# Robust Feature Selection Using Ensemble Feature Selection Techniques

Yvan Saeys, Thomas Abeel, and Yves Van de Peer

Department of Plant Systems Biology, VIB, Technologiepark 927, 9052 Gent,
Belgium and Department of Molecular Genetics, Ghent University, Gent, Belgium
{yvan.saeys,thomas.abeel,yves.vandepeer}@psb.ugent.be

**Abstract.** Robustness or stability of feature selection techniques is a topic of recent interest, and is an important issue when selected feature subsets are subsequently analysed by domain experts to gain more insight into the problem modelled. In this work, we investigate the use of ensemble feature selection techniques, where multiple feature selection methods are combined to yield more robust results. We show that these techniques show great promise for high-dimensional domains with small sample sizes, and provide more robust feature subsets than a single feature selection technique. In addition, we also investigate the effect of ensemble feature selection techniques on classification performance, giving rise to a new model selection strategy.

## 1 Introduction

Feature selection is an important preprocessing step in many machine learning applications, where it is often used to find the smallest subset of features that maximally increases the performance of the model. Besides maximizing model performance, other benefits of applying feature selection include the ability to build simpler and faster models using only a subset of all features, as well as gaining a better understanding of the processes described by the data, by focusing on a selected subset of features [1].

Feature selection techniques can be divided into three categories, depending on how they interact with the classifier. Filter methods directly operate on the dataset, and provide a feature weighting, ranking or subset as output. These methods have the advantage of being fast and independent of the classification model, but at the cost of inferior results. Wrapper methods perform a search in the space of feature subsets, guided by the outcome of the model (e.g. classification performance on a cross-validation of the training set). They often report better results than filter methods, but at the price of an increased computational cost [2]. Finally, embedded methods use internal information of the classification model to perform feature selection (e.g. use of the weight vector in support vector machines). They often provide a good trade-off between performance and computational cost [1,3].

During the past decade, the use of feature selection for knowledge discovery has become increasingly important in many domains that are characterized by

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 313–325, 2008.

a large number of features, but a small number of samples. Typical examples of such domains include text mining, computational chemistry and the bioinformatics and biomedical field, where the number of features (problem dimensionality) often exceeds the number of samples by orders of magnitude [3]. When using feature selection in these domains, not only model performance but also robustness of the feature selection process is important, as domain experts would prefer a stable feature selection algorithm over an unstable one when only small changes are made to the dataset. Robust feature selection techniques would allow domain experts to have more confidence in the selected features, as in most cases these features are subsequently analyzed further, requiring much time and effort, especially in biomedical applications.

Surprisingly, the robustness (stability) of feature selection techniques is an important aspect that received only relatively little attention during the past. Recent work in this area mainly focuses on the stability indices to be used for feature selection, introducing measures based on Hamming distance [4], correlation coefficients [5], consistency [6] and information theory [7]. Kalousis and coworkers also present an extensive comparative evaluation of feature selection stability over a number of high-dimensional datasets [5]. However, most of this work only focuses on the stability of single feature selection techniques, an exception being the work of [4] which describes an example combining multiple feature selection runs.

In this work, we investigate whether the use of ensemble feature selection techniques can be used to yield more robust feature selection techniques, and whether combining multiple methods has any effect on the classification performance. The rationale for this idea stems from the field of ensemble learning, where multiple (unstable) classifiers are combined to yield a more stable, and better performing ensemble classifier. Similarly, one could think of more robust feature selection techniques by combining single, less stable feature selectors. As this issue is especially critical in large feature/small sample size domains, the current work focuses on ensemble feature selection techniques in this area.

The rest of this paper is structured as follows. Section 2 introduces the methodology used to assess robustness of the algorithms we evaluated. Subsequently, we introduce ensemble feature selection techniques in section 3 and present the results of our experiments in section 4. We conclude with some final remarks and ideas for future work.

## 2 Robustness of Feature Selection Techniques

The robustness of feature selection techniques can be defined as the variation in feature selection results due to small changes in the dataset. When applying feature selection for knowledge discovery, robustness of the feature selection result is a desirable characteristic, especially if subsequent analyses or validations of selected feature subsets are costly. Modification of the dataset can be considered at different levels: perturbation at the instance level (e.g. by removing or adding samples), at the feature level (e.g. by adding noise to features), or a combination of both. In the current work, we focus on perturbations at the instance level.

### 2.1 Estimating Stability with Instance Perturbation

To measure the effect of instance perturbation on the feature selection results, we adopt a subsampling based strategy. Consider a dataset $\mathcal{X} = \{x_1, \ldots, x_M\}$ with $M$ instances and $N$ features. Then $k$ subsamples of size $\lceil xM \rceil$ $(0 < x < 1)$ are drawn randomly from $\mathcal{X}$, where the parameters $k$ and $x$ can be varied. Subsequently, feature selection is performed on each of the $k$ subsamples, and a measure of stability or robustness is calculated. Here, following [5], we take a similarity based approach where feature stability is measured by comparing the outputs of the feature selectors on the $k$ subsamples. The more similar all outputs are, the higher the stability measure will be. The overall stability can then be defined as the average over all pairwise similarity comparisons between the different feature selectors:

$$S_{\text{tot}} = \frac{2 \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{f}_i, \mathbf{f}_j)}{k(k-1)}$$

where $\mathbf{f}_i$ represents the outcome of the feature selection method applied to subsample $i$ $(1 \leq i \leq k)$, and $S(\mathbf{f}_i, \mathbf{f}_j)$ represents a similarity measure between $\mathbf{f}_i$ and $\mathbf{f}_j$.

To generalize this approach to all feature selection methods, it has to be noted that not all feature selection techniques report their result in the same form, and we can distinguish between feature weighting, feature ranking and feature subset selection. Evidently, a feature weighting can be converted to a feature ranking by sorting the weights, and a ranking can be converted to a feature subset by choosing an appropriate threshold. For the remainder of the paper, we choose the similarity function $S(.,.)$ to compare only results of the same type. Thus, $\mathbf{f}_i$ can be considered a vector of length $N$, where $\mathbf{f}_i^j$ represents (a) the weight for feature $j$ in the case of comparing feature weightings, (b) the rank of feature $j$ in the case of feature ranking (the worst feature is assigned rank 1, the best one rank $N$) and (c) $\mathbf{f}_i^j = 1$ if the feature is present in the subset, and zero otherwise in the case of feature subset selection.

### 2.2 Similarity Measures

Appropriate similarity measures for feature weighting, ranking and subset selection can be derived from different correlation coefficients. For feature weighting, the Pearson correlation coefficient can be used:

$$S(\mathbf{f}_i, \mathbf{f}_j) = \frac{\sum_l (\mathbf{f}_i^l - \mu_{\mathbf{f}_i})(\mathbf{f}_j^l - \mu_{\mathbf{f}_j})}{\sqrt{\sum_l (\mathbf{f}_i^l - \mu_{\mathbf{f}_i})^2 \sum_l (\mathbf{f}_j^l - \mu_{\mathbf{f}_j})^2}}$$

For feature ranking, the Spearman rank correlation coefficient can be used:

$$S(\mathbf{f}_i, \mathbf{f}_j) = 1 - 6 \sum_l \frac{(\mathbf{f}_i^l - \mathbf{f}_j^l)^2}{N(N^2 - 1)}$$

For feature subsets, we use the Jaccard index:

$$S(\mathbf{f}_i, \mathbf{f}_j) = \frac{|\mathbf{f}_i \cap \mathbf{f}_j|}{|\mathbf{f}_i \cup \mathbf{f}_j|} = \frac{\sum_l I(\mathbf{f}_i^l = \mathbf{f}_j^l = 1))}{\sum_l I(\mathbf{f}_i^l + \mathbf{f}_j^l > 0)}$$

where the indicator function $I(.)$ returns 1 if its argument is true, and zero otherwise.

Finally, it is important to note that robustness of feature selection results should not be considered per se, but always in combination with classification performance, as domain experts are not interested in a strategy that yields very robust feature sets, but returns a badly performing model. Hence, these two aspects need always be investigated together.

## 3 Ensemble Feature Selection Techniques

In ensemble learning, a collection of single classification or regression models is trained, and the output of the ensemble is obtained by aggregating the outputs of the single models, e.g. by majority voting in the case of classification, or averaging in the case of regression. Dietterich [8] shows that the result of the ensemble might outperform the single models when weak (unstable) models are combined, mainly because of three reasons: a) several different but equally optimal hypotheses can exist and the ensemble reduces the risk of choosing a wrong hypothesis, b) learning algorithms may end up in different local optima, and the ensemble may give a better approximation of the true function, and c) the true function cannot be represented by any of the hypotheses in the hypothesis space of the learner and by aggregating the outputs of the single models, the hypothesis space may be expanded.

### 3.1 The Ensemble Idea for Feature Selection

Similar to the case of supervised learning, ensemble techniques might be used to improve the robustness of feature selection techniques. Indeed, in large feature/small sample size domains it is often reported that several different feature subsets may yield equally optimal results [3], and ensemble feature selection may reduce the risk of choosing an unstable subset. Furthermore, different feature selection algorithms may yield feature subsets that can be considered local optima in the space of feature subsets, and ensemble feature selection might give a better approximation to the optimal subset or ranking of features. Finally, the representational power of a particular feature selector might constrain its search space such that optimal subsets cannot be reached. Ensemble feature selection could help in alleviating this problem by aggregating the outputs of several feature selectors.

### 3.2 Components of Ensemble Feature Selection

Similar to the construction of ensemble models for supervised learning, there are two essential steps in creating a feature selection ensemble. The first step involves creating a set of different feature selectors, each providing their output, while the second step aggregates the results of the single models. Variation in the feature selectors can be achieved by various methods: choosing different feature selection techniques, instance level perturbation, feature level perturbation, stochasticity in the feature selector, Bayesian model averaging, or combinations of these techniques [8,9]. Aggregating the different feature selection results can be done by weighted voting, e.g. in the case of deriving a consensus feature ranking, or by counting the most frequently selected features in the case of deriving a consensus feature subset.

In this work, we focus on ensemble feature selection techniques that work by aggregating the feature rankings provided by the single feature selectors into a final consensus ranking. Consider an ensemble $E$ consisting of $s$ feature selectors, $E = \{F_1, F_2, \ldots, F_s\}$, then we assume each $F_i$ provides a feature ranking $\mathbf{f}_i = (f_i^1, \ldots, f_i^N)$, which are aggregated into a consensus feature ranking $\mathbf{f}$ by weighted voting:

$$\mathbf{f}^l = \sum_{i=1}^{s} w(\mathbf{f}_i^l)$$

where $w(.)$ denotes a weighting function. If a *linear aggregation* is performed using $w(\mathbf{f}_i^l) = \mathbf{f}_i^l$, this results in a sum where features contribute in a linear way with respect to their rank. By modifying $w(\mathbf{f}_i^l)$, more or less weight can be put to the rank of each feature. This can be e.g. used to accommodate for rankings where top features can be forced to influence the ranking significantly more than lower ranked features.

## 4 Experiments

In this section, we present the results of our analysis of ensemble feature selection techniques on large feature/small sample size domains. First, the data sets and the feature selection techniques used in this analysis are briefly described. Subsequently, we analyze two aspects of ensemble feature selection techniques: robustness and classification performance. All experiments were run using Java-ML[1], an open source machine learning library.

### 4.1 Data Sets

Datasets were taken from the bioinformatics and biomedical domain, and can be divided into two parts: microarray datasets (MA) and mass spectrometry (MS) datasets (Table 1). For each domain, three datasets were included, typically consisting of several thousands of features and tens of instances in the case of microarray datasets, and up to about 15,000 features and a few hundred of instances in the case of mass spectrometry datasets. Due to their high dimensionality and low sample size, these datasets pose a great challenge for both classification and feature selection algorithms. Another important aspect

[1] Available at `http://java-ml.sourceforge.net`

**Table 1.** Data set characteristics. Sample to dimension rate (SDR) is calculated as $100M/N$.

| | Name | # Class 1 | # Class 2 | # Features | SDR | Reference |
|---|---|---|---|---|---|---|
| MA | Colon | 40 | 22 | 2000 | 3.1 | [15] |
| | Leukemia | 47 | 25 | 7129 | 1.0 | [16] |
| | Lymphoma | 22 | 23 | 4026 | 1.1 | [17] |
| MS | Ovarian | 162 | 91 | 15154 | 1.7 | [18] |
| | Prostate | 69 | 253 | 15154 | 2.1 | [19] |
| | Pancreatic | 80 | 101 | 6771 | 2.7 | [20] |

of this data is the fact that the outcome of feature selection techniques is an essential prerequisite for further validation, such as verifying links between particular genes and diseases. Therefore, domain experts require the combination of feature selection and classification algorithm to yield both a high accuracy as well as robustness of the selected features.

### 4.2 Feature Selection Techniques

In this work, we focus on the application of filter and embedded feature selection techniques. We discarded wrapper approaches because they commonly require on the order of $N^2$ classification models being built if a complete ranking of $N$ features is desired. Filter methods require no model being built, and embedded models only build a small amount of models. Thus, the wrapper approach, certainly when used in the ensemble setting is computationally not feasible for the large feature sizes we are dealing with. We choose a benchmark of four feature selection techniques: two filter methods and two embedded methods. For the filter methods, we selected one univariate and one multivariate method. Univariate methods consider each feature separately, while multivariate methods take into account feature dependencies, which might yield better results. The univariate method we choose was the Symmetrical Uncertainty (SU, [10]):

$$SU(F,C) = 2\frac{H(F) - H(F|C)}{H(F) + H(C)}$$

where $F$ and $C$ are random variables representing a feature and the class respectively, and the function $H$ calculates the entropy. As a multivariate method, we choose the RELIEF algorithm [11], which estimates the relevance of features according to how well their values distinguish between the instances of the same and different classes that are near each other. Furthermore, the computational complexity of RELIEF $\mathcal{O}(MN)$ scales well to large feature/small sample size data sets, compared to other multivariate methods which are often quadratic in the number of features. In our experiments, five neighboring instances were chosen. When using real-valued features, equal frequency binning was used to discretize the features.

As embedded methods we used the feature importance measures of Random Forests [12] and linear support vector machines (SVM). In a Random Forest (RF), feature importance is measured by randomly permuting the feature in the out-of-bag samples and calculating the percent increase in misclassification rate as compared to the out-of-bag rate with all variables intact. In our feature selection experiments we used forests consisting of 10 trees.

For a linear SVM, the feature importance can be derived from the weight vector of the hyperplane [13], a procedure known as recursive feature elimination (SVM_RFE). In this work, we use SVM_RFE as a feature ranker: first a linear SVM is trained on the full feature set, and the $C$-parameter is tuned using an internal cross-validation of the training set. Next, features are ranked according to the absolute value of their weight in the weight vector of the hyperplane, and the 10% worst performing features are discarded. The above procedure is then repeated until the empty feature set is reached.

### 4.3 Ensemble Feature Selection Techniques

For each of the four feature selection techniques described above, an ensemble version was created by instance perturbation. We used bootstrap aggregation (bagging, [14]) to generate 40 bags from the data. For each of the bags, a separate feature ranking was performed, and the ensemble was formed by aggregating the single rankings by weighted voting, using linear aggregation.

### 4.4 Robustness of Feature Selection

To assess the robustness of feature selection techniques, we focus here on comparing feature rankings and feature subsets, as these are most often used by domain experts. Feature weightings are almost never used, and instead converted to a ranking or subset. Furthermore, directly comparing feature weights may be problematic as different methods may use different scales and intervals for the weights.

To compare feature rankings, the Spearman rank correlation coefficient was used, while for feature subsets the Jaccard index was used. The last one was analyzed for different subset sizes: the top 1% and top 5% best features of the rankings were chosen.

To estimate the robustness of feature selection techniques, the strategy explained in section 2.1 was used with $k = 10$ subsamples of size $0.9M$ (i.e. each subsample contains 90% of the data). This percentage was chosen because we use small sample datasets and thus cannot discard too much data when building models, and further because we want to assess robustness with respect to relatively small changes in the dataset. Then, each feature selection algorithm (both the single and the ensemble version) was run on each subsample, and the results were averaged over all pairwise comparisons.

Table 2 summarizes the results of the robustness analysis across the different datasets, using the linear aggregation method for ensemble feature selection. For each feature selection algorithm, the Spearman correlation coefficient (Sp)

**Table 2.** Robustness of the different feature selectors across the different datasets. Spearman correlation coefficient, Jaccard index on the subset of 1% and 5% best features are denoted respectively by Sp, JC1 and JC5.

| Dataset | | SU | | Relief | | SVM_RFE | | RF | |
|---|---|---|---|---|---|---|---|---|---|
| | | Single | Ensemble | Single | Ensemble | Single | Ensemble | Single | Ensemble |
| Colon | Sp | 0.61 | 0.76 | 0.62 | 0.85 | 0.7 | 0.81 | 0.91 | 0.99 |
| | JC5 | 0.33 | 0.49 | 0.44 | 0.64 | 0.47 | 0.45 | 0.44 | 0.79 |
| | JC1 | 0.3 | 0.55 | 0.45 | 0.56 | 0.44 | 0.5 | 0.01 | 0.64 |
| Leukemia | Sp | 0.68 | 0.76 | 0.58 | 0.79 | 0.73 | 0.79 | 0.97 | 0.99 |
| | JC5 | 0.48 | 0.57 | 0.39 | 0.54 | 0.53 | 0.58 | 0.8 | 0.91 |
| | JC1 | 0.54 | 0.6 | 0.44 | 0.55 | 0.49 | 0.57 | 0.36 | 0.8 |
| Lymphoma | Sp | 0.59 | 0.74 | 0.49 | 0.76 | 0.77 | 0.81 | 0.96 | 0.99 |
| | JC5 | 0.31 | 0.49 | 0.35 | 0.53 | 0.54 | 0.54 | 0.74 | 0.9 |
| | JC1 | 0.37 | 0.55 | 0.42 | 0.56 | 0.43 | 0.46 | 0.22 | 0.73 |
| Ovarian | Sp | 0.93 | 0.95 | 0.91 | 0.97 | 0.91 | 0.95 | 0.96 | 0.99 |
| | JC5 | 0.76 | 0.79 | 0.66 | 0.78 | 0.75 | 0.79 | 0.7 | 0.93 |
| | JC1 | 0.84 | 0.85 | 0.85 | 0.88 | 0.8 | 0.84 | 0.1 | 0.83 |
| Pancreatic | Sp | 0.57 | 0.65 | 0.46 | 0.73 | 0.69 | 0.77 | 0.9 | 0.99 |
| | JC5 | 0.2 | 0.24 | 0.16 | 0.3 | 0.43 | 0.41 | 0.52 | 0.76 |
| | JC1 | 0.13 | 0.15 | 0.09 | 0.19 | 0.41 | 0.36 | 0.01 | 0.48 |
| Prostate | Sp | 0.88 | 0.91 | 0.9 | 0.96 | 0.81 | 0.92 | 0.96 | 0.99 |
| | JC5 | 0.68 | 0.7 | 0.61 | 0.71 | 0.6 | 0.63 | 0.72 | 0.88 |
| | JC1 | 0.67 | 0.7 | 0.52 | 0.64 | 0.6 | 0.6 | 0.13 | 0.78 |
| **Average** | **Sp** | **0.71** | **0.8** | **0.66** | **0.84** | **0.77** | **0.84** | **0.94** | **0.99** |
| | **JC5** | **0.46** | **0.55** | **0.44** | **0.58** | **0.55** | **0.57** | **0.65** | **0.86** |
| | **JC1** | **0.47** | **0.57** | **0.46** | **0.57** | **0.53** | **0.56** | **0.14** | **0.71** |

and Jaccard index on the subset of 1% (JC1) and 5% best features (JC5) are shown. In general, it can be observed that ensemble feature selection provides more robust results than a single feature selection algorithm, the difference in robustness being dependent on the dataset and the algorithm.

RELIEF is one of the less stable algorithms, but clearly benefits from an ensemble version, as well as the Symmetrical Uncertainty filter method. SVM_RFE on the other hand proves to be a more stable feature selection method, and creating an ensemble version of this method only slightly improves robustness. For Random Forests, the picture is a bit more complicated. While for Sp and JC5, a single Random Forest seems to outperform the other methods, results are much worse on the JC1 measure. This means that the very top performing features vary a lot with regard to different data subsamples. Especially for knowledge discovery, the high variance in the top selected features by Random Forests may be a problem. However, also Random Forests clearly benefit from an ensemble version, the most drastic improvement being made on the JC1 measure. Thus, it seems that ensembles of Random Forests clearly outperform other feature selection methods regarding robustness.

The effect of the number of feature selectors on the robustness of the ensemble is shown in Figure 1. In general, robustness is mostly increased in the first steps, and slows down after about 20 selectors in the ensemble, an exception being the Random Forest. In essence, a single Random Forest can already be seen as an ensemble feature selection technique, averaging over the different trees in the forest, which can explain the earlier convergence of ensembles of Random Forests.

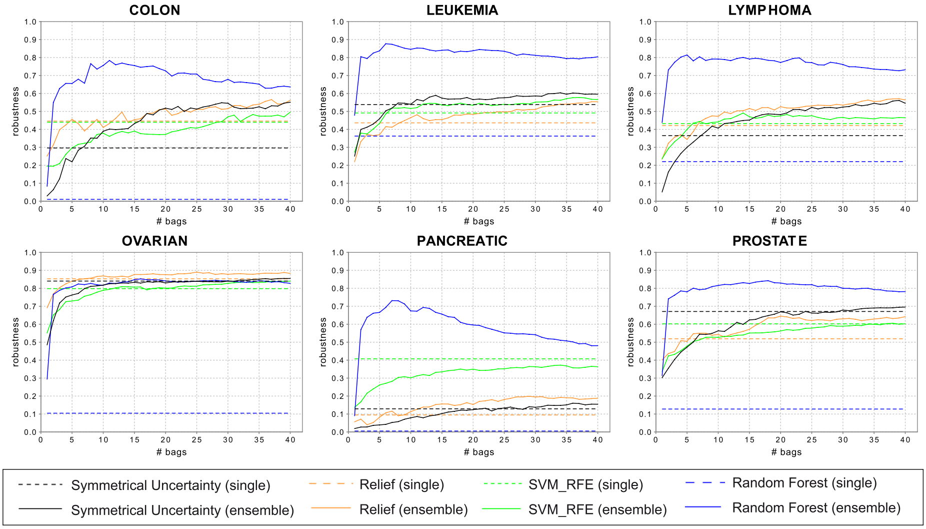

**Fig. 1.** Robustness in function of the ensemble size. Robustness is measured using the Jaccard index on the 1% top ranked features.

One could wonder to what extent an ensemble of Random Forests would be comparable to just one single Random Forest consisting of more trees. Preliminary experiments on the datasets analysed in this work suggest that larger Random Forests often lead to less robust results than smaller ones. Hence, if robust feature selection results are desired, it would be computationally cheaper to average over a number of small Random Forests in an ensemble way, rather than creating one larger Random Forest.

### 4.5 Robustness Versus Classification Performance

Considering only robustness of a feature selection technique is not an appropriate strategy to find good feature rankings or subsets, and also model performance should be taken into account to decide which features to select. Therefore, feature selection needs to be combined with a classification model in order to get an estimate of the performance of the feature selector-classifier combination. Embedded feature selection methods like Random Forests and SVM_RFE have an important computational advantage in this respect, as they combine model construction with feature selection.

To analyze the effect on classification performance using single versus ensemble feature selection, we thus set up a benchmark on the same datasets as used to assess robustness. Due to their capacity to provide a feature ranking, as well as their state-of-the-art performance, Random Forests and linear SVMs were included as classifiers, as well as the distance based k-nearest neighbor algorithm (KNN). The number of trees in the Random Forest classifier was set to 50, and the number of nearest neighbors for KNN was set to 5.

**Table 3.** Performance comparison for the different feature selector-classifier combinations. Each entry in the table represents the average accuracy using 10-fold cross-validation.

| Dataset | | SU | | RELIEF | | SVM_RFE | | RF | | All |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Single | Ensemble | Single | Ensemble | Single | Ensemble | Single | Ensemble | features |
| Colon | SVM | 0.87 | 0.89 | 0.91 | 0.91 | 0.93 | 0.96 | 0.87 | 0.74 | 0.87 |
| | RF | 0.89 | 0.91 | 0.86 | 0.89 | 0.8 | 0.86 | 0.79 | 0.67 | 0.79 |
| | KNN | 0.81 | 0.87 | 0.87 | 0.87 | 0.87 | 0.94 | 0.83 | 0.7 | 0.79 |
| Leukemia | SVM | 0.98 | 0.96 | 0.98 | 0.99 | 1.0 | 0.99 | 0.91 | 0.88 | 1.0 |
| | RF | 1.0 | 0.99 | 1.0 | 1.0 | 0.98 | 0.98 | 0.94 | 0.94 | 0.85 |
| | KNN | 0.99 | 0.98 | 0.98 | 0.96 | 1.0 | 0.99 | 0.88 | 0.83 | 0.86 |
| Lymphoma | SVM | 0.96 | 1.0 | 0.94 | 1.0 | 1.0 | 1.0 | 0.9 | 0.78 | 0.94 |
| | RF | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.84 | 0.72 | 0.74 |
| | KNN | 0.98 | 0.98 | 0.92 | 0.98 | 1.0 | 1.0 | 0.84 | 0.74 | 0.68 |
| Ovarian | SVM | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.99 | 0.82 | 1.0 |
| | RF | 0.99 | 0.98 | 0.98 | 0.98 | 1.0 | 0.99 | 0.95 | 0.73 | 0.92 |
| | KNN | 0.97 | 0.97 | 0.97 | 0.97 | 0.99 | 0.99 | 0.96 | 0.66 | 0.92 |
| Pancreatic | SVM | 0.54 | 0.56 | 0.59 | 0.62 | 0.75 | 0.81 | 0.58 | 0.57 | 0.64 |
| | RF | 0.66 | 0.66 | 0.63 | 0.64 | 0.6 | 0.68 | 0.53 | 0.52 | 0.55 |
| | KNN | 0.57 | 0.57 | 0.64 | 0.63 | 0.6 | 0.61 | 0.53 | 0.48 | 0.55 |
| Prostate | SVM | 0.94 | 0.94 | 0.95 | 0.96 | 0.96 | 0.98 | 0.93 | 0.8 | 0.96 |
| | RF | 0.94 | 0.95 | 0.94 | 0.95 | 0.92 | 0.95 | 0.9 | 0.82 | 0.89 |
| | KNN | 0.96 | 0.95 | 0.94 | 0.94 | 0.97 | 0.97 | 0.87 | 0.82 | 0.89 |
| **Average** | **SVM** | **0.88** | **0.89** | **0.90** | **0.91** | **0.94** | **0.96** | **0.86** | **0.77** | **0.90** |
| | **RF** | **0.90** | **0.91** | **0.89** | **0.9** | **0.87** | **0.9** | **0.83** | **0.73** | **0.79** |
| | **KNN** | **0.88** | **0.89** | **0.89** | **0.89** | **0.91** | **0.92** | **0.82** | **0.71** | **0.78** |

For each classifier, we analyzed all combinations with the four feature selection algorithms explained in section 4.4. Classification performance was assessed using a 10-fold cross-validation setting, using accuracy as the performance measure. For each fold, feature selection was performed using only the training part of the data, and a classifier was built using the 1% best features returned by the feature selector, as it was often observed in these domains that only such a small amount of features was relevant [3]. For this experiment, $k = 40$ bootstraps of each training part of the fold were used to create the ensemble versions of the feature selectors. This model was then evaluated on the test part of the data for each fold, and results were averaged over all 10 folds. The results of this experiment are displayed in Table 3.

Averaged over all datasets, we can see that the best classification results are obtained using the SVM classifier, using the ensemble version of RFE as feature selection mechanism. Also for the other classifiers, the combination with the ensemble version of RFE performs well over all datasets. Given that the ensemble version of RFE was also more robust than the single version (Table 2, JC1 rows), this method can thus be used to achieve both robust feature subsets and good classification performance.

In general, it can be observed that the performance of ensemble feature selection techniques is about the same (or slightly better) than the version using a single feature selector, an exception being the Random Forest feature selection technique. Comparing the performance of the Random Forest ensemble feature selection version to the single version, it is clear that the substantial increase in robustness (see Table 2, JC1 rows) comes at a price, and results in lower accuracies for all datasets.

Comparing the results of ensemble feature selection to a classifier using the full feature set (last column in Table 3), it can be observed that in most cases performance is increased, an exception again being the Random Forest feature selector. However, this performance is now obtained at the great advantage of using only 1% of the features. Furthermore, the selected features are robust, greatly improving knowledge discovery and giving more confidence to domain experts, who generally work by iteratively investigating the ranked features in a top-down fashion.

If robustness of the feature selection results is of high importance, then a combined analysis of classification performance and robustness, like the one we presented here, would be advisable. In the case of single and ensemble methods performing equally well, the generally more robust ensemble method can then be chosen to yield both good performance and robustness. In other cases, an appropriate trade-off between robustness and classification performance should be chosen, possibly taking into account the preference of domain experts.

### 4.6 Automatically Balancing Robustness and Classification Performance

In order to provide a formal and automatic way of jointly evaluating the trade-off between robustness and classification performance, we use an adaptation of the F-measure [21]. The F-measure is a well known evaluation performance in data mining, and represents the harmonic mean of precision and recall.

In a similar way, we propose the robustness-performance trade-off (RPT) as being the harmonic mean of the robustness and classification performance.

$$\mathrm{RPT}_\beta = \frac{(\beta^2 + 1)\ \text{robustness performance}}{\beta^2\ \text{robustness} + \ \text{performance}}$$

The parameter $\beta$ controls the relative importance of robustness versus classification performance, and can be used to either put more influence on robustness or on classification performance. A value of $\beta = 1$ is the standard formulation, treating robustness and classification performance equally important.

Table 4 summarizes the results for the different feature selector-classifier combinations when only 1% of the features is used ($\mathrm{RPT}_1$). For the robustness measure, the Jaccard index was used, while for classification performance the accuracy was used. It can be observed that in almost all cases, the ensemble feature selection version results in a better RPT measure. The best RPT values were obtained using the ensemble version of the Random Forest feature

**Table 4.** Harmonic mean of robustness and classification performance ($RPT_1$) for the different feature selector-classifier combinations using 1% of the features

| Dataset | | SU | | RELIEF | | SVM_RFE | | RF | |
|---|---|---|---|---|---|---|---|---|---|
| | | Single | Ensemble | Single | Ensemble | Single | Ensemble | Single | Ensemble |
| Colon | SVM | 0.45 | 0.68 | 0.6 | 0.69 | 0.6 | 0.66 | 0.02 | 0.69 |
| | RF | 0.45 | 0.69 | 0.59 | 0.69 | 0.57 | 0.63 | 0.02 | 0.65 |
| | KNN | 0.44 | 0.67 | 0.59 | 0.68 | 0.58 | 0.65 | 0.02 | 0.67 |
| Leukemia | SVM | 0.7 | 0.74 | 0.61 | 0.71 | 0.66 | 0.72 | 0.52 | 0.84 |
| | RF | 0.7 | 0.75 | 0.61 | 0.71 | 0.65 | 0.72 | 0.52 | 0.86 |
| | KNN | 0.7 | 0.74 | 0.61 | 0.7 | 0.66 | 0.72 | 0.51 | 0.81 |
| Lymphoma | SVM | 0.53 | 0.71 | 0.58 | 0.72 | 0.6 | 0.63 | 0.35 | 0.75 |
| | RF | 0.53 | 0.69 | 0.58 | 0.7 | 0.6 | 0.62 | 0.35 | 0.72 |
| | KNN | 0.54 | 0.7 | 0.58 | 0.71 | 0.6 | 0.63 | 0.35 | 0.73 |
| Ovarian | SVM | 0.91 | 0.92 | 0.92 | 0.94 | 0.89 | 0.91 | 0.18 | 0.82 |
| | RF | 0.91 | 0.91 | 0.91 | 0.93 | 0.89 | 0.91 | 0.18 | 0.78 |
| | KNN | 0.9 | 0.91 | 0.91 | 0.92 | 0.88 | 0.91 | 0.18 | 0.74 |
| Pancreatic | SVM | 0.21 | 0.24 | 0.16 | 0.29 | 0.53 | 0.5 | 0.02 | 0.52 |
| | RF | 0.22 | 0.24 | 0.16 | 0.29 | 0.49 | 0.47 | 0.02 | 0.5 |
| | KNN | 0.21 | 0.24 | 0.16 | 0.29 | 0.49 | 0.45 | 0.02 | 0.48 |
| Prostate | SVM | 0.78 | 0.8 | 0.67 | 0.77 | 0.74 | 0.74 | 0.23 | 0.79 |
| | RF | 0.78 | 0.81 | 0.67 | 0.76 | 0.73 | 0.74 | 0.23 | 0.8 |
| | KNN | 0.79 | 0.81 | 0.67 | 0.76 | 0.74 | 0.74 | 0.23 | 0.8 |
| **Average** | **SVM** | **0.6** | **0.68** | **0.59** | **0.69** | **0.67** | **0.69** | **0.22** | **0.74** |
| | **RF** | **0.6** | **0.68** | **0.59** | **0.68** | **0.65** | **0.68** | **0.22** | **0.72** |
| | **KNN** | **0.6** | **0.68** | **0.59** | **0.68** | **0.66** | **0.68** | **0.22** | **0.71** |

selector, which can be explained by the very high robustness values (see Table 2), compared to the other feature selectors.

## 5 Conclusions and Future Work

In this work we introduced the use of ensemble methods for feature selection. We showed that by constructing ensemble feature selection techniques, robustness of feature ranking and feature subset selection could be improved, using similar techniques as in ensemble methods for supervised learning. When analyzing robustness versus classification performance, ensemble methods show great promise for large feature/small sample size domains. It turns out that the best trade-off between robustness and classification performance depends on the dataset at hand, giving rise to a new model selection strategy, incorporating both classification performance as well as robustness in the evaluation strategy. We believe that robustness of feature selection techniques will gain importance in the future, and the topic of ensemble feature selection techniques might open many new avenues for further research. Important questions to be addressed include the development of stability measures for feature ranking and feature subset selection, methods for generating diversity in feature selection ensembles, aggregation methods to find a consensus ranking or subset from single feature selection models and the design of classifiers that jointly optimize model performance and feature robustness.

## References

1. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
2. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. 97(1-2), 273–324 (1997)
3. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
4. Dunne, K., Cunningham, P., Azuaje, F.: Solutions to instability problems with sequential wrapper-based approaches to feature selection. Technical report TCD-2002-28. Dept. of Computer Science, Trinity College, Dublin, Ireland (2002)
5. Kalousis, A., Prados, J., Hilario, M.: Stability of feature selection algorithms: a study on high-dimensional spaces. Knowl. Inf. Syst. 12(1), 95–116 (2007)
6. Kuncheva, L.: A stability index for feature selection. In: Proceedings of the 25th International Multi-Conference on Artificial Intelligence and Applications, pp. 390–395 (2007)
7. Krízek, P., Kittler, J., Hlavác, V.: Improving Stability of Feature Selection Methods. In: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, pp. 929–936 (2007)
8. Dietterich, T.: Ensemble methods in machine learning. In: Proceedings of the 1st International Workshop on Multiple Classifier Systems, pp. 1–15 (2000)
9. Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging. Statistical Science 14, 382–401 (1999)
10. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C (1988)
11. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Proceedings of the 7th European Conference on Machine Learning, pp. 171–182 (1994)
12. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
13. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning 46(1-3), 389–422 (2002)
14. Breiman, L.: Bagging Predictors: Machine Learning 24(2), 123–140 (1996)
15. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA 96(12), 6745–6750 (1999)
16. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286, 531–537 (1999)
17. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403(3), 503–511 (2000)
18. Petricoin, E.F., Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B.: Use of proteomics patterns in serum to identify ovarian cancer. The Lancet 359(9306), 572–577 (2002)
19. Petricoin, E.F., Ornstein, D.K., Paweletz, C.P., Ardekani, A., Hackett, P.S., Hitt, B.A., Velassco, A., Trucco, C.: Serum proteomic patterns for detection of prostate cancer. J. Natl. Cancer Inst. 94(20), 1576–1578 (2002)
20. Hingorani, S.R., Petricoin, E.F., Maitra, A., Rajapakse, V., King, C., Jacobetz, M.A., Ross, S.: Preinvasive and invasive ductal pancreatic cancer and its early detection in the mouse. Cancer Cell. 4(6), 437–450 (2003)
21. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)