Enhanced K-Nearest Neighbour Algorithm for Large-scale Hierarchical Multi-label Classification *

Xiao-lin Wang, Hai Zhao, and Bao-liang Lu**

Shanghai Jiao Tong University,
800 Dongchuan Rd., Shanghai, China
arthur.xl.wang@gmail.com, {zhaohai,blu}@cs.sjtu.edu.cn

Abstract. Large-scale hierarchical classification tasks typically have tens of thousands of classes as well as a large number of samples. In this paper we proposed an enhanced K-Nearest Neighbour method (EKNN) to address such tasks. EKNN employs a novel thresholding strategy to predict multiple labels. We analyze the complexity of EKNN, and present its performances at the second Pascal challenge on large scale hierarchical text classification. EKNN turns out to be ranked at the first place in two tracks and ranked sixth in another track in terms of classification accuracy.

Keywords: large-scale classification, KNN, thresholding strategy

1 Introduction

Test categorization, as a key technology of data mining, has received intensive study for decades. Many classification methods have been proposed, and proved to be quite effective on benchmark data sets such as Reuters-21578 and RCV1 [12]. However, large-scale hierarchical classification problems typically have tens of thousands of classes, where many established techniques such as the 1-vs-Rest multiclass classification fail due to computational complexity.

A popular solution to these large-scale hierarchical classification problems is the ensemble learning approach of top-down method. Top-down method first builds a hierarchical network of classifiers, and then classifies a sample by passing it down this network of classifiers from the root node. At each parent node that the sample reaches, the child nodes whose confidence scores meet a predefined requirement are invoked to carry the sample on. When the sample reaches the bottom leaf nodes eventually, the predictions can be made [13, 16, 27].

* This work was partially supported by the National Natural Science Foundation of China (Grant No. 60903119, Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No.2008AA02Z315).

** correspondent author

Top-down method has proved to be quite efficient in solving these large-scale hierarchical classification problems. In the first Pascal challenge on Large Scale Hierarchical Text Classification (LSHTC), quite a few participants have adopted the top-down method [11]. However, the top-down method has some significant weaknesses. One weakness is the deficiency of classification accuracy, that is, its accuracy is generally lower than the flat one-vs-rest approach, which is caused by the so-called error propagation in deep levels of the hierarchy [2, 3, 23, 25]. Another weakness is that the training of the root classifier has to be performed on the whole training set, which is usually quite time consuming.

In this paper, we propose an Enhanced K-Nearest Neighbour method, noted as EKNN, to solve large-scale hierarchical classification problems. K-Nearest Neighbour algorithm (KNN) is a method of classifying samples based on closest training samples in the feature space [5]. Adopting KNN model instead of the top-down method provides a variety of merits including no error propagation, ease to parallel and so on. The main challenge of applying KNN is how to efficiently find the k nearest neighbors of a given sample and how to raise the classification accuracy, which are addressed by EKNN. The second Pascal challenge on Large Scale Hierarchical Text Classification (LSTHC2) provides us a great opportunity to test EKNN [1]. The results indicate that EKNN possesses both efficiency and accuracy in solving these large-scale tasks.

This paper is organized as follows. We first review related works on Sec. 2. Then our EKNN method is presented in Sec. 3. After that we report the experimental results on LSHTC2 datasets in Sec. 4. A hierarchical voting scheme for KNN is discussed in Sec. 5. Finally we conclude this paper in Sec. 6.

2 Related Works

2.1 *K*-Nearest Neighbour Algorithm

K-Nearest Neighbor (KNN) is a type of instance-based learning method, which classifies samples based on closest training samples in the feature space. It is amongst the simplest of all machine learning algorithms [5]. BM25 is a widely used similarity measurement in KNN, through which the k nearest training samples for a given test sample are found [17, 18](see Sec. 3.1 for details).

As the KNN classifier requires storing and searching the whole training set which becomes too costly when this set grows large, many researchers have attempted to reduce the redundancy of the training set to alleviate this problem [8]. Hart proposes the Condensed Nearest Neighbour (CNN) method which stores only a subset of the training set [9]. Gate proposes the Reduced Nearest Neighbour (RNN) rule that aims to further reduce the stored subset after having applied CNN [7].

The classification accuracy is always a great concern for classification methods. A variety of voting methods have been studied in order to improve classification accuracy [14, 24]. Zhang and Zhou propose a voting method for multi-labeled tasks based on the "maximum a posteriori" principle [29]. Large-margin

[1] `http://lshtc.iit.demokritos.gr/`

principle and discriminative model have also been incorporated into KNN [15, 28].

2.2 Thresholding Methods

Thresholding method is a general solution to multi-labeled classification for a large group of classification algorithms which yield a confidence score for each candidate class [26, 20, 21]. One-vs-rest SVMs combined with proper thresholding strategies are recognized as one of the-state-of-art method for multi-labeled classification [13, 19]. A recent study of thresholding methods can be found at [10].

Two sorts of thresholding strategies are frequently used [6, 26]. One is Score-cut strategy (S-cut) which predicts a class if its confidence score exceeds a predefined threshold. The other is Rank-cut strategy (R-cut) which predicts the top-n classes with the highest confidence scores. Our empirical research shows that S-cut with proper thresholds can achieve higher accuracies than R-cut, while R-cut is more robust and always provides stable and competent performance.

In this paper, we propose a novel thresholding strategy which maintains both the accuracy of S-cut and the robustness of R-cut.

3 Methods

Our method is to enhance the KNN algorithm to solve large-scale hierarchical multi-label classification. Fig. 1 presents the workflow of the enhanced KNN algorithm. Note that the inverted index is employed for the sake of efficiency (see Sec. 3.3 for details).

Our method works as follows:

1. find the k nearest neighbors according to BM25 similarity;
2. the k nearest neighbors weighted vote on candidate classes;
3. decide the final predictions via the thresholding strategy.

The following three subsections present the related details.

3.1 Samples' Similarity and Weighted Voting

The similarity of BM25, with which the k nearest training documents for a given test document are found, is computed according to the following formula [18],

$$BM25(s_a, s_b) = \sum_{w \in s_a \cap s_b} f'(w, s_a) * f'(w, s_b) * idf(w) \tag{1}$$

$$f'(w, s) = \frac{(k_1 + 1) f(w, s)}{k_1 + (1 - b + b\frac{|s|}{l_a})} \tag{2}$$

$$idf(w) = log\frac{N - n(w) + 0.5}{n(w) + 0.5} \tag{3}$$

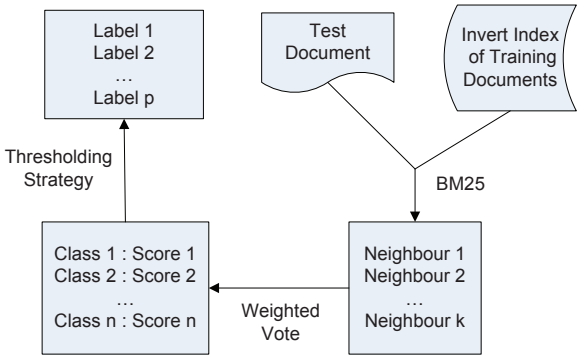


Fig. 1: The workflow of EKNN algorithm

where s_a and s_b are two documents, w is a common word of s_a and s_b. $f(w, s)$ is the term frequency of w in the document s, k_1 and b are the two parameters to scale its value, and l_a is the average length of training documents. $idf(w)$ is the inverse document frequency, N is the number of training documents, and $n(w)$ is the document frequency of w.

The k nearest neighbors perform weighted voting on candidate classes, according to the following formula,

$$score(c|s_e) = \sum_{i=1}^{k} \gamma(s_i, c) BM25(s_i, s_e)^{\alpha} \tag{4}$$

where c is a candidate class. s_e is the test document, and s_i is one of its k neareast neighbors.$\gamma(s_i, c)$ indicates whether c is label of s_i. α is a parameter of positive real number. This voting formula performs better than others according to our pilot experiments.

3.2 Distinctive Score-Cut Thresholding Strategy

Distinctive Score-cut strategy (DS-cut) employs different thresholds of scores for the candidate classes ranked at different positions. This strategy is formulated as follows:

Suppose s_e is a test sample. c_1,c_2,. . . ,c_k are its top-k candidate classes, and s_1, s_2, . . . , s_k are the corresponding confidence scores which are sorted in descending order. DS-cut employs the following formula:

$$\delta(c_i) = \begin{cases} 1 \text{ if } s_i \geq t_i \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

where $\delta(c_i)$ indicates whether the candidate class c_i is accepted, t_i is the predefined threshold for the position i. Note that $\delta(c_i)$ is considered only if $\delta(c_1)$,$\delta(c_2)$,. . . ,$\delta(c_{i-1})$ all output 1.

In real-world applications, the confidence scores owned by different samples may vary greatly. Therefore, scaling the confidence scores according to the largest one can gain extra robustness, which leads to the following Distinctive Scaled-Score-cut (DSS-cut) strategy,

$$\delta(c_i) = \begin{cases} 1 \text{ if } s_i/s_1 \geq t_i \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

where t_i and other settings are the same as in Eq. 5. t_1 should be set 1.0, so DSS-cut always accept the top candidate class. Then the t_2,t_3,...,t_k are tuned by the data set to optimize the performance.

3.3 Complexity Analysis

The main computation of EKNN is finding the k nearest training samples for a given test sample. The technique of inverted index is taken to raise the speed of this process. Inverted index is an index data structure storing a mapping from a word to the documents containing this feature. This technique has been widely used in full-text document retrieval and web-page retrieval [1, 30].

With the inverted index, the computational complexity of EKNN on a given test document is

$$Complexity(s) = \sum_{i=1}^{m} \mathcal{F}(w_i) \tag{7}$$

where s is a test document which contains m unique words w_i. $\mathcal{F}(w_i)$ is the document frequency of w_i, that is, the number of training documents containing w_i.

The complexity presented by Eq. 7 actually greatly decreases in the practical document classification. First, words with large document frequencies are common words, thus they have less impact on a document's categories and can be ignored. Second, the document frequencies of words obey the Power law distribution [4], thus a large part of words in a document actually have low document frequencies.

4 Experiments

The second Pascal challenge on Large Scale Hierarchical Text Classification (LSTHC2) provides a great platform for us to test EKNN. We apply EKNN to all three tasks and achieve promising results.

4.1 Experimental Settings

EKNN totally has three sorts of parameters – BM25 similarity, the weighted voting and the DSS-cut thresholding strategy. For each task, we split the original training set into a new training set and a development set, and use them to tune the parameters. Tab. 1 presents the parameters' settings of EKNN at LSHTC2. All these parameters are tuned by a grid-search strategy, which targets at maximize the criterion of example-based F_1.

Table 1: The parameters' settings at LSHTC2

Parameter	Task	Value
k_1 at Eq. 1	all	6.0
b at Eq. 1	all	0.9
k at Eq. 4	all	30
α at Eq. 4	all	3.0
t_i at Eq. 6	DMOZ	$(1.0, 1.0, 1.0, 1.0, 1.0)$
	Wiki. Small	$(1.0, 0.5, 0.5, 0.5, 0.5)$
	Wiki. Large	$(1.0, 0.5, 0.4, 0.4, 0.4)$

4.2 Experimental Results

Classification Accuracy is measured by a group of criterions at LSHTC2 such as accuracy, exampled-based F_1 and so on [22]. Fig. 2 shows the comparisons between EKNN and the best of other participants' results. EKNN ranks the sixth place in the DMOZ track and the first place in the rest two tracks of Wikipedia-small and Wikipedia-large in terms of the accuracy criterion.

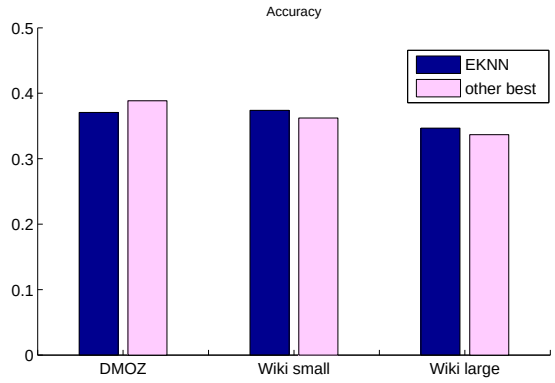
To compare the efficiencies of different thresholding strategies, we construct a test set named WikiSmallTest from the track of Wikipedia-small [2]. We randomly pickup 1000 samples from the origin train data set to form a new test set, and take the rest training samples as a new training set. The experimental results are present at Tab. 2, where SSD-cut performs better than both R-cut and S-cut.

Table 2: The experimental results of different thresholding strategies

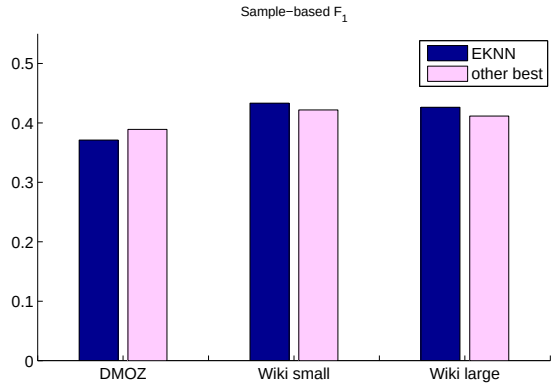
Thresholding	Param.	E-F_1	E-P	E-R
RSS-cut	(1.0, 0.5, 0.5, 0.5, 0.5)	0.444	0.467	0.505
R-cut	r=1	0.396	0.519	0.349
	r=2	0.394	0.378	0.467
	r=3	0.334	0.254	0.581
S-cut		0.323	0.282	0.491

Efficiency is an important factor in processing large-scale tasks. The computational complexity of our method is analyzed in Sec. 3.3. Tab 3 presents the practical time cost of EKNN at LSTHC2 as well as the sizes of data sets.

[2] The oracle is shut down when writing this paper



(a)



(b)

Fig. 2: Classification accuracy comparisons (a) accuracy (b) exampled-base F_1

Table 3: The efficienty aspect of experimental results

Data set	#classes	# Words	# Train docs	# Test docs	Time cost	
					Per doc(secs)	Total(hrs)
DMOZ	27,875	594,158	394,756	104,263	0.223	6.4
Wiki. Small	36,504	346,299	456,886	81,262	0.026	0.6
Wiki. Large	325,056	1,617,899	2,365,436	452,167	0.373	46.8

5 Discussion

As an attempt to further improve the classification accuracy of KNN in the context of hierarchical classification, a hierarchical voting method for KNN is developed by us.

Like the widely used top-down approach to hierarchical classification, we try to estimate the confidence that a child node is chosen for a parent node. Suppose b_1, b_2, ..., b_k are the label sets of the k nearest neighbors of a given test sample s_e. $H = \{(p, c)|c \text{ is a child node of } p\}$ is the hierarchy. Then confidence of s_e choose a child node c from a parent node p is estimated as,

$$\begin{aligned} z_{pc} &\approx P(c|p) \\ &= \frac{\sum_{i=1}^{k} \theta(b_i, c) s_i}{\sum_{i=1}^{k} \theta(b_i, p) s_i} \end{aligned} \tag{8}$$

$$\theta(b_i, n) = \begin{cases} 1 \text{ if } n \text{ is the ancestor of some element at } b_i \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

where s_i is the similarity score the nearest neighbor.

Then, for each candidate class, a sequence of confidence values, which are corresponding to the path from the root to this node, can be generated, as follows.

$$Z_l = (z_{n_0 n_1}, z_{n_1 n_2}, \ldots, z_{n_{p-1} n_p})$$

where l is candidate class, n_0 is the root node, and n_q is the node l, that is, n_0, n_1, ..., n_p is a route from the root to the node of l.

After that, meta criterions of the confidence sequence can be taken as the confidence value of the candidate class. In this paper, the following criterions are tried by us:

$$min(Z_l) = \min_{i=1}^{p-1} z_{n_i n_{i+1}} \tag{10}$$

$$avg(Z_l) = \frac{1}{p} \sum_{i=1}^{p-1} z_{n_i n_{i+1}} \tag{11}$$

$$prod(Z_l) = \prod_{i=1}^{p-1} z_{n_i n_{i+1}} \tag{12}$$

In the end, thresholding strategies are applied to the confidence scores of candidate classes in order to give the final predictions.

The experimental results of this hierarchical voting method on the data set of WikiSmallTest are presented at Tab. reftab:h-vote. Its performances are worse than the straightforward weighted voting of nearest neighbors (see the last line of the table).

Table 4: The experimental results of KNN hierarchical voting

Criterion	Thresholding	E-F_1	E-P	E-R
min	R-Cut,r=2	0.352	0.336	0.420
avg	R-Cut,r=2	0.223	0.208	0.271
prod	R-Cut,r=2	0.362	0.344	0.435
non-hierachy	R-Cut,r=2	0.394	0.378	0.467

6 Conclusion

In this paper, we propose a enhanced KNN method, noted as EKNN, to solve large-scale hierarchical classification problems. EKNN enhances the conventional KNN algorithm in the following aspects,

- employing inverse index to find k nearest neighbors;
- multinomial weighted voting;
- DSS-cut thresholding strategy to predict multiple labels.

EKNN exhibits promising accuracy and efficiency at LSHTC2.

In addition, we also discuss a hierarchical voting method for KNN in this paper. But this method performs worse than the straightforward voting method on LSHTC2 tasks. We plan to focus on how to better incorporate the knowledge of hierarchy into EKNN in the future.

References

1. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463 (1999)
2. Bennett, P., Nguyen, N.: Refined experts: improving classification in large taxonomies. In: Proc. of SIGIR'09. pp. 11–18. ACM (2009)
3. Ceci, M., Malerba, D.: Classifying web documents in a hierarchy of categories: a comprehensive study. Journal of Intelligent Information Systems 28(1), 37–78 (2007)
4. CLAUSET, A., ROHILLA SHALIZI, C., NEWMAN, M.: Power-law distributions in empirical data. SIAM review 51(4), 661–703 (2009)
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on 13(1), 21–27 (1967)
6. Fan, R., Lin, C.: A study on threshold selection for multi-label classification. Department of Computer Science, National Taiwan University (2007)
7. Gates., G.: The reduced nearest neighbour rule. IEEE Transactions on Information Theory 18, 431–433 (1972)
8. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: Knn model-based approach in classification. On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE pp. 986–996 (2003)
9. Hart., P.: The condensed nearest neighbour rule. IEEE Transactions on Information Theory 14, 515–516 (1968)

10. Ioannou, M., Sakkas, G., Tsoumakas, G., Vlahavas, I.: Obtaining bipartitions from score vectors for multi-label classification. In: Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on. vol. 1, pp. 409–416. IEEE
11. Kosmopoulos, A., Gaussier, E., Paliouras, G., Aseervatham, S.: The ECIR 2010 large scale hierarchical classification workshop. In: ACM SIGIR Forum. vol. 44, pp. 23–32. ACM (2010)
12. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
13. Liu, T.Y., Yang, Y., Wan, H., Zeng, H.J., Chen, Z., Ma, W.: Support vector machines classification with a very large-scale taxonomy. ACM SIGKDD Explorations 7(1), 36–43 (2005)
14. M. Kubat, M.J.: Voting nearest-neighbour subclassifiers. Proceedings of the 17th International Conference on Machine Learning pp. 503–510 (2000)
15. Min, M., Stanley, D., Yuan, Z., Bonner, A., Zhang, Z.: Large-margin knn classification using a deep encoder network. Arxiv preprint arXiv:0906.1814 (2009)
16. Montejo-Ráez, A., Ureña-López, L.: Selection strategies for multi-label text categorization. Advances in Natural Language Processing pp. 585–592 (2006)
17. Murata, M., Kanamaru, T., Shirado, T., Isahara, H.: Using the k nearest neighbor method and bm25 in the patent document categorization subtask at ntcir-5. In: Proc. of NTCIR-5 Workshop Meeting (2005)
18. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Overview of the Third Text REtrieval Conference (TREC-3). pp. 109–126. NIST (1995)
19. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)
20. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)
21. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. Data Mining and Knowledge Discovery Handbook pp. 667–685 (2010)
22. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering (2010)
23. Wang, X.L., Lu, B.L.: Flatten hierarchies for large-scale hierarchical text categorization. In: Proc. of fifth international conference on digital information management. pp. 139–144 (2010)
24. Wilson, D., Martinez, T.: Reduction techniques for exemplar-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
25. Xue, G.R., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: Proc. of SIGIR'08. pp. 619–626. ACM (2008)
26. Yang, Y.: A study of thresholding strategies for text categorization. In: Proc. of SIGIR'01. pp. 137–145 (2001)
27. Yang, Y., Zhang, J., Kisiel, B.: A scalability analysis of classifiers in text categorization. In: Proc. of SIGIR'03. pp. 96–103. ACM (2003)
28. Zhang, H., Berg, A., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2126–2136. IEEE (2006)
29. Zhang, M., Zhou, Z.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)
30. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Computing Surveys (CSUR) 38(2), 6–es (2006)