Neurocomputing 73 (2010) 2346–2352

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Finding optimal classifiers for small feature sets in genomics and proteomics

Gregor Stiglic [a,*], Juan J. Rodriguez [b], Peter Kokol [a,c]

[a] University of Maribor, Faculty of Health Sciences, Zitna ulica 15, 2000 Maribor, Slovenia
[b] University of Burgos, c/ Francisco de Vitoria s/n, 09006 Burgos, Spain
[c] University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia

ARTICLE INFO

Available online 31 May 2010

Keywords:
Gene expression analysis
Machine learning
Feature selection
Rotation Forest

ABSTRACT

The classification of genomic and proteomic data in extremely high dimensional datasets is a well-known problem which requires appropriate classification techniques. Classification methods are usually combined with gene selection techniques to provide optimal classification conditions—i.e. a lower dimensional classification environment. Another reason for reducing the dimensionality of such datasets is their interpretability, as it is much easier to interpret a small set of ranked genes than 20 thousand genes. This paper evaluates the classification performance of Rotation Forest classifier on small subsets of ranked genes for two dataset collections consisting of 47 genomic and proteomic classification problems. Robustness and high classification accuracy is shown to be an important feature of Rotation Forest when applied to small sets of genes.

1. Introduction

There are many new classification methods and variants of existing techniques for classification problems. One of them is Random Forest that was first presented by Breiman and Cutler in [1]. It has proven to be a fast, robust and very accurate technique that can be compared with other leading classifiers (e.g. Support Vector Machines [2] and some of the most efficient ensemble-based classification techniques) [3]. Most of these techniques are also used in genomic and proteomic classification problems where classifiers need to be specialized for high dimensional problems. Another option is to integrate feature pre-selection into the classification process, thereby reducing the initial feature set before classification takes place. Most of the early experiments using microarray gene expression datasets used simple statistical methods of gene ranking to reduce the initial set of attributes. Recently, more advanced feature selection methods from the field of machine learning have been applied to the pre-selection step in genomic and proteomic classification problems. Although a small number of genes is preferred, in line with Wang et al. [4], we try to avoid extremely small subsets of genes, where subsets with only two or three genes are used for classification.

This paper evaluates a widely used feature selection technique to determine the most appropriate number of features that should be retained in the pre-selection step to achieve the best classification performance. It moreover introduces one of the most recent classification techniques called Rotation Forest [5] to extensive genomic and proteomic classification using small sets of genes. Rotation Forest classifiers have been previously used in the classification of genomic and proteomic datasets using either a fixed number of selected features or completely different experimental settings [6,7]. The study also evaluates classier performance when applied to different numbers of selected features.

Section 2 of this paper describes the ensemble based classification model known as Rotation Forests. The rest of the paper is organized as follows: in Section 3, we review the classification methods under comparison; in Section 4, we present the results of our experiments by comparing the classification accuracy of Rotation Forests with three other classification methods. Section 5 concludes the paper and provides some pointers for future research into Rotation Forests used for genomic and proteomic classification problems.

2. Rotation Forest

Rotation Forest is a novel classification technique that was initially presented by Rodríguez et al. [5] and was applied to several machine learning problems. In order to obtain successful ensembles, the member classifiers should be accurate and diverse. The sampling process in bagging and Random Forest is introduced in order to obtain diverse classifiers, although using a subset of the examples to train the classifiers can degrade the accuracy of

* Corresponding author.
E-mail addresses: gregor.stiglic@uni-mb.si (G. Stiglic), jjrodriguez@ubu.es (J.J. Rodriguez), kokol@uni-mb.si (P. Kokol).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.02.024

the member classifiers, especially if the number of samples is small. Hence, a natural question is whether diverse classifiers may be obtained without discarding any information in the dataset.

Most ensemble methods can be used with any classification method, but decision trees are one of the most commonly used. There are ensemble methods designed specifically for decision trees, such as Random and Rotation Forests. The latter is based on the sensitivity of decision trees to axis rotations; the classifiers obtained with different rotations of a dataset can vary greatly. This sensitivity is usually considered a disadvantage, but it can be very beneficial when the trees are used as members of an ensemble. The trees obtained from a rotated dataset can still be accurate, because they use all the information available in the dataset, but at the same time they can be very diverse.

As in bagging and Random Forests, each member of the ensemble is trained with a different dataset. These datasets are obtained from a random transformation of the original training data. In Rotation Forests, the transformation of the dataset consists of the following steps:

- Features are randomly grouped in k groups, where k is a parameter of the method.
- For each group of features:
- A dataset consisting of all the examples and the features in the group is created.
- A subset of the classes is randomly selected. All the examples of these classes are removed from the current dataset.
- A subset of randomly chosen examples is eliminated from the new dataset (by default 25% of samples are removed).
- Principal component analysis (PCA) is applied to the remaining samples in the dataset.
- PCA components are considered as a new set of features. None of the components are discarded.
- All training samples are transformed using new variables selected by PCA for each group.
- A classifier is built from the transformed training set.
- A further classifier is built by repeating the first step, in case the final number of classifiers in the ensemble is not reached.

A more formal description of the method is available in [5]. Moreover, the source code of the method is available as part of Weka [8].

This transformation produces a rotation of the axis. The transformed dataset has as many examples as the original dataset, all the information that was in the original dataset remains in the transformed dataset, because none of the components are discarded, and all the training examples are used to train all the ensemble methods.

The number of features in each group (or the number of groups) is a parameter of the method. The optimal value for this parameter depends on the dataset and could be selected with an internal cross validation. Nevertheless, in this work the default value was used, and groups were formed using 3 features. The selection of the optimal value for this parameter would notably increase the time required for training the classifiers and would put Rotation Forests at an advantage with respect to other ensemble methods that do not optimize the value of any parameters.

The elimination of classes and examples from the dataset is performed because PCA is a deterministic method, and it would not be surprising (especially for big ensembles) if some members of the ensemble had the same (or very similar) variable groupings. Hence, an additional source of diversity was needed. This elimination is only done for the dataset on which PCA was performed; all the examples were used for training the classifiers in the ensemble.

3. Feature selection and classification techniques

The main idea of feature selection in genomic and proteomic datasets is to select a subset of variables that can significantly improve the time complexity and accuracy of a classification model. In such datasets, an initial set of features consists of thousands of gene expression values. With such a large amount of features, it is especially interesting to search for dependency between the optimal number of selected features and the accuracy of the classification model. There are two large groups of feature selection techniques in bioinformatics—classical methods based on statistical theory and more advanced machine-learning-based methods. Different metrics such as distance metrics, information measures, correlation and consistency metrics can be used to arrive at the features that will be used to efficiently separate the samples by their class value. This study primarily focuses on classification methods, and a simple, widely used method based on the t-test statistic [9] was used for feature selection in all experiments.

In addition to the t-test based feature selection method, the experiments presented in this paper all used a set of four classification techniques:

- Random Forests
- Rotation Forests
- Support Vector Machines (SVM)
- k-Nearest Neighbours (k-NN)

A machine learning software framework known as Weka, which implements all the above-mentioned methods except the t-test filter developed by the authors, was used for all experiments. All the above-mentioned methods except for Rotation Forest, as explained earlier, are briefly described in the remainder of this section.

3.1. Random Forests

Breiman upgraded the idea of bagging by combining it with the random feature selection for decision trees. Thus, he created Random Forests, where each member of the ensemble is trained on a bootstrap replicate as in bagging. Decision trees are then grown by selecting the feature to split at each node from a randomly selected number of features. The number of chosen features is set at $\log_2(k+1)$ as in [1], where k is the total number of features.

Random Forests is an ensemble building method that works well even with noisy content in the training dataset and is considered to be one of the most competitive methods that is comparable to boosting [10].

3.2. Support Vector Machines (SVM)

SVM are increasingly popular classifiers in many areas, including bioinformatics [2]. The most basic variant of SVM uses linear kernel and seeks to find an optimal hyperplane that separates samples of different classes. When classes can be linearly separated, the hyperplane is located so that there is a maximal distance between the hyperplane and the nearest sample of any class. In cases when samples cannot be linearly separated, there is no optimal separating hyperplane; in such cases, we try to maximize the margin while allowing for some classification errors. A Weka implementation of SMO using the sequential minimal optimization (SMO) training mechanism proposed by Platt [11,12] was used in all the experiments in this study. It offers very quick and reliable learning of the SVM-based decision models.

3.3. k-Nearest Neighbours (k-NN)

Nearest neighbour classifier is a typical case-based classifier, in which all samples are stored for later use in the classification process [13]. It aims to classify samples according to similarities or distances between them. A class value is defined using class values of k nearest samples. Similarity to neighbouring samples is calculated using distances between samples that are usually measured by Euclidean distance.

Another important parameter that has to be set is number of neighbours to be used for calculation of class value. We use Weka's option in our experiments to select the best k from 1 to 5 using hold-one-out cross-validation. Voting for the final class is weighted according to the distance from the current sample to the k nearest neighbours.

k-NN based classifiers are most useful in cases with continuous attribute values that also include genomic and proteomic datasets. The k-NN classification process is suitable for a lower number of samples (e.g. in gene expression datasets), as computational costs rise significantly with higher numbers of samples.

4. Experimental settings and results

All experiments were completed using a 10-fold cross validation procedure due to the small number of samples in some datasets. To avoid feature selection bias, as discussed in Ambroise and McLachlan [14], a separate feature selection process was completed for each training set during the 10-fold cross validation. The feature selection step was done using the t-test statistic to select features with the lowest p-values. Different numbers of features were selected to observe the classification performance in low and high dimensional spaces. The number of selected features was defined as 2^x where $x=[2\ldots9]$.

The first collection of 11 proteomic and genomic datasets used in this study is available from the Kent Ridge Biomedical Data Set Repository (KRBDSR) [15] where additional information including references to original work for each of the datasets may be found. Table 1 represents brief description of datasets from the Kent Ridge repository. A second collection of 36 microarray datasets was taken from the Gene Expression Machine Learning Repository (GEMLeR) [16] and contains pair-wise comparisons of 9 different cancer tissue types. The number of samples in the first collection ranges from 47 to 253 samples in the largest dataset, which makes an average of 117 samples per dataset. The second collection comprises datasets that contain larger numbers of samples with an average of 343.3 samples per dataset, including four datasets with over 500 samples. Both collections were analysed separately in most of the experiments for the purposes of comparing the performance of the classification methods on both smaller and larger datasets. Another important fact that separates the two collections of datasets is the biological meaning of the problems. In the GEMLeR datasets, we seek to separate two types of cancer tissues, whereas KRBDSR contains problems that are usually regarded as more complex, where we compare normal samples with disease state samples.

Table 1
Details for genomic and proteomic datasets from Kent Ridge repository.

Dataset	Original work	Genes	Patients	Classes
ALLAML	Golub et al.	7129	72	2
Breast	Van't Veer et al.	24481	97	2
CNS	Mukherjee et al.	7129	60	2
Colon	Alon et al.	2000	62	2
DLBCL	Alizadeh et al.	4026	47	2
DLBCL-NIH	Rosenwald et al.	7399	240	2
DLBCL-Tumor	Shipp et al.	6817	77	2
Lung	Gordon et al.	12533	181	2
Lung-Michigan	Beer et al.	7129	96	2
Ovarian	Petricoin et al.	15154	253	2
Prostate	Singh et al.	12600	102	2

Classification accuracy was measured in terms of percentages of correctly classified samples in all experiments. However, averaging the results over all datasets does not appear to be an appropriate technique to compare the performance of classification, and an alternative and fairer method of comparison was used in its place. With the help of SPSS statistics software suite version 17.0 and Friedman's non-parametric test, we calculated the average rankings from comparisons between each of the four classification methods. A classification method was assigned with an average ranking of between 1 (worst classification on all datasets) and 4 (best classification on all datasets).

4.1. Assessment of classification performance

Fig. 1 shows the average rank scores for the Kent Ridge collection of datasets as a function of the number of selected features. There are two winners in this comparison—Rotation Forest classifier achieved the highest rank in 5 cases and SVM won in 3 cases based on average rank. However, one should be aware that average accuracy rates at 128 or 256 features differ significantly from those at very low numbers of selected features (see Fig. 4). The best accuracy rates for the two methods under comparison were achieved at 256 selected features. It can therefore be said that SVM would only be the optimal solution in higher dimensional spaces.

Our second experiment compared four classification methods which were applied to the 36 GEMLeR datasets. Fig. 2 shows the average ranking calculated from accuracy rates on 36 datasets. It may be observed that Rotation Forest outperforms all the other methods and that Random Forest also outperforms SVM in the majority of cases. One explanation could be the sample size. Decision tree algorithms, on which both Rotation Forest and Random Forests are based, are known to perform better as the number of samples increases. The highest classification accuracy results were once again achieved at 256 selected features.

More effective visualization of the performance of the four classification methods for different numbers of selected features was obtained by using correspondence analysis, which is a multivariate statistical technique that is conceptually similar to PCA, but which scales the data (that has to be positive) so that rows and columns are treated equally [17]. The dimensions are computed so as to maximize the distances between the row or column points. The extraction of the dimensions (usually two or

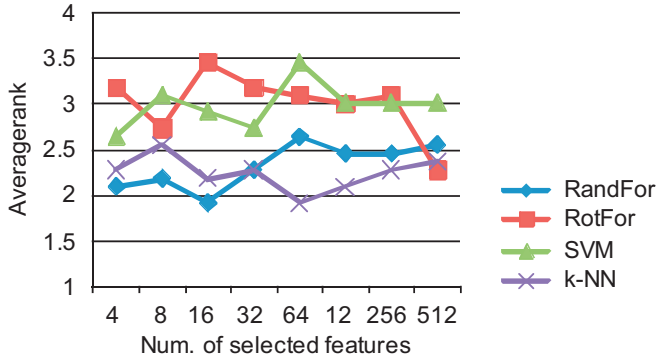

Fig. 1. Average ranking for all four classification methods using different numbers of pre-selected features on the Kent Ridge repository datasets.

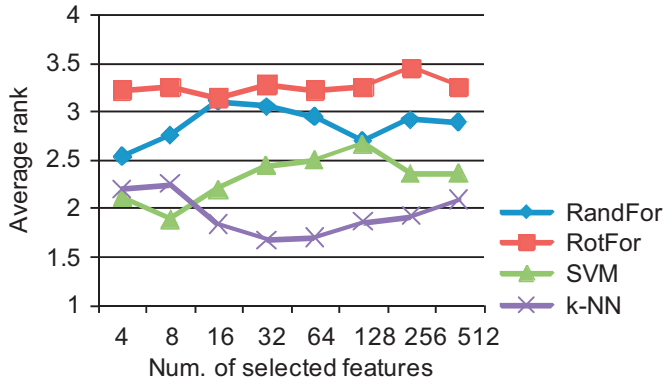

Fig. 2. Average ranking for all four classification methods applied to the GEMLeR datasets using different numbers of pre-selected features.

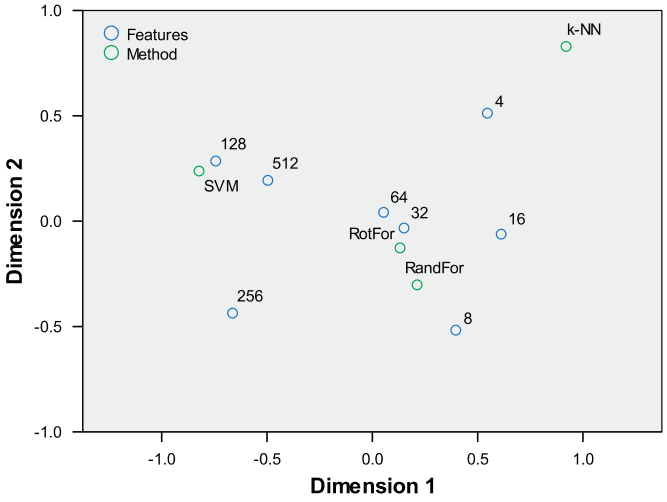

Fig. 3. Correspondence analysis using symmetrical normalization and the chi-square based distance metric for all four classification methods (Random Forests, Rotation Forest, SVM and k-NN).

three) is similar to principal component extraction in factor analysis. Simple counting was used for each method at different numbers of selected features—i.e. a method was awarded 1 point if it outperformed the other three methods in terms of accuracy for the given dataset. Fig. 3 demonstrates the results of correspondence analysis in which the classification method groups associated with a certain number of selected features may be observed. Correspondence analysis method from the SPSS statistical software package using symmetrical normalization and the chi-square based distance metric were applied to produce Fig. 3.

The nearest neighbours classifier can be seen in the upper right corner close to 4—i.e. a very small number of selected features. Observing Figs. 1 and 2, it can be confirmed that k-NN performs best when the number of selected features is small. In contrast, SVM, which appears on the extreme left, very near the points representing high dimensional feature selection, is therefore the most suitable for cases where more features are selected. It may also be seen that Rotation Forests and Random Forests lie very close together due to their similar characteristics. At a more abstract level, looking at dimensions 1 and 2, it may be noted that Random Forest and Rotation Forest lie very close together while SVM as the most complex method lies far away from k-NN which could be understood to be the simplest method—especially as dimension 1 describes this "complexity" of the classifiers very well.

Fig. 4 displays average classification accuracy rates to highlight the different characteristics of the datasets used in this study. In addition to simple accuracy, area under ROC curve, or simply area under curve (AUC) was measured in this experiment as a further performance metric. Results of the average AUC rates are presented in Fig. 5. It should be stated once again that Figs. 4 and 5 are not intended to compare the performance of single methods as ranking based methods serve this purpose much better. It is notable that in the case of the GEMLeR datasets, a smaller number of genes is needed to stabilize the classification accuracy rates. One of the explanations might be that, in general, it is much easier to discern two tissue types than two types of disease or two types of disease prognosis.

Furthermore, even greater differences may be observed between SVM and Rotation Forest when comparing average AUC rates (Fig. 5). The following experiments show whether this weakness can be avoided by using either parameter tuning or different SVM kernels.

The second experiment demonstrates the superior performance of the Rotation Forest classifier over all feature selection settings. However, time complexity is a significant drawback that detracts from this method, especially in real-time systems. Fig. 6 shows time of building for each of the four classifiers averaged over all of the 47 datasets used in the first two experiments. The lengthy time complexity of the Rotation Forest classifier is clearly evident, but one can expect very good results when it is used. For extremely large numbers of pre-selected genes, Rotation Forest will become very slow, which is mainly due to the PCA transformations that have to be performed for each tree in each ensemble.

4.2. Optimizing the classification performance

Diaz-Uriarte and Alvarez de Andres state in [2] that under some circumstances Random Forest classifier can outperform the current state-of-the-art SVM classifier. This claim was later rejected by Statnikov et al. [18] in a rigorous evaluation that put SVM back into so called current "best of class" classifier. Based on a paper by Rodríguez et al. [5], which demonstrated that Rotation Forest outperforms Random Forests on most datasets, a further experiment comparing Rotation Forest to SVM on all 47 genomic and proteomic datasets was designed. To allow for parameter tuning as described in [18], we employed Weka's GridSearch component that allows two parameters to be tuned. Due to time complexity, we limited the parameter tuning to only one parameter for Rotation Forest—i.e. the classifier was cross-validated at a different number of selected features (n) while the number of iterations stayed fixed at 100. In the case of SVM, GridSearch was used to set the optimal number of selected features and the penalty parameter $C\{0.0001, 0.01, 1, 100\}$. Two steps of extension were allowed, which means that parameter $C=100^2$ may also be used, if nested cross-validation shows the best accuracy rate at $C=100$, which can be further extended to $C=100^3$. Using two steps of extension, SVM GridSearch covers an 8×8 grid where n ranges from 4 to 512 and C from 100^{-4} to 100^3.

Fig. 7 shows the difference in classification accuracy and AUC for SVM and Rotation Forest on all 47 datasets. Rotation Forest performed better with regard to accuracy (24 wins, 6 draws, 17 losses), while SVM performed better when AUC was observed (28 wins for SVM v. 19 for Rotation Forest). Wilcoxon's signed ranks test was used to check for statistically significant differences between the two methods under comparison. No significant differences were detected ($p=0.088$) when accuracy was compared. When comparing AUC ranking for 47 datasets, there

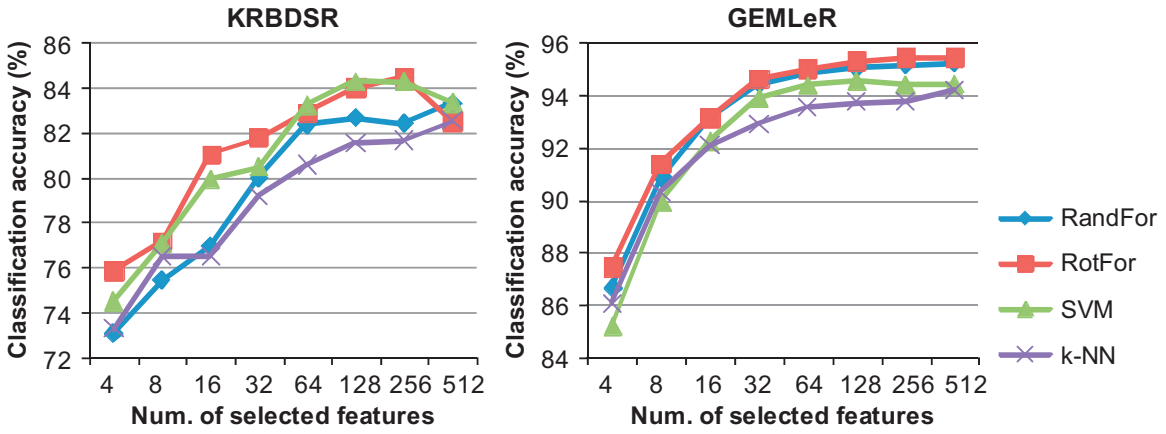

Fig. 4. Average accuracy rates as a function of the different number of selected features using 11 Kent Ridge Biomedical Data Set Repository datasets (left) and 36 GEMLeR datasets (right).

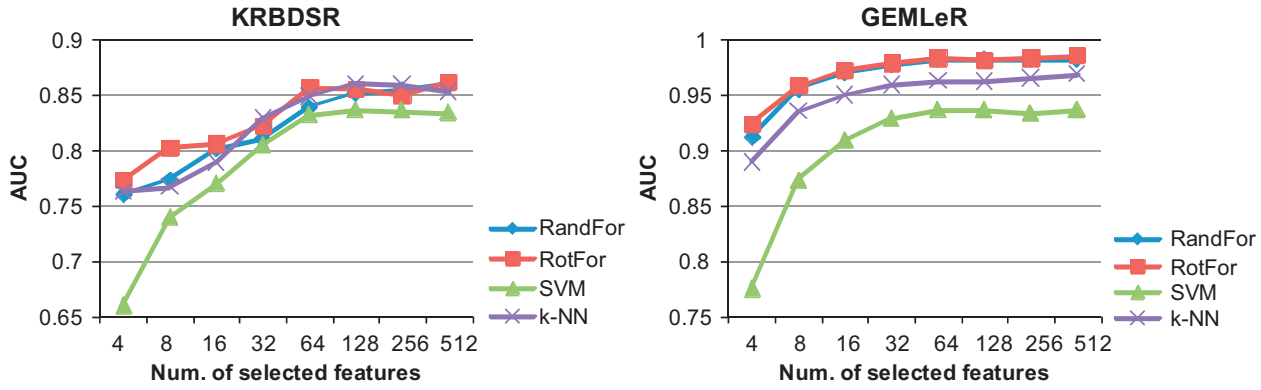

Fig. 5. Average AUC rates as a function of the different number of selected features using 11 Kent Ridge Biomedical Data Set Repository datasets (left) and 36 GEMLeR datasets (right).

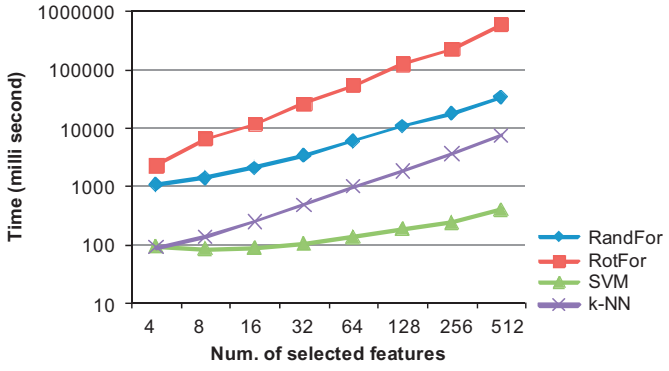

Fig. 6. Average time in milliseconds needed to build a classifier.

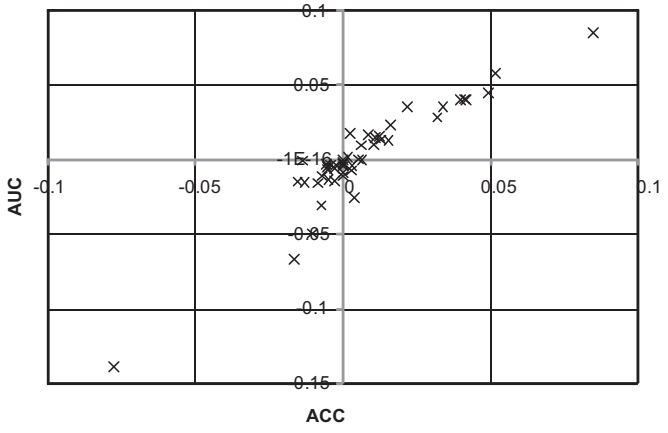

Fig. 7. Differences in accuracy (ACC) and AUC when comparing tuned Rotation Forest to SVM on 47 datasets. Positive differences represent datasets where Rotation Forest outperformed SVM.

was even less evidence of statistically significant differences ($p=0.743$).

Important information on the performance of both methods can be observed in data collected from the internal cross-validation based tuning, in which the best parameters of 10 cross-validations for each dataset were stored. Fig. 8 displays the surface plot of the 8×8 matrix for two tuned parameters of the SVM classifier. One can observe that the lower values of either parameter C or n were left practically unused in the process of tuning the SVM. The optimal solutions were mostly found in the upper 4×4 matrix of the parameter values. For parameter C the most frequently used setting was $C=1$ (244 out of 470 tunings), followed by $C=10^6$ (121 cases). Those two settings alone were used in more than 3/4 of all evaluations (47 datasets, each with 10 cross-validations), which provides potentially important information on the most appropriate parameters when no tuning is allowed.

On account of time complexity, only the feature selection parameter was tuned for both methods. Fig. 9 displays a

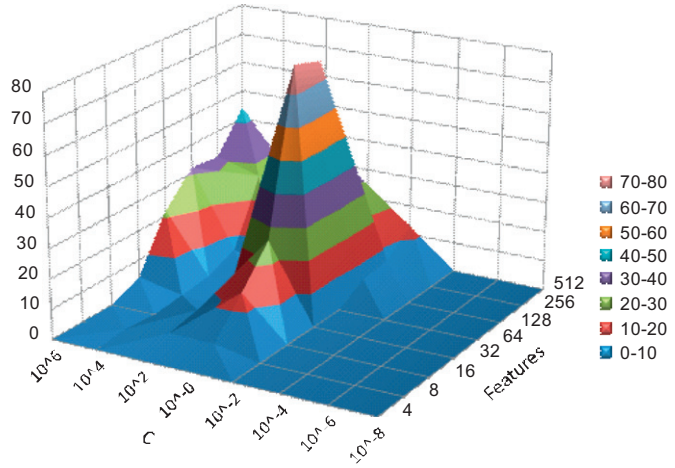

Fig. 8. Surface plot for SVM parameter tuning from 10-fold cross-validations on 47 datasets.

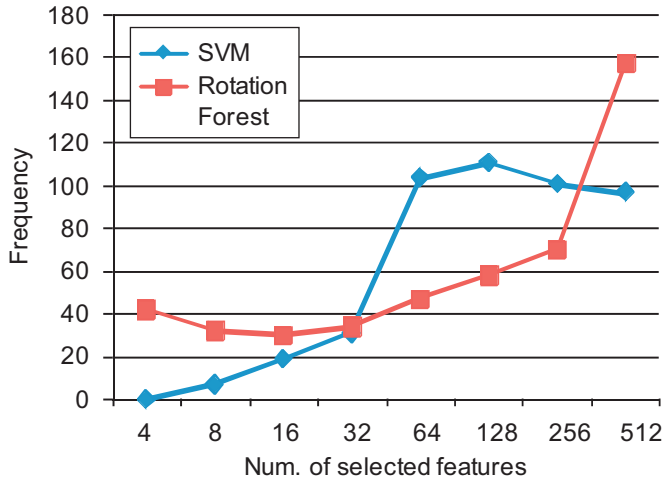

Fig. 9. Frequency of selection during parameter tuning of the number of selected features for Rotation Forest and SVM.

comparison of feature selection settings frequency during parameter tuning for Rotation Forest and SVM. The results are unexpected to some extent, especially in the case of Rotation Forest where in most cases 512 features were selected, whereas the previous experiments (Fig. 4) demonstrated that Rotation Forest failed to return the best accuracy rates at 512 selected features.

In the case of SVM, it may be seen that the 4 settings with the highest number of features (64, 128, 256 and 512) were selected in the majority of tunings. It may therefore be assumed that it is very difficult to choose the optimal number of features to select when SVM is used. However, based on our results, it is evident that this number should be higher than 50 selected features.

4.3. Non-linear kernels

Since linear kernels used in an initial parameter tuning experiment might not be the most suitable for comparison to non-linear Rotation Forest, another experiment using non-linear SVM kernels was conducted. When introducing polynomial and RBF kernels, one should be aware of additional parameters that

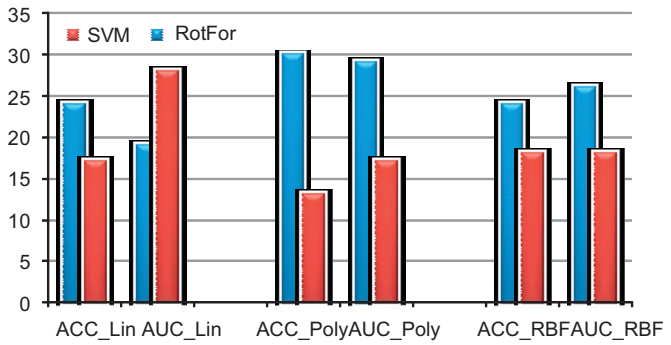

Fig. 10. Comparison of Rotation Forest and tuned SVM classifiers using linear, polynomial and RBF kernels (ties are not displayed).

can be tuned such as polynomial kernel exponents and gamma parameters of the RBF kernel. Following the initial parameter tuning, we set the number of selected features to 128 (see Fig. 9) and tuned another two parameters. Based on previous results (see Fig. 8) the initial tuning settings for parameter C that determines the complexity of the built classifier were set to {10, 10^2, 10^3, 10^4}. The degree of exponent was set at intervals between 3 and 6 in steps of 1, with the RBF gamma interval set to {1, 10^{-1}, 10^{-2}, 10^{-3}}. Two grid extensions in either positive or negative direction were allowed for all parameters.

As can be observed from Fig. 10, neither of the non-linear kernel based methods could match the classification performance of Rotation Forest. Observing the results, one could argue that the polynomial kernel should have selected an exponent setting of 1 more often if linear kernel produces the best results. This might be true, but since we use grid search optimization which allows two degrees of extension, it is very unlikely that 1 will be chosen for this parameter (performance should improve in cases where the exponent is 2 and then additionally where it is 1 in internal cross-validation based tuning).

Fig. 11 displays surface plots that demonstrate the proportion of different values for all tuned parameters. Both plots hint at the possibility of improving the classification using higher values for the complexity parameter C. However, increasing parameter C can significantly increase the chances of over fitting the classifier [19,20].

5. Discussion and conclusions

This paper has presented an extensive evaluation and comparison of four machine learning classifiers using different pre-selected sets of features (genes) on 47 different genomic and proteomic datasets. It has demonstrated that Rotation Forest classifier offers advantages over Random Forest classifier and moreover, very competitive results when compared to SVM. The results of our first experiment suggest that Rotation Forest should be considered in cases where small groups of genes are needed. The robustness of Rotation Forest classifier is also demonstrated in comparisons of the AUC with other methods for different numbers of selected features. Those results confirm the ones obtained in our previous work, where more complex feature selection methods were used. Fig. 12 presents results from [21] where ReliefF and SVM-RFE were used for feature selection.

In the second experiment, the performance of SVM could not match either Rotation Forest or Random Forest in most cases. On the other hand it has to be mentioned that all GEMLeR datasets represent very similar problems—i.e. tissue to tissue comparison

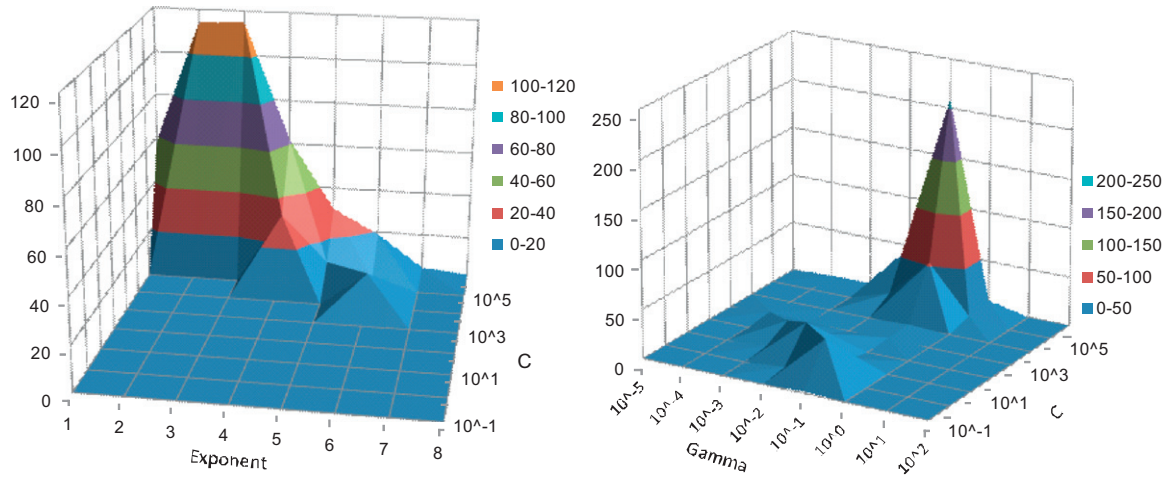

Fig. 11. Surface plots for SVM tuning using polynomial (left) and RBF (right) kernels.

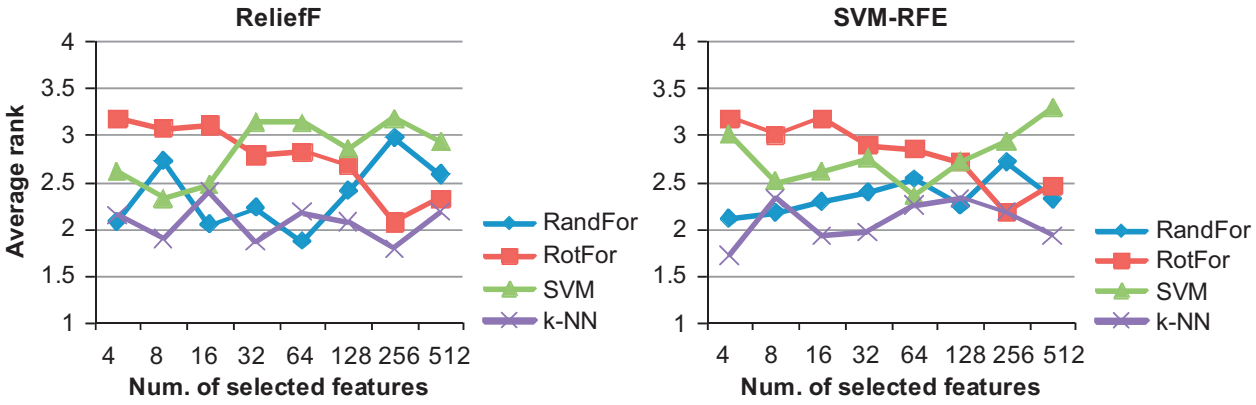

Fig. 12. Comparison of 4 classifiers using ReliefF and SVM-RFE for selection of attributes.

which is regarded as a less complex problem in comparison to some other prognostic and diagnostic microarray problems. By tuning parameters using internal cross-validation on a training set, one can even further enhance the performance of both Rotation Forest and the SVM classifier. Our final experiment included 47 genomic and proteomic datasets. It demonstrated that there is no significant difference between SVM and Rotation Forest, but Rotation Forest is the only classifier that is not significantly outperformed by SVM. The results show that Rotation Forest performs better on a majority of datasets when accuracy is considered, whereas SVM should be preferred for better AUC performance when linear kernels are used for SVM. Additionally, Rotation Forest performs better in cases when short lists of significant genes are required.

References

[1] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.
[2] V. Vapnik, in: Statistical Learning Theory, John Wiley, New York, 1998.
[3] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd international Conference on Machine Learning (ICML '06), vol. 148, 2006, pp. 161–168.
[4] L. Wang, F. Chu, W. Xie, Accurate cancer classification using expressions of very few genes, IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (1) (2007) 40–53.
[5] J.J. Rodríguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1619–1630.
[6] G. Stiglic, P. Kokol, Effectiveness of rotation forest in meta-learning based gene expression classification, IEEE Symposium on Computer-Based Medical Systems (2007) 243–250.
[7] K.H. Liu, D.S. Huang, Cancer classification using rotation forest, Computers in Biology and Medicine 38 (5) (2008) 601–610.
[8] I.H. Witten, E. Frank, in: Data Mining: Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann, San Francisco, 2005.
[9] P. Baldi, A.D. Long, A Bayesian framework for the analysis of microarray expression data: regularized *t*-test and statistical inferences of gene changes, Bioinformatics 17 (6) (2001) 509–519.
[10] T.G. Dietterich, Ensemble learning, in: M.A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, 2nd ed., The MIT Press, Cambridge, MA, 2002, pp. 405–408.
[11] J. Platt, Machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, 1998.
[12] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, Neural Computation 13 (3) (2001) 637–649.
[13] T. Mitchell, in: Machine Learning, McGraw Hill, 1997.
[14] C. Ambroise, G.J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proceedings of the National Academy of Sciences USA 99 (2002) 6562–6566.
[15] Kent Ridge Biomedical Data Set Repository, 〈http://datam.i2r.a-star.edu.sg/datasets/krbd/〉 (last accessed 19/01/09).
[16] Gene Expression Machine Learning Repository (GEMLeR), 〈http://gemler.fzv.uni-mb.si/〉 (last accessed 19/01/09).
[17] M. Greenacre, in: Theory and Applications of Correspondence Analysis, Academic Press, 1983.