Text Categorization with Diversity Random Forests

Chun Yang[1], Xu-Cheng Yin[1,⋆], and Kaizhu Huang[2]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
xuchengyin@ustb.edu.cn
[2] Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China

Abstract. Text categorization (TC), has many typical traits, such as large and difficult category taxonomies, noise and incremental data, etc. Random Forests, one of the most important but simple state-of-the-art ensemble methods, has been used to solve such type of subjects with good performance. most current Random Forests approaches with diversity-related issues focus on maximizing tree diversity while producing and training component trees. There are much diverse characteristics for component trees in TC trained on data of noise, huge categories and features. Consequently, given numerous component trees from the original Random Forests, we propose a novel method, Diversity Random Forests, which diversely and adaptively select and combine tree classifiers with diversity learning and sample weighting. Diversity Random Forests includes two key issues. First, by designing a matrix for the data distribution creatively, we formulate a unified optimization model for learning and selecting diverse trees, where tree weights are learned through a convex quadratic programming problem with given sample weights. Second, we propose a new self-training algorithm to iteratively run the convex optimization and automatically learn the sample weights. Extensive experiments on a variety of text categorization benchmark data sets show that the proposed approach consistently outperforms state-of-the-art methods.

1 Introduction

Classification techniques, especially text categorization, have many applications in Data Mining (DM) and Information Retrieval (IR), e.g., spam detection, sentiment detection, personal email sorting and document ranking [1]. Typical issues in text categorization and recommendation systems are large and difficult category taxonomies, huge samples, noise and incremental data, and various features. Classifier ensemble is a potential solution for such type of subjects. Many research efforts demonstrated that the Random Forests approach [2] is the most important but simple state-of-the-art ensemble for classification, consequently, for text categorization.

Random Forests can exploit implicit and explicit diversities together. The method combines the "Bagging" idea for instance sampling with the implicit diversity and the random selection of variables for feature selection with the explicit diversity. Generally, the performance of a classifier ensemble (including Random Forests) relies on not only the accuracy but also the diversity of component trees. Consequently, how to diversely generate and combine diverse classifiers plays an important role in Random Forests.

⋆ Corresponding author.

C.K. Loo et al. (Eds.): ICONIP 2014, Part III, LNCS 8836, pp. 317–324, 2014.

On the field of Random Forests research, there are many researches for improving Random Forests with diversity-related issues, most of which focus on maximizing tree diversity while producing and training component trees. Liu et.al.[3] proposed Max-diverse Ensemble method, which has the maximum diversity and uses only simple probability averaging without any feature selection criterion or other random elements. Later, Liu et.al.[4] proposed Coalescence method, which coalesces a number of points in the random-half of the spectrum and is found to perform better than any single operating point in the spectrum, without the need to tune to a specific level of randomness.

Obviously, In TC, there are a lot of diverse characteristics for component trees which are trained on data of noise, large categories and huge features, i.e., some trees or a subset of trees by properly selecting will be much diverse from each other. Alternatively, we improve Random Forests with diversity from pruning ensemble, as ensemble of the partial available component trees may be better than that of the whole [5]. Given numerous component trees from the original Random Forests, we want to diversely and adaptively select and combine tree classifiers with diversity learning.

Moreover, in classifier ensemble, all existing diversity measures are calculated on the training set, which means the performance of optimization relies on the samples of training set besides the diversity learning itself [6,7,8]. In some relative fields, researchers suggest sample weighting is needed to correct for imperfections in the samples that might lead to bias and other departures between the sample and the reference population. Adaboost[9] is one of the most famous sample weighting models.

Consequently, given numerous component trees from the original Random Forests, we propose a novel method, **D**iversity **R**andom **F**orests (DRF), which diversely and adaptively select and combine tree classifiers with diversity learning and sample weighting. Diversity Random Forests uses a self-training algorithm to iteratively run the convex optimization and automatically learn the sample weights. Each iteration of this self-training algorithm consists of two main steps: (1) calculate tree weights by solving an optimization problem with sample weights known, and then (2) update sample weights. In the first step, diversity learning with sample weights is converted into a unified convex quadratic programming optimization model, by creatively setting the sample distribution as a diagonal matrix. In the second step, sample weights are automatically and adaptively updated with a dynamically damped learning trick. Therefore, the whole self-training algorithm has a good convergence performance. Moreover, experimental results on a variety of text categorization benchmark data sets definitely show that our proposed approach has very promising performance.

The rest of the paper is organized as follows. The DRF model is presented in Section 2, and more details on the learning algorithm is described in Section 3. Section 4 shows extensive experimental results. Finally, conclusion is drawn in Section 5.

2 Diversity Random Forests

2.1 Random Forests

Random Forests [2] are an ensemble learning method for classification. It generates a multitude of decision trees based on bootstrap samples of the training data and outputs the class that is the mode of the classes output by individual trees. For each node of a

tree, m variables are randomly chosen and the best split based on these m variables is calculated based on the bootstrap data. Traditionally, m is set to $\lceil\sqrt{u}\rceil$, where u stands for the number of variable. Each decision tree results in a classification and is said to cast a weighted vote for that classification, and Random Forests returns the class that received the most votes.

As various theoretical and empirical studies shows[10,11,12], Random Forests are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting. In fact, they are considered to be one of the most accurate general-purpose learning techniques available.

In the paper, we formulate an optimization model based on the original Random Forests model [2]. Moreover, instead of the original output, the oracle output $\mathbf{O}$ of Random Forests is used for the optimization. Let the number of samples set be N, and the number of component trees L. $\mathbf{O}$ is a $N \times L$ matrix, and element

$$O_{ij} = \begin{cases} 1 \text{ the } j^{th} \text{ tree classified the } i^{th} \text{ sample correctly} \\ -1 \text{ otherwise} \end{cases} \tag{1}$$

2.2 Diversity Random Forests Model

As an ensemble approach, Random Forests can be improved by pruning component trees. Specially, for weighted-vote Random Forests, the improvement is equivalent to a mathematical optimization problem with tree weights. Define tree weights vector $\mathbf{w} = [\mathrm{w}_1, \mathrm{w}_2, ..., \mathrm{w}_L]$, where $\sum_{j=1}^{L} \mathrm{w}_j = 1,\ \mathrm{w}_j \geq 0$. Traditionally, $\mathbf{w}$ is learned by

$$\begin{aligned} \mathbf{w_{opt}} &= argmin_{\mathbf{w}} f_1(\mathbf{w}, \mathbf{P}) \\ s.t.\ \mathbf{w_{opt}} &\succeq 0,\ \mathbf{1^T w_{opt}} = 1. \end{aligned} \tag{2}$$

where $\mathbf{P}$ is the accuracy of each tree on training set. $\mathbf{P} = [P_1, P_2, ..., P_L]^T$, where $P_j = \sum_{i=1}^{N} O_{ij}$. The optimization function in Equation (2) usually has functional relationship f_1 with the accuracy $\mathbf{P}$.

Previous works show that a multi-criteria searching for an ensemble that maximizes both accuracy and diversity leads to more accurate ensembles than a single optimization criterion. Thus, consider diversity in component trees of Random Forests and add a regularization term about diversity to expand Equation (2) as,

$$\begin{aligned} \mathbf{w_{opt}} &= argmin_{\mathbf{w}} f_1(\mathbf{w}, \mathbf{P}) + \lambda div(\mathbf{w}) \\ s.t.\ \mathbf{w_{opt}} &\succeq 0,\ \mathbf{1^T w_{opt}} = 1. \end{aligned} \tag{3}$$

In Equation (3), $div(\mathbf{w})$ is the diversity of ensemble with classifier weights $\mathbf{w}$. If use pairwise diversity method, $div(\mathbf{w})$ can be calculated as an average,

$$\begin{aligned} div(\mathbf{w}) &= \mathbf{w}^T \mathbf{D} \mathbf{w} \\ \mathbf{D} &= f_D(\mathbf{O}^T \mathbf{O}, \mathbf{1}_{N\times 1}^T \mathbf{O}) \end{aligned} \tag{4}$$

where $\mathbf{D}$ is the diversity matrix of component trees, which has functional relationship f_D with $\mathbf{O}^T\mathbf{O}$ and $\mathbf{1}^T\mathbf{O}$.

In the paper, the Disagreement(dis) [13] is chosen to measure diversity, which is calculated by,

$$\mathbf{D_{dis}} = \frac{1}{2N}(N\mathbf{1}_{L\times L} - \mathbf{O}^T\mathbf{O}) \tag{5}$$

If use the average accuracy to calculate $f_1(\mathbf{w}, \mathbf{P})$, and the pairwise diversity Disagreement to calculate $div(\mathbf{w})$, then Equation (3) equals,

$$\begin{aligned} \mathbf{w_{opt}} &= argmin_{\mathbf{w}} - \lambda \mathbf{w}^T \mathbf{D_{dis}} \mathbf{w} - \mathbf{P}\mathbf{w} \\ s.t.\ \ \mathbf{w_{opt}} &\succeq 0,\ \mathbf{1^T w_{opt}} = 1. \end{aligned} \tag{6}$$

One issue of the optimization is how to determine the parameter λ. However, empirical analysis shows that the recognition rate has a very little change when the value λ changes.

More importantly, the performance of optimization function is totally different because of different training set selection. Considering the influence of training set, we expand Equation (6) as,

$$\begin{aligned} \mathbf{w_{opt}} &= argmin_{\mathbf{w}} - \lambda \mathbf{w}^T \mathbf{D_{dis,\Omega}} \mathbf{w} - \mathbf{P_\Omega}\mathbf{w} \\ s.t.\ \ \mathbf{w_{opt}} &\succeq 0,\ \mathbf{1^T w_{opt}} = 1. \end{aligned} \tag{7}$$

where Ω is a parameter of the data distribution (sample weights). This (Equation (7)) is the model of our Diversity Random Forests.

To simplify calculation and remain the optimization as a convex problem, we creatively set Ω as a $N \times N$ diagonal matrix, and $diag(\Omega)_i = \Omega_{ii}$ stands for the weight of sample x_i, where $diag(\Omega)_i \geq 0,\ \mathbf{1}^T diag(\Omega) = 1$. Thus, $\mathbf{P_\Omega}$ and $\mathbf{D_{dis,\Omega}}$ can be calculated by,

$$\begin{aligned} \mathbf{P_\Omega} &= \mathbf{1}^T \Omega \mathbf{O} \\ \mathbf{D_{dis,\Omega}} &= \tfrac{1}{2}(1_{L\times L} - \mathbf{O}^T \Omega \mathbf{O}) \end{aligned} \tag{8}$$

Consequently, the optimization (7) can be simplified to a convex quadratic programming problem with a given Ω.

3 DRF Algorithm

It is difficult to find the solution for the optimization in Equation (7) without both $\mathbf{w}$ and Ω. However, with known Ω, the optimization is simplified to a quadratic programming problem. Thus, we propose an iterative learning algorithm, **D**iversity **R**andom **F**orests (DRF) Algorithm, which is shown in Algorithm 1.

In Algorithm 1, the validation set is bootstrapped from the original training set of Random Forests. We assume the sample weights parameter Ω_{t+1} has a relationship with Ω_t, and use a dynamically damped trick, i.e., the damped factor $\beta_t \in [0, 1]$ and $\beta_t \leq \beta_{t+1}$. In the paper, we set β_t as,

$$\beta_t = \frac{1}{t} \tag{9}$$

Algorithm 1: DRF Algorithm

Input:

Tr: the validation set. $|Tr| = N$

$H = \{h_1, h_2, ..., h_L\}$: the component tree set, $|H| = L$.

M: pairwise diversity method.

Output:

$\mathbf{w}$: the component tree weights.

Parameter:

T: the max epoch.

Ω_t: a diagonal matrix, and $diag(\Omega_t)_i$ is the weight of sample x_i used to calculate $\mathbf{w}$ on the t^{th} turn.

Ω_t^*: a diagonal matrix, and $diag(\Omega_t^*)_i$ is the updated weight of sample x_i on the t^{th} turn.

ϵ_t: the error rate on the t^{th} turn.

β_t: a parameter that $\beta_t \in [0, 1]$, and $\beta_t \leq \beta_{t+1}$.

Procedure:

1: Set $diag(\Omega_1)_i = 1/N$.
2: **For** $t = 1, 2, ..., T$;
3: Use Equation (7) and (8) to calculate $\mathbf{w}$.
4: Calculate ϵ_t by $\mathbf{w}$ and Tr.
5: Use ϵ_t to calculate updated weight Ω_t^*.
6: $\Omega_{t+1} = \beta_t \Omega_t^* + (1 - \beta_t)\Omega_t$
7: **End**

The updated weight matrix Ω_t^* increases the weights of easily wrong-classified samples. We update Ω_t^* by DRF-Exp, which gets the idea from the adaptive reweighting step in Boosting [9]. In Boosting, a distribution of weights over training samples is adaptively maintained, and component trees are created sequentially with each tree concentrating on instances that are not well learnt by previous ones. With this mechanism, the learning process is more efficient. Similarly, DSWL-Exp updates Ω_t^* by,

$$\begin{aligned} \alpha &= \tfrac{1}{2} ln \frac{1 - \epsilon_t}{\epsilon_t} \\ diag(\Omega_{t+1}^*)_i &= \frac{diag(\Omega_t^*)_i exp(-\alpha m_i)}{Z_{t+1}} \end{aligned} \tag{10}$$

where Z_{t+1} is a normalization factor, then $diag(\Omega_{t+1}^*)$ is a valid distribution.

4 Experiments

We evaluated the performance of DRF by comparing against some state-of-art methods, such as Multinomial Naive Bayesian, J48, Support Vector Machines and Random Forests, on a variety of document collections.

4.1 Experimental Data

The detailed characteristics of the various document collections used in our experiments are available in [14].[1] More information for the data sets is presented in Table 1.

[1] `http://sourceforge.net/projects/weka/files/datasets/text-datasets/19MclassTextWc.zip`

Table 1. Benchmark Datasets

DataSet	Source	Docs	Words	Classes	DataSet	Source	Docs	Words	Classes
fbis	TREC	2463	2000	17	re1	Reuters	1657	3758	25
la1s	TREC	3204	13472	6	tr11	TREC	414	6429	9
la2s	TREC	3075	13472	6	tr12	TREC	313	5804	8
oh0	OHSUMED	1003	3182	10	tr21	TREC	336	7902	6
oh10	OHSUMED	918	3012	10	tr23	TREC	204	5832	6
oh15	OHSUMED	1050	3238	10	tr31	TREC	927	10128	7
oh5	OHSUMED	913	3100	10	tr41	TREC	878	7454	10
ohscal	OHSUMED	11162	11465	10	tr45	TREC	690	8261	10
re0	Reuters	1504	2886	13	wap	WebACE	1560	8460	20

4.2 Experimental Setup

The experiment compares DRF with some state-of-art methods, e.g., Multinomial Naive Bayes(MNB), J48, Support Vector Machines(SVM,[15]), Random Forests(RF, [2]). Both Multinomial Naive Bayes and J48 classifier are generated by WEKA,[2] and Random Forests classifier is generated by Matlab toolbox. [3] For each method, all parameters are set by default. In SVM, the Linear kernel is used, and the best c and g parameter is selected by cross validation from $c = 2^{-5}, 2^{-4}, ..., 2^{5}$, $g = 2^{-5}, 2^{-4}, ..., 2^{5}$.

In the experiment, 5-fold cross validation is performed on each data set. We assign Ranks to evaluate the methods' performance on each data set [16]. Mark the best method Rank 1, and the worse, the larger. Then calculate the average Rank for each method. Moreover, we also calculated the average recognition rate (AVE).

4.3 Results

The experimental results are shown in Table 2. In addition, the highest recognition rate for each data set is highlighted in boldface. As shown in Table 2, we can observe:

- Among four state-of-art methods(J48, MNB, SVM, RF), the best rank corresponds to RF(2.4), followed by SVM(2.8), MNB(3.6) and J48(4.6). On most data sets, RF achieves the best recognition rate, and is slightly worse than SVM on 'fbis', 're1', 'tr11', 'tr21', 'tr41' and 'wap' data sets. These results show that RF is a powerful technique for text categorization.
- Moreover, DRF ranks 1.6, and is 0.9% higher than Random Forests for the average classification precision. On most data sets, DRF achieves an 1%-4% higher recognition rate than RF, except on 'la1s', 'la2s' and 'wap'. That is to say, in TC, our proposed method, DRF, can utilize diversity in component trees and select a proper subset of trees in RF for ensemble.
- Specifically, by selecting training sets (calculate the sample weights) carefully, DRF has the minimum Rank and largest average recognition rate, and outperforms

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] https://code.google.com/p/randomforest-matlab/

J48, MNB, SVM and RF. In most cases, DRF achieves the best performance when there are enough training data for learning component trees, tree weights and sample weights. Consequently, our methods obtain the best rank (1.6) in all experimental approaches.

Table 2. Comparison of recognition rate (%) (Average±Standard Deviation).

Datasets	J48	MNB	SVM	RF	DRF
fbis	72.03 ± 2.07	77.30 ± 1.84	82.79 ± 1.07	82.74 ± 1.17	$\mathbf{83.35 \pm 1.20}$
la1s	75.56 ± 1.93	87.45 ± 0.51	87.83 ± 1.11	$\mathbf{88.08 \pm 1.61}$	88.05 ± 1.55
la2s	76.33 ± 1.66	88.78 ± 1.03	88.85 ± 1.17	$\mathbf{88.93 \pm 1.60}$	88.80 ± 1.60
oh0	81.05 ± 4.99	$\mathbf{88.43 \pm 3.09}$	85.14 ± 2.85	88.03 ± 2.66	88.03 ± 2.66
oh10	68.38 ± 3.06	78.00 ± 3.80	76.29 ± 4.54	80.95 ± 6.79	$\mathbf{81.14 \pm 6.79}$
oh15	72.39 ± 5.08	$\mathbf{82.04 \pm 1.81}$	76.88 ± 3.74	80.49 ± 5.08	81.04 ± 5.10
oh5	80.71 ± 5.13	87.47 ± 3.01	85.84 ± 4.68	87.58 ± 2.74	$\mathbf{89.32 \pm 2.74}$
ohscal	70.23 ± 5.10	73.99 ± 1.14	76.63 ± 1.49	80.87 ± 1.21	$\mathbf{80.93 \pm 3.21}$
re0	70.68 ± 1.96	76.87 ± 4.32	81.25 ± 4.30	81.32 ± 5.30	$\mathbf{81.52 \pm 5.26}$
re1	77.43 ± 4.43	79.05 ± 6.16	81.83 ± 4.23	81.81 ± 5.86	$\mathbf{82.35 \pm 5.86}$
tr11	77.06 ± 3.24	84.07 ± 3.07	87.20 ± 1.58	84.53 ± 2.87	$\mathbf{88.41 \pm 2.87}$
tr12	79.21 ± 4.05	81.76 ± 7.43	85.93 ± 4.02	87.19 ± 5.33	$\mathbf{87.84 \pm 5.33}$
tr21	77.95 ± 7.25	60.09 ± 6.01	86.00 ± 4.21	85.31 ± 4.53	$\mathbf{86.28 \pm 4.46}$
tr23	$\mathbf{92.68 \pm 5.17}$	69.07 ± 9.13	83.34 ± 4.66	83.89 ± 8.54	86.30 ± 8.54
tr31	93.53 ± 1.37	95.04 ± 1.35	97.09 ± 0.82	97.19 ± 2.52	$\mathbf{97.52 \pm 2.52}$
tr41	92.03 ± 2.67	93.97 ± 2.94	$\mathbf{94.76 \pm 1.69}$	92.94 ± 2.35	93.96 ± 2.35
tr45	91.01 ± 1.50	82.46 ± 3.78	89.28 ± 4.36	90.29 ± 4.51	$\mathbf{92.75 \pm 4.51}$
wap	65.38 ± 2.58	79.94 ± 3.94	$\mathbf{84.49 \pm 1.98}$	82.71 ± 2.15	81.23 ± 2.15
AVE	78.54	81.43	85.08	85.83	**86.60**
Ranks	4.6	3.6	2.8	2.4	**1.6**

5 Conclusion

Random Forests approach is widely considered as an effective method to improve accuracy of various component trees, which has a variety of applications in information retrieval and data mining, e.g., text categorization, image retrieval, and recommendation systems. By improving Random Forests from ensemble pruning aspect, we propose a convex mathematical model for ensembling components in Random Forests, which takes into account both diversity learning and sample weighting. We also propose an iterative self-training algorithm for DRF, where the optimization problem is simplified as a convex quadratic programming problem at each iteration. In the experiments, DRF is compared with other state of art methods, e.g., J48, Multinomial Naive Bayes, Support Vector Machines and Random Forests. A series of experiments on benchmark data sets show that our proposed method achieves very encouraging results for text categorization.

Acknowledgments. The research was partly supported by the National Natural Science Foundation of China (61105018, 61175020).

References

1. Manning, C.D., Prabhakar, R., Hinrich, S.: Introduction to Information Retrieval. Cambridge University Press (2008)
2. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
3. Liu, F.T., Ting, K.M., Fan, W.: Maximizing tree diversity by building complete-random decision trees. In: Proceeding of PAKDD, pp. 605–610 (2005)
4. Liu, F.T., Ting, K.M., Yu, Y., Zhou, Z.H.: Spectrum of variable-random trees. J. Artif. Intell. Res. 32, 355–384 (2008)
5. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence 137, 239–263 (2002)
6. Yin, X.-C., Huang, K., Hao, H.-W., Iqbal, K., Wang, Z.-B.: Classifier ensemble using a heuristic learning with sparsity and diversity. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part II. LNCS, vol. 7664, pp. 100–107. Springer, Heidelberg (2012)
7. Yin, X.C., Huang, K., Hao, H.W., Iqbal, K., Wang, Z.B.: A novel classifier ensemble method with sparsity and diversity. Neurocomputing 134, 214–221 (2014)
8. Yin, X.C., Huang, K., Yang, C., Hao, H.W.: Convex ensemble learning with sparsity and diversity. Information Fusion 20, 49–59 (2014)
9. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proceedings of ICML, pp. 148–156 (1996)
10. Biau, G.: Analysis of a random forests model. J. Mach. Learn. Res. 13, 1063–1095 (2012)
11. Genuer, R., Poggi, J.M., Tuleau-Malot, C.: Variable selection using random forests. Pattern Recognition Letters 31(14), 2225–2236 (2010)
12. Verikas, A., Gelzinis, A., Bacauskiene, M.: Mining data with random forests: A survey and results of new tests. Pattern Recognition 44(2), 330–349 (2011)
13. Skalak, D.B.: The sources of increased accuracy for two proposed boosting algorithms. In: Proceeding of AAAI, pp. 120–125 (1996)
14. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Proceedings of European PKDD, pp. 424–431 (2000)
15. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Trans. Intelligent Systems and Technology 2(3), 1–27 (2011), `http://www.csie.ntu.edu.tw/cjlin/libsvm`
16. Brazdil, P., Soares, C.: A comparison of ranking methods for classification algorithm selection. In: Proceedings of ECML, pp. 63–74 (2000)