Constraint scores for semi-supervised feature selection: A comparative study

Mariam Kallakech, Philippe Biela, Ludovic Macaire, Denis Hamad

Constraint scores for semi-supervised feature selection: A comparative study

Mariam Kalakech[a,b], Philippe Biela[a,b], Ludovic Macaire[b], Denis Hamad[c]

[a] *Hautes Etudes d'Ingénieur, 13 rue de Toul, F-59046, Lille, France*
[b] *LAGIS FRE CNRS 3303, Université de Lille 1 Nord de France, Bâtiment P2, Cité Scientifique, F-59655 Villeneuve d'Ascq, France*
[c] *LISIC, ULCO, 50, rue Ferdinand Buisson, F-62228 Calais, France*

Abstract

Recent feature selection scores using pairwise constraints (must-link and cannot-link) have shown better performances than the unsupervised methods and comparable to the supervised ones. However, these scores use only the pairwise constraints and ignore the available information brought by the unlabeled data. Moreover, these constraint scores strongly depend on the given must-link and cannot-link subsets built by the user. In this paper, we address these problems and propose a new semi-supervised constraint score that uses both pairwise constraints and local properties of the unlabeled data. Experimental results show that this new score is less sensitive to the given constraints than the previous scores while providing similar performances.

Keywords: Feature selection, Pairwise constraints, Kendall's coefficient, Constraint scores, Laplacian score, Fisher score

1. Introduction

In machine learning and pattern recognition applications, the processing of high dimensional data requires large computation time and capacity storage. Though, it leads to poor performances when the dimensionality to sample size ratio is high. To improve performances, the sample dimensionality is reduced thanks to feature extraction or selection schemes (Liu and Motoda (1998); Yu and Liu (2003)). Let us notice that feature extraction transforms the original input space into a new low dimensional space by combining the initial features, while feature selection retains the most relevant ones in order to build a low dimensional feature space.

Data samples can be either unlabeled or labeled, leading to the development of unsupervised and supervised feature selection techniques. Unsupervised feature selection measures the feature capacity of keeping the intrinsic data structure in order to evaluate its relevance (Dy and Brodley (2004)). Supervised feature selection consists in evaluating feature relevance by measuring the correlation between the feature and class labels (Yu and Liu (2004)).

Supervised feature selection requires sufficient labeled data samples in order

to provide a discriminating feature space. However, the sample labeling process by the human user is fastidious and expensive. That is the reason why in many real applications, we have huge unlabeled data and small labeled samples. To deal with this "lack labeled-sample problem", recent semi-supervised feature selection schemes have been developed by Zhao et al. (Zhao and Liu (2007a); Zhao and Liu (2007b)). They propose a semi-supervised feature relevance criterion which takes into account both unlabeled data and labeled samples. Unfortunately, this score requires to define the classifier step in order to compare the initial labels of the samples and those provided by the classifier (Ng et al. (2001)).

Beside class labels, there is another kind of user supervision information called the pairwise constraints (Bar-Hillel et al. (2005)). It consists to simply specify whether a pair of data samples must be regrouped together (must-link constraints) or cannot be regrouped together (cannot-link constraints). Zhang et al. (Zhang et al. (2008)) propose to evaluate the feature relevance by scores which only take into account these constraints. Zhao et al. define another score which uses both the pairwise constraints defined by the user and the unlabeled nearest neighbors of the samples (Zhao et al. (2008)). However, this score considers the neighbors of each sample without explicitly

taking into account its local density property.

These authors have experimentally shown that the features selected thanks to these constraint-based scores, may provide results which are comparable with those given by supervised approaches. Unfortunately, these scores strongly depend on the given constraint subsets built by the user. So, when the user slightly modifies the constraint subset, the feature scores may also change.

In this paper, we first present a review of the feature selection scores. We then propose a semi-supervised score which uses both pairwise constraints and the local properties of the unlabeled data. We experimentally demonstrate that this score, thanks to the contribution of the unlabeled data, is less sensitive to constraint changes than the classical constraint scores, while providing satisfying classification results.

Previous works compare the performances of the feature scores by considering the accuracy rates obtained by a classifier operating in the selected feature space. In order to measure the sensitiveness of the scores to constraint changes, we could estimate the dispersion of accuracy rates with respect to different subsets of constraints. Though, this evaluation depends on the be-

havior of the used classifier. So, we propose to only examine the dispersion of the feature ranks provided by the examined scores thanks to the Kendall's coefficient (Grzegorzewski (2006)).

The paper is organized as follows. In section 2, we review different supervised and unsupervised feature selection scores. Then, we introduce the spectral semi-supervised score in section 3. Recent constraint scores are detailed in section 4 and our semi-supervised constraint score is presented in section 5. In section 6, we study the relationships between the constraints given by the user and the feature ranks obtained by the different scores. Comparative experimental results are provided in section 7 in order to assess the efficiency of our semi-supervised constraint score.

2. Supervised and unsupervised feature selection

Given a dataset of n samples defined in a d-dimensional feature space, let us denote $\mathcal{X} = (x_{ir})$ $i = 1, \ldots, n$; $r = 1, \ldots, d$; the associated data matrix where x_{ir} is the r^{th} feature value of the i^{th} data. Each of the n rows of the matrix $\mathcal{X}$ represents a data sample $x_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$, while each of the d columns of $\mathcal{X}$ defines the feature values $f_r = (x_{1r}, \ldots, x_{nr})^T \in \mathbb{R}^n$.

2.1. Supervised feature selection

The principle of supervised feature selection consists in examining the correlation between projected data samples and their class labels on each feature axis. It looks for features on which the classes are compact and far from each others. For this purpose, one uses the well known Fisher criterion to evaluate the feature relevance (Bishop (1996)).

By considering the sample coordinates on the feature f_r, each class ω, ω=1, ..., c, populated with n_ω labeled samples is characterized by its mean $\mu_{\omega r}$ and its variance $\sigma^2_{\omega r}$. Moreover, let us denote μ_r the mean of all data samples on the feature f_r.

The Fisher score F_r used to evaluate the relevance of the feature f_r is defined by:

$$F_r = \frac{\sum_{\omega=1}^{c} n_\omega (\mu_{\omega r} - \mu_r)^2}{\sum_{\omega=1}^{c} n_\omega \sigma^2_{\omega r}}. \quad (1)$$

In order to select the most relevant features, they are sorted according to the decreasing order of their Fisher score F_r.

2.2. Unsupervised feature selection

Unsupervised feature selection consists in evaluating the relevance of each feature by examining the dispersion of the data samples projected on its axis. A feature is considered as being relevant when the data samples projected on this feature axis are scattered as much as possible.

So, the variance score V_r is used to evaluate the relevance of the feature f_r:

$$V_r = \frac{1}{n} \sum_{i=1}^{n} (x_{ir} - \mu_r)^2 . \tag{2}$$

The features are sorted according to the decreasing order of V_r, in order to select the most relevant ones.

Rather than measuring the data dispersion along a feature axis, one examines the local properties of the data. The basic idea is to assume that the input data pairwise distances are preserved in the relevant feature space. So, similar samples have to be close when they are projected on a relevant feature axis.

According to the spectral graph theory (von Luxburg. (2007)), data samples x_i, $i = 1, \ldots, n$ are represented by n nodes in a graph structure. The edge between two connected nodes i and j is weighted by a similarity level defined

by:

$$s_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2t^2}\right), \tag{3}$$

where $\|x_i - x_j\|^2$ represents the squared euclidean distance between x_i and x_j in the d-dimensional initial feature space (von Luxburg. (2007)). The parameter t has to be tuned in order to represent the local dispersion of the data (Zelink-Manor and Perona (2005)).

The weights of the graph are represented by a similarity matrix S $(n \times n)$. From S, we calculate the Laplacian matrix defined as $L = D - S$, where D $(n \times n)$ is the diagonal matrix, $D_{ii} = \sum_{j=1}^{n} s_{ij}$. It is interesting to note that the degree D_{ii} of a node i can be considered as a local density measure at x_i.

In an unsupervised context, He et al. assume that the projections (coordinates) of similar data on the examined feature axis have to be as close as possible (He et al. (2005)). They propose to compute the Laplacian score L_r of a feature f_r as:

$$L_r = \frac{\sum_{ij}(x_{ir} - x_{jr})^2 s_{ij}}{\sum_i \left(x_{ir} - \overline{f}_r\right) D_{ii}}. \tag{4}$$

It is easy to demonstrate that:

$$L_r = \frac{\widetilde{f}_r^T L \widetilde{f}_r}{\widetilde{f}_r^T D \widetilde{f}_r} \tag{5}$$

where $\widetilde{f}_r = f_r - \overline{f}_r$ and $\overline{f}_r = \frac{\sum_{i=1}^{n} x_{ir} D_{ii}}{\sum_{i=1}^{n} D_{ii}}$.

By considering D_{ii} as a density probability measure, $\overline{f}_r$ is the weighted feature average.

In order to select relevant features, they are sorted according to the ascending order of L_r.

He et al. have experimentally demonstrated that the classifier operating in the feature space selected by the Laplacian score L_r, outperforms the classifier operating in the feature space selected by the variance score V_r. Indeed, L_r takes into account the locality structure of the data samples.

3. Semi-supervised feature selection

In many applications, we have huge unlabeled and a few labeled data samples. Indeed, labeling all the data samples by the user is time consuming and fastidious. In that context, the labeled data subset is usually too small to carry sufficient information for the supervised selection while unsupervised

approaches ignore this label information which could be yet interesting for feature selection.

That is the reason why more interest has been addressed for a new challenge in feature selection called "small labeled-sample problem". Semi-supervised feature selection methods bring slutions by considering both labeled and unlabeled data subsets.

Zhao et al. propose to couple the Laplacian score with normalized mutual information (NMI) in order to introduce a new semi-supervised feature selection score (Zhao and Liu (2007a); Zhao and Liu (2007c)). This score compares the initial labels of the data samples with the labels provided by a classifier operating with the examined feature f_r. It is defined as:

$$M_r = \alpha \frac{\sum_{ij}(x_{ir} - x_{jr})^2 s_{ij}}{\sum_i \left(x_{ir} - \overline{f}_r\right) D_{ii}} + (1-\alpha)\left(1 - NMI\left(\widehat{f}_r, y\right)\right). \qquad (6)$$

where $\widehat{f}_r$ is the cluster indicator of the samples generated by the classifier and y is the initial class label vector (Zhao and Liu (2007a)). The first term of equation (6) calculates the Laplacian score of the feature f_r, while the second term estimates the corresponding classification error of $\widehat{f}_r$ according to the labeled data. The term α is a regularization parameter set to 0.1 by (Zhao and Liu (2007a)), in order to favor the contribution of the labeled data. That

is the reason why the selected features by using equation (6) mainly depend on the labeling decision achieved by the classifier.

4. Constraint scores

The prior knowledge about the data can be represented according to two different ways: class labels and pairwise constraints. Class labels require to have detailed information about the classes and to precisely indicate the label of each data sample. Pairwise constraints simply mention for some pairs of data samples that they are similar, i.e. must be regrouped together (must-link constraints), or that they are dissimilar, i.e. cannot be regrouped together (cannot-link constraints).

The user has to build the subset $\mathcal{M}$ of must-link constraints and the subset $\mathcal{C}$ of cannot-link constraints defined as:

$\mathcal{M} = \{(x_i,x_j)$, *such as* x_i *and* x_j *must be linked*$\}$.

$\mathcal{C} = \{(x_i,x_j)$, *such as* x_i *and* x_j *cannot be linked*$\}$.

The cardinals of these subsets are usually much lower than the number $\binom{n}{2}$ of all possible pairwise constraints.

These pairwise constraints are easier to be obtained by the user than the class labels. They simply formalize that two data samples belong or not to

the same class without detailed information about the classes in presence. Indeed, labeled data samples can be transformed into must-link and cannot-link constraints but not vice versa. It consists in putting must-link constraint between two data samples which share the same label and cannot-link constraint between two data samples sharing different labels.

Zhang et al. have recently proposed a constraint feature selection scheme which uses only a subset of must-link and cannot-link constraints (Zhang et al. (2008)).

In the context of the spectral theory (von Luxburg. (2007)), two specific graphs are built:

- The must-link graph $G^{\mathcal{M}}$ where a connection is established between two nodes i and j if there is a must-link constraint between their corresponding samples (nodes) x_i and x_j.

- The cannot-link graph $G^{\mathcal{C}}$ where a connection is established between two nodes i and j if there is a cannot-link constraint between their corresponding samples (nodes) x_i and x_j.

The connection weights between two nodes of the graphs $G^{\mathcal{M}}$ and $G^{\mathcal{C}}$ are respectively stored by the similarity matrices $S^{\mathcal{M}}$ $(n \times n)$ and $S^{\mathcal{C}}$ $(n \times n)$,

and are given by:

$s_{ij}^{\mathcal{M}}=$

$$\begin{cases} 1 \text{ if } (x_i,x_j) \in \mathcal{M} \text{ or } (x_j,x_i) \in \mathcal{M} \\ 0 \text{ otherwise,} \end{cases} \tag{7}$$

$s_{ij}^{\mathcal{C}}=$

$$\begin{cases} 1 \text{ if } (x_i,x_j) \in \mathcal{C} \text{ or } (x_j,x_i) \in \mathcal{C} \\ 0 \text{ otherwise.} \end{cases} \tag{8}$$

The matrices $S^{\mathcal{M}}$ and $S^{\mathcal{C}}$ are used to define the constraint Laplacian matrices $L^{\mathcal{M}} = D^{\mathcal{M}} - S^{\mathcal{M}}$ and $L^{\mathcal{C}} = D^{\mathcal{C}} - S^{\mathcal{C}}$, where $D^{\mathcal{M}}$ and $D^{\mathcal{C}}$ are the degree matrices defined by $D_{ii}^{\mathcal{M}} = \sum_{j=1}^{n} s_{ij}^{\mathcal{M}}$ and $D_{ii}^{\mathcal{C}} = \sum_{j=1}^{n} s_{ij}^{\mathcal{C}}$.

In order to measure the constraint preserving ability of the feature f_r, Zhang et al. define two constraint scores C_r^1 and C_r^2:

$$C_r^1 = \frac{\sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{M}}}{\sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{C}}} = \frac{f_r^T L^{\mathcal{M}} f_r}{f_r^T L^{\mathcal{C}} fr}, \tag{9}$$

$$
\begin{aligned}
C_r^2 &= \sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{M}} - \lambda \sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{C}} \\
&= f_r^T L^{\mathcal{M}} f_r - \lambda f_r^T L^{\mathcal{C}} f_r,
\end{aligned}
\tag{10}
$$

where λ is a regularization parameter used to balance the contribution of the two constraints terms of C_r^2. Must-link constraints are favored by setting $0 < \lambda < 1$.

The lower these two scores are, the more relevant the feature is. Zhang et al. have experimentally shown that the features selected by C^1 and C^2 provide similar performances when λ is well balanced.

5. Constraint scores for semi-supervised feature selection

The scores presented in section 4 use only the available constraints and do not take into account the unlabeled data contribution. Zhao et al. define another score which uses both unlabeled data and pairwise constraints in order to retrieve both locality properties and discriminating structures in the data samples (Zhao et al. (2008)). They build a new graph $G^{\mathcal{W}}$ which connects samples having high probability of sharing the same label:

- $G^{\mathcal{W}}$ is the within-class graph: two nodes i and j are connected if (x_i,x_j) or (x_j,x_i) belongs to $\mathcal{M}$, or if the two samples are unlabeled but they

are sufficiently close to each other (by using the k-nearest neighbor graph)

The edges in the graphs $G^{\mathcal{W}}$ are weighted by using the similarity matrix $S^{\mathcal{W}}$ $(n \times n)$ and are expressed as:

$s_{ij}^{\mathcal{W}}=$

$$\begin{cases} \gamma \text{ if } (x_i,x_j) \in \mathcal{M} \text{ or } (x_j,x_i) \in \mathcal{M} \\ 1 \text{ if } x_i \text{ or } x_j \text{ is unlabeled} \\ \text{but node } i \in KNN(j) \text{ or node } j \in KNN(i) \\ 0 \text{ otherwise} \end{cases} \tag{11}$$

where γ is a constant parameter which has been empirically set to 100 in (Zhao et al. (2008)).

Zhao et al. introduce a Laplacian score, called the locality sensitive discriminant analysis score and defined as:

$$C_r^3 = \frac{\sum_{i,j} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{W}}}{\sum_{i,j} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{C}}} = \frac{f_r^T L^{\mathcal{W}} f_r}{f_r^T L^{\mathcal{C}} f_r}, \tag{12}$$

where $L^{\mathcal{W}} = D^{\mathcal{W}} - S^{\mathcal{W}}$, $D^{\mathcal{W}}$ being the degree matrix defined by $D^{\mathcal{W}}_{ii} = \sum_{j=1}^{n} s^{\mathcal{W}}_{ij}$. This score implicitly takes into account the unlabeled data but favors pairs of must-link data by assigning them high weights in the matrix $S^{\mathcal{W}}$. Moreover, the similarity matrix $S^{\mathcal{W}}$ represents the links between the k-nearest neighbors of the data by binary weighting them. By mainly considering the must-link constraints, C^3 is very close to C^2, and both neglect the unlabeled data samples.

Though, taking into account the unlabeled data samples should catch the data structure and make less sensitive a feature score against the given constraint subset. That leads us to propose another semi-supervised constraint score which is less sensitive to the constraints chosen by the user. Given the matrices S, $S^{\mathcal{M}}$ and $S^{\mathcal{C}}$, the semi-supervised constraint score is defined as :

$$C^4_r = \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} . \frac{f_r^T L^{\mathcal{M}} f_r}{f_r^T L^{\mathcal{C}} f_r}. \tag{13}$$

The proposed score C^4_r is the simple product between the Laplacien score L_r (see equation (4)) processed with the unlabeled data and the constraint score C^1_r (see equation (9)) defined by Zhang:

$$C^4_r = L_r . C^1_r, \tag{14}$$

So, it takes into account both the unlabeled data thanks to L_r and the available constraints thanks to C_r^1 in order to evaluate the relevance of the feature f_r. As for the other scores, the features are ranked in ascending order according to score C^4 in order to select the most relevant ones.

6. Constraint subset influence in feature selection process

The constraint scores introduced above evaluate the relevance of the features based on the must-link and cannot-link constraints. However, these scores strongly depend on the given constraint subsets $\mathcal{M}$ and $\mathcal{C}$. Indeed, changing the subset of available constraints could lead to a large change in the feature ranks and so, in the selected features. To illustrate this problem, we examine the following toy example.

6.1. Toy example

Let us consider four 3-dimensional data samples:

$A(-3,-1,1)$; $B(-3,1,1)$; $C(-1-1,1)$ and $D(1,-3,-1)$.

A, B and C belong to the first class whereas D is assigned to the second class. We can see that the feature f_3 is the single feature which corresponds to the class label (1 for the first class, -1 for the second class). So, an efficient

feature selection algorithm should identify the feature f_3 as the most relevant one.

Let us consider that the user builds one single must-link constraint and one single cannot-link constraint from the labeled samples. The constraint scores introduced in sections 4 and 5 are used to rank the 3 features based on these constraints.

The different feature ranks are shown in table 1. Pairs $\{(A,B)\}$, $\{(A,C)\}$ and $\{(B,C)\}$ are the single must-link constraints that can be built from the data samples. Pairs $\{(A,D)\}$, $\{(B,D)\}$ and $\{(C,D)\}$ are the single cannot-link constraints that can be also built from the data samples. So, there are 9 constraint combinations which correspond to 9 cells in table 1. The first, the second, the third and the fourth row of each cell represent the ranks of the 3 features given by C^1, C^2 (α is fixed to 0.1 as suggested by the authors), C^3 and C^4 scores, respectively. The sign '=' between two features means that their scores are equal.

For example, let us examine the cell which corresponds to the must-link constraint $\{(A,C)\}$ and the cannot-link constraint $\{(B,D)\}$. Scores C^1 and C^4 select the features f_2 and f_3 as the best features. and the feature f_1 as the third feature. Scores C^2 and C^3 sort feature f_2 as the best feature, feature

f_3 as the second feature and feature f_1 as the third feature.

$\mathcal{C}/\mathcal{M}$		$\{(A,B)\}$	$\{(A,C)\}$	$\{(B,C)\}$
$\{(A,D)\}$	C^1	$\mathbf{f_3 = f_1}$, $\mathbf{f_2}$	$\mathbf{f_3 = f_2}$, $\mathbf{f_1}$	$\mathbf{f_3}$, $\mathbf{f_1}$, $\mathbf{f_2}$
	C^2	$\mathbf{f_1}$, $\mathbf{f_3}$, $\mathbf{f_2}$	$f_3 = f_2$, f_1	f_3, f_1, f_2
	C^3	f_1, f_3, f_2	$\mathbf{f_3}$, $\mathbf{f_2}$, $\mathbf{f_1}$	f_3, f_1, f_2
	C^4	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	f_3, f_1, f_2
$\{(B,D)\}$	C^1	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	$\mathbf{f_3}$, $\mathbf{f_1 = f_2}$
	C^2	f_1, f_3, f_2	$\mathbf{f_2}$, $\mathbf{f_3}$, $\mathbf{f_1}$	f_3, $f_1 = f_2$
	C^3	f_1, f_3, f_2	f_2, f_3, f_1	f_3, f_2, f_1
	C^4	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	f_3, f_1, f_2
$\{(C,D)\}$	C^1	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	f_3, $f_1 = f_2$
	C^2	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	f_3, $f_1 = f_2$
	C^3	f_3, f_1, f_2	f_3, f_2, f_1	f_3, f_2, f_1
	C^4	$f_3 = f_1$, f_2	$f_3 = f_2$, f_1	f_3, f_1, f_2

Table 1: Feature ranks by semi-supervised scores using the chosen constraints.

First, table 1 shows that the feature f_3 is always ranked as the first feature by C^1 and C^4 whatever the given constraint subset. C^2 and C^3 do not retain feature f_3 as the first feature three times. This table also shows

that 7 different ranks (indicated in bold in table 1) of the 3 features are obtained by comparing each of the 4 examined scores for the 9 constraint combinations. This simple example clearly shows that the features ranks deduced from these scores strongly depend on the chosen constraints. So, we propose to measure how the feature ranks vary with respect to the subset of constraints.

6.2. Kendall's coefficient

The performances of the examined scores are generally compared by measuring the accuracy rates obtained from well classified data projected on the selected features. To measure the dependence of the scores on the given constraint subsets, we could estimate the disparity of the accuracy rates obtained with different constraint subsets. This evaluation requires to define a labeling decision step, such as k-nearest neighbor classifier, which operates in the selected feature space. Since this decision step influences the quality of classification, the comparison between the score performances may depend on the used classifier.

We prefer to examine only the feature ranks deduced from the scores, so that our study is not corrupted by the decision step. More precisely, we study the

concordance between feature ranks when the constraint subset changes.

Given the sample set $\mathcal{X} = (x_1,...,x_n)$ and the associated class label of each data sample, we randomly pick up a subset $\mathcal{S}_q$ of pairwise constraints ($\mathcal{S}_q = \mathcal{M}_q \cup \mathcal{C}_q$). Then, we rank the d features f_r, according to the different constraint scores C_r^* detailed in sections 4 and 5 ($* = 1, 2, 3, 4$).

Let us denote R_{qr}^* the rank of the feature f_r when the score C_r^* considers the constraint subset $\mathcal{S}_q$. In order to evaluate the influence of constraint subset, the feature selection is run p times. For a score C^*, the ranks of the d features obtained with the p different subsets $\mathcal{S}_q$ are represented by the matrix R^* ($p \times d$):

$$R^* = \begin{bmatrix} R_{11}^* \; R_{12}^* \; ... \; R_{1d}^* \\ R_{21}^* \; R_{22}^* \; ... \; R_{2d}^* \\ \; ... \quad ... \quad ... \quad ... \quad ... \\ R_{p1}^* \; R_{p2}^* \; ... \; R_{pd}^* \end{bmatrix}$$

The q^{th} row of R^* represents the d feature ranks by using the subset S_q ($q = 1,...,p$) while the r^{th} column represents the ranks of the feature f_r by using the p different constraint subsets. Therefore, each row of R^* is a per-

mutation of the d feature ranks which depends on the available constraint subset S_q.

We use the Kendall's coefficient to measure the concordance or agreement between the feature ranks with p constraint subsets (Grzegorzewski (2006)). The Kendall's coefficient K^* takes into account the different rows of the matrix R^* and is defined as (Siegel and Castellan (1988)):

$$K^* = \frac{12\Delta^*}{p^2\left(d^3 - d\right) - p.\tau^*}, \tag{15}$$

where $\Delta^* = \sum_{r=1}^{d}\left(R_r^* - \bar{R}^*\right)^2$, $R_r^* = \sum_{q=1}^{p} R_{qr}^*$, $\bar{R}^* = \frac{1}{d}\sum_{r=1}^{d} R_r^*$ and $\tau^* = \sum_{v=1}^{m}\left(\tau_v^{*3} - \tau_v^*\right)$

The term τ_v^* is the number of tied ranks in each of the m groups of ties in R^*. The sum τ^* is computed over all the groups of ties found in all p rows of the table R^*.

The Kendall's coefficient which measures the dispersion Δ^* of the feature ranks, ranges from 0 (no agreement) to 1 (complete agreement).

6.3. Kendall's coefficient for the toy example

Let us examine the toy example of section 6.1. We propose to compute the Kendall's coefficient from table 1 in order to measure the concordance of

the feature ranks obtained by each examined constraint score. The Kendall's coefficients K^1 (see eq. (9)), K^2 (see eq. (10)), K^3 (see eq. (12)) and K^4 (see eq. (13)) are respectively 0.4325, 0.2258, 0.3333 and 0.4333.

These low values reflect the dependence of the feature ranks on the available constraints subsets.

This example shows that the Kendall's coefficient K^4 of C^4 is slightly higher than the other ones. The objective of this simple example, with low sample population, is not to compare the Kendall's coefficient of the different scores, but to arise that the selected features depend on the given constraint subset. This conclusion concurs with that of Sun et al. (Suna and Zhang (2010)).

7. Comparative experimental results

In this section, we first measure the sensitivity of the scores against the given constraints. So, we compare the Kendall's coefficient obtained by the tested scores (C^1, C^2, C^3, C^4).

We also compare the performances obtained by a classifier operating in the feature space selected by C^4 score and those obtained with the classical feature selection methods: Fisher score (supervised), Laplacian score (unsupervised), C^1, C^2 (constraint scores) and C^3 (semi-supervised constraint

score).

7.1. Examined databases

For this purpose, our experiments are achieved with six well known and largely used benchmark databases, and more precisely the 'Wine', 'Image segmentation' and 'Vehicle' databases from the UCI repository (Blake et al. (1998)), the face database 'ORL' (Samaria and Hartert (1994)) and the two gene expression databases, i.e., 'Colon Cancer'(Alon et al. (1999)) and 'Leukemia'(Golub et al. (1999)). These databases have been retained since the features are numeric and since the label information of each data sample is clearly defined.

In our experiments, we first normalize the features between 0 and 1, so that different features have the same scale. For each dataset, we follow an Holdout partition and choose the first half of samples from each class as the training data and the remaining data for testing.

7.2. Results on UCI datasets and ORL database

First, we achieve experiments on 3 UCI datasets and the the 'ORL' face database. Here is a brief description of the four considered databases:

- **'Wine' database**

  This database contains 178 13-dimensional data (d=13) regrouped into 3 classes having 59, 71 and 48 instances, respectively. We randomly select 30, 36 and 24 data samples from each class to build the training data. The remaining data are organized as the test data subset.

- **'Image segmentation' database**

  This database contains 210 19-dimensional data (d=19) regrouped into 7 classes, each class having 30 instances. We randomly select 15 data samples from each class to build the training data and the remaining data constitute the test data subset.

- **'Vehicle' database**

  This database contains 846 18-dimensional data (d=18) regrouped into 4 classes, having 212, 217, 218 and 199 instances, respectively. We randomly select 106, 109, 109 and 100 data samples from each class to build the training data.

- **'ORL' database**

  The 'ORL' database (Olivetti Research Laboratory) contains a set of face images representing 40 distinct subjects. There are 10 different

Figure 1: Sample face images from the ORL database (2 subjects).

images per subject, so that the database contains 400 images. For each subject, the images have been acquired according to different conditions: lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses) (see figure 1).

In our experiments, original images have been normalized (in scale and orientation) so that the two eyes are aligned at the same horizontal position. Then, the facial areas have been cropped in order to build images of size 32×32 pixels. Thus, each image can be represented by a 1024-dimensional sample data. The gray level of each pixel is quantified with 256 levels.

We randomly select 5 images from each class (subject) to build the training data. The remaining data are organized as the test data subset.

7.2.1. Kendall's coefficient results

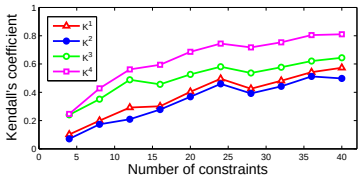
In our experiments, the feature selection is performed on the training data and features are ranked according to the different scores. At each feature selection run q, q=1,..., p, we simulate the generation of pairwise constraints as follow: we randomly select pairs of samples from the training data and create must-link or cannot-link constraints depending on whether the underlying classes of the two samples are the same or different. We iterate this scheme until we obtain $\frac{card(S_q)}{2}$ must-link constraints and $\frac{card(S_q)}{2}$ cannot-link constraints.

This operation is repeated over $p = 100$ runs in order to measure the Kendall's coefficients K^1, K^2, K^3 and K^4.

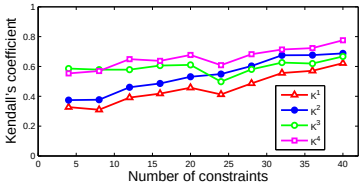
We achieve experiments for different cardinals of S_q ranging from 4 (2 must-link and 2 cannot-link) to 40 constraints as Zhang et al. do.

Figure 2 shows the Kendall's coefficients K^1, K^2, K^3 and K^4 calculated over $p = 100$ runs for different numbers of constraints. At each run, the same given constraint subset is of course considered by the four tested criteria. The low values of K^1, K^2, K^3 and K^4 show clearly that the selected features using C^1, C^2, C^3 and C^4 depend on the constraint subsets.

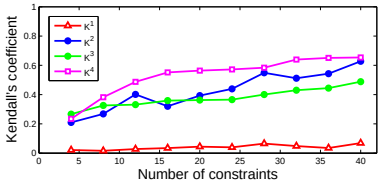
By examining the different curves of figure 2, we see that the different coeffi-



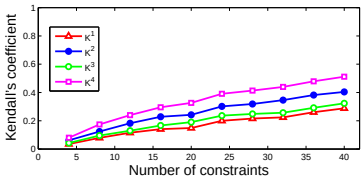
(a) 'Wine'.



(b) 'Image segmentation'.



(c) 'Vehicle'.



(d) 'ORL'.

Figure 2: Kendall's coefficient for different number S_q of constraints on the four databases.

cients K^1, K^2, K^3 and K^4 increase with the number $card(S_q)$ of constraints. The higher the number $card(\mathcal{S}_q)$ of constraints is, the more complete the supervision information is. Ideed, when $card(\mathcal{S}_q)$ is high, the semi-supervised learning context tends to become a supervised learning context. That explains why the agreement of feature ranks increases when $card(\mathcal{S}_q)$ increases in the different curves of figure 2.

We can also notice that K^4 has the highest values for the different cardinals of constraint subsets in the four databases. This shows that our score C^4 is less sensitive to the must-link and cannot-link constraints built by the user

than the classical scores.

We also observe that the values of the Kendall's coefficients on the 'ORL' database are lower than those obtained with the other databases. Indeed, the number of features extracted from the 'ORL' database (1024 features) is bigger than that of the other databases.

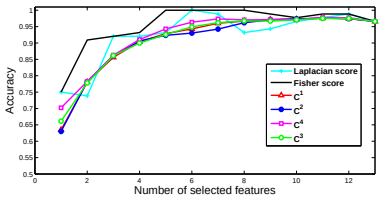
Zhao and Zhang have experimentally shown that by picking-up a small number $card(\mathcal{S}_q)$ of constraints, their scores allow to select the features which carry a good discriminating power. Figure 2 shows that, between two different runs, the feature ranks change, and so the selected features change when $card(\mathcal{S}_q)$ is low. So, the quality of data discrimination would strongly vary with respect to the chosen constraint subsets.

7.2.2. Comparison of the performances

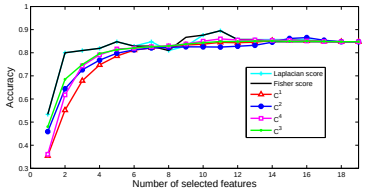
We also compare the performances obtained by the nearest neighbor classifier which operates in the feature space selected thanks to the considered supervised, unsupervised and semi-supervised feature scores.

The classification accuracies of the test data are used to evaluate the performance of each criterion.

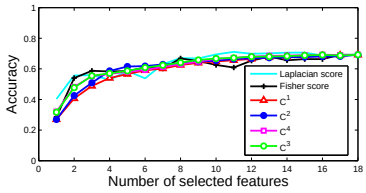
As for measuring the Kendall's coefficients, the rates of good classification



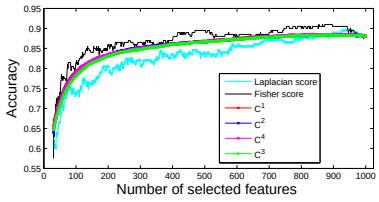
(a) 'Wine'.



(b) 'Image segmentation'.



(c) 'Vehicle'.



(d) 'ORL'.

Figure 3: Accuracy vs. different number of selected features on the four datasets. 10 pairwise constraints including 5 must-link and 5 cannot link are used.

are averaged over $p = 100$ runs with different generations of constraints.

Figure 3 shows the plots of accuracy vs. the desired number of selected features on the 'Wine' 3(a), 'Image segmentation' 3(b), 'Vehicle' 3(c) and 'ORL' 3(d) databases, respectively when $card(\mathcal{S}_q)$ is set to 10 .

From figure 3 , we can see that the classification rates obtained with constraint scores (C^1, C^2, C^3 and C^4) range between those obtained with the unsupervised method (Laplacian score) and those obtained with the supervised method (Fisher score). Since $card(\mathcal{S}_q)$ is set to 10, these semi-supervised scores use a bit more supervision information than Laplacian score, but much fewer than Fisher Score.

So, these results confirm that a classification scheme reaches higher performance thanks to a supervised learning than thanks to a semi-supervised learning. They also confirm that taking into account a few constraints allows to improve classification results, compared with the unsupervised learning.

However, the curves of C^1, C^2, C^3 and C^4 are confused in figure 3. It is also difficult to compare the performances of these scores since they are averaged over 100 runs with different generations of constraints.

That leads us to compare these scores while examining their accuracies at each of the 100 runs. For a fixed number of selected features, in each of

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	198	238	219	**180**
10 constraints	195	210	185	**162**
40 constraints	180	299	168	**136**

Table 2: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'wine' database.

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	208	228	**168**	179
10 constraints	228	210	**154**	183
40 constraints	184	177	153	**146**

Table 3: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'image' database.

the 100 runs, we propose to rank the 4 criteria in descending order of their accuracy. Let us denote $rank_q^*$ the rank of the criterion C^* at the run q. This rank takes the values 1, 2, 3 or 4. At each run q, the method having the highest accuracy is ranked as 1 and the method with the lowest accuracy value, is ranked as 4. Methods with the same accuracy have the same rank. We calculate a rank sum T^* for each semi-supervised constraint score as follow:

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	237	239	232	**219**
10 constraints	271	246	**189**	240
40 constraints	290	240	208	**207**

Table 4: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'vehicle' database.

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	**194**	212	311	200
10 constraints	**185**	189	250	196
40 constraints	**190**	202	300	203

Table 5: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'ORL' database.

$$T^* = \sum_{q=1}^{100} rank_q^*, \tag{16}$$

where * is 1, 2, 3 or 4 corresponding to the score C^1, C^2, C^3 or C^4 respectively. The method with the lowest rank sum is considered as being the score which provides the best results.

For the 'Wine', 'Image segmentation', 'Vehicle' and 'ORL' databases, the accuracy of each of the 4 semi-supervised criteria seems to be stable when

the number of desired features is higher than 6, 5, 8 and 300, respectively (see figure 3). So, we propose to calculate the rank total of each of the semi-supervised criteria by considering the 6 first selected features for the 'Wine database', the 5 first selected features for the 'Image segmentation' database, the 8 first selected features for the 'Vehicle' database and the 300 first selected ones for the 'ORL' database .

Tables 2, 3, 4 and 5 show the rank sum T^* for different numbers S_q of constraints (4, 10 and 40). Each cell indicates the rank sum of the tested score with the considered number of constraints. From these tables, we can clearly see that T^1, T^2, T^3 and T^4 are very close. The features selected thanks to C^4 provide accuracy rates comparable with those obtained by the features selected by C^1, C^2 and C^3. Indeed, our score provides the lowest rank sum T (indicated in bold) for 6 times over the 12 rows of tables 2, 3, 4 and 5.

7.3. Results on gene expression databases

In this subsection, several experiments are carried out on two gene expression databases, i.e., 'Colon Cancer'(Alon et al. (1999)) and 'Leukemia'(Golub et al. (1999)).

- **'Colon Cancer'**

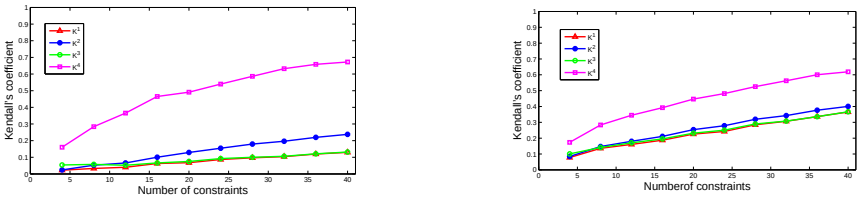
  This database contains the expression of 2000 genes measured on 62 tissues (40 tumors and 22 normals).We randomly select 20 and 11 data samples from each class to build the training data. The remaining data are organized as the test data subset.

- **'Leukemia'**

  This database contains information on gene-expression in samples from human acute myeloid (AML) and acute lymphoblastic leukemias (ALL). From the originally measured 6817 probe sets we removed genes that were not present in at least one sample so a total of 5147 genes are used in the experiments.

  Because Leukemia has a predefined partition of the objects into training (27 ALL and 11 AML) and testing (20 ALL and 14 AML) subsets, the ensembles on this dataset are performed on the predefined training and testing sets.

Figure 4 shows the Kendall's coefficients K^1, K^2, K^3 and K^4 calculated over $p = 100$ runs for different numbers of constraints on the 'Colon Cancer' and the 'Leukemia' databases. As for the previous databases, K^4 has the high-

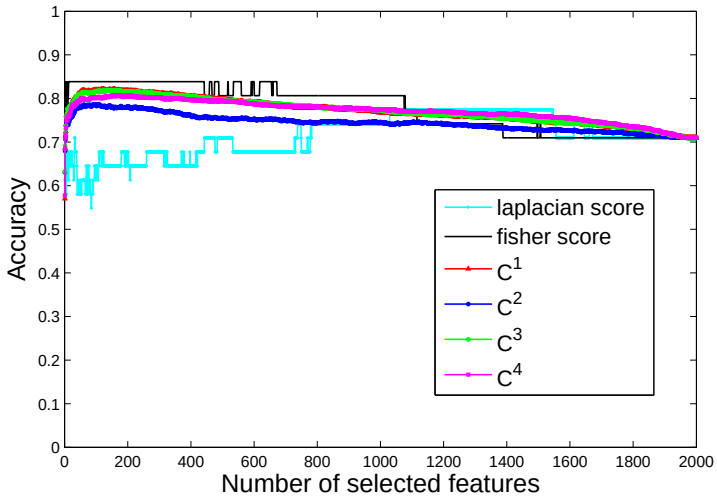


(a) 'Colon Cancer'. (b) 'Leukemia'.

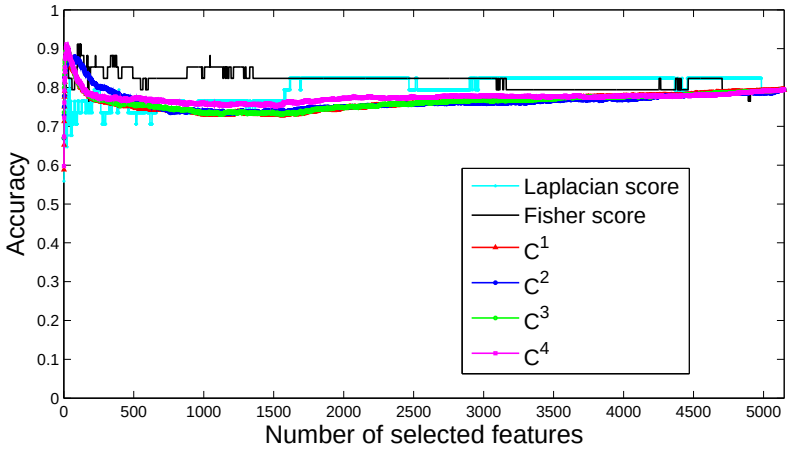
Figure 4: Kendall's coefficient for different number S_q of constraints on the 'Colon Cancer' and the 'Leukemia' databases.

est values for the different cardinals of constraint subsets in these two gene expression databases. This confirm the fact that our score C^4 is less sensitive to the must-link and cannot-link constraints built by the user than the classical scores. Moreover, since the number of features is very high, 2000 for the Colon Cancer database and 5147 for the leukemia one, we can notice that the gap between K^4 on a hand and K^1, K^2 and K^3 on the other hand is higher on these databases than the previous ones (a voir).

Figure 5 shows the plots for accuracy vs. different numbers of selected features on Colon Cancer and Leukemia databases. $card(S_q)$ here is set to 60 as Sun et al. do (Suna and Zhang (2010)), so 60 pairwise constraints including 30 must-link and 30 cannot-link are used. From figure 5, we can see that the classification rates obtained with constraint scores (C1, C2,



(a) 'Colon Cancer'.



(b) 'Leukemia'.

Figure 5: Accuracy vs. different number of selected features on the gene expression-databases. 60 pairwise constraints including 30 must-link and 30 cannot link are used.

C3 and C4) range between those obtained with the unsupervised method (Laplacian score) and those obtained with the supervised method (Fisher score). These semi-supervised constraint scores even outperforms the supervised Fisher score on the 'Colon Cancer' database for a number of selected features more than 1050.

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	**148**	195	235	158
10 constraints	156	242	203	**153**
40 constraints	169	286	153	**138**
60 constraints	157	271	149	**144**

Table 6: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'Colon Cancer' database.

Tables 6 and 7 show the rank sum T^* for different numbers S_q of constraints (4, 10, 40 and 60) on the 'Colon Cancer' and the 'Leukemia' databases respectively. The rank total of each of the semi-supervised criteria is calculated considering the half of the original features of each of the gene expression databases (see(Suna and Zhang (2010))).

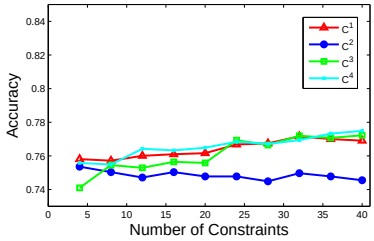
From these tables, we can see that, for the 'Colon Cancer' database, our score provides the lowest rank sum T (indicated in bold) for 3 times over the

$card(S_q)$ / T	T^1	T^2	T^3	T^4
4 constraints	184	240	194	**182**
10 constraints	176	276	150	**134**
40 constraints	170	196	148	**118**
60 constraints	186	198	168	**126**

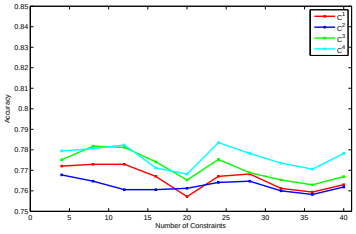
Table 7: The rank sums of the different semi-supervised criteria for different number S_q of constraints on the 'Leukemia' database.

4 rows of table 6. For the 'Leukemia database', our score provides the lowest rank sum T (indicated in bold) for the different numbers of constraints (4, 10, 40 and 60). These results improve the fact that the features selected thanks to C_4 provide accuracy rates comparable with those obtained by the features selected by C_1, C_2 and C_3.

Furthermore, figure 6 shows the plot for accuracy under fixed number of selected features (half of the number of original features) vs. different numbers of pairwise constraints on the gene expression databases. For almost all the number of constraints, our score C_4 achieves higher accuracy than C_1, C_2 and C_3.



(a) 'Colon Cancer'.



(b) 'Leukemia'.

Figure 6: Accuracy vs. different number of pairwise constraints (for C_1, C_2, C_3 and C_4) on the gene expression databases. The desired number of selected features is half of the number of original features

.

8. Conclusion

Constraint scores which use pairwise constraints for semi-supervised feature selection have shown good performances of classification. Unfortunately, these scores depend on the subset of constraints built by the user since they do not take into account the information provided by the unlabeled data. In this paper, we propose a new semi-supervised constraint score that considers both the pairwise constraints and the local properties of the unlabeled data. Moreover, we study the relationships between the features selected by the constraint scores and the constraints chosen by the user. We measure the sensitiveness of the scores to constraint changes by the Kendall's coefficient.

To the best of our knowledge, this is the first work that studies the influence of the constraint subsets change on the features selected by the constraint scores.

Experimental results on three UCI datasets and a face database show that the proposed score is less sensitive to the constraint changes while selecting features that provide satisfying classification results.

References

Alon, U., Barkai, N., Notterman, D., Gishdagger, K., Ybarradagger, S., Mackdagger, D., Levine, A., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Science of the USA 96 (12), 745–6750.

Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D., 2005. Learning a mahalanobis metric from equivalence constraints. Journal of Machine Learning Research, 937–965.

Bishop, C., Jan. 1996. Neural Networks for Pattern Recognition. Oxford University Press, USA.

570 Blake, C., Keogh, E., Merz, C., 1998. UCI repository of machine learning
571 databases. http://www.ics.uci.edu/ mlearn/MLRepository.html.

572 Dy, J., Brodley, C., 2004. Feature selection for unsupervised learning. Journal
573 of Machine Learning Research 5, 845–889.

574 Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M.,
575 Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A.,
576 Bloomfield, C. D., 1999. Molecular classification of cancer: class discovery
577 and class prediction by gene expression monitoring. Science 286, 531–537.

578 Grzegorzewski, P., 2006. The coefficient of concordance for vague data. Com-
579 putational Statistics and Data Analysis 51 (1), 314–322.

580 He, X., Cai, D., Niyogi, P., Dec. 2005. Laplacian score for feature selection.
581 In: Proceedings of the Advances in Neural Information Processing Systems
582 ('NIPS 05'). Vancouver, British Columbia, Canada.

583 Liu, H., Motoda, H., Aug. 1998. Feature extraction construction and selection
584 a data mining perspective, 1st Edition. Springer.

585 Ng, A., Jordan, M., Weiss, Y., Dec. 2001. On spectral clustering: Analysis

and an algorithm. In: Proceedings of the Advances in Neural Information Processing Systems ('NIPS 01'). Canada, pp. 849–856.

Samaria, F., Hartert, A., 1994. Parameterisation of a stochastic model for human face identification. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision 'ACV 94'. Sarasota, Florida, pp. 138–142.

Siegel, S., Castellan, N., 1988. Nonparametric statistics for the behavioral sciences, 2nd Edition. McGraw-Hill, New York.

Suna, D., Zhang, D., Jun. 2010. Bagging constraint score for feature selection with pairwise constraints. Pattern Recognition 43, 2106–2118.

von Luxburg., U., 2007. A tutorial on spectral clustering. Statistics and Computing 17 (4), 395–416.

Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. Vol. 6. pp. 856–863.

Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224.

Zelink-Manor, L., Perona, P., Dec. 2005. Self-tuning spectral clustering. In:

Proceedings of the Advances in Neural Information Processing Systems 'ANIPS 05'. Vancouver, British Columbia, Canada, pp. 1601–1608.

Zhang, D., Chen, S., Zhou, Z., Oct. 2008. Constraint score: A new filter method for feature selection with pairwise constraints. Pattern Recognition (41), 1440–1451.

Zhao, J., Lu, K., He, X., 2008. Locality sensitive semi-supervised feature selection. Neurocomputing 71 (10-12), 1842–1849.

Zhao, Z., Liu, H., Apr. 2007a. Semi-supervised feature selection via spectral analysis. In: Proceedings of the SIAM International Conference on Data Mining 'ICDM 07'. Minneapolis.

Zhao, Z., Liu, H., 2007b. Semi-supervised feature selection via spectral analysis. Tech. rep., Computer Science and Engineering (CSE) Department, Ari-zona State University (ASU).

Zhao, Z., Liu, H., Aug. 2007c. Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on Machine learning 'ICML 07'. ACM, Corvalis, Oregon, pp. 1151–1157.