Constrained Laplacian Score for Semi-supervised Feature Selection

Khalid Benabdeslem and Mohammed Hindawi

University of Lyon1 - GAMA, Lab.
43 Bd du 11 Novembre, 69622 Villeurbanne, France
kbenabde@univ-lyon1.fr,
mohammed.hindawi@insa-lyon.fr

Abstract. In this paper, we address the problem of semi-supervised feature selection from high-dimensional data. It aims to select the most discriminative and informative features for data analysis. This is a recent addressed challenge in feature selection research when dealing with small labeled data sampled with large unlabeled data in the same set. We present a filter based approach by constraining the known Laplacian score. We evaluate the relevance of a feature according to its locality preserving and constraints preserving ability. The problem is then presented in the spectral graph theory framework with a study of the complexity of the proposed algorithm. Finally, experimental results will be provided for validating our proposal in comparison with other known feature selection methods.

Keywords: Feature selection, Laplacian score, Constraints.

1 Introduction and Motivation

Feature selection is an important task in machine learning for high dimensional data mining. It is one of the effective means to identify relevant features for dimension reduction [1]. This task has led to improved performance for several UCI data sets [2] as well as for real-world applications over data such as digital images, financial time series and gene expression microarrays [3].

Generally, feature selection methods can be classified in three types: filter, wrapper or embedded. The filter model techniques examine intrinsic properties of the data to evaluate the features prior to the learning tasks [4]. The wrapper based approaches evaluate the features using the learning algorithm that will ultimately be employed [5]. Thus, they "wrap" the selection process around the learning algorithm. The embedded methods are locally specific to models during their construction. They aim to learn the feature relevance with the associated learning algorithm [6].

Moreover, Feature selection could be done in three frameworks according to class label information. The most addressed framework is the supervised one, in which feature relevance can be evaluated by their correlation with the class label [7]. In unsupervised feature selection, without label information, feature relevance can be evaluated by their capability of keeping certain properties of the data, such as the variance or the separability. It is considered as a much harder problem, due to the absence of class labels that would guide the search for relevant information [8].

The problem becomes more challenging when the labeled and unlabeled data are sampled from the same population. It is more adapted with real-world applications where labeled data are costly to obtain. In this context, the effectiveness of semi-supervised learning has been demonstrated [9]. The authors in [10] introduced a semi-supervised feature selection algorithm based on spectral analysis. Later, they exploited intrinsic properties underlying supervised and unsupervised feature selection algorithms, and proposed a unified framework for feature selection based on spectral graph theory [11]. The second known work in semi-supervised selection deals with a wrapper-type forward based approach proposed by [12] which introduced unlabeled examples to extend the initial labeled training set.

Furthermore, utilizing domain knowledge became an important issue in many machine learning and data mining tasks [13,14,15]. Several recent works have attempted to exploit pairwise constraints or other prior information in feature selection. The authors in [16] proposed an efficient algorithm, called SSDR (with different variants: SSDR-M, SSDR-CM, SSDR-CMU), which can simultaneously preserve the structure of original high-dimensional data and the pairwise constraints specified by users. The main problem of these methods is that the proposed objective function is independent of the variance, which is very important for the locality preserving for the features. In addition, the similarity matrix used in the objective function uses the same value for all pairs of data which are not related by constraints. The same authors proposed a constraint score based method [17,18] which evaluates the relevance of features according to constraints only. The method carries out with little supervision information in labeled data ignoring the unlabeled data part even if it is very large. The authors in [19] proposed to solve the problem of semi-supervised feature selection by a simple combination of scores computed on labeled data and unlabeled data respectively. The method (called C^4) tries to find a consensus between an unsupervised score and a supervised one (by multiplying both scores). The combination is simple, but can dramatically bias the selection for the features having best scores for labeled part of data and bad scores for the unlabeled part and vice-versa.

In the contrast of all cited methods, our proposal uses a new developed score by constraining the well known Laplacian score (unsupervised) [20] that we will detail in the next section. The idea behind our proposal is to assess the ability of features in preserving the local geometric structure offered by unlabeled data, while respecting the constraints offered by labeled data.

Therefore, our semi-supervised feature selection algorithm is based on a filter approach. We think that one important motivation to have a filter method for feature selection is the specificity of the semi-supervised data. This is because, in this paradigm, data may be used in the service of both unsupervised and supervised learning. On the one hand, semi-supervised data could be used in the goal of data clustering, then using the labels to generate constraints which could in turns ameliorate the clustering. In this context, "good" features are those which better describe the geometric structure of data. On the other hand, semi-supervised data could be used for supervised learning, i.e. classification or prediction of the unlabeled examples using a classifier constructed from the labeled examples. In this context, "good" features are those which are better correlated with the labels. Subsequently, the use of a filter method makes the feature selection process independent from the further learning algorithm whether it is supervised or unsupervised. This is important to eliminate the bias of feature selection in both cases, i.e. good features in this case would be those which compromise between better description of data structure and better correlation with desired labels.

2 Related Work

In semi-supervised learning, a data set of N data points $X = \{x_1, ..., x_N\}$ consists of two subsets depending on the label availability: $X_L = (x_1, ..., x_l)$ for which the labels $Y_L = (y_1, ..., y_l)$ are provided, and $X_U = (x_{l+1}, ..., x_{l+u})$ whose labels are not given. Here data point x_i is a vector with m dimensions (features), and label $y_i \in \{1, 2, ..., C\}$ (C is the number of different labels) and $l + u = N$ (N is the total number of instances). Let $F_1, F_2, ..., F_m$ denote the m features of X and $f_1, f_2, ..., f_m$ be the corresponding feature vectors that record the feature value on each instance.

Semi-supervised feature selection is to use both X_L and X_U to identify the set of most relevant features $F_{j1}, F_{j2}, ..., F_{jk}$ of the target concept, where $k \leq m$ and $j_r \in \{1, 2, ..., m\}$ for $r \in \{1, 2, ..., k\}$.

2.1 Laplacian Score

This score was used for unsupervised feature selection. It not only prefers those features with larger variances which have more representative power, but it also tends to select features with stronger locality preserving ability. A key assumption in Laplacian Score is that data from the same class are close to each other. The Laplacian score of the r^{th} feature , which should be minimized, is computed as follows [20]:

$$L_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i (f_{ri} - \mu_r)^2 D_{ii}} \tag{1}$$

where D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$, and S_{ij} is defined by the neighborhood relationship between samples ($x_i = 1, .., N$) as follows:

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\lambda}} & if\ x_i\ and\ x_j\ are\ neighbors \\ 0 & otherwise \end{cases} \tag{2}$$

where λ is a constant to be set, and x_i, x_j are neighbors means that x_i is among k nearest neighbors of x_j , $\mu_r = \frac{1}{N}\sum_i f_{ri}$.

2.2 Constraint Score

In general, domain knowledge can be expressed in diverse forms, such as class labels, pairwise constraints or other prior information.

The constraint score guides the feature selection according to pairwise instance level constraints which can be classified on two sets: Ω_{ML} (a set of Must-Link constraints) and Ω_{CL} (a set of Cannot-Link constraints)

– **Must-Link constraint** (***ML***): involving x_i and x_j , specifies that they have the same label.
– **Cannot-Link constraint** (***CL***): involving x_i and x_j , specifies that they have different labels.

Constraint score of the r^{th} feature, which should be minimized, is computed as follows [17]:

$$C_r = \frac{\sum_{(x_i,x_j)\in\Omega_{ML}} (f_{ri} - f_{rj})^2}{\sum_{(x_i,x_j)\in\Omega_{CL}} (f_{ri} - f_{rj})^2} \tag{3}$$

3 Constrained Laplacian Score

The main advantage of Laplacian score is its locality preserving ability. However, its assumption that data from the same class are close to each other, is not always true. In fact, there are several cases where the classes overlap in some instances. Thus, two close instances could naturally have two different labels and vis-versa. Furthermore, for constraint score, the principle is mainly based on the constraint preserving ability. This few supervision information is certainly necessary for feature selection, but not sufficient when ignoring the unlabeled data part especially if it is very large. For that, we propose a Constrained Laplacian Score (CLS) which constraints the Laplacian score for an efficient semi-supervised feature selection. Thus, we define CLS, which should be minimized, as follows:

$$CLS_r = \frac{\sum_{i,j} (f_{ri} - f_{rj})^2 S_{ij}}{\sum_i \sum_{j|\exists k,(x_k,x_j)\in\Omega_{CL}} (f_{ri} - \alpha^i_{rj})^2 D_{ii}} \tag{4}$$

where :

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\lambda}} & if\ x_i\ and\ x_j\ are\ neighbors\ or\ (x_i, x_j) \in \Omega_{ML} \\ 0 & otherwise \end{cases} \tag{5}$$

and:

$$\alpha^i_{rj} = \begin{cases} f_{rj} & if\ (x_i, x_j) \in \Omega_{CL} \\ \mu_r & otherwise \end{cases} \tag{6}$$

Since the labeled and unlabeled data are sampled from the same population generated by target concept, the basis idea behind our score is to generalize the Laplacian score for semi-supervised feature selection. Note that if there are no labels ($l = 0, X = X_U$) then $CLS_r = L_r$ and when ($u = 0, X = X_l$), CLS represents an adjusted C_r, where the ML and CL information would be weighted by S_{ij} and D_{ii} respectively in the formula.

With CLS, on the one hand, a relevant feature should be the one on which those two samples (neighbors or related by an ML constraint) are close to each other. On the other hand, the relevant feature should be the one with a larger variance or on which those two samples (related by a CL constraint) are well separated.

4 Spectral Graph Based Formulation

The spectral graph theory [11] represents a solid theoretical framework which has been the basis of many powerful existing feature selection methods such as ReliefF [21], Laplacian [20] , sSelect [10], SPEC[22] and Constraint score [17].

Similarly, we give a graph based explanation for our proposed Constrained Laplacian Score (CLS). A reasonable criterion for choosing a relevant feature is to minimize the object function represented by CLS. Thus, the problem is to minimize the first term $T_1 = \sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}$ and maximize the second one $T_2 = \sum_i \sum_{j|\exists k,(x_k,x_j)\in \Omega_{CL}} (f_{ri} - \alpha^i_{rj})^2 D_{ii}$. By resolving these two optimization problems, we prefer those features respecting their pre-defined graphs, respectively. Thus, we construct a k-neighborhood graph G_{kn} from X (data set) and Ω_{ML} (ML constraint set) and a second graph G_{CL} from Ω_{CL} (CL constraint set).

Given a data set X, let $G(V, E)$ be the complete undirected graph constructed from X, with V is its node set and E is its edge set. The i^{th} node v_i of G corresponds to $x_i \in X$ and there is an edge between each nodes pair (v_i, v_j), whose weight $w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\lambda}}$ is the dissimilarity between x_i and x_j.

$G_{kn}(V, E_{kn})$ is a subgraph which could be constructed from G where E_{kn} is the edge set $\{e_{i,j}\}$ from E such that $e_{i,j} \in E_{kn}$ if $(x_i, x_j) \in \Omega_{ML}$ or x_i is one of the k-neighbohrs of x_j. $G_{CL}(V_{CL}, E_{CL})$ is a subgraph constructed from G with V_{CL} its node set and $\{e_{i,j}\}$ its edge set such that $e_{i,j} \in E_{CL}$ if $(x_i, x_j) \in \Omega_{CL}$.

Once the graphs G_{kn} and G_{CL} are constructed, their weight matrices, denoted by S^{kn} and S^{CL} respectively, can be defined as:

$$S^{kn}_{ij} = \begin{cases} w_{ij} & if\ x_i\ and\ x_j\ are\ neighbors\ or\ (x_i, x_j) \in \Omega_{ML} \\ 0 & otherwise \end{cases} \tag{7}$$

$$S^{CL}_{ij} = \begin{cases} 1 & if (x_i, x_j) \in \Omega_{CL} \\ 0 & otherwise \end{cases} \tag{8}$$

Then, we can define :

- For each feature r, its vector $f_r = (f_{r1}, ..., f_{rN})^T$
- Diagonal matrices $D^{kn}_{ii} = \sum_j S^{kn}_{ij}$ and $D^{CL}_{ii} = \sum_j S^{CL}_{ij}$
- Laplacian matrices $L^{kn} = D^{kn} - S^{kn}$ and $L^{CL} = D^{CL} - S^{CL}$

Algorithm 1. CLS

```
**Input:** Data set $X$
1: Construct the constraint set ($\Omega_{ML}$ and $\Omega_{CL}$) from $Y_L$
2: Construct graphs $G_{kn}$ and $G_{CL}$ from $(X, \Omega_{ML})$ and $\Omega_{CL}$ respectively.
3: Calculate the weight matrices $S^{kn}$, $S^{CL}$ and their Laplacians $L^{kn}$, $L^{CL}$ respectively.
for $r = 1$ to $m$ do
    4: Calculate $CLS_r$
end for
5: Rank the features $r$ according to their $CLS_r$ in ascending order.
```

Following some simple algebraic steps, we see that:

$$T_1 = \sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}^{kn} = \sum_{i,j}(f_{ri}^2 + f_{rj}^2 - 2f_{ri}f_{rj})S_{ij}^{kn} \tag{9}$$

$$= 2(\sum_{i,j} f_{ri}^2 S_{ij}^{kn} - \sum_{i,j} f_{ri} S_{ij}^{kn} f_{rj}) \tag{10}$$

$$= 2(f_r^T D^{kn} f_r - f_r^T S^{kn} f_r) \tag{11}$$

$$= 2 f_r^T L^{kn} f_r \tag{12}$$

Note that satisfying the graph-strutures is done according to α_{rj}^i in the equation (6). In fact, when $\Omega_{CL} = \varnothing$, we should maximize the variance of f_r which would be estimated as:

$$var(f_r) = \sum_i (f_{ri} - \mu_r)^2 D_{ii}^{kn} \tag{13}$$

The optimization of (13) is well detailed in [20]. In this case, $CLS_r = L_r = \frac{f_r^T L^{kn} f_r}{f_r^T D^{kn} f_r}$. Otherwise, we develop as above the second term (T_2) and obtain $2 f_r^T L^{CL} D^{kn} f_r$. Subsequently, $CLS_r = \frac{f_r^T L^{kn} f_r}{f_r^T L^{CL} D^{kn} f_r}$ seeks those features that respect G_{kn} and G_{CL}. The whole procedure of the proposed CLS is summarized in Algorithm 1.

Lemma 1. *Algorithm 1 is computed in time $O(m \times max(N^2, Log\, m))$.*
Proof. The first step of the algorithm requires l^2 operations. Steps 2-3 build the graph matrices requiring N^2 operations. Step 4 evaluates the m features requiring mN^2 operations and the last step ranks features according to their scores with $m\, Log(m)$ operations. $\square$

Note that the "small-labeled" problem becomes an advantage in our case, because it supposes that the number of extracted constraints is smaller since it depends on the number of labels, l. Thus, the cost of the algorithm depends considerably on u, the size of unlabeled data X_U.

To reduce this complexity, we propose to apply a clustering on X_U. The idea aims to substitute this huge part of data by a smaller one $X_U' = (p_1, ..., p_K)$ by preserving the geometric structure of X_U, where K is the number of clusters. We propose to use Self-Organizing Map (SOM) based clustering [23] that we briefly present in the next section.

Lemma 2. By clustering X_U the complexity of *Algorithm 1 is reduced to* $O(m \times max(u, Log\, m))$.

Proof. The size of labeled data is very smaller than the one of unlabeled data, $l << u < N$ and the clustering of X_U provides at most $K = \sqrt{u}$ clusters. Therefore, Algorithm 1 is applied over a data set with size equal to $\sqrt{u}+l \simeq \sqrt{u}$. This allows to decrease the complexity to $O(m \times max(u, Log\, m))$. □

4.1 SOM Algorithm

SOM is a very popular tool used for visualizing high dimensional data spaces. It can be considered as doing vector quantization and/or clustering while preserving the spatial ordering of the input data rejected by implementing an ordering of the codebook vectors (also called prototype vectors, cluster centroids or reference vectors) in a one or two dimensional output space. The SOM consists of nodes organized on a regular low-dimensional grid, called the map. More formally, the map is described by a graph (V, E). V is a set of K interconnected nodes having a discrete topology defined by E. For each pair of nodes (c, s) on the map, the distance $\delta(c, s)$ is defined as the shortest path between c and s on the graph. This distance imposes a neighborhood relation between nodes.

Each node c is represented by an m-dimensional reference vector $p_c = p_c^1, ..., p_c^m$ from $\mathcal{M}$ (the set of all map's nodes), where m is equal to the dimension of the input vectors $x_i \in X_U$ (unlabeled data set). The SOM training algorithm resembles K-means. The important distinction is that in addition to the best matching reference vector, its neighbors on the map are updated.

More formally, we define an assignment function γ from $\mathbb{R}^m$ (the input space) to $\mathcal{M}$ (the output space), that associates each element x_i of $\mathbb{R}^m$ to the node whose reference vector is "closest" to x_i. This function induces a partition $P = P_c; c = 1...K$ of the set of observations where each part P_c is defined by: $P_c = \{x_i \in X_U; \gamma(x_i) = c\}$.

Next, an adaptation step is performed when the algorithm updates the reference vectors by minimizing a cost function, noted $\mathcal{E}(\gamma, \mathcal{M})$. This function has to take into account the inertia of the partition P, while insuring the topology preserving property. To achieve these two goals, it is necessary to generalize the inertia function of P by introducing the neighborhood notion attached to the map. In the case of individuals belonging to $\mathbb{R}^m$, this minimization can be done in a straight way. Indeed, new reference vectors are calculated as:

$$p_s^{t+1} = \frac{\sum_{i=1}^{u} h_{sc}(t) x_i}{\sum_{i=1}^{u} h_{sc}(t)} \tag{14}$$

where $c = arg\min_s \|x_i - w_s\|$, is the index of the best matching unit of the data sample x_i, $\|.\|$ is the distance mesure, typically the Euclidean distance, and t denotes the time. $h_{sc}(t)$ is the neighborhood function around the winner unit c.

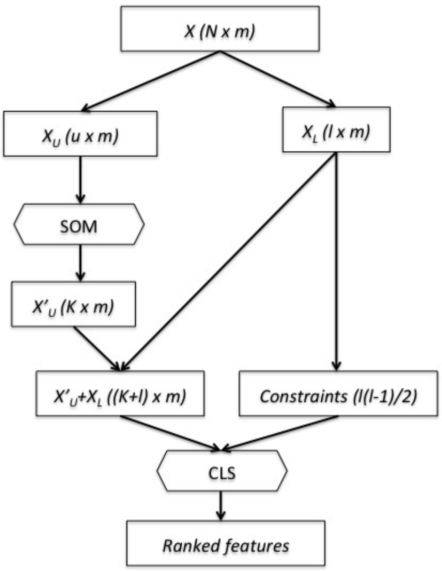

Fig. 1. Semi-supervised feature selection framework

In practice, we often use $h_{sc} = e^{-\frac{\delta_{sc}}{2T^2}}$ where T represents the neighborhood raduis in the map. It is decreased from an initial value T_{max} to a final value T_{min}.

Subsequently, as explained above, SOM will be applied on the unsupervised part of data (X_U) for obtaining X'_U with a size equal to the number of SOM' nodes (K). Therefore, CLS will be performed on the new obtained data set ($X_L + X'_U$). Note that any other clustering method could be applied over X_U, but here SOM is chosen for its ability to well preserve the topological relationship of data and thus the geometric structure of their distribution. Finally, the feature selection framework is represented in the Figure 1.

5 Results

5.1 Data Sets and Methods

In this section, we present an empirical study on several databases downloaded from different repositories. "Iris", "Wave", "Ionosphere", "Sonar" and "Soybean" in [2]. Microarray data sets, "Leukemia" and "Colon cancer" in [24] and [25] respectively. Face-image data sets, "Pie10P" and "Pix10P" which can be found in http://featureselection.asu.edu/datasets.php. The whole data sets information is detailed in Table 1.

The data sets are voluntarily chosen for evaluating the clustering performance of our proposal, CLS, and comparing it with other state of the art techniques. The concerned methods are listed below:

Table 1. Data sets

Data sets	N	m	$\#classes$
Iris	150	4	3
Wave	5000	40	3
Ionosphere	351	34	2
Sonar	208	60	2
Soybean	47	35	4
Leukemia	72	7129	2
Colon cancer	62	2000	2
Pie10P	210	2420	10
Pix10P	100	10000	10

- Variance score, is based on variance for feature selection [26].
- Fisher score, is based on variance and all labels for feature selection [27].
- Laplacian score, is only based on geometric structure of data [20].
- Constraint score (CS or CScore), selects the feature according to few supervision information, extracted from labeled data [17].
- C^4, is a semi-supervised feature selection by a simple combination of Laplacian score and CS [19].
- ReliefF, estimates the significance of features according to how well their values distinguish between the instances of the same and different classes that are near to each other [21].
- F2+r[4] and F3+r (SPEC), spectral feature selection methods [22].

The experimental results will be presented on three folds. First, we test our algorithm on data sets whose the relevant features are known. Second, we do some comparisons with known powerful feature selection methods and finally, we apply the algorithm on databases with huge number of features. In most experiments, the λ value is set to 0.1 and $k = 10$ for building the neighborhood graph. For the semi-supervised data, we chose the first labeled examples for all data sets (with different labels). We did no selection neither on the level of examples to be labeled, nor on the generated constraints.

5.2 Validation of Feature Selection

In this section, we are particularly interested on the two first data sets ("Iris" and "Wave") which are popularly used in machine learning and data mining tasks.

In "Iris", one class is linearly separable from the other two which are not linearly separable from each other. Out of the four features it is known that the features F3 (petal length) and F4 (petal width) are more important for the underlying clusters than F1 (sepal length) and F2 (sepal width) Figure 2. The sub-figure (c) shows the data projected on the subspace constructed by F3 and F4, whereas the sub-figure (b) shows the data projected on the subspace of F1 and F2. In [20], it was reported that by using variance score [26], the

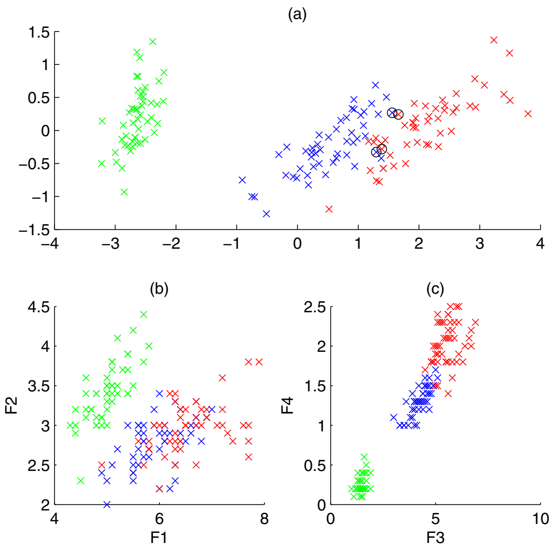

Fig. 2. 2D-Visualization of "Iris"

four features are sorted as (F3, F1, F4, F2). With $k \geq 15$, Laplacian score sorts these four features as (F3, F4, F1, F2). It sorts them as (F4, F3, F1, F2) when $3 \leq k < 15$. By using CLS, the features are sorted as (F3, F4, F1, F2) for any value of k (between 1 and 20). For explaining the difference between the two scores, we chose for this data set, $l = 10$ generating 45 constraints. Two of CL-type constraints are constructed from the pairs $(73^{th}, 150^{th})$ and $(78^{th}, 111^{th})$ according to the labels of the points Figure.2(a)[1] (The concerned points are represented by rounds). Since, the data points between brackets are close, with the Laplacian score, the edges $e_{73,150}$ and $e_{78,111}$ are constructed in the associated k-neighborhood graph and affect the feature selection process. With our method, these edges never exist because of the CL constraint property even if k is small. For that, the scores obtained by CLS are smaller than the ones obtained by Laplacian score. We also observed an important gap on scores between the relevant variables ($CLS_3 = 1.4 \times 10^{-3}$, $CLS_4 = 2.7 \times 10^{-3}$) and the irrelevant ones ($CLS_1 = 1.07 \times 10^{-2}$, $CLS_2 = 1.77 \times 10^{-2}$). In fact, In the region where the points belong to the two non-linearly separable classes, Laplacian score is biased by the dissimilarity which could affect the ranking of features for their selection, while CLS is able to control this problem with the help of constraints.

The waveform of Brieman data set "Wave" consists of 5000 instances divided into 3 classes. This data set is composed of 21 relevant features (the first ones) and 19 noise features with mean 0 and variance 1. Each class is generated from a combination of 2/3 "base" waves. We tested our feature selection algorithm

[1] Figure 2(a) is obtained by PCA.

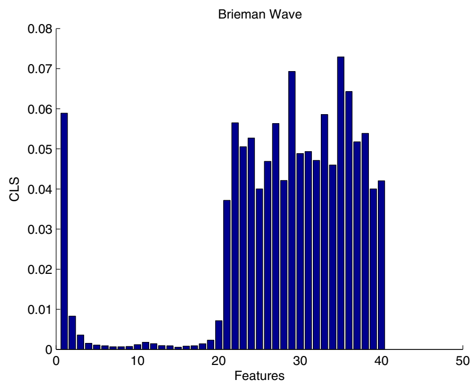

Fig. 3. Results of CLS on features of "Wave" data set

with $l = 8$ (28 constraints) and the dimension of the map (26×14) for SOM algorithm. We can see in Figure 3 that the features (21 to 40) have high values on CLS. The noise represented by these features is clearly detected.

5.3 Comparison of the Feature Selection Quality

For comparing our feature selection approach with another ones, the nearest neighborhood (1-NN) classifier with Euclidean distance is employed for classification after feature selection. For each data set, the classifier is learned in the first half of samples from each class and tested on the remaining data. We tested the Accuracy behavior of the ranking feature function represented by CLS for comparing it with those of other methods cited in [17]. These experiments were applied on three data sets "Ionosphere", "Sonar" and "Soybean" with 5 labeled instances for each one (so 10 pairwise constraints were generated).

Figure 4 indicates that, in most cases, the performance of CLS is comparable to Fisher Score [26] and significantly better than that of Variance, Laplacian and Constraint scores. This verifies that merging supervision information of labeled data with geometrical structure of unlabeled data is very useful in learning feature scores. Table 2 compares the averaged accuracy under different number of selected features. Here the values after the symbol $\pm$ denote the standard deviation. From Table 2 and Figure 4 we can find that, the performance of CLS is almost always better than that of Variance, Laplacian score and Constraint score and is comparable with Fisher Score. More specifically, CLS is superior to Fisher Score on "Soybean" and "Ionosphere" and is inferior on "Sonar". Note that Fisher score uses all labels when CLS score uses just 5 labels for each data set.

Then, we compare the performance of CLS with that of Fisher and constraint scores when different levels of supervision are used. Figure 5 shows the plots for accuracy under desired number of selected features vs. different numbers of labeled data (for Fisher Score) or pairwise constraints (for CScore and CLS) on the three data sets ("Ionosphere", "Sonar" and "Soybean"). Here the desired

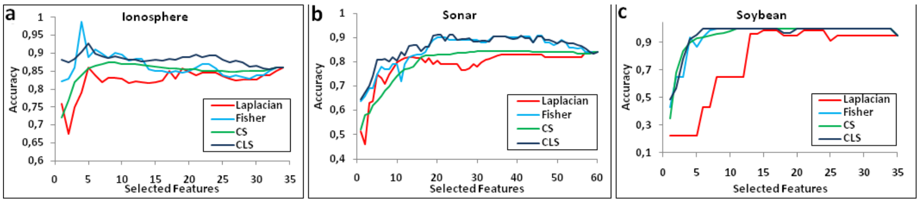

Fig. 4. Accuracy vs different numbers of selected features

Table 2. Averaged accuracy of different algorithms on "Ionosphere", "Sonar" and "Soybean"

Data sets	Variance	Laplacian	Fisher	CS	CLS
Ionosphere	82.2±3.8	82.6±3.6	86.3±2.5	85.1±2.9	**86.73±2.1**
Sonar	79.3±6.3	79.5±7.2	**86.4±6.9**	80.7±7.8	83.3±1.7
Soybean	88.9±12.7	79.4±28.4	94.5±12.1	93.5±11.6	**95.06±1.3**

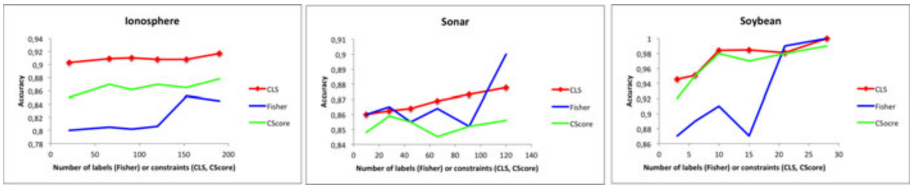

Fig. 5. Accuracy vs. different numbers of labeled data (for Fisher Score) or pairwise constraints (for CScore and CLS)

number of selected features is chosen as half of the original dimension of samples. For all scores, the results are averaged over 100 runs. As shown in Figure 5, except on "Sonar", CLS is much better than the other two algorithms especially when only a few labeled data or constraints are used. On "Sonar", both CScore and CLS are inferior to Fisher Score when the number of labeled data (or constraints) is great; CLS is always better when this number is small. A closer study on Figure 5 reveals that, generally, the accuracy of CLS increases steadily and fast in the beginning (with few constraints) and slows down at the end (with relatively more constraints). It implies that too many constraints won't help too much to further boost the accuracy, and only a few constraints are required in CLS, which corresponds exactly to our initial problem concerning "small-labeled" data. While Fisher Score typically requires relatively more labeled data to obtain a satisfying accuracy.

5.4 Results on Gene Expression Data Sets

"Leukemia" and "Colon cancer" are gene expression databases with huge number of features. The microarray Leukemia data is constituted of a set of 72

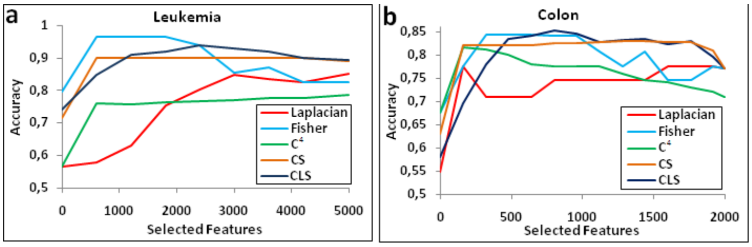

Fig. 6. Accuracy vs. different numbers of selected features on gene expression data sets

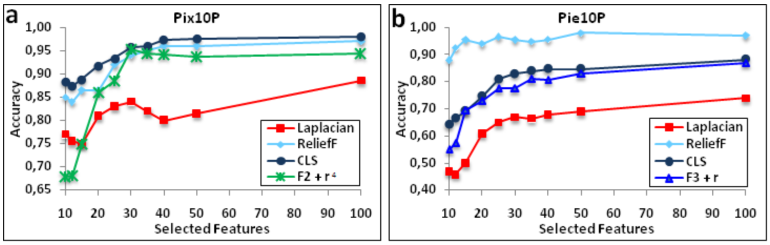

Fig. 7. Accuracy vs. different numbers of selected features on face-image data sets

samples, corresponding to two types of Leukemia called ALL (Acute Lymphocytic Leukemia) and AML (Acute Myelogenous Leukemia), with 47 ALL and 25 AML. The data set contains expressions for 7129 genes. While "Colon cancer" is a data set of 2000 genes measured on 62 tissues (40 tumors and 22 "normal"). We present our results on these data sets on comparison with Laplacian, Fisher, C^4 and CS scores, and that in case of Accuracy vs. Selected features. The results (Figure6) show that CLS records a comparable performance with other scores when the number of features is inferior to 2500 for "Leukemia" data set, and 500 for "Colon cancer" data set, then the performance of CLS is superior to other scores performance when increasing the number of features.

5.5 Results on Face-Image Data Sets

"Pie10P" and "Pix10P" are face-image data sets, each containing 10 persons. The validation on these data sets is presented in comparison with Laplacian, ReliefF scores on both data sets. In addition, results were compared with ($F2+r^4$) score on "Pix10P" data set and with (F3+r) score on "Pie10P" data set. We chose to compare our results with (F3+r) and ($F2+r^4$) because they achieved best results over the other variant scores proposed by authors in [22]. Experimentation results in Figure7 show that CLS outperforms significantly the other scores whatever the exploited number of features. Meanwhile, on "Pie10P" data set, CLS is higher than Laplacian and (F3+r) scores and inferior to ReliefF. Nevertheless, it could be shown that CLS has an excellent accuracy on "Pix10P" data set and very good one on "Pie10P" data set.

6 Conclusion

In this paper, we proposed a filter approach for semi-supervised feature selection. A new score function was developed to evaluate the relevance of features based on both, the locally geometrical structure of unlabeled data and the constrains preserving ability of labeled data. In this way, we combined two powerful scores, unsupervised and supervised, in a new one which is more generic for a semi-supervised paradigm. The proposed score function was explained in the spectral graph theory framework with the study of the complexity of the associated algorithm. For reducing this complexity we proposed to cluster the unlabeled part of data by preserving its geometrical structure before feature selection. Finally, experimental results on five UCI data sets and one microarray database show that with only a small number of constraints, the proposed algorithm significantly outperforms other filter based features selection methods.

There are a number of interesting potential avenues for future research. The choice of (λ, k) is discussed in [20] and [10], we tried to keep the same values that the authors used in their experiments in order to compare with their results. But even, the study of the influence of (λ, k) on our function score (with the treatment of constraints) is still interesting.

Another line of our future work is to study the constraint utility before integrating them for feature selection. In our proposal, we used the maximum number of constraints which could be generated from the labeled data. This could have ill effects over accuracy when constraints are incoherent or inconsistent. It would be thus more interesting to investigate in constraint selection for more efficient semi-supervised feature selection.

References

1. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(2), 153–158 (1997)
2. Frank, A., Asuncion, A.: Uci machine learning repository. Technical report, University of California (2010)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research (3), 1157–1182 (2003)
4. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the Twentieth International Conference on Machine Learning (2003)
5. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97(12), 273–324 (1997)
6. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by local linear embedding. Science (290), 2323–2326 (2000)
7. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(3), 131–156 (2000)
8. Dy, J., Brodley., C.E.: Feature selection for unsupervised learning. Journal of Machine Learning Research (5), 845–889 (2004)
9. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning. The MIT Press, Cambridge (2006)

10. Zhao, Z., Liu, H.: Semi-supervised feature selection via spectral analysis. In: Proceedings of SIAM International Conference on Data Mining (SDM), pp. 641–646 (2007)
11. Chung, F.: Spectral graph theory. AMS, Providence (1997)
12. Ren, J., Qiu, Z., Fan, W., Cheng, H., Yu, P.S.: Forward semi-supervised feature selection. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 970–976. Springer, Heidelberg (2008)
13. Basu, S., Davidson, I., Wagstaff, K.: Constrained clustering: Advances in algorithms, theory and applications. Chapman and Hall/CRC Data Mining and Knowledge Discovery Series (2008)
14. Xing, E., Ng, A., Jordan, M., Russel, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems, vol. 15, pp. 505–512 (2003)
15. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. Journal of Machine Learning Research 6, 937–965 (2005)
16. Zhang, D., Zhou, Z., Chen, S.: Semi-supervised dimensionality reduction. In: Proceedings of SIAM International Conference on Data Mining, SDM (2007)
17. Zhang, D., Chen, S., Zhou, Z.: Constraint score: A new filter method for feature selection with pairwise constraints. Pattern Recognition 41(5), 1440–1451 (2008)
18. Sun, D., Zhan, D.: Bagging constraint score for feature selection with pairwise constraints. Pattern Recognition 43(6), 2106–2118 (2010)
19. Kalakech, M., Biela, P., Macaire, L., Hamad, D.: Constraint scores for semi-supervised feature selection: A comparative study. Pattern Recognition Letters 32(5), 656–665 (2011)
20. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems, vol. 17 (2005)
21. Robnik-Sikonja, M., Kononenko, I.: Theoretical and empirical analysis of relief and relieff. Machine Learning 53, 23–69 (2003)
22. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the Twenty Fourth International Conference on Machine Learning (2007)
23. Kohonen, T.: Self Organizing Map. Springer, Berlin (2001)
24. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, L., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 15 286(5439), 531–537 (1999)
25. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Natl. Acad. Sci. 96(12), 6745–6750 (1999)
26. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
27. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley-Interscience, Hoboken (2000)