Influence of Hyperparameters on Random Forest Accuracy

Simon Bernard, Laurent Heutte, and Sébastien Adam

Université de Rouen, LITIS EA 4108
BP 12 - 76801 Saint-Etienne du Rouvray, France
{simon.bernard,laurent.heutte,sebastien.adam}@univ-rouen.fr

Abstract. In this paper we present our work on the Random Forest (RF) family of classification methods. Our goal is to go one step further in the understanding of RF mechanisms by studying the parametrization of the reference algorithm Forest-RI. In this algorithm, a randomization principle is used during the tree induction process, that randomly selects K features at each node, among which the best split is chosen. The strength of randomization in the tree induction is thus led by the hyperparameter K which plays an important role for building accurate RF classifiers. We have decided to focus our experimental study on this hyperparameter and on its influence on classification accuracy. For that purpose, we have evaluated the Forest-RI algorithm on several machine learning problems and with different settings of K in order to understand the way it acts on RF performance. We show that default values of K traditionally used in the literature are globally near-optimal, except for some cases for which they are all significatively sub-optimal. Thus additional experiments have been led on those datasets, that highlight the crucial role played by feature relevancy in finding the optimal setting of K.

Keywords: Supervised Learning, Ensemble Method, Random Forests, Decision Trees.

1 Introduction

Random Forest is a family of classifier ensemble methods that use randomization to produce a diverse pool of individual classifiers, as for Bagging [1] or Random Subspaces methods [2]. It can be defined as a generic principle of classifier ensemble that uses L tree-structured base classifiers $\{h(x, \Theta_k),\ k = 1, ...L\}$ where $\{\Theta_k\}$ is a family of independent identically distributed random vectors, and x is an input data. The particularity of this kind of ensemble is that each decision tree is built from a random vector of parameters. A Random Forest can be built for example by randomly sampling a feature subset for each decision tree (as in Random Subspaces), and/or by randomly sampling a training data subset for each decision tree (as in Bagging). Since they have been introduced in 2001, RFs have been studied in many ways, theoretically as well as experimentally [3,4,5,6,7,8]. In most of those works, it has been shown that RFs are particularly competitive with one of the most efficient learning principles, *i.e.* boosting [4,8]. However, the mechanisms that explain the good performance of RFs are not

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 171–180, 2009.

clearly identified and one has to admit that it is still a complex task for the practitioner to take full benefits of the potential of those methods. For example considering the reference RF method called Forest-RI, introduced by Breiman in [4] (see section 2), an important hyperparameter has been identified : the number K of features randomly selected at each node during the tree induction process. Yet, in those research works that have experimented this method, the value of K is arbitrarily or empirically set, and sometimes without any theoretical nor experimental justification.

In this paper we propose to go one step further in the understanding of RF mechanisms, by studying the parametrization of the reference algorithm Forest-RI. We propose to study the influence of the hyperparameter K on the classification accuracy, in order to empirically distinguish parametrization rules or tendencies, according to some characteristics of the classification problem. The final goal is to give elements to the practitioner in order to help him building RFs that accurately suit to the specificities of a classification problem. For that purpose we have experimented on several machine learning datasets the algorithm Forest-RI with different settings of K in order to study their influence on accuracy. We show that default values of K traditionally used in the literature are globaly near-optimal, except for some cases for which they are all significatively sub-optimal. We have then studied the relation between the nature of the feature space and RF performance, by studying feature relevancy. We highlight the crucial role played by feature relevancy in finding the optimal setting of K.

The paper is thus organized as follows: we recall in section 2 the Forest-RI principles and related works on its experimentation; in section 3, we describe our experimental protocol, the datasets used, and the results obtained with different settings of the hyperparameter K. We finally draw some conclusions and future works in the last section.

2 The Forest-RI Algorithm

One can see Random Forests as a family of methods, made of different decision tree ensemble induction algorithms, such as the Breiman Forest-RI method often cited as the reference algorithm in the literature [4]. In this algorithm the Bagging principle is used with another randomization technique called Random Feature Selection. The training step consists in building an ensemble of decision trees, each one trained from a bootstrap sample of the original training set — *i.e.* applying the Bagging principle — and with a decision tree induction method called Random Tree. This induction algorithm, usually based on the CART algorithm [9], modifies the splitting procedure for each node, in such a way that the selection of the feature used for the splitting criterion is partially randomized. That is to say, for each node, a feature subset is randomly drawn, from which the best splitting criterion is then selected as in traditionnal tree induction algorithms. To sum up, in the Forest-RI method, a decision tree is grown by using the following process :

- Let N be the size of the original training set. N instances are randomly drawn with replacement, to form the bootstrap sample, which is then used to build the tree.
- Let M be the dimensionality of the original feature space, and K a preliminary fixed hyperparameter so that $K \in [1, M]$. For each node of the tree, a subset of

K features is randomly drawn without replacement, among which the best split is then selected.

– The tree is thus built to reach its maximum size. No pruning is performed.

In this process the tree induction is directed by a single hyperparameter, *i.e.* the number K of randomly selected features. This number allows to introduce more or less randomization in the induction. Whereas this hyperparameter seems to be critical to induce accurate RF [3], no research work has been specifically devoted to the study of its setting and its real influence on performance, and only a few have empirically dealt with this issue.

In [6] for example, Geurts et al. have proposed a new method of RF induction, called Extras-Trees for Extremely Randomized Tree Ensemble, that modifies the Forest-RI algorithm to accentuate the randomization. Here the Random Feature Selection is still used but modified so that the best splitting criterion selection is one step further randomized. The authors have designed their experimental protocol to study the influence of K on performance. Even if this method is partly different from the Forest-RI algorithm, this work allows to draw some intuitions on the Random Forest behavior according to K. It highlights for example that its default setting $K = \sqrt{M}$, where M stands for the size of the original feature set, is most of times closed to the optimal setting, at least for the Extras-Trees method and on several representative datasets.

When introducing RF formalism, Breiman studied performance according to K [4]. In these experiments, a large number of RF has been grown on three datasets, for which the test set error rate has been monitored. Actually only one of those three experiments was really concerned by the Forest-RI algorithm, since the two others have been run with an induction algorithm that uses feature combinations as splitting criterions, instead of single features. Hence, even if some tendencies can be intuitively guessed, those experiments do not allow to conclude on RF behavior according to the setting of K. We also noticed that in his Forest-RI experiments, Breiman decided to use two values of $K : 1$ and $log_2 M + 1$. While the first value is intuitively interesting since it corresponds to a decision tree induction that selects in a fully random manner the splitting criterion among features for each node, the second one seems to be more arbitrary or at least is not justified.

In [3], a serie of tests with Forest-RI has been led on the well-known MNIST handwritten digit recognition dataset [10]. An interval of values of K has been found for which best accuracies have been reached; this interval does not contain neither $K = 1$ nor $K = M$ but contains the two values $K = \sqrt{M}$ and $K = log_2(M) + 1$. However we think that their primary conclusions need to be confirmed with a more rigourous experimental protocol and with several different machine learning datasets.

Finally, implementation and experimentation of the Forest-RI algorithm require to fix the value of the hyperparameter K but there actually does not exist any theoretical rule that can be used to fix it. As mentioned previously, only arbitrary default values are proposed in the literature and nothing guarantees that these values are close to the optimal setting. Thus one of the goal of the work presented in this paper is to bring elements of RF mechanism understanding by focusing on the hyperparameter K and on the way it acts on RF accuracy. For that purpose we have led a set of experiments

that are described in the following section. Notice that in the rest of this paper, the term Random Forest (RF) will always stand for a forest induced with the Forest-RI algorithm.

3 Investigating the Influence of Hyperparameter K on Accuracy

The hyperparameter K denotes the number of features randomly selected at each node during the tree induction process. It must be set with an integer in the interval $[1..M]$, where M stands for the dimensionality of the feature space. This number thus controls the strength of the randomization in the feature selection process, in such a way that the smaller the value of K, the stronger the randomization. In the case where $K = 1$ for example, each split (*i.e.* the feature used as splitting criterion) of the tree structure is randomly selected among all the available features. On the contrary, where $K = M$, no randomization is introduced in the split selection and each tree is thus grown following a traditional tree induction process. In this particular case, randomization is thus introduced only through the bagging principle. The main idea of our experiments is to study RF accuracy according to hyperparameter K, on several kinds of machine learning problems. We first describe in the following subsection the datasets used. We then detail our experimental protocol and results in the next two subsections.

3.1 Datasets

The description of the 12 datasets that have been used for these experiments is summurized in Table 1. 9 of these datasets have been selected from the UCI repository [11], because they concern different machine learning issues in terms of number of classes, number of features and number of samples. Three additionnal datasets on different handwritten digit recognition problems have been used: (i) the well-known MNIST database [10] with a 85 multiresolution density feature set $(1+2\times2+4\times4+8\times8)$ built from greyscale mean values as explained in [3]; (ii) Digits and DigReject both described in [12], on which a 330-feature set has been extracted, made from three state-of-the-art kinds of descriptors, *i.e.* a 117-statistical/structural feature set [13], a 128-feature set extracted from the chaincode (contour-based) [14], and the same 85-feature set as for MNIST.

Table 1. Dataset description

Dataset	# Samples	# Features	# Classes	Dataset	# Samples	# Features	# Classes
Digits	38142	330	10	Mfeat-karhunen	2000	64	10
DigReject	14733	330	2	Mfeat-zernike	2000	47	10
Letter	20000	16	26	MNIST	60000	84	10
Madelon	2600	500	2	Musk	6597	166	2
Mfeat-factors	2000	216	10	Pendigits	10992	16	10
Mfeat-fourier	2000	76	10	Segment	2310	19	7

3.2 Experimental Protocol

Our experiments aim at studying the evolution of RF accuracy according to different values of K. For all the RF induced in our experiments, the number of trees has been set to 100. This choice is based on an experimental work presented in [3], in which it has been shown that it is a reasonable value to grow an accurate RF. Moreover our goal is not here to reach intrinsic optimal performance. First, each dataset has been randomly split into training and testing subsets, as explained in the previous section. This splitting procedure has been repeated 50 times, so that 50 different training sets and testing sets are thus available, each respectively containing two thirds and one third of the original dataset. We denote by $T_i = (Tr_i, Ts_i)$ such a split, with $i \in [1..50]$ and where Tr_i and Ts_i stand respectively for the training set and the testing set. Then, for each T_i, the Forest-RI algorithm has been run for each value of K in $[1..M]$, where M stands for the total number of features. However, mainly for computational reasons, the definition domain of K has been sampled for some of the datasets for which the size of the feature space is too large, so that values of K have been picked at regular intervals between 1 and M. In the rest of this paper we denote by M' the number of values of K that have been tested for each dataset. Table 2 summarizes the number of runs in these experiments, according to the values of K and the number of splits. Algorithm 1 summarizes the whole experimental protocol applied to each dataset. This procedure outputs a table of $50 \times M'$ error rates according to different values of K, and for each dataset. Those results are presented and discussed in the next subsection.

Table 2. Numbers of runs for evaluating RF performance according to K

Dataset	M	M' = # values of K tested for each T_i	total # of runs
Digits	330	$\frac{M}{10} + 1 = 34$	$34 \times 50 = 1700$
DigReject	330	$\frac{M}{10} + 1 = 34$	$34 \times 50 = 1700$
Letter	16	16	$16 \times 50 = 800$
Madelon	500	$\frac{M}{10} + 1 = 51$	$51 \times 50 = 2550$
Mfeat-factors	216	$\frac{M}{3} + 1 = 73$	$73 \times 50 = 3650$
Mfeat-fourier	76	76	$76 \times 50 = 3800$
Mfeat-karhunen	64	64	$64 \times 50 = 3200$
Mfeat-zernike	47	47	$47 \times 50 = 2350$
Mnist	84	$\frac{M}{2} + 1 = 43$	$43 \times 50 = 2150$
Musk	166	$\frac{M}{3} + 1 = 56$	$56 \times 50 = 2800$
Pendigits	16	16	$16 \times 50 = 800$
Segment	19	19	$19 \times 50 = 950$
Total		586	29100

3.3 Results

Table 3 presents the results obtained with the experimental protocol detailed in Algorithm 1. With this protocol, a table of $50 \times M'$ error rates with respect to K is first obtained for each dataset. These series of error rates have been averaged so that to each dataset correspond M' accuracies, *i.e.* one mean value for every K, and the standard

Algorithm 1. Experimental Protocol 1

INPUT: N: number of samples in the original dataset.
INPUT: M: number of features in the original dataset.

OUTPUT: $\epsilon[50][M']$: 2D table used for storing each error rate obtained with Forest-RI.
for $i \in [1..50]$ **do**
Randomly draw without replacement $\frac{2}{3} \times N$ samples from the original dataset to form a training subset Tr_i. The remaining samples form the testing subset Ts_i, the couple (Tr_i, Ts_i) is denoted T_i.
for each k in $[1..M]$ **do**
$H(k) \leftarrow$ Grow a Random Forest with the Forest-RI algorithm, with $L = 100$ and $K = k$, on the training set Tr_i.
$\epsilon(i, k) \leftarrow$ Test the resulting RF on the testing set Ts_i.
end for
end for

deviation values have also been computed. Table 3 details some of those results according to five particular values of K. The first of those values, in the second column, corresponds to the "optimal" value of K, noted K^*, that is to say the value of K, among all the tested values, for which the maximum average accuracy has been reached. The second and third values correspond to $K = \sqrt{M}$ and $K = log_2 M + 1$ which are often used as default settings in the literature (see section 2). The two last columns present error rates for $K = 1$ and $K = M$. The number in brackets for columns 2, 3 and 4, represents the value of K that is either obtained from the experiments (K^* in column 2) or fixed for the experiments ($\sqrt{M}$ in column 3 or $log_2(M) + 1$ in column 4).

A first observation made from Table 3 is that the four particular values $K = \sqrt{M}$, $K = log_2(M) + 1$, $K = 1$ and $K = M$ rarely exactly correspond to the best parametrization of K. K^* is $\sqrt{M}$ for only 2 of the 12 datasets (Mfeat-zernike and

Table 3. Mean error rates for the different algorithms

Dataset	K^*	$K_{\sqrt{M}}$	$K_{log_2(M)+1}$	K_1	K_M
Digits	2.18 ± 0.12 (11)	2.20 ± 0.13 (18)	2.19 ± 0.12 (9)	2.61 ± 0.13	3.25 ± 0.19
DigReject	7.15 ± 0.34 (181)	7.70 ± 0.34 (18)	7.80 ± 0.35 (9)	9.02 ± 0.32	7.27 ± 0.30
Letter	4.16 ± 0.28 (3)	4.23 ± 0.23 (4)	4.30 ± 0.26 (5)	5.21 ± 0.27	7.33 ± 0.47
Madelon	17.60 ± 1.60 (261)	30.48 ± 1.94 (22)	34.54 ± 1.26 (10)	45.94 ± 1.57	18.60 ± 1.91
Mfeat-fac	3.56 ± 0.71 (10)	3.57 ± 0.58 (15)	3.58 ± 0.73 (9)	4.27 ± 0.72	4.61 ± 0.76
Mfeat-fou	16.81 ± 1 (19)	17.11 ± 1.05 (9)	17.25 ± 1.02 (7)	22.18 ± 1.19	18.66 ± 1.33
Mfeat-kar	4.30 ± 0.68 (6)	4.33 ± 0.69 (7)	4.30 ± 0.68 (6)	7.14 ± 0.83	8.38 ± 1.11
Mfeat-zer	22.26 ± 1.06 (7)	22.26 ± 1.06 (7)	22.56 ± 1.02 (5)	23.86 ± 0.90	24.87 ± 1.33
MNIST	5.06 ± 0.14 (24)	5.17 ± 0.14 (10)	5.32 ± 0.17 (8)	6.54 ± 0.17	6.54 ± 0.28
Musk	2.34 ± 0.34 (88)	2.40 ± 0.29 (13)	2.50 ± 0.26 (8)	4.03 ± 0.32	2.48 ± 0.34
Pendigits	0.97 ± 0.17 (4)	0.97 ± 0.17 (4)	1.01 ± 0.17 (5)	1.15 ± 0.18	1.50 ± 0.23
Segment	2.36 ± 0.53 (5)	2.44 ± 0.44 (4)	2.36 ± 0.53 (5)	3.23 ± 0.50	2.71 ± 0.56

Pendigits), $log_2(M)+1$ for only 2 of them (Mfeat-karhunen and Segment), and is never equal to 1 nor to M. Those settings are even sometimes quite far from the K^* value, as for the Madelon, DigRejects, MNIST and Musk datasets. One could conclude from this that it is not advised to systematically use one of those default settings of K and that it is necessary to further investigate a way to supply a better rule of parametrization. However, by examining standard deviation values and differences between results obtained for two settings of K closed from each other, we were wondering if this conclusion was correct. We can see from table 3 that they are most of the time of the same order. If we thus consider that values of K closed from each other produce classifiers statiscally equivalent in terms of accuracy, $K = \sqrt{M}$ can be considered as a good setting for K — at least close to the optimal setting — for 8 of the 12 datasets. Of course this statement needs to be experimentally investigated and proved, by performing a statistical test of significance such as the McNemar test for example. Nevertheless we think that it is reasonnable to deduce from our results that $K = \sqrt{M}$ is a good compromise for the parametrization of K.

Figure 1 presents our results as curves of averaged error rates with respect to values of K. On this figure one can first observe that all the curves exhibit the same global variation, that is to say a decreasing followed by an increasing when K increases. This confirms the primary conclusions drawn in [3], in which it has been found that the extrema values for K, *i.e.* $K = 1$ and $K = M$, are not advised to be used for building an accurate RF with Forest-RI. However, this common behavior strongly differs from a dataset to another in a more detailed analysis of the curves. We can distinguish three trends of variation in these 12 diagrams: for 5 of them (Mfeat-factors, Mfeat-fourier, Mfeat-karhunen, Mfeat-zernike and Digits) the minimum error rate is reached for a small value of K (marked with circles in the figure) and the increase of the curves from this point till $K = M$ is monotonical and almost linear; for 4 of them (Letter, Mnist, Pendigits and Segment) the trend is almost the same but with a more parabolical shape; for 3 of them (DigReject, Madelon and Musk) this increase of the curves is quasi null, and the minimum error rate is reached for a larger value of K (*i.e.* greater than $\frac{M}{2}$). These different behaviours are quite difficult to explain only with some characteristics of the datasets such as the number of classes or the number of features. Geurts et al. in [6] suggest that the nature of the features could explain some particularity of their Extra-Trees behavior for some particular cases. They conjecture that the more features that are irrelevant, the larger the value of K^*, since *a higher value of K would lead to a better chance of filtering out the irrelevant variables*. This statement has led us to focus on the relevancy of features for explaining accuracy variation according to values of K. For that purpose we have decided to evaluate the relevancy of each feature by measuring the information gain with respect to the class. In general terms, the expected information gain is the change in information entropy from a prior state to a state that takes some information [15]. It is often used in decision tree induction as criterion for node split selection. For our experiments, information gain has been measured for all the features on each Tr_i of each dataset. In that way, $50 \times M$ values of information gain have been computed for each dataset. Figure 2 synthesizes those results as curves of cumulative number of features with respect to information gain values, so that each dot of the curves indicates the number of features for which the information gain is smaller

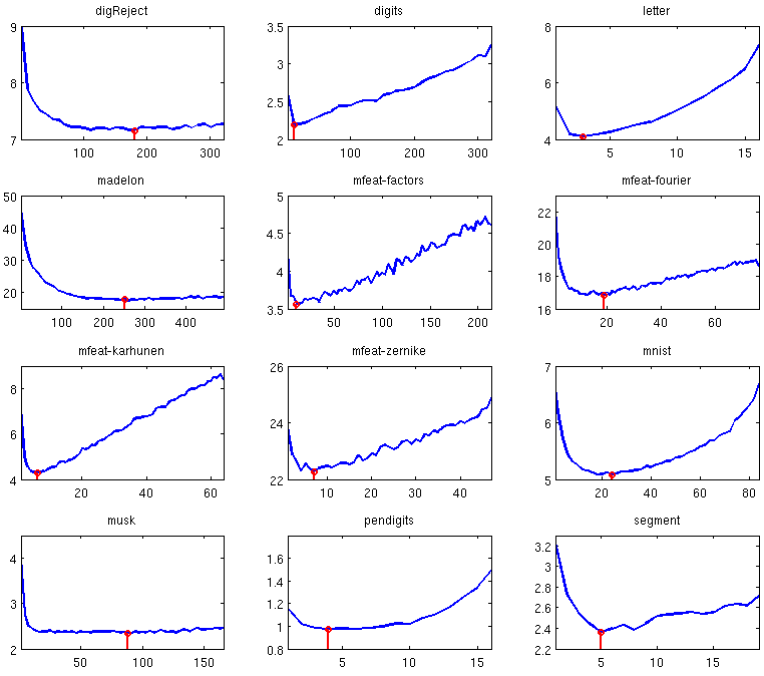

Fig. 1. Mean error rates with respect to values of K. The minimum error rates are marked on each diagram.

than or equal to the corresponding value on the x-axis. This representation allows to simultaneously observe the relevancy of all the features and gives an idea of how many of them are irrelevant. One can observe on this figure that values of information gain are globally greater (typically higher than 0.1) for datasets for which values of K^* are small regarding to M, as this is the case for exemple for Digits, Mfeat-Factors and Mfeat-Karhunen for which K^* is lower than $\frac{M}{10}$. On the contrary, for the three datasets for which K^* is higher than $\frac{M}{2}$ (DigReject, Madelon and Musk), the information gain values are always lower than about 0.1. This seems to prove that relevancy of features strongly explains the accuracy variation according to values of K.

The choice of K actually leans on a compromise between two needs : (i) to force, via randomness, the tree induction process to diversify choices of splitting criteria in order to induct trees different from each other (ii) to choose relevant features for splitting criteria in order to induct trees performant enough. A too strong randomization of the split selection produces trees that globally do not suit the problem enough, while not randomizing "enough" creates trees that tend to overfit the training data, and thus make them be similar from each other in terms of predictions. K acts thus as a trade-off for balancing performance and diversity of trees in the ensemble. However we believe that the impact on performance of the "amount" of randomization used in the split selection process, strongly depends on the global relevancy of features. If there are too few relevant features, the randomization will rapidly make the tree accuracy decrease, and

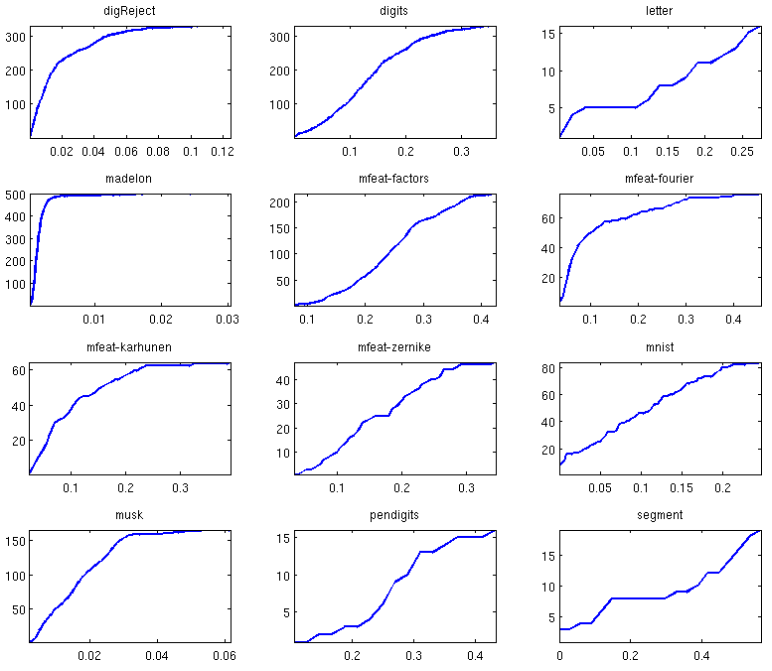

Fig. 2. Cumulative number of features with respect to information gain values

thus rapidly deteriorate the "individual performance of the trees versus the ensemble diversity" trade-off. On the contrary a large amount of strongly relevant features facilitates the overfitting of trees and weakens the randomization effects in split selection. We believe that this is the reason why the three datasets for which values of information gain are small (DigRejects, Madelon and Musk), present error rate curves that do not significantly raise for increasing values of K. Consequently we think that the relevancy of features is an important property that should be taken into account for determining a parametrization rule for the Forest-RI algorithm.

4 Conclusions

Investigations on RF parametrization have been presented in this paper, that have focused on the number K of features randomly selected at each node during the tree induction. This hyperparameter allows to control the strengh of the randomization in the split selection, in such a way that the smaller the value of K, the stronger the randomization. In this work several experiments have been led with the Forest-RI algorithm on different machine learning datasets and with different settings of K, in order to study its influence on RF performance. We have firstly shown that default settings traditionally used in the literature do not allow to produce the best possible RF in terms of accuracy, in a majority of cases. However our results illustrate that one of them, *i.e.* $K = \sqrt{M}$

with M being the dimensionality of the feature space, is a reasonnable setting to induct near-optimal RF. This statement should obviously be experimentally confirmed through the use of statistical test of significance such as the McNemar test for determining whether or not significant improvement can be made with an optimal setting of K in comparison with $K = \sqrt{M}$. In a second part of this experimental work we have focused on the relevancy of features to determine whether or not it could explain the accuracy variation according to K. We have shown that this property is crucial for finding the best setting of K since it can strongly modify the randomization effect of the random split selection procedure. As a consequence we think that the relevancy of features is an important property that should be taken into account for determining a parametrization rule for the Forest-RI algorithm. Our future works will focus on this open issue.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
3. Bernard, S., Heutte, L., Adam, S.: Using random forests for handwritten digit recognition. In: International Conference on Document Analysis and Recognition, pp. 1043–1047 (2007)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Breiman, L.: Consistency of random forests and other averaging classifiers. Technical Report (2004)
6. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 36(1), 3–42 (2006)
7. Latinne, P., Debeir, O., Decaestecker, C.: Limiting the number of trees in random forests. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 178–187. Springer, Heidelberg (2001)
8. Rodriguez, J., Kuncheva, L., Alonso, C.: Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619–1630 (2006)
9. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Chapman and Hall/Wadsworth, Inc., New York (1984)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
11. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
12. Chatelain, C., Heutte, L., Paquet, T.: A two-stage outlier rejection strategy for numerical field extraction in handwritten documents. In: International Conference on Pattern Recognition, Honk Kong, China, vol. 3, pp. 224–227 (2006)
13. Heutte, L., Paquet, T., Moreau, J., Lecourtier, Y., Olivier, C.: A structural/statistical feature based vector for handwritten character recognition. Pattern Recognition Letters 19(7), 629–641 (1998)
14. Kimura, F., Tsuruoka, S., Miyake, Y., Shridhar, M.: A lexicon directed algorithm for recognition of unconstrained handwritten words. IEICE Transaction on Information and System E77-D(7), 785–793 (1994)
15. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)