} = \begin{cases} \gamma & if\ (xi, xj) \in M\ or\ (xi, xj) \in M \\ 1 & if\ xi\ or\ xj\ is\ unlabled\ but\ node\ i \in KNN(j) or\ node\ j \in KNN(i) \\ 0 & otherwise \end{cases}$$

(γ) Constant parameter, which has been empirically set to 100 in Zhao *et al.*

Laplacian score introduced in Zhao *et al*, refer to as the locality sensitive discriminant analysis score as follows:

$$C_r^3 = \frac{\sum_{i,j}(xir - xjr)^2\, S_{ij}^{W}}{\sum_{i,j}(xir - xjr)^2\, S_{ij}^{C}} = \frac{f_r^T\, L^W\, fr}{f_r^T\, L^C\, fr}$$

Where $L^W = D^W - S^W$, (D^W) being the degrees matrix defined by $D_{ii}^{W} = \sum_{j=1}^{n} s_{ij}^{W}$. This score implicitly takes into consideration the unlabeled data but prefers the ML data by giving them high weights in the matrices (S^W). Furthermore, the similarity matrices (S^W) represent the connection between the K-N-N of the data by binary weighting them. By mainly considering the MLC, C^3 is too near to C^2, and the unlabeled data samples are neglected in both of them.

However, the unlabeled data samples should catch the data structure and make less sensitive a feature score against the given constraint subset. That makes us propose another semi-supervised constraint score (CS) which is less sensitive to the constraints chosen by the user. Given the matrices S, S^C and S^M the semi-supervised constraint score (CS) is defined as follows:

$$C_r^4 = \frac{\tilde{f_r}^T\, L\, \tilde{fr}}{\tilde{f_r}^T\, D\, \tilde{fr}} \cdot \frac{f_r^T\, L^M\, fr}{f_r^T\, L^C\, fr}$$

The proposed score C_r^4 is the simple product between the Laplacian Score (LS) L_r processed with the constraint score C_r^1 and unlabeled data defined as follow:

$$C_r^4 = L_r .\, C_r^1$$

These lead us to investigation both the available constraint to C_r^1 and the unlabeled data to L_r in order to evaluate the related of the features f_r.

2.6 Constraint Lplacian Score CLS

We give the characterization of the constraint laplacian score in this section. In fact, constraint laplacian score exploits the two parts of data, unlabeled and labeled. The labeled part is transferred to pairwise constraints, which can be classified in to two subsets: a set of must-link constraints (ML) and a set of cannot-link constraints (CL) [35].

- (ML) Must-link constraint, including two instances xi and xj, it shows that they are belong to the same label.
- (CL) Cannot-link constraint, including two instances xi and xj, it show that they are belong to the more than one label.

All constraints (must-link and cannot-link) are generated from each pair of examples in TL as we defined above, so their amount is equal to L(L−1)/2.

Fr is a features to correct, realize its vectors by fr = (fr1, fr2, frn). The constraint LS of Fr, which must be minimized, is counted by this formula:

$$\text{Constraint laplacian score (CLSr)} = \frac{\sum_{i,j}(fri-frj)^2\, S_{ij}}{\sum_i \sum_{j|\exists l,(xl,xj)\in \Omega CL}(fri-\propto_{rj}^{i})^2\, Dij}$$

Where *(D)* is a diagonal matrices with $D_{ii}=\sum_j S_{ij}$,$D_{ii}=\sum_j S_{ij}$, and S_{ij} is relaize by the nearest relations among examples (X_i=1,2,3,..,n) is $S_{ij}=e^{-\frac{||xi-xj||^2}{\lambda}}$ if $\big((xj, xi)\epsilon$ XU and xj, xi are neighbors$\big)$ or $(xi, xj)\epsilon$ ML otherwise S_{ij} = 0.

Where (xi,xj) are nearest mean the xi is between *k* n n of xj, and (λ) is a constant to be set:

$$\alpha_{rj}^{i}\begin{cases} frj & \text{if } (xi, xj) \in \Omega CL \\ \mu r & \text{if } i = j \text{ and } xi \in X_U \\ fri & \text{otherwise} \end{cases}$$

Where μr=$\frac{1}{n}\sum_i fri$ (the mean of the feature vector fr).

We can see the term a_{rj}^{i} calculates for each example *i* at the *r*th feature regarding its distance from the mean when xi belongs to unlabeled side, or the sum of distance

between the current example and the other *j* one's, which it is linked through a cannot-link constraints, the equations consist of three phases:

Phase1 fri: for deleting the term fri−a_rj^i when not in the aforementioned phases.

Phase2 frj: if xi related to labeled side.

Phase3 μr: if xi belongs to an unlabeled side.

Constraint laplacian score represents an enhanced version of the two scores: constraint-based [2] and Laplacian [30]. The laplacian score (LS) can be viewed as a special version of constraint laplacian score when the number of labels are zero (X=XU), and when there is the label (X=XL), constraint laplacian score can be regarded as an adjusted version of the constraint score (CS) [2]. In constraint laplacian score, we proposed a more efficient combination of both scores by a new score function, including the constraint-preserving ability of labeled data and the geometrical structure of unlabeled data.

On the one side, with constraint laplacian score [35] a relevant feature should be the one with a larger variance or in which these two instances (related by a CL constraint) are correctly separated, On the other side, the relevant feature should be the one in which these 2 instances (related by a must-link constraint or neighbors) are near to each other. We summarized the entire constraint laplacian score procedure in Algorithm 4.

Algorithm 4: CLS algorithm

One datasets X ($n \times m$), which have 2 subsets: X_U ($U \times m$), the subset of unlabeled and X_L ($L \times m$), the subset of labeled.

Inputs:

($F = \{F_1, F_2..., Fm\}$); (λ) and the neighborhood (k).

Step 1: build the constraints (*ML* and *CL*) from the labeled part.

Step 2: count the diagonal matrices *D and* the dissimilarity matrix *S*.

Step 3: for $i = 1$ to m do.

Step 4: Compute (*CLSr)* according to equation (1).

Step 5: end for.

We can say that the above is computed in time O (m×max (n^2, log m)). To decrease this complexity, it is proposed in constraint laplacian score (CLS) [35] to the clustering on unlabeled training instance (X_U). Here the idea was to replace this huge side of data by a smaller side $X'_U=(u_1, u_2, u_3,...,u_C)$ by keeping the geometric structure of labeled training instance (X_U), where *C* is represents the number of clusters. We suggested using (SOM) the self-organizing map-based clustering [36], for its ability to keep the topological relations of data properly and thus the geometric structure of their distribution. In this strategy, we decreased the complication to O (m × max (U,log m)), *U* is the size of labeled training instance X_U.

However, to construct the map, we can have used (PCA) a principal component analysis based heuristic proposed by [36] for automatically supplying the dimensions of the map and the initial number of clusters (C). The reference vectors are initialized linearly along the greatest eigenvectors of each one of them associated dataset.

2.7 Ensemble Constraint Laplacian Score (ENCLS)

Combine models are the most significant condition for a successful ensemble learning method, which differs from one to another. Hence, to keep diversity between committee members, we used two techniques. First, the diversity is further observed by applying the bootstrapping method second, a well-known ensemble method named random subspace method (RSM) [37] is employed to face the curse of dimensionality problem by constructing multiple classifiers; each trained on a different subset of examples projected on a smaller feature set RSMi. [38].

This method is described in detail in algorithm (5), X_U is a set of unlabeled training examples, X_L is a set of labeled training example, which described over the input space $F=\{F_1, F_2,\ldots, F_m\}$, our method makes a committee according to this phases, as described in Algorithm (5) in step four and step five. The committee is made as following: for each one of the ensemble components a, *i* replication from the labeled part X_L, b^i of the labeled dataset is attained by choosing examples from labeled part (X_L) with replacing and then projecting them over random subspace

method (RSM^i), a feature subspace by *M* randomly selected features (M<m). The unlabeled data port of data (X_U) is also projected over random subspace method (RSM^i) [10]. To produce labeled part of data XUi. Once each one of the ensemble component *i* is obtained, the (CLS) in Algorithm 4 is used to evaluate feature relevance as we see in the algorithm above (step 7). Finally, the ranking of all features is processed regarding their average relevance's as we see in (steps 13 - 15).

Algorithm 5: The Ensemble Constraint Laplacian Score (EnsCLS) algorithm

Require:

(X_L) is a Fixed of labeled training instances; (X_U) is a fixed of unlabeled training instances;

Input:

($F = \{F_1, F_2, \ldots, F_m\}$); with size ($N$)

step 1: Initialize the scores I(*fr*) to zero for each feature *Fr*

step 2: Reset the incidences O(*fr*) to zero for each features *Fr*

step 3: for $j = 1: N$ do

step 4: Random Subspace Method (RSM^i)= arbitrarily magnet *M* features from *F*

step 5: bootstrap sample from X_L projected onto $RSM^i = X_L, b^i$

step 6: the unlabeled sample X_U projected onto $RSM^i = X_U{}^i$

step 7: $imp^i = CLS\ (X_L, b^i, X_U{}^i)$ compute the (CLS) of each feature in

Random Subspace Method (RSM^i) using Algorithm 4

step 8: for each feature $Fr \in RSM^i$ do

step 9: I(*fr*) = I(*fr*) + imp^i(*fr*)

step 10: O(*fr*) = O(*fr*) + 1

step 11: end for

step 12: end for

step 13: for every one of the features $Fr \in F$ do

step 14: I(fr) = $\frac{I(fr)}{O(fr)}$

step 15: end for

step 16: rate the feature in (F) depending on their scores (I) from smaller to bigger

step 17: return F

Ensemble learning paradigms train multiple component learners and then combine their output results. While a single learner is known to produce very poor results as the learning algorithms break down with high-dimensional data. Ensemble approach is regarded to improve the robustness and the generalization ability of single learners, and the effective solution to overcome the dimensionality problem [3].

2.7.1 Problems in the Constraint Laplacian Score (CLS)

Here we explain problems of (CLS)

a) High dimensionality

The main shortcoming of constraint laplacian score was the application on high-dimensional data. Due to the fact of the euclidean distances between samples are an essential factor in the function score, it makes the calculation of such distances more biased when treating with very high-dimensional data resulting in bad feature scores, it adopts the use of the random manipulation strategy over the feature space Random Subspace Method (RSM) [10]. Thus, we create (N) random subspaces of the original features with a nearly equal occurrence probability for all features. The high dimension is then reduced in each subspace, and the distances calculated depending on the new reduced dimension are less biased [39].

b) Constraints

Instance-level constraints are produced from labels when we apply the Constraint Laplacian Score. Partially labeled in the semi-supervised, so the number of constraints (Ω=L (L−1)/2) and L represent the number of the labeled instances. Furthermore, the constraint set that we generated may consist of some noisy constraints, which proven to have adverse effects on performance of the learning. We suggest the use of the bagging method on the labeled part of data in each one of the random subspace. To improve the positive effects of the pairwise constraints, bootstrap aggregation (Bagging) made by sampling with replacement [1]. Therefore, we use bagging for some reason, one of them is to enhance diversity on pairwise constraints and then to compute a robust estimation of the feature score (FS) compared with minor changes in the pairwise constraint set. Moreover, when we have different bootstraps samples in different random subspaces, it helps to reduce the undesirable effects of the noisy constraints(C) [2].

c) Unlabeled instance diversity

Calculation of the constraint laplacian score suggested the technique of a clustering algorithm (SOM) [40]. To overcome the calculation complexity of the score function. Because of the fact that the complexity of the constraint laplacian score highly depends on the unlabeled side of data, this clustering helps to reduce this complexity [3].

2.8 Bootstrap Aggregation (Bagging)

Briman (1994) defined bootstrap aggregation (Bagging) predictors as a technique for generating multiple samples of a predictor and then uses these samples to initialize an (aggregated predictor). The averages of the aggregation over the samples when predicting a numerical outcome and does a plurality vote when predicting a class. We can get multiple samples by making bootstrap replicates of the learning set and then we can use these samples as new learning sets. Classification and regression are two methods used to test on real and simulated data sets using trees and subset selection in linear regression show to us that bootstrap aggregation (bagging) may give significant gains in accuracy. Instability of the prediction method is the

important element. If perturbing the learning set can result in significant changes in the predictor constructed, then bagging can get better accuracy [41].

2.8.1 Bagging, boosting and stacking in machine learning

This three technique are also called "meta-algorithms": approaches to doing the same work to combine several machine learning techniques into one predictive model in order to improve the predictive force (stacking alias ensemble), and bias (boosting) or decrease the variance (bagging),

Each one of these algorithms includes two steps:

Step 1: Combining the distribution into one "aggregated" model.

Step 2: Producing a distribution of simple Must-Link models on subsets of the original data.

2.8.2 Short description of all three methods (Bagging, Boosting and Stacking):

a) Bagging

Bootstrap aggregation (bagging) is a better way we can use it to decrease the (variance) of your prediction, that happened when we generate an additional data for training from the original dataset using combinations with repetitions to produce multi sets of the same size as the original data. In the case if we increasing the size of the training set we cannot improve the model predictive force, only we can decrease the (variance) [41].

b) Boosting

This method contains two steps, the first step is to uses subsets of the original data to produce a number of averagely performing models and after that, we combining them together using a specific cost function called majority vote to (boosts) their performance. It different from (bagging) that the subset creation is not random; it relies upon the performance of the previous models [42].

c) Stacking

This method is like boosting; in point, they also generate models from the original data, it only different from boosting in point that, that you do not have just an empirical formula for your weight function; rather you introduce a meta-level and use different approach, to determine what models perform well and what badly give these input data [43].

2.9 Bagging Constraint Score for FS

As it mentioned in previous topic constraint score (CS), it is a recently proposed method for feature selection by using pairwise constraints and identify if a pair of instance I in the same class or it belong to the different class. As it explicated the constraint score, that contains a little number of pairwise constraints, achieves comparable performance than those with fully supervised feature selection (FS) methods e.g. (fisher score). Anyhow, there is also the disadvantage of the constraint score one of them is that performance relies on the best selection on the composition and cardinality of constraint set, it can be a big challenging in practice. We deal with this problem by importing bootstrap aggregation (Bagging) into constraint score (CS) and it proposed a method called (BCS) bagging constraint score. In this method bagging constraint score (BCS); we perform multiple constraint scores, each of which uses a bootstrapped subset of originally given constraint set. Diversity analysis on individuals of ensemble leads us to know that resampling pairwise constraints are good for simultaneously increasing the diversity of individuals and the accuracy [1].

CHAPTER THREE

MATERIAL AND METHODS

3.1. Introduction

In this chapter, we are going to discuss a technique called (BCLS), and we will deal with semi-supervised feature selection data. The semi-supervised data has two parts labeled part and unlabeled part.

3.2 Methodology

In this work, we have four main steps, the first step is to resampling the original dataset into multiple sets and this process called bootstrap aggregating (bagging), the second step is to rank the features of each dataset that generated by (bagging), and the process used to rank the feature is called constraint laplacian score (CLS), the third step is to classify the data of all dataset that we generated before, and the fourth step is make voting between a classified datasets to find the best of them that have a high accuracy we called this process majority voting, so finally we will get the only dataset that has a higher accuracy among all of the datasets. We will explain all of these steps in details.

3.2.1 Bootstrap aggregating (Bagging)

Bootstrap aggregating is the first step of this work, Bootstrap aggregating is one of the machine learning algorithms. The main advantages of using this algorithm are to improve the accuracy; the stability; avoiding the overfitting; and reduce the variance. The descriptions of this technique include; the original data set of size n; bagging randomly generates m numbers of datasets by sampling from the original one; and random selection with replacement. Leo breiman proposed this technique in 1994 to

Improve the classification by combining classifications of randomly generated training sets, in the figure (3.1) below it declares more about this technique.

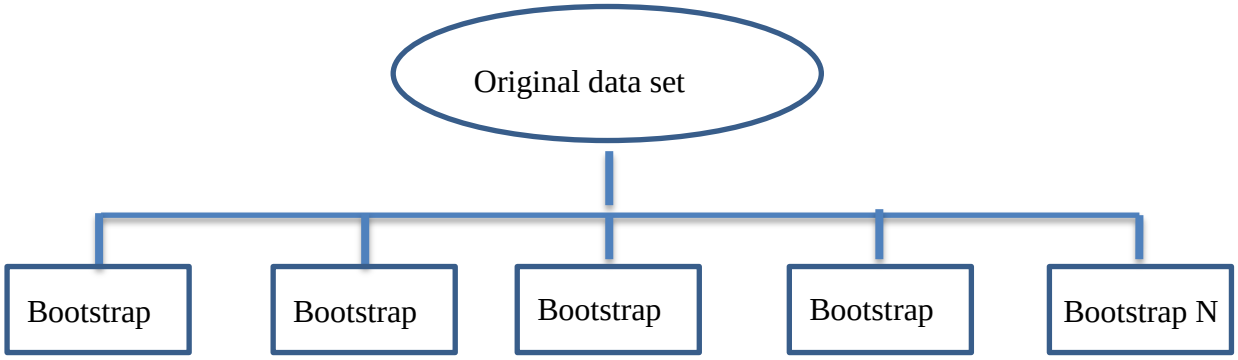


Figure 3.1 The idea of how bootstrap aggregating (Bagging) technique works.

3.2.2 Constraint laplacian Score

The second step in this work, is to rank the features of generated datasets by technique called constraint laplacian score (CLS); the idea of this technique is the constraining of the laplacian score by the information extracted from labeled part of data. The goal of CLS is to rank the ability of features in preserving the local geometric structure offered by unlabeled part of data with respecting the constraint of labeled part of data. An example of ranking the features is shown in figure (3.2).

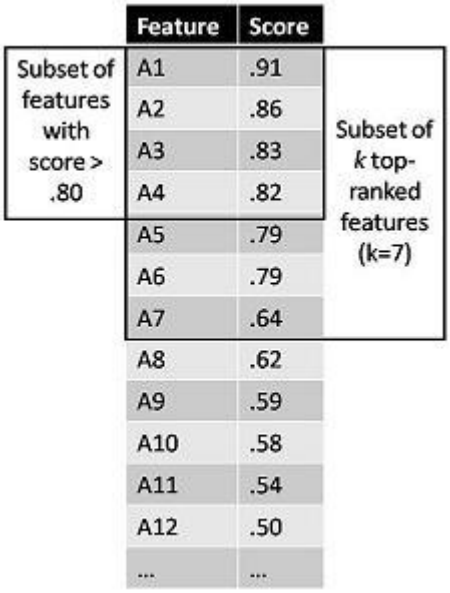


Figure 3.2 An example of ranking the features.

3.2.3 Classification

In this work classification is the third step, after we ranked the feature for each generated dataset; we start to classify each generated dataset individually, the idea of classification is to classify an instance of datasets into predefined category. A classifier will compute the probability for each instance, and this probability makes the instance belong to one of the labels. There are many types of classification but in this work we used the K-NEAREST NEIGHBOR (K-N-N) which, is considered as a simple lazy learner algorithm, it classify the dataset depend on their similarity with the neighbors, K stands for number of dataset items that are considered for the classification. Figure (3.3); is an example of how k-n-n works.

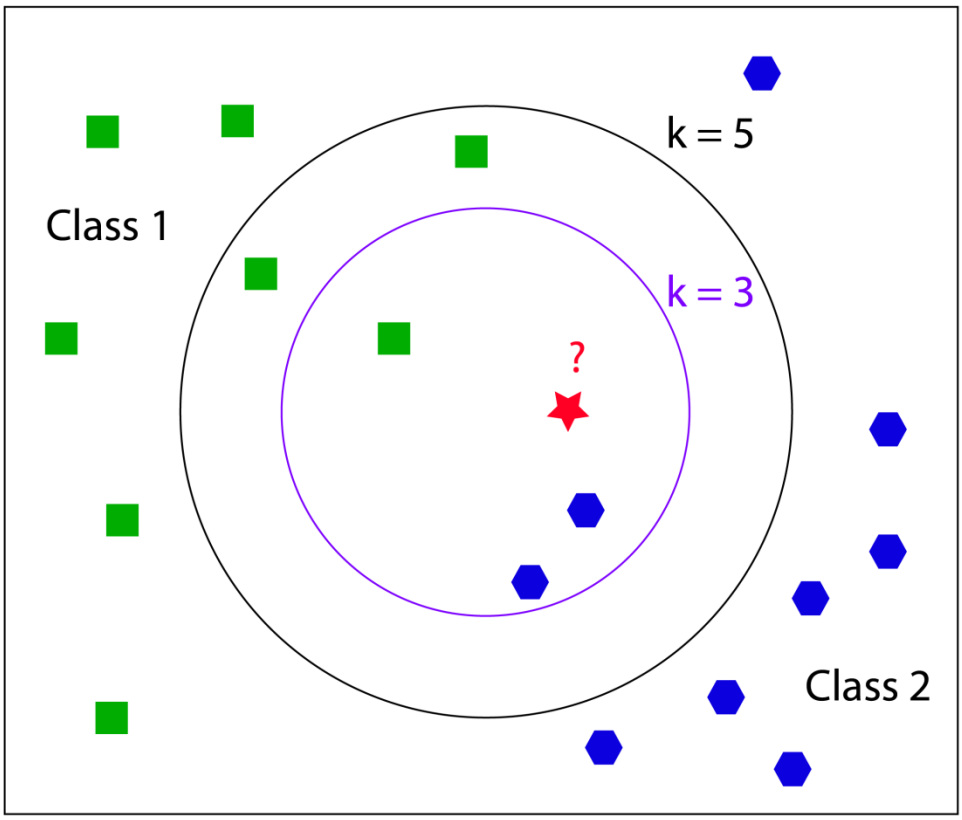


Figure 3.3 Explain the idea of K-N-N and in this example, we have two classes' class1 and class2 and considering K is equal 3 and equal to 5 in another one.

The second type of classification that we used in this work is the naïve base, Bayesian classifiers are sort as a statistical classifier. It can expect class membership probabilities, as for example the probability that a given sample belongs to a certain class. Bayesian classifier is based on Bayes' theorem. It assumes that the effect of an

attribute value on a given class is independent of the values of the other attributes. This is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered “naive”.

In computer science literature and statistics, naive Bayes method is known under more than one name, including independence Bayes and simple Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method.

3.2.4 Majority Voting

Majority voting is a type of ensemble learning, it is a decision rule that selects alternatives which have a majority, that is, more than half the votes, when can also say that it combine the classifier’s prediction, as for example, we can say we combine four classifier that classifies a training sample as follows:

- Classifier 1 → class 0
- Classifier 2 → class 1
- Classifier 3 → class 0
- Classifier 4 → class 0

So here via majority voting, we would classify the sample as “class 0”.

The idea of majority voting is shown in figure (3.4).

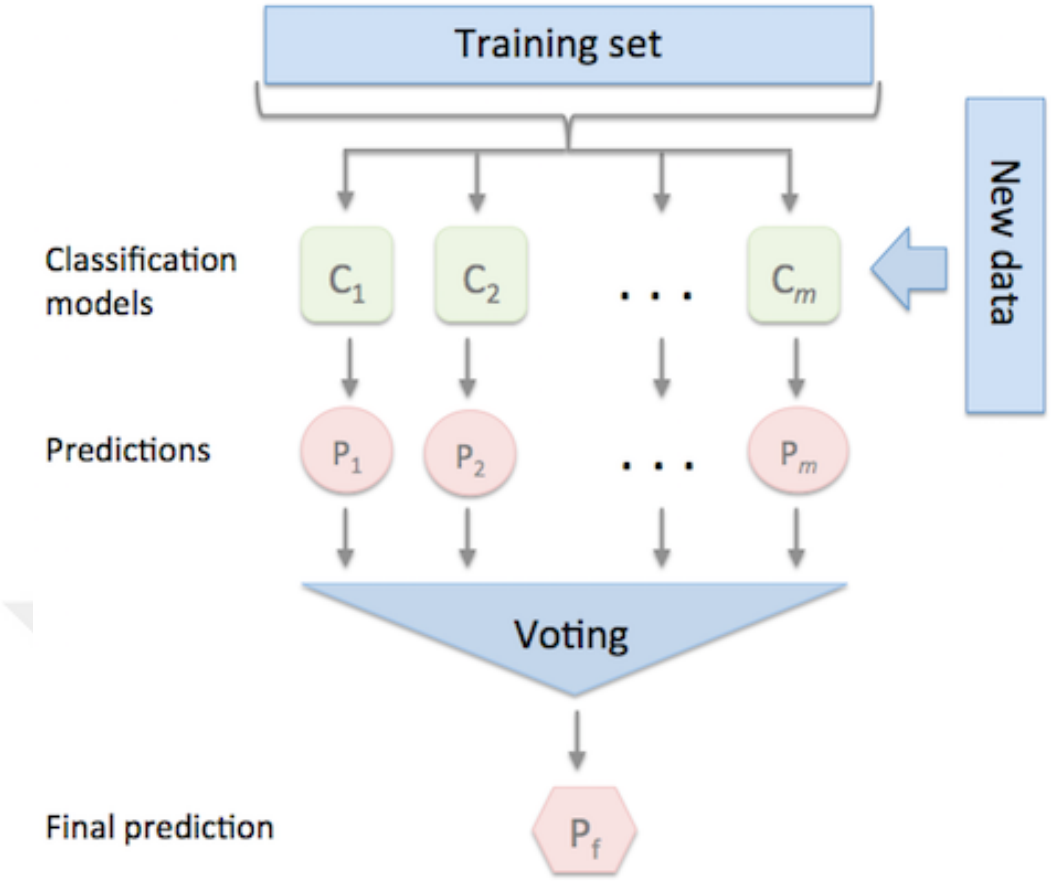


Figure 3.4 The idea of majority voting.

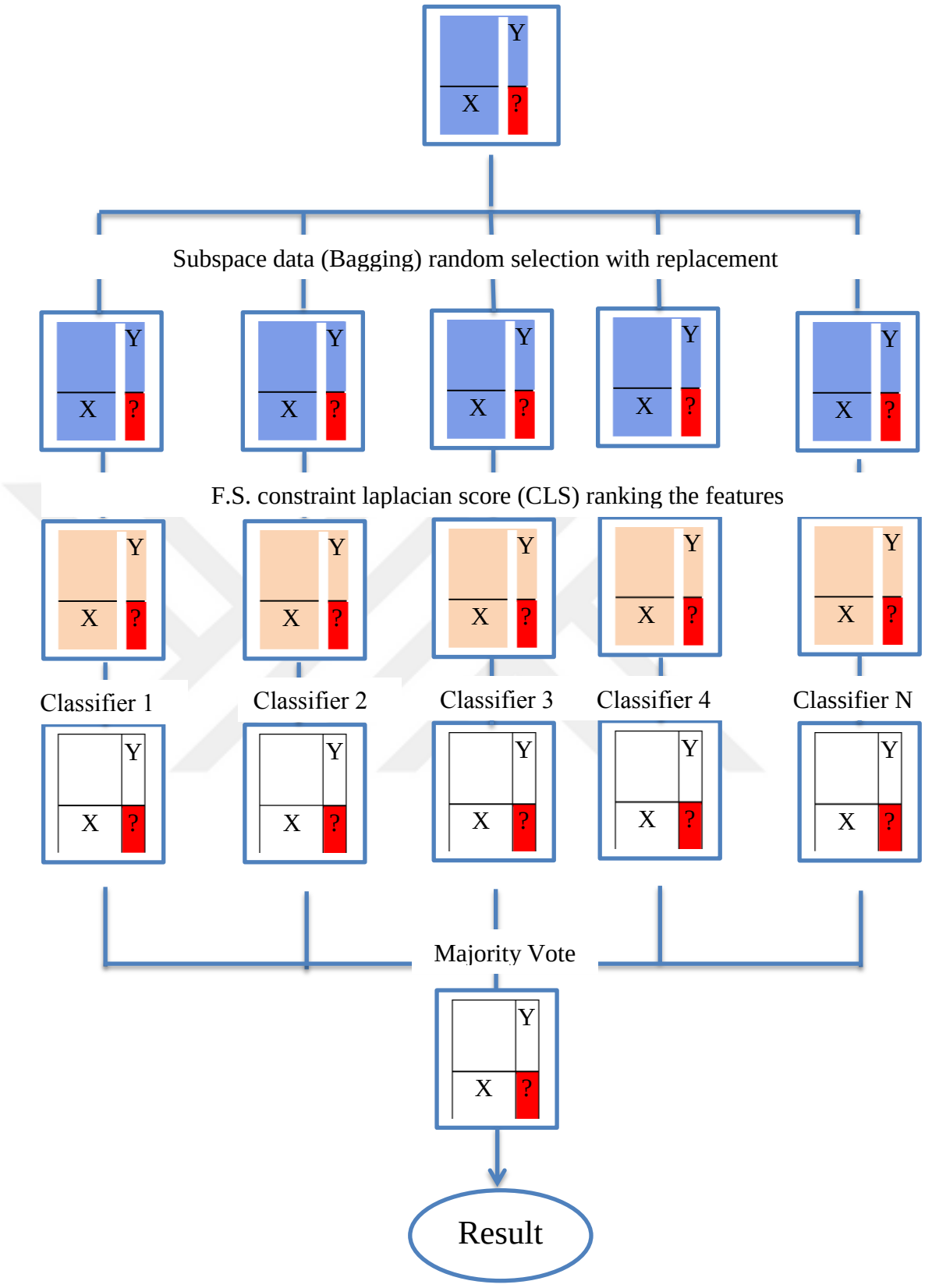


Figure 3.5 All the steps of this work (bagging constraint laplacian score for efficient semi-supervised feature selection).

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Introduction

In this chapter, we will discuss the main idea of 'bagging' constraint laplacian score, the experiments, datasets parameters and the result of the experiments. Comparing the new method with the previous methods 'Fisher-Score' and 'Multi-Cluster Feature Selection' (MCFS) also will be compared.

4.2 Datasets used

The datasets used in this work the **WINE, AR10P, PIE10P, PX10P, ORL10P, YALE, COIL20P and SONAR,** which taken from open source feature selection repository[1] with **WINE** and **SONAR** from machine learning repository[2].

4.2.1 The WINE dataset

The wine dataset includes the results of a chemical examination of wines developed in a particular region of Italy. Three kinds of wine is denoted in the (178) samples, with the results of (13) features chemical examination recorded for each sample.

4.2.2 The AR10P dataset

It is face dataset represented in the 130 samples and 2400 numbers of features.

4.2.3 The PIE10P dataset

They collected a database of (41,368) images of 68 people between (October – December) 2000. Via protraction the CMU 3D Room they were capable of image the

[1] http://featureselection.asu.edu/datasets.php
[2] https://archive.ics.uci.edu/ml/datasets

An entire person under 13 various poses, 43 various illumination conditions, and with four different expressions. We called this Expression (PIE) dataset.

4.2.4 The PIX10P dataset

The PIX10P dataset is face image dataset represented in the 100 samples and 10000 numbers of features.

4.2.5 The ORL10P dataset

This dataset includes a group of face images and these face images taken between (April 1992) (April 1994) at the lab. It was used in the context of a face recognition project carried.

4.2.6 The YALE dataset

This dataset includes 165 gray scale images in (GIF) format of 15 characters, there are 11 images per object, one for each various facial expression: left-light, w/glasses, sad, center-light, w/no glasses, sleepy, right-light, happy, normal, surprised, and a wink.

4.2.7 The COIL20 dataset

This dataset includes 20 patterns. The images of each pattern taken five degrees apart as the pattern are rotated on a turntable and each one of these patterns has 72 images, and the size of each one of these images is (32x 32) pixels, with 256 gray grades per pixel, and a 1024-dimensional vector denotes each image.

4.2.8 The SONAR dataset

This dataset includes 111 objects gained by bouncing sonar signals off a metal cylinder at different angles and under the different status. Figure (4.1) described the characteristics of these datasets.

Table 4.1 characteristics of the datasets used in the experiments

Dataset	#Instances	#Features	# of classes	Type	% of label
WINE	178	13	3	Multivariate	30
AR10P	130	2400	10	Face	30
PIE10P	210	2420	10	Face	30
PIX10P	100	10000	10	Face	30
ORL10P	100	10304	10	Face	30
YALE	165	1024	15	Face	30
COIL20	1440	1024	20	Face	30
SONAR	208	60	2	Multivariate	30

4.3 Experiments

In the first part of experiments, we will show and explain the result of our work using the K-N-N method for classification step.

In the second part of experiments, we will show and explain the result of our work using the naïve base method for classification step.

In Figure 4.1, we see the difference between the data blocks after sampling the original dataset and applying the laplacian logarithm.

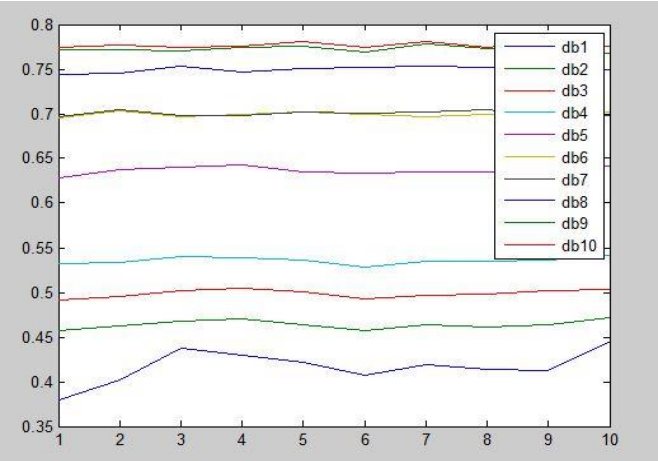


Figure 4.1 The data blocks after using the laplacian score on the YALE dataset

As shown in figure (4.2), the accuracy of BCLS is better than the other two methods Fisher score and MCFS on the face image datasets YALE and PIE10P.

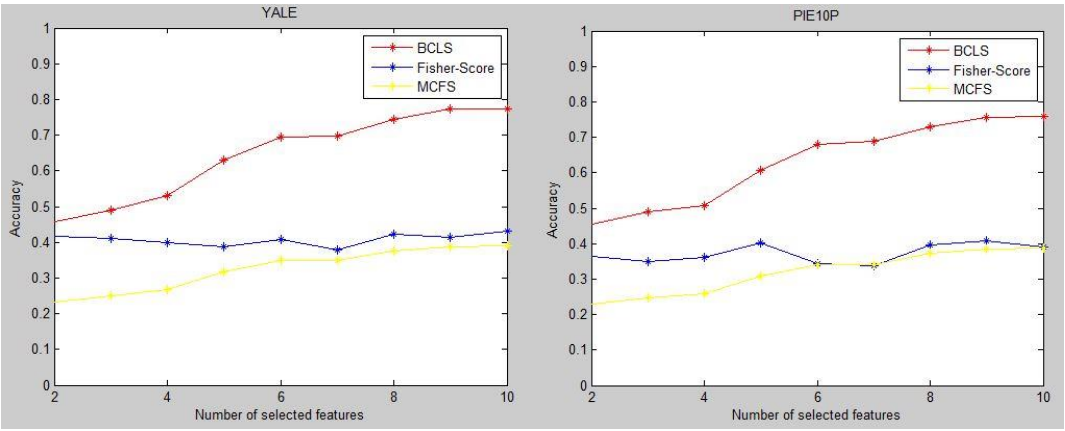


Figure 4.2 The accuracy vs. a number of selected features after the bagging constraint laplacian score, Fisher score and MCSF methods on YALE and PIE10P datasets.

In Figure 4.3, the accuracy of BCLS is better than Fisher score and MCFS and the accuracy of Fisher score are at the same level during all iterations but in the MCFS the accuracy becomes better in last than the first iterations. In this Figure we use the PIX10P dataset.

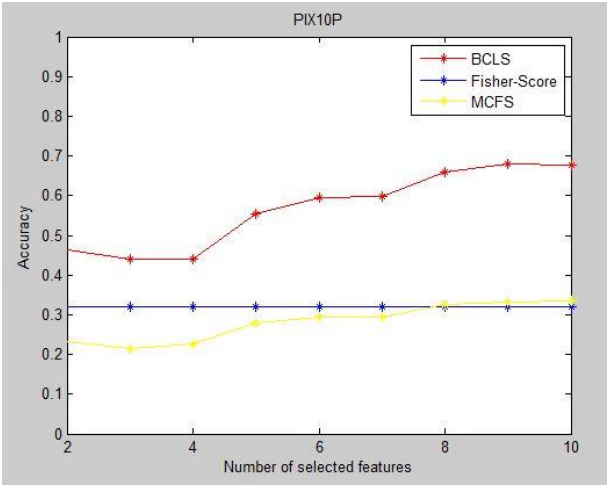


Figure 4.3 The accuracy vs. a number of selected features after the bagging constraint laplacian score, Fisher score and MCSF methods on PIX10P datasets.

In the Wine dataset, the accuracy of BCLS is better than the other two methods during all iterations as we see in the below Figure (4.4).

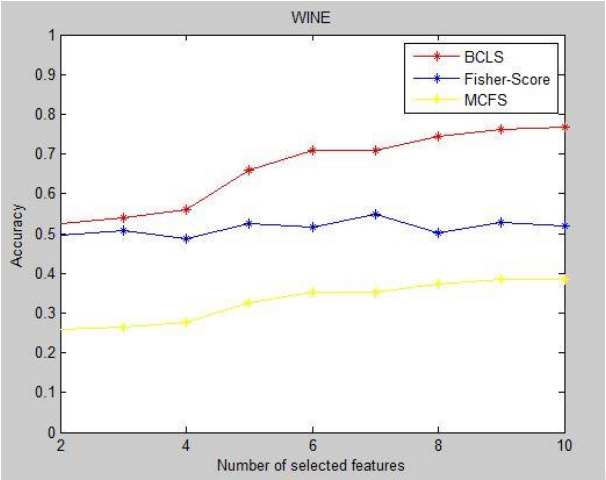


Figure 4.4 The accuracy vs. a number of selected features after the bagging constraint laplacain score, Fisher score and MCSF methods on WINE datasets.

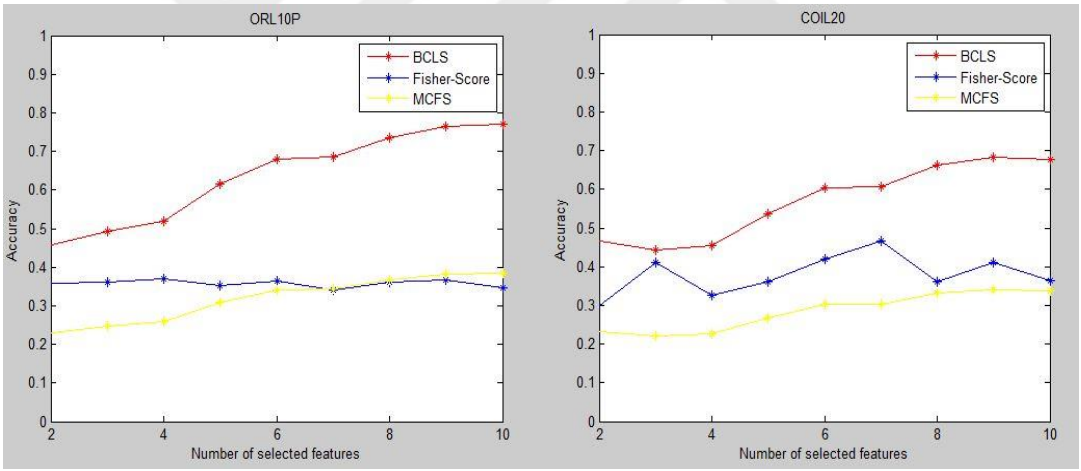


Figure 4.5 the Accuracy vs. number of selected features after the bagging constraint laplacain score, Fisher score and MCSF methods on COIL20 and ORL10P datasets.

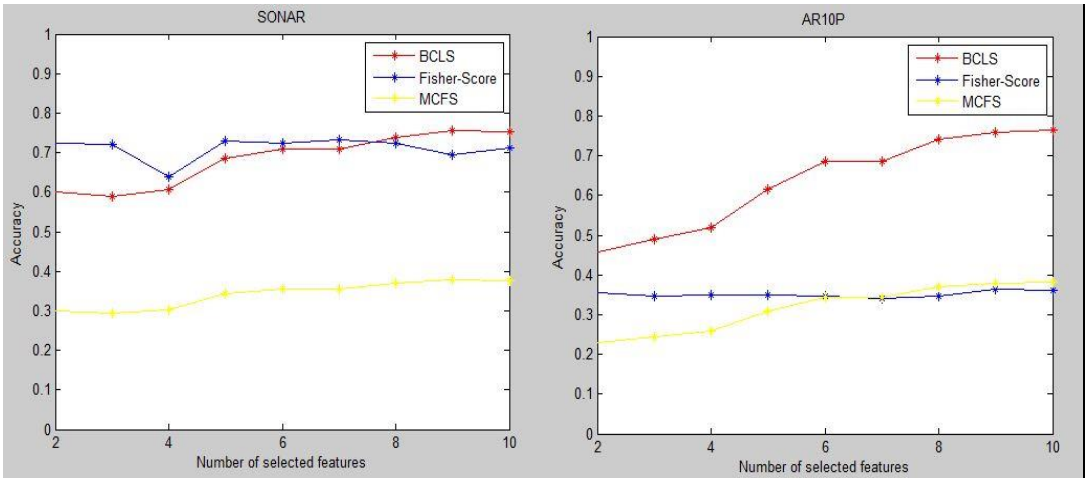


Figure 4.6 The accuracy vs. a number of selected features after the bagging constraint laplacian score, Fisher score and MCSF methods on SONAR and AR10P datasets.

In the below results we change the classification algorithm from the K-N-N to the naïve base algorithm, as we will see the difference of using these two classification algorithms on the accuracy of the data.

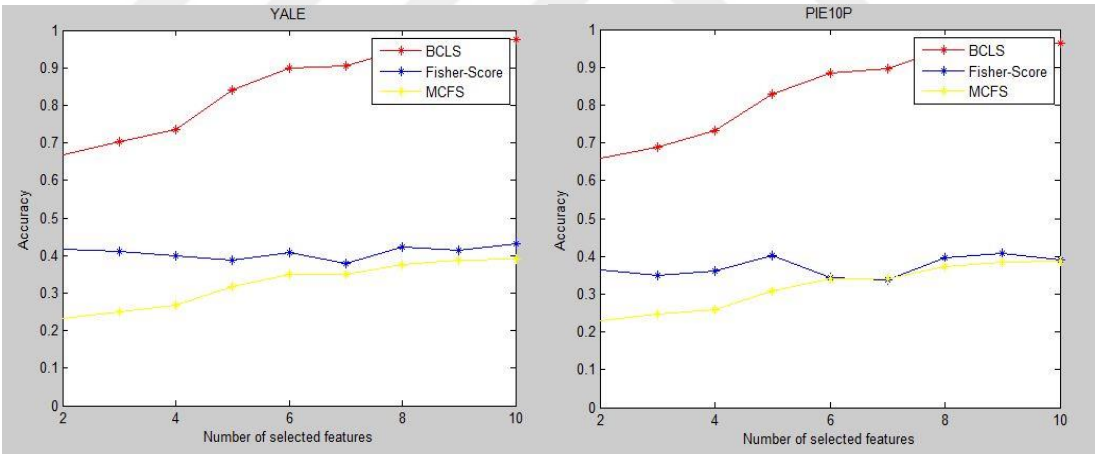


Figure 4.7 The accuracy vs. a number of selected features after the bagging constraint laplacian score, using the naïve base algorithm for classification, Fisher score and MCSF methods on YALE and PE10P datasets.

Figure (4.8) shows that, the accuracy of BCLS is better than the fisher score and the multi-cluster feature selection (MCFS). The accuracy of BCLS is increase when the number of data increases.

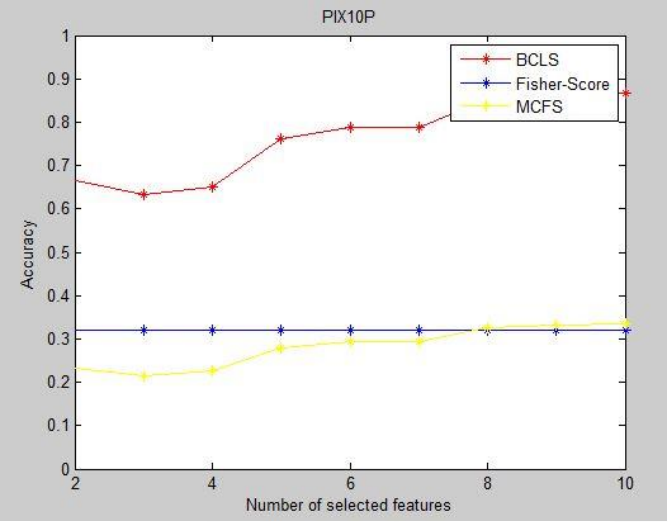


Figure 4.8 The accuracy vs. a number of selected features after the bagging constraint laplacian score, using the naïve base algorithm for classification Fisher score and MCSF methods on PIX10P datasets.

In the figure (4.9) the accuracy of BCLS go down into negative side in the first two iterations and after those two iterations the accuracy goes up and increases when the number of selected feature increases.

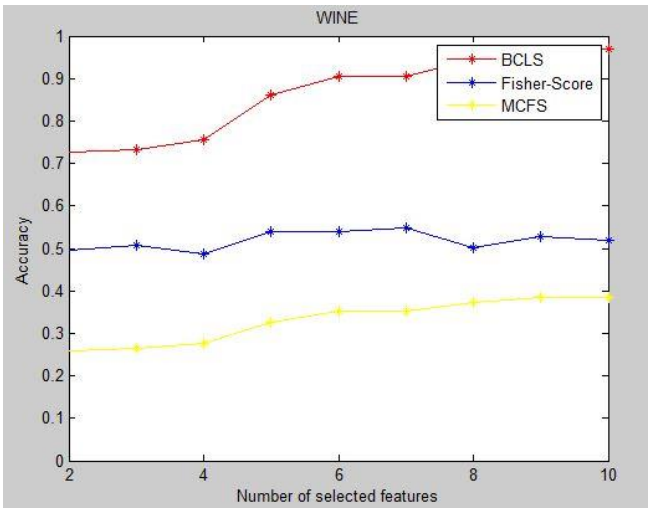


Figure 4.9 The accuracy vs. a number of selected features after the bagging constraint laplacian score, using the naïve base algorithm for the classification, Fisher score and MCSF methods on WINE datasets.

Applying the BCLS on the wine datasets increases the accuracy during all iterations and is better than the other two algorithms as we see in figure (4.9).

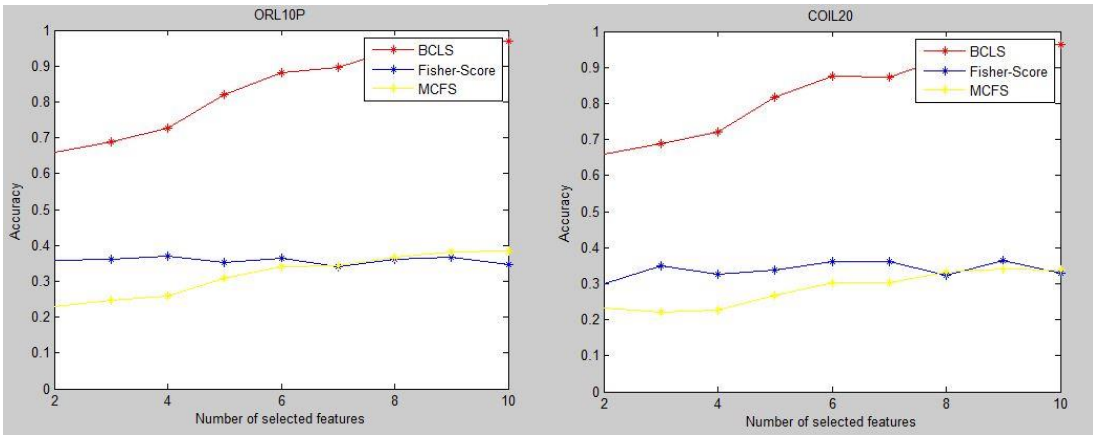


Figure 4.10 The accuracy vs. number of selected features after the bagging constraint laplacian score using the naïve base algorithm for the classification, Fisher score and MCSF methods on COIL20 and ORL10P datasets.

In the figure (4.10) above the accuracy of both datasets ORL10P and COIL20 is almost same that is because the features value of both datasets is too close to each other, and as we see the both have good accuracy when apply the BCLS on them, but the ORL10 is little better than the COIL20.

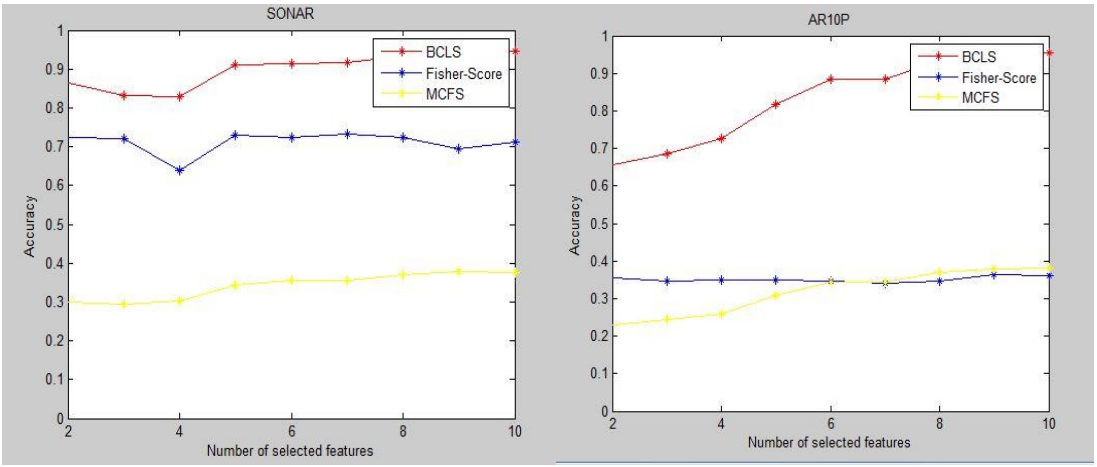


Figure 4.11 the Accuracy vs. number of selected features after the bagging constraint laplacian score using the naïve base algorithm for the classification, Fisher score and MCSF methods on SONAR and AR10P datasets.

As we see in all results above applying the naïve base for classification generates better results than applying the K-N-N algorithms.

The advantages of using the naïve base are that it is too simple; we are only doing a group of counts. If the NB conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models, so you need less training data. Moreover, even if the NB assumption does not hold, an NB classifier still often performs surprisingly well in practice. A good bet if you want to do some kind of semi-supervised learning, or want something embarrassingly simple that performs pretty well.

Even the results of BCLS using the K-N-N algorithms have a good accuracy as we see in the mentioned results. However, if we compare between both of them the naïve base is better than K-N-N in general.

CHAPTER FIVE

CONCLUSION & RECOMMENDATION

5.1 Conclusion

In this study, we suggested an algorithm called constraint laplacian score (CLS) to solve the semi-supervised feature selection problem, the main drawback of this algorithm is the choice of the little supervision data, represented by pairwise constraints, constraints are proven to have some noise, which may deteriorate the learning performance.

In this work, we override the negative effects of constraints set by the variation of their source; all this done by an ensemble technique using bootstrap aggregating (bagging) and random subspace strategy then deleting any redundancy in selected feature.

The experiments result shows for us that the ensemble feature selection methods are not always superior to those with randomly selected constraints.

5.2 Recommendation

In the future work, we can use different algorithms to rank the feature with the idea of BAGGING, that mean after sampling the original dataset into data blocks. We can suggest a different method for each data block as example of method for ranking the features (fisher score, constraint score and laplacian score) as example, after we sample the dataset into N numbers of data blocks we can use a fisher score method for the first (five) data blocks and for the second (five) data block we can use the constraint score method, this can be a good idea to get good results.

Another idea can be work on it in the future for the classification step we can apply another method instead of the K-nearest neighbor for classifying the data there are some methods we can use it as example of this method we can use the support vector machine (SVM), this step of classifying the data become after

Bootstrapping the original data into data block as we did in this work and ranking the same data using on of feature selection techniques.

REFERENCES

[1] Sun, D., Zhang, D., (2010). Bagging constraint score for feature selection with pairwise constraints, *Pattern Recognition*, **43(6),** 2106-2118.

[2] Zhang, D., Chen, S., Zhou, Z. H., (2008). Constraint Score, A new filter method for feature selection with pairwise constraints, *Pattern Recognition*, **41(5)**, 1440-1451.

[3] Benabdeslem, K., Elghazel, H., Hindawi, M., (2016). Ensemble constrained Laplacian score for efficient and robust semi-supervised feature selection, *Knowledge and Information Systems*, **49(3)**, 1161-1185.

[4] Yu, L., Liu, H., (2003). Feature selection for high-dimensional data, a fast correlation-based filter solution, In *ICML*, **3**, 856-863.

[5] Kohavi, R., John G., (1997). Wrappers for feature subset selection, Artificial Intelligence, **97(1)**, 273-324.

[6] Kalakech, M., Biela P., Macaire L., Hamad D., (2011). Constraint scores for semi-supervised feature selection: A comparative study, *Pattern Recognition Letters*, **32(5)**, 656-65.

[7] Davidson, I., Liu L., (2009), editors. *Encyclopedia of Database Systems*. Boston, MA: Springer US.

[8] Clark, P., Niblett, T., (1989). *The CN2 Induction Algorithm Machine Learning*, **3(4)**:261-83.

[9] Arauzo-Azofra, A., Benitez, J., Castro, J., (2008). Consistency measures for feature selection, *Journal of Intelligent Information Systems*, **30(3)**, 273-92.

[10] Panov, P., Džeroski, S., Clare, C., (2007). Combining Bagging and Random Subspaces to Create Better Ensembles. In, editors. Advances in Intelligent Data Analysis VII: *7th International Symposium on Intelligent Data Analysis*, IDA, Ljubljana, Slovenia, 6-8, Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg.

[11] Breiman, L., (2001). Random Forests, *Machine Learning*, **45(1)**, 5-32.

[12] Vincent, N., Claire, C., (2003). Bootstrapping conference classifiers with multiple machine learning algorithms, *Proceedings of the 2003 conference on Empirical methods in natural language processing*, Association for Computational Linguistics.

[13] Skurichina, M., Kuncheva L., Duin, R., (2002). Bagging and Boosting for the Nearest Mean Classifier, Effects of Sample Size on Diversity and Accuracy, editors. Multiple Classifier Systems: *Third International Workshop*, MCS Cagliari, Italy, June 24–26, Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg.

[14] Welling, M., Max, G., (2005). Fisher linear discriminant analysis, Department of Computer Science, University of Toronto, **3**, 1-4.

[15] Guyon, F., Isabelle, D., André, E., (2003). An introduction to variable and feature selection, *Journal of machine learning research*, **3**,1157-1182.

[16] David, A., Dickey, C., (2012), Introduction to Predictive Modeling with Examples, *SAS Global Forum Statistics and Data Analys*is.

[17] Blum, A., Langley, P., (1997). Selection of relevant features and examples in machine learning, *Artificial Intelligence*, **97(1)**, 245-71.

[18] Janecek, M., Andreas, A., (2008). On the Relationship Between Feature Selection and Classification Accuracy, *FSDM*.

[19] Ghodsi, B., Ali, G., (2006). Dimensionality reduction a short tutorial, *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada* 37: 38.

[20] Kumar, K., Batt, J., (2016). Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms, *International Journal of Computer Applications*, **150**(12).

[21] Liu, H., Hiroshi, M., (2012). *Feature selection for knowledge discovery and data mining*, Springer Science & Business Media, **454**.

[22] Duch, W., (2006). Filter methods, *Feature Extraction*. Springer Berlin Heidelberg.

[23] Kohavi, R., Dan, S., (1995). Feature Subset Selection Using the Wrapper Method for machine learning: Overfitting and Dynamic Search Space Topology, *KDD*.

[24] Lal, D., Thomas, N., (2006). Embedded methods. *Feature extraction*. Springer Berlin Heidelberg.

[25] Kakade, Z., Sham M., Shai, S., Ambuj, T., (2012). Regularization techniques for machine learning with matrices, *Journal of Machine Learning Research*, 1865-1890.

[26] Zhao, Z., Huan, L., (2007). Semi-supervised Feature Selection via Spectral Analysis, *Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics.

[27] Chapelle, O., Bernhard, S., Alexander, Z., (2009). Semi-Supervised Learning, *IEEE Transactions on Neural Networks*, **3**, 542-542.

[28] Zhu, X., (2005). Semi-supervised learning literature survey.

[29] Chung, F., He, X., (1996). Lectures on spectral graph theory, *CBMS Lectures, Fresno*.

[30] He, X., Deng C., (2005). Laplacian score for feature selection, *Advances in neural information processing systems*.

[31] Belkin, M., Partha, N., (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering, **14**.

[32] He, X., Partha, N., (2004). Locality Preserving Projections, *Advances in Neural Information Processing Systems.*

[33] Zhu, L., Linsong, M., Daoqiang, Z., (2012). Iterative laplacian score for feature selection, *Chinese Conference on Pattern Recognition*, Springer Berlin Heidelberg.

[34] Gu, Q., Zhenhui L., Jiawei, H., (2012). Generalized fisher score for feature selection, *arXiv preprint arXiv*.

[35] Benabdeslem, K., Hindawi, M., (2011). Constrained Laplacian score for semi-supervised feature selection, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg.

[36] Guthikonda, M., (2005). Kohonen self-organizing maps, *Wittenberg University*.

[37] Ho, T., (1998). The random subspace method for constructing decision forests, *IEEE transactions on pattern analysis and machine intelligence*, 832-844.

[38] Freund, Y., Robert E., (1996). Experiments with a new boosting algorithm, **96**.

[39] Parsons, L., Ehtesham, H., Huan, L., (2004). Subspace clustering for high dimensional data, a review, *ACM SIGKDD Explorations Newsletter*, 90-105.

[40] Bullinaria, A., (2004). Self-Organizing Maps, Fundamentals, *Introduction to Neural*.

[41] Breiman, L., (1996). Bagging predictors, *Machine learning*, 123-140.

[42] Schapire, A., Robert, E., (2003). The boosting approach to machine learning, An overview, *Nonlinear estimation and classification*, Springer New York, 149-171.

[43] Witten, H., Eibe, F., (2005). Data Mining: Practical machine learning tools and techniques.