Similarity-based constraint score for feature selection

Abderezak Salmi, Kamal Hammouche, Ludovic Macaire

Similarity-Based Constraint Score For Feature Selection

Abderezak Salmi[a], Kamal Hammouche[a,*], Ludovic Macaire[b]

[a] *Université Mouloud Mammeri, Laboratoire Vision Artificielle et Automatique des Systèmes (LVAAS), Tizi-Ouzou, Algeria*
[b] *Université Lille, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France*

Abstract

To avoid the curse of dimensionality resulting from a large number of features, the most relevant features should be selected. Several scores involving must-link and cannot-link constraints have been proposed to estimate the relevance of features. However, these constraint scores evaluate features one by one and ignore any correlations between features. In addition, they compute distances in the high-dimensional original feature space to evaluate the similarity between samples. So, they would be corrupted by the curse of dimensionality. To deal with these drawbacks, we propose a new constraint score based on a similarity matrix that is computed in the selected feature subspace and that makes it possible to evaluate the relevance of a feature subset at once. Experiments on benchmark databases demonstrate the improvement brought by the proposed constraint score in the context of both supervised and semi-supervised learnings.

Keywords: Constraint score, Feature selection, Pairwise constraints, Similarity matrix.

1. Introduction

In machine learning and pattern recognition applications, such as data mining and image analysis, datasets are often characterized by a large number of

*Corresponding author
Email addresses: salmi-abderezak@hotmail.fr (Abderezak Salmi), kamal_hammouche@yahoo.fr (Kamal Hammouche), ludovic.macaire@univ-lille.fr (Ludovic Macaire)

features. The processing of such high-dimensional data requires large memory storage and high computational time, and may lead to poor learning performance [1, 2]. To address these drawbacks, the dimensionality of data is often reduced by using feature selection to remove redundant or irrelevant features. Typically, feature selection methods can be categorized into three types: filter, wrapper, and embedded methods [2, 3]. Filter methods evaluate features independently of the classification algorithm, while wrapper methods exploit a classification algorithm to evaluate the relevance of features. Embedded methods embed feature selection into the learning algorithm. Because filter methods are not dependent on any classification scheme, they have better generalization ability than wrapper and embedded methods [2, 3].

According to the availability of prototypes (i.e., labeled data samples that represent classes), feature selection methods can also be divided into unsupervised, supervised, and semi-supervised approaches [1, 2, 3]. Supervised feature selection only uses prototypes to measure the correlation of each feature with the class labels, while unsupervised feature selection uses only unlabeled data samples to evaluate the feature capacity to preserve the intrinsic data structure [1]. Semi-supervised feature selection takes into account both prototypes and unlabeled data samples to evaluate the relevance of features.

In supervised and semi-supervised learning frameworks, besides class labels of prototypes, the available information can be also expressed by must-link and cannot-link constraints. A must-link constraint specifies that two data samples belong to the same class, while a cannot-link constraint specifies that two data samples belong to different classes [4]. Pairwise constraints can be provided by the user or easily generated from a small number of prototypes.

Must-link and cannot-link constraints are used to estimate the relevance of features via score functions, called constraint scores [1, 2]. Zhang et al. [5] proposed two supervised constraint scores that use only pairwise constraints to evaluate the relevance of features. Zhao et al. [6] defined a semi-supervised constraint score that analyzes both pairwise constraints and unlabeled data samples for feature selection. Kalakech et al. [1] combined an unsupervised score com-

puted from unlabeled data samples with a supervised score that is computed from the pairwise constraints. This score is predicted to be less sensitive to constraint changes. Two semi-supervised constraint scores that assess the ability of a feature to preserve the local properties of unlabeled data samples while respecting pairwise constraints, have been proposed by Benabdeslem et al. in [7] for the former and in [8] for the latter. More recently, Yang et al. introduced a new semi-supervised constraint score which takes advantage of the local geometrical structure of unlabeled data samples as well as constraints deduced from prototypes [9, 10].

Because the above-mentioned constraint scores are part of the filter approach, they all evaluate features one by one [2]. The score of a feature subset is estimated by the sum of the individual feature scores, and the evaluation of a feature subspace ignores correlations between features. Thus, learning algorithms that operate in a subspace of individually relevant features do not necessarily provide favorable results [8]. In addition, the constraint scores proposed in the literature are based on the Laplacian of a similarity matrix. Because the similarity matrix is computed in the original feature space, state-of-the-art feature scores can also be corrupted by the curse of dimensionality.

In this paper, we propose a new constraint score that evaluates the relevance of features in the context of both supervised and semi-supervised learning. Our score assesses the ability of features to respect the available set of pairwise constraints. As this score can be used for feature selection, it is based on a similarity matrix that is computed in the considered feature subspace. Unlike existing constraint scores that are applied to each feature, our score can evaluate a subset of several features simultaneously. The proposed score is then used as a criterion by a sequential forward selection scheme to identify the most relevant subset of features with tractable computation [11].

The performance of the constraint scores is measured by the classification accuracy of the test data commonly obtained by the nearest neighbor classifier. Previous studies use the entire training dataset with true class labels as prototypes by the classifier, while only few prototypes are used by the constraint

scores. By using conditions that are similar to those in real-life applications, in this paper we propose using only available information. In the supervised context, only the prototypes involved in pairwise constraint generation are used by the classifier. In the semi-supervised context, we follow the same strategy proposed by Kalakech et al. [12] that first uses the constrained K-means algorithm [4] to classify the unlabeled training data samples and then uses the classified samples as prototypes in classifying the test data. However, in our case, instead of using the constrained K-means algorithm, we use constrained spectral clustering, which is based on the same similarity matrix concept used by the constraint scores.

The remainder of this paper is organized as follows. Section 2 provides a brief definitions on spectral graph theory and pairwise constraints generation. In Section 3 a primary state-of-the-art constraint scores is presented. Our proposed constraint score and the feature selection procedure are presented in Section 4. Experimental results achieved with benchmark databases pertaining to supervised and semi-supervised feature selection are provided and discussed in Section 5.

2. Preliminaries

Constraint scores are based on the concepts of spectral graph theory and pairwise constraints. In this section, we briefly give some notations and definitions related to these two concepts.

2.1. Spectral graph theory

Let $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times d}$ denote the set of n training data samples defined in a d-dimensional feature space, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{ir}, \ldots, x_{id}] \in \mathbb{R}^d$ is the i-th data sample of X, and x_{ir},$(r = 1, \ldots, d)$ is the r-th feature value of the i-th data sample. Let $F_d = \{f_1, f_2, \ldots, f_r, \ldots, f_d\}$ denote the set of d feature vectors of X, where $f_r = [x_{1r}, x_{2r}, \ldots, x_{nr}]^T \in \mathbb{R}^n$ is the r-th feature vector.

In spectral graph theory, dataset X is represented by a complete undirected weighted similarity graph $G = (V, E, W)$ in which each data sample in X corresponds to a node. V is a non-empty set that contains all nodes, and E is the set of edges between any two nodes in V. Each edge in E is weighted by a similarity value w_{ij} $(i, j = 1, 2, \ldots, n)$ between two nodes. The similarity matrix W that gathers similarities between all pairs of nodes is positive semi-definite and symmetric. Generally, the similarity w_{ij} between two data samples x_i and x_j is computed by the following Gaussian kernel function [13]:

$$w_{ij} = \texttt{exp}\left(-\frac{\delta^2(x_i, x_j)}{2\sigma^2}\right) \quad i, j = 1, 2, \ldots, n \tag{1}$$

where $\delta(x_i, x_j)$ is the Euclidean distance between two data samples x_i and x_j, and σ is a scaling parameter.

Dataset X can also be represented by the similarity matrix W^{KNN} of the nearest neighbor subgraph G^{KNN} whose node v_i is connected to v_j when x_j is one of the K-nearest neighbors ($KNNs$) of x_i such that:

$$w_{ij}^{KNN} = \begin{cases} w_{ij} & \text{if } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The constraint scores are often formulated from a given unnormalized Laplacian matrix of W, that is defined as follows:

$$L = D - W \tag{3}$$

where $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix whose elements are $d_{ii} = \sum_{j=1}^{n} w_{ij}$ [13, 14].

2.2. Pairwise constraint

In supervised and semi-supervised learning frameworks, often a limited amount of information on the training dataset is available. This prior knowledge is expressed by only few labeled data samples. It can also be expressed by pairwise constraints that mention if data samples belong to the same class (must-link)

or to different classes (cannot-link). Pairwise constraints can be given by the user or are generated from labeled data samples.

In this paper, we consider that only few labeled data samples (prototypes) characterize the k classes ω^l, $l = 1, \ldots, k$. Let $X^l \in \mathbb{R}^{p \times d}$, $(X^l \subset X)$ be the set of p prototypes that are associated with class ω^l. From the overall set of prototypes denoted X^P $\left(X^P = \bigcup_{l=1,\ldots,k} X^l\right)$, we can build set M of $(k \cdot p^2 - k \cdot p = k \cdot p \cdot (p-1))$ must-link pairs that are composed of two prototypes belonging to the same class:

$$M = \left\{(x_i, x_j) \in X^2 \mid \exists\ l = 1, \ldots, k \text{ so that } x_i \in X^l \text{ and } x_j \in X^l\right\} \quad (4)$$

We can also build set C of $(k \cdot (k-1) \cdot p^2)$ cannot-link pairs that are composed of two prototypes belonging to different classes:

$$C = \left\{(x_i, x_j) \in X^2 \mid \exists\ (l, m);\ l \neq m;\ \text{so that } x_i \in X^l \text{ and } x_j \in X^m\right\} \quad (5)$$

Of all possible data pairs that can be extracted from X, those belonging to M or C are called constrained pairs, while the remaining pairs are called unconstrained pairs. Furthermore, data that are not prototypes are called unlabeled data samples and are gathered in subset $X^U = X / X^P$.

In the context of spectral graph theory, two graphs G^M and G^C are built from the sets of must-link constraints M and cannot-link constraints C. The similarity matrices $W^M \in \mathbb{R}^{n \times n}$ and $W^C \in \mathbb{R}^{n \times n}$ are defined as follows:

$$w_{ij}^M = \begin{cases} 1 & \text{if } (x_i, x_j) \in M \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$w_{ij}^C = \begin{cases} 1 & \text{if } (x_i, x_j) \in C \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

3. Constraint scores

The performance achieved by learning algorithms such as classification or clustering depends on similarity, which is based on the Euclidean distance in the original d-dimensional feature space. Because these features are not always relevant, many authors select the best ones based on constraint scores that combine the concepts of spectral graph theory and pairwise constraints.

3.1. Constraint scores for supervised feature selection

In a supervised learning framework, only pairwise constraints are used to evaluate the relevance of features while unconstrained data samples are ignored.

Two constraint scores C_r^1 and C_r^2 of feature f_r are defined by Zhang et al. [5] as follows:

$$C_r^1 = \frac{\sum_{(x_i,x_j)\in M}(x_{ir}-x_{jr})^2}{\sum_{(x_i,x_j)\in C}(x_{ir}-x_{jr})^2} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij}^M}{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij}^C} \tag{8}$$

$$\begin{aligned} C_r^2 &= \sum_{(x_i,x_j)\in M}(x_{ir}-x_{jr})^2 - \lambda\sum_{(x_i,x_j)\in C}(x_{ir}-x_{jr})^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij}^M - \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2 w_{ij}^C \end{aligned} \tag{9}$$

where $\lambda \in [0,1]$ is a regularization parameter. In our experiments, λ is set to 1, as in [5]. These scores assume that the distance between the must-link data samples should be as low as possible, while the distance between the cannot-link data samples should be as high as possible.

The two scores can be formulated from the unnormalized constrained Laplacian matrices $L^M = D^M - W^M$ and $L^C = D^C - W^C$:

$$C_r^1 = \frac{f_r^T L^M f_r}{f_r^T L^C f_r} \tag{10}$$

$$C_r^2 = f_r^T L^M f_r - \lambda f_r^T L^C f_r \tag{11}$$

Lower scores indicate a more relevant feature.

3.2. Constraint scores for semi-supervised feature selection

Semi-supervised feature selection involves the analysis of both pairwise constraints and unlabeled data samples. It considers both the discriminating power of the pairwise constraints and the local properties of the unlabeled data samples.

Zhao et al. [6] introduced the semi-supervised constraint score C_r^3, called the locality sensitive discriminant analysis score, which combines the similarity

matrix W^C constructed from the cannot-link constraints (Eq. (7)) and the similarity matrix $W^{KNN1} \in \mathbb{R}^{n \times n}$ which is constructed from the set of must-link constraints and unlabeled data samples as follows:

$$w_{ij}^{KNN1} = \begin{cases} \gamma & \text{if } (x_i, x_j) \in M \\ 1 & \text{if } (x_i \in X^U \text{ or } x_j \in X^U) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where γ is a constant parameter. In our experiments, γ is set to 100 and K is set to 5 as in [6].

The constraint score C_r^3 is defined as

$$C_r^3 = \frac{\sum_{i=1}^n \sum_{j=1}^n (x_{ir} - x_{jr})^2 w_{ij}^{KNN1}}{\sum_{i=1}^n \sum_{j=1}^n (x_{ir} - x_{jr})^2 w_{ij}^C} = \frac{f_r^T L^{KNN1} f_r}{f_r^T L^C f_r} \tag{13}$$

where $L^{KNN1} = D^{KNN1} - W^{KNN1}$ is the unnormalized Laplacian matrix of W^{KNN1}, D^{KNN1} being the degree matrix computed from W^{KNN1}.

This score ensures that two data samples related by a cannot-link constraint are well separated. It also implicitly takes into account the unlabeled data samples but favors pairs of must-link data samples by assigning them high weights in the matrix W^{KNN1}. Moreover, the similarity matrix W^{KNN1} represents the links between the $KNNs$ of the unlabeled data samples by binary weighting them.

Kalakech et al. [1] proposed a semi-supervised constraint score C_r^4 that is less sensitive to the constraint sets by a simple combination of scores computed on prototypes and unlabeled data samples. More precisely, C_r^4 attempts to identify a trade-off between the unsupervised Laplacian score L_r and the supervised constraint score C_r^1 (see Eq. (8)) by multiplying both scores :

$$C_r^4 = L_r \cdot C_r^1 \tag{14}$$

The Laplacian score L_r of a feature f_r is defined as in [15] as follows:

$$L_r = \frac{\sum_{i=1}^n \sum_{j=1}^n (x_{ir} - x_{jr})^2 w_{ij}}{\sum_{i=1}^n (x_{ir} - \bar{f}_r)^2 d_{ii}} = \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} \tag{15}$$

where $\bar{f}_r = \frac{\sum_{i=1}^{n} x_{ir} d_{ii}}{\sum_{i=1}^{n} d_{ii}} = \frac{f_r^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}}$, $\tilde{f}_r = f_r - \bar{f}_r$ and $\mathbf{1} = [1, \ldots, 1]^T$.

L_r favors features with high variance and tends to select those with a strong ability to preserve locality, while C_r^1 seeks to select features with a strong ability to preserve pairwise constraints.

Finally, C_r^4 is defined in terms of Laplacian matrices as follows:

$$C_r^4 = \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} \cdot \frac{f_r^T L^M f_r}{f_r^T L^C f_r} \tag{16}$$

L and D are deduced from similarity matrix W (Eq. (1)).

Benabdeslem and Hindawi proposed another constraint score called the constrained Laplacian score that combines the similarity matrices W^C and W^{KNN2} for semi-supervised feature selection [7]. The similarity matrix W^{KNN2} is expressed as

$$w_{ij}^{KNN2} = \begin{cases} w_{ij} & \text{if } \big((x_i, x_j) \in M\big) \text{ or } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

The score C_r^5 is defined as follows [7, 16]:

$$C_r^5 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (x_{ir} - x_{jr})^2 w_{ij}^{KNN2}}{\sum_{i=1}^{n} \sum_{j=1 | \exists l, (x_l, x_j) \in C}^{n} (x_{ir} - \alpha_{jr}^i)^2 d_{ii}^{KNN2}} \tag{18}$$

where

$$\alpha_{jr}^i = \begin{cases} x_{ir} & \text{if } (x_i, x_j) \in C \\ \bar{f}_r & \text{if } \big(i = j\big) \text{ and } \big(x_i \in X^U\big) \\ x_{ir} & \text{otherwise} \end{cases} \tag{19}$$

C_r^5 represents an enhanced version of both scores: the Laplacian score L_r and the supervised constraint score C_r^1. In fact, the Laplacian score L_r can be seen as a special version of C_r^5 when there are no labeled data samples ($\alpha_{jr}^i = \bar{f}_r$), and when ($\alpha_{jr}^i = x_{ir}$), C_r^5 can be considered as an adjusted version of the constraint score C_r^1. Subsequently, C_r^5 is defined in terms of Laplacian matrices as follows [7]:

$$C_r^5 = \frac{f_r^T L^{KNN2} f_r}{f_r^T L^C D^{KNN2} f_r} \tag{20}$$

Finally, on the one hand, this score ensures that data samples that are neighbors or related by must-link constraints should be close together when they are projected to relevant features. On the other hand, the distance between data samples that are related by cannot-link constraints should be as high as possible.

A second semi-supervised constrained Laplacian score, referred to C_r^6, has been proposed by Benabdeslem and Hindawi in [8] as follows:

$$C_r^6 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2(w_{ij}^{KNN}+\mathcal{N}_{ij})}{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-\alpha_{jr}^{i})^2 d_{ii}^{KNN}} = \frac{f_r^T L^{KNN3} f_r}{f_r^T L^C D^{KNN} f_r} \tag{21}$$

where the diagonal matrix D^{KNN} is deduced from the similarity matrix W^{KNN} (see Eq. 2). N_{ij} is given as follows:

$$\mathcal{N}_{ij} = \begin{cases} -w_{ij} & \text{if } \big((x_i,x_j)\in M\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ w_{ij}^2 & \text{if } \big[\big((x_i,x_j)\in M\big) \text{ and } \big(x_i \notin KNN(x_j) \text{ and } x_j \notin KNN(x_i)\big)\big] \\ & \text{or } \big[\big((x_i,x_j)\in C\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big)\big] \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

The Laplacian matrix L^{KNN3} is computed from the similarity matrix $W^{KNN3} = w_{ij}^{KNN} + \mathcal{N}_{ij}$ defined as follows:

$$w_{ij}^{KNN3} = \begin{cases} w_{ij}^2 + w_{ij} & \text{if } \big((x_i,x_j)\in C\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ w_{ij}^2 & \text{if } \big((x_i,x_j)\in M\big) \text{ and } \big(x_i \notin KNN(x_j) \text{ and } x_j \notin KNN(x_i)\big) \\ w_{ij} & \text{if } \big(x_i \in X^U \text{ or } x_j \in X^U\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

C_r^6 combines the power of the local geometric structure offered by unlabeled data samples, with the constraint preserving ability offered by prototypes. The additional weight w_{ij}^2 was introduced in order to more differentiate the features in the both bad cases, i.e., when two data samples are related by a must-link constraint but are not neighbors and when two neighboring data samples are related by a cannot-link constraint.

Recently, Yang et al. introduced the new semi-supervised constraint score C_r^7, called constraint compensated Laplacian score, which takes advantage of the local geometrical structure of unlabeled data samples as well as constraint information deduced from labeled data samples [9, 10]:

$$C_r^7 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ir}-x_{jr})^2(w_{ij}^{KNN}+\bar{\mathcal{N}}_{ij})}{\sum_r + \sum_r^b - \sum_r^w} \tag{24}$$

$\bar{\mathcal{N}}_{ij}$ is given as follows:

$$\bar{\mathcal{N}}_{ij} = \begin{cases} 1-w_{ij} & \text{if } \big((x_i,x_j)\in M\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ \gamma w_{ij} & \text{if } \big((x_i,x_j)\in C\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ \lambda & \text{if } \big((x_i,x_j)\in M\big) \text{ and } \big(x_i \notin KNN(x_j) \text{ and } x_j \notin KNN(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where γ and λ are the parameters set to the empirical values of 0.9 and 0.5, respectively [9, 10].

$\sum_r$ is the variance of the rth feature of the whole dataset X, $\sum_r^b$ and $\sum_r^w$ are within-class variance and between-class variance of the rth feature of the labeled dataset (protoypes) X^P, respectively.

$$\sum_r = \sum_{i=1}^{n}(x_{ir}-\bar{f}_r)^2 d_{ii} \tag{26}$$

$$\sum_r^b = \sum_{l=1}^{k}|X^l|(\bar{f}_r^{(l)}-\bar{f}_r^P)^2 \tag{27}$$

$$\sum_r^w = \sum_{l=1}^{k}|X^l|(\sigma_r^{(l)})^2 \tag{28}$$

where $|X^l| = p$ is the number of prototypes of the lth class, $\bar{f}_r^P = \sum_{i=1|x_i\in X^P}^{n}\frac{x_{ir}}{k.p}$ is the mean of the rth feature of the labeled dataset, $\bar{f}_r^{(l)} = \sum_{i=1|x_i\in X^l}^{n}\frac{x_{ir}}{p}$ and $\sigma_r^{(l)}$ denote the mean and variance of the rth feature of the lth class, respectively.

The semi-supervised constraint score C_r^7 can be expressed in terms of Laplacian matrices as follows [10]:

$$C_r^7 = \frac{2(f_r)^T L^{KNN4} f_r}{(\tilde{f}_r)^T D^{KNN4}\tilde{f}_r + 2(\tilde{f}_r^P)^T L^P \tilde{f}_r^P - (\tilde{f}_r^P)^T D^P \tilde{f}_r^P} \tag{29}$$

The Laplacian matrix L^{KNN4} is computed from the similarity matrix $W^{KNN4} = W^{KNN} + \tilde{\mathcal{N}}_{ij}$ that is expressed as

$$w_{ij}^{KNN4} = \begin{cases} 1 & \text{if } \big((x_i, x_j) \in M\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ \lambda & \text{if } \big((x_i, x_j) \in M\big) \text{ and } \big(x_i \notin KNN(x_j) \text{ and } x_j \notin KNN(x_i)\big) \\ (1-\gamma)w_{ij} & \text{if } \big((x_i, x_j) \in C\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ w_{ij} & \text{if } \big(x_i \in X^U \text{ or } x_j \in X^U\big) \text{ and } \big(x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i)\big) \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

$L^P = D^P - W^P$ is the unnormalized Laplacian matrix of W^P, D^P being the degree matrix computed from W^P. The cells of W^P are set to $1/|X^l|$ when two data samples are prototypes that belong to the same class and to 0, otherwise:

$$w_{ij}^P = \begin{cases} 1/|X^l| & \text{if } x_i \in X^l \text{ and } x_j \in X^l \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

Note that $\tilde{f}_r^P = f_r^P - \bar{f}_r^P$.

The features providing the lowest scores C^3, C^4, C^5 C^6 and C^7 are the most relevant features. However, because these scores are based on the Laplacian matrices and the diagonal degree matrices, which are deduced from the similarity matrices, they are computed in the original d-dimensional feature space and can be corrupted by the curse of dimensionality.

4. Proposed constrained feature selection

Existing constraint scores estimate the relevance of each feature considered independently and separately from each other. Because these scores do not take into account the correlation between features, we propose a new constraint score that estimates the relevance of a subset of features at once.

4.1. Proposed constraint score

Unlike previous constraint scores that are based on Laplacian matrices, we propose a constraint score that is based only on similarity matrices to

evaluate the relevance of a subset of m features denoted $F_m = \{f_1, \ldots, f_m\}$ $(m = 1, 2, \ldots, d)$. The proposed score, denoted $\varepsilon^*(F_m)$, can be used in one of two learning contexts: $* =$ supervised (S) or semi-supervised (SS). In the supervised learning context, we use only the must-link and cannot-link constraint sets to select features using $\varepsilon^S(F_m)$. In the semi-supervised learning context, we exploit both the pairwise constraint sets and the information contributed by the unlabeled data samples to select features using $\varepsilon^{SS}(F_m)$.

The relevance of F_m is evaluated by means of the distance between the target similarity matrix $\hat{W}^*$, $* =$ S or SS, which is defined from the given constraints and a similarity matrix $W(F_m)$ that is computed with the subset F_m of features. The score $\varepsilon^*(F_m)$, which should be as low as possible, is expressed as the following square error:

$$\varepsilon^*(F_m) = \|W(F_m) - \hat{W}^*\|_2 \quad \text{with} \quad * = \text{S or SS} \tag{32}$$

where $\|\cdot\|_2$ is the Euclidean norm. Thus, $\varepsilon^*(F_m)$ can be rewritten as

$$\varepsilon^*(F_m) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij}(F_m) - \hat{w}^*_{ij}\right)^2 \tag{33}$$

$W(F_m) \in \mathbb{R}^{n\times n}$ is the similarity matrix computed on the dataset X with the subset of features F_m:

$$w_{ij}(F_m) = \texttt{exp}\left(-\frac{\delta^2\left(x_i^{(m)}, x_j^{(m)}\right)}{2\sigma^2}\right) \qquad i, j = 1, 2, \ldots, n \tag{34}$$

where $x_i^{(m)}$ is the vector of the i-th data point characterized by the subset F_m.

$\hat{W}^* \in \mathbb{R}^{n\times n}$ is the target matrix whose cells correspond to must-link pairs are set to 1, while the cells corresponding to the cannot-link pairs are set to 0.

4.1.1. Supervised constraint score

For supervised learning, the target similarity matrix $\hat{W}^S$ is defined as follows:

$$\hat{w}^S_{ij} = \begin{cases} 1 & \text{if } (x_i, x_j) \in M \\ 0 & \text{if } (x_i, x_j) \in C \\ w_{ij}(F_m) & \text{otherwise} \end{cases} \tag{35}$$

The cells of $\hat{W}^S$ corresponding to the unconstrained pairs are set to $w_{ij}(F_m)$ so that they are not taken into account by $\varepsilon^S(F_m)$, $(w_{ij}(F_m) - \hat{w}_{ij}^S = 0)$. Therefore, from Eqs. (33) and (35), it can be easily demonstrated that $\varepsilon^S(F_m)$ can be expressed as

$$\varepsilon^S(F_m) = \sum_{\substack{(x_i, x_j) \in M \\ (x_i, x_j) \in C}} \left(w_{ij}(F_m) - \hat{w}_{ij}^S \right)^2 \tag{36}$$

Finally, $\varepsilon^S(F_m)$ makes it possible to select the set of features with the best constraint preserving ability and without taking into account the unlabeled data samples. The concept behind $\varepsilon^S(F_m)$ is simple and natural. A relevant feature subset produces similarity $w_{ij}(F_m)$ between two must-link samples that is close to 1; however, a good feature subset should provide a similarity between cannot-link samples close to 0.

4.1.2. Semi-supervised constraint score

For semi-supervised learning new must-link pairs are constructed from the prototypes and unlabeled data samples to compute the binary target similarity matrix $\hat{W}^{SS}$:

$$\hat{w}_{ij}^{SS} = \begin{cases} 1 & \text{if } (x_i, x_j) \in M^{SS} \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

The cell (i, j) of $\hat{W}^{SS}$ is set to 1 (0 otherwise) when (x_i, x_j) belongs to M^{SS}, which is a new set of must-link pairs deduced from prototype subsets X^l, $l = 1, \ldots, k$ and unlabeled data samples as follows:

$$M^{SS} = \left\{ (x_i, x_j) \in X^2 | \exists l = 1, \ldots, k \quad \text{so that} \quad NP(x_i) \in X^l \quad \text{and} \quad NP(x_j) \in X^l \right\} \tag{38}$$

The nearest prototype $NP(x_i)$ is the prototype whose distance from a sample data point $x_i \in X$ in the original d-dimensional feature space is the smallest one:

$$NP(x_i) = \underset{y \in \bigcup_{l=1,\ldots,k} X^l}{\arg\min} \left(\delta^2(x_i, y) \right) \tag{39}$$

Thus, pair (x_i, x_j) belongs to M^{SS} when the nearest prototype of x_i and x_j both belong to the same subset X^l of prototypes. Because $NP(x_i)$ is x_i when x_i belongs to prototype subset X^l, set M is included in M^{SS} ($M \subset M^{SS}$). Therefore, M^{SS} increases the contribution of the pairwise constraints M for semi-supervised feature selection. An illustration of M^{SS} is provided in Fig. 1, in which only must-link pairs are represented for clarity. For comparison purposes, Fig. 1(d) displays the links corresponding to similarity matrix W^{KNN1} of Eq. (12), W^{KNN2} of Eq. (17), W^{KNN3} of Eq. (23) and W^{KNN4} of Eq. (30), which is used by the semi-supervised constraint score C^3, C^5, C^6 and C^7, respectively. These links are established from the pairwise constraints and the K-nearest neighbors ($KNNs$) of the data samples. To compare them with our set M^{SS}, we set K to 1 ~~for W^{KNN1}, W^{KNN2}, W^{KNN3} and W^{KNN4}~~ and define $1NN = \{(x_i, x_j) \in X^2 \mid x_i \in X^U$ or $x_j \in X^U$ but $x_i = 1NN(x_j)$ or $x_j = 1NN(x_i)\}$ to describe links that are generated by these four matrices. Figure 1 demonstrates that M^{SS} better respects the geometric structure of classes than W^{KNN1}, W^{KNN2}, W^{KNN3} and W^{KNN4} ~~$1NN$~~ by taking into account the margin between unlabeled samples and prototypes of different classes.

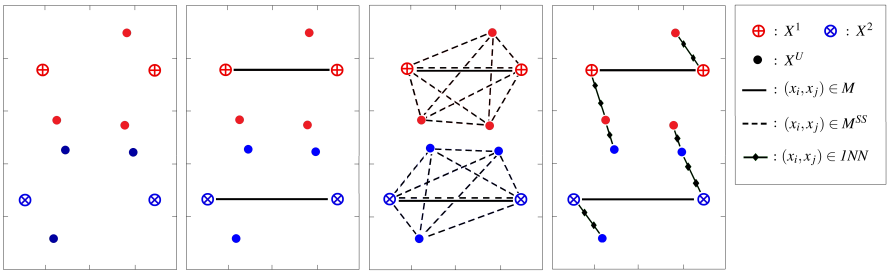


Figure 1: Must-link pairs in a semi-supervised learning context. (a) Set of unlabeled data samples (displayed with color of class labels for a better interpretation of results) and two prototype sets X^1 and X^2. (b) Set M. (c) Set M^{SS}. (d) ~~Set $1NN$~~ Links corresponding to similarity matrices W^{KNN1}, W^{KNN2}, W^{KNN3} and W^{KNN4}.

Finally, $\varepsilon^{SS}(F_m) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(w_{ij}(F_m) - \hat{w}_{ij}^{SS} \right)^2$ makes it possible to assess the ability of a feature subset to preserve the pairwise constraints provided

by the user and the extended pairwise constraints provided by M^{SS}. These extended pairwise constraints well represent the geometric structure of the classes.

The supervised constraint scores C^1 and C^2 do not compute the similarity between prototypes in any feature space, while the classical semi-supervised constraint scores C^c $(c = 3, \ldots, 7)$ evaluate the similarity between samples only in the original feature space. In contrast, our scores ε^S and ε^{SS} evaluate the similarity matrix $W(F_m)$ in the selected feature subspace F_m. This difference is essential because the classification or clustering of the data samples is performed in the selected feature subspace.

4.2. Feature selection procedure

Feature selection reduces data dimensionality by selecting the most relevant sets of original features. Because the state-of-the-art scores C^c $(c = 1, \ldots, 7)$ evaluate the relevance of each feature, the selection procedure ranks features with respect to one of the scores to identify the most relevant sets of features.

For illustration purposes, we consider the Dermatology database from UCI repository [17], which contains 366 samples characterized by 34 features and is regrouped into 6 classes. We randomly pick out $p = 3$ prototypes for each class. Figure 2 presents the variation in scores C^c $(c = 1, \ldots, 7)$ with respect to the number m of features. It should be noted that the score of a subset composed of the best m features in Fig. 2 is computed as the cumulative sum of the lowest scores divided by m. It can be seen that the C^1, C^3, C^4, C^5, C^6 and C^7 curves monotonically increase while C^2 monotonically decreases. As these scores monotonically vary, they do not allow for the identification of the optimal number of features.

Because our score ε^* can evaluate the relevance of a feature subset at once, it is more advantageous to use a selection procedure capable of combining features between them. For this, we use the sequential forward feature selection technique due to its simplicity [11].

To evaluate the relative relevance of d features, we first consider each feature

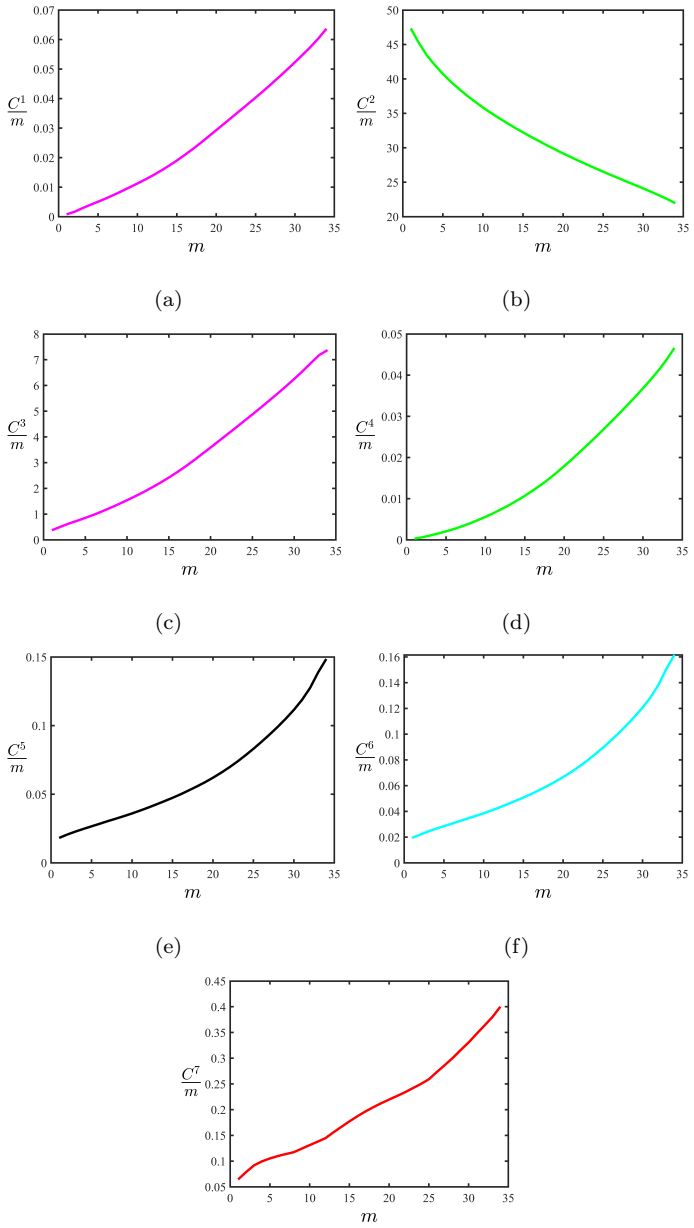


Figure 2: State-of-the-art constraint scores with respect to the number of selected features m for the Dermatology database: (a) $\frac{C^1}{m}$, (b) $\frac{C^2}{m}$, (c) $\frac{C^3}{m}$, (d) $\frac{C^4}{m}$, (e) $\frac{C^5}{m}$, (f) $\frac{C^6}{m}$, (g) $\frac{C^7}{m}$

one by one ($m = 1$). The feature f_r that minimizes $\varepsilon^*(F_1)$ with $F_1 = \{f_r\}$ is selected and is combined with each of the remaining $d - 1$ features. The corresponding $d - 1$ scores $\varepsilon^*(F_2)$ are then computed, and the pair of features that minimizes $\varepsilon^*(F_2)$ is retained. When m of d features have been selected, the $(m + 1)$-th feature that minimizes $\varepsilon^*(F_{m+1})$ when combined with the m previously chosen features is selected. This suboptimal procedure is iterated until d features have been ordered. The subset $F_{\hat{m}}$ that corresponds to the minimum of $\varepsilon^*(F_m)$ is finally selected (see Algorithm 1). ~~The pseudo-code for this feature selection procedure is outlined in Algorithm~~ 1.

Algorithm 1 Feature selection procedure.

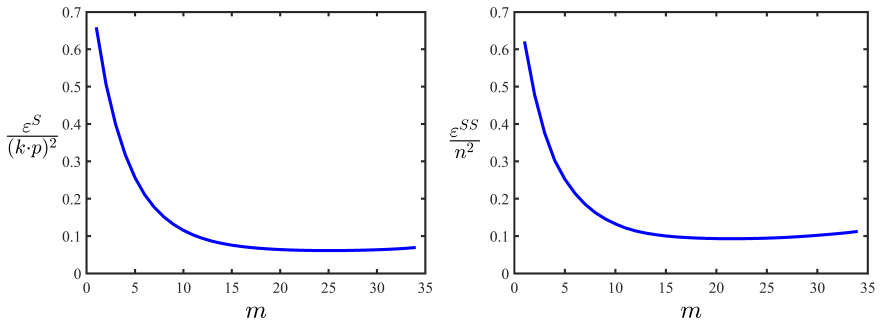
Input: Set of d feature $F_d = \{f_1, \ldots, f_r, \ldots, f_d\}$.

1. Create empty set of features $F_0 = \{\varnothing\}$.
2. For $m = 1$ to d
   a. Select the most relevant feature f_r^+
   $$f_r^+ = \underset{f_r \in F_d \setminus F_{m-1}}{\arg\min} \Big(\varepsilon^{SS}(F_{m-1} \cup \{f_r\})\Big).$$
   b. Update $F_m = F_{m-1} \cup \{f_r^+\}$.
3. Select the number $\hat{m}$ of features such that
   $$\hat{m} = \underset{m=1,2,\ldots,d}{\arg\min} \Big(\varepsilon^{SS}(F_m)\Big).$$

Output: Subset of $\hat{m}$ relevant features $F_{\hat{m}}$.

Figure 3 illustrates the variation of the proposed constraint scores ε^S and ε^{SS} with respect to the number of features m in the Dermatology database, from which $p = 3$ prototypes have been selected for each class. It should also be noted that as for all state-of-the-art scores C^c $(c = 1, \ldots, 7)$, ε^S is normalized by the total number $(k \cdot p)^2$ of constraints, and ε^{SS} is normalized by the total number n^2 of cells in the similarity matrix W. The curves of ε^S and ε^{SS} are both pseudo-convex . ~~The proposed supervised and semi-supervised constraint scores~~ and their minima can thus ~~help~~ used to identify the optimal number of features.

It should be emphasized that although this feature selection procedure eval-



(a) (b)

Figure 3: Proposed constraint scores with respect to the number of selected features m for the Dermatology database: (a) $\frac{\varepsilon^S}{(k\cdot p)^2}$ and (b) $\frac{\varepsilon^{SS}}{n^2}$

uates the features jointly and considers the dependency among them by virtue of our proposed constraint score, it belongs to the filter approach and retains all the advantages of this approach comparatively to wrapper approach (i.e., independence from the classifier, ~~high computation speed~~, simple and rapid implementation).

5. Experiments on benchmark databases

We evaluated and compared our proposed constraint score with several constraint scores and several well-known feature selection methods on datasets originating from benchmark databases. We first examined the supervised feature scores (ε^S, C^1, and C^2). Then, we assessed the performance attained by semi-supervised feature scores (ε^{SS}, C^3, C^4, C^5, C^6 and C^7). Because the data were scaled between 0 and 1, the scaling parameter σ used to compute the similarity matrices was set to 1. Feature selection procedures were performed on the training datasets and repeated over 100 runs. In each feature selection run, we automatically generated a set of pairwise constraints as follows. For each class, we randomly selected from the training dataset X, k prototype subsets X^l ($|X^l| = p$). Then, we deduced sets M, C, and M^{SS} of the pairwise constraints using Eqs. (4), (5), and (38), respectively.

5.1. Databases

Our experiments were performed using twelve ~~six~~ well-known and commonly used benchmark databases in the feature selection framework, namely, the Wisconsin Breast Cancer (WBCD), Image Segmentation, Wisconsin Diagnostic Breast Cancer (WDBC), Ionosphere, Dermatology, Libras Movement, Multiple Features and CNAE-9 databases from the UCI repository [17], and the Olivetti Research Laboratory (ORL), Yale, Pie10P, and ALLAML databases from the ASU feature selection repository [18]. The details of these databases are provided in Table 1. All the databases had numerical features, and the class label of each data sample was clearly defined. We normalized each dataset between 0 and 1 to ensure that the scales of all features were equal. For WBCD, 410 samples were used as the training dataset and 273 samples were used as the test dataset, while for WDBC, 376 samples were used as the training dataset and 193 samples were used as the test dataset [2]. For the remaining databases, ~~Ionosphere, Dermatology, Libras Movement, and ORL,~~ we used half of the data samples from each class as the training dataset and the remaining data samples as the test dataset [1]. It should be noted that the classes were ~~not~~ equiprobable for all databases with the exception of the WBCD, WDBC, Ionosphere, Dermatology and ALLAML ~~Libras Movement and ORL~~ databases. We extracted $(k \cdot p)$ prototypes from the training sample set of each database with p ranging from 2 to 4. ~~Table~~ 1 ~~also displays the number $(k \cdot p)$ of prototypes extracted from the training sample set of each database with p ranging from 2 to 4.~~ These prototypes were used to build pairwise constraints and select relevant features. They were also used in the classification step to evaluate the relevance of the selected features. It can be seen that the curse of dimensionality is faced, as the number of prototypes always remains less than the number of features d.

5.2. Supervised feature selection

For supervised learning, only k prototype subsets X^l are used to select features. The performance obtained by the proposed supervised constraint score ε^S is compared first with that attained by supervised constraint scores C^1 and

Table 1: Description of benchmark databases used in experiments

Database	#Features	#Samples	#Training/#Test	#Classes
WBCD	09	683	410/273	2
Image Segmentation	19	2310	1155/1155	7
WDBC	30	569	376/193	2
Ionosphere	34	351	176/175	2
Dermatology	34	366	183/183	6
Libras Movement	90	360	180/180	15
Multiple Feature	649	2000	1000/1000	10
CNAE-9	856	1080	540/540	9
Yale	1024	165	90/75	15
ORL	1024	400	200/200	40
Pie10P	2420	210	110/100	10
ALLAML	7129	72	37/35	2

C^2, then with that achieved by supervised feature selection methods including ReliefF [19] and minimal-redundancy-maximal relevance (mRMR) [20]. In addition to these filter feature selection methods, a wrapper-type supervised feature selection [21] is used for the comparison.

The relevance of the selected subset of features is evaluated by the accuracy measure of the test dataset. Each test data point is projected onto the retained feature space and is assigned to one of the k classes according to the nearest neighbor rule. In general, the entire training dataset is used as prototypes by the nearest neighbor classifier to classify the test data, whereas only few prototypes are used by the supervised constraint scores [1, 5, 7]. Here, we propose performing the selection and evaluation with only the same available k prototype subsets X^l ($|X^l| = p$). In this way, these two steps are performed in conditions similar to those of real-life applications.

Because the accuracy highly depends on the number of selected features and prototypes, we first evaluate accuracy with respect to the number of features m

for a given number of prototypes p. Then, for a relevant number of features, we compare the accuracy obtained by the supervised constraint scores with different numbers of prototypes.

5.2.1. Accuracy versus number of selected features

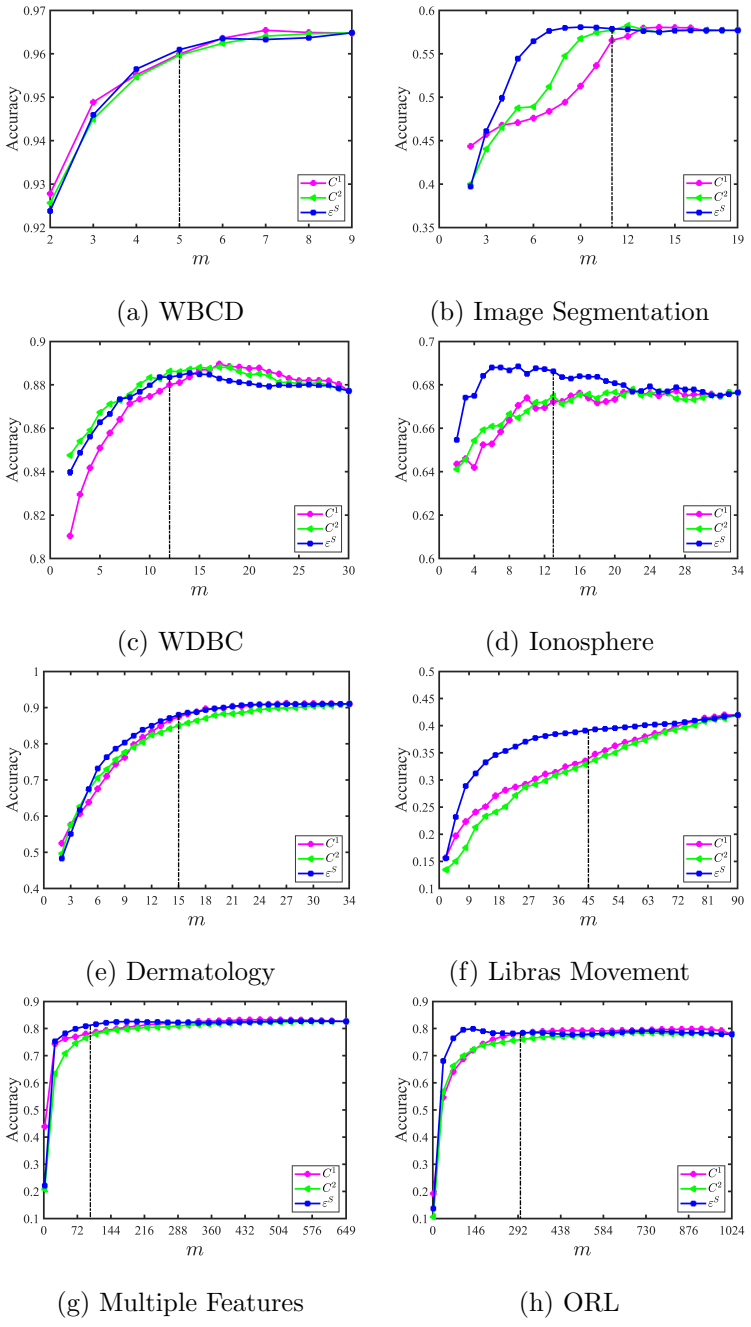
Figure 4 illustrates the average accuracy for ε^S, C^1, and C^2 obtained when the number of prototypes p is set to 3. From this figure, it can be seen that in most cases, the average accuracy obtained with the proposed scores ε^S is higher than that obtained with constraint scores C^1 and C^2. The curves of ε^S differ from the curves of C^1 and C^2 for the Image Segmentation, Ionosphere, ~~Dermatology,~~ and Libras Movement databases. However, the curves of ε^S, C^1, and C^2 overlap for the other databases. Because the performance of these scores is averaged over 100 runs with different generations of constraints, comparing them is difficult. As a result, we compare these scores by examining their accuracy at each of the 100 runs.

For a fixed number of selected features, in each of the 100 runs, we propose ranking the supervised constraint scores ε^S, C^1, and C^2 in descending order of accuracy. Let us denote $rank_q^{[c]}$ the rank of the supervised constraint scores $c = \varepsilon^S$, C^1, or C^2 at run q. The score with the highest accuracy is ranked 1, while the score with the lowest accuracy is ranked 3. Scores with the same accuracy have the same rank. We compute the rank sum $R^{[c]}$ for each score as follows:

$$R^{[c]} = \sum_{q=1}^{100} rank_q^{[c]} \tag{40}$$

The method with the lowest rank sum is considered to be the score that provides the best results.

For the WBCD, Image Segmentation, WDBC, Ionosphere, Dermatology, Libras Movement, Multiple Features, and ORL databases, the accuracy of the supervised constraint scores appears to be stable when the number of selected features is higher than $5, 11, 12, 13, 15, 45, 100$, and 300, respectively (see Fig. 4). The rank sum of each supervised constraint score is then computed by considering the first $5, 11, 12, 13, 15, 45, 100$, and 300 features for the WBCD, Image



(a) WBCD (b) Image Segmentation

(c) WDBC (d) Ionosphere

(e) Dermatology (f) Libras Movement

(g) Multiple Features (h) ORL

Figure 4: Accuracy versus number of selected features m by the supervised constraint scores on eight ~~six~~ benchmark databases

Segmentation, WDBC, Ionosphere, Dermatology, Libras Movement, Multiple Features, and ORL databases, respectively.

In Table 2, the row with an asterisk for each of the eight ~~six~~ databases, provides the rank sum when the number of prototypes p is set to 3. These rows indicate that ε^S provides the lowest rank sum for all databases (indicated in bold) except for the WDBC database. Thus, the accuracy provided by the features selected by ε^S is higher than those obtained by C^1 and C^2.

5.2.2. Accuracy versus number of prototypes

In this section, we compare the accuracy obtained by the supervised constraint scores ε^S, C^1, and C^2 for various number of prototypes p. For this purpose, we use the number of selected features that is provided in the previous section. Because the number of prototypes p must be higher than or equal to 2 to generate at least one must-link constraint by class, it ranges from 2 to 4.

Table 2 displays the rank sums obtained for the eight databases and demonstrates that the features selected by ε^S provide higher accuracy rates than those obtained by the features selected by C^1 and C^2. In fact, the score ε^S provides the lowest rank sum (indicated in bold) 19 times out of the 24 rows in Table 2.

The improvement provided by ε^S has two main causes. First, because ε^S estimates the relevance of a subset of features, it takes into account the correlation between them, whereas C^1 and C^2 ignore this correlation. Second, ε^S computes the similarity matrix between samples in the considered subset of features (see. Eq. (36)) whereas C^1 and C^2 do not compute the similarity between samples.

5.2.3. Comparison with other supervised feature selection methods

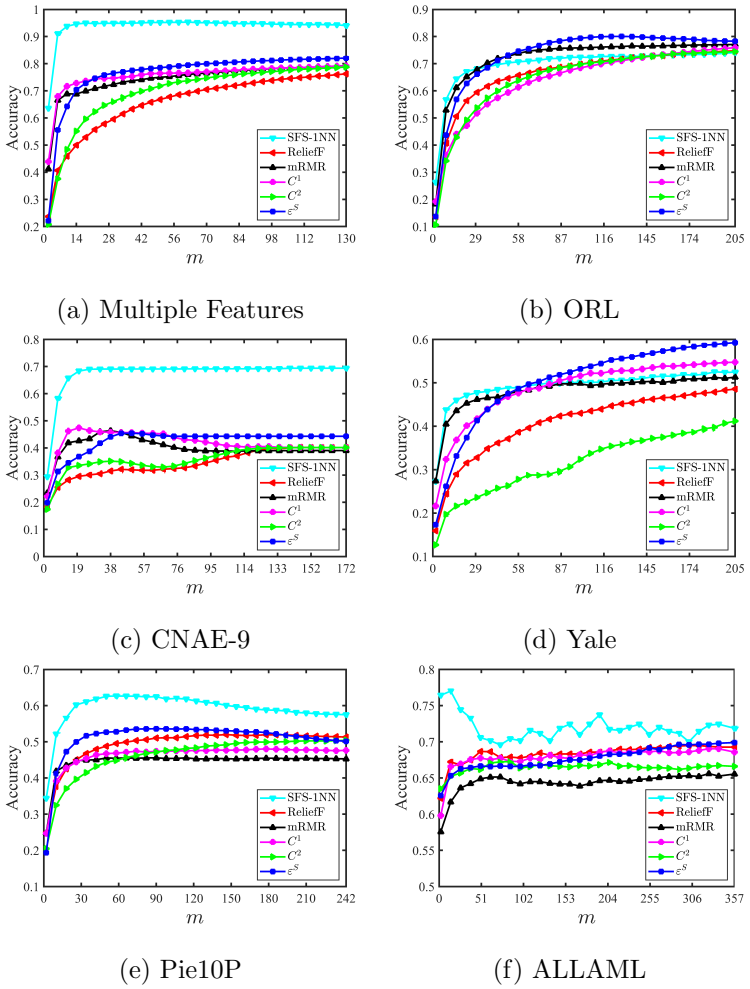
To further illustrate the effectiveness of our supervised constraint score ε^S, it is compared with well-established supervised feature selection methods including ReliefF [19] and mRMR [20]. A wrapper-type supervised feature selection that selects features in a forward sequential manner by means of nearest neighbor classifier (SFS-1NN) is also used for the comparison [21]. Figure 5 displays

Table 2: Rank sum of different supervised constraint scores for different numbers of prototypes

Database	m	p	ε^S	C^1	C^2
WBCD	5	2	**148**	171	161
		3*	**160***	169*	**160***
		4	172	167	**158**
Image Segmentation	11	2	196	**183**	192
		3*	**170***	209*	184*
		4	184	202	**182**
WDBC	12	2	**171**	188	**171**
		3*	186*	194*	**176***
		4	183	**176**	194
Ionosphere	13	2	**185**	192	**185**
		3*	**184***	194*	197*
		4	**179**	192	203
Dermatology	15	2	**170**	194	223
		3*	**164***	199*	227*
		4	**160**	202	223
Libras Movement	45	2	**127**	233	232
		3*	**118***	225*	251*
		4	**125**	223	234
Multiple Features	100	2	**156**	216	226
		3*	**133***	218*	246*
		4	**116**	238	245
ORL	300	2	**141**	184	248
		3*	**151***	160*	278*
		4	**100**	**100**	**100**

the average accuracies over 100 runs achieved by the supervised feature selection methods on high dimensionality databases compared with the r% selected features, when the number p of prototypes is set to 3. r is set to 5 for datasets

with a number d of features greater than 3000, 10 for datasets with a number of features in the range $[2000 - 3000]$, 20 for datasets with a dimensionality in the range $[300 - 2000]$ and 100 for lower dimensionality (less than 300). Figure 5 shows that the averaged accuracy on the features selected by ε^S is usually higher than that of the filter methods but remains lower than that of the wrapper method SFS-1NN. This observation is confirmed by the averaged accuracies over the top $r\%$ features that are reported in Table 3. To better assess the results obtained for each algorithm, we ranks the algorithms for each dataset separately, the best performing algorithm obtaining the rank of 1, the second best rank 2, etc. In case of ties, we assigns the same rank. Finally, the average rank over all databases is depicted in the bottom row of Table 3. We can find overall that our score ε^S outperforms the four other filter methods, but remains behind the wrapper method.



(a) Multiple Features (b) ORL

(c) CNAE-9 (d) Yale

(e) Pie10P (f) ALLAML

Figure 5: Accuracy versus number of selected features m by supervised feature selection methods on six high dimensionality databases

Table 3: Mean and standard deviations of accuracy over the $r\%$ selected features

Database	SFS-1NN	ReliefF	mRMR	C^1	C^2	ε^S
WBCD	96.75 ± 2.84	95.59 ± 3.28	94.99 ± 3.94	95.63 ± 3.19	95.51 ± 3.49	95.53 ± 3.38
Image Segmentation	60.77 ± 6.82	53.91 ± 7.68	55.55 ± 6.97	52.94 ± 7.03	53.80 ± 6.76	55.43 ± 6.58
WDBC	90.88 ± 5.37	87.29 ± 7.49	86.64 ± 7.99	87.41 ± 7.41	87.88 ± 7.35	87.60 ± 7.54
Ionosphere	77.66 ± 6.82	66.95 ± 8.80	66.56 ± 9.59	66.96 ± 8.72	67.02 ± 8.73	68.02 ± 8.44
Dermatology	90.90 ± 2.80	75.10 ± 5.60	83.96 ± 5.50	83.22 ± 5.10	82.47 ± 5.76	84.00 ± 5.29
Libras Movement	38.06 ± 5.01	32.25 ± 4.32	36.17 ± 4.27	33.41 ± 4.43	32.09 ± 4.53	37.32 ± 4.61
Multiple Feature	94.24 ± 1.01	65.99 ± 5.39	74.85 ± 4.55	75.91 ± 5.32	69.89 ± 4.37	76.97 ± 4.31
CNAE-9	67.96 ± 3.73	34.51 ± 6.06	40.38 ± 5.59	42.38 ± 5.72	36.04 ± 5.65	42.61 ± 5.27
Yale	49.57 ± 4.94	41.17 ± 5.73	48.49 ± 5.49	49.05 ± 5.56	31.75 ± 3.79	50.60 ± 5.49
ORL	70.40 ± 3.07	66.77 ± 3.17	72.75 ± 2.92	64.67 ± 3.50	65.19 ± 2.64	73.84 ± 2.44
Pie10P	59.60 ± 11.58	49.63 ± 7.14	44.90 ± 4.55	46.54 ± 5.30	46.45 ± 8.14	51.39 ± 6.50
ALLAML	71,95 ± 7,65	68.51 ± 13.27	64.50 ± 13.10	68.03 ± 13.20	66.51 ± 14.48	67.82 ± 14.48
Average rank	1.25	4.5	4.42	3.84	4.75	2.33

5.3. Semi-supervised feature selection

Semi-supervised learning analyzes the k prototype subsets X^l in addition to unlabeled samples that belong to the learning dataset X. Because our score ε^{SS} uses unlabeled samples to construct the new set M^{SS} of must-link pairwise constraints, it is interesting to first evaluate the relevance of M^{SS}. Then, the performance obtained by the proposed semi-supervised constraint score ε^{SS} is compared with that attained by the semi-supervised constraint scores C^c ($c = 3, \ldots, 7$) and by two semi-supervised feature selection methods namely the semi-supervised pairwise constraint-guided sparse (SCGS) learning method [22] and the ensemble constrained Laplacian score (EnsCLS) method [16].

For this purpose, we follow the same experimental scheme as that of supervised learning. First, we evaluate the accuracy with respect to the number of features m for a given number of prototypes p. Then, for a number of features m that are considered relevant, we compare the accuracy achieved by the semi-supervised constraint scores with different numbers of prototypes.

5.3.1. Correct must-link pairs of M^{SS}

The proposed semi-supervised constraint score ε^{SS} builds the set of new must-link pairs M^{SS} (see Eq. (38)), which extends set M. To assess the resulting set M^{SS}, we propose comparing it with M by computing the rate of new must-link pairs (NML). Then, the rate of correct new must-link pairs ($CNML$) is computed as follows to evaluate the relevance of these new constraints:

$$NML = \frac{|M^{SS}|}{|M|} \tag{41}$$

$$CNML = \frac{|\ \{(x_i, x_j) \in M^{SS} \setminus M\ \ |\ \ \omega(x_i) = \omega(x_j)\}\ |}{|M^{SS} \setminus M|} \tag{42}$$

where $\omega(x_i)$ is the true class of the data sample x_i. Here, $NML \geq 1$ and $0 \leq CNML \leq 1$, and higher NML and $CNML$ lead to an improved M^{SS}.

Table 4 lists the means of NML and $CNML$ for eight databases over 100 different sets of prototypes when the number of prototypes p ranges from 2 to

4. To interpret NML with set M, Table 4 also displays its size $|M| = k \cdot p^2 - k \cdot p$.

Table 4: NML and $CNML$ achieved for eight databases

Database	p	$\|M\|$	NML	$CNML$
WBCD	2	04	24037	0.8596
	3	12	7956.2	0.8647
	4	24	3915.4	0.8799
Image Segmentation	2	14	15234	0.4690
	3	42	4922.4	0.4966
	4	84	2408.2	0.5126
WDBC	2	04	19638	0.7962
	3	12	6392	0.8094
	4	24	3140.9	0.8337
Ionosphere	2	04	5450.1	0.5826
	3	12	1752.6	0.5962
	4	24	879.9	0.6008
Dermatology	2	12	534.2	0.8050
	3	36	174.4	0.8504
	4	72	86.5	0.8630
Libras Movement	2	30	81.19	0.4305
	3	90	25.0	0.5077
	4	180	12.0	0.5684
Multiple Features	2	20	5198.6	0.6269
	3	60	1698.3	0.6939
	4	120	841.3	0.7401
ORL	2	80	11.1	0.6324
	3	240	3.4	0.7694
	4	480	1.7	0.8780

This table demonstrates that for each database, NML decreases when the

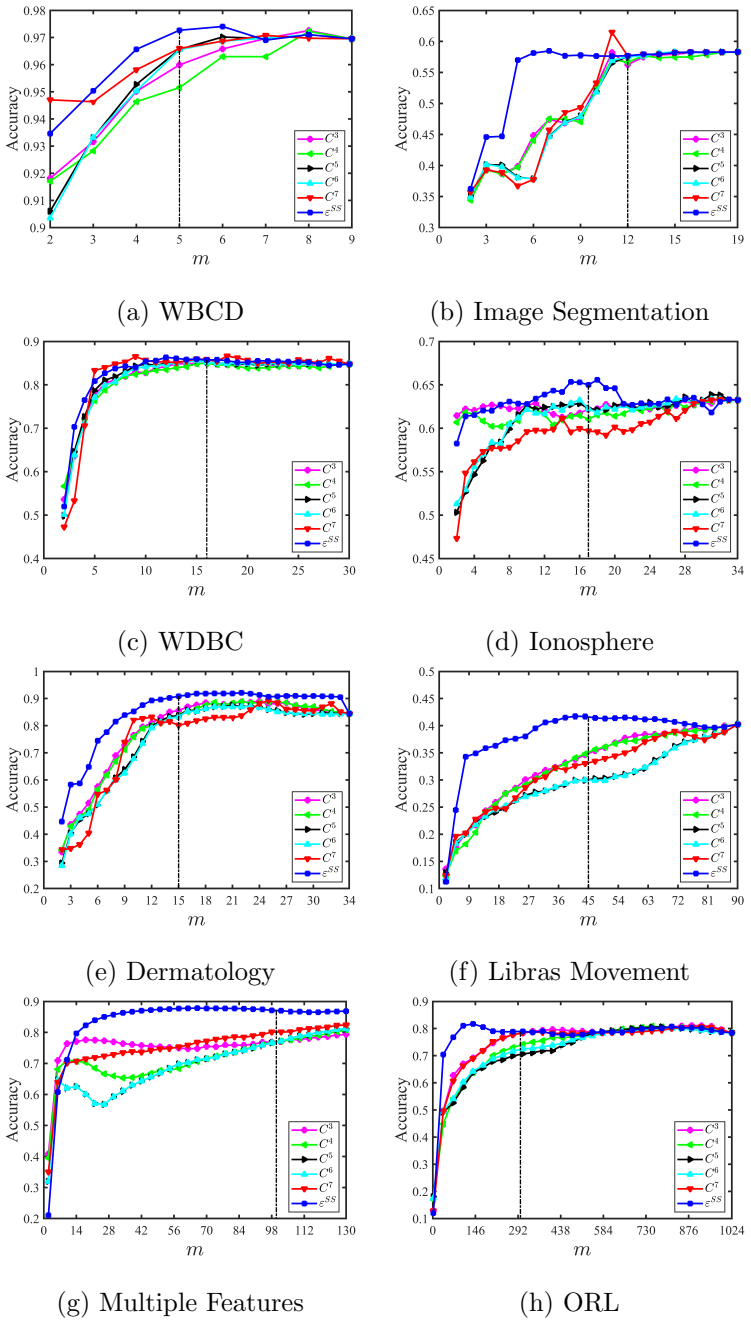
number of prototypes p increases. NML also varies strongly from one database to another and tends to be high when the number $|M|$ of available must-link pairs is low. Furthermore, $CNML$ is close to or exceeds 0.7 for all databases except for the Image Segmentation, Libras Movement and Ionosphere databases. Therefore, we can deduce that the majority of these new must-link pairwise constraints should be used to select features.

5.3.2. Accuracy versus number of selected features

The performance of semi-supervised constraint scores is measured according to the classification accuracy of the test dataset. In semi-supervised learning, the evaluation step is generally performed in the supervised context, in which the training dataset is used as prototypes by the nearest neighbor classifier to classify the test data [1, 6, 7]. Instead, the features are selected in the semi-supervised learning context in which only few prototypes are considered. To perform selection and evaluation in the same semi-supervised context that is similar to real-life applications, we follow the strategy proposed by Kalakech et al. [12] by applying the constrained spectral clustering detailed in [23] instead of the constrained K-means algorithm to classify the unlabeled training data samples. Then, we use the sample as prototypes to classify the test data subset. Finally, each test data point is projected onto the same retained feature subspace and is assigned to a class using the nearest neighbor rule, which uses the available prototypes in addition to the training data samples that have been previously classified.

Figure 6 presents the average accuracy for ε^{SS}, C^3, C^4, C^5, C^6 and C^7 obtained in the eight test sets using the semi-supervised learning evaluation over 100 runs when the number of prototypes p was set to 3. This figure indicates that our semi-supervised constraint score ε^{SS} provides higher accuracy than that achieved by the other semi-supervised constraint scores C^c $(c = 3, \ldots, 7)$. The curves of ε^{SS} clearly differ from those of C^c $(c = 3, \ldots, 7)$, which overlap.

We also determined the rank sum $R^{[c]}$ obtained by the semi-supervised constraint scores ε^{SS} and C^c $(c = 3, \ldots, 7)$ for the eight databases over 100 runs



(a) WBCD (b) Image Segmentation

(c) WDBC (d) Ionosphere

(e) Dermatology (f) Libras Movement

(g) Multiple Features (h) ORL

Figure 6: Accuracy versus number of selected features m by the semi-supervised constraint scores for eight benchmark databases

with $p = 3$ prototypes. The rank sum of each score was computed by considering the first $5, 12, 16, 17, 15, 45, 100$ and 300 features for the WBCD, Image Segmentation, WDBC, Ionosphere, Dermatology, Libras Movement, Multiple Features, and ORL databases, respectively (see Fig. 6). The rows with an asterisk in Table 5 indicate that ε^{SS} provides the lowest rank sum for all the databases (indicated in bold) except for the WDBC ~~Ionosphere~~ database. Thus, the accuracy provided by the features selected by ε^{SS} is higher than that obtained by C^c $(c = 3, \ldots, 7)$ in the majority of cases.

5.3.3. Accuracy versus number of prototypes

In this section, we compare the accuracy obtained by the semi-supervised constraint scores $\varepsilon^{SS}, C^3, C^4, C^5, C^6$ and C^7 versus the number of prototypes p. For this purpose, we use the number of selected features provided in Sect.5.3.2.

Table 5 displays the rank sum over 100 different sets of prototypes when the number of prototypes p ranges from 2 to 4. It can be seen that ε^{SS} provides the lowest rank sum 20 out of 24 times. These results indicate that the accuracy provided by features selected by ε^{SS} is higher than the accuracy of those obtained by C^c $(c = 3, \ldots, 7)$ in most cases.

In addition, ε^{SS} outperforms C^c $(c = 3, \ldots, 7)$ because it takes into account the correlation between features. Furthermore, it computes the similarity matrix between samples in the selected feature subspace, while C^c $(c = 3, \ldots, 7)$ compute the similarities in the high-dimensional original feature space.

5.3.4. Comparison with other semi-supervised feature selection methods

To further illustrate the effectiveness of our semi-supervised constraint score ε^{SS}, it is compared with other semi-supervised feature selection methods, namely the SCGS [22] and EnsCLS [16]. In addition, two unsupervised feature selection methods including the Laplacian score (LS) [15] and the spectral feature selection (Spec) method [24], which are based on graph theory and similarity matrices, are used for the comparison. Note that EnsCLS combines both a

Table 5: Rank sum of different semi-supervised constraint scores for different number of prototypes

Database	m	p	ε^{SS}	C^3	C^4	C^5	C^6	C^7
WBCD	5	2	254	257	374	246	**244**	251
		3*	**192***	304*	347*	250*	251*	209*
		4	**206**	307	332	253	250	221
Image Segmentation	12	2	**179**	271	284	211	218	**179**
		3*	**155***	282*	292*	160*	164*	156*
		4	**176**	210	242	192	196	**176**
WDBC	16	2	**282**	358	359	305	314	324
		3*	299*	351*	372*	303*	306*	**282***
		4	308	314	341	316	**301**	306
Ionosphere	17	2	**262**	334	332	321	323	379
		3*	**283***	326*	334*	314*	307*	414*
		4	**291**	332	333	307	303	434
Dermatology	15	2	**208**	316	367	282	292	482
		3*	**166***	319*	343*	314*	318*	498*
		4	**177**	306	374	309	324	466
Libras Movement	45	2	**132**	318	331	409	404	432
		3*	**128***	306*	326*	462*	458*	358*
		4	**138**	312	323	464	483	313
Multiple Features	100	2	**149**	486	369	311	311	364
		3*	**119***	422*	407*	378*	375*	300*
		4	**108**	375	454	382	379	297
ORL	300	2	**154**	246	378	508	518	240
		3*	**174***	210*	416*	567*	497*	189*
		4	200	**100**	500	600	200	400

resampling of data (bagging) and a random selection of features (random subspaces) strategy for generating different data views. The constraint score C^6 is

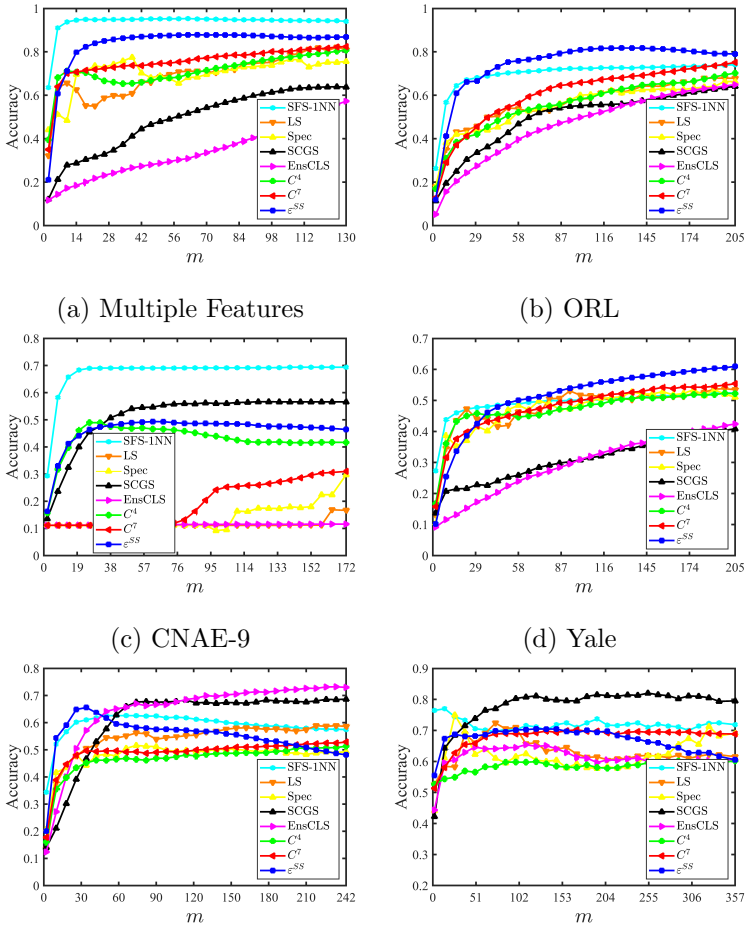
then used to measure feature relevance on each data replicate, and the score average of all features across all ensemble components is used to rank all features. In SCGS the semi-supervised feature selection is formulated as a optimization problem where the objective function contains three terms. The first term is the empirical loss on labeled data sample, the second term is a discriminant regularization term focusing on the local structure of data reflected by pairwise constraints, and the last term is an unsupervised estimation on intrinsic geometry distribution of the whole training data. The wrapper-type supervised feature selection (SFS-1NN) is used as baseline. For the sake of fairness, the selection and evaluation are performed in the same conditions as for the semi-supervised constraint scores. For the unsupervised methods, only the unlabeled training data samples are used for the feature selection and are then labeled by the constrained spectral clustering before being used as prototypes by the nearest neighbor classifier. Figure 7 displays the average accuracies over 100 runs achieved by the above feature selection approaches on high dimensionality datasets compared with the $r\%$ most relevant features, when the number of prototypes p is set to 3. The mean and standard deviation of accuracies, computed over the top $r\%$ features for all databases, are reported in Table 6. The averaged rank of each algorithm over all datasets, is reported in the bottom row of the Table 6. From Fig. 7 and Table 6, we can make the following observations:

- When we compare ε^{SS} with semi-supervised features selection methods, we notice that SCGS outperforms ε^{SS} for only 3 out 12 databases (CNAE9, Pie10P and ALLAML), and EnsCLS works better than ε^{SS} for only two databases (Ionosphere and Pie10P). ε^{SS} outperforms the two constraint scores C^4 and C^7 for all databases except for the ALLAML database where C^7 appears more efficient.

- By comparing ε^{SS} with the unsupervised features selection methods, we find that ε^{SS} outperforms LS and Spec for all databases except for ALLAML where Spec produces a better averaged accuracy and WDBC where LS displays the same averaged accuracy than ε^{SS}.

However, overall, ε^{SS} outperforms all of these competitive methods given its

lower average rank value. In some cases, ε^{SS} performs better results than the wrapper method SFS-1NN as for Multiple Features, Yale and ORL databases.



(a) Multiple Features (b) ORL

(c) CNAE-9 (d) Yale

(e) Pie10P (f) ALLAML

Figure 7: Accuracy versus number of selected features m by semi-supervised and unsupervised feature selection methods on six high dimensionality databases

Table 6: Mean and standard deviations of accuracy over the $r\%$ selected features

Database	LS	Spec	SCGS	EnsCLS	C^4	C^7	ε^{SS}
WBCD	95.36 ± 4.74	92.58 ± 7.42	92.98 ± 6.66	92.21 ± 8.06	95.14 ± 5.34	96.20 ± 4.16	96.34 ± 4.00
Image Segmentation	50.22 ± 6.51	54.39 ± 6.92	46.35 ± 8.29	38.20 ± 5.56	50.48 ± 7.69	50.63 ± 6.86	55.26 ± 5.98
WDBC	83.04 ± 7.74	82.05 ± 7.01	81.68 ± 8.60	73.41 ± 8.80	81.41 ± 7.67	82.41 ± 7.07	83.03 ± 7.29
Ionosphere	59.86 ± 8.43	64.06 ± 10.52	57.96 ± 8.95	65.29 ± 11.15	61.81 ± 9.19	59.74 ± 8.19	63.15 ± 10.53
Dermatology	70.41 ± 6.33	66.20 ± 6.29	71.07 ± 7.97	64.03 ± 5.60	77.84 ± 7.91	75.69 ± 6.49	84.98 ± 5.91
Libras Movement	32.07 ± 3.81	29.42 ± 4.01	32.19 ± 4.56	31.25 ± 3.94	32.59 ± 4.51	31.84 ± 4.74	38.46 ± 4.13
Multiple Feature	69.55 ± 4.28	70.60 ± 4.06	49.14 ± 4.82	34.07 ± 3.75	71.76 ± 5.74	75.66 ± 6.87	84.22 ± 4.41
CNAE-9	11.50 ± 0.40	14.09 ± 1.67	51.71 ± 6.50	11.39 ± 0.30	43.35 ± 7.28	19.13 ± 2.28	46.60 ± 6.78
Yale	48.10 ± 3.56	47.82 ± 4.00	30.84 ± 5.02	29.62 ± 4.27	47.4 ± 4.22	48.31 ± 5.56	51.76 ± 4.76
ORL	56.86 ± 2.49	55.38 ± 2.50	50.13 ± 3.02	46.86 ± 2.48	56.35 ± 2.95	61.09 ± 3.41	75.16 ± 2.33
Pie10P	54.09 ± 8.80	48.26 ± 7.19	61.36 ± 10.84	64.81 ± 7.67	46.90 ± 9.03	49.32 ± 8.49	55.56 ± 7.44
ALLAML	63.25 ± 14.16	62.74 ± 12.48	78.30 ± 12.35	61.55 ± 11.42	58.50 ± 15.19	68.16 ± 11.36	67.35 ± 13.31
Averaged rank	4.08	4.67	4.33	5.75	4.25	3.25	1.67

5.4. Discussion

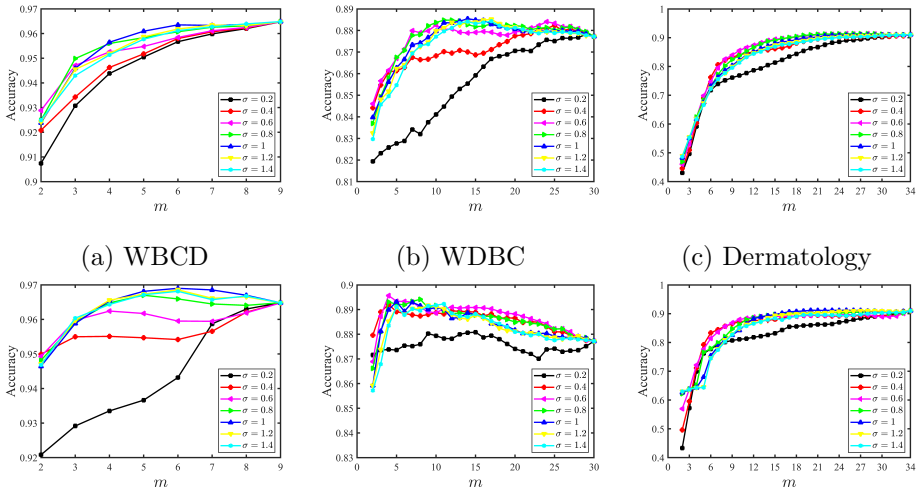
In this section, we first discuss the influence of the scaling parameter σ, the time running and the optimal number of selected features of the proposed constraint scores ε^S and ε^{SS} for feature selection. Then, we compare the accuracy provided by the supervised constraint score ε^S and the semi-supervised constraint score ε^{SS}.

5.4.1. Influence of the scaling parameter σ

Like the constraint scores $C^c, (c = 1, 2, ..., 7)$, our constraint scores ε^S and ε^{SS} depend on the scaling parameter σ used to compute the similarity matrices (see Eqs. 32 to 34). In previous experiments, the scaling parameter σ was set to 1. To analyze its effect on the performance of our constraint scores, we display in Fig. 8 the average accuracy over 100 runs obtained by our constraint scores ε^S and ε^{SS} on the both WBCD, WDBC and Dermatology databases with a number of prototypes p set to 3, when σ ranges from 0.2 to 1.4. To avoid the influence of the spectral clustering, the accuracy is assessed on the test dataset under the same conditions for the both scores, i.e., by using the same nearest neighbor classifier based only on the k prototype subsets X^l ($|X^l| = p$). The curves show that overall, σ has a significant influence on the results, but that a value between 0.6 and 1.2 is best suited to the experimented databases.

5.4.2. Time running

Our proposed score ε^* has the advantage to evaluate the relevance of a feature subset at once thanks to the sequential forward feature selection technique. In order to quantify the computational effort required by the feature selection based on our constraint scores ε^S and ε^{SS}, its running time is compared with that consumed by the wrapper method SFS-1NN that also uses the sequential forward selection as search algorithm. The running times of the mRMR and



(a) WBCD (b) WDBC (c) Dermatology

(d) WBCD (e) WDBC (f) Dermatology

Figure 8: Influence of the scaling parameter σ versus number of selected features m by the proposed constraint scores ε^S and ε^{SS}. Top row: results obtained by using ε^S. Bottom row: results obtained by using ε^{SS}

EnsCLS methods are also represented as an indication. Table 7 displays the computation times obtained by the above algorithms on some datasets of Table 1 with a Intel Core i7 3.60GHz, with 8GB RAM computer. Although the running times are computer dependent, they give an idea of the computation time required by the algorithms for various dimensionalities d, class numbers k, sample numbers n and numbers m of selected features. ε^S and ε^{SS} are relatively more time-consuming than mRMR and EnsCLS, even if that of ε^S remains comparable. In fact, the computation time of ε^S and ε^{SS} in itself is not very high, but it is rather the sequential forward selection procedure which increases their computation time. For ε^{SS}, the high computational time is mainly due to the procedure for propagating labels on unlabeled data samples, the complexity of which is proportional to the size of the data samples. Indeed, we can see on Table 7 that the time consumed by ε^{SS} on the Multiple Features dataset which contains 2000 data samples of dimension 649 is much larger than that obtained on the ALLAML dataset which contains 72 data samples of higher

dimension (7129). However, ε^S and ε^{SS} remain always faster than the wrapper method SFS-1NN. This proves that the calculation of scores is faster than the calculation of the accuracy by the 1NN classifier. It should be emphasized that the computational time cannot be considered as a crucial drawback because it is always possible to reduce it by adopting a faster search strategy, instead of the sequential forward selection, or by reducing the number of unlabeled data samples by clustering as in [8],[16].

Table 7: Run-time (in seconds) of feature selection methods

Database	m	mRMR	EnsCLS	SFS-1NN	ε^S	ε^{SS}
Multiple Features	20	0.135	3.207	318,66	0.491	202.35
	50	0.329	5.488	981,91	2.142	521.60
	100	0.651	8.277	2752.06	8.416	1111.0
ORL	20	0.241	1.605	85.478	2.783	6.429
	50	0.614	2.752	248,34	7.925	18.614
	100	1.190	4.117	602,79	20.78	46.412
ALLAML	20	1.343	0.067	97.033	0.294	3.603
	50	3.417	0.102	250,72	1.045	9.467
	100	7.021	0.146	535.37	3.560	22.452

5.4.3. Optimal number of selected features

The choice of feature number to select is open challenge. It is often specified in advance. The state-of-the-art constraint scores and the feature selection methods used in previous experiments are not able to determine the number of selected features. On the contrary, our scores ε^S and ε^{SS} offer a possibility to automatically determine the optimal number of features. Indeed, as mentioned in Sec. 4.2, the curve of constraint scores ε^S and ε^{SS} versus number of features presents a minimum that can be considered as the optimal number of features. To illustrate this strategy, in Fig. 9 we simultaneously display the average curves of ε^S and ε^{SS} and the average accuracies obtained with ε^S and

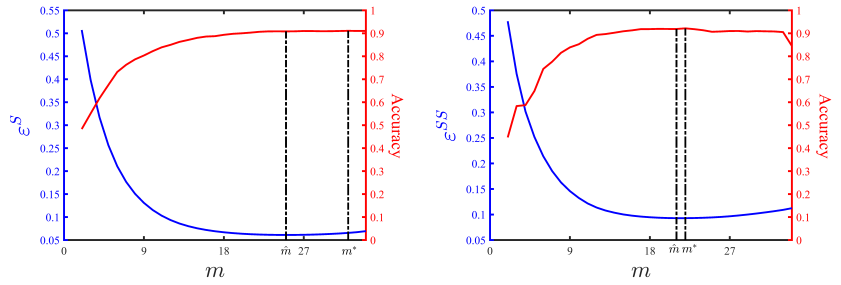
Table 8: Average accuracy and optimal number of selected features

Database	Constraint score ε^S		Constraint score ε^{SS}	
	$Acc(m^*)$	$Acc(\hat{m})$	$Acc(m^*)$	$Acc(\hat{m})$
WBCD	96.48 (9)	96.35 (**6**)	97.41 (**6**)	96.90 (7)
Image Segmentation	58.07 (**9**)	57.70 (18)	58.47 (**7**)	58.28 (18)
WDBC	88.54 (**14**)	87.97 (25)	86.33 (**12**)	85.24 (25)
Ionosphere	68.85 (**9**)	68.73 (12)	65.57 (18)	62.73 (**7**)
Dermatology	91.07 (32)	90.82 (**25**)	92.14 (22)	91.86 (**21**)
Libras Movement	42.03 (89)	41.95 (**90**)	41.73 (**41**)	39.73 (85)
Multiple Feature	82.06 (130)	75.27 (**24**)	87.85 (63)	84.24 (**23**)
CNAE-9	45.68 (**49**)	44.29 (86)	49.43 (**65**)	46.44 (172)
Yale	59.24 (205)	49.05 (**61**)	61.00 (205)	49.19 (**51**)
ORL	80.06 (125)	79.91 (**115**)	81.79 (136)	81.28 (**108**)
Pie10P	53.66 (91)	53.26 (**67**)	65.58 (**34**)	59.30 (60)
ALLAML	70.11 (351)	66.29 (**29**)	71.20 (132)	68.71 (**20**)

ε^{SS} versus number of features on Dermatology database. We can show that the average accuracy $Acc(\hat{m})$ obtained with the subset of features $F(\hat{m})$ that correspond to the minimum of ε^{SS} coincides or is close to the maximum value of accuracy $Acc(m^*)$. The average accuracies $Acc(\hat{m})$ and $Acc(m^*)$ as well as the corresponding optimal number $\hat{m}$ and m^* of features obtained on all databases are depicted in Table 8. From Table 8, we find that, in most cases, the average accuracies $Acc(\hat{m})$ and $Acc(m^*)$ are close but with a number $\hat{m}$ of features less than m^*.

5.4.4. Comparison between supervised constraint score ε^S and semi-supervised constraint score ε^{SS}

In this section, we compare the accuracy provided by the supervised constraint score ε^S and the semi-supervised constraint score ε^{SS}. For this purpose, the accuracy of the test dataset is obtained using the same nearest neighbor classifier based only on the k prototype subsets X^l ($|X^l| = p$). Figure 10 illustrates



(a) ε^S (b) ε^{SS}

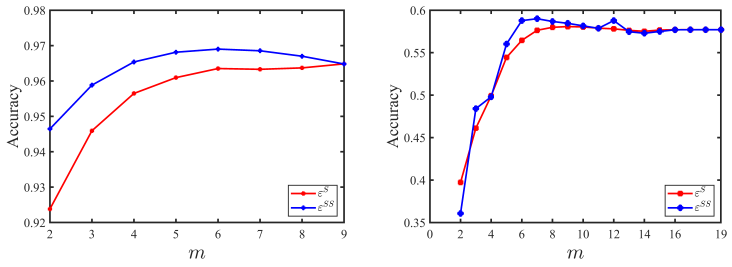
Figure 9: Accuracy-scores ε^S and ε^{SS} versus number of selected features for the Dermatology database

the accuracy obtained when the number of prototypes p is set to 3.

This figure indicates that the average accuracy over 100 runs obtained by the semi-supervised constraint score ε^{SS} is higher than that obtained by the supervised constraint score ε^S for all databases, with the exception of the Image Segmentation, Ionosphere and Libras Movement databases. These results can be explained by the relevance of the set M^{SS} of new must-link pairwise constraints used by the score ε^{SS}. Table 4 indicates that $CNML$ is close to 0.7 for all databases except for the Image Segmentation, Ionosphere and Libras Movement databases, in which $CNML$ remains less than or equal to 0.6. It can be deduced from this table that the improvement caused by ε^{SS} compared with ε^S is dependent on the relevance of M^{SS} compared to that of M. When most of the new must-link constraints in M^{SS} are correct, our semi-supervised constraint score outperforms the supervised constraint score in the majority of cases.

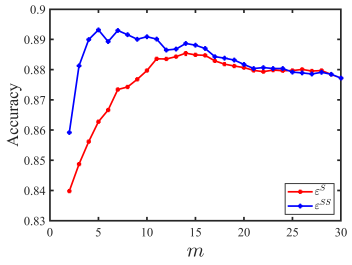
6. Conclusion

In this paper, we presented a new constraint score for feature selection in the context of both supervised and semi-supervised learning. This score evaluates a subset of features at one time, whereas state-of-the-art constraint scores evaluate only one feature at a time. This makes it possible to identify redun-

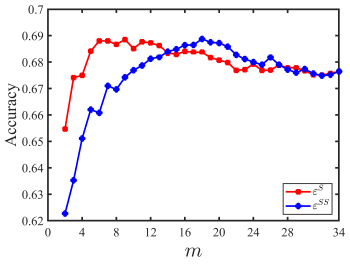


(a) WBCD

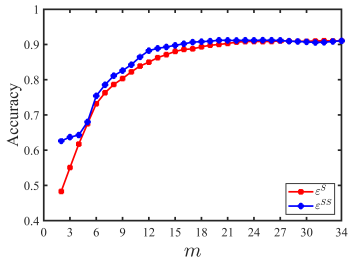
(b) Image Segmentation



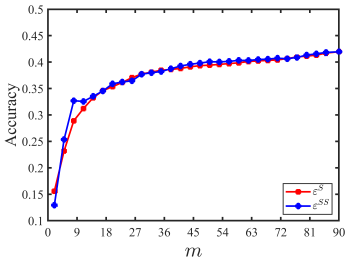
(c) WDBC



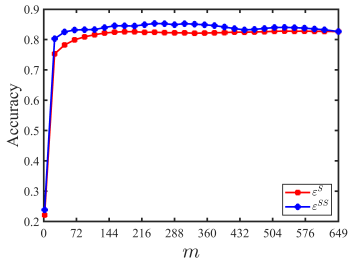
(d) Ionosphere



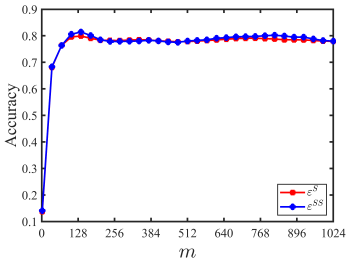
(e) Dermatology



(f) Libras Movement



(g) Multiple Features



(h) ORL

Figure 10: Comparison of accuracy rates obtained by ε^{SS} and ε^{S} scores for eight benchmark databases using nearest neighbor classifier

dant features and avoid the problem of correlation between features. Because our score evaluates the similarity between data samples in the examined feature subspace, selected features can be advantageously used by clustering.

In the context of supervised learning, our proposed score assesses the constraint-preserving ability of a feature subset. In the context of semi-supervised learning, new must-link constraints are deduced from those supplied by the user and from unlabeled data samples. When the majority of the new must-link constraints correspond to the structure of classes, they serve to improve the relevance of features selected by our score. Experiments with twelve ~~six~~ well-known benchmark databases demonstrate that in the context of both supervised and semi-supervised learning, the proposed constraint score outperforms the main state-of-the-art constraint scores and the well-established feature selection methods.

The proposed scores have the advantage of determining the number of characteristics to select. Preliminary tests have shown promising results, but more work is needed to improve them.

A possible way to increase the performance of our scores in terms of accuracy is to adaptively determine the scaling parameter σ. One solution envisaged is to proceed as for the calculation of our score, i.e., by means of the distance between the target similarity matrix defined from the given constraints and a similarity matrix that is computed with the candidate values of σ. Another possible solution for improving the relevance of selected features is to ensure of the coherence of pairwise constraints.

The proposed scores, especially in the semi-supervised context, are time-consuming. However, the computational time cannot be considered as a crucial drawback because it is always possible to reduce it. We plan to further investigate this in our future work.

In this paper, we consider that prior knowledge is represented by class prototypes from which pairwise constraints are deduced. In this case, we can compute new pairwise constraints from prototypes, and all available constraints contribute to efficient feature selection. In future work, we intend to generalize our score to prior knowledge that is formalized only by pairwise constraints. This

new score can then evaluate the relevance of each pairwise constraint provided by the user.

References

References

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

[2] R. Sheikhpour, M. A. Sarram, S. Gharaghani, M. A. Z. Chahooki, A survey on semi-supervised feature selection methods, Pattern Recognition 64 (2017) 141–158.

[3] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.

[4] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: International Conference on Machine Learning, volume 1, 2001, pp. 577–584.

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

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

[7] K. Benabdeslem, M. Hindawi, Constrained laplacian score for semi-supervised feature selection, in: Proceedings of the Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 204–218.

[8] K. Benabdeslem, M. Hindawi, Efficient semi-supervised feature selection: constraint, relevance, and redundancy, IEEE Transactions on Knowledge and Data Engineering 26 (2014) 1131–1143.

[9] X.-K. Yang, L. He, D. Qu, W.-Q. Zhang, M. T. Johnson, Semi-supervised feature selection for audio classification based on constraint compensated laplacian score, EURASIP Journal on Audio, Speech, and Music Processing 2016 (2016) 9.

[10] X.-K. Yang, L. He, D. Qu, W.-Q. Zhang, Semi-supervised minimum redundancy maximum relevance feature selection for audio classification, Multimedia Tools and Applications 77 (2018) 713–739.

[11] W. Siedlecki, J. Sklansky, On automatic feature selection, International Journal of Pattern Recognition and Artificial Intelligence 2 (1988) 197–220.

[12] M. Kalakech, P. Biela, D. Hamad, L. Macaire, Constraint score evaluation for spectral feature selection, Neural Processing Letters 38 (2013) 155–175.

[13] U. Von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (2007) 395–416.

[14] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 888–905.

[15] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, volume 18, 2005, pp. 507–514.

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

[17] M. Lichman, UCI machine learning repository, 2013. URL: `http://archive.ics.uci.edu/ml`.

[18] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, H. Liu, Advancing feature selection research, ASU feature selection repository (2010) 1–28.

[19] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Machine Learning 53 (2003) 23–69.

[20] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1226–1238.

[21] C. Li, S. Zhang, H. Zhang, L. Pang, K. Lam, C. Hui, S. Zhang, Using the k-nearest neighbor algorithm for the classification of lymph node metastasis in gastric cancer, Computational and Mathematical Methods in Medicine 2012 (2012).

[22] M. Liu, D. Zhang, Pairwise constraint-guided sparse learning for feature selection, IEEE Transactions on Cybernetics 46 (2016) 298–310.

[23] S. D. Kamvar, D. Klein, C. D. Manning, Spectral learning, in: International Joint Conference of Artificial Intelligence, Stanford InfoLab, 2003.

[24] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 1151–1157.