Constraint Selection-Based Semi-supervised Feature Selection.

Mohammed Hindawi, Kais Allab, Khalid Benabdeslem

To cite this version:

Mohammed Hindawi, Kais Allab, Khalid Benabdeslem. Constraint Selection-Based Semi-supervised Feature Selection.. Cook, Diane J. and Pei, Jian and 0010, Wei Wang and Zaïane, Osmar R. and Wu, Xindong. ICDM, Dec 2011, Vancouver,BC, Canada. IEEE, pp.1080-1085, 2011, 10.1109/ICDM.2011.42 . hal-00874960

HAL Id: hal-00874960
https://hal.science/hal-00874960

Submitted on 21 Oct 2013

Constraint Selection based Semi-supervised Feature Selection

Mohammed Hindawi
INSA of Lyon
20, avenue Albert Einstein
69100 Villeurbanne, France
mohammed.hindawi@insa-lyon.fr

Kaïs Allab
University of Lyon1
43, Boulevard du 11 Nov. 1918
69100 Villeurbanne, France
kais.allab@etu.univ-lyon1.fr

Khalid Benabdeslem
University of Lyon1
43, Boulevard du 11 Nov. 1918
69100 Villeurbanne, France
khalid.benabdeslem@univ-lyon1.fr

ABSTRACT

Dimensionality reduction is a significant task when dealing with high-dimensional data, this reduction can be done by feature selection, which means to select the most appropriate features for data analysis. It is a recent addressed challenge in feature selection research when handling small-labeled with large-unlabeled data sampled from the same population. The supervision information may be used in the form of pairwise constraints; these constraints have practically proven to have very positive effects on the learning performance. Nevertheless, selected constraints sets may have significant results (positive or negative) on learning performance. In this paper, we present a novel feature selection approach based on an efficient selection of pairwise constraints. This aims to grasp the most coherent constraints extracted from labeled party of data. We then evaluate the relevance of a feature according to its 'efficient' locality preserving and 'chosen' constraints preserving ability. Finally, experimental results will be provided for validating our proposal in comparison with other known feature selection methods.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology – Feature evaluation and selection

General Terms

Algorithms, Theory

Keywords

Dimensionality reduction, constraint selection, feature selection, pairwise constraints

1. INTRODUCTION

The rapid development of data acquisition tools has increased the accumulation of high-dimensional data, such as digital images, financial time series and gene expression microarrays. It was proved that the high-dimensionality can deteriorate the performance of data mining and machine learning process. Moreover, a too high sample-size to the number-of-dimensions ratio may result in the infeasibility of learning due to the “curse of dimensionality” [8]. Hence, the dimensionality reduction has become a fundamental tool for many data mining tasks. One of the existing methods to overcome this problem is the feature selection and extraction techniques [25]. Feature extraction methods can be categorized according to the viewpoint of label information availability into supervised and unsupervised ones. Fisher Linear Discriminant (FLD) [10], is an example of supervised feature extraction methods, which can extract the optimal discriminant vectors when class labels are available. For unsupervised feature extraction methods, Principal Component Analysis (PCA) [16] is an example that tries to preserve the global covariance structure of data when class labels are not available.

Similar to feature extraction, feature selection is one of the effective means to identify relevant features for dimensionality reduction [15]. Feature selection is a well addressed problem in machine learning and data mining communities. It is one of the effective means to identify relevant features for dimension reduction [13]. In fact, this task became very necessary with the accumulation of data having a huge number of features, that could have ill effect over learning algorithms. Moreover, feature selection has been well addressed in supervised and unsupervised paradigms with several works [9][13].

In the supervised feature selection context, the relevance of a feature can be evaluated by its correlation with the class label. In this context, the most known feature selection method is Fisher score [8], which seeks features with best discriminant ability with full class labels on the whole training data. Other powerful supervised feature selection methods, called ReliefF and RReliefF, were proposed by Robnik-Šikonja and al. [20]. The key idea behind these methods is to estimate 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. However, ReliefF and RReliefF do not help with removing redundant features. In [22], the authors proposed a new concept, predominant correlation, and proposed a fast filter method (FCBF) which can identify relevant features as well as redundancy among relevant features without pairwise correlation analysis. Hence, the authors in [27] proposed SPEC, a general framework of feature selection, and demonstrated that ReliefF is a special case of their proposal.

The unsupervised feature selection is considered as a much harder problem, due to the absence of class labels that would guide the

KDD'11, August 21–24, 2011, San Diego, CA, USA.

search for relevant information. In this way, the feature relevance can be evaluated by their capability of keeping certain properties of the data, such as the variance or the separability. Variance score [3] might be the simplest unsupervised evaluation of the features. It uses the variance along a dimension to reflect its representative power and those features with the maximum variance are selected. Another unsupervised feature selection method, Laplacian Score [14], makes a further step on Variance. It not only favors those features with larger variances which have more representative power, but also tends to selecting features with stronger locality preserving ability. This method is also generalized by the SPEC method in the unsupervised context. Laplacian score belongs to spectral feature selection family and will be more discussed in the next section.

The problem becomes more challenging when the data contains labeled and unlabeled examples 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 [4]. The authors in [26] introduced a semi-supervised feature selection algorithm (sSelect) 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 [5].

Furthermore, since domain knowledge became an important issue in many data mining tasks [2][21]; Several recent works have attempted to exploit pairwise constraints or other prior information in dimensionality reduction. Bar-Hillel and al. [1] proposed the constrained FLD (cFLD) for dimensionality reduction from equivalence constraints, as an interim-step for Relevant Component Analysis (RCA). However, cFLD can only deal with the must-link constraints. Zhang and al. [24] 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 [23] 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 party even if is very large. The authors in [17] 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 (Laplacian) and a supervised score (Constraint). The combination is simple, but can dramatically bias the selection for the features having best scores for labeled party of data and bad scores for the unlabeled party and vice-versa.

Supervision information is not limited to class labels only. Actually, the background information could be expressed by class labels, pairwise constraints, or any other prior information. In this paper we focus on the pairwise constraints, which are an instance level constraints, that specify that two instances have to be at the same class (Must-link constraints) or different classes (Cannot-link constraints). These constraints may be easier to obtain than class labels in some domains where it is hard to have an early decision on class label. In addition, these constraints can be generated from labeled data directly.

The importance of pairwise is practically proven, nevertheless, and unlikely to what might be expected; some constraint sets actually can decrease the learning performance [7]. Hence, the exploitation of constraint selection can result in more “useful” constraint sets to be presented to data.

In this paper, we present a general framework for semi-supervised dimensionality reduction. This framework is based on efficient selection of pairwise constraints (CSFS). This proposal uses a new developed score by efficiently combining the power of the local geometric structure offered by unlabeled data, with the constraint preserving ability offered by labeled data.

The rest of this paper is structured as follows: In section 2 we illustrate the related works through two powerful scores that we used to inspire our score function; in addition we demonstrate the constraint selection measure that we deploy in constraint selection. In section 3 we present a full description of our CSFS framework. We introduce in section 4 a spectral formulation of our score function with the technique we use to reduce the complexity caused by high-dimension data. The section 5 shows the results of our framework on real data sets with the comparisons of well known dimension reduction techniques. We then conclude our work in section 6 with the perspectives and possible forward research avenues.

2. RELATED WORKS

In this section we will discuss two scores on which we based to inspire the score function of our framework, we will illustrate in details the Laplacian score [14], and the constraint score [23] in addition to their limitations, but firstly we would present a formal definition of feature selection in semi-supervised learning.

In semi-supervised learning, a data set of N data points $\mathrm{X} = \{x_1, \ldots, x_\mathrm{N}\}$ consists of two subsets depending on the label availability: $X_L = (x_1, x_2, \ldots, x_l)$ for which the labels $Y_L = (y_1, y_2, \ldots, y_l)$ are provided, and $X = (x_{l+1}, x_{l+2}, \ldots, x_{l+u})$ which are non labeled. A data point x_i is a vector with m dimension (features), while label $y_i \in \{1, 2, \ldots, C\}$ (C is the number of different labels), and $l + u = N$ (N is the total number of instances). Let $F_1, F_2, \ldots, F_m$ denote the m features of X and $f_1, f_2, \ldots, f_m$ 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 $F_{j1}, F_{j2}, \ldots, F_{jk}$ of the target concept, where $k \le m$ and $j_r \in \{1, 2, \ldots, m\}$ for $r \in \{1, 2, \ldots, k\}$.

2.1 Laplacian Score

This score is used for unsupervised feature selection. It prefers those features with larger variances which have more representative power. In addition, it 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 [14]:

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

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:

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{l}} & if\, x_i \text{ and } x_j \text{ are neighbors} \\ 0 & otherwise \end{cases} \qquad (2)$$

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

2.2 Constraint Score

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 [23]:

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

3. CONSTRAINT SELECTION FOR FEATURE SELECTION

The aforementioned scores recorded important results in certain application scenarios; nevertheless, they have some limitations:

– Laplacian score: this score investigates the variance of the data in addition to the locality preserving ability of the features. Hence, a "good" feature for this score is the one at which two neighboring examples record close values. However, this score does not profit from the background information (the *CL* constraints in particular), which are provided to guide the learning process. In addition, the neighborhood choice is not clearly defined, that is, the variety of (*k*) choices has significant effects on results. We will discuss this problem in (section 4.1).

– Constraint score: Utilizing few labels of data, this score recorded better results than Fisher score which employs the whole labeled in feature selection process [23]. Nevertheless, this score has several drawbacks :

1. It just exercises the labeled data in the feature selection; such vital restriction may mislead the learning process, especially in a semi-supervised context, where the labeled party is normally larger than the labeled one.
2. As this score depends merely on the chosen constraint subset. The choice of constraints is still a problematic issue, which could derogate the performance of the feature selection process.

In order to overcome the listed restrictions, we propose an approach that will:

– Deploy a constraint selection in order to select the coherent subset of pairwise constraints extracted from the labeled data.

– Utilize the data structure in the definition of the neighborhood between examples.

3.1 Constraint Selection

While it was expected that different constraints sets would contribute more or less in improving clustering accuracy, it was found that some constraints sets actually decrease clustering performance. It was observed that constraints can have ill effects even when they are generated from the data labels that are used to evaluate accuracy, so this behavior is not caused by noise or errors in the constraints. Instead, it is a result of the interaction between a given set of constraints and the algorithm being used. So it is more important to know why do some constraint sets increase clustering accuracy while others have no effect or even decrease accuracy. For that, the authors in [7] have defined two important measures, informativeness and coherence, that capture relevant properties of constraint sets. These measures provide insight into the effect a given constraint set has for a specific constrained clustering algorithm. In this paper, we only use the coherence measure, which is independent of any learning algorithm.

The coherence represents the amount of agreement between the constraints themselves, given a metric *d* that specifies the distance between points. It does not require knowledge of the optimal partition P^* and can be computed directly. The coherence of a constraint set is independent of the algorithm used to perform constrained clustering. One view of an *ML(x,y)* (or *CL(x,y)*) constraint is that it imposes an attractive (or repulsive) force within the feature space along the direction of a line formed by *(x,y)*, within the vicinity of *x* and *y*. Two constraints, one an *ML* constraint (*m*) and the other a *CL* constraint (*c*), are incoherent if they exert contradictory forces in the same vicinity. Two constraints are perfectly coherent if they are orthogonal to each other and incoherent if they are parallel to each other. To determine the coherence of two constraints, *m* and *c*, we compute the projected overlap of each constraint on the other as follows (see Fig. 1 for examples).

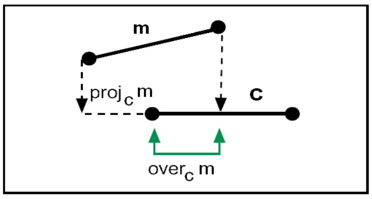

Figure 1. Projected overlap between two constraints: *ML(m)* over *CL(c)*. The coherence of the subset is null.

Let $\vec{m}$ and $\vec{c}$ be vectors connecting the points constrained by *m* and *c* respectively. The coherence of a given constraints set Ω is defined as a fraction of constraints pairs that have zero projected overlap:

$$Coh_d(\Omega) = \frac{\sum_{m\in\Omega_{ML},c\in\Omega_{CL}} \delta(over_c m = 0 \wedge over_m c = 0)}{|\Omega_{ML}||\Omega_{CL}|} \tag{4}$$

Where $over_c m$ represents the distance between the two projected points linked by m over c. δ is the number of the overlapped projections. More details can be found in [7].

From the equation(4), we can easily define a specific measure for each constraint as follows:

$$Coh_d(m) = \frac{\sum_{c\in\Omega_{CL}} \delta(over_c m = 0)}{|\Omega_{CL}|} \tag{5}$$

$$Coh_d(c) = \frac{\sum_{m\in\Omega_{ML}} \delta(over_m c = 0)}{|\Omega_{ML}|} \tag{6}$$

We show now how to select the relevant constraints according to their coherence. To be selected, a constrained α_i must be "fully" coherent, i.e. it must not overlap with any other constraint α_j ($\alpha_j \in \Omega_{CL}$ if $\alpha_i \in \Omega_{ML}$ and vice versa). This hard fashion to select constraints can be described in algorithm (Fig. 2).

```
Input:   Constraints set $\Omega = \{\alpha_i\}$
Initialize $\Omega_s = \emptyset$.
for $i = 1$ to $|\Omega|$ do
    if { $Coh_d(\alpha_i)=1$ }
        $\Omega_s = \Omega_s + \{\alpha_i\}$
    end if
end for
Output:  Selected constraints $\Omega_s$
```

Figure 2. Constraint Selection Algorithm

From this algorithm we obtain Ω_s, which is a set of coherent constraints of $ML(x_i,x_j)$, and $CL(x_i,x_j)$ in two subsets Ω'_{ML} and Ω_{CL} respectively.

3.2 Score Function

The advantage of Laplacian score is its survey of the respect of data structure, which is expressed by the variance and locality preserving ability. However, several studies proved that the exploitation of background information improves the performance of the learning process. Furthermore, for constraint score, the principle is mainly based on the constraint preserving ability. This little supervision information is certainly necessary for feature selection, but not sufficient when ignoring the unlabeled data party especially if it is very large.

For that, we propose a Constraint Selection for Feature Selection Score (φ) which constraints the Laplacian score by the Constraint score for an efficient semi-supervised feature selection. Thus, we define (φ) score, which should be minimized, as follows:

$$\varphi_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2(S_{ij} + N_{ij})}{\sum_{i,j}(f_{ri} - a^i_{rj})^2 D_{ii}} \tag{7}$$

Where:

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{l}} & if\, x_i \text{ and } x_j \text{ are neighbors} \\ 0 & otherwise \end{cases} \tag{8}$$

And:

$$N_{ij} = \begin{cases} -e^{-\frac{\|x_i - x_j\|^2}{l}} & if\, x_i \text{ and } x_j \text{ are neighbors and } (x_i,x_j)\in\Omega_{ML} \\ \left(e^{-\frac{\|x_i - x_j\|^2}{l}}\right)^2 & if\, x_i \text{ and } x_j \text{ are neighbors and } (x_i,x_j)\in\Omega_{CL} \\ & OR \\ & if\, x_i \text{ and } x_j \text{ are not neighbors and } (x_i,x_j)\in\Omega_{ML} \\ 0 & otherwise \end{cases} \tag{9}$$

$$a^i_{rj} = \begin{cases} f_{rj} & if\, (x_i,x_j)\in\Omega'_{CL} \\ \mu_r & otherwise \end{cases} \tag{10}$$

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 and the constraint scores for semi-supervised feature selection. Note that if there are no labels ($l = 0, X = X_U$) then $\varphi_r = L_r$ and when ($u = 0, X = X_l$), φ 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 φ score, 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.

To assess the previous concept, we use a weight N_{ij}. The motivation of adding N_{ij} to our score (over the Laplacian score) is not the integration of pairwise constraints into the score only, but it also adds a sensibility dimension to feature score in the following cases:

When we have two samples joint by a *ML* constraint but not neighbors $(S_{ij} + N_{ij}) = \left(e^{-\frac{\|x_i - x_j\|^2}{l}}\right)^2$, or when two neighboring samples are joints by a *CL* constraint $(S_{ij} + N_{ij}) = \left(e^{-\frac{\|x_i - x_j\|^2}{l}}\right)^2 + e^{-\frac{\|x_i - x_j\|^2}{l}}$. In both cases, the weight $\left(e^{-\frac{\|x_i - x_j\|^2}{l}}\right)^2$ is used in order to more differentiate the features in the both "bad cases".

4. SPECTRAL GRAPH BASED FORMULATION

In this section, we give a spectral graph based explanation for our proposed CSFS score (φ). A reasonable criterion for choosing a relevant feature is to minimize the objective function represented by φ. The principle consists thus to minimize the first term $T_1 = \sum_{i,j}(f_{ri} - f_{rj})^2(S_{ij} + N_{ij})$ and maximize the second one $T_2 = \sum_{i,j}(f_{ri} - a^i_{rj})^2 D_{ii}$. 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 node pair (v_i, v_j), the weight of this edge is the dissimilarity between x_i and x_j:

$$w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{l}}$$

$G_{kn}(V, E_{kn})$ 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-neighbors 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 $(S^{kn} + N^{kn})$ and S^{CL} respectively, can be defined as:

$$S^{kn}_{ij} = \begin{cases} w_{ij} & if\, x_i\; and\; x_j\; are\; neighbors \\ 0 & otherwise \end{cases} \tag{11}$$

$$N^{kn}_{ij} = \begin{cases} -w_{ij} & if\, x_i\; and\; x_j\; are\; neighbors\; and\; (x_i,x_j) \in \Omega_{ML} \\ & if\, x_i\; and\; x_j\; are\; neighbors\; and\; (x_i,x_j) \in \Omega_{CL} \\ w_{ij}^2 & OR \\ & if\, x_i\; and\; x_j\; are\; not\;\; neighbors\; and\; (x_i,x_j) \in \Omega_{ML} \\ 0 & otherwise \end{cases} \tag{12}$$

$$S^{CL}_{ij} = \begin{cases} 1 & if (x_i,x_j) \in \Omega_{CL} \\ 0 & otherwise \end{cases} \tag{13}$$

Then, we can define:

– For each feature r, its vector $f_r = (f_{r1}, \ldots, f_{rN})^T$

– Diagonal matrices $D^{kn}_{ii} = \sum_j S^{kn}_{ij}$, $D^{CL}_{ii} = \sum_j S^{CL}_{ij}$ and

$$DN^{kn}_{ii} = \sum_j N^{kn}_{ij}$$

– Laplacian matrices $L^{kn} = (D^{kn} + DN^{kn}) - (S^{kn} + N^{kn})$ and

$$L^{CL} = D^{CL} - S^{CL}$$

Following some simple algebraic steps, we see that:

$$T_1 = \sum_{i,j} (f_{ri} - f_{rj})^2 (S^{kn}_{ij} + N^{kn}_{ij}) = \sum_{i,j} (f_{ri}^2 + f_{rj}^2 - 2 f_{ri} f_{rj})(S^{kn}_{ij} + N^{kn}_{ij})$$
$$= 2\Big(\sum_{i,j} f_{ri}^2 (S^{kn}_{ij} + N^{kn}_{ij}) - \sum_{i,j} f_{ri}(S^{kn}_{ij} + N^{kn}_{ij}) f_{rj}\Big)$$
$$= 2\big(f_r^T (D^{kn} + DN^{kn}) f_r - f_r^T (S^{kn} + N^{kn}) f_r\big)$$
$$= 2 f_r^T L^{kn} f_r$$

Note that the respecting of different graph-structures is done according to a^i_{rj} in the equation(13). In fact, when $\Omega_{CL} = \emptyset$, we should G_{kn} maximize the variance of f_r this would be estimated as:

$$var(f_r) = \sum_i (f_{ri} - m_r)^2 D^{kn}_{ii} \tag{14}$$

The optimization of (14) is well detailed in [14]. In this case, $\varphi_r = L_r = \frac{f_r^T L^{kn} f_r}{f_r^T D^{kn} f_r}$.

Otherwise, we develop as above the second term (T_2) and obtain $2 f_r^T L^{CL} D^{kn} f_r$. Subsequently, $\varphi_r = \frac{f_r^T L^{kn} f_r}{f_r^T L^{CL} D^{kn} f_r}$ seeks those features that respect G_{kn} and G_{CL}.

The whole algorithm of the proposed score φ is summarized in (Fig. 3).

Note. The step 6 of the algorithm (Fig. 3) is computed in time $O(mN^2)$.

Notice 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 party of data by a smaller one $X_U' = (p_1, \ldots, p_K)$ by preserving the geometrical structure of X_U, where K is the number of clusters. We propose to use Self-Organizing Map (SOM) based clustering [18] which 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.

```
Input:   Data set $X$
1: Construct the constraint set ($\Omega_{ML}$ and $\Omega_{CL}$) from $Y_L$
2: Select the coherent set ($\Omega_{ML}$ and $\Omega_{CL}$) from ($\Omega_{ML}$ and $\Omega_{CL}$)
3: Construct graphs $G_{kn}$ and $G_{CL}$ from $(X, \Omega_{ML})$ and $\Omega_{CL}$ respectively.
4: Calculate the weight matrices $S^{kn}$, $S^{CL}$ and their Laplacians $L^{kn}$, $L^{CL}$ respectively.
5: Construct a clustering to Calculate $k_i$ for all examples
for $r = 1$ to $m$ do
6: Calculate $\varphi_r$
 end for
7: Rank the features $r$ according to their $\varphi_r$ in ascending order.
Output:  Ranked features
```

Figure 3. CSFS Feature Selection Algorithm

Lemma 1. By clustering X_U the complexity of step 6 in algorithm (Fig. 3) is reduced to $O(mu)$.

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, step 6 of the algorithm (Fig. 3) is applied over a data set with size equal to $\sqrt{u} + l$; $\sqrt{u}$. This allows decreasing the complexity to $O(mu)$. W

Subsequently, SOM will be applied on the unsupervised party of data (X_U) for obtaining $X_{U\text{¢}}$ with a size equal to the number of SOM' nodes (K). Therefore, φ will be performed on the new obtained data set ($X_L + X_{U\text{¢}}$).

4.1 Adaptive *k*-Neighborhood determination

The key assumption of Laplacian score is the assessment of locality preserving ability by features. Meanwhile, the principle of fixed *k*-nearest-neighbors for all instances may affect the locality preserving, because it is not guaranteed that the *k*-nearest-neighbors of an instance are "close" to it (Fig. 4-a). In this case, some "far" neighbors would be enrolled in the locality preserving measurement for the example at the hand.

Hence, we advise using a similarity based clustering approach on the whole instances, which allows revealing their locality structures. Then, the *k*-nearest-neighborhood relationship among them will depend on their membership to the same clusters. Hence, the adaptive *k* would be related to data structure and could be defined as follows: Two instances are neighbors if they belong to the same cluster. Consequently, each cluster has its own k which is the number of its elements (less one).

In (Fig. 4-b), calculating the score of X_1 does not need to look far, but it is calculated on the base of the instances belonging to its cluster. Accordingly, the score is less biased and the locality is more preserved.

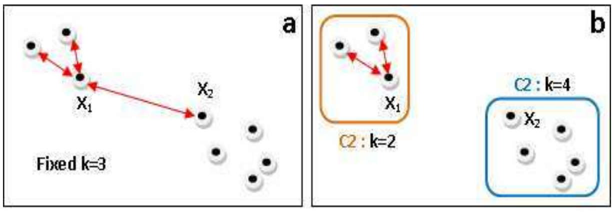

Figure 4. (a) Fixed k-nearest-neighborhood. (b) Adaptive k-nearest-neighborhood.

Finally, the feature selection framework is represented in (Fig. 5).

5. RESULTS

5.1 Data Sets

In this section, we present an empirical study on a broad range of data sets including four data sets downloaded from the UCI repository [11], i.e. *"Iris"*, *"Ionosphere"*, *"Sonar"* and *"Soybean"*. In addition we present the results on *"Leukemia"*, and *"colon cancer"* data sets, which can be found in [12][1] respectively. Moreover, for validating our framework on high-dimensional data, we present our results on *"Pie10P"* and *"Pix10P"* which are face image data sets containing 10 persons in each. The whole data sets information is detailed in (Table 3).

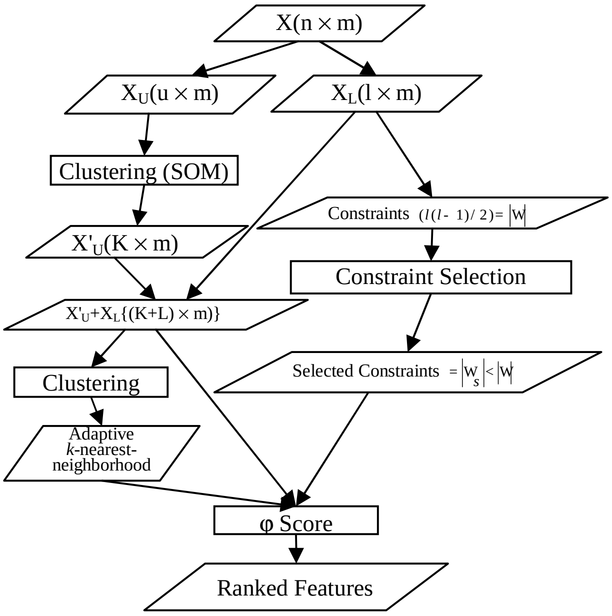

Figure 5. CSFS framework

For the construction of the SOM maps in the phase of unlabeled data clustering, we use a Principal Component Analysis (PCA) based heuristic proposed by Kohonen [18] for automatically providing the initial number of neurons and the dimensions of the maps (Table 1). The reference vectors are initialized linearly along the greatest eigenvectors of the associated data set X_U. Then, in order to determine the adaptive *k*-nearest-neighborhood constant, each SOM map is merged with the labeled data party X_l. The resulting data $(X_{U'} + X_l)$ is clustered by an Ascendant Hierarchical Clustering (AHC) for optimizing the number of clusters (by grouping neurons) [6]. In general, an internal index, like Davies Bouldin or Generalized Dunn [19], is used for cutting the dendrogram. Here, we used Davies Bouldin index to obtain the number of classes corresponding to the correct partition (#l) for each data set. Note that we obtain several values of *k* for each data set. These values are not manually determined but automatically settled based on the structure of each data set. For example, on Soybean data set, we obtained 4 clusters by (AHC). The numbers of instances belonging to clusters were 9, 16, 9 and 10 corresponding to 4 various values of *k* 8, 15, 8 and 9 respectively.

In order to compare our feature selection framework with other ones, the nearest neighborhood (1-NN) classifier with Euclidean distance is employed for classification. After feature selection phase, and for each data set, the classifier is learned in the first half of samples from each class and tested on the remaining data. In addition, the constant λ of our score function is set to 0.1 in all our experiments.

The empirical study that we would present is approached in four scenarios: Firstly, we would compare the performance of CSFS framework with Laplacian score; Constraint score and Fisher score, this comparison would be held on UCI data sets and concerns the accuracy of the classification vs. the number of selected features.

Secondly, we assess the relative performance of CSFS over other dimensionality reduction methods. We choose the PCA as the baseline. We also compare the performance of CSFS with SSDR-CMU and cFLD under different level of constraints. This comparison would be held also on UCI data sets but it concerns the classification accuracy vs. the number of selected constraints (while fixing the number of selected features).

The third scenario would be presented on higher-dimensional data, i.e. Leukemia and Colon Cancer. We evaluate the performance of CSFS framework on these data sets in comparison with Laplacian, Fisher, C[4] and CS scores. This comparison will concern the classification accuracy vs. both selected features and selected constraints (Laplacian score – fully unsupervised- is not applicable in accuracy vs. selected constraints case).

Table 1 : Data sets

Data set	N	m	#Class	Map' dimensions
Iris	150	4	3	11×5
Ionosphere	351	34	2	12×7
Sonar	208	60	2	9×7
Soybean	47	35	4	8×4
Leukemia	72	7129	2	7×5
Colon cancer	62	2000	2	6×6
Pie10P	210	2400	10	8×5
Pix10P	100	10000	10	9×7

Finally, we would validate our CSFS framework on higher dimensional images data sets, i.e. Pie10P and Pix10P. This validation is presented in comparison with Laplacian, ReliefF, F2+r[4] and F3+r scores.

In our experiments, we simulated the generation of pairwise constraints as follows:

We randomly selected samples of 25% from the labeled data belonging to each class, and then we created the must-link and cannot-link constraints depending on the underlying classes. Finally, we deployed our constraint selection framework in order to choose the most coherent subset of these constraints.

5.2 Results on UCI Data Sets

In this section we assess the relative performance of CSFS over other dimensionality reduction methods for classification. We choose the fully unsupervised Laplacian score as the baseline. We also test the performance of supervised Fisher score which uses the class labels of all the training data. We compare CSFS results with constraint score ones too. As mentioned before, after dimensionality reduction, nearest neighborhood (1-NN) classifier is employed for classification. In addition, the coherent constraints exploited on data sets are: (8 for Iris, 13 for Ionosphere,11 for Sonar and 7 for Soybean.

(Fig. 6) shows that CSFS always achieves the highest accuracy on all data sets. It can also be shown that in most cases the performance of Laplacian score is the worst. We believe that this is because Laplacian score does not use supervision information, i.e. labels (the constraints as a result).

In particular, CSFS outperforms constraint and Laplacian score significantly, while it outperforms or achieves similar accuracy to Fisher score in all cases.

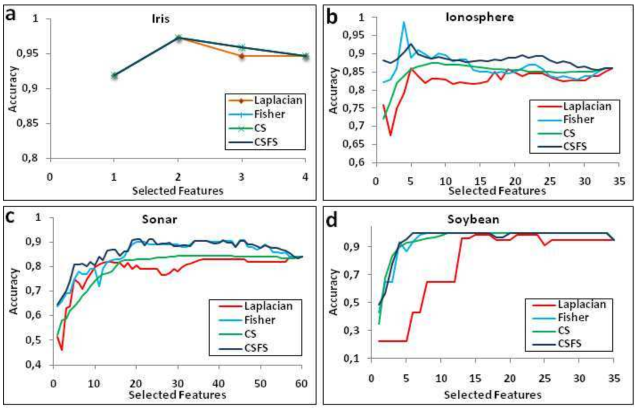

Figure 6. Accuracy vs different numbers of selected features

Note that Fisher uses the full labels of the data set while CSFS uses a subset of coherent constraints generated originally from a small-labeled data party (25%). It is remarkable too that CSFS scores good accuracy even with few number of selected features, these results verify that merging "useful" constraints extracted from supervision information with geometrical structure of unlabeled data is very useful in learning feature scores.

Then, we compare the performance of CSFS with that of PCA, cFLD and SSDR-CMU (Fig. 7). This comparison concerns the Accuracy vs. different number of constraints (we used 50% of selected features). Note that authors in [24] proposed the SDDR score with different variants (SSDR-M, SSDR-CM and SSDR-CMU), we compared our results with SSDR-CMU because it use the two types of pairwise constraints in addition to the unlabeled data, which means that it uses the same specifications that we consider in our score function. In addition, SSDR-CMU recorded better results than the other SSDR variants. The comparison of our framework with the listed scores is presented under different levels of selected constraints.

Note also that CSFS deploys just the coherent constraints from the whole constraints set generated from the labeled data. This can explain that the maximum number of selected constraints in the figure is far less than the maximum number of possible constraints.

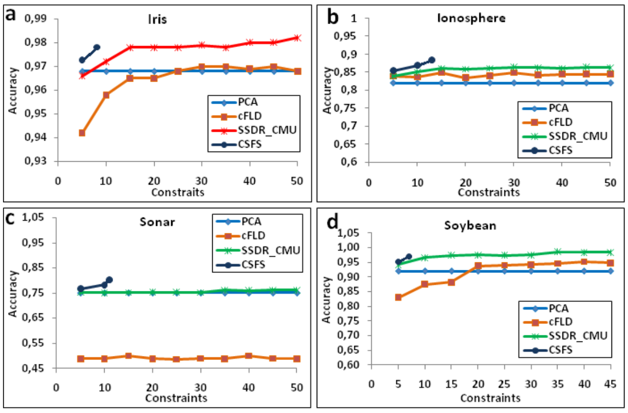

Figure 7. Accuracy vs. different numbers of selected constraints ("coherent" constraints for CSFS)

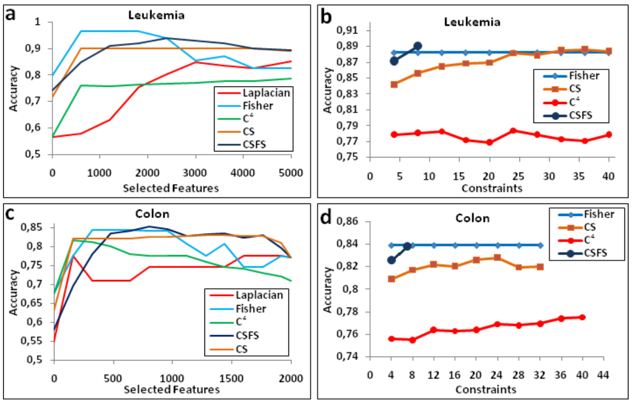

Figure 8. (a,c) Accuracy vs. different numbers of selected features, (b,d) Accuracy vs. different numbers of selected constraints

(Fig. 7) shows that CSFS outperforms the PCA and cFLD scores significantly, and it is comparable to SSDR-CMU on Soybean, outperforms it in Sonar and Ionosphere, but inferior to it on Iris when SSDR-CMU exploits the full constraints set. Note that CSFS achieves a high accuracy even when few "coherent" constraints are deployed. Another important notice from (Fig. 7) is that CSFS accuracy on Sonar and Ionosphere data sets is higher of the other scores accuracy even when they deploy the full constraints set, this validates the practically proven fact that the use of more "incoherent" constraints would have ill effects on learning performance (or it would have no effects in best cases).

5.3 Results on Leukemia and Colon Cancer 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 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 both cases: Accuracy vs. Selected features (The coherent constraints used for this case are: 7 for colon cancer and 8 for Leukemia), and Accuracy vs. the selected constraints (50% of the selected features were deployed). The results of accuracy vs. Selected features (fig. 8-a,c) show that CSFS 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 CSFS is superior to other scores performance when increasing the number of features.

While the results of accuracy vs. number of Selected constraints (fig. 8-b,d) show that CSFS outperforms other scores when using the full "coherent" constraint sets, and as on UCI data sets, the accuracy achieved by CSFS on Leukemia data set is not reached by other scores even when using the whole possible constraints set.

5.4 Results on face images data sets

As mentioned above, Pie10P & Pix10P are face images data sets containing 10 persons in each. The validation on these data sets is presented in comparison with Laplacian, Relief 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 [27].

Note that the coherent constraints used are (6 for Pix10P and 9 for Pie10P), Experimentation results in (Fig. 9) shows that CSFS outperforms significantly the other scores whatever the exploited number of features. Meanwhile, on Pie10P data set, CSFS is higher than Laplacian and (F3 + r) scores and inferior to ReliefF. Nevertheless, it could be shown that CSFS has an excellent accuracy on Pix10P data set and very good one on Pie10P data set. In addition, (Fig. 9) illustrates the stability of CSFS in comparison with other scores.

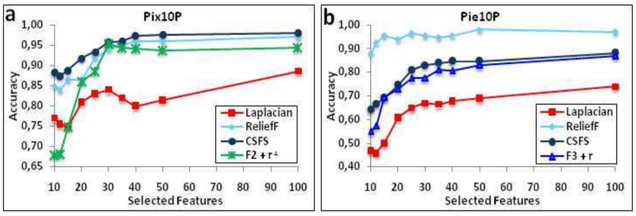

Figure 9. Accuracy vs. different numbers of selected features

Finally, Table (2) shows that regarding the average accuracy, CSFS has an excellent average accuracy, which is superior to all other scores except to Relief score on Pie10P data set.

Table 2. Averaged accuracy of different algorithms on "Pie10P" & "Pix10P" data sets

Data set	Laplacian	ReliefF	$F2+r^4$	F3+r	CSFS
Pie10P	0.74	0.97	0.78	0.87	0.91
Pix10P	0.88	0.97	0.94	0.93	0.98

CONCLUSION

In this paper, we proposed a framework for feature selection based on constraint selection for semi-supervised dimensionality reduction. A new score function was developed to evaluate the relevance of features based on both, the locally geometrical structure of unlabeled data and the constraints preserving ability of labeled data. The framework which we propose has three major advantages:

- It incorporates the labeled and unlabeled examples in a competent and flexible manner, so it could be utilized regardless of the percentage of the labeled data.
- It exploits a pairwise constraint selection, which results in a coherent constraint subset extracted from the labeled data.
- It surveys the structural neighborhood of data examples, which highlights the efficient locality preserving properties of the selected features.

Future work may include the amelioration of the choice of labels set from which constraints are generated. Other perspectives may be the choice of clustering algorithm between the constraints. In addition, we used a “hard” constraints selection which means to select only the constraints that are coherent with the full constraints set. This results in a little constraints number. Possible choice may be to adopt a “soft” constraints selection, in which the constraints coherence is calculated gradually, and the constraint is rejected if it is incoherent with the so far selected constraints, this may results in a higher constraints number, then it would be interested to judge if the learning quality would be more efficient with a great number of constraints softly selected than with a few number of constraints hardly selected.

6. REFERENCES

[1] Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. et Levine, A. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A*. 96, 12 (1999), 6745-6750.

[2] Basu, S., Davidson, I. et Wagstaff, K. 2008. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman and Hall/CRC.

[3] Bishop, C.M. 1996. *Neural Networks for Pattern Recognition*. Oxford University Press, USA.

[4] Chapelle, O., Scholkopf, B. et Zien, A. 2010. *Semi-supervised learning*. MIT.

[5] Chung, F.R. 1996. *Spectral Graph Theory*. American Mathematical Society.

[6] Dash, M. et Liu, H. 1997. Feature Selection for Classification. *Intelligent Data Analysis*. 1, (1997), 131--156.

[7] Davidson, I., Wagstaff, K.L. et Basu, S. 2006. Measuring constraint-set utility for partitional clustering algorithms. *in: Proceedings of the Tenth European Conference on Principles and Practice of Knowledge Discovery in Databases*. 4213, (2006), 115--126.

[8] Duda, R.O., Hart, P.E. et Stork, D.G. 2000. *Pattern Classification*. Wiley-Interscience.

[9] Dy, J.G., Brodley, C.E. et Wrobel, S. 2004. Feature selection for unsupervised learning. *Journal of Machine Learning Research*. 5, (2004), 845--889.

[10] Fisher, R. 1936. The use of multiple measurements in taxonomic problems. *Annals Eugen.* 7, (1936), 179-188.

[11] Frank, A. et Asuncion, A. 2010. {UCI} Machine Learning Repository.

[12] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. et Lander, E.S. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science (New York, N.Y.)*. 286, 5439 (Oct. 1999), 531-537.

[13] Guyon, I. et Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*. 3, (2003), 1157-1182.

[14] He, X., Cai, D. et Niyogi, P. 2005. Laplacian score for feature selection. *In NIPS* (2005).

[15] Jain, A. et Zongker, D. 1997. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19, (1997), 153--158.

[16] Jolliffe, I. 2002. *Principal Component Analysis*. Springer.

[17] Kalakech, M., Biela, P., Macaire, L. et Hamad, D. 2011. Constraint scores for semi-supervised feature selection: A comparative study. *Pattern Recognition Letters*. 32, 5 (2011), 656 - 665.

[18] Kohonen, T. éd. 1997. *Self-organizing maps*. Springer-Verlag New York, Inc.

[19] Mali, K. and Mitra, S. 2003. Clustering and its validation in a symbolic framework. *Pattern Recognition Letters*. 24, 14 (Oct. 2003), 2367-2376.

[20] Robnik-Šikonja, M. et Kononenko, I. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning*. 53, (Oct. 2003), 23–69.

[21] Xing, E., Ng, A., Jordan, M. et Russell, S. 2002. Distance Metric Learning, with Application to Clustering with Side-information. *Advances in Neural Information Processing Systems 15* (2002), 505-512.

[22] Yu, L. et Liu, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. *in ICML* (2003), 856-863.

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

[24] Zhang, D., Zhou, Z. et Chen, S. 2007. Semi-supervised dimensionality reduction. *In: Proceedings of the 7th SIAM International Conference on Data Mining* (2007), 11–393.

[25] Zhao, J., Lu, K. et He, X. 2008. Locality sensitive semi-supervised feature selection. *Neurocomputing*. 71, (Juin. 2008), 1842–1849.

[26] Zhao, Z. et Liu, H. 2007. Semi-supervised Feature Selection via Spectral Analysis. *- Proceedings of the 7th SIAM International Conference on Data Mining* (2007).

[27] Zhao, Z. et Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th international conference on Machine learning* (New York, NY, USA, 2007), 1151–1157.