Multi-label Classification of Short Text Based on Similarity Graph and Restart Random Walk Model

Xiaohong Li, Fanyi Yang, Yuyin Ma, Huifang Ma

Multi-label Classification of Short Text based on Similarity Graph and Restart Random Walk Model

Xiaohong Li*, Fanyi Yang, Yuyin Ma, Huifang Ma

Northwest Normal University College of Computer Science and Engineering, Northwest Normal University, Lanzhou China
xiaohongli@nwnu.edu.cn
1521497745@qq.com

Abstract. A multi-label classification method of short text based on similarity graph and restart random walk model is proposed. Firstly, the similarity graph is created by using data and labels as the node, and the weights on the edges are calculated through an external knowledge, so the initial matching degree of between the sample and the label set is obtained. After that, we build a label dependency graph with labels as vertices, and using the previous matching degree as the initial prediction value to calculate the relationship between the sample and each node until the probability distribution becomes stable. Finally, the obtained relationship vector is the label probability distribution vector of the sample predicted by the method in this paper. Experimental results show that we provides a more efficient and reliable multi-label short-text classification algorithm.

Keywords: Multi-label Classification, Short Text, Similarity Graph, Restart Random Walk, WordNet

1 introduction

Traditional single-label classification learning means that each sample has a unique category label, where each label belongs to a mutually exclusive label set L (| L |> 1). However, In practical applications, usually a sample belong to multiple categories at the same time, we call such data as multi-label data[1]. For example, a news report can could be classified into "entertainment" and "technologies", simultaneously. A movie can be both an "action movie" and a "thriller. The multi-label classification is significantly different from the traditional single-label classification. The correlation and co-occurrence between categories lead to those existed single-label classification method cannot be directly applied to the multi-label classification problem. But also multi-label classification is gradually becoming the current research hotspot and difficulty, especially in the fields of text classification, gene function classification, image semantic annotation, etc.

Researchers is finding the optimal classification algorithm to improve the classification accuracy of multi-label data. There are two most common ideas for multi-label classification[2]. One is to convert multi-label dataset into single-label dataset, and then apply traditional data classification algorithm to them(abbreviated as PT). Binary Relevance (BR)[3] is a typical PT method. BR considers the prediction of each label as an

independent single classification problem, and designs an independent classifier for each label, and trains each classifier using all training data. However, it ignores the interrelationships between tags, and often fails to achieve satisfying classification performance. Guo[4]propose a improved binary relevance algorithm, it sets two layers to decompose the multi-label classification problem into L-independent binary classification problems respectively. Liu[5] propose a classifier chain algorithm based on dynamic programming and greedy classifier chain algorithm to search for global optimal labels, which compensated for the Classifier Chain algorithm(CC) defects sensitive to label selection[6]. Label Powerset(LP)[7] encodes every label permutation as a binary number and obtains new labels. Another idea is to modify existing single-label learning algorithm to solve multi-label learning problem. For example, the MLkNN algorithm calculates the prior probability of each label through statistics in the label set, and the probability of the sample with labeled and no label, and then predicts whether the sample has label[8]. Tsoumakas[9] proposed the Random k-Labelsets method to decompose the initial label set into several small random subsets, and use the Label Powerset algorithm to train the classifier. In addition, other researchers have also used various methods for multi-label classification research [10, 11, 12, 13]. In the data prediction training process, the existing multi-label classification algorithms either ignore the interdependence between category labels, or ignore the important influence of initial features on the predicted value, and even add these tags to the original features as an additional function. It makes the feature set that has a very high dimension more complicated. Even if the dependency relationship between category labels is fully utilized, the multi-label classification algorithm ignores the initial prediction value between the label set and the training set, it maks the multi-label classification inaccurate.

We propose a multi-label short-text classification algorithm which combines the similarity graph and the restart random walk model (abbreviated as *SGaRW*). On the one hand, the similarity graph is used to calculate the original relationship between the text and the labels, and on the other hand, we utilize the restarted random walk model to calculate the potential semantic relationships between the labels and the labels. Finally, reasonable fusion is performed to make multi-label classification result more accurate.

2 Preliminary and Background

We review the existing basic concepts and define the problem of multi-label classification in this section.

2.1 Multi-label Classification

Fundamentally, multi-label classification can be considered as a label ranking problem[14,15]. This correlation is scored based on the correlation between the test sample and each category label, and then the label to which the sample belongs is determined based on the score value. Assume that $\boldsymbol{X}=\{x_1,x_2,\ldots,x_n\}$indicates the sample set, $Y=\{y_1,y_2,\ldots,y_m\}$ is label set, and $D=\{(x_i, Y_i)|1\leq i \leq n\}$ is dataset, $Y_i \subseteq Y$ is label set of

the sample x_i. Thus prediction of the label for sample x could be expressed as following vector H(x).

$$H(x) = (h_1(x), \mathrm{L}\ , h_i(x), \mathrm{L}\ , h_m(x)) \tag{1}$$

In the vector, $h_i(x) \in [0,1]$ describes relevancy between sample x and label y_i. Multi-label classification is to achieve a classifier h: $X \rightarrow 2^Y$ using training data. Given new sample x, the classifier can predict label set of the sample x subsumes. Therefore, multi-label classification is to seek an optimal classification algorithm to construct a high-precision score vector H(x) to achieve the purpose of accurate classification.

2.2 Similarity Graph

Similarity graph[16,17] built based on WordNet is a directed weighted graph G=(V, E) is used to calculate semantic similarity among nodes in the graph, V={*itemsset*, *senseset*}, *itemsset* is a collection of nodes (*item*) that represent words, *senseset* is a set composed of nodes (*sense*) that represent senses. According to the corresponding relationship between them, a directed edges <v_i, v_j> is added between two sense nodes, or between an *item* and a *sense*, or between two *items*. weight on the edge is signed as w_{ij}, w_{ij} represents the probability of thinking of the node v_j definitely when seeing the current node v_i, therefore, the weight w_{ij} reflects a conditional probability. So the similarity graph can be called a probability graph.

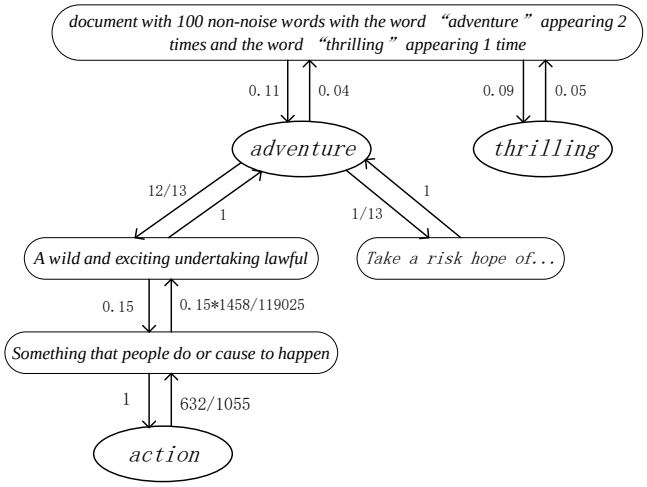

Fig. 1. similarity graph

Such as the similarity graph shown in Fig.1, *itemsset* ={ adventure, thrilling, action }, *senseset*={ "a wild and exciting undertaking lawful", "take a risk in the hope of ... ", "something that people do or cause to happen" }. In WordNet, the word "adventure" has two meanings in total, the using frequency of the first meaning is 0.92, and using frequency of the second meaning is 0.08, so the weight from the *item* node "adventure" to the first *sense* node " a wild and exciting undertaking lawful " is 0.92, which means

that the probability that someone is interested in the first meaning after seeing the word " adventure " is 0.92. In turn, The weight from *sense* node "a wild and exciting undertaking lawful" to *item* node "adventure" is 1, which means that someone must think of the word "adventure" when they see either “a wild and exciting undertaking lawful” or “take a risk in the hope of ... ”.

3 Implement Multi-label Classification of Short Text

Implementing multi-label classification of text is divided into two stages in this paper. At the first step, we create a similarity graph based on the text content, and calculates the original relationship between the text and the label set, which is the initial predicted value $\boldsymbol{H}(x)$. In the second phase, a label dependency graph is constructed and the restart random walk algorithm is performed on this graph to mine the potential semantic relationships between the labels. When the algorithm converges, we can obtain a vector consisting of the probability that the text belongs to each label, so we get labels which belong to the text.

3.1 Calculate Initial Association Between Sample and Labels

We consider short texts as *sense* nodes, map labels to *item* nodes, and create similarity graph $G_1=(V_1, E_1)$. Then affinity score between the text and the label on a directed path can be defined as the product of the weights of all adjacent edges between the text node and the label node on the path[18], as shown in formula (2):

$$affinity_{pt}(v_{doc} \mid v_{label}) = \prod_{\substack{v_i, v_j \in pt \\ (v_i, v_j) \in E\,1}} P_{pt}(v_i \mid v_j) \tag{2}$$

Where, $affinity_{pt}(v_{doc}|v_{label})$ is affinity score from v_{doc} to v_{label}, nodes sequence $pt=<v_{doc},\ldots v_i, v_j,\ldots v_{label}>$ is a directed path from v_{doc} to v_{label}, $p_{pt}(v_i|v_j)$ is weight on the edge between v_i and v_j in G_1, which support calculating affinity scores between two nodes. According to Markov model, as the path length increases, the value of the conditional probability decreases. The longer the path, the less evidence of an intimate relationship between the two nodes.

We also know that there is more than one directed path from v_{doc} to v_{label} in the similarity graph G_1. So affinity scores of the text-to-label on the entire graph G_1 can be expressed as the sum of the affinity score on all directed paths between these two nodes. $Aff'(v_{doc}, v_{label})$ denote affinity scores in formula (3)

$$Aff'(v_{doc}, v_{label}) = \sum affinity_{pt}(v_{doc} \mid v_{label}) \tag{3}$$

Due to the asymmetric nature of the affinity scores, the final affinity scores between two nodes can be obtained by formula (4).

$$Aff(v_{doc}, v_{label}) = \frac{Aff'(v_{doc}, v_{label}) + Aff'(v_{label}, v_{doc})}{2} \tag{4}$$

Treat the affinity scores between v_{doc} and v_{label} as the correlation score $h_i(x)$ between sample x and label y_i, That is:

$$h_i(x) = Aff(x, y_i) \tag{5}$$

Taking all the labels into account, we can get the correlation scores of the sample x and all labels in label set Y, as shown in formula (6)

$$\boldsymbol{H(x)} = [Aff(x, y_1), ...Aff(x, y_i)...Aff(x, y_m)]^T \tag{6}$$

3.2 Random Walk on Label Dependency Graph

3.2.1 Obtain Dependency among Labels.

We construct graph $G_2=(V_2, E_2)$ to encode dependency among labels. Vertices in the graph G_2 represent labels in Y. If the label y_i and y_j mark the text x at the same time, add an edge between y_i and y_j, and the weight w_{ij} is defined as the number of samples labeled by labels y_i and y_j commonly:

$$w_{ij} = \left| \{ x_k \mid y_i \in x_k \wedge y_j \in x_k \} \right| \quad if \ i \neq j \tag{7}$$

The adjacency matrix is used to store graph G_2 and m × m dimensional symmetric matrix is obtained. Therefore, the obtained matrix after utilizing equation (9) to make it asymmetric is represented as $\boldsymbol{S}$, and its element s_{ij} is used to represent the jump probability from label y_i to label y_j, m_j is number of non-zero elements in the j-th column.

$$s_{ij} = \frac{w_{ij}}{m_j} \tag{8}$$

3.2.2 Restart Random Walk.

.Random walk with restart[19] is defined as equation (9), it starts from a random node to retrieve graph. The retriever iteratively transmits to its neighborhood with the probability that is proportional to their edge weights, or it has some probability α to return to the starting point, until the steady-state is reached.

$$\boldsymbol{P}_i = a\boldsymbol{S}\boldsymbol{P}_i + (1-a)\boldsymbol{H} \tag{9}$$

Since prediction of every label can be delivered to other labels to some extent, label prediction related to samples not only is determined by samples, but also could be strengthened by other labels. We uses random walk model to predict multiple labels of a sample. Additionally, initial probability between sample x and each label is defined as $1/m$.

$$\begin{cases} P(Y)_x(0) = [\frac{1}{m}, ..., \frac{1}{m}]^T_m \\ P(Y)_x^{(t+1)} = a\boldsymbol{S}\boldsymbol{P}(Y)_x^{(t)} + (1-a)\boldsymbol{H}(x) \end{cases} \tag{10}$$

$P(Y)_x^{(t)}$ is probability distribution vector which represent the relationship between the sample and each label at time t. $\boldsymbol{S}$ is probability transformation matrix. $\boldsymbol{H(x)}$ is aforementioned initial prediction value vector of labels of sample x. The process continues

until $\boldsymbol{P}(\mathrm{Y})_x$ converges. Prediction of the label is updated repeatedly, dependency among labels could be utilized sufficiently.

4 Experimental Result and Analysis

In this section, we explain the means by which similarity graph and restart random walk model are evaluated, whilst providing a description of the multi-label dataset and other settings used in the experimental study. Finally, the experimental results on the dataset and the statistical analysis are discussed.

4.1 Dataset

The data used in the experiment is English movie titles and overviews collected manually, it is called Movies dataset. Dataset statistics is shown as table 1, in which label density equals to size of label set q divided by potential of the label set c, indicating probability that a label appears.

Table 1. Several statistical value

Dataset name	Size	Size of label set(q)	Label density	Size of elements in Y(c)
Movies	2000	14	0.212	2.972

4.2 Evaluation Metrics

Traditional single classification performance evaluation metrics, such as recall and accuracy, cannot be used directly to evaluate the multiple-label classification performance. Therefore, we use the following three metrics to measure the performance of our method.

4.2.1 Hamming Loss

. Hamming Loss[20] measures classification error based on single-label classification, that is, labels that belong to the sample do not appear in the labels set, but labels that the sample do not have appear. Smaller value means better performance of a classification model. The best is when it is 0. It is defined as:

$$Hamming\text{-}loss(x_i, y_i) = \frac{1}{|D|}\sum_{i=1}^{D}\frac{xor(x_i, y_i)}{|L|} \tag{11}$$

$|D|$ represents total number of samples. $|L|$ represents total number of labels. x_i and y_j represent prediction result and true label respectively.

4.2.2 Jaccard Index

Jaccard Index[21] measures how similar two sets are. It is defined as size of intersection divided by size of union. Bigger value means better performance of a classification model. It is defined as:

$$Jaccard(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{12}$$

4.2.3 Accuracy-score

Accuracy-score[22] is used to compute accuracy of prediction. In multi-label classification, the function returns accuracy of subsets. The accuracy is 1 if entire prediction labels are consistent with real labels, meaning it reaches the best performance, otherwise is 0. It is defined as following:

$$accuracy(y,\hat{y})=\frac{1}{|L|}\sum_{i=1}^{|L|}1(\hat{y}_i=y_i) \tag{13}$$

$\hat{y}_i$ is prediction value of the *i*-th sample and y_i is corresponding real value.

4.3 Experimental Result and analysis

Three experiments are designed to evaluate performance of algorithm of this paper on multi-label text classification. 1) Analyze influence of different α on algorithm, 2) Compare and analyze results by change the size of training set and test set v, 3) Compare our algorithm with other algorithms.

Experiment 1. Analyze influence of different α on our method. Let α=0.0001, 0.00007, 0.00004, 0.00001 to operate experiment respectively. From table 1 we see that three metrics reach all the largest when α=0.00007, and the result is optimal at this point. When α is larger than 0.00007 or smaller than 0.00007, the performance of the algorithm tend to be poor. Generally speaking, influence of α on performance is limited(not exceeds ±0.36%).

Table 2. Experimental results when s values are different

α	Hamming-loss	Jaccard	Accuracy-score
0.00010	0.1073	0.6588	0.8957
0.00007	**0.1043**	**0.6610**	**0.8959**
0.00004	0.1032	0.6609	0.8958
0.00001	0.1050	0.6573	0.8950

Experiment 2. Use training set *t* and test set *v* of different sizes and analyze experimental results. Use training set whose size is 300, 600, 900, 1200, 1500 and test set whose size is 100, 300, 500. As Table 3 shows, when the size of test set |*t*|=100 and the size of training set |*v*| is 300, the performance outperforms the others. When |*t*|=300 and |*v*|=1200, the result is wonderful. In summary, the performance obtained by our method is optimal when |*t*| = 500 and |*v*| = 1500.

Next, we select a group of data with the best classification performance for further comparative analysis. Specifically, |*t*|=500, and the size of training set *v* has different scales. It can be observed from Fig. 2 that Hamming's Loss gradually decreases and Accuracy-score continues to increase as the size of the training set increasing, it means that classification performance of the algorithm tend to get better when the ratio between training data and test data increases. When |*v*| = 1500, the classification score both reaches the optimal.

Table 2. Experimental results when the training set *v* is different from the test set *t* scale

Test(t)	Training(v)	Hamming-loss	Jaccard	Accuracy-score
|*t*|=100	|*v*|=300	**0.1097**	**0.5609**	**0.8359**
	|*v*|=600	0.1171	0.6512	0.8929
	|*v*|=900	0.1246	0.6017	0.8731
	|*v*|=1200	0.1464	0.5524	0.8536
|*t*|=300	|*v*|=900	0.1219	0.5803	0.8781
	|*v*|=1200	**0.1117**	**0.6088**	**0.8583**
	|*v*|=1500	0.1245	0.5734	0.8754
|*t*|=500	|*v*|=1000	0.1403	0.5219	0.8613
	|*v*|=1500	**0.1073**	**0.6657**	**0.8786**

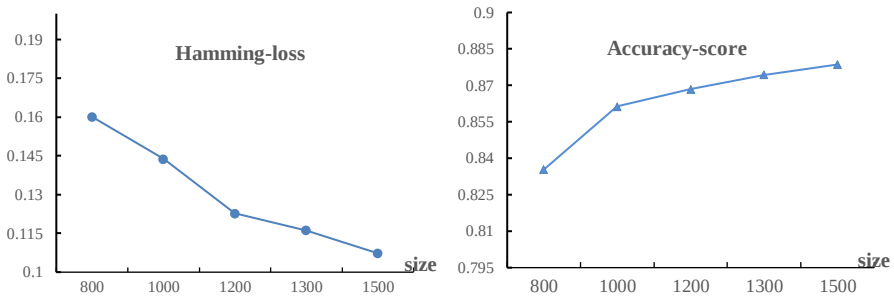

Fig. 2. when |t|=500, Changes in Hamming loss and accuracy-score with different |v|

Experiment 3. To demonstrate how our method improves multi-label text classification performance, we compare our method with other methods in comparison to those existed similar methods, they are *BR*, *LP*, *CC* and *MLkNN* and so on. It should be noted that the parameter value of the *MLkNN* algorithm is set to k = 20, and parameters in other algorithms use the default value. The classifiers for the *BR*, *LP* and *CC* use the Naive Bayes classification.

Fig.3 show that *SGaRW* algorithm has a larger Accuracy-score value compared with *MLkNN*, it indicates that the accuracy of the labels of the text predicted by our method is higher. The Jaccard index of our method is greater than *MLkNN*, while the Hamming loss is less than *MLkNN*. In other words, using the *SGaRW* algorithm will make the labels that do not belong to the text appear in the predicted label set as little as possible,

which reduce the error rate a lot. Comparison with *BR*, *LP*, *CC* and *MLkNN* algorithms shows that *SGaRW* algorithm has great advantage over other algorithms.

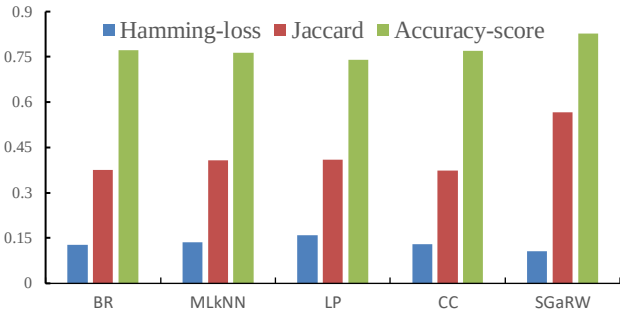

Fig. 3. Comparison of the different algorithms

5 Conclusion

We introduces a novel method *SGaRW* algorithm combining similarity graph and random walk model, which can resolve multi-label text classification problems efficiently. Utilizing prior information from WordNet to build similarity graph, and computing initial match value between labels and texts on it. Then a label dependency graph is constructed, and random walk with restart is been run on it. Finally labels of the text are determined. Core of the future work is to consider expanding dataset, introduce short text semantic understanding to improve performance of short text multi-label classification and optimize effectiveness of the algorithm furtherly.

6 Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 61762078, 61862058, 61967013), Youth Teacher Scientific Capability Promoting Project of NWNU (No. NWNU-LKQN-16-20).

References

1. Tsoumakas G., Katakis I., Vlahavas I:Mining Multi-label Data. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA(2009)
2. Zhang M L , Zhou Z H: Multi-label neural networks with applications to functional Genomics and Text Categorization.J. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10):1338-1351
3. Trohidis K, Tsoumakas G, Kalliris G: Multi-label classification of music by emotion. J. Eurasip Journal on Audio Speech & Music Processing, 2011, 2011(1):4
4. Guo T, Li G Y: An Improved Binary Relevance Algorithm for multi-label classification. J. Applied Mechanics and Materials, 2014, 536-537:394-398

5. Liu W , Tsang I W: On the optimality of classifier chain for multi-label classification. In: International Conference on Neural Information Processing Systems. MIT Press, 2015
6. Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification. J. Machine learning, 2011, 85(3): 333
7. Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning. J. Pattern Recognition, 2007, 40(7):2038-2048
8. Tsoumakas G , Katakis I , Vlahavas I: Random k-labelsets for multi-label classification. J. IEEE Transactions on Knowledge & Data Engineering, 2011, 23(7):1079-1089
9. Read J. A pruned problem transformation method for multi-label classification. In: New Zealand Computer Science Research Student Conference (NZCSRS 2008). 2008, 143150: 41
10. Jizhao Q , Hua J I , Huaxiang Z: Modified algorithm with label-specific features for multi-label learning. J. Computer Engineering and Applications, 2013, 49(22):163-166
11. J. Huang, G. Li, S. Wang, W. Zhang and Q. Huang: Group sensitive Classifier Chains for multi-label classification. In: IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015:1-6
12. Huang J , Li G , Huang Q , et al: Learning label-specific features and class-dependent labels for multi-label classification. J. IEEE Transactions on Knowledge & Data Engineering, 2016, 28(12):3309-3323
13. Qiao L , Zhang L , Sun Z , et al: Selecting label-dependent features for multi-label classification. J. Neurocomputing, 2017
14. Li X , Ouyang J , Zhou X: Supervised topic models for multi-label classification. Elsevier Science Publishers B. V. 2015
15. Soleimani H , Miller D J . Semi-supervised multi-label topic models for document classification and sentence labeling. In: Acm International on Conference on Information & Knowledge Management. ACM, 2016.
16. Stanchev L: Creating a Similarity Graph from WordNet. In: 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). Association for Computing Machinery, New York, NY, USA, Article 36, 1–11
17. Stanchev L: Semantic document clustering using a similarity graph. In: IEEE Tenth International Conference on Semantic Computing. IEEE, 2016:1-8
18. Stanchev L: Creating a probabilistic graph for WordNet using markov logic network. In: 6th International Conference on Web Intelligence, Mining and Semantics. 2016: 1-12
19. Tong H, Faloutsos C, Pan J Y: Fast random walk with restart and its applications. In: 6th international conference on data mining (ICDM'06). IEEE, 2006: 613-622.
20. Díez J, Luaces O, del Coz J J, et al: Optimizing different loss functions in multi-label classifications. J. Progress in Artificial Intelligence, 2015, 3(2): 107-118
21. Hamers L , Hemeryck Y , Herweyers G , et al: Similarity measures in scientometric research: the Jaccard Index versus salton's cosine formula. J. Information Processing & Management, 1989, 25(3):315-318
22. Hubley A M . Using the Rey-Osterrieth and modified Taylor complex figures with older adults: a preliminary examination of accuracy score comparability. J. Archives of Clinical Neuropsychology the Official Journal of the National Academy of Neuropsychologists, 2010, 25(3):197