Pre-filtering Features in Random Forests for Microarray Data Classification

Nicoletta Dessì, Gabriele Milia, Barbara Pes
Università degli Studi di Cagliari, Dipartimento di Matematica e Informatica,
Via Ospedale 72, 09124 Cagliari, Italy
{dessi, milia.ga, pes}@unica.it

Abstract. Random forests have been applied, with promising results, in analyzing datasets with large dimensionality and are receiving increasing attention for classification of microarray datasets. This paper examines random forests from an experimental perspective. It first aims at confirming their effectiveness in microarray data classification, but its main contribution is two-fold: to evaluate the effects of a filtering process which precedes the actual construction of the random forest and, in addition, to provide some insights about the behavior of random forest critical parameters, i.e. the forest size and the number of variable chosen at each split in growing trees. We experimented tuning these critical parameters in a public microarray dataset within a filter method. The paper gives suggestions on the optimal choice of these parameters and presents results which compare well with state-of-art methods for micro-array classification.

Keywords: Microarray classification, Random Forests, Gene selection.

1 Introduction

Microarray technology allows the collection of large-scale gene expression data and is currently used in medical diagnosis for identifying genes that play an important role in the pathogenesis of complex diseases. Such identification requires facing the challenge of handling datasets where the number of genes, namely features, is much larger than the number of samples. Even though thousands of genes are usually investigated, only a very small number of them show a correlation with the disease in question, which makes training and testing of general classification methods difficult.

Decision trees are among the popular machine learning methods. Being produced by a greedy algorithm, a single tree may generate an unstable classification model with poor generalization accuracy as a small change to the data can result in a very different model. The proposal of random forests [1], a method for classification based on the repeated growing of trees through the introduction of a random perturbation, tries to counteract such instability averaging the outcome of a great number of models fitted to the same dataset. At each node of the trees, a small subset of randomly selected features, instead of all features, are considered to split the node. As a sub-product of this technique, the identification of variables that are important in a great number of models provides suggestions in terms of variable selection.

Good generalization performance is critical for many learning algorithms, in general, and for microarray data classification in particular as it remarks on the performance of the algorithm on novel data. As demonstrated in [1], the generalization error is influenced by two factors, the correlation between the random trees and their individual strengths. Breiman [1] further derives that as the number of random trees becomes large (tends to infinity), the generalization error converges to a limit.

Random forests have been applied, with promising results, in analyzing datasets with large dimensionality. Extending these studies to develop random forests for microarray data analysis presents an interesting research goal as it has been observed [2] that random forest performance tends to decline when the number of features is huge and the proportion of truly informative features is small, such as with microarray data. Thus, the effectiveness of a random forest classification process is largely dependent on its capability in facing the curse of dimensionality of gene expression data.

Pre-filtering features is a popular procedure that considerably reduces the dataset dimensionality in building a classifier. The interest is to remove irrelevant variables and to score individual features according to their discriminative power, i.e. their capacity of separating the classes. Filtering results in a ranked list where features appear in descending order of relevance. Within this approach, a first critical issue is the choice of a threshold value denoting the cut-off point of the list of ranked features. When applied before growing a random forest, this process has to face an additional issue: asserting values for the two critical parameters of the random forest, i.e. the number of variables randomly chosen at each split and the number of the trees in the forest.

The aim of this paper is two-fold: to evaluate the effects of a filtering process which precedes the actual construction of the random forest and, in addition, to provide some insights about the behavior of random forest critical parameters.

To demonstrate the effectiveness of the random forest method, we first carried out a classification process that grows random forests on a public microarray dataset without pre-filtering features. We also experimented tuning the random forest critical parameters on the whole dataset. As a next step, to evaluate the impact of a filter approach, we conducted experiments on tuning critical parameters within different choices of the cut-off point. Specifically, for each cut-off, the related subset of top-ranked features was considered to grow random forests. Experiments investigated a trade-off between the number of trees and the number of variables randomly chosen at each split.

Our studies indicate that growing few trees, with one variable randomly chosen at each split, on small subsets of pre-filtered features presents results which compare well with state-of-art methods that use random forest as basic procedure for micro-array data classification.

The paper is organized as follows. Section 2 gives a short background about random forests and related work in microarray data analysis. Section 3 presents the experiments whose results are discussed in Section 4. Finally, Section 5 presents conclusions.

2 Background and Related Work

Given a training set with N cases and M features, a random tree is constructed as follows:

1. N cases are sampled at random (with replacement) from the original dataset; they represent the new training set to construct the tree.
2. A number *mtry* much smaller than M is specified and it is held constant during the forest growing. Each node is split using the best split among a randomly selected subset of *mtry* features.
3. Each tree is grown fully (without pruning).

The random forest, in most cases, is more difficult for humans to interpret than a decision tree [3] but this algorithm presents several interesting advantages, making it suitable for analysis of microarray data. According to [1] [4] [5] [6], (i) it can be used for both binary and multi-category classification, (ii) it can handle thousands of input features (without feature deletion) even when there are a few cases, (iii) it runs efficiently on high-dimensional datasets, (iv) it is relatively insensitive to irrelevant features, (v) it provides an embedded measure of variable importance, (vi) it is robust against overfitting.

In more detail, Svetnik et al. [4] found that for datasets with a large number of features, a random forest can be trained in less time than a single decision tree because the method tests effectively only *mtry* features (rather than M) and it does not do any pruning at all.

As previous mentioned, the critical parameters of a random forest are the number of trees, namely *ntree*, and number of random features to split each node of a tree, namely *mtry*. The value of *mtry* can range from 1 to M and common default values are $\sqrt{M}$ or log(M) [7].

Breiman [1] asserts that the default parameterization of the random forest often leads to excellent performance, but recent studies suggest a fine-tuning of the parameters. Specifically, papers [5], [6] employed a large numbers of trees (500, 1000, 2000), but Zhang and Wang [8] demonstrated that is not necessary to use the full forest for satisfying prediction performance. In their study the size of the optimal sub-forest is in the range of tens and some sub-forests can even overcome the original forest in terms of prediction accuracy on a breast cancer prognosis dataset. Our study aims to validate this thesis using both a different dataset (a diagnostic dataset) and a different approach.

Within high-dimensional classification problems, *mtry* is analyzed in [9] which suggests that this parameter must be sufficiently large in order to have a high probability to capture important variables, i.e. variables highly related to the class. Liaw and Wiener [10] underlined that, if the number of genes is large and the percentage of truly information is small, choosing large values of *mtry* may give better performance. However *mtry* = 1 can give very good performance for some datasets, even though, in this case, the trees are essentially making random splits. Armaratunga et al. [2] proposed a filtering approach to reduce the contribution of trees whose nodes are populated by non-informative features. Specifically, they

choose the splitting subset at each node by weighted random sampling instead of simple random sampling.

3 Experiments

The microarray dataset used in this paper is the public colon tumor dataset of Alon et al. [11] which contains 2000 genes and 62 samples coming from colon cancer patients. The samples belong to 40 tumor biopsies collected from tumor parts (labelled as 'negative') and 22 normal biopsies collected from healthy tissues of the same patients (labelled as 'positive'). This dataset is recognized as one of the noisiest microarray benchmarks.

The dataset was analyzed using the Weka data mining environment [12]. The overall analysis was implemented using leave-one-out cross-validation (LOOCV) in order to obtain a more thorough classification result even though expensive from a computational point of view.

The experiments were divided into two classes:

1. *Tuning on the whole dataset.* Here we used the whole dataset for growing different random forests within the following initial parameters: (i) number of trees = {10**,** 20, 30, 50, 100, 200, 300, 500, 1000}, (ii) *mtry* = {1, 2, 3, 5, 10}. The choice (ii) is motivated by the aim of exploring *mtry* values smaller than the common default values.

2. *Tuning on filtered subsets.* First, we ranked the features of the original dataset. We chose as ranking method Information Gain [13], which is a widely adopted ranking method in analyzing microarray data. It outputs a ranking of the features, based on which we selected different subsets of high-ranked features. For the purpose of simplicity, henceforth we denote these subsets as TOP10 (i.e. the first 10 top-ranked features), TOP30 (i.e. the first 30 top-ranked features) and so on. Then, we used these subsets for growing random forests using different parameter configurations. The values considered are as follows: (i) number of trees = **{**10, 20, 30, 50, 100, 200, 300), (ii) *mtry* = {1, 2, 3, 5, 10}.

The others parameters of the algorithm not mentioned above were used with their default value [12]. The performance of the method was evaluated using the value of area under the curve (AUC) of the receiver operating characteristic (ROC) curve in order to synthesize the information of sensitivity and specificity. Furthermore, AUC metric is not sensitive to unbalanced distributions and is more discriminative than the accuracy metric [14].

4 Results and Discussion

This section reports results for different settings of *ntree* and *mtry* and shows both the random forest stability to changes in its parameters and the effectiveness of the pre-filtering. For the sake of synthesis, the most significant results are summarized for each class of experiments.

Tuning on the whole dataset. First, we analyzed the behavior of random forest method on the whole dataset. For different values of *mtry*, Fig.1 shows the effects of changes in the parameter *ntree* on the AUC. As asserted by [1], the behavior of AUC is asymptotic: as the number of random trees becomes large (*ntree* >100), the AUC value converges to a limit. Smaller *ntree* values lead to more unstable values of AUC and *ntree* = 50 or *ntree* = 100 seem quite adequate, with further increases having negligible effects.

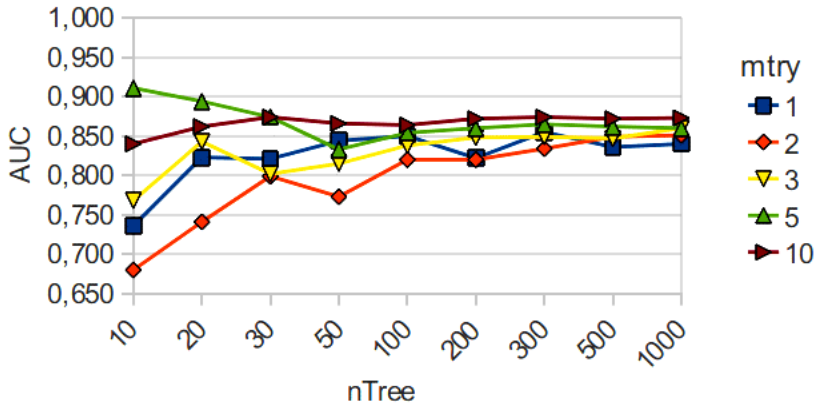

Fig. 1. AUC versus *ntree* for *mtry* = {1, 2, 3, 5, 10}

As Fig.1 shows, for small values of *ntree* (i.e. *ntree* = {10-50}), the choice of high values of *mtry* (*mtry* >= 5) results in higher values of AUC while the choice = 1 is better suited for *ntree* > 50. This seems to suggest that, when we choose to grow a forest with a small number of trees, we need to set higher values for *mtry* in order to increase the probability of randomly selecting informative variables.

In conclusion, experiments seem to suggest that the values *mtry* = 1 and *ntree* = 100 are a good option on the whole dataset which consists of 2000 features. Results confirm what asserted in [10]: *mtry* = 1 can give very good performance for some datasets, even though, in this case, the trees are essentially making random splits.

Tuning on filtered subsets. Here, we compare previous results with results obtained by growing random forests on some subsets of pre-filtered features. Based on previous findings, we set *mtry* = 1. Fig. 2 shows the classification performance on the whole dataset against the performance on the subsets TOP10, TOP100, TOP500. Again, we notice the asymptotic behavior of AUC. Highest AUC values are obtained with a few trees in TOP10 and TOP100. The effectiveness of the pre-filtering process

is considerable as the random forests grown on the filtered subsets greatly outperform the random forests built on the original dataset. As the whole dataset has a very large number of variables but it is expected that only very few are important, filtering increases the probability to capture informative variables. As a consequence, growing a few trees on small sized subsets results in higher AUC values compared with the forest grown on the original dataset.

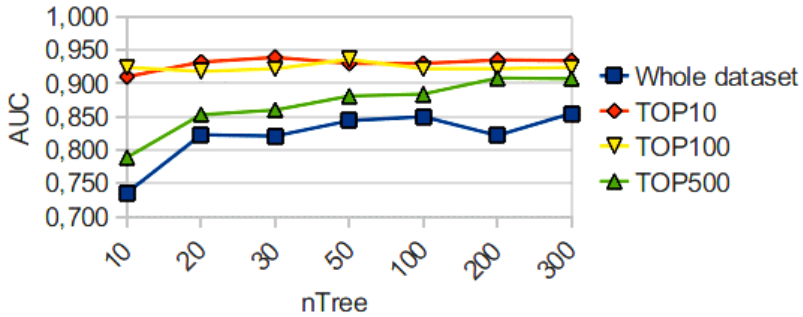

Fig.2. AUC: the whole dataset against TOP10, TOP100, TOP500 (*mtry* = 1)

Fig.3 and Fig. 4 show in detail AUC values on each filtered subset (*mtry* =1). For subsets in Fig. 3, we note that the smallest subset (i.e. TOP10) reaches the best AUC value within small values of *ntree* (*ntree* = 30). As Fig. 4 illustrates, as the subset size increases, the asymptotic behavior is more evident, but the AUC is lower and it is necessary growing random forests which contain a number of trees more and more greater.

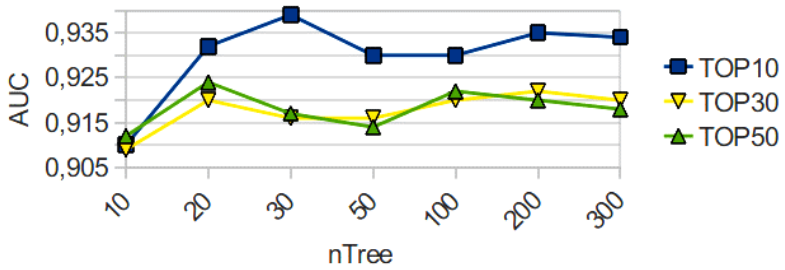

Fig.3. AUC versus *ntree* for TOP10,TOP30,TOP50 (*mtry* = 1)

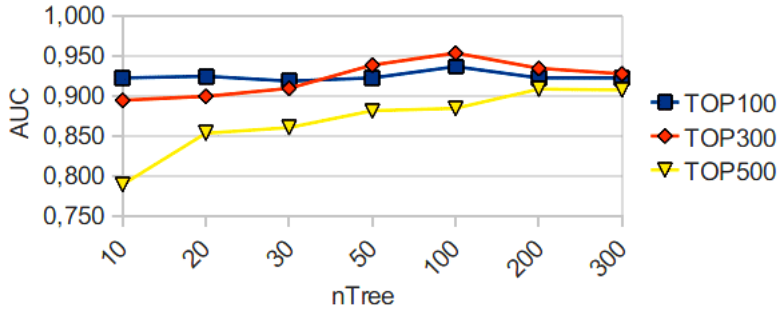

Fig.4. AUC versus *ntree* for TOP100, TOP300, TOP500 (*mtry* = 1)

In conclusion, tuning on the filtered subsets seems to suggest that the values *mtry* = 1 and *ntree* = 30 are a good option for a random forest on TOP10 which results in AUC values that considerably outperform the random forest on the whole dataset (2000 features).

As Table 1 shows, results on TOP10 within *mtry* = 1 and *ntree* = 30 compare well with results obtained in literature [5] by analogous experiments on random forests (with a different parameter setting) and by state-of-art methods.

Table 1. Comparison with results in literature

Method	**AUC**	
	On the full set of genes	**Using a filtered subset**
This paper	0,911	0,939
RF [5]	0,867	0,917
SVM [5]	0,867	0,938

5 Conclusions

This paper has presented an approach to microarray data classification that builds upon the known strengths of the random forests. Our method attempts to eliminate irrelevant variables by pre-filtering. Results on a diagnostic microarray dataset confirm what expected on the basis of similar studies on filtering methods when applied to microarray data classification. Our analysis suggests that the random forest method with pre-filtering allows the construction of a more parsimonious classification model with predictive performance better than the model implemented on the whole dataset. To our knowledge, this is the first work dedicated to the topic. However, we have demonstrated the effectiveness of a pre-filtering process with just a

single dataset and with only one filter method. In order to assert our first conclusions, it is necessary to further conduct experiments on different datasets and with more filter methods.

As such, our further research is currently exploring the pre-filtering advantages deriving from the application of random forest methods to additional microarray datasets.

Acknowledgments. This research was supported by RAS, Regione Autonoma della Sardegna (Legge regionale 7 agosto 2007, n. 7 "Promozione della ricerca scientifica e dell'innovazione tecnologica in Sardegna") in the project "DENIS: Dataspaces Enhancing the Next Internet in Sardinia".

References

1. Breiman, L.: Random forests. Mach Learn, vol. 45, pp. 5--32 (2001)
2. Amaratunga, D., Cabrera, J., Lee, Y.S.: Enriched random forest. Bioinformatics, vol.24, pp. 2010--2014, (2008)
3. Berthold, M. R.: Guide to Intelligent Data Analysis. Springer London (2010)
4. Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R., Feuston, B.P.: Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information and Computer Sciences 43, pp. 1947--1958 (2003)
5. Statnikov, A., Wang, L., Aliferis, C.F.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics 9:319 (2008)
6. Dìaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7:3 (2006)
7. Chen, X., Wang, M., Zhang, H.: The use of classification trees for bioinformatics. WIREs Data Mining Knowl Discov, vol. 1, pp. 55--63 (2011)
8. Zhang, H., Wang, M.: Search for the smallest random forest. Stat Interface 2:381 (2009)
9. Genuer, R., Poggi, J.M., Tuleau, C.: Random Forests: some methodological insights. INRIA (2008)
10. Liaw, A., Wiener, M.: Classification and Regression by random forest. R News, Vol. 2/3 pp. 18--22 (2002)
11. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS, vol. 96, pp. 6745--6750 (1999)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, vol. 11, Issue 1 (2009).
13. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining, Pearson (2006).
14. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report, HPL-2003-4, HP Laboratories (2003).