

# Forward Semi-Supervised Feature Selection

Jiangtao Ren<sup>1</sup>, Zhengyuan Qiu<sup>1</sup>, Wei Fan<sup>2</sup>, Hong Cheng<sup>3</sup>, and Philip S. Yu<sup>4</sup>

<sup>1</sup> Department of Computer Science, Sun Yat-Sen University, Guangzhou, China \*  
issrjt@mail.sysu.edu.cn, qzhengy@mail2.sysu.edu.cn

<sup>2</sup> IBM T.J.Watson Research, USA  
weifan@us.ibm.com

<sup>3</sup> Computer Science Department, UIUC, USA  
hcheng3@uiuc.edu

<sup>4</sup> Computer Science, University of Illinois at Chicago, USA  
psyu@cs.uic.edu

**Abstract.** Traditionally, feature selection methods work directly on labeled examples. However, the availability of labeled examples cannot be taken for granted for many real world applications, such as medical diagnosis, forensic science, fraud detection, etc, where labeled examples are hard to find. This practical problem calls the need for “semi-supervised feature selection” to choose the optimal set of features given both labeled and unlabeled examples that return the most accurate classifier for a learning algorithm. In this paper, we introduce a “wrapper-type” forward semi-supervised feature selection framework. In essence, it uses unlabeled examples to extend the initial labeled training set. Extensive experiments on publicly available datasets shows that our proposed framework, generally, outperforms both traditional supervised and state-of-the-art “filter-type” semi-supervised feature selection algorithms [5] by 1% to 10% in accuracy.

**Key words:** feature selection, semi-supervised learning

## 1 Introduction

Feature selection is an important data processing step in high dimensional data learning tasks. Traditionally, feature selection methods use information from “labeled data” to find the most informative or most useful feature subsets [1, 2], but the information in the “unlabeled” data is not used. When the size of the “labeled” data is limited, it is difficult to select an ideal feature subset only on “labeled” data. Recently, there have seen considerable interests in learning with labeled and unlabeled data [3]. In many learning tasks, the effectiveness of semi-supervised learning has been demonstrated [4]. Zhao and Liu [5] introduced a semi-supervised feature selection algorithm based on spectral analysis. Later, they exploited intrinsic properties underlying supervised and unsupervised feature selection algorithms, and proposed a unified framework for feature

---

\* Supported by the National Natural Science Foundation of China under Grant No. 60703110

selection based on spectral graph theory [6]. Yet these algorithms are “filter” models which 1) do not select feature subset for specific learning method and 2) sample selection bias is ignored. It is important if one can employ certain strategy to extend the initial training set with unlabeled data to overcome the biased distribution problem, and in the same time, perform a “wrapper type” feature selection for specific learning model.

In this paper, we introduce a “wrapper-type” forward semi-supervised feature selection framework. It uses the mechanism of random selection on unlabeled data to form new training sets, and the most frequently selected feature is added to the result feature subset in each iteration. With the introduction of randomly selected data with predicted labels, the sufficiency and diversity of the training sets can be improved, which in return helps to choose the most discriminative features.

In order to evaluate the effectiveness of our framework, we give formal analysis as well as conduct extensive experimental study on the algorithm. First, bipartite graph has been employed to formally show that unlabeled data is helpful in feature selection. Secondly, we have conducted extensive experiments with extremely few labeled instances (such as, only 6 labeled instances) which can reflect the scenario of data limitation. The results of these experiments show that the proposed “wrapper-type” framework, generally, outperforms the traditional feature selection method and state-of-the-art “filter-type” semi-supervised feature selection algorithms [5] by 1% to 10% in accuracy. It performs especially well when the size of the labeled data set is very small.

## 2 The Framework

Supervised sequential forward feature selection (SFFS) is one of the most widely used feature selection algorithms. Conceptually, it is an iterative process starting with an empty feature subset. In each iteration, one feature is chosen among the remaining features. To determine which feature to add, it tests the accuracy of a model built on the incremented feature subset. The feature that results in the highest accuracy is selected. Normally, the process terminates when no additional features could result in an improvement in accuracy or the feature subset already reaches a predefined size. Since this process can be easily implemented and is usually quite effective, it remains one of the widely adopted supervised feature selection methods. In this work, we extend it to take unlabeled data into account.

### 2.1 Our Approach

We propose a new framework of forward feature selection, which performs feature selection on both labeled and unlabeled data. Our algorithm uses SFFS and wrapper model to select *startfn* features initially, then the *startfn* features are used to train a classifier, the classifier is then used to predict the labels of the unlabeled data. Then the randomly selected *samplingRate%* unlabeled data with predicted labels is combined with labeled data to form a new training set.

```

Input:  $L, U, sizeFS, samplingRate, samplingTimes, maxIterations,$ 
          $startfn, fnstep$ 
Output:  $resultfs$ 
1 Perform feature selection on  $L$  using SFFS, select  $startfn$  features to
  form the current feature subset  $currentfs$ ;
2  $ReducedL \leftarrow L * currentfs$ ;
3  $ReducedU \leftarrow U * currentfs$ ;
4 for  $iteration \leftarrow 1$  to  $maxIterations$  do
5    $Predicted \leftarrow classifier(ReducedL, ReducedU)$ ;
6   for  $rand \leftarrow 1$  to  $samplingTimes$  do
7     Randomly select  $samplingRate\%$  of instances from  $Predicted$ ,
     and add it into  $L$  to form a new dataset  $NewDataset$ ;
8     Perform feature selection on  $NewDataset$  using SFFS, select
      $fnstep$  features to form feature subset  $fs[rand]$ ;
9   end
10  Count the frequency of every feature in  $fs$ , add the most frequent
    and not in  $currentfs$  feature into  $currentfs$ ;
11   $ReducedL \leftarrow L * currentfs$ ;
12   $ReducedU \leftarrow U * currentfs$ ;
13  if  $SIZE(currentfs) == sizeFS$  then break;
14 end
15  $resultfs \leftarrow currentfs$ ;

```

**Fig. 1.** Forward semi-supervised feature selection (FW-SemiFS)

Afterwards, the new training dataset is used to select  $fnstep$  features based on SFFS and the learner. The random selection and feature selection process repeat  $samplingTimes$  times and  $samplingTimes$  groups of features are selected. And we count the frequency of every feature in the  $samplingTimes$  groups of features, and the one with the most frequency is added to form a new feature subset. This process repeats until the size of the feature subset reaches a predefined number.

The algorithm is described in detail in Figure 1.  $L$  denotes the labeled data,  $U$  denotes the unlabeled data;  $sizeFS$  denotes the predefined number of selected features;  $samplingRate$  denotes the sampling rate according to the unlabeled data with predicted labels;  $samplingTimes$  denotes the randomly sampling times;  $maxIterations$  denotes the max iteration times;  $startfn$  denotes the start feature number;  $fnstep$  denotes the number of features selected in every step.  $resultfs$  denotes the output feature subset. In our algorithm, “\*” denotes the features reduction operator.

## 2.2 Method Analysis

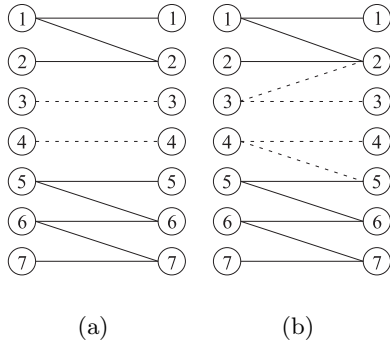
The classifier “wrapped” in the feature selection algorithm is used to predict the labels of the unlabeled instances as well as to evaluate the effectiveness of the chosen feature subset. Obviously, a more accurate classifier can select more effective feature subsets to represent the target distribution. Next, we will

demonstrate why the use of unlabeled data can improve the “accuracy” of the classifier at each step of feature selection.

Feature selection can be formulated as:  $f_1(X') = f(X)$  subject to make  $X'$  as small as possible, where  $f_1$  and  $f$  are the target functions on chosen feature subset  $X'$  and full feature set  $X$ , respectively. When unlabeled examples are used in feature selection, its process is similar to “co-training”, where  $X'$  and  $X$  are interpreted as the two “views” in co-training. The classifier constructed in one view is expected to be helpful for the other.

Now we can adopt the bipartite graph model to facilitate our interpretation on why “wrapper classifier” with unlabeled examples can improve the performance of feature selection. The training process can be regarded as a bipartite graph  $G_D(X'; X)$  [7]. In Figure 2, each node on the left-hand side denotes one instance in  $X'$  view and the one on the right-hand side denotes the same instance in  $X$  view. Any two instances in the labeled dataset will be connected by an edge if they belong to the same class, and those edges are shown in solid lines. Let  $S$  be a finite dataset that consists of the labeled dataset, then  $G_S$  is a bipartite graph in  $S$  whose components show the concept distribution in  $G_S$ . In the case of unlabeled instances, each vertex on the left, depending on the its prediction on  $X'$  view, will be connected to a vertex on the right. Thus, it is connected to the most probable category.

Next, we analyze why unlabeled data can improve the generalization accuracy of the classifier on the platform of bipartite graph. Let  $G_D$  be the bipartite graph in the all-real-data distribution  $D$ . As we know, the components in  $G_D$  reflect the concept distribution on the real dataset, and we can achieve completely correct prediction if we get the components in  $G_D$ . But it is impossible to achieve this when the dataset  $D$  is infinite. The work of co-training model is to find components that are much “similar” to those in  $G_D$  by the use of unlabeled data. Given  $S$ , as the predictions on the unlabeled data increase, the edges will be added to the bipartite graph and the number of components will



**Fig. 2.** Graphs  $G_D$  and  $G_S$

drop as components merge together. Intuitively, the components in  $G_S$  are more similar to the components in  $G_D$ , leading to a more accurate prediction on unlabeled data. For example, as demonstrated in Figure 2, vertex 1 and 2 represent labeled instances with the same true labels, vertex 5, 6 and 7 represent labeled instances with the same true labels but different from 1 and 2. On the other hand, vertex 3 and 4 represent unlabeled instances. Before knowing the predicted labels of 3 and 4, there are four components in Figure 2(a). But by introducing the predicted labels of vertex 3 and 4 (for example, vertex 3 has a predicted label

the same as vertex 2, and vertex 4 has a predicted label the same as vertex 5), unlabeled vertex 3 can be connected to vertex 1 and 2 to form a new component, and unlabeled vertex 4 can be connected to vertex 5, 6 and 7 to form another new component. Then the number of components is reduced to 2, as illustrated in Figure 2(b). Blum and Mitchell [7] had demonstrated that the components in  $G_S$  will be similar to those in  $G_D$  if the unlabeled set is large enough. Thus, they capture the real concept distribution of the real dataset and give the classifier higher prediction accuracy.

Based on the above analysis, it is clear that the accuracy of the wrapper classifier can be improved by the iteration process of FW-SemiFS. In other words, the effectiveness of FW-SemiFS can be improved by applying unlabeled data.

### 3 Experiment

In order to evaluate the effectiveness of our proposed algorithm, we have conducted extensive experiments on several datasets. Table 1 summarizes the information of the datasets that we used. In our experiment, the labeled data and unlabeled data are randomly selected from the whole dataset, and the left is used as testing data.

#### 3.1 Experiment Settings and Evaluation Method

In FW-SemiFS experiments, the parameters *startfn*, *fnstep*, *samplingRate*, and *samplingTimes* are set to 5, 6, 50%, and 10, respectively. For comparison, SFFS and Semi-supervised Laplacian Score (called SLS for short) are also conducted. SFFS is a supervised feature selection algorithm described in the previous section; and SLS is a semi-supervised feature selection algorithm which is similar to Zhao and Liu’s algorithm [5], but it uses Laplacian score as its ranking criterion. We chose three machine-learning models, NaiveBayes, NNge, and k-NN, to evaluate the effectiveness of the algorithms. Specifically, NNge is the nearest neighbor like algorithm using non-nested generalized examples; k-NN is  $k$  nearest neighbor classifier whose parameter  $k$  is set to 5.

For FW-SemiFS and SLS, we employ both the labeled dataset and unlabeled dataset to perform semi-supervised feature selection. But for SFFS approach, we only employ the labeled dataset to perform feature selection. After selecting the feature subset *ResultFS*, we construct the classifier only with the labeled dataset

Table 1. Dataset summary

Dataset	#Labeled	#Unlabeled	#Testing	#Features	#Classes
German	6	100	294	20	2
Ionosphere	6	100	245	34	2
Mushroom	6	100	294	22	2
Sonar	6	100	102	60	2
Waveform	6	100	294	21	3
wdbc	6	100	463	30	2
ColonTumor	6	30	26	2000	2

**Table 2.** The accuracy comparison between SFFS, SLS, and FW-SemiFS

Dataset	NaiveBayes				NNge				k-NN			
	Full	SFFS	SLS	SemiFS	Full	SFFS	SLS	SemiFS	Full	SFFS	SLS	SemiFS
German	52.72	54.08	51.59	<b>55.44</b>	67.01	65.65	58.84	<b>70.41</b>	64.63	63.61	60.20	<b>67.01</b>
Ionosphere	73.47	73.47	67.76	<b>75.51</b>	71.84	71.43	65.04	<b>74.15</b>	72.38	70.61	52.93	<b>73.61</b>
Mushroom	89.80	86.73	<b>91.50</b>	85.71	<b>86.28</b>	83.45	82.54	84.35	65.31	63.95	62.24	<b>69.05</b>
Sonar	52.94	52.61	48.37	<b>59.15</b>	68.63	<b>70.59</b>	<b>70.59</b>	<b>70.59</b>	56.86	61.76	59.80	<b>63.73</b>
Waveform	38.44	<b>46.94</b>	39.80	40.48	61.90	<b>69.39</b>	58.84	<b>69.39</b>	42.52	51.70	51.36	<b>52.72</b>
wdbc	82.72	77.97	<b>87.90</b>	82.51	90.28	<b>92.87</b>	91.79	91.58	<b>92.22</b>	89.85	85.31	87.04
ColonTumor	39.74	76.92	53.85	<b>84.62</b>	37.18	47.44	<b>55.13</b>	35.90	53.85	35.90	<b>58.98</b>	41.03

**Table 3.** Means and standard deviations of accuracies

Dataset	Method	NaiveBayes		NNge		k-NN	
		Mean	StDev	Mean	StDev	Mean	StDev
German	SFFS	54.49	2.16	66.14	1.43	62.35	4.98
	SLS	52.99	0.89	63.14	5.35	62.86	2.45
	SemiFS	<b>55.93</b>	2.22	<b>69.26</b>	2.77	<b>63.82</b>	3.92
Ionosphere	SFFS	72.32	1.48	72.50	1.08	71.51	1.07
	SLS	68.98	2.78	65.17	3.17	60.50	6.28
	SemiFS	<b>74.64</b>	0.99	<b>74.46</b>	1.00	<b>73.92</b>	0.53
Mushroom	SFFS	88.65	3.17	84.76	0.84	61.31	4.68
	SLS	<b>88.80</b>	2.44	82.41	3.78	<b>63.39</b>	5.53
	SemiFS	87.33	3.28	<b>85.06</b>	0.54	61.78	4.68
Sonar	SFFS	53.29	2.41	<b>71.57</b>	1.13	56.25	4.85
	SLS	50.51	2.27	68.20	2.91	58.09	3.07
	SemiFS	<b>58.58</b>	0.98	70.10	0.72	<b>61.21</b>	4.64
Waveform	SFFS	<b>43.47</b>	3.21	62.48	3.53	50.30	6.74
	SLS	40.18	1.22	60.89	3.65	51.79	3.84
	SemiFS	41.71	3.17	<b>63.58</b>	3.13	<b>52.42</b>	6.06
wdbc	SFFS	79.70	3.24	91.28	0.90	85.12	5.50
	SLS	<b>85.81</b>	3.34	91.20	0.60	86.00	0.73
	SemiFS	82.82	1.11	<b>91.62</b>	0.66	<b>87.39</b>	1.04
ColonTumor	SFFS	52.31	4.34	44.27	2.51	39.06	1.80
	SLS	43.42	3.14	<b>57.78</b>	3.00	<b>61.45</b>	2.89
	SemiFS	<b>54.02</b>	1.99	40.00	1.95	53.85	5.24

and *ResultFS*, and then the unseen testing dataset is employed to evaluate the classification accuracy.

### 3.2 Empirical Results

Table 2 shows the experiment results of the SFFS, SLS and FW-SemiFS methods when the size of the selected feature subset is 10, as well as the classification accuracy without feature selection (denoted as “Full” in the table). The numbers are shown in bold when it is the highest one among “Full”, “SFFS”, “SLS” and “SemiFS”. From table 2, we could find that the accuracies of FW-semiFS are higher than that of two other feature selection algorithms in 13 out of 21 cases.

For further view of the algorithm comparison, we conduct statistical analysis of the experiment results. For every dataset and every learning algorithm, we run SFFS, SLS and FW-SemiFS respectively according to 16 different feature subset sizes (from 5 to 20), and then get 16 groups of feature subsets and related accuracies. Each group has three feature subsets and three related classification accuracies according to SFFS, SLS and FW-SemiFS, respectively. After that, we

calculate the mean and standard deviation of these 16 accuracies for each group, which are listed in table 3.

From Table 3, we can see that the mean of the classification accuracies of FW-SemiFS are higher than that of two others most of the time. On German and Ionosphere datasets, the means of FW-SemiFS are the highest ones for NaiveBayes, NNge, and k-NN learners, along with small standard deviation; Although, for German dataset, the standard deviation is larger than that of two other algorithm, the mean of FW-SemiFS is much higher than that of two others. The similar phenomena can be also observed for Sonar and wdbc datasets.

## 4 Conclusion and Future Work

We have explored the use of unlabeled examples to facilitate “wrapper-type” forward semi-supervised feature selection. The proposed algorithm works in an iterative procedure. In each step, unlabeled examples receive labels from the classifier constructed on currently selected feature subset. Then a random sample of now “labeled” unlabeled examples is concatenated with the training set to form a “joint” dataset, where a wrapper-based feature selection is performed. Experiment results show that the proposed approach, generally, can obtain 1% to 10% higher accuracy than other supervised and semi-supervised feature selection algorithms.

**Future Work** The work discussed in this paper represents techniques based on random selection mechanism. In the future, we plan to extend the techniques by using prediction confidence as a criterion to select those unlabeled data which is most probable to have correctly predicted labels.

## References

1. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, **17** 4 (2005) 491-502
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research*, **3** (2003) 1157-1182
3. Seeger, M.: Learning with labeled and unlabeled data. Technical report, (2000)
4. Zhu, X.: Semi-Supervised Learning Literature Survey. Computer Sciences Technical Report 1530, University of Wisconsin-Madison, (2005)
5. Zhao, Z., Liu, H.: Semi-supervised Feature Selection via Spectral Analysis. *SIAM International Conference on Data Mining (SDM-07)*, (2007)
6. Zhao, Z., Liu, H.: Spectral Feature Selection for Supervised and Unsupervised Learning. *Proceeding of the 24th International Conference on Machine Learning (ICML-07)*, (2007)
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*, (1998) 92-100