Efficient semi-supervised feature selection by an ensemble approach

Mohammed Hindawi[1], Haytham Elghazel[2], Khalid Benabdeslem[2]

[1] INSA de Lyon
LIRIS, CNRS UMR 5205
F-69621, France
mohammed.hindawi@insa-lyon.fr
[2] University of Lyon1
LIRIS, CNRS UMR 5205
F-69622, France
{haytham.elghazel, khalid.benabdeslem}@univ-lyon1.fr

Abstract. Constrained Laplacian Score (CLS) is a recently proposed method for semi-supervised feature selection. It presented an outperforming performance comparing to other methods in the state of the art. This is because CLS exploits both unsupervised and supervised parts of data for selecting the most relevant features. However, the choice of the little supervision information (represented by pairwise constraints) is still a critical issue. In fact, constraints are proven to have some noise which may deteriorate the learning performance. In this paper we try to override any negative effects of constraints set by the variation of their sources. This is done by an ensemble technique using both a resampling of data (bagging) and a random subspace strategy. The proposed approach generates a global ranking of features by aggregating multiple Constraint Laplacian Scores on different views of the available labeled and unlabeled data . We validate our approach by empirical experiments over high-dimensional datasets and compare it with other representative methods.

Key words: Feature selection, semi-supervised learning, constraint score, ensemble methods.

1 Introduction

In nowadays machine learning applications, data acquisition tools have well developed making it's easier to get continuously a voluminous rough data. The huge quantity of data in its turn, has a dramatically deterioration effects on both stocking and treating the data via the classical learning algorithm due to the "curse of dimensionality". In order to override this problem, feature selection has become one of the most important techniques to reduce the dimensionality. Feature selection can be defined as the process of choosing the most relevant features of data. The relevance of a feature may differ according to the learning context, which may be roughly divided into supervised, unsupervised and semi supervised feature selection.

In supervised feature selection, where all data instances are labeled, a relevance of a feature is measured according to its correlation with label information. Then a 'good' feature would be the one at which the instances with the same labels record the same (or closer) values, and vice-versa [16]. Unsupervised feature selection is considered as a much harder problem due to the absence of labels; hence the relevance of a feature is measured according to its ability in preserving some data characteristics (e.g. variance) [11]. Actually, the supervised feature selection methods outperform the unsupervised ones due to the presence of labels which represent the background knowledge about the data. However, labels availability is not always guaranteed, this is because labels -generally- require experts' intervention which is costly to obtain. Adding the aforementioned idea of rapid data acquisition tools development, a more frequent case in machine learning applications is to provide labeling information for a small part of data, then the data is called 'semi-supervised', which in its turn produces the so-called "small-labeled sample problem" [34].

In [3] we proposed Constrained Laplacian Score (CLS) as a semi-supervised scoring method which makes profit of the data structure and the label information (transformed into pairwise constraints). CLS has scored an outstanding performance towards other competitive methods. However, the method was sensible to the noise in the constraints set. To tackle this problem, we later proposed a Constrained Selection based Feature Selection framework (CSFS) [19] in which we enhanced the function score in order to be more efficient. In order to overcome the problem of noisy constraints, CSFS exploits a constraint selection process according to a coherence measure (proposed in [8]), which considers that two constraints are incoherent if they represent two contradictive powers and coherent if not. When the constraint selection is done, the remaining constraints are obviously fewer but more efficient. CSFS outperformed the results of its ancestor CLS, this could be explained by the amelioration of the scoring function and the elimination of the constraint noise. However, CSFS had two critical points : firstly, even if they are efficient, the size of selected constraint set was rather small, this has led in some cases to dramatic minimization of the constraints use feasibility. In addition, CSFS and CLS are based on the Euclidian distance between instances in the computation of feature scores, in this case the calculation of such distance becomes less reliable when data is of high dimensionality.

To overcome the two mentioned problems, we present an ensemble-based framework called EnsCLS (for Ensemble Constraint Laplacian Score) for semi-supervised feature selection. EnsCLS combines both a resampling of data (bagging) and a random selection of features (random subspaces or RSM for short) strategy. The CLS score is then used to measure features relevance on each replicate of data and the score average of all features across all ensemble components is considered. A combination of these two main strategies (bagging and RSM) for producing feature ranking, leads to an exploration of distinct views of inter-pattern relationships and allows to (i) compute robust estimates of variable importance against small changes in the pairwise constraint set, and (ii) to mitigate the curse of dimensionality.

The rest of the paper is organized as follows: Section 2 reviews recent studies on semi-supervised feature selection and ensemble methods. Section 3 briefly recalls Constraint Laplacian Score algorithm. Then we discuss the details of the proposed EnsCLS algorithm in Section 4. Experiments using relevant high-dimensional benchmarks and real datasets are presented in Section 5. Finally, we conclude this paper in Section 6.

2 Related works

In this section, we briefly present the semi-supervised feature selection and the semi-supervised ensemble approaches that appeared recently in the literature.

2.1 Feature selection

With the advent of semi-supervised feature selection, some unsupervised methods are adopted to this context by ignoring the few label information. Laplacian score [18], as example, determines a feature relevance according to the variance of data along it. The variance is an important measure of data, nevertheless, labeled data also carry valuable information and represent the background knowledge about the domain. At the other hand, another score, called constraint score [33], depends only on the few available labeling information which is transformed into constraints. Actually, constraint score proved that utilizing a few number of constraints it may perform competitively to other full labels methods (like Fisher score [13]), this had made constraint score more adaptive to the small-labeled sample problem. However, constraint score ignores the "large" unlabeled data part which carry the real data structure. In addition, the performance of constraint score is severely influenced by the choice of the constraint set. To overcome this problem, authors in [30] proposed a bagging approach (BS) to the constraint score in order to ameliorate the overall classification accuracy. The main drawback of the method is -as mentioned- the still ignorance of the unlabeled part of data which is generally far larger than the labeled one. In order to make profit of the both labeled and unlabeled parts of data, a score called C4 [23] has proposed a simple multiplication of the Laplacian and Constraint scores in order to compromise between the two scores. However, the method is biased towards the features with good Laplacian score but bad constraint score and vice-versa.

2.2 Ensemble learning

Ensemble methods have been called the most influential development in Data Mining and Machine Learning in the last decade. They combine multiple models into one usually more accurate than the best of its components. This improvement of performances relies on the concept of diversity which states that a good classifier ensemble is an ensemble in which the examples that are misclassified are different from one individual classifier to another. Dietterich [10] states that

"A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse". Many methods have been proposed to generate accurate, yet diverse, sets of models. Bagging [5], boosting [14] and Random Subspaces [20] are the most popular examples of this methodology. While Bagging obtains a bootstrap sample by uniformly sampling with replacement from original training set, boosting resamples or reweighs the training data by emphasizing more on instances that are misclassified by previous classifiers. Likewise bagging, random subspaces method (RSM) are another excellent source of obtaining diversity through feature set manipulation that provides different views of data and allows to improve the quality of classification solutions.

Recently, besides classification ensemble, there also appears clustering [29, 31] and semi-supervised learning [26, 32, 17] ensemble for which it has been shown that combining the strengths of a diverse set of clusterings or semi-supervised learners can often yield more accurate and robust solutions. Last but not least, considerable attention was paid to exploiting the power of ensemble with a view to identify and remove the irrelevant features in a supervised [6, 27], unsupervised [21, 22, 12] and semi-supervised [2] setting.

3 Constraint Laplacian Score

In this section we present a brief description of the CLS score [3] upon which we depend in our framework. In fact, CLS utilizes both parts of data, labeled and unlabeled. The labeled part is transformed into pairwise constraints, which can be classified on two subsets: Ω_{ML} (a set of Must-Link constraints) and Ω_{CL} (a set of Cannot-Link constraints)

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

Let X be a dataset of n instances characterized by p features. X consists of two subsets: X_L for labeled data and X_U for unlabeled data.

Let r be a feature to evaluate. We define its vector by $f_r = (f_{r1}, ..., f_{rn})$. The CLS of r, which should be minimized, is computed by:

$$CLS_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i \sum_{j|\exists l,(x_l,x_j)\in\Omega_{CL}} (f_{ri} - \alpha^i_{rj})^2 D_{ii}} \tag{1}$$

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

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

Algorithm 1 CLS

Require:

A data set $X(n \times p)$, which consists of two subsets: $X_L(L \times p)$, the subset of labeled training instances and $X_U(U \times p)$, the subset of unlabeled training instances; the input space ($F = \{f_1, \ldots, f_p\}$); the constant λ and the neighborhood degree k.

1: Construct the constraint sets (Ω_{ML} and Ω_{CL}) from the labeled part: X_L.
2: Calculate the dissimilarity matrix S and the diagonal matrix D.
3: **for** $r = 1$ **to** p **do**
4: Calculate CLS_r according to eq(1).
5: **end for**

where λ is a constant to be set, and x_i, x_j are neighbors means that x_i is among k nearest neighbors of x_j.

$$\alpha_{rj}^{i} = \begin{cases} f_{rj} & if\ (x_i, x_j) \in \Omega_{CL} \\ \mu_r & if\ i = j\ and\ x_i \in X_U \\ f_{ri} & otherwise \end{cases} \tag{3}$$

where $\mu_r = \frac{1}{n}\sum_i f_{ri}$ (the mean of the feature vector f_r).

CLS represents an enhanced version of both scores Laplacian [18] and Constraint-based [33]. In fact, Laplacian score can be seen as a special version of CLS when there are no labels ($X = X_U$), and when ($X = X_L$), CLS can be considered as an adjusted version of constraint score [33]. In CLS, we proposed a more efficient combination of both scores by a new score function, including the geometrical structure of unlabeled data and the constraint-preserving ability of labeled data.

With CLS, on the one hand, a relevant feature should be the one on which those two instances (neighbors or related by an ML constraint) are close to each other. On the other hand, the relevant feature should be the one with a larger variance or on which those two instances (related by a CL constraint) are well separated. We present the whole procedure of CLS in Algorithm 1.

Note that this algorithm is computed in time $O(p \times max(n^2, log\ p))$. To reduce this complexity, we proposed in our prior work [3] to apply a clustering on X_U. The idea was to substitute this huge part of data by a smaller one $X'_U = (u_1, ..., u_K)$ by preserving the geometric structure of X_U, where K is the number of clusters. We proposed to use the Self-Organizing Map (SOM) based clustering [24], for its ability to preserve the topological relationship of data well and thus the geometric structure of their distribution. With this strategy, we reduced the complexity to $O(p \times max(U, log\ p))$, where U is the size of X_U.

4 Ensemble Constraint Laplacian Score

In this section we present our ensemble based approach of constrained laplacian score for semi-supervised feature selection.

As discussed before, the most important condition for a successful ensemble learning method is to combine models which are different from each other. Thus, to maintain diversity between committee members, we have employed two strategies. Firstly, a well known ensemble method named RSM [20], is employed to face the curse of dimensionality problem by constructing multiple classifiers each one trained on different subset of examples projected on a smaller feature set RSM^i. Secondly, the diversity is further maintained, by applying the *bootstrapping method* [14].

The formal description of our approach is given in Algorithm 2. Given a set of labeled training examples X_L, and a set of unlabeled training examples X_U, described over the input space $F = \{f_1, \ldots, f_p\}$, our approach constructs a committee according to the following steps. First, as described in the steps 3 and 4 of Algorithm 2, the committee is constructed as follows : For each ensemble component i, a replication ${X_{L,b}}^i$ of the labeled data set is obtained by selecting instances from X_L with replacement and then projecting them over RSM^i, a feature subspace with m randomly selected features ($m < p$). The unlabeled data part X_U is also projected over RSM^i to generate ${X_U}^i$. Once each ensemble component i is obtained, the CLS score in Algorithm 1 is used to measure features relevance (step 6). A ranking of all features is finally obtained with respect to their average relevances over all ensemble members (steps from 7 to 9).

A single learner is known to produce very bad results as the learning algorithms break down with high-dimensional data. Ensemble learning paradigms train multiple component learners and then combine their output results. Ensemble techniques are considered as an effective solution to overcome the dimensionality problem and to improve the robustness and the generalization ability of single learners,

By using bagging in tandem with random feature subspaces, our framework try to deal with three different problems in the CLS score:

- **High dimensionality :** The major drawback of CLS was the application on high-dimensional data. This is because the Euclidian distances between examples (over all features) is an essential factor in the function score (S_{ij} in equation (1)). This makes the calculation of such distances less reliable when dealing with very high-dimensional data leading to bad features scores. Motivated by this, we adopt the use of the random manipulation strategy over the feature space (RSM). Hence, we create N random subspaces of the original features with a nearly equal apparition probability for all features. The high dimension is then reduced in each subspace and the distances calculated upon the new reduced dimension is more reliable. Consequently, working on the projected random subspace allows us to mitigate the curse of dimensionality and also help in enhancing the diversity between ensemble components.
- **Constraints :** In CLS, instance level constraints are generated directly from labels. In semi-supervised context such labels are few, then the number of constraints ($\Omega = L(L-1)/2$ where L is the number of the labeled instances)

is rather few too. Moreover, the generated constraint set may contain some noisy constraints which were proven to have deteriorate effects on the learning performance. In order to improve the positive effects of the pairwise constraints, we propose the use of bagging method on the labeled part of data in each random subspace. The bagging is made by sampling with replacement. The reason for using bagging is to enforce diversity on pairwise constraints and then to compute a robust estimation of feature score against small changes in the pairwise constraint set. Furthermore, the different bootstrap samples in different random subspaces helps in reducing the undesirable effects of the noisy constraints.

– **Unlabeled instance diversity :** The computation of CLS score implied the application of a clustering algorithm (SOM) to overcome the computation complexity of the score function. This is due to the fact that the complexity of the CLS score is highly dependent to the unlabeled part of data. Such a clustering was proved to considerably reduce this complexity. In this work, based on the random subspace approach, we keep the use of SOM algorithm in each subspace. Doing this, not only the computational complexity is reduced, but also the diversity is gained by the diversity of clusterings obtained in the different subspaces.

Algorithm 2 The EnsCLS algorithm

Require:
Set of labeled training examples (X_L); set of unlabeled training examples (X_U); input space ($F = \{f_1, \ldots, f_p\}$); committee size (N)

Initialize the scores $\mathbf{I}(f_r)$ to zero for each feature r
for $i = 1 : N$ **do**
  RSM^i = randomly draw m features from F
  ${X_{L,b}}^i$ = bootstrap sample from X_L projected onto RSM^i
  ${X_U}^i$ = the unlabeled sample X_U projected onto RSM^i
  $imp^i = CLS({X_{L,b}}^i, {X_U}^i)$ compute the constraint laplacian score of each feature in RSM^i using Algorithm 1
  for each feature $r \in RSM^i$ **do**
    $\mathbf{I}(f_r) = \mathbf{I}(f_r) + \frac{imp^i(f_r)}{N}$
  end for
end for
rank the features in F according to their scores $\mathbf{I}$ in ascending order.
return F

5 Experimental results

In this section, we provide empirical results on several benchmark and real high-dimensional datasets and compare EnsCLS against over state-of-the-art semi-

Table 1. The datasets used in the experiments

Dataset	# patterns	# features	# classes	Reference
BasesHock	1993	4862	2	[36]
Leukemia	73	7129	2	[15]
Lymphoma	96	4026	9	[1]
Madelon	2598	500	2	[4]
PcMac	1943	3289	2	[36]
PIE10P	210	2420	10	[36]
PIX10P	100	10000	10	[36]
Prostata	102	12533	2	[28]
Relathe	1427	4322	2	[36]

supervised feature ranking algorithms. EnsCLS is compared with four other feature selection methods: (1) the original CLS score [3], (2) the Constrained Selection based Feature Selection framework (CSFS) [19], two ensemble-based feature evaluation algorithms, including (3) the Bagging constraint Score (BS) [30], and (3) the wrapper-type Semi-Supervised Feature Importance approach (SSFI) [2]. Nine benchmark and real labeled datasets were used to assess the performance of feature selection algorithms. They are described in Table 1. We selected these datasets as they contain thousands features and are thus good candidates for feature selection. Most of these datasets have already been used in various empirical studies [35, 2] and cover different application domains: Biology, image and text analysis.

5.1 Evaluation framework

To make fair comparisons, the same experimental settings in [3] was adopted here for CLS and CSFS approaches, *i.e.*, the neighborhood graph with a neighborhood degree of 10, and the λ value is set to 0.1. For BS, we set the ensemble size to 100, as around this value the quality of this method is less insensitive to the increase of the ensemble size (*c.f.* [30]). EnsCLS and SSFI are tuned similarly. The number of features per bag is $m = \sqrt{p}$, where p is the size of the input space. The committee size N is computed using the following formula:

$$N = 10 \times ceil\left(\frac{\log(0.01)}{\log(1 - 1/\sqrt{p})}\right). \tag{4}$$

This formula ensures that each feature is drawn ten times at a confidence level of 0.01. Furthermore, as suggested by the authors in [2], the number of iterations *maxiter* and the sample size n in SSFI are set to 10, and 1, respectively.

For each dataset, experimental results are averaged over 10 runs. At each run, the whole dataset is splitted (in a stratified way) into a training partition with 2/3 of the observations and a test partition with the remaining 1/3 observations. Training set is further splitted into labeled and unlabeled datasets. As in [35],

Algorithm 3 Feature Evaluation Framework

for each dataset X **do**
  build a randomly stratified partition (Tr, Te), from X where $|Tr| = \frac{2}{3}.|X|$ and $|Te| = \frac{1}{3}.|X|$;
  Generate labeled data X_L by randomly sampling from Tr 3 instances per class;
  $X_U = Tr \backslash X_L$;
  SF_{CLS} = Apply *CLS* with $X_L \cup X_U$;
  SF_{CSFS} = Apply *CSFS* with $X_L \cup X_U$;
  SF_{BS} = Apply *BS* with $X_L \cup X_U$;
  SF_{SSFI} = Apply *SSFI* with $X_L \cup X_U$;
  SF_{EnsCLS} = Apply *EnsCLS* with $X_L \cup X_U$;
  for $i = 1$ *to* 20 **do**
    Select top i features from SF_{CLS}, SF_{CSFS}, SF_{BS}, SF_{SSFI} and SF_{EnsCLS};
    $Tr_{CLS} = \Pi_{SF_{CLS}}(Tr)$;
    $Tr_{CSFS} = \Pi_{SF_{CSFS}}(Tr)$;
    $Tr_{BS} = \Pi_{SF_{BS}}(Tr)$;
    $Tr_{SSFI} = \Pi_{SF_{SSFI}}(Tr)$;
    $Tr_{EnsCLS} = \Pi_{SF_{EnsCLS}}(Tr)$;
    Train the *Baselearner* using Tr_{CLS}, Tr_{CSFS}, Tr_{BS}, Tr_{SSFI} and Tr_{EnsCLS} and record accuracy obtained on Te;
  end for
end for

the labeled sample set X_L consists of randomly selected 3 patterns per class, and the remaining patterns are used as unlabeled sample set X_U. In order to assess the quality of a feature subset obtained with the aforementioned semi-supervised procedures, we train a SVM classifier (using LIBSVM package [7]) on the whole labeled training data and evaluate its accuracy on the test data. The latter is taken as the score for the feature subset. The details of the evaluation framework are shown in Algorithm 3. As mentioned above, the process specified in Algorithm 3 is repeated 10 times. The obtained accuracy is averaged and used for evaluating the quality of the feature subset selected according to each algorithm.

5.2 Results

In Figure 1, we plotted the accuracies of the above feature selection approaches against the 20 most important features. As may be observed, EnsCLS outperforms the other four methods by a noticeable margin. The major observations from the analysis of these plots are three-fold:

– EnsCLS usually has better performances than CLS and CSFS. This firstly validates the motivation behind our method EnsCLS that ensemble strategy

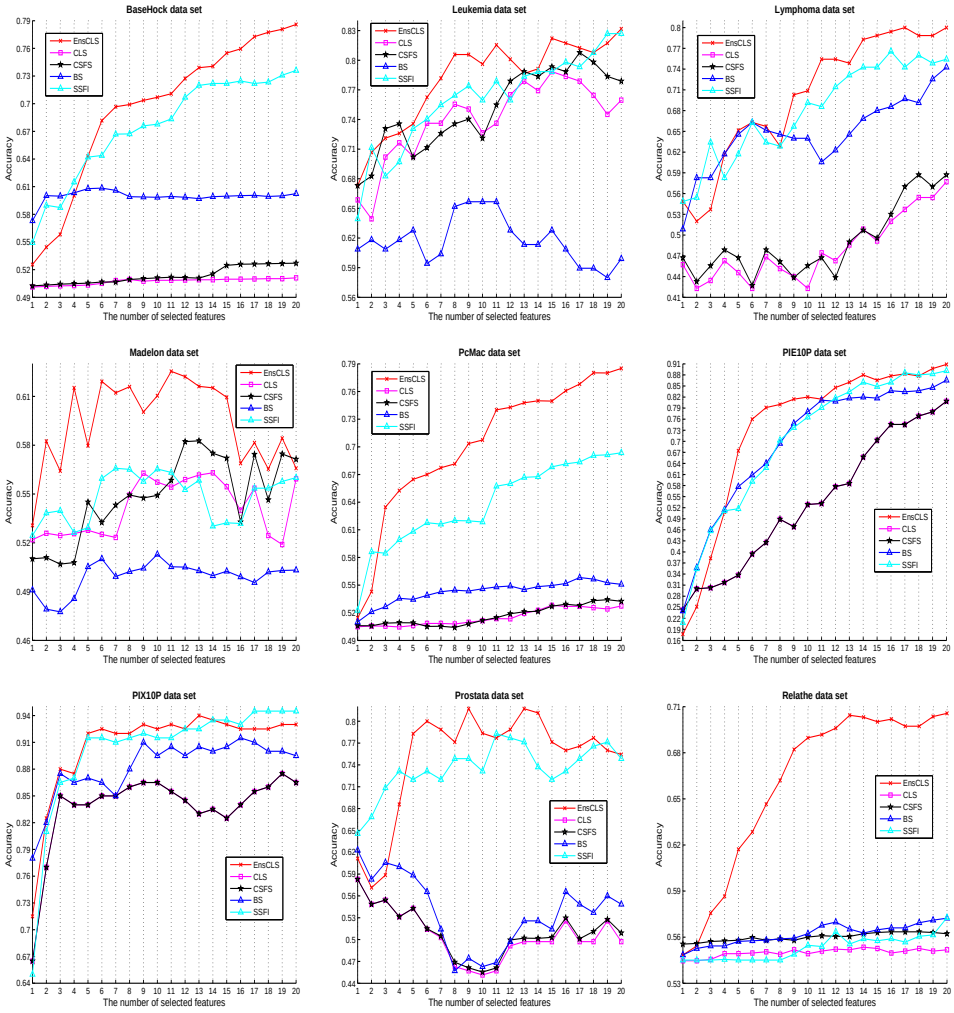

Fig. 1. Accuracy vs. different numbers of selected features. The number of labeled instances per class is set to 3.

has the potential to improve the quality and the stability of the CLS score and also confirms the effectiveness of this ensemble strategy to rank the features properly, compared to the powerful constraint selection method used in CSFS.

– EnsCLS seems to combine more efficiently the labeled and unlabeled data for feature evaluation and it shows promise for scaling to larger domains in a semi-supervised way in view of the good performance on BaseHock, PcMac, Madelon and Relathe datasets. This suggests the ability of the proposed

Table 2. Mean and standard deviations of accuracy over the 20 most important features. Bottom row of the table present average rank of accuracy mean used in the computation of the Friedman test.

Data	**EnsCLS**	**CLS**	**CSFS**	**BS**	**SSFI**
BaseHock	0.695±0.01	0.507±0.00	0.513±0.01	0.600±0.05	0.675±0.03
Leukemia	0.781±0.06	0.740±0.10	0.751±0.11	0.618±0.04	0.760±0.08
Lymphoma	0.702±0.03	0.480±0.03	0.490±0.03	0.647±0.04	0.680±0.06
Madelon	0.594±0.01	0.542±0.04	0.549±0.03	0.499±0.01	0.548±0.04
PcMac	0.703±0.01	0.515±0.01	0.517±0.01	0.543±0.03	0.638±0.02
PIE10P	0.734±0.07	0.535±0.04	0.535±0.04	0.696±0.11	0.701±0.07
PIX10P	0.907±0.03	0.837±0.05	0.837±0.05	0.882±0.03	0.902±0.03
Prostata	0.749±0.04	0.507±0.02	0.511±0.02	0.538±0.08	0.735±0.10
Relathe	0.660±0.01	0.550±0.00	0.560±0.00	0.562±0.02	0.553±0.00
Av Rank	1.0000	4.6667	3.6667	3.3333	2.3333

ensemble method of CLS to rank the relevant features accurately, compared to especially the other ensemble semi-supervised feature selection approaches (BS and SSFI), by exploiting efficiently the topological information from the unlabeled data.

– A closer inspection of the plots reveals that the accuracy on the features selected by EnsCLS generally increases swiftly at the beginning (the number of selected feature is small) and slows down afterwards. This suggests that EnsCLS ranks the most relevant features first and that a classifier can achieve a very good classification accuracy with the top 5 features while the other methods require more features to achieve comparable results.

Fore sake of completeness, we also averaged the accuracy for different numbers of selected features. The averaged accuracies of EnsCLS and the other methods over the top 20 features are depicted in Table 2. In order to better assess the results obtained for each algorithm, we adopt in this study the methodology proposed by [9] for the comparison of several algorithms over multiple datasets. In this methodology, the non-parametric Friedman test is firstly used to evaluate the rejection of the hypothesis that all the classifiers perform equally well for a given risk level. It ranks the algorithms for each dataset separately, the best performing algorithm getting the rank of 1, the second best rank 2 etc. In case of ties it assigns average ranks. Then, the Friedman test compares the average ranks of the algorithms and calculates the Friedman statistic. If a statistically significant difference in the performance is detected, we proceed with a *post hoc* test. The Nemenyi test is used to compare all the classifiers to each other. In this procedure, the performance of two classifiers is significantly different if their average ranks differ more than some critical distance (CD). The critical distance depends on the number of algorithms, the number of data sets and the critical value (for a given significance level p) that is based on the Studentized range statistic (see [9] for further details). In this study, based on the values in Table 2,

the Friedman test reveals statistically significant differences ($p < 0.05$) between all compared approaches.

Furthermore, we present the result from the Nemenyi posthoc test with average rank diagrams as suggested by Demsar [9]. These are given on Figure 2. The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the rightmost side of the diagram. The algorithms that do not differ significantly (at $p = 0.1$) are connected with a line. The critical difference CD is shown above the graph (here CD=1.8336).

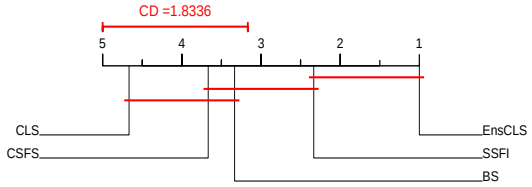

Fig. 2. Average ranks diagram comparing the feature selection algorithms in terms of accuracy over different number of selected features.

Overall, EnsCLS performs best. However, its performances are not statistically distinguishable from the performances of SSFI. Another interesting observation upon looking at the average rank diagrams and Table 2 is that, almost in all cases the ensemble methods, *i.e.* EnsCLS, SSFI and BS, achieve better performances than those of single methods including CLS and CSFS, respectively.

The statistical tests we use are conservative and the differences in performance for methods within the first group (EnsCLS and SSFI) are not significant. To further support these rank comparisons, we compared, on each dataset and for each pair of methods, the accuracy values in Table 2 using the paired t-test (with $p = 0.1$). The results of these pairwise comparisons are depicted in Table 3 in terms of "Win-Tie-Loss" statuses of all pairs of methods; the three values in each cell (i, j) respectively indicate how times many the approach i is significantly better/not significantly different/significantly worse than the approach j. Following [9], if the two algorithms are, as assumed under the null-hypothesis, equivalent, each should win on approximately $n/2$ out of n data sets. The number of wins is distributed according to the binomial distribution and the critical number of wins at $p = 0.1$ is equal to 7 in our case. Since tied matches support the null-hypothesis we should not discount them but split them evenly between the two classifiers when counting the number of wins; if there is an odd number of them, we again ignore one.

In Table 3, each pairwise comparison entry (i, j) for which the approach i is significantly better than j is boldfaced. From this table, the analogous trend between EnsCLS and other feature selection methods can be observed as in Table

Table 3. Pairwise t-test comparisons of FS methods in terms of accuracy. Bold cells (i, j) highlights that the approach i is significantly better than j according to the sign test at $p = 0.1$.

	EnsCLS	**CLS**	**CSFS**	**BS**	**SSFI**
EnsCLS	–	**8/1/0**	**8/1/0**	**8/1/0**	**5/4/0**
CLS	0/1/8	–	2/3/4	2/2/5	0/3/6
CSFS	0/1/8	4/3/2	–	2/2/5	1/2/6
BS	0/1/8	5/2/2	5/2/2	–	0/2/7
SSFI	0/4/5	**6/3/0**	**6/2/1**	**7/2/0**	–

2 and Figure 2, *i.e.*, EnsCLS and SSFI usually have better performances than all other methods. On the other hand, It can be seen from Table 3 that EnsCLS significantly outperforms SSFI.

6 Conclusion

Constraint Laplacian Score (CLS) which uses pairwise constraints for feature selection has shown good performance in our previous work [3]. However, one important problem of such approach is how to best use the available constraints for dealing with low-quality ones that may deteriorate the learning performance. Instead of making efforts on choosing constraints for single feature selection, as recently done in the CSFS approach [19], we address, in this paper, this important issue from another view. We propose a novel semi-supervised feature selection method called Ensemble Laplacian Constraint Score (EnsCLS for short), which firstly combines both data resampling (bagging) and random subspace strategies for generating different views of the data. Once each ensemble component is obtained, the CLS score is used to measure features relevance. A ranking of all features is finally obtained with respect to their average relevances over all ensemble members.

Extensive experiments on a series of benchmark and real datasets have verified the effectiveness of our approach compared to other state-of-the-art semi-supervised feature selection algorithms and confirm the ability of the used ensemble strategy to rank the relevant features accurately. They also show that the proposed EnsCLS method can utilize labeled and unlabeled data in a more effective way than Constraint Laplacian Score. Furthermore, they indicate that our method which inject some randomness for manipulating the available unlabeled and labeled data (constraints) is superior to the recently proposed CSFS method which actively selects constraints to improve the quality of the CLS score.

Future substantiation through more experiments on biological databases containing several thousands of variables and through evaluating the stability of the feature selection method [25, 27] when small changes are made to the data are currently being undertaken. Moreover, comparisons using different numbers of pairwise constraints will be reported in due course.

References

1. A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.
2. Hasna Barkia, Haytham Elghazel, and Alex Aussem. Semi-supervised feature importance evaluation with ensemble learning. In *ICDM*, pages 31–40, 2011.
3. K. Benabdeslem and M. Hindawi. Constrained laplacian score for semi-supervised feature selection. In *Proceedings of ECML-PKDD conference*, pages 204–218, 2011.
4. C.L Blake and C.J Merz. Uci repository of machine learning databases, 1998.
5. L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.
6. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
7. Chih. Chung Chang and Chih. Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
8. I. Davidson, K. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proceedings of ECML/PKDD*, 2006.
9. Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
10. T.G. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classier Systems*, pages 1–15, 2000.
11. J.G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, (5):845–889, 2004.
12. Haytham Elghazel and Alex Aussem. Unsupervised feature selection with ensemble learning. *Machine Learning*, pages 1–24, 2013.
13. R. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen*, 7:179–188, 1936.
14. Y Freund and R.E. Shapire. Experiments with a new boosting algorithm. In *13^{th} International Conference on Machine Learning*, pages 276–280, 1996.
15. T.R. Golub, Slonim, D.K., P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, and H. Coller. Molecular classication of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
16. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, (3):1157–1182, 2003.
17. M. F. Abdel Hady and F. Schwenker. Combining committee-based semi-supervised learning and active learning. *Journal of Computer Science and Technology*, 25(4):681–698, 2010.
18. X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems, 17*, 2005.
19. M. Hindawi, K. Allab, and K. Benabdeslem. Constraint selection based semi-supervised feature selection. In *Proceedings of international conference on Data Mining*, pages 1080–1085, 2011.
20. Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.
21. Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Consensus unsupervised feature ranking from multiple views. *Pattern Recognition Letters*, 29(5):595–602, 2008.

22. Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742–2756, 2008.
23. M. Kalakech, P. Biela, L. Macaire, and D. Hamad. Constraint scores for semi-supervised feature selection: A comparative study. *Pattern Recognition Letters*, 32(5):656–665, 2011.
24. T. Kohonen. *Self organizing Map*. Springer Verlag, Berlin, 2001.
25. Ludmila I. Kuncheva. A stability index for feature selection. AIAP'07, pages 390–395, 2007.
26. M. Li and Z. H. Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1088–1098, 2007.
27. Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *ECML/PKDD (2)*, pages 313–325, 2008.
28. Dinesh Singh, Phillip G. Febbo, Kenneth Ross, Donald G. Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. D'Amico, and Jerome P. Richie. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.
29. A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
30. Dan Sun and Daoqiang Zhang. Bagging constraint score for feature selection with pairwise constraints. *Pattern Recognition*, 43:2106–2118, 2010.
31. A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.
32. Y. Yaslan and Z. Cataltepe. Co-training with relevant random subspaces. *Neurocomputing*, 73(10-12):1652–1661, 2010.
33. D. Zhang, S. Chen, and Z. Zhou. Constraint score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, 41(5):1440–1451, 2008.
34. Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 641–646, 2007.
35. Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *SDM*, pages 641–646, 2007.
36. Zheng Zhao, Fred Morstatter, Shashvata Sharma, Salem Alelyani, and Aneeth Anand. Feature selection, 2011.