

Semi-Supervised Feature Importance Evaluation with Ensemble Learning

Hasna Barkia*, Haytham Elghazel* and Alex Aussem*

**Université de Lyon, 69000, Lyon, France; Université de Lyon 1, Laboratoire GAMA, EA 4608, 69622 Villeurbanne.*

*Email: hasna.barkia@etu.univ-lyon1.fr; haytham.elghazel@univ-lyon1.fr; alex.aussem@univ-lyon1.fr
http://gama.univ-lyon1.fr/MLKD*

Abstract—We consider the problem of using a large amount of unlabeled data to improve the efficiency of feature selection in high dimensional datasets, when only a small set of labeled examples is available. We propose a new semi-supervised feature importance evaluation method (SSFI for short), that combines ideas from co-training and random forests with a new permutation-based out-of-bag feature importance measure. We provide empirical results on several benchmark datasets indicating that SSFI can lead to significant improvement over state-of-the-art semi-supervised and supervised algorithms.

Keywords-Feature Selection, Semi-Supervised Learning, Ensemble Method, Co-training, Bagging, Random Subspaces Method

I. INTRODUCTION

The identification of relevant subsets of random variables among thousands of potentially irrelevant and redundant variables is a challenging topic of pattern recognition research that has attracted much attention over the last few years. In supervised learning, feature selection algorithms use only information from labeled data to find the relevant subsets of variables, i.e., those that conjunctively prove useful to construct an efficient classifier from data. It enables the classification model to achieve good or even better solutions with a restricted subset of features [1], [2], [3]. However, in many real-world applications, the amount of labeled data is very limited and it becomes difficult to identify and remove the redundant and irrelevant variables from the feature set, especially in high dimension. This situation arises naturally in many real-world applications, where large amount of data can be collected cheaply and automatically, but when manual labeling of samples remains extremely time consuming and/or cannot be taken for granted. In this case, unsupervised feature selection methods could be envisaged to exploit the information conveyed by the a large amount of unlabeled training data [4], [5], [6], [7]. Broadly speaking, the feature selection in unsupervised learning aims at finding relevant subsets of variables that produce "natural" groupings by grouping "similar" objects together based on some similarity measure. Clearly, the combination of both paradigms (supervised and unsupervised) allows the merging of sophisticated semi-supervised approaches that

can handle both labeled and unlabeled data. The problem of semi-supervised feature selection has attracted a great deal of interest recently and its effectiveness has already been demonstrated in many applications [8], [9], [10], [11].

On the other hand, databases have increased many fold in recent years. Important recent problems (i.e., DNA data in biology) often have the property that there are hundreds or thousands of features, with each one containing only a small amount of information. 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, and therefore has been a hot topic during the past years. Although considerable attention has been given on the problem of constructing an accurate and diverse ensemble committee for supervised and unsupervised learning, and using this committee to estimate the feature importance [12], [6], [5], little attention has been given to exploiting the power of ensemble with a view to identify and remove the irrelevant features in a semi-supervised setting.

The way internal estimates are used to measure variable importance in the Random Forests (RF) paradigm [12] have been influential in our thinking. In this study, we show that these ideas are also applicable to semi-supervised feature selection. We propose a novel semi-supervised feature importance evaluation method termed SSFI as a shorthand. The algorithm ranks features through an ensemble framework, in which a feature's relevance is evaluated by its predictive accuracy using both labeled and unlabeled data. SSFI combines both data resampling (*bagging*) and random subspace strategies for generating an ensemble learner using a co-training style algorithm. A combination of these two main strategies for producing ensemble of classifiers leads to exploration of distinct views of inter-pattern relationships. Once each ensemble member is obtained, an extension of the RF permutation importance measure [12], using the labeled and unlabeled data together, is proposed to measure feature's relevance. A ranking of all features is finally obtained with

respect to their relevances in all obtained semi-supervised classifiers.

The rest of the paper is organized as follow: Section 2 reviews recent studies on semi-supervised feature selection and ensemble methods. Section 3 introduces the SSFI framework and describes how variable importance used in RF can be extended in semi-supervised context by using both labeled and unlabeled data. Experiments using relevant high-dimensional benchmarks and real datasets are presented in Section 4.

II. RELATED WORK

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

A. Semi-Supervised feature selection

The key for designing an effective semi-supervised feature selection algorithm is to develop a framework, under which the relevance of a feature can be evaluated by both labeled and unlabeled data in a natural way. Recently, several studies have focused on semi-supervised feature selection. Like in supervised and unsupervised FS, these methods can be divided into three categories, depending on how they interact with the learning algorithm: *filter*, *wrapper* and *embedded* approaches. *Filter methods* discover the relevant and redundant features through analyzing the correlation and dependence among features without involving any learning algorithms [10], [11]. The most common filter strategies are based on feature ranking. Feature ranking is a relaxed version of feature selection which ranks all features with respect to their relevances and chooses the top ranked features as the working feature vector manually. Therefore, feature ranking can be viewed as a kind of flexible feature subset selection approach. Feature ranking has been well studied for semi-supervised classification. Zhao et al. [10] proposed a semi-supervised feature ranking algorithm, referred to as *Sselect*, based on the spectral graph theory. Their method first constructs a neighborhood graph using original data, and then evaluates each feature vector by transforming it into a cluster indicator and checking whether it is consistent with label information. It has demonstrated promising results on some benchmark datasets. In [11], a semi-supervised feature selection algorithm, called Locality Sensitive Discriminant Feature (LSDF) was proposed. Unlike Fisher score which makes use of only labeled data points and Laplacian score which makes use of only unlabeled data points, the proposed algorithm makes use of both labeled and unlabeled data points. It tries to discover both geometrical and discriminant structure in the data. using two graphs, i.e., within-class graph and between-class graph. The within-class graph connects data points which share the same label or are sufficiently close to each other, while the between-class graph connects data points which are sufficiently close to

each other but have different labels. The importance of the features is characterized by its degree of preserving these graph structures. Specifically, a feature is considered as "good" if at this dimension nearby points, or points sharing the same label, are close to each other, while points with different labels are far apart. However, the presence of a large amount of irrelevant features often leads to inexact neighborhood mapping and causes both aforementioned methods to fail [8].

On the other hand, *wrapper methods* perform a search in the space of feature subsets, guided by the outcome of the learning model. Typically, a criterion is firstly defined for evaluating the quality of a candidate feature subset and wrapper approaches aim to identify a feature subset such that the learning algorithm trained on this feature subset can achieve the optimal value of the predefined criterion. In [9], a forward search based semi-supervised feature ranking method is proposed. It uses the mechanism of random selection on unlabeled data to form new training sets, and the most frequently selected feature, using supervised sequential forward search strategy, is added to the result feature subset in each iteration. In this method, the subset of features derived from the random training sets used may not be adequate, but once the feature is chosen, it will never be eliminated.

In contrast to filter and wrapper approaches, the search for an optimal subset of features with *embedded methods* is built into the model construction making these techniques specific of a given learning algorithm. Recently, Zenglin et al. [13] proposed a semi-supervised feature selection method that works in an embedded way. The feature selection process is integrated to the semi-supervised classifier by taking advantage of manifold regularization. In the proposed method, an optimal subset of features is identified by maximizing a performance measure that combines classification margin with manifold regularization. The manifold regularization in the proposed feature selection method assures that the decision function is smooth on the manifold constructed by the selected features of the unlabeled data.

B. Semi-supervised ensemble learning

Semi-supervised learning has been widely applied in many real-world application domains such as medical diagnosis, fraud detection and pattern recognition. Semi-supervised learning methods are used in order to make use of unlabeled data in addition to the labeled data for better classification. According to the feature spaces used, semi-supervised learning (SSL) algorithms can be divided into single-view and multiple-view algorithms. One of the most successful single-view algorithms is the Self-Training algorithm in which a single classifier is initially trained using a small amount of labeled data. Then it adds the most confident unlabeled data incrementally into the labeled dataset and retrains the underlying classifier with

the augmented training set. On the other hand, Co-training is one of the most attractive multi-view SSL algorithms. Introduced by Blum and Mitchell in [14], in Co-training two classifiers are initially trained using two redundant and independent sets of features (views). Then in each further iteration, each classifier classifies the unlabeled examples, adds the examples about which it is most confident into the training set. The aim is that the most confident examples with respect to one classifier can be informative with respect to the other. Although co-training has emerged as a powerful method in some fields, the requirement on two sufficient and redundant attribute subsets is too strong to be met in many real-world applications. Therefore, many extensions of co-training have been proposed in the literature to deal with this problem. The proposed alternatives are generally ensemble-based and differ on the strategy they used to generate component classifiers. Methods for constructing ensembles include manipulation of the training samples by resampling (*bootstrap aggregation or bagging*) [15], [16], [17], [18], [19], [20] or using *random subspaces* [15], [19], [21], [22].

In [18], an ensemble co-training style method named Co-Forest is proposed. It extends the co-training paradigm by incorporating a well-known ensemble learning algorithm named Random Forest [12] to tackle the problems of how to determine the most confident examples to label and how to produce the final hypothesis. Co-Forest uses bootstrap sample data from training set and trains random trees. At each iteration each random tree is reconstructed by newly selected examples for its concomitant ensemble. Furthermore, [16] gives another extension of the usage of RF to semi-supervised learning problems. In order to incorporate unlabeled data, the main idea consists to use the predicted labels of the unlabeled data as additional optimization variables. The authors in [16] perform an iterative deterministic annealing-style training algorithm maximizing both the multi-class margin of labeled and unlabeled samples.

Another ensemble semi-supervised learning approach is given by the work in [15] named Co-training By Committee (CoBC). In this work, an ensemble of diverse classifiers is used instead of redundant and independent views. The committee of diverse accurate classifiers is initially constructed by using a successful ensemble learning algorithms: Bagging or random subspace method. At each iteration and for each classifier, a subset of unlabeled examples are drawn randomly from the whole unlabeled dataset and classified using the concomitant ensemble. The most confident examples to label are then determined and the committee members are retrained using their updated training sets.

It should be noted that all extensions of Co-training that requires bootstrapping may need a lot of labeled samples in order to be successful. For high dimensional datasets, the classifiers trained on small bootstrapped data samples using single feature view may face the "large p, small n problem" (the size of the training set is much smaller than the number

of dimensions in the feature vector) and, thus, may cause an overfitting problem.

As a solution, *random subspace methods* (RSM) are one of the successful methods used for producing an ensemble of classifiers and dealing with high dimensional datasets. RASCO [21] algorithm combines the ideas of Co-training and random subspace methods. Instead of using two feature subspaces, it uses random feature splits in order to train different classifiers. The unlabeled data samples are labeled and added to the training set based on the combination of decisions of the classifiers trained on different feature splits. The intuition behind this is that each classifier can complement another one. RASCO has been shown to perform better than Co-training method. In [22], instead of totally random feature subspaces, the authors propose to produce relevant random subspaces by means of drawing features with probabilities proportional to their relevances measured by the mutual information between features and class labels. The results obtained on different datasets show that the proposed algorithm, termed as Rel-RASCO, outperforms both RASCO and Co-training methods.

Another similar semi-supervised learning approach to RASCO, that uses support vector machines, was proposed to be used for content based image retrieval [19]. Authors in [19] propose to use bagging and random subspace strategy in the same framework since they are especially effective when the original classifier is not very stable and can generate more diversified classifiers.

III. THE METHOD

In this section, we discuss our semi-supervised feature importance evaluation method, that combines ideas from co-training and RF with a new permutation-based out-of-bag feature importance measure.

A. Committee construction

As discussed before, the most important condition for a successful ensemble learning method is to combine models which are different from each other, i.e. that make error on different training examples. Thus, to maintain diversity between committee members, we have employed two strategies. Firstly, a well known ensemble method named *RSM*, 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*. The formal description of our approach is given in Algorithm 1. Given a set of labeled training examples L , and a set of unlabeled training examples U , described over the input space $F = \{f_1, \dots, f_p\}$, our approach constructs a committee according to the following steps.

First, as described in the steps from 3 to 11 of the Algorithm 1, the initial committee is constructed as follows :

Algorithm 1 $SSFI(L, U, F, K, N, n, maxiter, BaseLearn)$

Require:

set of labeled training examples (L), set of unlabeled training examples (U), input space ($F = \{f_1, \dots, f_p\}$), number of classes (K), committee size (N), sample size (n), maximum number of iterations ($maxiter$) and base learning algorithm ($BaseLearn$)

- 1: Get the class prior probabilities, $\{Pr_k\}_{k=1}^K$
- 2: Set the class growth rate, $n_k = n \times Pr_k$ where $k = 1, \dots, K$

Initial committee construction H

- 3: $H = \emptyset$
- 4: **for** $i = 1 : N$ **do**
- 5: RSM^i = randomly draw m features from F
- 6: L_{bag}^i = bootstrap sample from L projected onto RSM^i
- 7: U_{bag}^i = bootstrap sample from U projected onto RSM^i
- 8: $L_{oob}^i = L \setminus L_{bag}^i$, $U_{oob}^i = U \setminus U_{bag}^i$
- 9: $h^i = BaseLearn(L_{bag}^i)$
- 10: $H = H \cup h^i$
- 11: **end for**

Committee refinement using SSL ensemble method

- 12: $t = 1$
- 13: **repeat**
- 14: **for** each $h^i \in H$ **do**
- 15: $\pi^i = SelectConfidentExamples(i, H, U_{bag}^i, \{n_k\}_{k=1}^K)$
- 16: $L_{bag}^i = L_{bag}^i \cup \pi^i$, $U_{bag}^i = U_{bag}^i \setminus \pi^i$
- 17: $h^i = BaseLearn(L_{bag}^i)$
- 18: **end for**
- 19: $t = t + 1$
- 20: **until** ($t > maxiter$ **OR** no committee member changes)

Feature relevance estimate

- 21: $imp = 0$
 - 22: **for** each $h^i \in H$ **do**
 - 23: $[O_{data}^i, O_{label}^i, O_{ccnf}^i] = BuildOOBMatrix(i, H, L_{oob}^i, U_{oob}^i, K)$
 - 24: **for** each $f \in RSM^i$ **do**
 - 25: randomly permute the values of f over the O_{data}^i examples to form O_{perm}^i
 - 26: **for** each $x \in O_{perm}^i$ **do**
 - 27: **if** ($h^i(x) \neq O_{label}^i(x)$) **then**
 - 28: $imp(f) = imp(f) + O_{con,f}^i(x)$
 - 29: **end if**
 - 30: **end for**
 - 31: **end for**
 - 32: **end for**
 - 33: rank the features f according to $imp(f)$
 - 34: **return** F and imp
-

For each committee member h^i , L_{bag}^i and U_{bag}^i are selected with replacement, from L and U respectively, and projected over RSM^i , a feature subspace with m randomly selected features ($m < p$). Then, each component h^i is constructed according to a given *baseLearner* based on its corresponding labeled training examples L_{bag}^i .

Second, according to the steps from 12 to 20 in the Algorithm 1, the co-training method trains each h^i , by asking a subset of the concomitant classifiers to label examples from U_{bag}^i for it, then a set π^i , which consists of the n_k most confident examples assigned to each class k , is removed from U_{bag}^i , and incrementally added into L_{bag}^i . Then a new h^i is retrained over the augmented set L_{bag}^i . A formal description is given in the Algorithm 2, to describe how the most confident examples are selected.

The co-training steps are repeated until a maximal number of iteration is reached or the committee is no longer changing.

B. Confidence Measure

An important factor that affects the performance of any Co-Training style algorithm is how to measure the confidence about the labeling of an unlabeled example which determines its probability of being selected. An inaccurate confidence measure leads to adding mislabeled examples to the labeled training set which leads to performance degradation during the SSL process.

In the Algorithm 2, a formal description is given, to explain how the most confident examples are selected. In order to improve the accuracy of a committee member h^i , its unlabeled examples, U_{bag}^i will be labeled by the other components. More specifically, for a given unlabeled example x , let H_x be the concomitant ensemble of h^i , which contains only members where x is out of bag. In order to guarantee the consistence of the learning process and an accurate labeling for unlabeled data, we have chosen to label a given unlabeled example x , only by the members h^j of its corresponding H_x . Thus, a given example x , in a given iteration t , will have the same label for all the committee members $h^i \in H$, where x is $\in U_{bag}^i$.

Then, for each unlabeled example $x \in U_{bag}^i$, each committee member $h^j \in H_x$, will label it, in order to generate the class probability distribution for the given x . Then a majority voting method is applied over H_x , in order to attribute the final class label of x : As described in the Algorithm 3, each classifier from H_x is asked to label x , in order to generate the class probability distribution for the given x . Then the class which receives the maximal votes, is assigned to the example x , with a label confidence equal to the degree of agreement on the labeling, i.e. the number of classifiers that agree on the label assigned by H_x .

C. Out-of-bag based feature relevance measure

In our approach, the Random subspace method is combined to bootstrapping. Actually, in each bootstrapped labeled and unlabeled set, almost 33% are left oob, i.e., they are not used for the construction of the corresponding model. We refer to them as U_{oob}^i and L_{oob}^i . Thus, these patterns can be used to estimate non biased feature relevancies. The first step consists to build the Out Of Bag information structure $O^i = [O_{data}^i, O_{label}^i, O_{cnf}^i]$ as described in the Algorithm 4. For each classifier, we select the well predicted instances from L_{oob}^i and U_{oob}^i using h^i to form the set O^i . Clearly, for the labeled examples, an example is well predicted, if the class label given by h^i corresponds to the real label. Its label confidence is set to 1. For the unlabeled examples, the right label is unknown. Also, the key idea is to assume that an unlabeled example x is “well labeled” by h^i , if the label given by h^i is the label given by the majority vote given by the committee H_x . In that case, its label confidence will be set to the degree of agreement for winning label among the members of H_x . Second, the values of the f^{th} feature in the O_{data}^i , are randomly permuted to form O_{perm}^i , and h^i is used to predict the label of the new Out Of Bag patterns. The procedure is repeated for every feature $f \in \{f_1, \dots, f_p\}$. At the end of the run, the sum of all the example’s confidence for which the predicted label in the O_{perm}^i differs from the initial predicted label in the initial O_{data}^i , is computed. The latter value is averaged over N , i.e., the committee size. The resulting value is taken as the importance of the feature f . The key idea in our approach is the use of label’s confidence in the evaluation of the feature importance. So the unlabeled examples play an important role in the feature importance evaluation.

Algorithm 2 *SelectConfidentExamples*($i, H, U_{bag}^i, \{n_k\}_{k=1}^K$)

Require:

- a committee (H), current committee member index (i), pool of unlabeled examples (U_{bag}^i), growth rate ($\{n_k\}_{k=1}^K$) and number of classes (K)
 - 1: $\pi^i = \emptyset$
 - 2: **for** each $x \in U_{bag}^i$ **do**
 - 3: $H_x = \{h^j \in H | x \in U_{oob}^j\}$
 - 4: $[label(x), conf(x)] = MeasureConfidence(x, H_x, K)$
 - 5: **end for**
 - 6: Rank the examples in U_{bag}^i by decreasing order of confidence and select the n_k most confident examples for each class k
 - 7: **for** each $x \in U_{bag}^i$ **do**
 - 8: **if** (x is selected as confident) **then**
 - 9: $\pi^i = \pi^i \cup \{x, label(x)\}$
 - 10: **end if**
 - 11: **end for**
 - 12: **return** π^i
-

Algorithm 3 *MeasureConfidence*(x, H_x, K)

Require:

- an unlabeled training example (x), a committee of classifiers for which x is out-of-bag (H_x) and number of classes (K)
- 1: Apply H_x to generate the class probability distribution for x as $P(x) = \{p_k(x) : k = 1, \dots, K\}$
 - 2: $conf(x) = \max_{1 \leq k \leq K} P(x)$
 - 3: $label(x) = \operatorname{argmax}_{1 \leq k \leq K} P(x)$
 - 4: **return** $conf(x)$ and $label(x)$
-

Algorithm 4 *BuildOOBMatrix*($i, H, L_{oob}^i, U_{oob}^i, K$)

Require:

- a committee (H), current committee member index (i), out-of-bag labeled examples of h^i (L_{oob}^i), out-of-bag unlabeled examples of h^i (U_{oob}^i) and number of classes (K)
- 1: $O_{data}^i = 0, O_{label}^i = 0, O_{conf}^i = 0$
 - 2: **for** each $x \in L_{oob}^i$ **do**
 - 3: **if** ($h^i(x) == L_{oob}^i(x)$) **then**
 - 4: $O_{data}^i = O_{data}^i \cup \{x\}$
 - 5: $O_{label}^i(x) = h^i(x)$
 - 6: $O_{conf}^i(x) = 1$
 - 7: **end if**
 - 8: **end for**
 - 9: **for** each $x \in U_{oob}^i$ **do**
 - 10: $H_x = \{h^j \in H | x \in U_{oob}^j\}$
 - 11: $[label(x), conf(x)] = \text{MeasureConfidence}(x, H_x, K)$
 - 12: **if** ($h^i(x) == label(x)$) **then**
 - 13: $O_{data}^i = O_{data}^i \cup \{x\}$
 - 14: $O_{label}^i(x) = h^i(x)$
 - 15: $O_{conf}^i(x) = conf(x)$
 - 16: **end if**
 - 17: **end for**
 - 18: **return** O_{data}^i, O_{label}^i and O_{conf}^i
-

D. Why should our approach work

There are several advantages with the proposed method. First, SSFI will outperform RF when the available labeled training set is small. RF relies on the available training data for encouraging diversity. So if the size of the training set is small as for semi supervised setting, then the diversity among the ensemble members will be limited. Consequently, the ensemble error reduction will be small. SSFI incrementally adds newly-labeled examples to the training set. Therefore, it can improve the diversity and the average error of ensemble members constructed by RF and then improve the feature ranking paradigm. Second, since SSFI uses a diverse ensemble creation method, the measure of feature importance based on ensemble is more accurate than using a single classifier. Third, It is also worth mentioning that

Table I
THE DATASETS USED IN THE EXPERIMENTS

Datasets	# patterns	# features	# classes	Reference
Baseshock	1993	4862	2	[24]
Colon	62	2000	2	[25]
Leukemia	73	7129	2	[26]
Madelon	2598	500	2	[23]
Orlraws	100	10304	10	[24]
Ovarian	54	1536	2	[27]
Pcmac	1943	3289	2	[24]
SMK-CAN	187	19993	2	[24]
Toxicology	171	5748	2	[24]
Warpar10P	130	2400	10	[24]

the way feature importance measure is performed, in our approach differs, from the feature importance measure in RF as well as its recent extensions: *Co-forest* [18] and *semi-supervised random forest* [16]. In *Co-forest*, the variable importance measure can not be estimated from *OOB* samples since the bootstrap sample used to train each random tree is discarded after the first iteration. In semi-supervised RF, *OOB* data are all labeled. However, since the amount of labeled data is very small, the diversity of oob data is not sufficient. The out-of-bag estimates are biased as they depend on too few data.

IV. EXPERIMENTS

In this section, we provide empirical results on several benchmark and real high-dimensional datasets and compare SSFI against over state-of-the-art semi-supervised and supervised algorithms feature ranking algorithms. SSFI is compared with three other feature selection methods : (1) Breiman’s supervised random forests (RF) [12] taken as our gold standard ensemble supervised feature ranking approach, (2) the wrapper-type Forward Semi-Supervised Feature Selection (FwFS) [9], and (3) the filter-type Semi-Supervised Feature Selection via Spectral Analysis (sSelect) [10]. Ten benchmark and real labeled datasets, mostly selected from the *UCI Machine Learning Repository* [23], and from *ASU feature selection Repository* [24], were used to assess the performance of SSFI. They are described in Table I. We selected these datasets as they contain thousands features with comparatively much smaller sample size (e.g., *Leukemia*, *Toxicology*, *Orl10p*, *ovarian*, *colon* and *SMK-CAN*) and are thus good candidates for feature selection. Most of these datasets have already been used by other authors for testing the performance of their feature selection algorithms [5], [10], [9], [6].

A. Evaluation framework

To make fair comparisons, the same experimental settings in [10] was adopted here for *sSelect* approach, i.e., a neighborhood graph with a neighborhood size of 10, and the λ value is set to 0.1. For *FwFS*, we set $sizeFS = 10$, $SamplingRate = 0.5$, $SamplingTimes = 10$, $fnsteps =$

6 and $startfn = 5$, as suggested by the authors in [9]. RF and SSFI are tuned similarly. The number of features per bag is \sqrt{p} . The committee size N is computed using the following formula:

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

This formula ensures that each feature is drawn ten times at a confidence level of 0.01. Furthermore, the number of iterations $maxiter$ and the sample size n in our approach are set to 10, and 1, respectively. As we have to compare our approach with RF that uses *decisiontree*, the *treefit* Matlab implementation of decision tree is used as the base classifier in *FwFS* and *SSFI* for fair comparison. 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 [9], [10], the labeled sample set L consists of randomly selected 3 patterns per class, and the remaining patterns are used as unlabeled sample set U .

In order to assess the quality of a feature subset obtained with the aforementioned semi-supervised procedure, we train a classifier (a decision tree) 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 5. As mentioned above, the process specified in Algorithm 5 is repeated 10 times. The obtained accuracy is averaged and used for evaluating the quality of the feature subset selected according to each algorithm. In Figure 1, we plotted the accuracies of the above four approaches against the 10 most important features.

B. Results

Figure 1 shows the plots for accuracy vs. different numbers of selected features. As may be observed, SSFI outperforms the other three methods by a noticeable margin, especially on BaseHock, Leukemia, Madelon, PC-Mac and Toxicology, and sSelect performs the worst. SSFI 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 Leukemia, SMK-CAN and Orlraws. More importantly, SSFI outperforms RF on most datasets except on Warpar10P. However, we would like to mention that the labeled training examples in Warpar10P contains 30 examples for this dataset, which is an important amount of data (25% of the whole dataset). As expected, we a general trend that is observed in Figure 1 is that the more features we select, the better accuracy we achieve. Again, this is not surprising. However, it is worth mentioning that the accuracy of SSFI generally increases swiftly at the beginning

Algorithm 5 Feature Evaluation Framework

- 1: **for** each dataset X **do**
 - 2: build a randomly stratified partition (Tr, Te) , from X where $|Tr| = \frac{2}{3} \cdot |X|$ and $|Te| = \frac{1}{3} \cdot |X|$;
 - 3: Generate labeled data L by randomly sampling from Tr 3 instances per class;
 - 4: $U = Tr \setminus L$;
 - 5: $SF_{sSelect} = \text{Apply } sSelect \text{ with } L + U$;
 - 6: $SF_{FwFS} = \text{Apply } FwFS \text{ with } L + U$;
 - 7: $SF_{SSFI} = \text{Apply } SSFI \text{ with } L + U$;
 - 8: $SF_{RF} = \text{Apply } RF \text{ with } L$;
 - 9: **for** $i = 1$ to 10 **step** 1 **do**
 - 10: Select top i features from $SF_{sselect}$, SF_{FwFS} , SF_{SSFI} and SF_{RF} ;
 - 11: $Tr_{sSelect} = \Pi_{SF_{sSelect}}(Tr)$;
 - 12: $Tr_{FwFS} = \Pi_{SF_{FwFS}}(Tr)$;
 - 13: $Tr_{SSFI} = \Pi_{SF_{SSFI}}(Tr)$;
 - 14: $Tr_{RF} = \Pi_{SF_{RF}}(Tr)$;
 - 15: Train the *Baselearner* using $Tr_{sSelect}$, Tr_{FwFS} , Tr_{SSFI} and Tr_{RF} and record accuracy obtained on Te ;
 - 16: **end for**
 - 17: **end for**
-

(the number of selected feature is small) and slows down at the end. These experiments suggest that SSFI ranks the features properly 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. Note also that the high computational complexity of FwFS is a major drawback with large dimensional data. In addition, the accuracy of FwFS often tend to decrease as more features are included.

Fore sake of completeness, we also averaged the accuracy for different numbers of selected features. The averaged accuracies between SSFI and the other methods over the top 10 features are depicted in Table II. Again, as may be observed, SSFI clearly outperforms RF, Sselect and Fw-semiFS by a noticeable margin, on all datasets except for Warpar10P where RF works the best. SSFI is significantly better then all three approaches ($p < 0.02$) according to the Wilcoxon signed-rank test (unsufficient number of datasets and classifiers to apply the Friedman test) [28]. Finally, these experiments confirm the ability of the proposed permutation feature importance measure to rank the relevant features accurately, compared to a powerful fully supervised approach like RF, by exploiting efficiently the information from the unlabeled data.

V. CONCLUSION

We discussed a new semi-supervised feature importance evaluation method, called SSFI, combining ideas from co-

Table II
ACCURACY AVERAGED OVER THE 10 MOST IMPORTANT FEATURES

data	SSFI	FwFS	RF	sSelect
Basehock	0.6080	0.5193	0.5445	0.5064
Colon	0.5645	0.5404	0.5327	0.5431
Leukemia	0.7707	0.6396	0.6950	0.6246
Madelon	0.5580	0.5024	0.5076	0.5008
Orlraws	0.6952	0.6200	0.6563	0.6145
Ovarian	0.8372	0.7444	0.7583	0.6327
Pcmac	0.5953	0.5312	0.5120	0.5123
SMKCAN	0.5671	0.5339	0.5408	0.5275
Toxicology	0.4777	0.3656	0.4008	0.3435
Warpar10P	0.4432	0.3952	0.4876	0.2966

training and random forests with a new permutation-based out-of-bag feature importance measure. Both labeled and unlabeled out-of-bag instances were used to evaluate the relevance of the features. Empirical results on ten benchmark datasets indicated that SSFI lead to significant improvement over state-of-the-art semi-supervised algorithms. More importantly, SSFI was shown to outperform Random Forests on several datasets in terms of feature selection accuracy. The method also shows promise to deal with very large domains. Future substantiation through more experiments on biological databases containing several thousands of variables are currently being undertaken.

REFERENCES

- [1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [2] J. Hua, W. Tembe, and E. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, pp. 409–424, 2009.
- [3] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, pp. 95–116, 2007.
- [4] J. Dy and C. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.
- [5] H. Elghazel and A. Aussem, "Feature selection for unsupervised learning using random cluster ensembles," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010)*, 2010, pp. 168–175.
- [6] Y. Hong, S. Kwong, Y. Chang, and R. Qingsheng, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognition*, vol. 41, no. 9, pp. 2742–2756, 2008.
- [7] P. Mitra, A. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.
- [8] Y. Cheng, Y. Cai, Y. Sun, and J. Li, "Semi-supervised feature selection under logistic i-relief framework," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, 2008, pp. 1–4.
- [9] J. Ren, Z. Qiu, W. Fan, H. Cheng, and P. S. Yu, "Forward semi-supervised feature selection," in *Proceedings of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008)*, 2008, pp. 970–976.
- [10] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007, pp. 641–646.
- [11] J. Zhao, K. Lu, and X. He, "Locality sensitive semi-supervised feature selection," *Neurocomputing*, vol. 71, no. 10-12, pp. 1842–1849, 2008.
- [12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [13] Z. Xu, I. King, M. R. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Transactions on Neural Networks*, vol. 21, no. 7, pp. 1033–1047, 2010.
- [14] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT 1998)*, 1998, pp. 92–100.
- [15] M. F. A. Hady and F. Schwenker, "Combining committee-based semi-supervised learning and active learning," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 681–698, 2010.
- [16] C. Leistner, A. Saffari, J. Santner, and H. Bischof, "Semi-supervised random forests," in *Proceedings of the 12th International Conference on Computer Vision (ICCV 2009)*, 2009, pp. 506–513.
- [17] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.
- [18] M. Li and Z. H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1088–1098, 2007.
- [19] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, 2006.
- [20] M.-L. Zhang and Z.-H. Zhou, "Exploiting unlabeled data to enhance ensemble diversity," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010)*, 2010, pp. 619–628.
- [21] J. Wang, S. Luo, and X. Zeng, "A random subspace method for co-training," in *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008*, 2008, pp. 195–200.

- [22] Y. Yaslan and Z. Cataltepe, "Co-training with relevant random subspaces," *Neurocomputing*, vol. 73, no. 10-12, pp. 1652–1661, 2010.
- [23] C. Blake and C. Merz, "Uci repository of machine learning databases," 1998.
- [24] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, and A. Anand, "Feature selection," 2011.
- [25] A. Ben-Dor, L. Bruhn, A. Laboratories, N. Friedman, M. Schummer, I. Nachman, U. Washington, U. Washington, and Z. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7, pp. 559–584, 2000.
- [26] T. Golub, Slonim, D.K., P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, and H. Coller, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.
- [27] M. Schummer, W. V. Ng, and R. E. Bumgarnerd, "Comparative hybridization of an array of 21,500 ovarian cdnas for the discovery of genes overexpressed in ovarian carcinomas," *Gene*, vol. 238, no. 2, pp. 375–385, 1999.
- [28] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

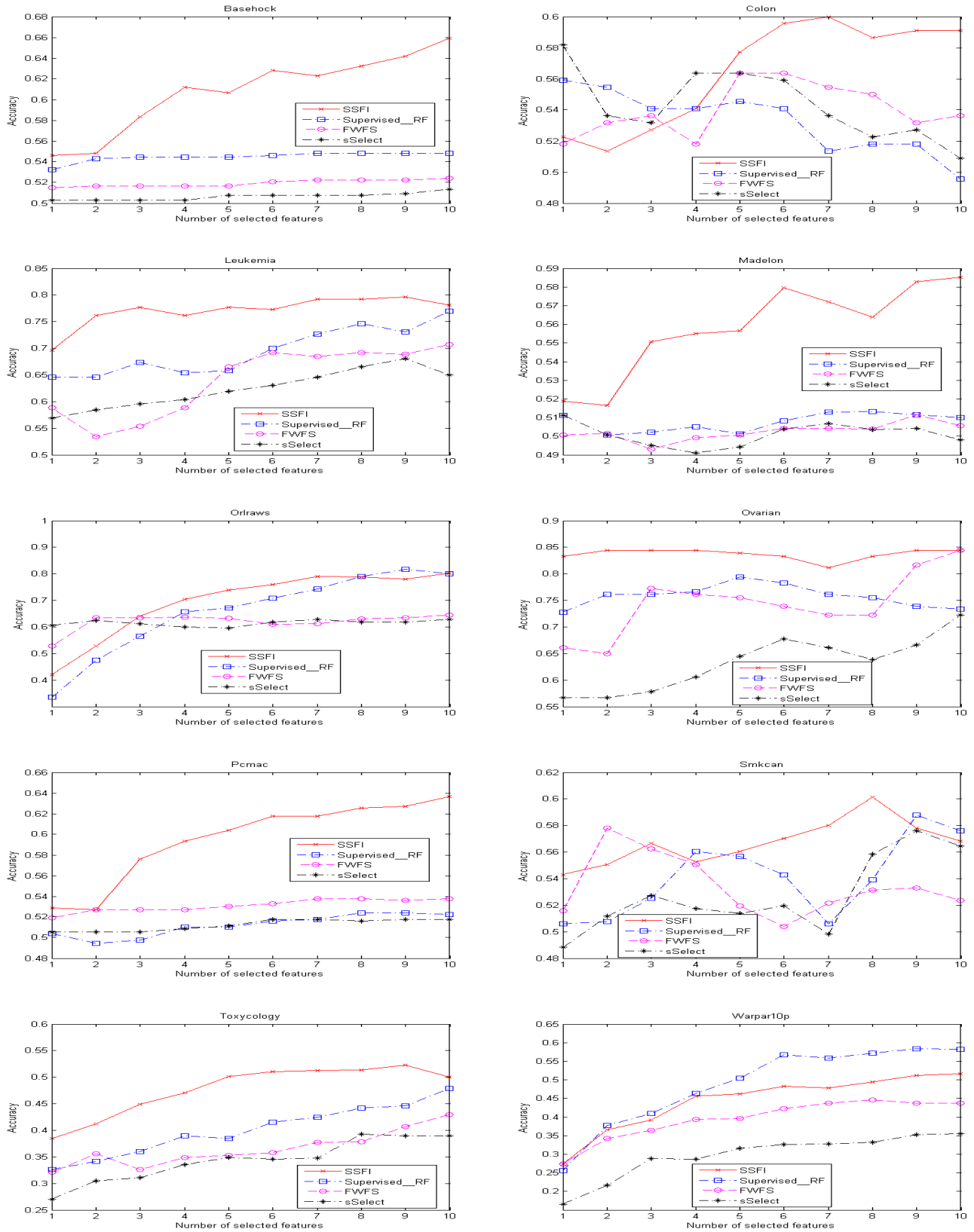


Figure 1. Accuracy evaluated on a test set as the number of most relevant features fed as input to the classifier is varied.