A Deep Forest Method for Classifying E-Commerce Products by Using Title Information

Jin Dai*†, Tianyu Wang†‡, and Shaowei Wang†
*School of Information Science and Engineering, Jinling College, Nanjing University, Nanjing 210089, China
†School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
‡National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
Email: 030308@jlxy.nju.edu.cn, {tianyu.alex.wang,wangsw}@nju.edu.cn

***Abstract*—E-commerce platforms, such as Amazon, eBay and Tmall, are flooded with various types of products. These platforms need to classify the products to facilitate product management and recommendation, which however can be very costly by using manual work. Recently, ML-based classification technology, e.g. SVM and DL, has been widely used in industry to classify e-commerce products by using the text information in the titles given by the merchants. However, current techniques can be inefficient and inaccurate when the number of categories is large and the data scale is small, as in the e-commerce product classification problem. In this paper, we propose a novel machine learning method for the problem, referred to as gcForest, which utilizes the cascade forest of decision trees and multi-grained scanning mechanisms. After preprocessing the product title information by using a word examination technology, the TF-IDF algorithm, we carry out a serials of experiments with 4000 samples belonging to 35 categories of products. The experiment results show that the classification accuracy using gcForest is 92.38%, which outperforms SVM with RBF kernel (86.88%), SVM with linear kernel (89.73%) and CNN (86.86%).**

***Index Terms*—GcForest, product classification, word segmentation.**

I. Introduction

With the rapid development of Internet technology and e-commerce platforms, online shopping now has a profound impact on the development of enterprises and people's daily lives [1]. Manual classification can be inefficient and inaccurate considering the massive tags of e-commerce products and the diverse descriptions of text information in the product titles. Moreover, a large mount of new products constantly emerge on e-commerce platforms with various types every day, which makes real time classification a challenging task.

Recently, with the progress of artificial intelligence (AI), machine learning (ML) based classification technology has been widely used in industry to classify products by using the text information in the titles given by the merchants. Support vector machine (SVM), which is a well known classifier for both linear and non-linear classification problems, has been used in text classification problem [2]. In [3], the authors propose a rule-based question classifier based on SVM by using information extracting, which has been used in search of online resource. The experimental accuracy with 165 question categories is 96.9%. In [4], SVM is applied to document classification problem with both original features and processed low dimension features, which achieves a classification accuracy with 68.1% and 76.7%, respectively. In [5], the authors propose a SVM classifier to classify scientific documents in Wikipedia by using the topic methods. The classification accuracy with 150 topics is 84.60%.

Deep learning (DL) is another widely used classification technology, which achieves success in the field of image classification, and it is also introduced in text-based classification problems [6]. In [7], the authors propose a new learning mechanism to train Text-CNN to classify text and non-text components in images, including binary text/non-text information, text area mask and character label. The experimental results show that f-measure is 82%. In [8], the authors propose the application of deep convolutional neural network (CNN) architecture to analyze handwritten documents, based on the segmented word recognition method. The average accuracy of the experiments is 86.59%.

However, both SVM and DL have their own defects for the considered e-commerce product classification problem. SVM is more suitable for binary classification problems, but the considered problem is a multi-class problem with a large number of product categories. DL usually demands a large amount of training data [9], but the considered problem usually does not have a large number of samples for each category. In 2017, Zhou and Feng propose a novel tree-based deep network for supervised learning, referred to as gcForest, which applies the multi-grained scanning and the cascade deep forest tehniques [10]. It shows great performance in multi-classification with small-scale samples. In [11], gcForest has been used in detecting spammer, by using a group of email features which can reflect the mail users behavior. The classification accuracy is 93.28%. In [12], gcForest is used as an image clothes classifier with a classification accuracy 90.2%.

In the paper, we consider an e-commerce product classification problem using the product title information. We extract 4000 title samples that belong to 35 product categories from a real e-commerce platform. Specifically, we propose a gcForest classification method for the considered problem. Firstly, we apply attribute data dictionary and feature vector technology

This work was partially supported by the National Natural Science Foundation of China (61801208, 61931023, 61671233), the Jiangsu Science Foundation (BK20170650), the Postdoctoral Science Foundation of China (BX201700118, 2017M621712), the Jiangsu Postdoctoral Science Foundation(1701118B), and the open research fund of National Mobile Communications Research Laboratory (2019D02).

978-1-7281-4905-9/20/$31.00 ©2020 IEEE

to preprocess the title samples and then utilize gcForest to train the classification model, which has 2 completely random forests and 2 random forests, each owning 500 decision trees. We randomly selected 80% data as the training set, and the remaining 20% data as the test set. In our experiments, the classification accuracy using gcForest is 92.38%, which outperforms SVM with RBF kernel (86.88%), SVM with linear kernel (89.73%) and CNN (86.86%).

The rest of the paper is organized as follows. In Section II, we propocess the original data samples by using the attribute data dictionary and feature vector. In Section III, we utilize gcForest method for the considered problem. In Section IV we give the experimental results and analyze them. Finally, in Section V, we conclude the paper.

II. Data Set Instruction

A. Data Set Establishing

The original data set contains 4000 samples belonging to 35 categories of products, each sample of which consists of a paragraph of text and a label that represents its category. The original data set is denoted by

$$\mathcal{P} = \left\{p_1, \quad p_2, \quad \dots \quad p_n, \quad \dots \quad p_{N=4000}\right\}. \tag{1}$$

Each sample $p_n \in \mathcal{P}$ is given by

$$p_n = \left(t_n, \quad l_n\right), \tag{2}$$

in which t_n is a paragraph of text information within 100 Chinese words, and l_n is the label that represents the category of the corresponding product.

Here, we give an example of p_n, in which t_n is given by

$$t_n = '\mathit{Jiulou\ Pavilion, Jingdezhen, Teacup, Ceramic, Tea\ cup,\ Cup,\ Ceramic\ medium\ men's\ office\ cup\ double\ line\ cup}\ 450ml, \mathit{Snowscape,\ Without\ gift\ box}'. \tag{3}$$

The text message t_n describes the product information of a commodity cup, including its brand, material, style, capacity and target group positioning. This product is tagged with a categorical label l_n, given by

$$l_n = '\mathit{Kitchen\ utensils} > \mathit{Tea\ set\&Coffee\ set} > \mathit{Cup}', \tag{4}$$

which implies that it belongs to a large class $Kitchen\ utensils$, a medium class $Tea\ set\&Coffee\ set$ and a small class Cup. All categories consist of 3 parts as shown in the example p_n, i.e., large class, medium class and small class. The set of all categories is denoted by

$$\mathcal{C} = \left\{tag_1, \quad tag_2, \quad \dots, \quad tag_m, \quad \dots \quad tag_{M=35}\right\}. \tag{5}$$

B. Data Preprocessing

As these samples are unstructured text data, we preprocess the samples into structured data that can be used in the gcForest training. Our data preprocessing method consists of two parts, i.e., the construction of attribute data dictionary and the construction of product feature vector.

In construction of attribute data dictionary, for each category $tag_m \in \mathcal{C}$, the set of the corresponding samples is given by

$$\mathcal{P}_m = \{(t, l) \in \mathcal{P} | l = tag_m\}. \tag{6}$$

We use $\mathcal{P}_m$ as the input of term frequency-inverse document frequency (TF-IDF) [13], which outputs 16 keywords with the highest frequency, denoted by

$$\mathcal{W}_m = \left\{w_{m,1}, \quad w_{m,2}, \quad \dots \quad w_{m,16}\right\}. \tag{7}$$

Then, the attribute data dictionary $\mathcal{D}$ is defined as the union of all keywords from all categories, which is given by

$$\mathcal{D} = \bigcup_{m=1}^{M} \mathcal{W}_m. \tag{8}$$

In our data set, after removing the repeated words in attribute data dictionary, we obtain 441 keywords, and attribute data dictionary is rewritten as

$$\mathcal{D} = \left\{d_1, \quad d_2, \quad \dots \quad d_k, \quad \dots \quad d_{K=441}\right\}. \tag{9}$$

The construction of attribute data dictionary is summarized in Algorithm 1.

For each sample $p_n \in \mathcal{P}$, we define a feature vector $\boldsymbol{a_n}$, given by

$$\boldsymbol{a_n} = \left(a_{n,1}, \quad a_{n,2}, \quad \dots \quad a_{n,k}, \quad \dots \quad a_{n,K=441}\right), \tag{10}$$

in which $a_{n,k}$ is defined as follows

$$a_{n,k} = \begin{cases} 1, & t_n \text{ contains } d_k \\ 0, & t_n \text{ does not contains } d_k \end{cases}. \tag{11}$$

In addition, the category label $l_n \in p_n$ is replaced by a integer $c_n \in [1,\ 35]$, given by

$$c_n = m, \quad if\ l_n = tag_m. \tag{12}$$

Then the original sample $p_n = (t_n,\ l_n)$ is replaced by a structured sample $q_n = (\boldsymbol{a_n},\ c_n)$, and the total data set is given by

$$\mathcal{Q} = \bigcup_{n=1}^{N} \{q_n\}. \tag{13}$$

The construction of product feature vector is summarized in Algorithm 2.

III. GcForest Model

In this section, we first introduce basic element of decision forest in gcForest method, then describe two key technologies, i.e. multi-grained scanning and cascade forest, and show how it is applied in the considered e-commerce product classification problem.

Algorithm 1 Construction of attribute data dictionary.

Input: Product set $\mathcal{P}$; category set $\mathcal{C}$;
Output: Attribute data dictionary set $\mathcal{D}$;

```
Initialize D = ∅;
for Every element tag_m in the C do
    for Every element p_n in the P do
        if l_n = tag_m then
            string = string + t_n;
        end if
    end for
    TF-IDF(string, 16) return W_m
    D = D ∪ W_m;
end for
return D;
```

Algorithm 2 Construction of product feature vector.

Input: Product set $\mathcal{P}$; attribute data dictionary set $\mathcal{D}$; categories set $\mathcal{C}$;
Output: Training data set $\mathcal{Q}$

```
Initialize Q = ∅;
for Every element p_n in the P do
    for Every element tag_m in the C do
        if l_n = tag_m then
            c_n = m;
        end if
    end for
    Initialize a_n = 0;
    for Every element d_k in the D do
        if t_n contains d_k then
            a_{n,k} = 1;
        end if
    end for
    Q = Q ∪ {(a_n, c)};
end for
return Q;
```

A. Decision Forest

This gcForest classifier adopts the idea of decision tree. Each tree in a forest grows using a split node based on a feature selected from a sample. Each forest contains 500 decision trees, which is a hyper-parameter that can be specified. Fig. 1 gives an example with 3 categories, then the each classification probability distribution is a 3-dim vector, which can be calculated at the leaf node in each tree for the sample. And the class vector is obtained by averaging the classification probability distribution of all decision trees in the forest. Then, each forest gives a 35-dim class vector as the output considered the problem with 35 categories of products.

B. Multi-Grained Scanning

Multi-grained scanning in gcForest can effectively deal with the relationship between the features. Among them, sliding window is the most important part of multi-grained

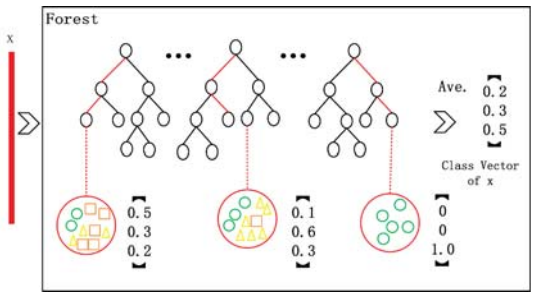

Fig. 1: Decision forest.

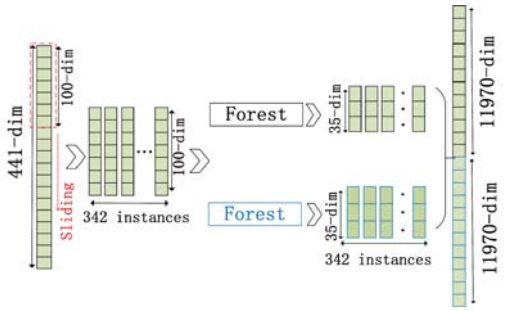

Fig. 2: Multi-grained scanning with a completely random forest (black) and a random forest (blue).

scanning, which can realize feature's reuse through multi-scale sliding windows, and generate new feature vectors with higher dimensions for the sample.

In the paper, each sample is presented in the form of sequential data, and three kinds of sliding windows are used to scan the original features in training data set, with window sizes of 100, 200 and 300 dimensions respectively. As shown in Fig. 2, each sample contains the original data of 441 dimensional feature vector $\boldsymbol{a_n}$. After scanning by a 100-dim sliding window, 342 100-dim feature sub-mappings are generated. All feature submaps access 1 completely random forest and 1 random forest to generate class vectors, as shown in Fig. 2. Then, the output 684 class vectors, each vector has 35-dim for 35 categories of products, from 2 forests are integrated as a new 23940-dim feature vector $\boldsymbol{x_n}$, which is the result of the multi-grained scanning of the 100-dim sliding window. Similarly, after the samples are scanned by 200-dim and 300-dim sliding windows, one 16940-dim feature vector $\boldsymbol{y_n}$ and the other 9940-dim feature vector $\boldsymbol{z_n}$ are generated, respectively in Fig. 3. Then a original sample $\boldsymbol{a_n}$ with 441 features is eventually converted to three higher dimension feature vectors, i.e., $\boldsymbol{x_n}$, $\boldsymbol{y_n}$ and $\boldsymbol{z_n}$, with a total of 50820-dim features.

C. Cascade Forest

In gcForest, the original input feature vector of cascade forest comes from the output of multi-grained scanning. Cascade forest of gcForest owns multiple levels, like deep

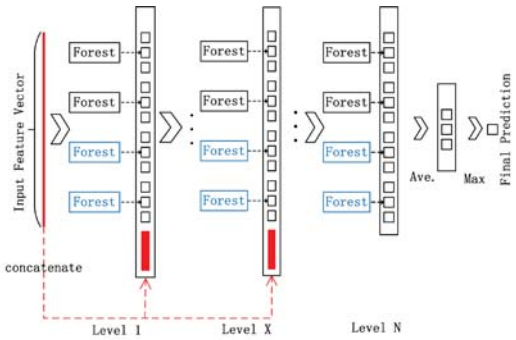

Fig. 3: Cascade forest with two completely random forests and two random forests in each level.

neural networks. In the paper, each level of cascade forest is composed of two completely random forests and two random forests in Fig. 3. In our considered problem, each forest produces a 35-dim class vector. The input of the next level forms by concatenating the original input feature vector to the back of the the four class vectors. Then, the number of levels keeps growing until a certain level triggers the termination condition, that the desired accuracy is reached or the maximum number of levels is reached. Finally, gcForest calculates the average value of the four class vectors in the last level, and select the class with maximum value as the classification result. Specifically, this concatenation mechanism can realize the cross-level process of the sample feature vector in gcForest method.

D. Overall Structure of GcForest

In Fig. 4, we summarize the overall structure of gcForest, including multi-grained scanning and cascade forest. Each sample feature vector a_n is processed by multi-grained scanning to generate the new high dimensional vectors, i.e., $\boldsymbol{x_n}$, $\boldsymbol{y_n}$, and $\boldsymbol{z_n}$. Then, $\boldsymbol{x_n}$, $\boldsymbol{y_n}$, and $\boldsymbol{z_n}$ respectively access the associated levels to participate in the classification of each level, using concatenation mechanism of cascade forest. Finally, gcForest generates a class number as the final prediction in the cascade forest, which is the category of products.

IV. Experimental Results

We provide the gcForest hyper-parameter setting for the product classification, through which the classification effect of gcForest model and other traditional model are compared and analyzed, including SVM with RBF kernel, SVM with linear kernel and CNN.

A. Hyper-Parameters Adjustment for GcForest

GcForest requires fewer hyper-parameters adjustment and owns a simpler structure compared with the neural network algorithm. In particular, it is not sensitive to the hyper-parameter setting, then satisfactory performance can be achieved with the same hyper-parameter setting even on different type of data sets. In addition, gcForest has no strict requirements on the size of the data sets, even on the small data sets, as we consider the problem. In Table I, the gcForest's hyper-parameters used in product classification are shown. The model hyper-parameters include 2 completely random forests and 2 random forests, each owning 500 decision trees, 3 sliding Windows in three different dimensions, i.e., 100-dim, 200-dim and 300-dim, and the terminal condition of tree growth in the forests.

TABLE I: Hyper-parameters in the gcForest for classification

Type	Name	Value
Multi-grained scanning	Number of forest	2
	Type of forest	Completely random forest and random forest
	Terminal condition of tree growth	Till pure leaf or reach depth 10
	Sliding window size	100-dim, 200-dim and 300-dim
Cascade forest	Number of forest	4
	Maximum levels	100
	Type of forest	Completely random forest and random forest
	Number of trees in each forest	500
	Class vector size	35
	Terminal condition of tree growth	Till pure leaf

B. Experimental Verification

The experiments are based on a feature vector data set with 4000 samples including 35 product categories, which is preprocessed from text-based product information from the real e-commerce platform. We randomly take 80% of the feature vector data set as the training set, and the remaining data as the test set. We use gcForest model to compare as traditional ML-based classification model, i.e., SVM with RBF kernel, SVM with linear kernel and CNN. Table II shows the comparison results of these classification models in the model training and test experiments. The accuracy of SVM with RBF kernel is 86.88%, and the accuracy of SVM with linear kernel is 89.73%, and the accuracy of CNN is 86.86%, and gcForest has the highest accuracy of 92.38%.

The gcForest classification model, that we propose, shows the obvious performance advantages compared with the traditional ML-based classification models for multi-classification problems with small-scale data samples problem, as we can see the experiment results. The multi-grained scanning of gcforest can effectively solve sequential text attribution relationship problem. In addition, sample feature vectors can be linked across multi-level in cascade forest makes training more accurate. Then gcForest can train a model with better classification performance using a small number of samples.

C. Experimental analysis

As shown in TABLE III, the classification accuracy of 3 levels are respectively obtained. The accuracy of large class

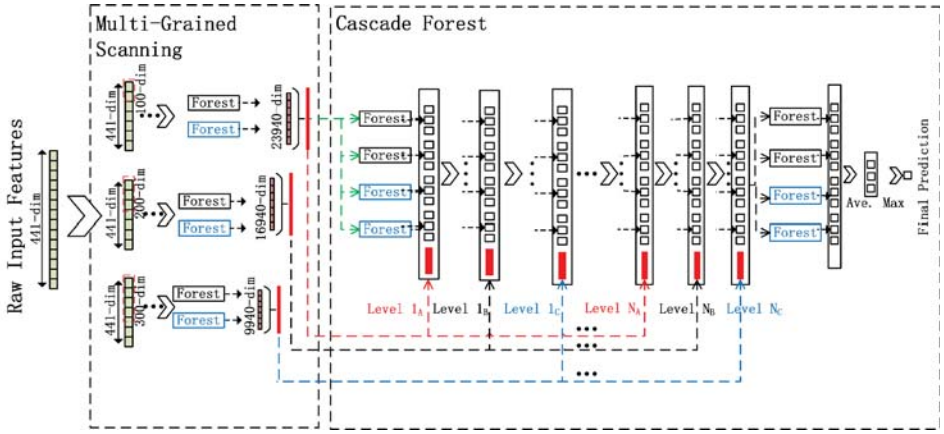

Fig. 4: Construction of gcForest.

TABLE II: Performance comparison of different methods

Method	Accuracy
SVM with RBF kernel	86.88%
SVM with linear kernel	89.73%
CNN	86.86%
GcForest	92.38%

TABLE III: Comparison of accuracy between different categories

Type	Accuracy
Large class	99.79%
Medium class	95.33%
Small class	92.38%

is 99.79%, the accuracy of medium class is 95.33% and the accuracy of small class is 92.38%. We can find that the classification of large categories is better than the classification of medium class and small class, which is close to 100%. This is due to the high degree of differentiation between the figure vectors of samples owned different large categories, which is helpful to classification and recognition. The similarity of goods in the middle category or the small increase significantly, then the classification accuracy decreases. The excellent performance in large class makes us believe that the gcForest model can replace the manual classification.

V. CONCLUSION

In this paper, we utilize gcForest model to a product classifier for text-based title information classification problem. We propose a preprocessing method for the original text data from the real e-commerce platform, and the advice of the hyper-parameters setting in gcForest. In the experiments, the classification accuracy using gcForest is 92.38%, which outperforms SVM with RBF kernel (86.88%), SVM with linear kernel (89.73%) and CNN (86.86%). Gcforest has shown great potential in solving e-commerce product classification problem that has a large number of categories and comparatively small scale data samples. We believe that this method can replace a large amount of manual work in predicting e-commerce product forms.

REFERENCES

[1] Z. Kozareva, "Everyone likes shopping! multi-class product categorization for e-commerce," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1329–1333, May 2015.

[2] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif Intell Rev*, vol. 52, pp. 273–292, June 2019.

[3] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artif Intell Rev*, vol. 35, pp. 137–154, Feb. 2011.

[4] C. S. Jothi and D. Thenmozhi, "Machine learning approach to document classification using concept based features," *International Journal of Computer Applications*, vol. 118, May 2015.

[5] D.-T. Vo and C.-Y. Ock, "Learning to classify short text from scientific documents using topic models with various types of knowledge," *Expert Syst Appl*, vol. 42, pp. 1684–1698, Feb. 2015.

[6] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *Int J Res Mark*, vol. 36, pp. 20–38, Mar. 2019.

[7] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE T Image Process*, vol. 25, pp. 2529–2541, Mar. 2016.

[8] S. Sudholt and G. A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *Proceedings of the IEEE International Conference on Frontiers in Handwriting Recognition 2016*, pp. 277–282, Oct. 2016.

[9] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng, and K. Li, "A parallel random forest algorithm for big data in a spark cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, pp. 919–933, Apr. 2017.

[10] Z.-H. Zhou and J. Feng, "Deep forest," *National Science Review*, vol. 6, pp. 74–86, Oct. 2018.

[11] M. Zhou, S. Zhang, Y. Qiu, H. Luo, and Z. Wu, "Entropy-based spammer detection," in *Proceedings of the ACM International Conference on Internet Multimedia Computing and Service 2018*, pp. 1–6, Aug. 2018.

[12] L. Han, Z. Haihong, Y. Erxin, B. Yuming, and L. Huiying, "A clothes classification method based on the gcforest," in *Proceedings of the IEEE International Conference on Image, Vision and Computing 2018*, pp. 429–432, Dec. 2018.

[13] H. Christian, M. P. Agus, and D. Suhartono, "Single document automatic text summarization using term frequency-inverse document frequency (tf-idf)," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 7, pp. 285–294, Dec. 2016.