

A Hierarchical ML-kNN Method for Complex Emotion Analysis on Customer Reviews

Hongwei Qi^{1, a}, Yanquan Zhou^{1, 2, b} and Qi Guo^{1, c}

¹School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100000, China

²Engineering Research Center of Information Networks, Ministry of Education

^aqhwh2014@126.com, ^bzhouyanquan@bupt.edu.cn, ^cclass33gq@163.com

Keywords: Sentiment classification, multi-label learning, kNN, hierarchical.

Abstract. Aim at multi-label sentiments classification of customer reviews. This paper presents a hierarchical ML-kNN algorithm. Different from the traditional ML-kNN algorithm, this algorithm can capture the connection of labels by introducing auxiliary labels. The experimental results show that our proposed approach can improve the performances of multi-label sentiments classification of customer reviews. In addition, due to the effective use of auxiliary information, our algorithm can greatly reduce the interference of noises.

1. Introduction

Sentiment analysis as an important branch of NLP is reflected its research value in recent years. For different types of text, sentiment analysis technology can effectively identify potential emotion which is a close combination with the cognitive psychology disciplines.

In recent years, due to the rapid development of the Internet, the UGC (User Generated Content) on the Internet produced an exponential growth trend. Internet e-commerce as a free trade platform, which contains a large number of customer reviews, and it contains a lot of emotional factors in these customer reviews. How to effectively use these data effectively sentiment analysis is being more and more attention of researchers.

In the traditional sentiment classification task, researchers usually divide sentiments into two categories: positive and negative [1, 2]. Its concern is the overall sentiment of the text, namely that the text paragraph with subjective feelings uniquely contain only positive emotion or negative emotion. In real life, when people's views often contain a variety of different emotions. Table 1 shows a part of the customer reviews from an e-commerce website, as can be seen, the user is satisfied with the purchased goods, but blame the service of site. This indicates that user may also express a variety of emotions in a single paragraph.

Table 1 Multi-label emotions for customer reviews

S1	The screen is large enough, good call quality, multi-application software, but photographic imaging results are not ideal, cost can be, is to buy on price, order date and return the difference exceeds millet week, super bad mood.
S2	Looked under, good mobile performance, very suitable for the elderly to use, did not pay attention when buying telecom cards only. Found immediately return, but return several times without success, and only then return when shipping day!

Based on the traditional multi-label learning algorithm ML-kNN, we propose HML-kNN (Hierarchical ML-kNN) algorithm. Different from the traditional ML-kNN algorithm, our method can effectively capture the correlation between labels.

We will introduce the related work of sentiment classification and multi-label learning in Section 2. Section 3 presents the detail about our method. The Experimental results and discussions presented in Section 4. The conclusions and future word are in Section 5.

2. Related work

2.1 Sentiment classification

Sentiment analysis is an important part of NLP, which appeared later than other NLP tasks. It was first reflected in studies of Turney [1] and Pang [2]. They divided movie and product reviews into positive and negative sentiment by using rule-based and statistical machine learning methods. Taboada [3] et al. use emotion dictionary to compute the sentiment polarity of text. Pang [4] and Snyder [5] extended comment emotional category for multiple stars. Kim [6] and Kalchbrenner [7] use Convolution neural network method to classify text sentiments.

Common of these studies are only concerned with overall sentiment while ignoring the case of text that may contain a variety of sentiments. In recent years, multi-label learning machine learning researchers began to receive attention, at the same time began to appear for some research of complex sentiments of text. Liu et al. [8] used a variety of multi-label learning method analyze the complex sentiments of microblog. Xu et al. [9] used a layered combination method to analyze the complex sentiments of microblog.

2.2 Multi-label learning

In the traditional classification task, researcher often considered the only category tag associated with a sample, but in real life, the sample is often more than one classification. In recent years, multi-label study became a hot topic in the field of machine learning, from the research status now, the solution to this problem can be basically divided into two categories: Problem transformation and Algorithm adaptation [10, 11].

Problem transformation method transformed multi-label classification problem into other known problem by processing data. Binary Relevance (BR) algorithm[12] convert multi-label classification task into several independent binary classification task. Classifier Chain (CC) algorithm[13] transform independent binary classification problem into chain form. Calibrated Label Ranking (CLR) algorithm [14] combine labels in pair, and transform multi-label classification problem into the label ranking problem. Random k-Labelsets (RAKEL) algorithm [15] transform the multi-label classification problem into an ensemble of multi-class classification problem.

Algorithm adaptation method retrofit existing machine learning algorithms so that it can be applied to multi-label classification learning. ML-kNN algorithm [16] transform traditional kNN algorithm by maximizing the posterior probability. Rank-SVM algorithm [17] transform the traditional SVM algorithm. LEAD algorithm [18] use Bayesian networks to build relationship between the labels.

3. Hierarchical ML-kNN

In the above multi-label learning methods, most require human participate in the process of feature selection. Different feature selection methods often lead to different results, how to select features is a problem that most machine learning methods need to be resolved. Too many features could easily lead to disaster dimension, and often resulting classifier over-fitting, and too little features often result in loss of information. ML-kNN algorithm determine the labels of target sample by calculating the similarity between samples. Considering the particular nature of text, we can take words as the feature, and calculate the Jaccard similarity between samples. In this way, we can avoid the manual process involved in feature selection, and greatly reduce the impact of different features.

But, traditional ML-kNN algorithm only consider the target label set, therefore it ignores the correlation between labels. To avoid losing such association information, we have improved the traditional ML-kNN algorithm and propose HML-kNN algorithm, it captures the correlation between labels through the auxiliary labels.

An entity on real-world whose labels are usually organized, and we can divide its labels through different hierarchy. As shown in Figure 1, label set $T = \{t_1, t_2, \dots, t_q\}$ contains q labels and label set $L = \{l_1, l_2, \dots, l_r\}$ contains r labels. There is a connection from the relationship from T to L . In practical problems, we generally regard L as the target set and T as the auxiliary set. Next, we will introduce detailed HML-kNN algorithm.

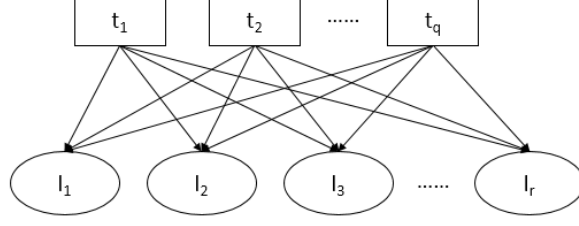


Fig. 1 Hierarchical connection between labels

For convenience of description, we first formal definition of the problem. Suppose X is instance space, $T = \{t_1, t_2, \dots, t_q\}$ is the auxiliary label space with q possible auxiliary labels, $L = \{l_1, l_2, \dots, l_r\}$ is the target label space with r possible target labels. Given a training set $D = \{(x_i, T_i, L_i) | T_i \subseteq T, L_i \subseteq L, 1 \leq i \leq m\}$, our goal is to find one or more labels on target labels set L which correspond to an entity sample. In other words, we hope to get a classifier h that for any sample x , it can give a corresponding label set which is a subset of the target label set L , formal representation is $h(x) \subseteq L$.

Specifically, for a test sample x , we first use the traditional ML-kNN algorithm to train an temporary classifier for determining the auxiliary labels, $h^*(x) = T^* \subseteq T$. When predicting the target labels, we need to consider the impact of the auxiliary labels, therefore, the final classification has the following form:

$$h(x|T^*) = h(x|h^*(x)) = L^* \subseteq L \quad (1)$$

Where $h(x|h^*(x))$ is the target classifier, L^* is predicted target labels set which is a subset of target labels space L .

We suppose $N_k(x)$ is a sample set which is composed of the k nearest neighbors samples of x in the training set D . C_j^T represents the number of samples in $N_k(x)$ which have auxiliary label t_j , and C_s^L represents the number of samples in $N_k(x)$ which have target label l_s . Their calculation method is as follows:

$$C_j^T = \sum_{(x^*, T^*, L^*) \in N_k(x)} \llbracket t_j \in T^* \rrbracket \quad (2)$$

$$C_s^L = \sum_{(x^*, T^*, L^*) \in N_k(x)} \llbracket l_s \in L^* \rrbracket \quad (3)$$

Where the value of $\llbracket \cdot \rrbracket$ equals to 1 when the expression \cdot is established, and equals to 0 otherwise.

And then, we let H_j^T represents the event that auxiliary label t_j associated with x , and $\neg H_j^T$ represents the event that auxiliary label t_j not related to x . Thus, we can get a classifier by traditional ML-kNN which used for predict the auxiliary labels of x :

$$T^* = h^*(x) = \{t_j | \frac{P(C_j^T | H_j^T) \cdot P(H_j^T)}{P(C_j^T | \neg H_j^T) \cdot P(\neg H_j^T)} > 1, 1 \leq j \leq q\} \quad (4)$$

In order to predict the target labels of x , we need the result T^* . As similar to the above, we let H_s^L represents the event that target label l_s associated with x , and $\neg H_s^L$ represents the event that auxiliary label l_s not related to x , then the target classifier should has the following form:

$$h(x) = \{l_s | f(x, l_s) > 1, 1 \leq s \leq r\} \quad (5)$$

$$\text{where } f(x, l_s) = \frac{P(H_s^L | C_s^L, T^*)}{P(\neg H_s^L | C_s^L, T^*)} \quad (6)$$

In accordance with Bayes' theorem, $f(x, l_s)$ can be rewritten as the following form:

$$f(x, l_s) = \frac{P(H_s^L | C_s^L, T^*)}{P(\neg H_s^L | C_s^L, T^*)} = \frac{P(C_s^L, T^* | H_s^L) \cdot P(H_s^L)}{P(C_s^L, T^* | \neg H_s^L) \cdot P(\neg H_s^L)} \quad (7)$$

We assume that each label between independent in the respective hierarchy (auxiliary or target), then the following equation holds:

$$P(T^*) = \prod_{t_{j^*} \in T^*} P(H_{j^*}^T) \quad (8)$$

Therefore, $f(x, l_s)$ should further be rewritten like this:

$$f(x, l_s) = \frac{P(H_s^L) \cdot P(C_s^L, T^* | H_s^L)}{P(\neg H_s^L) \cdot P(C_s^L, T^* | \neg H_s^L)} = \frac{P(H_s^L) \cdot P(C_s^L | H_s^L) \cdot \prod_{t_{j^*} \in T^*} P(H_{j^*}^T | H_s^L)}{P(\neg H_s^L) \cdot P(C_s^L | \neg H_s^L) \cdot \prod_{t_{j^*} \in T^*} P(H_{j^*}^T | \neg H_s^L)} \quad (9)$$

Based on Bayes' theorem and Maximum Likelihood, $P(H_j^T | H_s^L)$ satisfies the equation:

$$P(H_j^T | H_s^L) = \frac{P(H_s^L | H_j^T) \cdot P(H_j^T)}{P(H_s^L)} = \frac{P(l_s | t_j) \cdot P(H_j^T)}{P(H_s^L)} \quad (10)$$

Where $P(l_s | t_j)$ can be obtained by computing the number of samples in the training set that has a corresponding label.

Finally, by integrating the formulas above, we can get the target classifier:

$$h(x) = \{l_s | f(x, l_s) > 1, 1 \leq s \leq r\} \quad (11)$$

$$\text{where } f(x, l_s) = \frac{P(C_s^L | H_s^L) \cdot P(H_s^L)^{(1-|T^*|)} \cdot \prod_{t_{j^*} \in T^*} P(l_s | t_{j^*}) \cdot P(H_{j^*}^T)}{P(C_s^L | \neg H_s^L) \cdot P(\neg H_s^L)^{(1-|T^*|)} \cdot \prod_{t_{j^*} \in T^*} P(\neg l_s | t_{j^*}) \cdot P(H_{j^*}^T)} \quad (12)$$

Where $|T^*|$ represents the number of predicted auxiliary labels.

Next, we introduce the estimation method of the probabilities above. For $P(H_j^T)$ and $P(H_s^L)$, we can use the frequency of each label that appears on the training set to estimate the probability. Specifically, given a training set $D = \{(x_i, T_i, L_i) | T_i \subseteq T, L_i \subseteq L, 1 \leq i \leq m\}$, we compute the priori probability according to the following method:

$$P(H_j^T) = \frac{\alpha^T + \sum_{i=1}^m \mathbb{1}[t_j \in T_i]}{2\alpha^T + m}; \quad P(\neg H_j^T) = 1 - P(H_j^T) \quad (1 \leq j \leq q) \quad (13)$$

$$P(H_s^L) = \frac{\alpha^L + \sum_{i=1}^m \mathbb{1}[l_s \in L_i]}{2\alpha^L + m}; \quad P(\neg H_s^L) = 1 - P(H_s^L) \quad (1 \leq s \leq r) \quad (14)$$

Where α^T and α^L are smoothing parameter used to balance the influence of the labels which does not appear. When they take the value of 1, the method is called Laplace smoothing which we generally used. For conditional probability $P(C_s^L | H_s^L)$ and $P(C_s^L | \neg H_s^L)$, we use the following formulas:

$$P(C_s^L | H_s^L) = \frac{\alpha^T + G_s^L(C_s^L)}{(k+1)\alpha^T + \sum_{i=0}^k G_s^L(i)} \quad (1 \leq s \leq r, 0 \leq C_s^L \leq k) \quad (15)$$

$$P(C_s^L | \neg H_s^L) = \frac{\alpha^T + \widetilde{G}_s^L(C_s^L)}{(k+1)\alpha^T + \sum_{i=0}^k \widetilde{G}_s^L(i)} \quad (1 \leq s \leq r, 0 \leq C_s^L \leq k) \quad (16)$$

Where $G_s^L(i)$ represents the number of samples in training set having such properties: there exactly are i samples in its k nearest neighbors which are associated with the label l_s , and itself is associated with l_s too. Similarly, $\widetilde{G}_s^L(i)$ represents the number of samples which have i samples in its k nearest neighbors associated with the label l_s but itself not associated with l_s .

4. Experiment

We randomly selected 34,799 customer reviews of mobile phones from *www.jd.com* website. After analyzed the characteristics of the data, we found the sentiments of these reviews can be divided into 6 major categories: satisfaction, disappointment, admiration, reproach, like, dislike. Furthermore, we found these data can be divided into three different categories by different object: (1) reviews for purchase experience; (2) reviews for service providers; (3) reviews for goods (mobile phone). Naturally, our goal is to analyze the sentiments of reviews, and the second classifications can as our auxiliary information.

In accordance with the above two different partitioning strategy, we twice for corpus annotation. The first is labeled in accordance with the object of the review, and the second in accordance with the sentiment of the review. Table 2 shows the number of object in the first tagged labels and Table 3 shows the distribution of each object. Table 4 shows the number of sentiments in the second tagged labels and Table 5 shows the distribution of each sentiment.

Table 2 Number of object in the first tagged labels

#Object	#Examples
0	3764
1	21502
2	8636
3	897

Table 3 Distribution of each object in the first tagged labels

Object	#Examples
Event	4074
Agent	9221
Object	28170

Table 4 Number of sentiments in the second tagged labels

#Sentiments	#Examples
0	3122
1	23559
2	7239
3	862
4	16
5	1

Table 5 Distribution of each sentiment in the second tagged labels

Sentiment	#Examples
satisfaction	3207
disappoint	1852
admiration	2192
reproach	7455
like	9449
dislike	16537

We split the data into training set which contains 27824 samples and test set which contains 6975 samples, and follow the method of Section 3 and set k value of 5. In the experiment, we use Jaccard similarity to calculate neighbor. We use 5 kinds of multi-label evaluation indicators to assess our results, they are Hamming Loss, One-Error, Coverage, Ranking Loss, Average Precision. Wherein, the first four smaller the evaluation value represents the better result, the fifth evaluation larger value indicates the better result. We compared the results of traditional ML-kNN method and our Hierarchical ML-kNN method, Table 6 shows the results of our experiments.

Table 6 The results of traditional ML-kNN and Hierarchical ML-kNN method

	ML-kNN	HML-kNN
Hamming Loss	0.0809	0.0809
One-Error	0.4852	0.4852
Coverage	1.8416	1.8400
Ranking Loss	0.2869	0.2865
Average Precision	0.4688	0.4692

As can be seen from the experimental results, hierarchical ML-kNN method works on Coverage, Ranking Loss, Average Precision indicators are better than the traditional ML-kNN method. It suggest that hierarchical ML-kNN method can effectively capture the different levels of association between the labels while adding auxiliary labels compared to traditional ML-kNN method.

5. Conclusion and future work

This paper analyzed the complex sentiments of customer reviews and proposed a hierarchical ML-kNN method based on traditional ML-kNN method. This method can effectively capture the correlation between labels by addition of auxiliary labels. Experimental results show that hierarchical ML-kNN method has better results than traditional ML-kNN method. Next, we consider lead hierarchical thinking into other multi-label classification method and analyze complex sentiments of different areas text.

Acknowledgments

This work was supported by the National Natural Science Foundation of China with grant No. 71231002.

References

- [1] Turney P D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews[J]. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2002:417—424.
- [2] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques[J]. Proceedings of Emnlp, 2009:79—86.
- [3] Taboada M, Brooke J, Tofiloski M, et al. Lexicon-based methods for sentiment analysis[J]. Computational linguistics, 2011, 37(2): 267-307.
- [4] Pang B, Lee L. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales[J]. Proceedings of the Acl, 2005:115-124.
- [5] Snyder B, Barzilay R. Multiple Aspect Ranking using the Good Grief Algorithm[C]// In Proc. 2010:300--307.
- [6] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
- [7] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
- [8] Liu S M, Chen J H. A multi-label classification based approach for sentiment classification[J]. Expert Systems with Applications, 2015, 42(3): 1083-1093.
- [9] Xu H, Yang W, Wang J. Hierarchical emotion classification and emotion component analysis on chinese micro-blog posts[J]. Expert Systems with Applications, 2015, 42(22): 8745-8752.
- [10] Tsoumakas G, Katakis I. Multi-label classification: An overview[J]. Dept. of Informatics, Aristotle University of Thessaloniki, Greece, 2006.
- [11] Zhang M L, Zhou Z H. A review on multi-label learning algorithms[J]. Knowledge and Data Engineering, IEEE Transactions on, 2014, 26(8): 1819-1837.
- [12] Boutell M R, Luo J, Shen X, et al. Learning multi-label scene classification[J]. Pattern recognition, 2004, 37(9): 1757-1771.
- [13] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification[J]. Machine learning, 2011, 85(3): 333-359.
- [14] Fürnkranz J, Hüllermeier E, Mencía E L, et al. Multilabel classification via calibrated label ranking[J]. Machine learning, 2008, 73(2): 133-153.
- [15] Tsoumakas G, Vlahavas I. Random k-labelsets: An ensemble method for multilabel classification[M]//Machine learning: ECML 2007. Springer Berlin Heidelberg, 2007: 406-417.
- [16] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning[J]. Pattern recognition, 2007, 40(7): 2038-2048.
- [17] Elisseeff A, Weston J. A kernel method for multi-labelled classification[C]//Advances in neural information processing systems. 2001: 681-687.
- [18] Zhang M L, Zhang K. Multi-label learning by exploiting label dependency[C]//Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010: 999-1008.