Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 258054, 13 pages
doi:10.1155/2012/258054

Research Article

Applying Randomness Effectively Based on Random Forests for Classification Task of Datasets of Insufficient Information

Hyontai Sug

Division of Computer and Information Engineering, Dongseo University, Busan 617-716, Republic of Korea

Correspondence should be addressed to Hyontai Sug, hyontai@yahoo.com

Received 20 July 2012; Revised 8 October 2012; Accepted 8 October 2012

Academic Editor: Hak-Keung Lam

Random forests are known to be good for data mining of classification tasks, because random forests are robust for datasets having insufficient information possibly with some errors. But applying random forests blindly may not produce good results, and a dataset in the domain of rotogravure printing is one of such datasets. Hence, in this paper, some best classification accuracy based on clever application of random forests to predict the occurrence of cylinder bands in rotogravure printing is investigated. Since random forests could generate good results with an appropriate combination of parameters like the number of randomly selected attributes for each split and the number of trees in the forests, an effective data mining procedure considering the property of the target dataset by way of trial random forests is investigated. The effectiveness of the suggested procedure is shown by experiments with very good results.

1. Introduction

Because rotogravure printing is used to print in a large volume, it is important to prevent process delays for higher productivity. But, when rotogravure printing is being performed, sometimes a series of bands appear in the cylinder of printing machine so that it ruins the printouts. When this happens, a pressman should do appropriate action to remove the bands from the cylinder, resulting in process delays up to even several hours. In order to reduce the delays, preventive maintenance activity is more desirable, if we can predict possible occurrence of the bands accurately in advance [1]. So many researchers tried to increase the predictive accuracy for the task [2–5], and decision tree-based methods and neurocomputing-based methods have been used mostly for the task. It is known that a weak point of decision trees is relatively poor accuracy compared to other data mining methods like neural networks, because decision trees fragment datasets and prefer majority classes, even

if the size of available datasets is small. In order to overcome the problem, a large number of decision trees could be generated for a single dataset based on some random sampling method and could be used for classification. Random forests [6, 7] are a representative data mining method that uses many trees for that purpose. Random forests are known to be robust for real world datasets that may not have enough information as well as may have missing and erroneous data. Because a related dataset called "cylinder bands" is a real world dataset that contains such properties in the domain of rotogravure printing, and random forests have different performance depending on the values of parameters of the algorithm with respect to the property of given dataset, therefore, in this paper we want to find some best predictive accuracy with random forests to predict the cylinder bands by examining the property of the dataset by way of trial random forests and effective search.

Several research results have been published to find better classification models for the so-called "cylinder bands" dataset, after the first paper [2] related to the task was published. They generated rules based on C4.5 decision tree algorithm [8] to improve the heuristics that can predict possible occurrence of bands or a series of grooves in the cylinder during printing. But, because the rules are based on a single decision tree, the prediction accuracy is somewhat limited. After the first paper, other researchers have tried also to find better knowledge models with respect to accuracy.

As an effort to find the knowledge models of better performance, fuzzy lattice neurocomputing (FLN) models based on competitive clustering and supervised clustering were suggested [3]. Later, the researchers of FLN models found that the data space can be divided into subspaces based on class values of each data instance. So depending on fitness of each data instance to data space, five fit algorithms were suggested [4]; FLN tightest fit, FLN ordered tightest fit, FLN first fit, FLN selective fit, and FLN max tightest fit. A fit is called tightest, if the lattice-join of any data instance in the same class causes a contradiction. The FLN tightest fit was the first one among the five FLN models, and the accuracy of FLN ordered tightest fit is the best accuracy among the fuzzy lattice neurocomputing models. FLN models have the time complexity of $O(n^3)$ to train, which means that it is a polynomial time algorithm, so it will take some long computing time, if the size of input data is large [9].

Some other researchers tried to find better knowledge models of performance based on randomness in attribute selection and training datasets. Random subspace method [6] tries to select the subsets of attributes randomly and applies aggregating to find better classification models. SubBag method [5] tries BAGGING [10] and random subspace method together. BAGGING stands for Bootstrap AGGregatING. So in BAGGING several equally sized training sets are made using sampling with replacement, and trained knowledge models vote for classification or prediction. It was combined with decision tree algorithm based on C4.5 and rule generator named JRip which is based on RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [11]. According to experiments with a variety of datasets RIPPER algorithm gave better accuracy and could treat larger datasets than the rule generation method of C4.5, and the algorithm was known to be robust for noisy datasets also. SubBag and BAGGING with JRip showed competitive results with FLN tightest fit.

More recently, random forests were tried with some fixed parameter values. Because each decision tree in random forests is independent, parallel training of each decision tree in the random forests was tried with a concurrent programming language called erLang [12]. Boström generated the random forests based on decision tree algorithm that uses information gain [13]. The random forests of 100 trees, 1000 trees, 10,000 trees, and 100,000 trees were generated and showed comparable results to FLN tightest fit and SubBag method. Table 1 summarizes all the previous works.

Table 1: Comparison of accuracy for "cylinder bands" data set in different data mining methods.

Number	Method	Accuracy (%)	Year
1	FLN tightest fit [3]	78.33	2000
2	FLN ordered tightest fit [4]	85.56	2003
3	Concurrent random forests [12]	80.19	2011
4	SubBag-JRip [5]	78.15	2007
5	BAGGING-JRip [5]	79.26	2007

Table 2: The accuracy of trial random forests (RF) for each class with different parameters.

	($T = 1$, $R = 39$)	($T = 100$, $R = 6$)	($T = 100$, $R = 1$)
Class accuracy			
"Band"	18.4%	70.2%	74.6%
"No band"	92.9%	91.7%	91.7%
Accuracy of RF	61.4815%	**82.5926**%	**84.4444**%

2. The Method

2.1. Random Forests

Random forests suggested by Breiman [7] are based on BAGGING, use many decision trees with some random selection of attributes to split each node in the tree, and do no pruning. In other words, random forests use bootstrap method [14] in sampling to generate a training set, and the training set is used to build a tree, and since bootstrap method uses sampling with replacement, each training set can have some duplicate instances and could compensate the insufficiency of data to train somewhat.

After sampling some conventional decision tree generation algorithms like C4.5 or CART can be applied, but without pruning. When random selection of attributes to split each node is applied, the number of candidate attributes for split is limited by some predefined number, say R. R may be given by user, or default value can be used. Default R value is the first integer less than $\log_2 A + 1$ [7, 15], and the half and double of the number are also recommended for further search [16]. So, depending on which number is used, the degree of randomness in tree generation is affected.

The other factor that affects the accuracy of random forests is the number of decision trees, say T in the forests. Because the trees in the forests are generated samples with random sampling with replacement, appropriate T value could compensate the insufficiency of data for training. According to Breiman tens to hundreds of decision trees are enough as T value, because thousands of trees may not give better performance than the smaller number of trees in the random forests. Moreover, we may have different accuracy of random forests depending on how many trees are in the forests, but small difference in the number of trees may not give different accuracy.

2.2. Optimization Procedure

Decision tree algorithms have the tendency of neglecting minor classes to achieve overall best accuracy, so smaller R value in random forests can alleviate the tendency. Minor classes are classes that have less number of instances possibly having more conflicting class values.

Table 3: The accuracy of random forests for the data set "cylinder bands."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (T)
12	77.5926	78.5185	79.4444	575
6	82.037	82.7778	83.3333	475, 500
3	83.7037	84.0741	84.4444	200, 500, 675
2	83.7037	84.0741	84.6296	1000
1	84.4444	85.0	**85.7407**	325

```
Procedure:
Begin
  Check if the grid search could be effective by generating trial random forests;
  /* |A|: the number of conditional attributes */
  R := the double of the first integer less than log2|A| + 1;
  I = 100; F = 1000; D = 25;
  Do
      For t = I to F by increasing D
        /* Generate random forests of t trees in which R attributes are picked
        randomly to split each node */
        Generate Random_forests (R, t);
      End For;
      R := the first integer larger than R/2;
    Until R = 1;
End.
```

Procedure 1

Depending on the composition of given datasets, this discrimination of minor or major can be varied. Note that setting R = (the number of attributes) makes the random forests conventional decision trees without pruning. Moreover, because preparing training sets with random sampling with replacement or bootstrapping has the effect of oversampling, it could duplicate training instances that can result in better accuracy. Therefore, appropriate combination of R and T value can generate better results. In other words, because appropriate T value could supplement training instances for better accuracy and appropriate or smaller R value could mitigate the decision tree's property of neglecting minor classes, we could find best random forests.

We often use the default R value with some fixed T value, because we believe that the values would be good for their datasets, since the values were often recommended as other researchers did [12, 17, 18]. But, we understand that the parameters should be set well to reflect the fact that we may not have enough instances to train.

This is the reason why we generate trial random forests in three different ways: {R = the number of attributes, $T = 1$}, {R = the default number of attributes to pick randomly, $T = 100$}, {$R = 1$, $T = 100$}. Note that with parameter {R = the number of attributes, $T = 1$} the splitting criteria of decision tree algorithm will be used 100% as conventional decision tree algorithms. By setting R value smaller we can mitigate the splitting criteria so that decision tree's property of preferring major classes can be mitigated.

As for our target dataset, the total number of instances in our target dataset called "cylinder bands" is 540, and it has two classes, "band" and "no band," and 39 conditional attributes like "cylinder number" as nominal attribute and "viscosity" as numeric attribute. The number of instances in class "band" and "no band" is 228 and 312, respectively. So the size of the dataset is small. This means that we may not have enough instances for accurate classification. The procedure to find the best random forests is shown in Procedure 1.

In Procedure 1, there are four parameters to be defined, I, F, D, and R. I represents the initial number of trees in random forests. F represents the final number of trees in random forests. D represents the increment of the number of trees in the random forests in the for-loop. I and F are set 100 and 1000, respectively, in the experiment. I was set 100 to consider small enough number of trees in the forests. F was set 1,000 because the parameter showed the best results in average by Boström's experiment [12]. In the experiment, 100 trees, 1,000 trees, 10,000 trees, and 100,000 were generated for 34 datasets with default R value and ranked 1 to 4 based on accuracy. The average rank of 100 trees, 1,000 trees, 10,000 trees, and 100,000 trees is 3.12, 2.06, 2.44, and 2.38, respectively. For cylinder bands dataset, the accuracy of 1,000 trees and 100,000 trees is 79.81% and 80.19%, respectively. But, because the rank of 1,000 trees is the best in average for the 34 datasets, we use 1,000 trees for generalization. D was set to 25, because we found that smaller numbers than 25 generated almost the same accuracies. One may set smaller D value as R becomes smaller for more searches. R represents the number of randomly selected attributes to generate each decision tree in random forests. It is initialized by the double of the first integer less than $\log_2|A| + 1$, where $|A|$ is the number of attributes. The initial value of R was inspired by Breiman's recommendation, because smaller R value could generate better results for most cases [16, 19]. But one may set the value as the total number of attributes, if more through search is necessary. This necessity for rare cases could be raised by inspecting trial random forests also. For example, if the accuracy of random forests with $\{R$ = the number of attributes, $T = 1\}$ is greater than the accuracy of random forests with $\{R$ = the default number of attributes to pick randomly, $T = 100\}$ or $\{R = 1$, $T = 100\}$, we should initialize R with the total number of attributes. On the other hand, if the accuracy of random forests with $\{R = 1, T = 100\}$ is greater than the accuracy of random forests with $\{R$ = the default number of attributes to pick randomly, $T = 100\}$, we can set as above and do the grid search. R value is decreased during iteration. We consider R value to be up to 1, because the dataset is small, which means we may not have enough information for accurate classification, so we want randomness in tree building process to be maximized as the search proceeds. For details about random_forests (R, t), you may refer to Breiman's [7].

3. Experiments

3.1. Experiments for the Dataset "Cylinder Bands"

The dataset was obtained from UCI machine learning repository [20]. The number of attributes is 39. Among the 39 attributes, 19 attributes are nominal attributes and the other 20 attributes are numeric attributes. About 4.8% of attribute values is missing.

We first check if our suggested method could find better random forests effectively by generating trial random forests. If we generate random forests with the parameters ($T = 1$, $R = 39$), the accuracy of the random forests is 61.4815%, and the accuracy of each class is 18.4% for class "band" and 92.9% for class "no band" with 10-fold cross-validation. Because

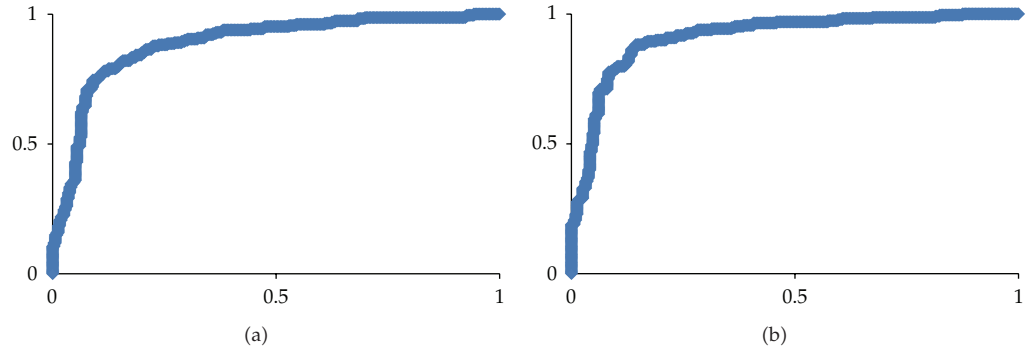

Figure 1: ROC curves for $R = 6$ (a) and $R = 1$ (b).

Table 4: Confusion matrix.

	$R = 6$ (default)				$R = 1$		
		TRUE				TRUE	
		Band	No band			Band	No band
Predicted	Band	162	66	Predicted	Band	177	51
	No band	24	288		No band	26	286

Table 5: Comparison of accuracy in other data mining methods.

Number	Method	Accuracy (%)	Sensitivity	Specificity	AUC (%)	Remarks
1	FLN tightest fit [3]	78.33	0.6375	0.91	NA	2/3 for training, 1/3 for testing, once
2	FLN ordered tightest fit [4]	85.56	0.7875	0.91	NA	2/3 for training, 1/3 for testing, once
3	Concurrent random forests [12]	80.19	NA	NA	89.32	10-fold cross validation
4	SubBag-JRip [5]	78.15	NA	NA	NA	10-fold cross validation
5	BAGGING-JRip [5]	79.26	NA	NA	NA	10-fold cross validation
6	C4.5 [8]	70.19	0.303	0.994	62.6	10-fold cross validation
7	Suggested random forests	**85.74**	0.7763	0.9167	91.45	10-fold cross validation

Table 6: The number of attributes and instances of data sets.

Data set	The number of attributes	The number of instances
Cylinder bands	39	540
Bridges	12	108
Dermatology	33	306
Post Operative	8	90

Table 7: The accuracy of trial random forests (RF) for each class with different parameters.

Bridges	($T = 1$, $R = 12$)	($T = 100$, $R = 4$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
1	0%	0%	0%
2	0%	100%	100%
3	0%	0%	27.3%
4	100%	100%	93.2%
5	0%	15.4%	23%
6	0%	0%	0%
7	0%	50%	60%
Accuracy of RF	41.5094%	**63.2075**%	**65.0943**%
Dermatology	($T = 1$, $R = 34$)	($T = 100$, $R = 6$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
1	97.3%	100%	100%
2	77%	85.2%	85.2%
3	97.2%	100%	100%
4	87.8%	87.8%	87.8%
5	90.4%	100%	100%
6	85.0%	95.0%	100%
Accuracy of RF	90.9836%	**95.6284**%	**96.4881**%
Post Operative	($T = 1$, $R = 8$)	($T = 100$, $R = 4$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
1	0%	0%	0%
2	26.7%	4.2%	0%
3	58.1%	85.9%	92.2%
Accuracy of RF	52.2222%	**62.2222**%	**65.5556**%

Table 8: The accuracy of random forests for the data set "Bridges."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (T)
8	53.7736	53.7736	53.7736	All
4 (default)	62.2642	63.2075	63.2075	All except 625, 750, 775, 800, 825
2	64.1509	66.0377	**66.9811**	200
1	65.0943	66.9811	**66.9811**	All except 100, 125, 175, 850, 875, 900

the dataset has 39 attributes, it is like conventional decision tree without pruning in which bootstrap method is applied. So, from the trial random forests, we can understand that the class "band" has very limited data instances for correct classification.

In order to see if more randomness and bootstrapping may give better results, we try random forests of parameters like ($T = 100$, $R = 6$) and ($T = 100$, $R = 1$). Note that $R = \lfloor \log_2 39 + 1 \rfloor = 6$ is the default value [15] for the number of attributes to pick randomly, while $R = 1$ is not. The result is summarized in Table 2.

From Table 1, we can expect that we may find better accuracy as we perform grid search by giving smaller R and larger T value in generating random forests. In order to find best possible results, we decrease R value from the initial number of attributes. But, because

Table 9: The accuracy of random forests for the data set "Dermatology."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (t)
12	95.9016	96.4481	96.4481	All except 100, 125, 500, 725, 800, 850 ~ 1000
6 (default)	95.6284	96.1749	96.4481	950
3	96.4481	96.9945	97.2678	200, 225, 250
2	96.7213	96.9945	97.2678	425
1	96.4481	97.2678	**97.541**	500, 525, 575, 600, 625, 750 ~ 950

Table 10: The accuracy of random forests for the data set "Post Operative."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (T)
8	60	61.1111	**65.5556**	100
4 (default)	60	61.1111	62.2222	200, 675, 725, 900, 925, 950, 975, 1000
2	61.1111	63.3333	64.4444	375, 625, 700, 725, 750, 775, 800, 825, 875
1	63.3333	65.5556	**65.5556**	200, 275, 300, 525, 550, 575, 650, 675, 700, 725, 950, 975, 1000

we do not know exactly which T value will generate the best result for a given R value, and very small increase in T value may generate the similar accuracy to previous ones, we increase the T value in given interval as we iterate.

In the experiment 10-fold cross-validation was used. Hence, the dataset is divided into ten equal subsets and each subset was used for test while nine other subsets were used for training. Random forests in weka were utilized for the experiment. Weka is a data mining package written in Java [21]. Table 3 shows the best accuracy based on Procedure 1 in which R and T were varied.

For each iteration the initial number of trees is 100, and 25 trees are incremented at each step to find proper number of trees in the forests, and the final number of trees in the forests is 1000. From the results in Table 3, we can see that we could get better accuracy as R value decreases. Table 4 shows the confusion matrix of the result for default and suggested R value.

Figure 1 shows ROC curves for $R = 6$ and $R = 1$. AUC for $R = 6$ is 88.8%, and $R = 1$ is 91.45%.

In Table 3, the accuracy of the random forests having 325 decision trees when the number of randomly selected attribute is one is 85.7407%, and this accuracy is yet the best accuracy according to literature survey. Table 5 summarizes the survey to compare the accuracy in other methods.

In Table 5, the accuracy of fuzzy lattice neurocomputing models is given at row 1 [3] and row 2 [4]. The training and testing were done once, so the experiments are less objective than other experiments. The result of 100,000 trees which is the best in the concurrent random forests [12] is presented at the 3rd row of the table. It is based on default value in the number

Table 11: The number of attributes and instances of data sets.

Data set	The number of attributes	The number of instances
DB world	4,703	64
Lung cancer	57	32

Table 12: The accuracy of trial random forests (RF) for each class with different parameters.

DB world	($T = 1$, $R = 46$)	($T = 100$, $R = 6$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
0	80%	91.4%	94.3%
1	86.2%	89.7%	96.6%
Accuracy of RF	82.8125%	**90.625**%	**95.3125**%
Lung cancer	($T = 1$, $R = 11$)	($T = 100$, $R = 4$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
1	66.7%	77.8%	77.8%
2	53.8%	76.9%	69.2%
3	50.0%	70.0%	80.0%
Accuracy of RF	56.25%	**75.0**%	**75.0**%

of attributes to pick randomly. It was generated by using Dell PowerEdge R815 sever with 48 cores and 64 GB memory so that it took a lot of computing resources, while our random forests were generated by using a Pentium PC with 2 GB main memory. SubBag with JRip [5] has some poorer result than the others as we can see at the 4th row. In the experiment BAGGING with JRip has better accuracy between the two experiments using JRip as we can see at the 5th row. 50 JRip classifiers were used for aggregation in the experiment. The 6th row of the table contains the accuracy of single decision tree of C4.5 that is the base of the first paper for the dataset [2]. From the value of sensitivity and specificity, we can understand that C4.5 has the tendency of neglecting minor classes. The last row shows the result of the suggested method. All in all, we can say that our random forests produced a very competitive result. Some other advantage of our method is high availability than other referred data mining methods. For example, several data mining tools that provide random forests are available like Salford system's [22], *R* [23], and weka, and so forth.

3.2. *Experiments for Other Datasets Having the Property of the Number of Attributes < the Number of Instances*

In order to see the suggested procedure can find better results than conventional application of random forests, other three datasets in different domain called "Bridges," "Dermatology," and "Post Operative" in UCI machine learning repository were tried. Dataset "Bridges" has 12 conditional attributes, 108 instances, and 7 classes. Dataset "Dermatology" has 33 conditional attributes, 366 instances, and 6 classes. Dataset "Post Operative" has 8 conditional attributes, 90 instances, and 3 classes. Table 7 has the results of trial random forests for each dataset. Note that all the four datasets including "cylinder bands" have the property, of the number of attributes < the number of instances as in Table 6.

Table 7 shows trial random forests for the three datasets.

Table 13: The accuracy of random forests for the data set "DB world."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (t)
12	90.625	90.625	90.625	All
6 (default)	90.625	90.625	90.625	All
3	93.75	93.75	93.75	All
2	93.75	93.75	93.75	All
1	93.75	95.3125	**95.3125**	All except 550, 650 ~ 850, 900 ~ 1000

Table 14: The accuracy of random forests for the data set "lung cancer."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (t)
8	65.625	65.625	71.875	575, 600, 625, 650
4 (default)	71.875	71.875	**75.0**	100, 125, 175, 200, 225, 275, 300, 325, 375, 400, 425, 450, 475, 550
2	68.75	71.875	**75.0**	125, 150, 175, 200
1	71.875	71.875	**75.0**	100 ~ 325

As we can see in Table 7, because we could generate better results with $R = 1$. Tables 8, 9, and 10 have the results of grid search for the datasets. Table 8 has the results of experiments for the dataset "Bridges."

For "Bridges" dataset the same best accuracy of 66.9811% was found at $R = 2$ and $R = 1$. But, while the accuracy was found only once at $R = 2$, the accuracy was found 34 times at $R = 1$. Table 9 has the results of experiments for the dataset "Dermatology."

Table 10 has the results of experiments for the dataset "Post Operative."

For "Post Operative" dataset the same best accuracy of 65.5556% was found at $R = 8$ and $R = 1$. But, while it was found only once at $R = 8$, it was found 19 times at $R = 1$. As we can see in Tables 8, 9, and 10, we could find better results based on the suggested procedure in other datasets also.

3.3. *Experiments for Another Datasets Having the Property of the Number of Attributes > the Number of Instances*

Because we have considered datasets having the property of the number of attributes < the number of instances, two other datasets in UCI machine learning repository, "DB world" and "lung cancer," that have the property of the number of attributes > the number of instances, were tried. Table 11 summarizes the datasets.

Because the two datasets might have many irrelevant attributes, preprocessing to select major attributes was performed first. It is based on weka's correlation-based feature subset (CFS) selection method [24] with best first search. For "DB world" and "lung cancer" datasets 46 and 11 attributes are selected, respectively. Table 12 shows the results of trial random forests.

Table 15: The accuracy of trial random forests (RF) for each class with different parameters.

DB world	($T = 1$, $R = 4703$)	($T = 100$, $R = 13$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
0	74.3%	91.4%	94.3%
1	79.3%	79.3%	48.3%
Accuracy of RF	**76.5625**%	**85.9375**%	73.4375%
Lung cancer	($T = 1$, $R = 57$)	($T = 100$, $R = 4$: def.)	($T = 100$, $R = 1$)
Accuracy per class			
1	33.3%	55.6%	22.2%
2	38.5%	69.2%	46.2%
3	40%	40%	60%
Accuracy of RF	37.5%	**56.25**%	**43.75**%

Table 13 has the results of experiments for the dataset "DB world."

Table 14 has the results of experiments for the dataset "lung cancer."

Note that the trial random forests of the dataset "lung cancer" have the same accuracy at R = default and $R = 1$. So we have the best accuracy at both R values. Experiments were done without attribute selection for the datasets of "DB world" and "lung cancer" to compare. Table 15 shows the results of trial random forests.

Table 16 has the results of experiments for the dataset "DB world." As the values for R, additional numbers like the whole number of attributes and $1/3$ of it were used also, because we know that the dataset contains many irrelevant attributes. The setting is based on Genuer et al.'s idea [17].

Table 17 has the results of experiments for the dataset "lung cancer." As the values for R, additional numbers like the whole number of attributes and $1/3$ of it were used also.

If we compare the best accuracies in Tables 13 and 16 for the dataset "DB world," the best accuracy of preprocessed dataset is 95.3125% and that of original dataset is 90.625%. Moreover, if we compare the best accuracies in Tables 14 and 17 for the dataset "lung cancer," the best accuracy of preprocessed dataset is 75.0% and that of original dataset is 59.375%. Therefore, we can conclude that our method is effective and the trial random forests well reflect whether the grid search is needed or not.

4. Conclusions

Rotogravure printing is very favored for massive printing tasks to print millions of copies. Hence, it is important to prevent process delays for better productivity. In order to reduce the delays preventive maintenance activity is more desirable, if we can predict some possible occurrence of bands in the cylinder. Therefore, more accurate prediction is important to reduce the delays. Random forests are known to be robust for missing and erroneous data as well as insufficient information with good performance, and moreover, they can utilize the fast building property of decision trees, so they do not require much computing time in most datasets for data mining, even though the forests have many trees. Hence, they are good for real word situation of data mining, because in the real world, lots of datasets have the property.

Because random forests have high possibility to generate better results when the combinations of parameters like the number of randomly picked attributes (R) and the

Table 16: The accuracy of random forests for the data set "DB world."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (t)
4703	82.8125	85.9375	87.5	125, 600, 625, 650, 725
1568	85.9375	89.0625	**90.625**	500, 575, 600, 625, 650, 675, 725, 850 ~ 925, 975, 1000
523	89.0625	90.0625	**90.625**	All except 100, 125, 700
175	89.0625	89.0625	**90.625**	100 ~ 175
59	85.9375	87.5	89.0625	100, 175, 275, 300
26	85.9375	87.5	87.5	All except 125, 150, 200 ~ 300, 500, 950, 975, 1000
13 (default)	81.25	82.8125	87.5	200, 225
7	82.8125	84.375	87.5	225, 250
4	76.5625	78.125	81.25	325, 1000
2	76.5625	79.6875	81.25	225, 350
1	73.4375	78.125	79.6875	375, 675, 775, 800

Table 17: The accuracy of random forests for the data set "lung cancer."

Number of attr. to pick randomly (R)	Worst accuracy (%)	Median accuracy (%)	Best accuracy (%)	Number of trees in random forests resulting in the best accuracy (t)
57	56.25	56.25	56.25	All
19	46.75	53.125	56.25	200, 325, 575, 600, 625, 650
12	43.75	46.875	56.25	175
6 (default)	46.875	50.0	**59.375**	150
3	46.875	50.0	56.25	300, 325
2	43.75	50.0	53.125	200, 250 ~ 300
1	34.375	37.5	56.25	150

number of trees in the forests (T) are good for given datasets, an effective procedure considering the properties of both of the datasets and random forests is investigated to find good results. Among R and T, because different R values could affect the accuracy of random forests very much, we suggest generating trial random forests to see the possibility of better results. Among the used six datasets, the five datasets showed that $R = 1$ is the best choice, while one dataset showed default R value and $R = 1$ is the best choices. $R = 1$ can be the best choice means that we need maximum randomness to spilt, because the datasets do not have sufficient information for correct classification. So for some datasets the default R value with appropriate number of trees could be the best choice, but for some other datasets smaller R value could be the best. In this sense, the trial random forests can do the role of a compass for further grid search.

Acknowledgment

This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2010.

References

[1] B. Evans and D. Fisher, "Using decision tree induction to minimize process delays in printing industry," in *Handbook of Data Mining and Knowledge Discovery*, W. Klösgen and J. M. Żytkow, Eds., pp. 874–880, Oxford University Press, 2002.

[2] B. Evans and D. Fisher, "Overcoming process delays with design tree induction," *IEEE Expert*, vol. 9, no. 1, pp. 60–66, 1994.

[3] V. G. Kaburlasos and V. Petridis, "Fuzzy Lattice Neurocomputing (FLN) models," *Neural Networks*, vol. 13, no. 10, pp. 1145–1170, 2000.

[4] A. Cripps, V. G. Kaburlasos, N. Nguyen, and S. E. Papadakis, "Improved experimental results using Fuzzy Lattice Neurocomputing (FLN) classifiers," in *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications (MLMTA '03)*, pp. 161–166, Las Vegas, Nev, USA, June 2003.

[5] P. Panov and S. Džeroski, "Combining bagging and random subspaces to create better ensembles," *Lecture Notes in Computer Science*, vol. 4723, pp. 118–129, 2007.

[6] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[9] "Big O notation," 16. 070 Introduction to computers and programming, MIT, http://web.mit.edu/16.070/www/lecture/big_o.pdf.

[10] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[11] W. W. Cohen, "Fast effective rule Induction," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123, Tahoe City, Calif, USA, 1995.

[12] H. Boström, "Concurrent learning of large-scale random forests," in *Proceedings of the Scandinavian Conference on Artificial Intelligence*, pp. 20–29, Trondheim, Norway, 2011.

[13] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[14] B. Efron and R. Tibshirani, "Improvements on cross-validation: the .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, 1997.

[15] Class Random Forest, http://weka.sourceforge.net/doc/weka/classifiers/trees/RandomForest.html.

[16] L. Breiman and A. Cutler, "Random Forests," http://www.stat.berkeley.edu/users/breiman/RandomForests/.

[17] R. Genuer, J. Poggi, and C. Tuleau, "Random Forests: some methodological insights," Tech. Rep. inria00340725, INRIA, 2008.

[18] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2-3, pp. 18–22, 2002.

[19] L. Breiman and A. Cutler, "Random Forests," http://www.stat.berkeley.edu/users/breiman/RandomForests/.

[20] A. Frank and A. Asuncion, "UCI machine learning repository," University of California, School of Information and Computer Science, Irvine, Calif, USA, 2010, http://archive.ics.uci.edu/ml.

[21] WEKA, http://www.cs.waikato.ac.nz/ml/weka/.

[22] "Salford systems-random forests," http://www.salford-systems.com/en/products/randomforests.

[23] "The R project for statistical computing," http://www.r-project.org/.

[24] M. A. Hall, *Correlation-based feature subset selection for machine learning [Ph.D. thesis]*, The University of Waikato, Hamilton, New Zealand, 1999.