On Comparison of Feature Selection Algorithms

Payam Refaeilzadeh and **Lei Tang** and **Huan Liu**
Department of Computer Science & Engineering
Arizona State University
Tempe, Arizona 85287
{Payam, L.Tang, Huan.Liu}@asu.edu

Abstract

Feature selection (FS) is extensively studied in machine learning. We often need to compare two FS algorithms (A_1, A_2). Without knowing true relevant features, a conventional way of evaluating A_1 and A_2 is to evaluate the effect of selected features on classification accuracy in two steps: selecting features from dataset D using A_i to form D'_i, and obtaining accuracy using each D'_i, respectively. The superiority of A_1 or A_2 can be statistically measured by their accuracy difference. To obtain reliable accuracy estimation, $k-$fold cross-validation (CV) is commonly used: one fold of data is reserved in turn for test. FS may be performed only once at the beginning and subsequently the results of the two algorithms can be compared using CV; or FS can be performed k-times inside the CV loop. At first glance, the latter is the obvious choice for accuracy estimation. We investigate in this work if the two really differ when comparing two FS algorithms and provide findings of bias analysis.

Introduction

Feature selection (FS) is the process of reducing dimensionality by removing irrelevant features (Guyon & Elisseeff 2003). It is usually applied as a pre-processing step in machine learning tasks. FS is employed in different applications with a variety of purposes: to overcome the *curse of dimensionality*, to remove irrelevant and redundant features (Blum & Langley 1997) thus improving classification performance, to streamline data collection when the measurement cost of attributes are considered (e.g., drug design targeting at specific genes), to speed up the classification model construction, and to help unravel and interpret the innate structure of datasets (John, Kohavi, & Pfleger 1994). FS algorithms broadly fall into two categories[1]: the *filter* model and the *wrapper* model (John, Kohavi, & Pfleger 1994). The *filter* model relies on some intrinsic characteristics of data to select features without involving classification learning; the *wrapper* model, typically uses a classifier to evaluate feature quality. The wrapper model is often computationally more expensive than the filter model, and its selected features are biased toward the classifier used (Guyon & Elisseeff 2003). A large number of *filter* algorithms have been proposed in literature (Liu & Yu 2005). This work investigates evaluation methods for FS algorithms of the *filter* model.

Feature Selection Evaluation

Feature selection evaluation aims to gauge the efficacy of a FS algorithm. It boils down to the evaluation of its selected features, and is an integral part of FS research. In evaluation, it is often required to compare a new proposed feature selection algorithm with existing ones.

The evaluation tasks would have been simple, if the ground truth (the true relevant features) were known. However, this is almost never the case for real-world data. Since we don't have the ground truth for real-world dat

Indirect Evaluation

As is well known, a classifier achieving good classification performance on training data does not necessarily generalize well on new test data, or the classifier might *overfit* the training data. Hence, k-fold cross validation (CV) or its variants are widely adopted to separate training and test data. In this process, data is split into k equal-sized *folds*. In each of the k iterations of the CV loop, one fold is held out for testing while the remaining $k-1$ folds are used for training the classifier. The averaged performance, typically accuracy[2], is reported. *The labeling information of the test set should never be used during classifier training*. When incorporating FS in classifier learning, we clearly should perform FS **inside** the cross-validation loop (Method **IN** as shown in Figure 1). That is, for each iteration, feature selection is applied to the training set before classifier construction.

When treating FS as a pre-processing step for dimensionality reduction, would it be appropriate to separate FS from classifier learning? That is, performing feature selection first and then gauging the efficacy of the FS algorithm via CV in comparison with the quality of selected features of some baseline algorithm. This method performs FS **outside** the CV loop (Method **OUT** as shown in Figure 2)[3] This com-

[1] This work focuses on supervised FS. Unsupervised FS studies how to select features without class labels (Dy & Brodley 2004).

[2] In this paper, we focus on accuracy. Other measurements can be precision, recall, F-measure, AUC, etc.

[3] This method seems a *de facto* one in many FS papers.

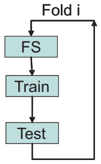

Figure 1: Method IN

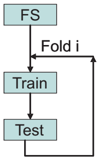

Figure 2: Method OUT

monly used method was not questioned for a long time, until the publication of (van 't Veer *et al.* 2002). In this work, the authors used the full set of data instances to select genes (features) and then perform CV to estimate classification accuracy after FS. Upon its publication in *Nature*, it was criticized for its "information leakage" from the test data[4]. After revising their methodology to Method IN, the authors published a Web supplement to their article, reporting a decrease in CV accuracy from 83% to 73%.

At first glimpse, performing FS outside the CV loop (OUT) is unacceptable as it is tantamount to peeking at the hold-out test data. The "leakage" of label information from the test data during FS will lead to an "optimistic" *accuracy estimate*, which is also observed in (van 't Veer *et al.* 2002). However, in comparing two FS algorithms, the leakage occurs to both. So the question is:

> *Does this bias in accuracy estimation really matter when comparing two feature selection algorithms?*

A Closer Look into Method IN & OUT

Clearly, the quality of selected features is highly correlated with the number of instances available during training. In order to select good features, we should use all the available data. However, k-fold CV complicates the above as it is essentially making an approximation that, by holding out $1/k$ instances for testing, a comparison made using 90% of the data (in case of 10-fold CV) can be generalized to a comparison made using all the data. This approximation is compulsory for classification evaluation because otherwise we would not be able to 'fairly' measure accuracy, but may be unnecessary for feature selection.

Method IN is holding out one fold for FS and may be too conservative. Comparatively, method OUT performs FS using all the instances. As it "peeks" at the label information before evaluation, the classification performance may be too optimistic. Since these biases can affect the results of *both* algorithms under comparison, it is unclear which bias would yield more accurate comparison results. Moreover, holding out one fold for FS in method IN could exacerbate the *small sample problem* with FS (Jain & Zongker 1997) as in many applications the available data is just a small sample of the whole population. A large variance in FS performance can result from small samples. Method OUT alleviates this problem by using all the available instances for FS.

In light of the above arguments, we conjecture that (1) OUT may be too liberal in estimating accuracy since it uses all available data for FS, and (2) IN may be too conservative as it holds out one fold of data in FS. Recall that our goal is to compare two feature selection algorithms. We are thus seeking to answer the following questions: (1) Do IN and OUT indeed have different bias? (2) Does this bias affect the outcome of pair-wise comparison? (3) Which method should be adopted when the number of samples are small/large? A great number of FS algorithms have been proposed in literature, and the answers to these questions will determine if some experimental results using either IN or OUT should be reconsidered. Hence, it is important to evaluate the two evaluation methods and determine which is more reliable.

Evaluating the Evaluation

Ultimately we seek to answer the question: which evaluation method (IN or OUT) is more truthful in comparing two feature selection algorithms? To answer this question, we must know the ground truth about which method is truly better. To obtain the ground truth we must compare the two FS algorithms in a special experimental setup that does not contain any of the controversial elements from CV and methods IN and OUT. This special setup, called TRUTH, is as follows: For a particular data, we create 100 training and 100 test sets. This ensures a large sample. We further ensure that these 200 sets are independent. Note that creating 200 independent sets would require a very large number of instances. With real-world datasets there are simply not enough instances to create such a setup. For this reason we resort to using synthetic data to study the properties of IN and OUT.

We use each of the 100 train set to perform feature selection using a pair of FS algorithms (A_1 and A_2). We then train the classifier on the resulting pair of data (composed of only selected features). Finally we measure the accuracy of the pair of trained classifiers (one for the subset of features selected by A_1, and one for A_2) on the corresponding test set. At the end we have 100 paired measurements of accuracy for A_1 and A_2. We perform a paired t-test to determine whether A_1 is better than A_2 or if they are equivalent. Thus this method (TRUTH) yields a conclusion (E_{TRUTH}) with one of three possible values: Win (A_1 is better), Loss (A_2 is better) or Draw (there is no significant difference).

We also perform CV twice on *each* of the 100 training sets, one time using method IN and one time using method OUT. We use the same folds for both methods. We then perform a paired t-test on the results of each CV experiment. This allows us to obtain 100 comparison conclusion for OUT (E_{OUT}) and 100 for IN (E_{OUT}) about which FS algorithm is better or if they are the same. Ultimately we have 100 paired conclusions of IN and OUT and a single TRUTH. Based on this data, we can perform a statistical significance test to determine which method is more truthful.

We generated two data sources with continuous feature values, one linearly separable and the other non-linearly separable with feature values independently drawn from a uniform distribution. Both have 60 features only 10 of which are relevant. We also generated three discrete data sources according to the specification for the MONKS' problems (Thrun *et al.* 1991). We added Gaussian random

[4] Most notably (Molla *et al.* 2004) contains a detailed discussion.

Table 1: N and F in Synthetic Data Experiments.

Dataset	#Instances (N)	#Features (F)
Continuous	$\{125, 250, 375, 500, 625, 750, 875, 1000\}$	$\{3, 5, 10, 20, 40\}$
Discrete	$\{40, 60, 80, 100, 120, 140, 160, 180\}$	$\{1, 2, 3, 4, 5\}$

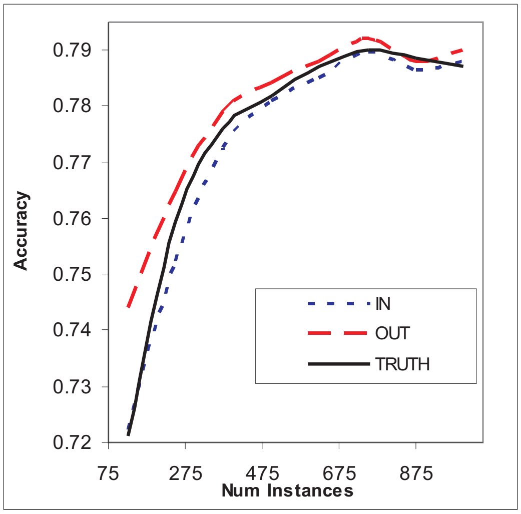

Figure 3: Average accuracy (as determined by each evaluation method: IN, OUT and TRUTH) of the 1NN classifier on non-linear synthetic data when 10 features are selected by FCBF.

noise to the target value for the continuous data before binarizing it into two classes. Noise was added to MONK3 according to the MONK's problems specification. For each experiment, we fixed the number of training instances (N), the number of features selected by the FS algorithms (F), the classifier and the pair of FS algorithms. We repeated the experiment with different values for F and N, with four different classifiers: SVM, Naïve Bayes (NBC), Nearest Neighbor (1NN) and decision tree (C4.5); and three FS algorithms: ReliefF (Kononenko 1994), FCBF (Yu & Liu 2003), and information gain (IG).

The different values for N and F are shown in Table 1. For the continuous data, the number of relevant features (R) is 10, so we vary F from $R/4$ to $4R$. The MONK data only contained 6 features so we included all possible values of F. The values for N were selected because we observed that the accuracy for most models stabilized at around 500 instance so we let N vary from 125 to $1,000$. Similarly for the MONK data marginal accuracy dropped off at around 100 instances so N varies from 40 to 180.

Accuracy Estimation Bias

We first look at whether IN and OUT indeed both demonstrate a bias. The results show that OUT often over-estimates the true accuracy, while IN often under-estimates the true accuracy. For example, Figure 3 shows a plot of average accuracy for IN, OUT and TRUTH in a particular case (1NN classifier on non-linear synthetic data when 10 features are selected by FCBF). Note that the TRUTH accuracy curve is enveloped by IN below and OUT above. We observed similar trends for other classifiers, feature selection algorithms and number of features.

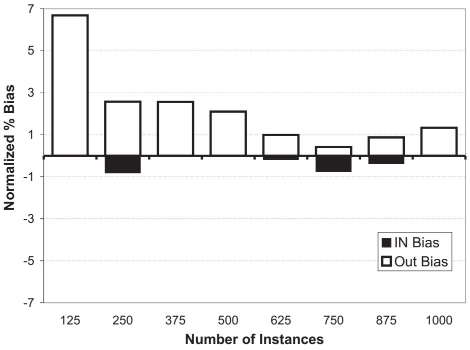

Figure 4: Normalized bias for IN and OUT on non-linear synthetic data averaged over all factors

Formally, bias is defined as the expected value for the difference between an estimator and the true value of the parameter being estimated. Suppose we are trying to estimate some parameter θ using some estimator $\hat{\theta}$. In such a case $\hat{\theta}$'s bias is given by:

$$Bias = E(\hat{\theta} - \theta)$$

where E is the expected value (average over many estimates). If $\hat{\theta}$ has a tendency to under-estimate, the bias will be negative, and if it has a tendency to over-estimate, the bias will be positive. For accuracy estimation, θ is the true mean accuracy and $\hat{\theta}$ is the average accuracy obtained by CV after IN or OUT. Unfortunately we do not know the true mean accuracy. To approximate the bias, we can use the values from method TRUTH but the problem is that method TRUTH yields a distribution, not a fixed value. To deal with this issue we utilized the following method.

For each experiment we generated 100 datasets, yielding 100 TRUTH accuracy estimates. We also performed 10-fold CV on *each* of these datasets, yielding 100 sets of 10 accuracy estimates for IN and another 100 sets for OUT. To simulate the bias we perform a two-sample hypothesis test, comparing each of the 100 sets of 10 accuracy estimates with the 100 accuracy estimates obtained using method TRUTH. If the null hypothesis is rejected we conclude that the CV results are biased, otherwise we assume that cross validation demonstrated no bias. We used the version of the Student's t-test intended for use with samples with unequal variances (L.Welch 1947). We observed that the variance is much lower for TRUTH, likely due to the much larger sample size and the much larger test set; and OUT has slightly lower variance than IN. We track the bias for IN and OUT on each CV experiment and average them over all 100 sets. The obtained bias is an absolute accuracy bias, in order to make it comparable across different factors such as number of instances, we normalized the bias by dividing it by the average TRUTH accuracy. We observed that OUT indeed

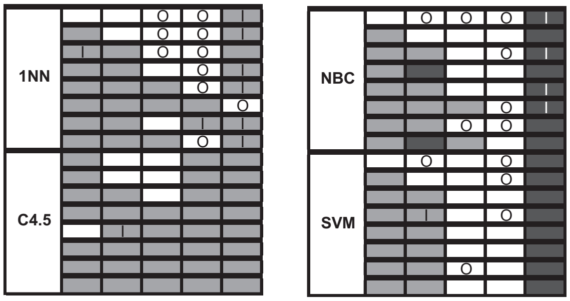

Figure 5: Results for comparing FCBF vs. ReliefF on the synthetic linear data. According to the binomial test: I - IN was better, O - OUT was better. According to the method TRUTH: black - ReliefF is better, white - FCBF is better, gray - there is no difference.

demonstrated a positive bias, while IN demonstrated a negative bias. Figure 4 shows the normalized bias for non-linear synthetic data averaged over all factors. Notice that the bias for IN is consistently small (below 1%), while the bias for OUT is quite large for small number of instances. We observed similar trends with other data sources. IN's bias was never observed to exceed 5% while OUT's bias reached 20% for some cases!

Which Method is More Truthful?

We now look at the central question about which method (IN or OUT) is better when comparing two feature selection algorithms. Since our experimental setup yielded 100 paired conclusions for IN and OUT, we further determined the truth about which of the two FS algorithms is better. In order to determine if one method is significantly better than the other, we count the frequency of two events:
I : the number of times that IN is truthful but OUT is not
O : the number of times that OUT is truthful but IN is not
If IN and OUT are the same we would expect these two events to be opposing events of equal probability in Bernoulli trials. We assume that IN and OUT are the same:

$$H_0 : \quad E_{IN} = E_{OUT}$$

Given a particular pair of frequencies for I and O, we can calculate the exact probability that these frequencies came from the described binomial process by summing over one tail of the binomial distribution.

$$p = \sum_{k=0}^{\min(I,O)} \binom{I+O}{k} \left(\frac{1}{2}\right)^{k} \left(\frac{1}{2}\right)^{I+O-k}$$

If this exact probability (p) that the null hypothesis is true is sufficiently low we can reject the null hypothesis in favor of one of these alternatives:

$$H_1 : \begin{cases} E_{IN}\ better\,than\ E_{OUT} & if\ (p < 0.5\alpha) \wedge (I > O) \\ E_{OUT}\ better\,than\ E_{IN} & if\ (p < 0.5\alpha) \wedge (O > I) \end{cases}$$

Table 2: Confusion Matrices for comparing FS algorithms on Synthetic Data. I - IN is better, O - OUT is better, S - no significant difference between IN and OUT, W - Win (the first FS algorithm is better), L - Loss (the second FS algorithm is better), D - Draw (no significant difference).

	FCBF vs. IG			FCBF vs. ReliefF			ReliefF vs. IG		
	I	S	O	I	S	O	I	S	O
				Linear					
W	0	1	0	4	14	0	0	36	21
D	0	153	0	10	75	0	9	73	0
L	0	6	0	0	35	22	4	17	0
				Non-Linear					
W	0	0	0	2	15	0	0	54	13
D	0	154	0	12	63	0	12	63	0
L	0	6	0	1	54	13	2	16	0
				MONK1					
W	0	0	0	0	43	1	5	47	15
D	1	151	0	0	47	0	0	49	0
L	0	7	0	6	48	15	0	43	1
				MONK2					
W	0	1	0	0	7	0	1	33	5
D	0	158	0	2	112	0	4	110	0
L	0	1	0	2	33	4	0	7	0
				MONK3					
W	0	4	0	0	7	0	0	6	0
D	0	153	0	1	146	0	0	144	1
L	0	3	0	0	6	0	0	9	0

After compiling the results, it is interesting to discover that neither IN nor OUT was superior in all cases. We present the results for comparing FCBF with ReliefF on the synthetic linear data in Figure 5. This figure shows the results for all classifiers, and all the varied numbers of instances and features. Within each classifier, the number of instances increases from top to bottom and the number of selected features increases from left to right. The cells marked with O are cases where the binomial test indicated that OUT was the better method, while the cells marked with I are cases where the binomial test indicated that IN was the better method. As we can see there are many cases where OUT was the better method. We overlayed this figure with the TRUTH conclusion and discovered a pattern. In this figure the black cells are cases where FCBF was in fact the better method, white cells are cases where ReliefF was in fact the better method and gray cells are cases where there was no significant difference. As you can see there are more O's in the white regions, indicating that when ReliefF is the winner, OUT is better at detecting it, and there are more I's in the gray and black regions, indicating that when ReliefF is the loser or is the same as its competitor, IN is better at detecting it. We observed a similar trend with respect to ReliefF in other data sources except for MONK3. Table 2 shows a summary of these results.

The results indicate that some FS algorithms are sensitive to the evaluation method and some are not. When comparing FCBF and IG, IN and OUT were determined to be the same in nearly all cases and across all data sources; however, when comparing ReliefF with another algorithm, it is more

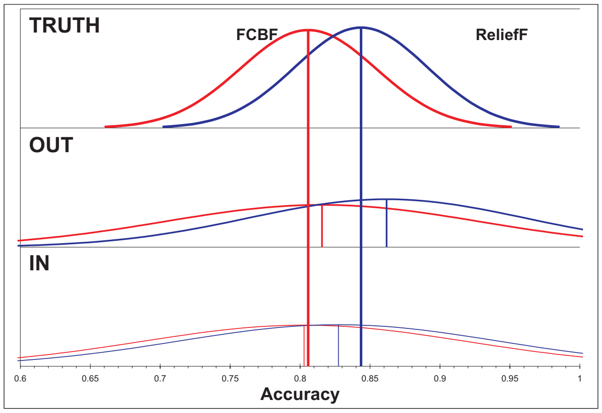

Figure 6: An example demonstrating the effect of bias imbalance. (SVM accuracy on 125 instances of synthetic linear data suing FCBF and ReliefF to select 20 features)

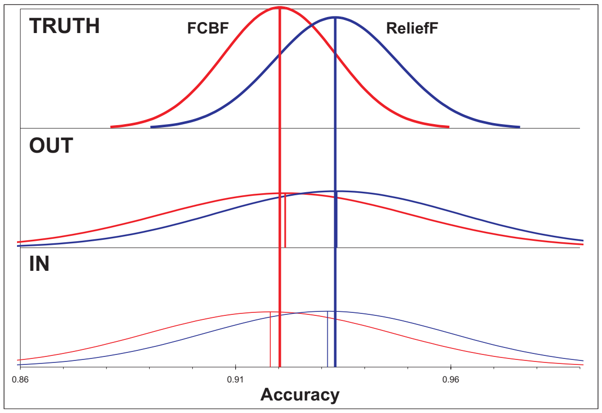

Figure 7: SVM accuracy on 1000 instances of synthetic linear data using FCBF and ReliefF to select 20 features

likely that one method (IN or OUT) is better than the other. In cases where there was in fact no difference between the two FS algorithms, OUT was more likely to be wrong by saying that there is a difference. An even more interesting finding is that when there was a difference between the two FS algorithms, no single evaluation method was consistently better than the other. Instead, we observed an asymmetric pattern, where the sensitive FS algorithm (ReliefF) seemed to consistently favor one evaluation method (OUT) over the other, when compared against different FS algorithms and across different data sources. Conversely when this algorithm lost to its competitor, the other evaluation method (IN) was superior across different factors. We further observed that for each outcome of method TRUTH, the superiority of one method over another does not have any correlation with the number of instances. The only exception was the case of a draw between two algorithms. In this case we found that even though IN was more often the better method its superiority decreased with increasing instances.

How Can OUT be better?

We have demonstrated that both IN and OUT are biased in accuracy estimation, but OUT's bias is often larger in magnitude than that of IN. The mystery then becomes how can OUT ever be the better method? We examined this issue more closely and discovered that the superiority of IN or OUT has less to do with their bias and more to do with their *bias imbalance*. We define bias imbalance as the absolute difference in the bias that a method exerts on the two FS algorithms under comparison. When two FS algorithms are in fact different, a bias imbalance can have a positive effect when the bias imbalance causes the measured mean accuracy rates to be driven further apart; or it can have a negative effect when the bias imbalance causes the mean accuracy rates to be driven closer together. Generally, we would like the bias imbalance to be as small as possible, as to ensure accurate comparison of two FS algorithms.

Consider the example shown in Figure 6: with OUT, FCBF and ReliefF are visibly separable, whereas with IN a hypothesis test may incorrectly conclude that they come from the same distribution. The vertical lines show the means, also summarized in Table 3. Though OUT shows a larger bias than IN, particularly for ReliefF, IN shows a larger *bias imbalance*, making OUT the better method in this case. Remarkably, such bias imbalance patterns seem consistent across the data, namely OUT having a lower bias imbalance when ReliefF is superior and IN having the lower imbalance when ReliefF is inferior or equivalent to its competitor. Figure 7 shows an example where IN and OUT both arrive at the correct conclusion. Although IN's mean difference of FCBF and ReliefF is smaller than that in Figure 6 ($\sim 1\%$ as compared to $\sim 4\%$), the two distributions are more distinguishable because (1) bias imbalance is not very large, and (2) their variances are smaller (likely due to a larger number of instances).

Related Work

Reunanen presents findings about evaluation methods for FS when *using wrapper models*, pointing out that the wrapper accuracy on *training data* using leave-one-out CV is not necessarily consistent with that on the independent *test set*. It does not address issues specific to pairwise comparison of FS algorithms, and it studies only wrapper models. Our work concentrates on filter models. We study how to conduct pairwise comparison of FS algorithm using 10-fold CV with paired t-test. 10-fold CV is recommended as it tends to provide less biased estimation of the accuracy (Kohavi 1995). As suggested in (Salzberg 1997), when comparing two or more learning algorithms, appropriate hypothesis testing should be performed instead of comparing only average accuracy of k-fold CV. Paired t-test is one such test taking into account the variance in accuracy estimates (Dietterich 1998), and is often used in machine learning.

Alternatives to 10-fold CV can be found in literature: (a) (Dietterich 1998) recommends 5×2-fold CV for its low Type I error; (b) Since a high degree of sample overlap in a typical CV can lead to an underestimated variance of

Table 3: Mean Accuracy and Bias Imbalance Summary for Figure 6

	FCBF		ReliefF		t-test ...		
	Accuracy	Bias	Accuracy	Bias	concludes	is truthful?	Bias Imbalance
OUT	0.8156	0.0098	0.8619	0.0183	ReliefF better	Yes	0.0085
TRUTH	0.8058	n/a	0.8436	n/a	ReliefF better	n/a	n/a
IN	0.8028	-0.003	0.8274	-0.0162	No difference	No	0.0132

performance difference, (Nadeau & Bengio 1999) proposes to correct the sample variance by scaling it with a ratio based on the number of instances in training and test data; (c)(Bouckaert 2003) suggests 10 $\times$ 10-fold CV followed by adjusted paired t-test on the resulting 100 samples using 10 degrees of freedom; and (d) (Bouckaert 2004) demonstrates that "sorted runs sampling" scheme followed with t-test outperforms other evaluation schemes if the replicability of an evaluation result is considered. In theory, one can replace 10-fold CV with any one of the above. We focus on 10-fold CV because no matter what kind of bias each of the above methods has, these biases would affect the ranking of compared algorithms in a similar way. For example, 5 $\times$ 2-fold CV makes each training fold smaller, IN could be more conservative than using 10-fold CV; as OUT still uses the full training data for FS, it remains optimistic. The fact that many past FS studies employed 10-fold CV in their evaluation has also influenced our choice of 10-fold CV.

Conclusions

This study results in the following findings: (1) IN and OUT have different biases, and bias is not a major factor in determining whether IN or OUT is more truthful in pair-wise comparison; (2) some FS algorithms are sensitive to FS evaluation methods; (3)for the greater majority of cases, IN and OUT are not significantly different; (4) IN and OUT almost never give completely opposite conclusions; (5) when two FS algorithms perform identically, IN is often a better method to indicate so; and (6) for other two cases where (a) A_1 is better and (b) A_2 is better, if IN is better for case (a), then OUT is better for case (b).

If the end goal of the pair-wise comparison is to show that a new algorithm is superior to some baseline algorithm, and we wish to minimize our chance of making a mistake, we recommend using IN. Since we do not know which of the three outcomes (A_1 is better or A_2 is better or there is no difference between A_1 and A_2) is most probable, assuming that they are equally likely, we recommend method IN to get the edge in two out of the three cases. Before running experiments with real-world data, we can also consider first running experiments with synthetic data using two FS algorithms we plan to compare, as to observe their sensitivity to the evaluation methods.

On the other hand, if our end goal is to select the best subset of features for a particular dataset, we recommend to run both methods IN and OUT, trust the method indicating that one algorithm is better than the other, and use that better algorithm to select features using the entire dataset. In the worst case scenario, the selected features will be no worse than the subset selected by the alternative algorithm.

References

Blum, A. L., and Langley, P. 1997. Selection of relevant features and examples in machine learning. *AI* 97:245–271.

Bouckaert, R. R. 2003. Choosing between two learning algorithms based on calibrated tests. In *ICML*.

Bouckaert, R. R. 2004. Estimating replicability of classifier learning experiments. In *ICML*.

Dietterich, T. G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* 10(7):1895–1923.

Dy, J. G., and Brodley, C. E. 2004. Feature selection for unsupervised learning. *JMLR* 5:845–889.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *JMLR* 3:1157–1182.

Jain, A., and Zongker, D. 1997. Feature selection: Evaluation, application, and small sample performance. *IEEE TPAMI* 19(2):153–158.

John, G. H.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant feature and the subset selection problem. In *ICML*.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*.

Kononenko, I. 1994. Estimating attributes: Analysis and extensions of relief. In *ECML*, 171–182.

Liu, H., and Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE TKDE* 17:491–502.

L.Welch, B. 1947. The generalization of 'student's' problem when several different population variances are involved. *Biometrika* 34:28–35.

Molla, M.; Waddell, M.; Page, D.; and Shavlik, J. 2004. Using machine learning to design and interpret gene-expression microarrays. *AI Mag.* 25(1):23–44.

Nadeau, C., and Bengio, Y. 1999. Inference for the generalization error. In *NIPS*.

Reunanen, J. 2003. Overfitting in making comparisons between variable selection methods. *JMLR* 3:1371–1382.

Salzberg, S. 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery* 1(3):317–328.

Thrun, S. B.; etal. 1991. The MONK's problems: A performance comparison of different learning algorithms. Technical Report CS-91-197, Pittsburgh, PA.

van 't Veer, L. J.; etal. 2002. gene expression profiling predicts clinical outcome of breast cancer. *Nature*.

Yu, L., and Liu, H. 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *ICML*