IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 33, No. 1, February, 2019, pp. 15-19
Journal homepage: http://iieta.org/journals/ria

Knowledge Text Classification Based on Virtual Category Tree

Wei Zhao[1], Guangyu Wang[2], Bo Peng[1*]

[1] Jilin Police College, Jilin 130000, China
[2] Aviation University Air Force, Jilin 130000, China

Corresponding Author Email: 22135770@qq.com

https://doi.org/10.18280/ria.330103

Received: 15 December 2018
Accepted: 3 February 2019

Keywords:
knowledge text, classification, virtual category tree

ABSTRACT

The existing multi-layer classification methods often learn the sample data repeatedly with a top-down classification model. To solve the problem, this paper developed a multi-layer text classification method based on Virtual Category tree (VC tree). The classifier was designed in a bottom-up manner, aiming to reduce the repetition and time of sample learning. In the top-down classification, the similarity between the preprocessed document vector and the associated classifier was calculated, and the maximum similarity was selected to determine the category of the document, and then the document was directly attributed to the leaf node. The experimental results showed that the proposed method outperformed the support vector machine (SVM)-based classification and saved the time of text classification.

1. INTRODUCTION

Text Classification, or Text Categorization, is the process of classifying a given text into one or more predetermined text categories according to its content. Text classification is a typical supervised learning process. According to the text set that have been marked, a relationship model between text feature and text category is obtained through learning, and then applied for classifying new texts. As a key technology for processing and organizing a large amount of text data, text classification can solve the problem of information clutter to a large extent. It's convenient for users to accurately locate and shunt the required information, and it is of great practical significance for high-efficient information management and utilization.

In recent years, many scholars at home and abroad have conducted in-depth research on knowledge text classification models and algorithms, and have achieved certain results. Sweeney et al. [1] proposed the classical k-anonymity model, which requires each tuple in the data table after data release to correspond to at least k indistinguishable tuples, making it impossible for attackers to infer the specific information in the anonymous post-tuples, thereby resisting the linkage attacks. By producing a certain number of indistinguishable individuals, it makes the attackers unable to distinguish the individual to which the private information belongs, thereby preventing the linkage attacks. Many scholars have conducted researches on data privacy and usability problems from different aspects based on the classic K-anonymity model.

Literature [2] further elaborated the k-anonymity model and proposed a cluster-based k-anonymity algorithm; Benjamin CM Fung et al. [3] defined information entropy to measure the balance between privacy and information, and proposed a top-down TDS generalization algorithm, which can both generalize the discrete attributes and the continuous attributes, and process the redundant data in each iteration of generalization operations, retain the data information, thereby guiding the classification information while meeting the privacy requirements. Xu et al. [4] conducted local generalization on the attributes of quasi-identifiers and proposed two anonymous algorithms to ensure data availability. Based on data availability, Shen Yanguang et al. [6] adopted a security multi-party computation method to propose a distributed PPC4.5 classification decision-tree algorithm for privacy protection. Based on the classification decision-tree model, literatures [7] and [8] respectively proposed improved singular value decomposition and multi-dimensional suppression idea to instruct anonymous processing. Zhao [9] and Yang [10] both conditionally constrain the sensitive attributes, and respectively proposed a sensitivity-based personalized (a, l)-anonymous model and a micro-aggregation algorithm based on sensitive attribute entropy. Literature [11] proposed an anonymous model based on classification utility, which determines the attribute maximum classification capacity by calculating the mutual information of each quasi-identifier attribute, targeting attribute classification capacity rather than privacy requirement, the method ultimately obtained accurate classification of anonymous data. Literature [12] proposed a privacy protection method considering attribute weights, which introduced a weight generalization path to the attributes of quasi-identifiers, and restrained the utility difference under different specific data application and analysis occasions.

Based on the research of existing text classification methods and incremental learning algorithms, this paper proposes a multi-layer text classification method based on VC tree. Aiming at repeated sample data learning problem of the existing multi-layer classification methods which often use the top-down classification model, this paper proposes to use a bottom-up approach to construct the classifiers, so as to reduce the cost and time of repeated sample learning. Furthermore, combined with the actual application requirements, it designs an incremental learning algorithm for multi-layer classification method.

2. CLASSIFICATION MODEL CONSTRUCTION

The learning process of text classification can use machine learning algorithms and probability statistics to mine the associations between documents and categories in the learning data set and construct related classification models. For the multi-layer text classification method 106.101.109110114.148 mentioned earlier, the classification models mainly include the category tree and the directed acyclic graph (DAG). The former organizes the categories of text into a tree-shaped structure, and the classification process classify the categories in turn according to two basic patterns: top-down and around-graph; the latter chooses the graph method to query all the document category information involved in DAG. The text classification model constructed in this paper selects the VC tree, in which the leaf nodes represent all classification categories, the details are shown in the following figure.

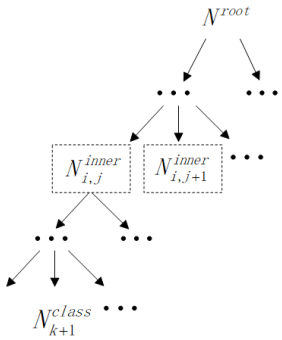

Figure 1. Category tree

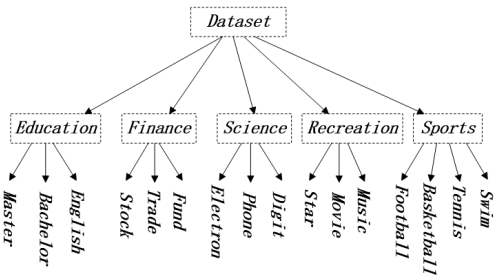

Figure 2. VC tree

Definition 1: VC tree, is defined as a four-tuple $(N^{root},\{N_{ij}^{inner}\},\{N_k^{class}\},\{(N_f,N_s)\})$, wherein, N^{root} is a virtual category formed by all categories, among all nodes, it belongs to the parent category; N_{ij}^{inner} represents to add a j-th virtual category with depth i in the VC tree, and the leaf nodes do not include the root node; N_k^{class} represents the k -th real category, which contains the leaf nodes of the VC tree; (N_f,N_s) describes the parent-child relationship between the nodes, it is a branch of the VC tree, wherein $N_f \in \{N^{root}\}\cup\{N_{ij}^{inner}\}$,$N_f \in \{N_{ij}^{inner}\}\cup\{N_k^{class}\}$.

An example of VC tree is shown as Figure 2, there are three types of nodes:

(1) Root node: $Dataset$ represents the virtual category to which all documents belong;

(2) Inner nodes: {Education, Finance, Science, Recreation, Sports} is a manually-introduced virtual category set;

(3) Category nodes: {Master, Bachelor, English, Stock, Trade, Fund, Electron, Phone, Digit, Star, Movie, Music, Football, Basketball, Tennis, Swim} is the actual category set to which the documents will be assigned.

Whether the text classification model is reasonable or not will directly affect the subsequent classification results. The multi-layer classification method proposed in this paper is based on the learning samples and each layer of the hierarchical directory structure, all of which are classified by the binary classifier onevs. brothers, which can only be obtained through the learning of positive and negative example documents. Examples that play an important role in this category are positive examples, and those that cannot produce any value for this category are considered as negative examples. Generally, relevant documents of the category are selected as positive examples of the binary classifier of the category, and negative examples contain a lot of documents that are not involved in the category. For virtual categories and real categories, a binary classifier is constructed to determine whether a document belongs to a corresponding category. Definitions of commonly used concepts in text classification are given below:

Definition 2: Feature Word FW is defined as a five-tuple: (ID,TX,TF,DF,WT), $TX \in Dict$; $Dict$ represents the set of word entries in a dictionary; $ID = Dict(TX) \in$; $Dict(TX)$ represents the identifier of the word TXin the dictionary; $TF = (tf_1,tf_2,\dots,tf_k), tf_i = \sum_{d_j \in N_i^{class}} \left|TX_{d_j}\right|$, TF represents the frequency of the word TX in the training document set of each category, that is, the number of occurrences in the document set, K represents the number of actual categories; $DF = (df_1,df_2,\dots,df_K)$, $df_i = \left|\{d_j | d_j \in N_t^{class} \cap TX \in d_j\}\right|$, DF represents the document frequency of the word TX in the training document set of each category, that is, the number of documents in which TX appears in the document feature words; $WT = Dictf(TX)$; $Dictf(TX)$ represents the word frequency factor of the word TX in the dictionary.

Definition 3: Feature space, F_{space} is defined as a set of key feature words $\{fw_i^{key} | fw_i^{key} : FW\} \subseteq \{fw_j | fw_j : FW\}$; the key feature words are selected from the feature word set of the training document set by method such as using information gain to calculate the weights.

Definition 4: Feature word set of the document$\boldsymbol{d_j}$, assuming that the set can also be represented by $s^{d_j} = \{(fw_i,v_i) | fw_i tx \in d_j$, wherein, $v_i = fw_i tf.log(|D^{train}\backslash |fw_{i.df}); fw_i : FW\}$, V_i represents the weight of the feature word fw_i in document d_j calculated by tf.idf.

Definition 5: Sample set D 's feature word set $S^D = \{S^{d_f} | d_i \in D\}$.

Definition 6: Document d_j 's eigenvector $fv^{d_j} = (v_1,v_2,\dots\dots,v_k), (fw_1 v_i) \in S^{d_j} \wedge fw_i \in F^{space}$. It represents the vector consisted by the weight of the feature word set of the document d_j mapped to the feature space FPU.

Definition 7: Document d_j 's category document set$D = \{x | x \in D^{train} \wedge Parent(x) = Parent(d_j)\}$ ($Parent(x)$ represents the directory node where the document x is located), D^{train} represents the entire sample set that is used for learning. d_j category sample set represents a collection of documents that have the same category with d_j.

Definition 8: Node N's positive example set $+ve^N = \underset{Parent(N_I)=N}{U} + ve^{N_i}$ represents the union of all the positive example sets of its child nodes.

Definition 9: Node N's negative example set $+ve^N = \bigcup_{Parent(N_I)=N} + ve^{N_i}$ represents the union of all the negative example sets of its child nodes.

3. CLASSIFIER CONSTRUCTION

The multi-layer text classification method based on VC tree is mainly to construct a virtual classification tree in the learning stage, and construct binary classifiers from the bottom through the learning process of the sample set to which it belongs. The classifier construction method is shown as Algorithm 1.

Time complexity analysis of the classifier construction algorithm based on VC tree: the training text set contains n texts, the number of categories is k; the construction of algorithm is divided into two stages: category construction and classifier construction. The previous stage is a process of constructing a category tree according to the directory hierarchy, and the category tree can be regarded as a binary tree structure, the traversal time complexity is the same as the hierarchical traversal time complexity of the binary tree. According to the nature of the binary tree, the maximum number of virtual nodes is $k-1$, the total node number is $k+k-1$, and the time complexity is $O(2k-1)$; in the classifier construction stage, the time spent include the subsequent traversal time of the category tree and the text processing time, each text only needs to be scanned , and its time complexity is $O(2k-1+n)$. Therefore, the total time complexity of Algorithm 1 is $O(n+4k-1)$, since k is much smaller than the text number n, so we can get that the algorithm's time complexity is $O(n)$ level.

Algorithm 1. Classifier construction algorithm based on VC tree

Input: Sample document set D^{train} for the hierarchical directory structure

Output: VC tree, each non-root node contains a binary classifier, the actual category number *K*

Algorithm steps:

Step 1. Construct VC tree root-node N^{root} for the directory of learning document set, the category sequence number t=0;

Step2. Conduct breadth-first traverse starting from the root directory of the learning document set:

a) For the j-th directory with the depth of the i layer: i. If the directory contains subdirectories, then create corresponding VC tree node N_{ij}^{inner} and take it as the child node of the newly created node with the depth of i layer; if the directory doesn't contain any subdirectory, $t=t+1$, then create the corresponding VC tree leaf node N_t^{class} and take it as the child node of the newly created node with the depth of $i-1$ layer;

b) When document d is read, perform preprocessing operations such as Chinese word segmentation on each sample in the category document set of d, and for the union set of text vectors related to all documents $d_f \in \bigcup_{i} N_i^{daxt} fv^{d_j}$, take it as the child node of its upper layer node N_t^{class}.

Step 3. The number of actual classification categories K=t;

Step 4. Create corresponding binary classifiers for all nodes on the VC tree with document vector, and conduct post-order traversal on each node:

a) If the current node is N^{root}, skip directly;

b) If the current node is N_k^{vector}, create a virtual binary classifier for it and each corresponding document eigenvector fv^{d_j} is calculated and taken as the positive sample set, ∅ is taken as the negative sample set;

c) For the remaining nodes N, the positive sample set $+ve^N$ is taken as a positive example, and the negative sample set $-ve^N$ is taken as a negative example for learning the corresponding binary classifier.

Step 5. For VC tree with document vector, cut off all the child nodes of N^{vector} type, and the algorithm ends.

After the multi-layer classification learning process, a VC tree model of multi-layer text classification method is constructed. Each leaf node corresponds to a real category, and each inner node corresponds to a virtual category.

4. TEXT AUTOMATIC CLASSIFICATION

This paper adopts the VC tree model to conduct automatic classification, starting from the root node, and the top-down operation pattern is chosen; then calculate the correlation between the document vector to be operated and the classifiers of different attributes in each layer, for the category to which the document belongs, select the maximum value, and then the documents are integrated into leaf nodes.

Time complexity analysis of top-down text automatic classification algorithm: the text set to be classified contains n texts, and the category tree leaf nodes have k categories. For each text, the classification process is the layer-by-layer traversal process of the category tree, then the time complexity of Algorithm 2 is $n*O(2k-1)$, that is $O(n*(2k-1))$ level, since k is much smaller than n the number of texts to be classified, so the algorithm's time complexity is $O(n)$ level.

Algorithm 2. Top-down automatic text classification

Input: D^{test} the set of documents to be classified, VC tree model

Output: D^{result}, the document set with hierarchical directory structure

Algorithm steps:

Step 1. Create a directory structure according to the hierarchical structure of VC tree;

Step 2. Read an unclassified document d_j from D^{test} the document set to be classified, perform Chinese word segmentation, feature dimension reduction, vector representation, and other pre-processing procedures, and calculate its eigenvector fv^{d_j};

Step 3. Starting from the root node in the VC tree model, $N^{temp}=N^{root}$, traverse down layer by layer:

a) If $N^{teme} \in \{N_i^{class} | 1<i<K$, $N^{teme} \in \{N_i^{class} | 1<i<K$, that is, classify till the leaf nodes, go to the next step;

b) Otherwise, calculate the similarity of all child nodes between d^f and N^{teme} and take its maximum value MAX($\underset{Parent(N_t)=N^{temp}}{SIM}(N^{teme}, N_K)$. The similarity can be calculated by L-S distance, cosine distance, kernel function and other methods. At last, modify the value of N^{teme} to N_K^{temp}, and the corresponding category is the category to which document d belongs, mark d_j as N_K^{temp}.

Step 4. The corresponding category of $N^{teme}$is the category to which document d_j belongs, mark d_j as N^{teme}.

Step 5. If there are still unmarked documents in the set of documents to be classified, return to Step 2; otherwise, copy all the marked documents into the corresponding directory structure.

5. EXPERIMENT SIMULATION

The experimental data are all from NEDS, obtained from the upper three layers, the collection method is relatively random. The data content includes 3288 documents, all of which were integrated effectively and a comprehensive data set was obtained. According to the semantic structure characteristics webpages, the pre-processing extracted the text using the text extraction tool, and then the Chinese word data attributes were classified by using the Chinese word segmentation system ICTCLAS developed by the Chinese Academy of Sciences. The system adopted the SVM-light classifier, which was designed and developed by Joachims.

The experiment adopted the classification model proposed in this paper, and screened out the hierarchical structure of the document set and the text segmentation processing data, then with the help of the multi-layer SVM classifier of VC tree, the data type of the text was classified automatically. In the classification results, the misclassified single-document can be dragged through the interface into the correct category so as to trigger the single-document incremental learning behavior of the system; in the system functions, the system's sample set incremental learning behavior can be triggered by specifying incremental sample set with the same directory structure as the historical sample set. Through above method, correction and reconstruction of the classification model were completed, so that the classification results are more accurate and effective. The functions of each functional module of the system had been developed, and the sample set had been tested, showing good performance and accuracy.

Compare the experimental results of multi-layer classification and single-layer classification. The single-layer classifier implemented by the traditional SVM was compared with the multi-layer classification based on VC tree. The learning set selected 50 texts*16 category documents, and the test set selected 2488 other documents. The evaluation criteria for single-layer classification only adopted the standard accuracy, the recall rate and the F1 value.

It can be seen from the data in Table 1 that the multi-layer classification method proposed in this paper showed better performance than the planar classification in classifying categories with "close-distant" differences. Except for English and Swim, the accuracy and recall rate of the other categories had increased to varying degrees. The F1 value of the standard recall rate and accuracy increased from 0.9413 to 0.9502. Figure 3 shows the comparison curve of classification result F1 values of the multi-layer classification and single-layer classification, except for some categories, the F1 values of other multi-layer classification had better performance than the planar classification, and its overall performance is better than the planar classification.

Table 1. Classification effect comparison and evaluation

Classification	number of documents		Plane classification			Multi-layer classification		
	Training	Test	recall rate	Accuracy	F_1	recall rate	Accuracy	F_1
Stock	50	163	0.9448	0.9625	0.9536	0.9571	0.9630	0.9600
Fund	50	155	0.9677	0.9615	0.9646	0.9677	0.9615	0.9646
Trade	50	158	0.9557	0.9264	0.9408	0.9557	0.9679	0.9618
Bachelor	50	150	0.8200	0.9840	0.8945	0.8267	0.9764	0.8953
Master	50	153	0.9673	0.8970	0.9308	0.9739	0.9198	0.9460
English	50	194	0.9897	0.9320	0.9600	0.9845	0.9272	0.9550
Electron	50	152	0.9737	0.9548	0.9642	0.9868	0.9677	0.9772
Phone	50	153	0.9608	0.9608	0.9608	0.9804	0.9615	0.9709
Digit	50	150	0.9133	0.9648	0.9384	0.9333	0.9655	0.9492
Basketball	50	156	0.9679	0.9618	0.9649	0.9744	0.9560	0.9651
Tennis	50	153	0.9150	0.9722	0.9428	0.9281	0.9726	0.9498
Swim	50	148	0.9730	0.9664	0.9697	0.9662	0.9662	0.9662
Football	50	154	0.610	0.9193	0.9397	0.9545	0.9304	0.9423
Movie	50	150	0.9533	0.9108	0.9316	0.9533	0.9408	0.9470
Star	50	150	0.8467	0.8819	0.8639	0.9067	0.8889	0.8977
Music	50	149	0.9463	0.9338	0.9400	0.9396	0.9524	0.9459
average			0.9400	0.9431	0.9413	0.9493	0.9511	0.9502

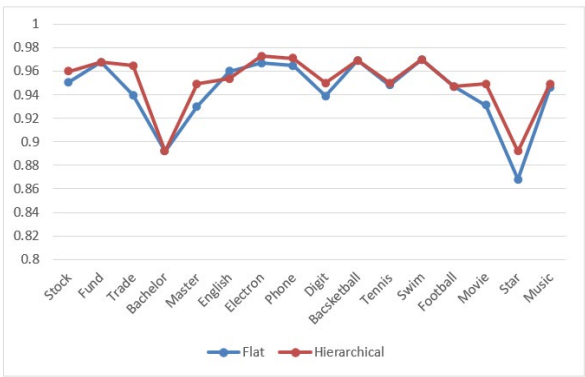

Figure 3. F1 result comparison of classification experiment

6. CONCLUSION

The existing multi-layer classification methods often learn the sample data repeatedly with a top-down classification model. To solve the problem, this paper developed a multi-layer text classification method based on virtual classification tree. The classifier was designed in a bottom-up manner, aiming to reduce the repetition and time of sample learning. In the top-down classification, the similarity between the preprocessed document vector and the associated classifier was calculated, and the maximum similarity was selected to determine the category of the document, and then the document was directly attributed to the leaf node. Experiments showed that the multi-layer text classification method based

on VC tree had better performance than SVM classification and reduced the time of text classification.

ACKNOWLEDGMENT

This work was supported in part by Jilin Province Education Department "13th Five-Year" Science and Technology Project No.2016558.

REFERENCES

[1] Sweene, Y.L. (2002). K-anonymity: A model for protecting privacy. International Journal on Uncertainty Fuzziness and Knowledge based Systems, 10(5): 571-578. https://doi.org/10.1142/S0218488502001648

[2] Aggarwal, G., Feder, T., Kenthapadi, K., Khuller, S., Panigrahy, R., Thomas, D., Zhu, A. (2010). Achieving anonymity via clustering. ACM Transactions on Algorithms, 6(3): 1-19. https://doi.org/10.1145/1798596.1798602

[3] Fung, B.C.M., Wang, K., Yu, P.S. (2005). Top-down specialization for information and privacy preservation. Proceedings of the 21st IEEE International Conference on Data Engineering(ICDE2005), pp. 205-216. https://doi.org/ 10.1109/ICDE.2005.143

[4] Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W. (2006). Utility-based anonymization using local recoding. Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (SIGKDD). Philadelphia, PA, USA, pp. 785-790. https://doi.org/10.1145/1150402.1150504

[5] Li, T.C., Li, N.H. (2009). On the tradeoff between privacy and utility in Data publishing. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, pp. 517-525. https://doi.org/ 10.1145/1557019.1557079

[6] Shen, Y.G., Shao, H., Zhang, Y.Q. (2010). Research on privacy preserving distributed decision-tree classification algorithm. Application Research of Computers, 27(8): 3070-3072. https://doi.org/10.3969/j.issn.1001-3695. 2010.08.069

[7] Li, G., Xi, M. (2012). An improved privacy-preserving classification mining method based on singular value decomposition. ACTA Electronica Sinica, 40(4): 739-744. https://doi.org/10.3969/j.issn.0372-2112.2012.04.019

[8] Kisilevich, S., Rokach, L., Elovici, Y. (2010). Efficient multidimensional suppression for K-anonymity. IEEE Transactions on Knowledge and Data Engineering, 22(3): 334-347. https://doi.org/10.1109/TKDE.2009.91

[9] Zhao, S., Chen, L. (2015). Personalized(a,l)-anonymity method based on sensitivity. Computer Engineering, 41(1): 115-120. https://doi.org/10.3969/j.issn.1000-3428.2015.01.021

[10] Yang, J., Wang, C., Zhang, J.P. (2014). Micro-aggregation algorithm based on sensitive attribute entropy. ACTA Electronica Sinica, 42(7): 1327-1337. https://doi.org/10.3969/j.issn.0372-2112.2014.07.013

[11] Li, J.Y., Liu, J.X., Bai, G.M. (2011). Information based data anonymization for classification utility. IEEE Transactions on Knowle Dgeand Data Engineering, 70(12): 1030-1045. https://doi.org/10.1016/j.datak.2011.07.001

[12] Xu, Y., Qin, X.L., Yang, Y.T., Yang, Z.X., Huang, C. (2012). A QI weight-aware approach to privacy preserving publishing data set. Journal of Computer Research and Development, 49(5): 913-924.