5. Wei-Tek T., Peide Z., Balasooriya J. et al., An Approach for Service Composition and Testing for Cloud Computing, in 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS), 2011: 631-636.
6. Babcock C., Management strategies for the cloud revolution: McGraw-Hill, USA, 2010.
7. Zou G., Chen Y., Yang Y. et al., AI Palming and Combinatorial Optimization for Web Service Composition in Cloud Computing, in proceedings of International Conference on Cloud Computing and Virtualization (CCV-10), 2010: 28-36.
8. Calheiros R. N., Buyya R., De Rose C. A. F., Building an automated and self-configurable emulation test bed for applications, Software: Practice and Experience, 2010, 40(S): 405-429.
9. Hurwitz J. D., Cloud computing for dummies: Wiley Pub., Inc., Indianapolis, IN, 2009.
10. Hoffa C., Mehta G., Freeman T. et al., On the Use of Cloud Computing for Scientific Workflows, in IEEE Fourth International Conference on e-Science, Indiana Univ.. IN. USA. 2008: 640-645.

Random Forest based Online Topic Detection using Topic Graph Cluster

Qian Chen[1,2] , Zhiguo Gui[1,3,4,*] , Xin Guo[2] , Yang Xiang[5]

1 School of Information and Communication Engineering,
North University of China, Taiyuan, Shanxi,030051, China
2 School of Computer and Information Technology, Shanxi University,
Taiyuan, Shanxi, 030006, China
3 Key Laboratory of Instrumentation Science and Dynamic Measurement,
North University of China, Taiyuan, Shanxi, 030051, China
4 National Key laboratory for Electronic Measurement Technology, Taiyuan,030051, Shanxi, China
5 School of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China
** Corresponding author: gui_zg@163.com*

Abstract
We proposed an online topic detection approach using random forest based on topic graph cluster which models a topic in the form of graph comprised of terms and the edges among terms. The topic graph structure can largely enhance the semantic information hidden in the corpus, thus avoided the shortcoming of bag-of-words. Random

Forest was used to simplify the online topic detection process in a considerate way thus gain low complexity in terms of time and space. Our approach can link two different corpuses, and investigate the linkage among topics between two corpuses. Furthermore, RF-OTD can detect outliers and novelty in text stream by computing the proximity of two topic graph. Experimental results showed that, compared to baseline topic detection algorithm, our approach gain better performance in computational efficiency, consistency and semantic explanation.
Keywords: ONLINE TOPIC DETECTION, TOPIC GRAPH, RANDOM FOREST

1. Introduction

Topic detection is a fundamental task in the field of information retrieval especially public opinion monitoring which is a fully unsupervised learning task with no prior knowledge[1]. How to represent topic without loss of semantic and detect the topic hidden in text collection is a challenging problem, and topic was frequently used in user modeling, especially emotional topic[2]. With the development of web2.0, text data grows exponentially and comes as stream in social media, such as twitter[3] and blogs[4], which poses new challenge in topic detection[5]. Existed work based on term co-occurrence frequency and probabilistic topic model treats topics as a distribution over term space, which overlooks the semantic information hidden in topic[6]. Most previous studies have focused on how to analyze text corpus rather than text stream. Text data's multi-dimension becomes extremely common, and we need to process topic detection in real time. The complexity of traditional algorithm made it loss their efficiency when facing high-speed text stream. Thus there are three challenges in total in topic detection for text stream. Firstly, traditional algorithm such as topic model and variational inference or sampling methods is based on probabilistic statistics, which ignore the semantic information hidden in corpus, thus a novel semantic based topic representation is needed. Secondly, text stream comes in bulk, and traditional method suffers higher time and space complexity, thus online topic detection with simple algorithm is strongly needed. Lastly, traditional topic detection primarily based on slide window technology, and therefore loss the correlativity among topics in different time slice. Traditional detection methods are performed on time slice, and topics between time slice corpuses can be hard to identify. There existed work LDA, OLDA and PAM[6-8] which treat this problem well, but fail to perform in log-linear complexity, especially in text stream.

The basic idea in this paper is: first, topic modeling with term graph, secondly, extract topic in one slice of text stream using topic graph. Thirdly, Random Forest is used to detection topic online. This paper is organized as follows: In section 2, we gave a brief introduction and background about topic modeling with topic graph as well as random forest. We built topics using a network called topic graph where topics were represented as concept nodes and their weight edge among them to avoid semantic loss. In order to gain lower time and space complexity we choose random forest to construct online topic detection, which can also detect the correlations among topics in different time slice in section 3. Section 4 and section 5 is followed by giving an experiment and the result compared to the baseline topic detection.

2. Background

In order to reduce the complexity of topic detection, we need to partition text stream into multiple sub-collections, for online text stream, one can partition data according to time slices, each of which is a fixed time slot. For large text stream, it can be divided by a fixed number N, in which N pieces of text is a sub-collection. Our topic graph based online detection construct topic graph cluster for each sub-collection, and topics in each collection is obtained in units of topic graph. With the arrival of text sub-collection containing N text, an updating operation is performed for current topic graph cluster and previous topic graph cluster. Topic graph cluster for the whole text collection was updated and generated.

2.1. topic graph

Formally, we take the concept and relation between concepts into account. This can be done by using an adjacent matrix over term space in which the relation weight over any two concept nodes implies the closeness strength. The basis of building a topic network from corpus is vector space model. Given a corpus, we can build a term-document matrix, and each column is a vector representing a document using *TF-IDF*.

After that, *term-document* matrix is decomposed and then transformed into a *term-term* model using Equation (1).

$$t_{ij} = \begin{cases} 0 & \text{if } i = j \\ \sum_{k=1}^{|A|} \min(w_{ik}, w_{jk}) & \text{if } i \neq j \end{cases} \quad (1)$$

Finally, we set a threshold to get rid of edges of which weight was less than that threshold, and an edge between the term pair with non-zero weight can

be created. We can use this graph to build topic by segment the whole graph into several non-connected graphs. It's clear that larger the threshold is and fewer topics we obtained.

2.2. Random Forest

Combination model is a research hotspot in recent years; one can gain a strong model by combining multiple relatively simple models. Random forests is first put forward by Leo Breiman [9], a classic instance of model combined with decision tree, which is mainly used for classification and regression. We choose random forest for our classification algorithm since its high precise, efficiency, low complexity and simplicity, and it has the properties that no variable selection is needed, and it can deal with large data.

For classification, the idea of random forests are randomly constructing multiple decision trees to form a forest, and the decision trees are considered to be independent identically distributed. The class label $y = \{C_1, C_2, ...\}$ is obtained by each decision tree via samples, i.e., each tree vote for the samples, then samples we are predicted for the category that gets the most votes. The judgment to vote made by each tree is independent. This process is similar to the judges scoring mechanism in a game or election process[10]. Although the ability of each tree or the perspective of each judge is limited, when all the power of decision tree gathered together, the forest becomes powerful, thus can get a more suitable decision.

3. Random Forest based topic detection using topic graph

After introducing topic graph and random forest, we try to maintain semantic relation in order to discover new semantic structure and relation base on original topic graph structure and the relation among terms. We choose RF for two reasons, one is the inherent properties of RF, and the other is for incremental process.

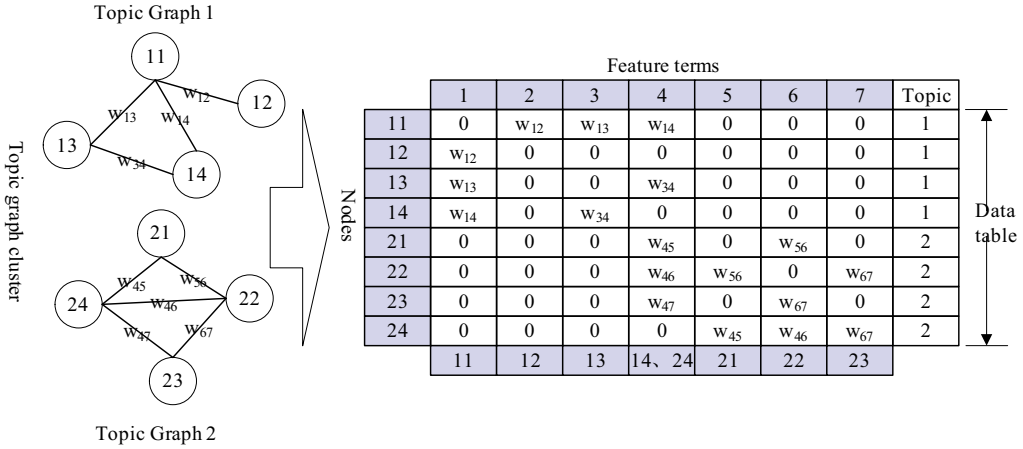
3.1. model training

The training process of Random forest can be done in several steps.

(1) Transforming topic graph cluster into data table. For convenience, topic graph in sub-collection need to be represented as sheets of records, all of which form table with size of $M*(|V|+1)$, where $|V|$ is the size of selected term vocabulary. Each row stands for a node in a topic graph, where the value of each element is the weight between current node and the column node. The last column denotes the topic ID in which the current node belongs to. That is to say, this model classifies a node according to the relationship between current node and other node in the same topic graph. We take these data as our training set, which is used to make prediction via random forests. Since the same word node in topic graph cluster may be in different topic graph, column size should be greater than or equal to the size of row. The mapping from the topic graph cluster to data table is shown in Figure 1.

(2) Random sampling. This step is to build data set for each tree in random forest, and here we choose bagging algorithm[11] which is a relatively simple model combination algorithm, and can increase the stability and accuracy of classification or regression. For dataset D with size of n, we select n' samples by randomly sampling with replacement. There can be duplicate data for sampling with replacement. When data quantity is big enough, the remaining data is about one third of the total amount[12], which is used for out-of-bag (OOB) error estimation. Random sampling and repeat exceedingly reduce over fitting.

(3) Building classification tree. Decision tree can be divided in to two type, classification and regression tree. For the value to be predicted is discrete, we will build classification tree. Each tree is trained by evaluating the gini impurity of input variable in terms



	1	2	3	4	5	6	7	Topic
11	0	w_{12}	w_{13}	w_{14}	0	0	0	1
12	w_{12}	0	0	0	0	0	0	1
13	w_{13}	0	0	w_{34}	0	0	0	1
14	w_{14}	0	w_{34}	0	0	0	0	1
21	0	0	0	w_{45}	0	w_{56}	0	2
22	0	0	0	w_{46}	w_{56}	0	w_{67}	2
23	0	0	0	w_{47}	0	w_{67}	0	2
24	0	0	0	0	w_{45}	w_{46}	w_{67}	2
	11	12	13	14、24	21	22	23	

Figure 1. Mapping from topic graph cluster to data table.

of predicted variable according to each cycle of sampling data. All tree is combined into the whole forest. The detail is as follows.

We selected m out of $|V|$ feature terms in random, and computed the frequency these samples occurs for each possible composition $a_1=x_1$, $a_2=x_2$, ..., $a_m=x_m$, m dimensional column list was generated. For all samples, evaluate Gini impurity of each feature term in terms of class labeled y Gini(a_i)(i = 1, …, m). Feature term a_i of the minimal gini impurity is chosen as the first split variable. We use gini to select variable because the smaller gini is, the larger the probability that the consequent node sample come from the same category. Since the variables are real numbers, if variable is splitted according to real number, the classification tree branch data set is divided into small pieces, lead to overfitting. We divided node sample into two parts $\{S_1^j, S_2^j\}$, $x \geq x_j$ and $x<x_j$ respectively according to the possible value of a_i that it takes for each x_j (j= 1,…,t), and estimate gini impurity $\text{Gini}_j(a_i)$ for a_i in terms of x_j.

$$\text{Gini}_j(a_i) = \frac{S_1^j}{S}\text{Gini}(S_1^j) + \frac{S_2^j}{S}\text{Gini}(S_2^j)$$

$$= \frac{S_1^j}{S}\left(1-\sum_{y=1}^{Y}\left(\frac{S_{y1}^j}{S_1^j}\right)^2\right) + \frac{S_2^j}{S}\left(1-\sum_{y=1}^{Y}\left(\frac{S_{y2}^j}{S_2^j}\right)^2\right) \quad (2)$$

Thus we can get $j' = \underset{j}{\text{argmin}}\,\text{Gini}_j(a_i)$ that minimize gini impurity. In this method, the gini of variable a_i won't change with j', so all variable needs to traverse to get further spliting every time the split variable is selected, not only the variable that never be traversed.

(4) The out-of-bag error estimates. When build each classification tree, about one third records have not been selected as the training set, and this part of the record is used as OOB error estimates. Category vote result is obtained by trained classification tree for remaining records. For each sample, calculate the vote result givien by classification tree when it is used as oob sample, and return the result category that gets the most vote as the class label of that sample. the proportion of the number of mis-classifited samples and the total samples is oob error estimates[12]. Oob error estimation diagram as shown in Figure 2.

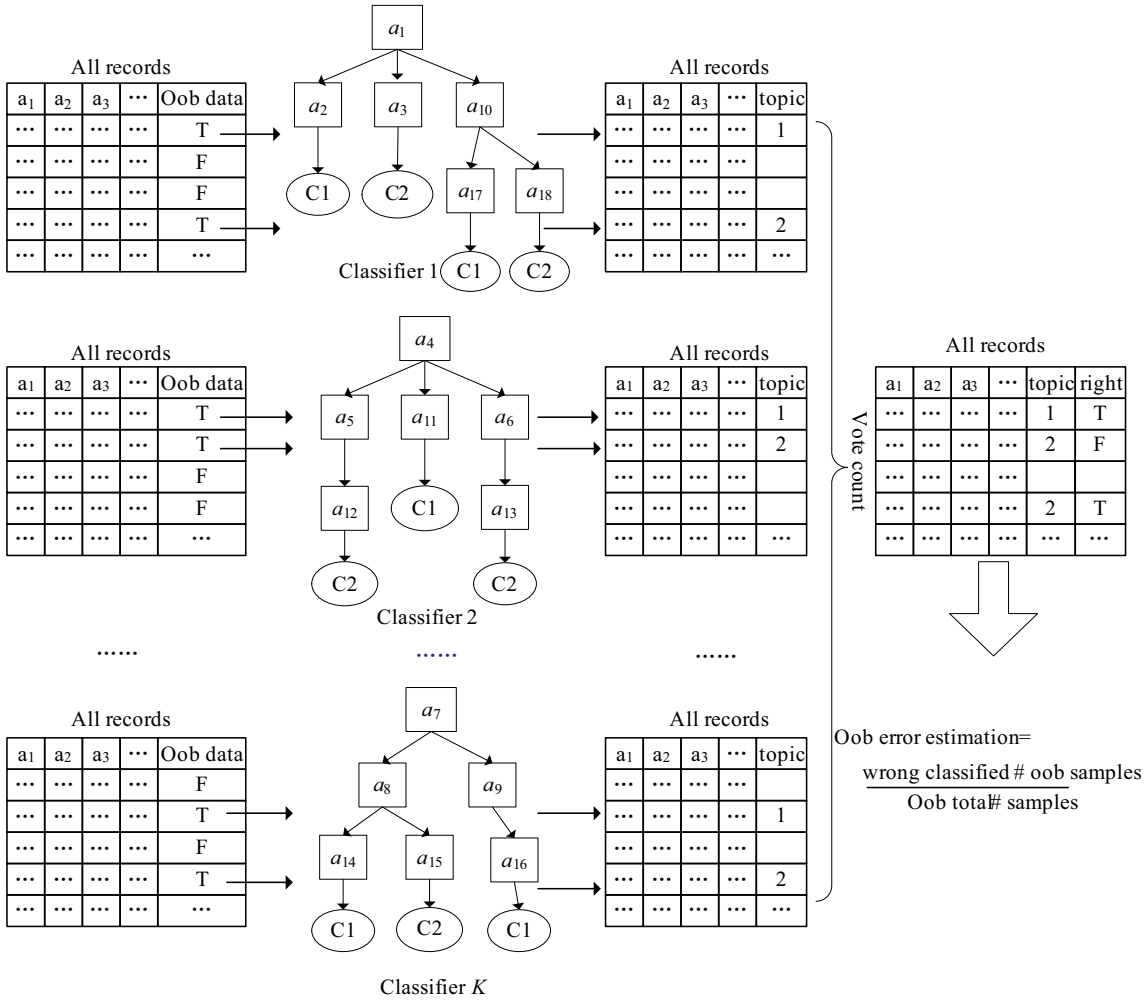


Figure 2. Mapping from topic graph cluster to data table

RF can be regarded as the collection of tree-shaped classifiers which formed a more powerful classifier $\{h(\boldsymbol{x}, \Theta_k), k=1, \ldots\}$. $\boldsymbol{x}$ is the input variable, i.e. one record, and Θ_k is a identified and independent distributed(iid) variable, which determined the growing process of the k-th classifier. The class marginal function can be represented as

$$mf(\boldsymbol{x}, y) = \frac{1}{K}\sum_{k=1}^{K} I(h(\boldsymbol{x}, \Theta_k) = y) - \max_{j \neq y} \frac{1}{K}\sum_{k=1}^{K} I(h(\boldsymbol{x}, \Theta_k) = j) \quad (3)$$

where, $mf(\boldsymbol{x}, y)$ represents the margin of the maximum average votes number between the correctly classified average vote number, while $I(.)$ is the index function. The larger the marginal is, the better the effect we get. Thus, the generalized error (GE) of random forest can be represented as

$$ge = P_{\boldsymbol{x},y}(mf(\boldsymbol{x}, y) < 0), \quad (4)$$

and when the number of classifier get larger. The GE of a random forest can converge in a relatively index as

$$ge^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}. \quad (5)$$

The error rate of RF depends on two aspects: one is based on the original interval function relationship between the classification tree, greater the correlation is, higher the error rate is; the other is classification tree's intensity, which is in fact the average deviation of the right votes number and the wrong votes number. Bigger intensity indicated higher classification accuracy, and the classifier is stronger, and thus get lower error rate. While the number of variable m is big indicates more powerful classification tree, but greater correlation, and vice versa. In order to get minimum generalized error, greater intensity and less relevance is desirable. Thus we have to find the trade-off of strength and relevance.

Generalized error of random forests are determined by the correlation between the strength of the classification tree and the correlations among classification trees, while the OOB error estimates are unbiased estimation of the generalization error of random forest, therefore, the best value for m occurs when the oob error arrive at its minimum[12]. Originally we can choose root of M as the initial value of m, M is the number of all variables, and then to calculate the initial value gradually do OOB error estimate, so as to find the best value of m.

(5) Balance error. There is phenomenon that the error of some class label is very low, while that of other class label is huge, and this is so-called error unbalanced phenomenon. This is due to that the sample number of different categories differs a lot. Therefore, we need to balance the error, and we can inverse proportion of different categories according to the number of samples set with different weights. Having Y class labels, for example, the total sample number is S, the sample number of each class is $S_1, S_2, \ldots, S_y$, then the weight of each class label cw_y is

$$cw_y = \frac{[S_1, S_2, \ldots S_Y]}{S_y \cdot \sum_{i=1}^{Y} S_i} \quad (6)$$

where $[S_1, S_2, \ldots, S_y]$ denotes the least common multiple. We can evaluate error under each class according to the weight, and adjust weight value aptly to get fairly balanced result. It is noted that balanced error will amplify the total error to a certain degree. Our model is completed. We can also find the novel record in training set based on this model, which can inversely correct original topic graph. For the upcoming new topic graph cluster, through the model of the new topic graph cluster formation of each record category prediction as well as new categories of testing, so as to complete incremental update of two topic graph cluster.

3.2. Correlation prediction

We can do online topic detection via correlation prediction which can be done by following actions. Represent the topic graph cluster of newly coming text subset in the form of data table. Re-predict the record table's topic label in each classifier using the well trained random forest, and summarize the vote result of all classifier trees as the current record's prediction result. Since the same term of a topic graph cluster may be in different topic graph, one term may be classified into different topic class for the difference of related feature term. Each record can be viewed as a star structure, representing the relationship of each term and other terms. Class prediction will break a new topic graph cluster into several star-structured elements, and then attributed to the topic graph in original cluster.

Since all the elements were attributed to existed topic graph, while new collection may contain new topic, novelty detection on these elements is needed to perform before integrating into topic graphs, which may form new topic graph.

3.3. Novelty detection

Proximity evaluation. Each record represents the relationship between a feature term and another term in a topic graph. Similarity between each pair of records can be expressed as a proximity matrix. When traveling classifier tree, if two records are of the same poll, the proximity value increase 1, and standardiza-

tion operation is performed on proximity matrix. The proximity of two record r_i and r_i is evaluated as

$$pr(r_i, r_j) = \frac{\sum_{k=1}^{K} I_k(y_i = y_j)}{K} \quad (7)$$

where K is the number of classifier tree, and y_i is the category of record r_i. $I_k(.)$ is the indicator function, which mean if condition in brackets is meet the value of that function is 1, otherwise 0. it is obvious that the proximity degree matrix is symmetric, positive definite, and matrix element is less than or equal to 1, and diagonal elements are 1.

Novelty record detection. We can detect novel record by finding one that has the smallest proximity compared to other records in the same class. Note that novel record is in terms of the original corpus. The average of proximity of a record of class y is

$$avgpr(r_i) = \sum_{y_j = y, i \neq j} (pr(r_i, r_j))^2 . \quad (8)$$

So the original novelty degree of r_i is

$$rpoutlier(r_i) = \frac{1}{avgpr(r_i)} . \quad (9)$$

We standardize for $rpoutlier(r_i)$, and get novelty degree of r_i. Thus novelty record detection can be performed.

4. Results and Discussion

We select NIPS12 as our data, and we generate topic graph cluster according to section 2.1, then set the threshold so that each time slice have less than 20 topics, and we transform the cluster into data table, transforming a fully unsupervised learning problem into a classification problem. We set the threshold so as to remove the redundant information between terms. Table 1 demonstrates the corresponding relation between threshold and total topic number. From the table we can see that the total topic number reach to its maximum when threshold equals to 0.8. Thus in our experiment we set the threshold to 0.8.

Table 1. Relation between threshold and topics number

thre	Topic num	thre	Topic num
0.1	3	0.6	8
0.2	4	0.7	12
0.3	4	0.8	18
0.4	4	0.9	11
0.5	5	1	3

4.1. NIPSdata

We choose nips12 research literature as our test corpus, which include abstraction from 13 years of papers in Conference on Neural Information Processing Systems(NIPS). For convenience, we download from url *http://www.cs.nyu.edu/~roweis/* for the word-frequency matrix data *nips12raw_str602* which was prepared for further research purpose, and there are 1740 papers in 13 years in total, and the size of word list is 13649. Detailed about topic graph cluster in NIPS12 is shown in Table 2.

Table 2. Detailed about topic graph cluster in NIPS12

Time	Topic no.	Topic term no.	Record no.
2000	3	18	10
2001	4	21	15
2002	3	23	16
2003	3	25	21
2004	4	28	24
2005	3	30	26
2006	3	27	21
2007	3	39	30
2008	3	41	34
2009	3	43	35
2010	3	51	39
2011	3	49	38
2012	3	44	34

4.2. Importance Selection

In order to gain better effect, feature selection is needed, and we use the *tf-idf* technology to rebuild the document-term matrix data, and transform the data into adjacent matrix and compute the word importance using random forest, and the result is shown in figure 3. We selected the most 100 important terms as our training feature. Figure 3 showed the 100 most important terms using two models: Random Forest and ExtrTrees[13]. We see from Figure 3 that the result is much alike to each other, and the result, along with the detail top 40 word list in Table 3 from 100 most important terms also verified the efficiency of our method. Another purpose of term selection is that we can largely reduce the computation cost while maintain the effectiveness.

Table 3. Top 40 word list from 100 most important term

	Word list
Top 10	timing, techniques, receiver, sharing, span, positions, stress, strings, optimum, panel
Top 20	suggest, finding, repetition, subsets, spa, increasing, software, foundation, study, logarithm
Top 30	fraction, sets, spatially, text, severe, tissue, exploratory, rev, successfully, piecewise
Top 40	shavlik, periods, parity, lattice, atkeson, divide, central, denoted, adversary, factorial

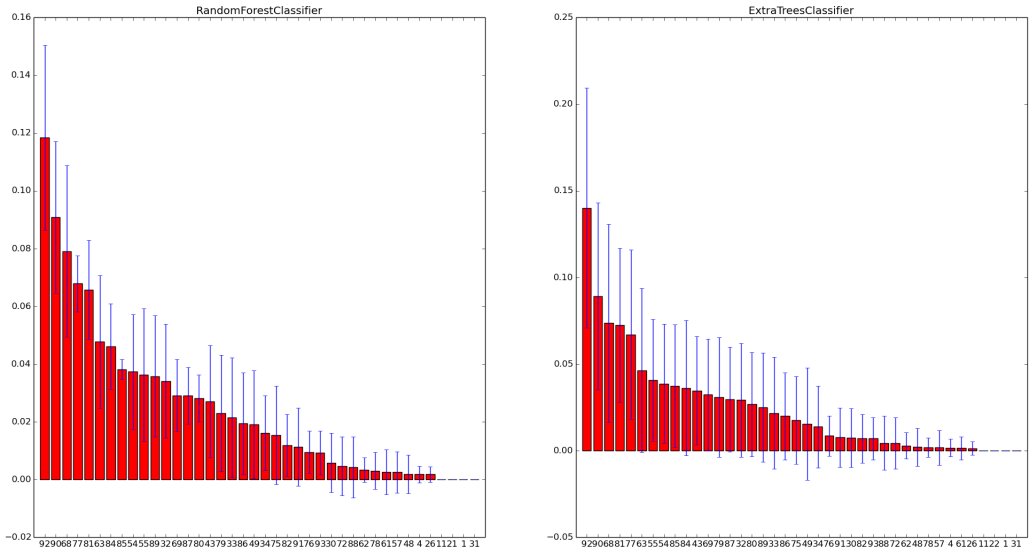


Figure 3. Top-100 important terms using RF, ExtraTrees

4.3. Online Topic Detection

For comparison, we set up experiment based on four kinds of algorithms, i.e. Decision Tree, ExtrTrees, RF and Adaboost classifier, three of which are ensemble learning models and one is merely tree algorithm. For demonstration, we compared these four algorithms in terms of pairwise feature terms, and in our experiment, these feature term-pairs were {92, 90}, {92, 68}, {90, 68}, {68, 77}, and {77, 81}. We firstly standardize our data, then trained the data one after the other by years to discover interesting topic.

5. Result and Analysis

Our experiment is designed in two incremental ways, i.e. (1) model training is based on the first year data, and the data in consequent years is used as test data; (2) iteration way: we trained data in previous year as labeled data, and data in next year is used as unlabeled testing data, and re-construct on topic cluster to form new data as new cycle labeled data, and again treat data in next year as unlabeled testing data, and so forth.

According to section 4.3, we showed experiment result in Figure 4, where there are 5 classes or topics which are marked with red, green, yellow, blue and cyan. From figure 4, we can see that decision tree is too strict, and have binarization effect. When used in first three term pairs, Adaboost gain poor performance. The effect of RF and ExtraTrees are nearly the same, but RF outperformed ExtraTrees in time complexity.

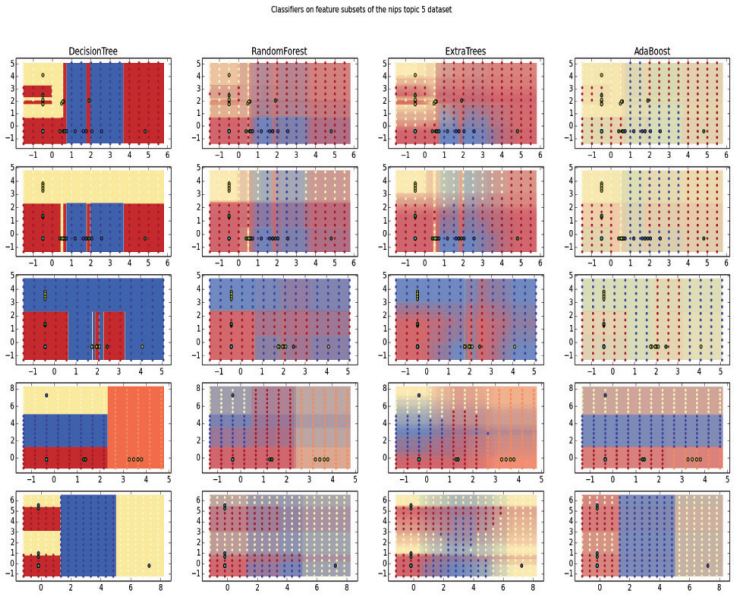


Figure 4. Five term-pairs with five class topic labels using DecisionTree[14], ExtraTrees[13], Random Forest and AdaBoost[15] classifier

6. Conclusion

According to empirical result, our online topic detection algorithm is simple and effective. In this paper, we used a topic graph construction technology to convert an unsupervised learning problem into a classification task, and trained the previous corpus topic graph data to predict the backward data, and partition all the record in a topic graph cluster into either existed topic class or a new type of topic. In construction process, we have update, merging, and add operation. Before training, we used the term importance evaluation to select top 100 topic terms, and it's clear that iteration way is better than the basic incremental way. Further, compared to adaboost, ExtraTrees, DecisionTree, RF gain better classification effect, and can be executed in distributed environment. Besides, no extension operation is needed for data table, and our method can be used to deal with large data. The larger the data is, the more obvious the advantage is.

Acknowledge

This work is supported by (1) the National Natural Science Foundation of China (61403238, 61502288, 61071192, 61271357, 61171178), (2) Natural Science Foundation of Shanxi Province (2014021022-1), (3) Outstanding Graduate Innovation Project of Shanxi Province (20123098), and (4) International S&T Cooperation Program of Shanxi Province(2013081035).

References

1. Gangemi, Aldo, Valentina Presutti, and Diego Reforgiato Recupero. "Frame-based detection of opinion holders and topics: a model and a tool."*Computational Intelligence Magazine, IEEE* 9.1 (2014): 20-30.
2. Yang, Zaihan, et al. «Parametric and Non-parametric User-aware Sentiment Topic Models.» Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015.
3. Burnside, Gérard, Dimitris Milioris, and Philippe Jacquet. «One Day in Twitter: Topic Detection Via Joint Complexity.» *SNOW-DC@ WWW*. 2014.
4. Zhou, Erzhong, Ning Zhong, and Yuefeng Li. «Extracting news blog hot topics based on the W2T Methodology.» *World Wide Web* 17.3 (2014): 377-404.
5. Papadopoulos S, Corney D, Aiello L M. SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media[C]//SNOW-DC@ WWW. 2014: 1-8.
6. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. the Journal of machine Learning research, 3, 993-1022.
7. Hoffman M, Bach F R, Blei D M. Online learning for latent dirichlet allocation[C]//advances in neural information processing systems. 2010: 856-864.
8. Li W, McCallum A. Pachinko allocation: DAG-structured mixture models of topic correlations[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 577-584.
9. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
10. Iverson, L. R., Prasad, A. M., Matthews, S. N., & Peters, M. (2008). Estimating potential habitat for 134 eastern US tree species under six climate scenarios. Forest Ecology and Management, 254(3), 390-406.
11. Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.
12. Aslam, J. A., Popa, R. A., & Rivest, R. L. (2007, August). On estimating the size and confidence of a statistical audit. In Proc. 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT'07).
13. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63(1), 3-42.
14. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.
15. Zhu, J., Zou, H., Rosset, S., & Hastie, T. (2009). Multi-class adaboost. Statistics and its Interface, 2(3), 349-360.