Inferring Statistically Significant Features from Random Forests

Jérôme Paul[a,*], Pierre Dupont[a]

[a] *Université catholique de Louvain – ICTEAM/Machine Learning Group, Place Sainte Barbe 2, B-1348 Louvain-la-Neuve, Belgium*

Abstract

Embedded feature selection can be performed by analyzing the variables used in a Random Forest. Such a multivariate selection takes into account the interactions between variables but is not straightforward to interpret in a statistical sense. We propose a statistical procedure to measure variable importance that tests if variables are significantly useful in combination with others in a forest. We show experimentally that this new importance index correctly identifies relevant variables. The top of the variable ranking is largely correlated with Breiman's importance index based on a permutation test. Our measure has the additional benefit to produce p-values from the forest voting process. Such p-values offer a very natural way to decide which features are significantly relevant while controlling the false discovery rate. Practical experiments are conducted on synthetic and real data including low and high-dimensional datasets for binary or multi-class problems. Results show that the proposed technique is effective and outperforms recent alternatives by reducing the computational complexity of the selection process by an order of magnitude while keeping similar performances.

Keywords: Feature Selection, Tree Ensembles, Significance tests, High-dimensional data analysis

*Corresponding author

Email addresses: jerome.paul@uclouvain.be (Jérôme Paul), pierre.dupont@uclouvain.be (Pierre Dupont)

[1] http://www.ucl.ac.be/mlg/

1. Introduction

Feature selection aims at finding a subset of most relevant variables for a prediction task. To this end, univariate filters, such as a t-test, are commonly used because they are fast to compute and their associated p-values are easy to interpret. However such a univariate feature ranking does not take into account the possible interactions between variables. In contrast, a feature selection procedure embedded into the estimation of a multivariate predictive model typically captures those interactions.

A representative example of such an embedded variable importance measure has been proposed by Breiman with its Random Forest (RF) [1] algorithm. While this importance index is effective to rank variables, it is difficult to decide how many such variables should eventually be kept. This question could be addressed through an additional validation protocol at the expense of an increased computational cost. In this work, we propose an alternative that avoids such additional cost and offers a statistical interpretation of the selected variables.

The proposed multivariate RF feature importance index uses out-of-bag (OOB) samples to measure changes in the distribution of class votes when permuting a particular variable. It produces p-values, corrected for multiple testing, measuring to which extent variables are useful in combination with other variables of the model. Such p-values offer a natural threshold for deciding which variables are statistically relevant.

The remainder of this document is organized as follows. Section 2 sets up the context and introduces our proposed variable importance measure relying on a χ^2 test. Section 3 describes the metrics, experimental protocol and datasets used to assess the performances of feature selection indices. Comparative experiments with state-of-the-art methods are reported in Section 4. Section 5 summarizes our contribution and discusses some possible future work.

2. Material and methods

This section presents a novel feature selection index from tree ensembles, typically a Random Forest. Section 2.1 introduces our notations and reviews Breiman's RF feature importance measure. Our proposed feature importance index is presented in Section 2.2, along with related work.

2.1. Context and notations

Let $X^{n\times p}$ be a data matrix consisting of n observations in a p-dimensional space and y a vector of size n containing the corresponding class labels. A RF model [1] is made of an ensemble of trees, each of which is grown from a bootstrap sample of the n data points. For each tree, the selected samples form the bag (denoted by B), the remaining samples form the out-of-bag (OOB) denoted by $\overline{B}$. Let $\mathcal{B}$ stand for the set of bags over the ensemble and $\overline{\mathcal{B}}$ be the set of corresponding OOBs. We have $|\mathcal{B}| = |\overline{\mathcal{B}}| = T$, the number of trees in the forest.

In order to compute feature importances, Breiman [1] proposes a permutation test procedure based on classification error. For each variable x_j, there is one permutation test per tree in the forest. For an OOB sample $\overline{B}_k$ corresponding to the k-th tree of the ensemble, one considers the original values of the variable x_j and a random permutation $\tilde{x}_j$ of its values on $\overline{B}_k$. The difference in prediction error using the permuted and original variable is recorded and averaged over all the OOBs in the forest. The higher this index, the more important the variable is assumed because it corresponds to a stronger increase of the classification error when permuting it. The importance measure J_a of the variable x_j is precisely defined as:

$$J_a(x_j) = \frac{1}{T} \sum_{\overline{B}_k \in \overline{\mathcal{B}}} \frac{1}{|\overline{B}_k|} \left(\sum_{i \in \overline{B}_k} I(h_k^{\tilde{x}_j}(i) \neq y_i) - I(h_k(i) \neq y_i) \right) \qquad (1)$$

where y_i is the true class label of the OOB example i, I is an indicator function, $h_k(i)$ is the class label of the example i as predicted by the tree estimated on the bag B_k, $h_k^{\tilde{x}_j}(i)$ is the predicted class label from the same tree while the values of the variable x_j have been permuted on $\overline{B}_k$. Such a permutation does not change the tree but potentially changes the prediction on the out-of-bag examples since its j-th dimension is modified after the permutation. Since the predictors with the original variable h_k and the permuted variable $h_k^{\tilde{x}_j}$ are individual decision trees, the sum over the various trees where this variable is present represents the ensemble behavior, respectively from the original variable values and its various permutations. Whenever a specific variable does not appear in a tree, the prediction cannot be affected by permuting its value, which means that the specific term corresponding to this tree in equation (1) is null.

2.2. A statistical feature importance index from RF

While J_a is able to capture individual variable importances conditioned to the other variables used in the forest, it is not easily interpretable. In particular, it does not define a clear threshold to highlight statistically relevant variables. In the following sections, we propose a statistical feature importance measure closely related to J_a, and compare it with existing approaches providing a statistical interpretation to feature importance scores.

2.2.1. Definition

In the present work, we combine the idea of Breiman's J_a to use a permutation test with an analysis of the tree class vote distribution of the forest. We propose to perform a statistical test that assesses whether permuting a variable significantly influences that distribution. The hypothesis is that removing an important variable signal by permuting it should change individual tree predictions, hence the class vote distribution.

One can estimate this distribution using the OOB data to simulate unseen examples. In a binary classification setting, for each data point in an OOB, the prediction of the corresponding tree can fall into one of the four following cases : correct prediction of class 1 (TP), correct prediction of class 0 (TN), incorrect prediction of class 1 (FP) and incorrect prediction of class 0 (FN). Summing the occurrences of those cases over all the OOBs gives an estimate of the class vote distribution of the whole forest on unseen examples. The same can be performed when permuting a particular feature x_j to evaluate the effect on the class vote distribution after perturbing this variable. The various counts obtained can be arranged into a 4×2 contingency table defined as follows for each variable x_j and its permuted version $\tilde{x}_j$:

	x_j	$\tilde{x}_j$
TN	$s(0,0)$	$s^{\tilde{x}_j}(0,0)$
FP	$s(0,1)$	$s^{\tilde{x}_j}(0,1)$
FN	$s(1,0)$	$s^{\tilde{x}_j}(1,0)$
TP	$s(1,1)$	$s^{\tilde{x}_j}(1,1)$

(2)

where

$$s(l_1, l_2) = \sum_{\overline{B}_k \in \overline{\mathcal{B}}} \sum_{i \in \overline{B}_k} I(y_i = l_1 \text{ and } h_k(i) = l_2) \tag{3}$$

and $s^{\tilde{x}_j}(l_1, l_2)$ is defined similarly with $h_k^{\tilde{x}_j}(i)$ instead of $h_k(i)$.

A Pearson's χ^2 test is then used to assess whether the frequencies of those events significantly differ from the original x_j and its permuted version $\tilde{x}_j$. Rejecting the null hypothesis with a low p-value $p_{\chi^2}(x_j)$ means that permuting variable x_j significantly influences the class vote distribution and, therefore, that x_j is important in the current predictive model. We note that, even on small datasets, there is no need to consider a Fisher's exact test instead of Pearson's χ^2 since cell counts are generally sufficiently large: the sum of all counts is twice the sum of all OOB sizes.

Since the importance of several features is typically assessed through this test on the same data, p-values must be corrected for multiple testing. We use the popular Benjamini-Hochberg correction [2] to control the false discovery rate. Let $p^{fdr}_{\chi^2}(x_j)$ be the value of $p_{\chi^2}(x_j)$ after FDR correction, the new importance measure is defined as

$$J_{\chi^2}(x_j) = p^{fdr}_{\chi^2}(x_j) \tag{4}$$

The proposed index can easily be generalized to multi-class problems, as used in some of the experiments reported in Section 3. In such cases, the contingency table (2) simply has $c^2 \times 2$ entries, where c is the number of classes.

This statistical importance index is closely related to Breiman's J_a. The two terms inside the innermost sum of Equation (1) correspond to counts of FP et FN for permuted and non permuted variable x_j. This is encoded by the second and third lines of the contingency table (2). There are some important differences between both approaches however. Firstly, J_a aggregates both type of errors in a single measure, which might loose important information in case of unbalanced class priors. Secondly, the central term of J_a (eq. (1)) is normalized by each OOB size while the contingency table of J_{χ^2} (eq. (2)) considers global counts. This follows from the fact that J_a estimates an average increase in classification error on the OOB samples while J_{χ^2} measures a distribution shift on those samples. Finally, the very nature of those importance indices differ. J_a is an average measure of differences between prediction performances whereas J_{χ^2} (eq. (4)) is a corrected p-value from a χ^2 test. The higher J_a the more important is the corresponding variable assumed. In contrast, the lower J_{χ^2} the stronger the evidence to reject the null hypothesis that permuting this variable does not affect the voting process of a RF. There is also a natural significance threshold for J_{χ^2} since any corrected p-value lower than 5% is commonly accepted as significant [3].

The time complexity of computing J_{χ^2} for p variables is exactly the same as with Breiman's J_a. If we assume, to simplify the analysis, that each tree node splits its instances into two sets of equal sizes until having one observation per leaf, then the depth of a tree is in $O(\log n)$ and the time complexity of classifying one example by a single tree is $O(\log n)$. The global time complexity of computing a ranking of p variables from an ensemble of T trees is in $O(T \cdot p \cdot n \cdot \log n)$. Algorithm 1 describes the computation of J_{χ^2} and motivates its time complexity analysis.

```
init(res)                                  // Set to 0 a p-dimensional vector; Θ(p)
for x_j ∈ Variables do                                                    // Θ(p)
    init(contTable) // Set to 0 the counts of a contingency table; Θ(1)
    for B̄_k ∈ B̄ do                                                        // Θ(T)
        x̃_j ← perm(x_j, B̄_k)                                              // Θ(n)
        for i ∈ B̄_k do                                                    // O(n)
            a ← h_k(i)                                                  // Θ(depth)
            b ← h_k^{x̃_j}(i)                                            // Θ(depth)
            contTable ← update(contTable, a, b, y_i)                    // Θ(1)
        end
    end
    res[x_j] ← χ²(contTable)                                              // Θ(1)
end
return res
```

Algorithm 1: Algorithm for computing the importance of all variables within a forest of $T = |\overline{\mathcal{B}}|$ trees.

2.2.2. Related work

In [4], the authors compare several ways to obtain a statistically interpretable index from a feature relevance score. Their goal is to convert feature rankings to statistical measures such as the false discovery rate, the family wise error rate or p-values. Their proposed methods typically make use of an external permutation procedure to compute some null distribution from which those metrics are estimated. The external permutation tests repeatedly compute feature rankings on dataset variations, *e.g.* for which some features are randomly permuted.

Similarly to our approach, this work can be applied to convert Breiman's J_a index to a statistically interpretable measure and to produce p-values on

which a prescribed threshold can be easily defined. Yet, the methods proposed in [4] are somewhat more complex since they rely on an additional resampling protocol. This *external* resampling encompasses the growing of many forests on top of the *internal* bootstrap mechanism at the tree level, used while growing each forest. This external resampling relies on additional meta-parameters such as the number N of external resamplings and the number of instances to be sampled. Among the various methods presented in [4] that resort on an external permutation procedure, two techniques specifically produce p-values. They both rely on feature rankings produced according to Breiman's J_a index but differ in the way a null distribution is estimated.

The *mr-Test* [5] repetitively samples $\frac{n}{2}$ examples out of n without replacement. It also assumes that a prescribed fraction (by default, $\frac{p}{2}$ out of p) of variables are irrelevant. Out of N resamplings ($N \geq 100$ is typically chosen, see section 3), the null distribution is defined as the rank distribution of the worst $\frac{p}{2}$ variables according to their average J_a values. For each remaining variable, its average rank over the N resamplings is compared to the null distribution, which defines its associated p-value.

The *1Probe* ranks N times the features of the whole dataset (the n observations) after introducing an additional non-informative feature randomly sampled from $\mathcal{N}(0, 1)$ at each iteration. The p-value of a feature is then estimated as the proportion of iterations for which the non-informative variable has a better rank according to J_a. For both methods, the final p-values are corrected for multiple testing using the Benjamini-Hochberg procedure [2].

The *mr-Test* and *1Probe* have a higher computational complexity than the evaluation of the J_{χ^2} importance index. Indeed, they multiply the cost of computing a ranking with Brieman's original J_a by the number N of external resamplings. We further analyze in section 4.4 the cost/performance trade-off of those approaches.

3. Experiments

This section describes various metrics to assess the performance of feature selection methods, our experimental protocol and the datasets on which we run our experiments.

3.1. Performance Metrics

The Balanced Classification Rate (BCR) is used to assess the predictive performances of a classifier estimated on the selected features. It is defined

as the mean of the classification accuracy in each class. BCR is preferred to the standard classification rate when dealing with unbalanced class priors. It also generalizes to multi-class problems more easily than ROC analysis. For a two-class problem, BCR is defined as the average between sensitivity and specificity:

$$BCR = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) \tag{5}$$

Its multi-class generalization takes the following form:

$$BCR = \frac{1}{c}\sum_{l=1}^{c}\frac{TC_l}{C_l} \tag{6}$$

where c is the number of classes, TC_l is the number of correct predictions of class l and C_l is the total number of samples of class l. This metric has been used, for instance, in the performance prediction challenge[2] held at WCCI 2006 precisely to deal with possible class imbalance while considering the calibration of specific models [6].

Stability of feature selection indices quantifies how selected sets of features vary after small perturbations of the datasets. The Kuncheva index (KI) [7] specifically measures to which extent K sets (typically obtained from various resamplings) of s selected features share common elements.

$$KI(\{S_1, ..., S_K\}) = \frac{2}{K(K-1)}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\frac{|S_i \cap S_j| - \frac{s^2}{p}}{s - \frac{s^2}{p}} \tag{7}$$

where p is the total number of features and $\frac{s^2}{p}$ is a term correcting the random chance, for 2 feature sets S_i and S_j, to share common features. This index ranges within $(-1, 1]$. The larger its value, the larger the number of commonly selected features. A value of 0 is the expected stability for a selection performed uniformly at random.

3.2. Experimental protocol

In order to evaluate the predictive performances and the stability provided by a feature selection technique, an *external* resampling protocol is used. The

[2]The evaluation metric in this challenge actually relied on BER, the balanced error rate, which conveys the same information since $BCR = 100\% - BER$.

goal is twofold. Firstly, resampling allows to assess how a particular classifier built on the selected features will predict the class of new data. Secondly, it mimics small perturbations in datasets to assess the stability of feature selection. The procedure consists in repeating N times the following steps:

randomly select a training set Tr made of 90% of the available data. The remaining 10% form the test set Te.

- train a forest of T trees on Tr and rank the features it uses
- for each number of selected features s
  - train a forest of 500 trees using only the first s features on Tr
  - save the BCR computed on Te and the set of s features

The statistics recorded at each iteration are then aggregated to provide the mean BCR and the KI values. The above protocol considers several feature set sizes s. The results presented in section 4.1 reports, on several datasets, how many out of s are actually significant features.

3.3. Datasets

Artificial datasets allow to control by design the signal present in different features. Our first experiments are inspired from [4] and conducted on artificial datasets with a linear decision boundary. Labels $y \in \{-1, 1\}^n$ are given by $y = \text{sign}(Xw)$ where $w \in \mathbb{R}^p$ and $X \in \mathbb{R}^{n \times p}$. Each dimension from the input data X is repetitively drawn from a $\mathcal{N}(0, 1)$ distribution. The number p of variables is set to 110. The first 10 weights w_i are randomly sampled from a uniform distribution $\mathcal{U}(0.5, 1)$. The other 100 weights are set to 0 such that only the first 10 variables are relevant. We draw $n = 500$ instances for a given run with a design matrix $X \in \mathbb{R}^{500 \times 110}$. Finally, 10% of the y labels are randomly flipped to add some noise to the classification task.

Experiments are also performed on real-life datasets, briefly described in Table 1 in terms of class priors and number of input features. We consider firstly four gene expression datasets from a microarray technology. The number of features p in those datasets is typically much larger than the number n of training examples. In such a challenging setting, feature selection is usually considered particularly important. The DLBCL [8] dataset aims at predicting the outcome of diffuse large b-cell lymphoma. The prediction task associated to the Lymphoma [9] dataset concerns the identification of different subtypes of this pathology. Golub's dataset [10] aims at identifying

Name	Class priors	p
DLBCL [8]	58/19	7129
Lymphoma [9]	22/23	4026
Golub [10]	25/47	7129
Prostate [11]	52/50	6033
Breast tissue [12]	22/21/14/49	9
Glass [12]	70/76/17/13/9/29	9
Wine [12]	59/71/48	13
Vehicle [12]	218/212/217/199	18
Musk1 [12]	269/207	166
Arrhythmia [12]	245/185	262

Table 1: Summary of the real-life datasets: class priors report the n values in each class, p represents the total number of features.

different types of cancer. Finally, the Prostate [11] dataset focuses on the diagnostic of prostate cancer or healthy patients from their gene expression. Since the number of features in those datasets is orders of magnitude higher than the number of available samples, a non-specific filter (*i.e.* without considering the class labels) is applied first to remove 75% of the features with the lowest variance on the training set.

In addition, we consider six lower dimensional datasets with, proportionally, a larger number of training examples. Breast tissue [12] is a four classes dataset made of impedance measurements to predict the type of observed tissue. The Glass [12] dataset aims at classifying fragments of glass into seven different types using proportions of chemical elements that compose each fragment. The Wine [12] dataset consists of chemical measurements aiming at predicting from which of three domains comes a particular wine. The purpose of the Vehicle [12] dataset is to distinguish between four vehicle types given some geometrical features extracted from their silhouettes. The Musk1 [12] dataset describes two kinds of molecules (musk and non-musk) in terms of shape and conformation of the molecules. Finally, the Arrhythmia [12] dataset aims at predicting the presence of cardiac arrhythmia from ECG measurements.

4. Results and discussion

The following sections present experiments that highlight properties of the J_{χ^2} importance measure. They show that J_{χ^2} actually provides an importance index from which a natural selection threshold can be chosen (Section 4.1). Our results also illustrate that J_{χ^2} is closely related to J_a (Section 4.2), the original Brieman's index, both in terms of variable rankings and predictive performances after building a classifier on the selected features. Further experiments described in Section 4.3 present predictive performances obtained when restricting the classifier to be built only from variables which are deemed statistically significant. Finally, Section 4.4 details the relative performance of the J_{χ^2} and the two competing approaches *mr-Test* and *1Probe*.

4.1. Selecting statistically relevant features with J_{χ^2}

The expected benefit of J_{χ^2} is to a offer a principled way to select key variables from a tree ensemble. One aims at restricting the selection to those variables that are deemed significant for characterizing the class vote distribution in such an ensemble. We assess here to which extent this criterion matches the selection of relevant variables by design on an artificial dataset (cf. Section 3.3).

A RF, built on the full dataset, is used to rank the variables according to their importance index. Similarly to [4], a given variable is considered significantly important, whenever its p-value falls below 0.05 after correcting for multiple testing. Figure 1 reports importance indices obtained by forests of various sizes and m values. This meta-parameter m corresponds to the number of variables randomly sampled as possible candidates in each tree node while growing the forest. The specific ensemble sizes are chosen according to [13] in which the stability of such ensembles is studied. This work shows in particular that a forest of 500 trees performs quite well in terms of predictive performances on such high dimensional datasets while a much larger ensemble of about 10,000 trees is required to reach a stable feature selection. In the four plots of Figure 1, the 10 informative features appear at the top of the rankings of J_a and J_{χ^2}.

The results reported in Figure 1 illustrate that the original (decreasing) Breiman's J_a index does not offer a clear threshold to decide which variables are relevant. Our (increasing) J_{χ^2} index appears to distinguish more clearly between relevant and irrelevant variables. It however requires a relatively

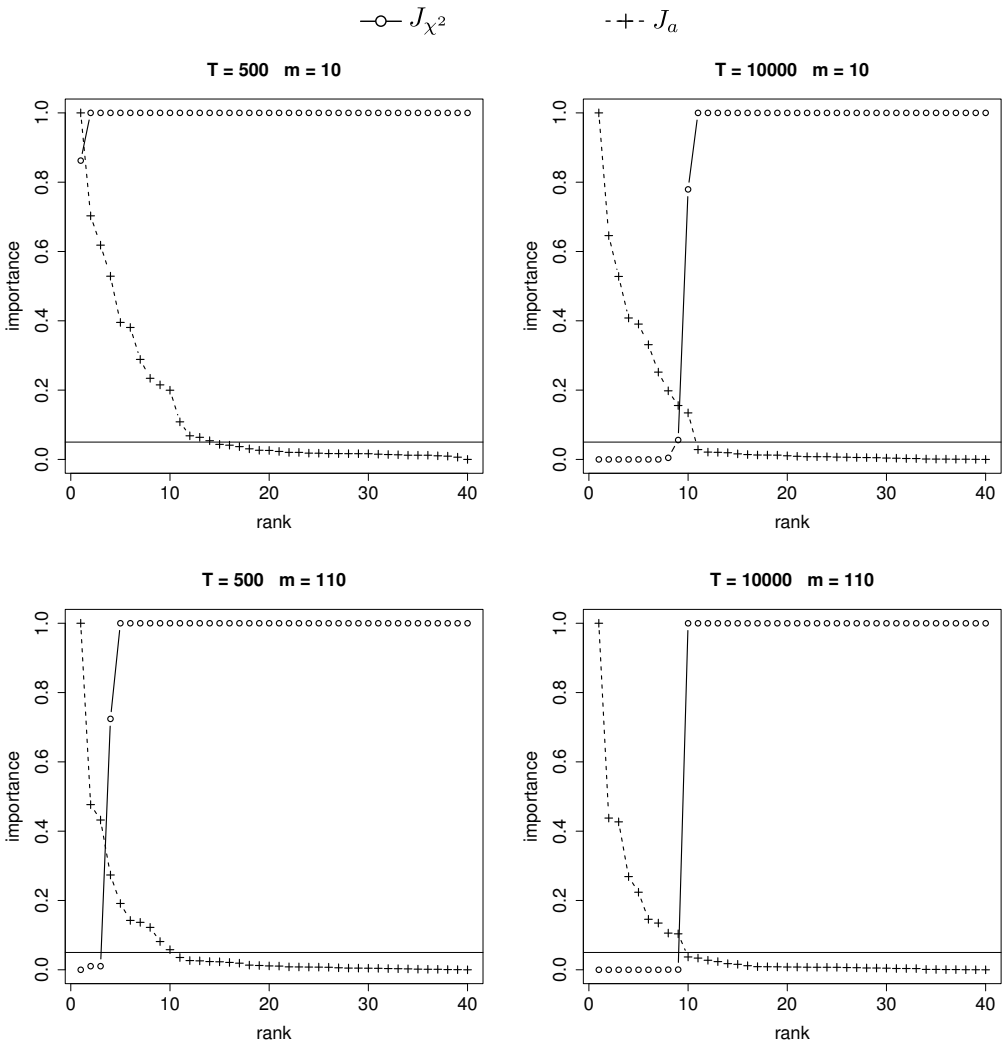

Figure 1: Importance indices computed on an artificial dataset with 10 informative features out of 110 features in total. Results are reported for various forest sizes (T) and m values (see text). For the sake of visibility, J_a has been rescaled between 0 and 1. The horizontal line is set at 0.05. $J_{\chi^2}(x_j)$ below this line are deemed statistically relevant. All 10 informative features appear at the top of each ranking in the four plots.

large number of trees to gain confidence that a feature is indeed relevant. When computed on small forests (left plots), J_{χ^2} may fail to identify variables as significantly important. Nevertheless those variables are still correctly ranked. Increasing the value of the m meta-parameter also tends to positively impact the identification of those variables when the number of trees is low. This beneficial effect appears less strongly as the number of trees increases. In general, the larger the forests the better, in terms of the significance of the test. Beyond significance, the effect size could also be assessed as briefly discussed in section 5.

4.2. Concordance with J_a

As discussed in Section 2.2.1, J_{χ^2} and J_a share some similarities and the same computational complexity to be evaluated. The left plot of Figure 2 compares the rankings of those two importance measures on one particular resampling of the DLBCL dataset (cf. Section 3.3). It shows that feature ranks in the top 500 are highly correlated. Spearman's rank correlation coefficient is 0.97 between both rankings.

The main differences are observed in the poorly ranked features, which are those very unlikely to be considered significant. While J_a penalizes features whose permuted versions would increase the prediction accuracy, J_{χ^2} would favor such features since they affect the class vote distribution. In particular, after rank 1,250 on the horizontal axis, features have a negative J_a value for they lower the prediction performance of the forest. Yet, since they influence the class vote distribution, they are considered more important by J_{χ^2}.

This behavior of J_{χ^2} could be considered undesirable but the actual effect is negligible in practice because the large ranks of those variables indicate that they are very unlikely to be eventually selected. This is further confirmed by the right plot of Figure 2 where the mean rank of each variable is computed over 200 resamplings. We can see that this effect totally disappears and is only due to random variations on those features. To sum up, this particular behavior of J_{χ^2} has virtually no practical impact since only top ranked features will typically be selected based on their low corrected p-values.

We further show that J_a and J_{χ^2} are also similar in terms of stability of the feature selection and predictive performances of the final classifier built from the selected features. Figure 3 presents the measurements made over 200-resamplings from the DLBCL dataset according to the number of features kept to train a RF as final classifier. It shows that the two indices

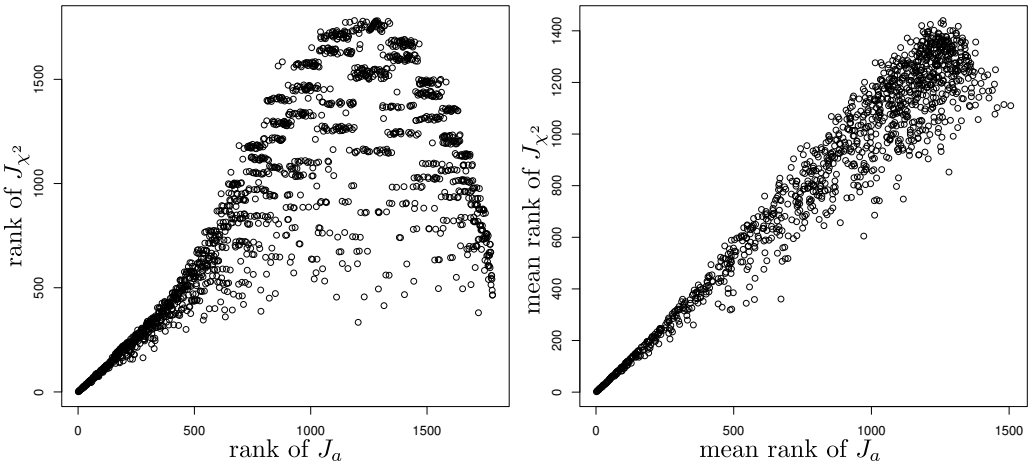

Figure 2: Left: Rankings produced by J_a and J_{χ^2} on one external resampling of the DLBCL dataset. Right: Mean rankings produced by J_a and J_{χ^2}, averaged over 200 resamplings of the DLBCL dataset. Approximately 1,800 features are ranked after prefiltering 75 % of features with the lowest variances.

behave very similarly when plotting predictive performance with respect to the number of selected features. Increasing the number of trees increases the feature selection stability in both cases, with a convergence of the stability curves observed from 5000 trees. Similar results have been reported for J_a in [13].

4.3. Prediction from significant features

The previous results show that J_{χ^2} ranks top features roughly the same way as J_a. Since J_{χ^2} is a corrected p-value, it is associated to a commonly accepted threshold equal to 0.05 to decide whether a variable should eventually be kept. We recall that this selection process is no longer univariate (as would be the case of a t-test) but is performed while considering each variable jointly with the others in the forest. One can wonder whether restricting the final classifier to be built strictly on the features which are deemed significant still offers good predictive results.

We follow here the protocol of Section 3.2 and observe that the number of significant variables increases with the number of trees considered. This is consistent with the results already presented in Section 4.1. Table 2 specifically reports the results obtained from genomic data. The number of variables eventually considered significant largely varies across datasets.

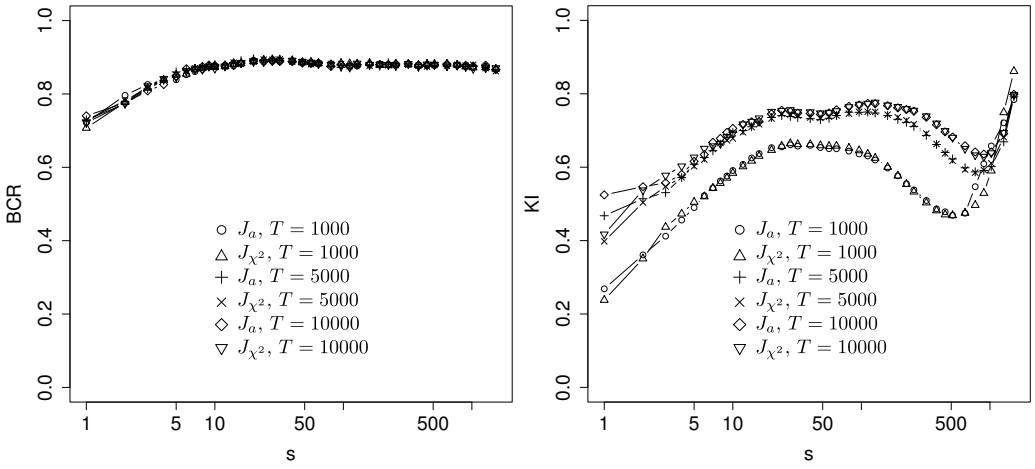

Figure 3: Average BCR and KI of J_a and J_{χ^2} over 200 resamplings of the DLBCL dataset according to the number s of selected features, for various numbers T of trees.

Almost no features are considered statistically significant on the DLBCL dataset (similar results are observed in [4]). For the three other datasets and provided the number of trees is sufficiently large, the predictive performances of a RF built on significant features only (according to J_{χ^2}) are similar to those of a RF built on the top 50 features according to J_a. While those predictive performances are close (especially with 10,000 trees) the differences are statistically significant in most cases. Yet, it is worth stressing that the final forest can be built on only a few key features with good predictive results.

Table 3 reports similar results on the lower dimensional datasets, with different ratios between the numbers of learning examples and input features, including multi-class problems. As before, the average number of significant features increases with the number of trees. On the Breast tissue, Glass, Wine and Vehicle datasets, results show that nearly all features appear significant even with only 500 trees. Overall these results are fully consistent with those observed on the high dimensional genomic datasets but the number of trees needed to highlight relevant features that lead to good predictive performances tends to be lower.

4.4. Comparison of J_{χ^2} to 1Probe and mr-Test

In this section we compare J_{χ^2} to 1Probe and mr-Test, two methods proposed in [4] and briefly reviewed in section 2.2.2. Section 4.4.1 compares

dataset	T	avg(s^{rel})	min(s^{rel})	max(s^{rel})	BCR	BCR^{50}
DLBCL	5000	0.04	0.00	1.00	0.50	0.89
	10,000	0.99	0.00	5.00	0.61	0.88
golub	5000	5.96	3.00	10.00	0.93	0.97
	10,000	10.82	8.00	14.00	0.96	0.97
lymphoma	5000	0.66	0.00	6.00	0.54	0.94
	10,000	4.85	2.00	9.00	0.93	0.94
prostate	5000	4.95	2.00	8.00	0.93	0.94
	10,000	7.92	6.00	11.00	0.93	0.94

Table 2: Various statistics obtained over 200-resamplings when keeping only significant features. T is the number of trees used to build the forest. *avg(s^{rel})* (resp. *max*, *min*) is the average (resp. maximum, minimum) number of significant features according to J_{χ^2}. *BCR* is the average predictive performance of a RF built from significant features only. BCR^{50} is the average BCR obtained when using the 50 best ranked features according to J_a.

the number of trees needed to highlight important variables from synthetic datasets. Section 4.4.2 compares the predictive performances and signatures obtained using only significant variables.

4.4.1. Discovery rates evaluated on synthetic datasets

The performances of 1Probe and mr-Test on synthetic datasets are assessed in [4] using $T = 1000$ trees and $N = 1000$ external resamplings. The total number of trees considered is therefore $1,000,000$ in contrast to J_{χ^2} which show comparable results with only $10,000$ trees (and no external resampling).

We aim here at comparing the 3 approaches with a similar computational budget and perform the same experiment on synthetic datasets as in Section 4.1 with $T = 10,000$ for J_{χ^2} and $N \times T = 100 \times 100$ for 1Probe and mr-Test. This setting appears to be inadequate for the latter approaches. The number of trees T in the forest for each resampling is too low to rank variables correctly. The two methods hardly find any of the important variables.

Better results are reported on Figure 4 where the 10 informative features appear at the top of the ranking of each of the three methods. The number of trees is increased to $T = 1000$ (a total of $100,000$ trees over all resamplings). J_{χ^2} (with 10 times fewer trees in total) and 1Probe are both able to highlight significant variables at the top of the ranking. In contrast, mr-Test does not

dataset	T	avg(s^{rel})	min(s^{rel})	max(s^{rel})	BCR	BCR*
Breast tissue	50	6.64	3.00	8.00	0.86	0.86
	100	7.75	5.00	8.00	0.85	0.86
	250	7.99	7.00	8.00	0.85	0.86
	500	8.00	8.00	8.00	0.85	0.86
	1000	8.00	8.00	8.00	0.85	0.86
	2500	8.01	8.00	9.00	0.85	0.86
	5000	8.21	8.00	9.00	0.85	0.86
	10,000	8.90	8.00	9.00	0.86	0.86
Glass	50	4.74	1.00	7.00	0.67	0.74
	100	6.66	3.00	8.00	0.73	0.74
	250	7.96	5.00	8.00	0.74	0.74
	500	8.00	8.00	8.00	0.74	0.74
	1000	8.01	8.00	9.00	0.74	0.74
	2500	8.49	8.00	9.00	0.74	0.74
	5000	8.94	8.00	9.00	0.74	0.74
	10,000	9.00	9.00	9.00	0.74	0.74
Wine	50	7.12	6.00	9.00	0.98	0.98
	100	7.85	7.00	10.00	0.98	0.98
	250	9.86	8.00	12.00	0.98	0.98
	500	11.04	10.00	12.00	0.98	0.98
	1000	11.49	11.00	13.00	0.98	0.98
	2500	12.76	11.00	13.00	0.98	0.98
	5000	12.99	12.00	13.00	0.98	0.98
	10,000	13.00	13.00	13.00	0.98	0.98
Vehicle	50	15.89	14.00	18.00	0.75	0.75
	100	17.32	16.00	18.00	0.75	0.75
	250	18.00	18.00	18.00	0.75	0.75
	500	18.00	18.00	18.00	0.75	0.75
	1000	18.00	18.00	18.00	0.74	0.74
	2500	18.00	18.00	18.00	0.74	0.74
	5000	18.00	18.00	18.00	0.74	0.74
	10,000	18.00	18.00	18.00	0.74	0.74
Musk1	50	0.02	0.00	1.00	0.50	0.89
	100	0.25	0.00	2.00	0.53	0.89
	250	1.91	0.00	5.00	0.67	0.89
	500	4.89	3.00	9.00	0.73	0.89
	1000	12.15	7.00	18.00	0.81	0.89
	2500	31.62	21.00	38.00	0.87	0.89
	5000	63.23	52.00	77.00	0.90	0.89
	10,000	102.08	89.00	118.00	0.90	0.89
Arrhythmia	50	0.00	0.00	0.00	0.50	0.84
	100	0.10	0.00	1.00	0.50	0.85
	250	1.09	0.00	2.00	0.60	0.85
	500	2.29	1.00	5.00	0.68	0.85
	1000	6.17	2.00	11.00	0.78	0.85
	2500	15.42	10.00	21.00	0.82	0.85
	5000	24.19	18.00	31.00	0.84	0.85
	10,000	36.13	29.00	42.00	0.85	0.85

Table 3: Various statistics obtained over 200-resamplings when keeping only significant features. T is the number of trees used to build the forest. *avg(s^{rel})* (resp. *max*, *min*) is the average (resp. maximum, minimum) number of significant features according to J_{χ^2}. *BCR* is the average predictive performance of a RF built from significant features only. *BCR** is the average BCR obtained when using the 50 best ranked features according to J_a or all the available features if the number of variables is less than 50.

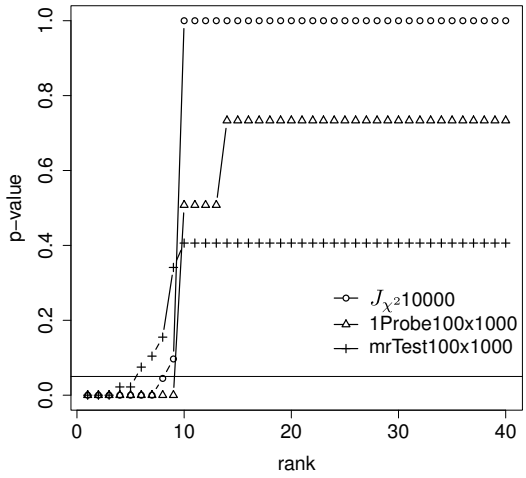

Figure 4: Importance indices computed on an artificial dataset with 10 informative features out of 110 features in total. The horizontal line is set to 0.05. p-values below this line are deemed statistically relevant. The 10 informative features are ranked at the top of those rankings.

consider them significant even though they are well ranked.

After repeating the above experiment on 10 runs for generating synthetic datasets, it appears that the number of true discoveries (i.e. p-value ≤ 0.05 and actually informative feature) is on average 6.8 for J_{χ^2}, 7.2 for 1Probe and 3.4 for mr-Test. A Friedman test [14] shows a significant difference between the performances of the three approaches (p-value of 6×10^{-3}).

The Nemenyi post-hoc test [14], illustrated by a critical difference diagram on Figure 5, shows that the difference in performances of 1Probe and J_{χ^2} is not significant while the mr-Test performs significantly worse. All methods have a very high precision with an average of 0.1 false discoveries (i.e. p-value ≤ 0.05 and not informative feature) for J_{χ^2}, 0.3 for 1Probe and 0 for mr-Test. None of those differences are statistically significant according to a Friedman test.

In summary, the ability of J_{χ^2} and 1Probe to discover informative variables and discard non-informative variables is essentially the same but 1Probe requires an order of magnitude more trees. In contrast, mr-Test is too conservative as it typically misses informative variables which are wrongly considered not significant.

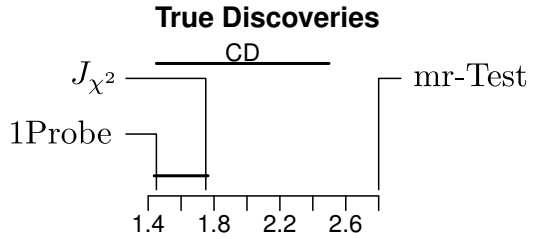

Figure 5: Nemenyi critical difference diagram. Importance indices are sorted according to their mean rank while counting the number of true discoveries. The critical difference is represented by the 'CD' black line.

4.4.2. Prediction from significant features

Similar experiments as those described in Section 4.3 are conducted on the datasets presented in Section 3.3. We report here comparative predictive performances between the 3 methods when estimating a final RF only on the features that are considered significant.

Table 4 reports the results obtained with 100 resamplings to evaluate those performances on the genomic datasets. Table 6 and Table 7 report results on the lower dimensional datasets. Whenever a specific approach does not select any feature as being significant in a particular resampling, the BCR is fixed to 0.5 since no classifier can be built from zero features and one has to resort on random guessing. The null distribution is also difficult to define with mr-Test when none of the $\frac{p}{2}$ worst features actually appears in the tree ensembles. In such cases, occurring in particular with few trees, a default BCR=0.5 is also considered.

As observed in [4], increasing the number of trees promotes the selection of larger subsets of features. On genomic datasets, considering 100×1000 trees with 1Probe provides very good predictive performances. mr-Test appears a lot more conservative and tends to select very few or no features. Whenever at least a few genes are considered significant with mr-Test (only for the Golub and Prostate datasets), good predictive performances are observed. The number of J_{χ^2}'s significant features also increases with the number of trees (cf. section 4.1), providing the best predictive performances with $100,000$ trees. The average number of selected features of J_{χ^2} typically falls between the results of mr-Test and 1Probe. A finer analysis of the gene signatures is presented in Table 5. It shows that nearly all features estimated to be significant by J_{χ^2} with $100,000$ trees belong to the feature sets selected with 1Probe. In summary, J_{χ^2} with 10,000 trees already offers good predictive

dataset	method	T or NxT	avg(s^{rel})	min(s^{rel})	max(s^{rel})	BCR
DLBCL	J_{χ^2}	10,000	0.88	0	4	0.60
	1Probe	100x100	0.02	0	1	0.50
	mrTest	100x100	0	0	0	0.50
	J_{χ^2}	100,000	27.53	22	37	0.89
	1Probe	100x1000	45.38	33	60	0.88
	mrTest	100x1000	0.03	0	2	0.50
golub	J_{χ^2}	10,000	10.80	8	13	0.96
	1Probe	100x100	0.42	0	2	0.64
	mrTest	100x100	0	0	0	0.50
	J_{χ^2}	100,000	40.83	35	50	0.97
	1Probe	100x1000	67.00	53	78	0.97
	mrTest	100x1000	9.87	3	15	0.95
lymphoma	J_{χ^2}	10,000	4.83	2	8	0.93
	1Probe	100x100	0.18	0	2	0.54
	mrTest	100x100	0	0	0	0.50
	J_{χ^2}	100,000	27.72	22	34	0.94
	1Probe	100x1000	36.45	24	51	0.94
	mrTest	100x1000	0.13	0	5	0.50
prostate	J_{χ^2}	10,000	7.89	6	11	0.94
	1Probe	100x100	2.98	1	6	0.92
	mrTest	100x100	0	0	0	0.50
	J_{χ^2}	100,000	41.52	34	53	0.94
	1Probe	100x1000	50.20	21	71	0.94
	mrTest	100x1000	7.70	5	11	0.93

Table 4: Various statistics obtained over 100-resamplings when keeping only significant features. T is the number of trees used to build the forest. For 1Probe and mr-Test, N indicates the number of external resamplings to compute the null distribution. *avg(s^{rel})* (resp. *max*, *min*) is the average (resp. maximum, minimum) number of significant features. *BCR* is the average predictive performance of a RF built only from features which are estimated significant.

dataset	methods	J_{χ^2} 10,000	J_{χ^2} 100,000	1Probe 100x1000	mrTest 100x1000
DLBCL	J_{χ^2}10,000	1.00	1.00	1.00	0.02
	J_{χ^2}100,000	0.03	1.00	0.98	0.00
	1Probe100x1000	0.02	0.60	1.00	0.00
	mrTest100x1000	1.00	1.00	1.00	1.00
golub	J_{χ^2}10,000	1.00	1.00	1.00	0.87
	J_{χ^2}100,000	0.27	1.00	0.99	0.24
	1Probe100x1000	0.16	0.61	1.00	0.15
	mrTest100x1000	0.95	1.00	1.00	1.00
lymphoma	J_{χ^2}10,000	1.00	1.00	1.00	0.02
	J_{χ^2}100,000	0.18	1.00	0.98	0.01
	1Probe100x1000	0.13	0.75	1.00	0.00
	mrTest100x1000	1.00	1.00	1.00	1.00
prostate	J_{χ^2}10,000	1.00	1.00	1.00	0.89
	J_{χ^2}100,000	0.19	1.00	0.93	0.19
	1Probe100x1000	0.16	0.78	1.00	0.16
	mrTest100x1000	0.92	1.00	1.00	1.00

Table 5: Average proportion of common genes between various feature sets. The entry at row i and column j in this matrix represents the average proportion, over 100-resamplings, of significant features selected by the index i that also belong to the feature set selected by method j for the same sampling.

performances on genomic datasets except on DLBCL where more trees are required. 1Probe typically requires an order of magnitude more trees to offer competitive results while mr-Test is often too conservative and selects too few important features.

Experiments on lower dimensional datasets are performed with a smaller number of trees. Table 6 shows that J_{χ^2} and 1Probe perform similarly in terms of predictive performances on the Breast tissue, Glass, Wine and Vehicle datasets. However, 1Probe has to grow 5000 trees to be able to select as many features as J_{χ^2} with only 1000 trees. On the Musk1 and Arrhythmia datasets, which contain many more features, 1Probe needs 100x100 (= 10,000) trees to find out relevant features that lead to predictive performances similar to those of J_{χ^2} with 2,500 trees, as shown in Table 7. For a fixed number of trees, J_{χ^2} appears to highlight more important features than 1Probe. In addition, results presented in Table 8 show that the majority of

the features selected by 1Probe with 100x100 trees are also selected by J_{χ^2} with 2,500 trees and that they are all selected by J_{χ^2} with 10,000 trees.

The mr-Test approach fails to select a sufficient amount of features to perform a good prediction on all datasets but Wine. Its assumption that $\frac{p}{2}$ variables are irrelevant seems clearly violated on the Breast tissue, Glass and Vehicle datasets, as attested by the number of variables selected by the two other methods on the same datasets.

To sum up, significant variables selected by J_{χ^2} or 1Probe also lead to good predictive performances on all datasets. However, 1Probe requires a much larger number of trees to reach those performances. The underlying assumption behind mr-Test is hardly met in practice, which probably explains its poor performances.

5. Conclusion and perspectives

We propose in this work a novel feature importance index from random forests. This index J_{χ^2} produces a feature ranking similar to Breiman's importance index, especially for top ranked features. It has the additional benefit of being a (corrected) p-value from a χ^2 test. Such approach defines a natural threshold to decide which features are estimated statistically important. Unlike a standard t-test, the proposed index is also multivariate as it evaluates the importance of each variable conditioned to the other variables present in the tree ensemble.

Experiments were conducted both on synthetic and real datasets, including low and high-dimensional datasets for binary or multi-class problems. They show that J_{χ^2} allows us to highlight informative features and discard non-informative ones. Computing J_{χ^2} has the same computational complexity as Breiman's index, which is a linear function of the number of trees and the total number of features to be evaluated. J_{χ^2} is also shown to outperform two recently proposed alternatives, known as 1Probe and mr-Test [4].

The selected features with J_{χ^2} offer similar predictive performances when included in a final classifier as compared to a selection by 1Probe. However, the total number of trees required to reach such performances is typically one order of magnitude smaller with J_{χ^2}, especially on high dimensional data. This computational benefit comes from the fact that J_{χ^2} is estimated on the out-of-bag samples which have been defined while growing the forest. In contrast, the existing alternatives include the cost of an additional resampling procedure. The second alternative, mr-Test, is also shown to be

dataset	method	T or NxT	avg(s^{rel})	min(s^{rel})	max(s^{rel})	BCR
Breast tissue	J_{χ^2}	1000	8.00	8.00	8.00	0.85
	1Probe	100x10	6.09	1.00	8.00	0.84
	mrTest	100x10	0.00	0.00	0.00	0.50
	J_{χ^2}	2500	8.01	8.00	9.00	0.85
	1Probe	100x25	7.99	7.00	8.00	0.85
	mrTest	100x25	0.04	0.00	2.00	0.50
	J_{χ^2}	5000	8.21	8.00	9.00	0.85
	1Probe	100x50	8.00	8.00	8.00	0.85
	mrTest	100x50	0.43	0.00	2.00	0.59
Glass	J_{χ^2}	1000	8.01	8.00	9.00	0.74
	1Probe	100x10	7.38	4.00	8.00	0.74
	mrTest	100x10	0.00	0.00	0.00	0.50
	J_{χ^2}	2500	8.49	8.00	9.00	0.74
	1Probe	100x25	8.00	8.00	8.00	0.74
	mrTest	100x25	1.10	0.00	4.00	0.48
	J_{χ^2}	5000	8.94	8.00	9.00	0.74
	1Probe	100x50	8.14	8.00	9.00	0.74
	mrTest	100x50	2.88	0.00	5.00	0.53
Wine	J_{χ^2}	1000	11.49	11.00	13.00	0.98
	1Probe	100x10	6.04	4.00	8.00	0.98
	mrTest	100x10	4.42	0.00	6.00	0.96
	J_{χ^2}	2500	12.76	11.00	13.00	0.98
	1Probe	100x25	9.88	7.00	12.00	0.98
	mrTest	100x25	5.88	5.00	7.00	0.98
	J_{χ^2}	5000	12.99	12.00	13.00	0.98
	1Probe	100x50	11.31	10.00	13.00	0.98
	mrTest	100x50	6.05	6.00	7.00	0.98
Vehicle	J_{χ^2}	1000	18.00	18.00	18.00	0.74
	1Probe	100x10	14.21	0.00	18.00	0.74
	mrTest	100x10	0.34	0.00	1.00	0.47
	J_{χ^2}	2500	18.00	18.00	18.00	0.74
	1Probe	100x25	17.72	16.00	18.00	0.75
	mrTest	100x25	1.00	1.00	1.00	0.40
	J_{χ^2}	5000	18.00	18.00	18.00	0.74
	1Probe	100x50	18.00	18.00	18.00	0.74
	mrTest	100x50	1.24	1.00	4.00	0.44

Table 6: Various statistics obtained over 200-resamplings when keeping only significant features. T is the number of trees used to build the forest. For 1Probe and mr-Test, N indicates the number of external resamplings to compute the null distribution. *avg(s^{rel})* (resp. *max*, *min*) is the average (resp. maximum, minimum) number of significant features. *BCR* is the average predictive performance of a RF built only from features which are estimated significant.

dataset	method	T or NxT	avg(s^{rel})	min(s^{rel})	max(s^{rel})	BCR
Musk1	J_{χ^2}	1000	12.15	7.00	18.00	0.81
	1Probe	100x10	0.01	0.00	1.00	0.50
	mrTest	100x10	0.00	0.00	0.00	0.50
	J_{χ^2}	2500	31.62	21.00	38.00	0.87
	1Probe	100x25	0.92	0.00	3.00	0.61
	mrTest	100x25	0.00	0.00	0.00	0.50
	J_{χ^2}	5000	63.23	52.00	77.00	0.90
	1Probe	100x50	6.51	0.00	15.00	0.76
	mrTest	100x50	0.00	0.00	0.00	0.50
	J_{χ^2}	10000	102.08	89.00	118.00	0.90
	1Probe	100x100	49.01	7.00	121.00	0.88
	mrTest	100x100	0.69	0.00	2.00	0.60
Arrhythmia	J_{χ^2}	1000	6.17	2.00	11.00	0.78
	1Probe	100x10	0.26	0.00	1.00	0.52
	mrTest	100x10	0.00	0.00	0.00	0.50
	J_{χ^2}	2500	15.42	10.00	21.00	0.82
	1Probe	100x25	1.80	1.00	4.00	0.65
	mrTest	100x25	0.00	0.00	0.00	0.50
	J_{χ^2}	5000	24.19	18.00	31.00	0.84
	1Probe	100x50	4.92	2.00	11.00	0.74
	mrTest	100x50	0.17	0.00	2.00	0.51
	J_{χ^2}	10000	36.13	29.00	42.00	0.85
	1Probe	100x100	13.44	6.00	21.00	0.82
	mrTest	100x100	2.18	0.00	5.00	0.66

Table 7: Various statistics obtained over 200-resamplings when keeping only significant features. T is the number of trees used to build the forest. For 1Probe and mr-Test, N indicates the number of external resamplings to compute the null distribution. *avg(s^{rel})* (resp. *max*, *min*) is the average (resp. maximum, minimum) number of significant features. *BCR* is the average predictive performance of a RF built only from features which are estimated significant.

dataset	methods	J_{χ^2} 2500	J_{χ^2} 10,000	1Probe 100x100	mrTest 100x100
Musk1	J_{χ^2}2500	1.00	1.00	0.89	0.02
	J_{χ^2}10,000	0.31	1.00	0.48	0.01
	1Probe100x100	0.62	0.99	1.00	0.02
	mrTest100x100	1.00	1.00	1.00	1.00
Arrhythmia	J_{χ^2}2500	1.00	1.00	0.69	0.15
	J_{χ^2}10,000	0.43	1.00	0.37	0.06
	1Probe100x100	0.82	1.00	1.00	0.18
	mrTest100x100	1.00	1.00	1.00	1.00

Table 8: Average proportion of common variables between various feature sets. The entry at row i and column j in this matrix represents the average proportion, over 200-resamplings, of significant features selected by the index i that also belong to the feature set selected by method j for the same sampling.

too conservative, or even inadequate, and consequently may miss important features which are not estimated to be significant.

We consider here tree ensembles in the specific form of Random Forests. This was originally motivated by the link to be drawn between J_{χ^2} and the original Breiman's index. Yet, J_{χ^2} can in principle be computed from any tree ensemble techniques leaving aside some out-of-bag samples while growing the ensemble. Those include at least bagging of trees, extremely randomized trees [15] and c-Forests [16]. The impact of considering J_{χ^2} jointly with these techniques is considered as future work.

On high dimensional data, increasing the number of trees (typically up to 10,000) is shown to be beneficial to correctly discover the informative variables and to discard irrelevant ones. This result is consistent with the study of feature selection stability from RF proposed in [13]. In general, enlarging the ensemble size naturally leads to increase the number of features that are deemed statistically significant. Beyond significance itself, it would also be interesting to study the effect size evaluated by such a statistical procedure. The Cramer's V measure [17] looks interesting in this regard.

Finally, this work show that measuring the distribution shift of class votes before and after permuting a feature in a tree ensemble conveys some useful information. The specific test to characterize such distribution shift is a bit less central. The J_{χ^2} test is convenient and appears to be effective in practice, yet one could certainly design other procedures.

For instance, a Kolmogorov-Smirnov (KS) test offers a particular non-parametric alternative. The test statistic here relies on the distribution of the out-of-bag classification accuracies (or balanced classification rates averaging specificity and sensitivity) across the various trees in the ensemble. The effect on such distributions after permuting a specific variable is assessed. Our preliminary results along those lines show that the KS procedure offers very similar results to those of J_{χ^2}, but at a higher computational cost.

In the same spirit, one could easily design further variants to focus on some specific function of the class confusion matrix and, for instance, to promote the selection of features that play a more critical role in the sensitivity of the classifier while putting less emphasis on the specificity as well.

Acknowledgements

We thank the anonymous reviewers for their fruitful comments. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the Fonds de la Recherche Scientifique de Belgique (FRS-FNRS).

References

[1] L. Breiman, Random Forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.
URL http://dx.doi.org/10.1023/A:1010933404324

[2] Y. Benjamini, Y. Hochberg, Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, Journal of the Royal Statistical Society. Series B (Methodological) 57 (1) (1995) 289–300. doi:10.2307/2346101.
URL http://dx.doi.org/10.2307/2346101

[3] S. Stigler, Fisher and the 5% level, CHANCE 21 (4) (2008) 12–12. doi:10.1007/s00144-008-0033-3.
URL http://dx.doi.org/10.1007/s00144-008-0033-3

[4] V. A. A. Huynh-Thu, Y. Saeys, L. Wehenkel, P. Geurts, Statistical interpretation of machine learning-based feature importance scores for biomarker discovery., Bioinformatics (Oxford, England) 28 (13) (2012)

1766–1774. doi:10.1093/bioinformatics/bts238.
URL http://dx.doi.org/10.1093/bioinformatics/bts238

[5] C. Zhang, X. Lu, X. Zhang, Significance of gene ranking for classification of microarray samples, IEEE/ACM Trans. Comput. Biol. Bioinformatics 3 (3) (2006) 312–320. doi:10.1109/TCBB.2006.42.
URL http://dx.doi.org/10.1109/TCBB.2006.42

[6] I. Guyon, A. R. S. A. Alamdari, G. Dror, J. Buhmann, Performance prediction challenge, in: International Joint Conference on Neural Networks, Vancouver, BC, Canada, 2006, pp. 2958 – 2965.

[7] L. I. Kuncheva, A stability index for feature selection, in: AIAP'07: Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference, ACTA Press, Anaheim, CA, USA, 2007, pp. 390–395.

[8] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, T. R. Golub, Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning., Nat Med 8 (1) (2002) 68–74. doi:10.1038/nm0102-68.
URL http://dx.doi.org/10.1038/nm0102-68

[9] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, L. M. Staudt, Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling., Nature 403 (6769) (2000) 503–511. doi:10.1038/35000501.
URL http://dx.doi.org/10.1038/35000501

[10] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, science 286 (5439) (1999) 531–537.

[11] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer cell 1 (2) (2002) 203–209.

[12] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL http://archive.ics.uci.edu/ml

[13] J. Paul, M. Verleysen, P. Dupont, The stability of feature selection and class prediction from ensemble tree classifiers, in: ESANN 2012, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2012, pp. 263–268.

[14] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
URL http://dl.acm.org/citation.cfm?id=1248547.1248548

[15] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63 (2006) 3–42, 10.1007/s10994-006-6226-1.
URL http://dx.doi.org/10.1007/s10994-006-6226-1

[16] C. Strobl, A. Boulesteix, A. Zeileis, T. Hothorn, Bias in random forest variable importance measures: Illustrations, sources and a solution, BMC Bioinformatics 8 (1) (2007) 25.

[17] H. Cramér, Mathematical Methods of Statistics, Princeton University Press, 1946.