Multi-Label Chinese Comments Categorization: Comparison of Multi-Label Learning Algorithms

Jiahui He[1], Chaozhi Wang[1], Hongyu Wu[1], Leiming Yan[1, *] and Christian Lu[2]

Abstract: Multi-label text categorization refers to the problem of categorizing text through a multi-label learning algorithm. Text classification for Asian languages such as Chinese is different from work for other languages such as English which use spaces to separate words. Before classifying text, it is necessary to perform a word segmentation operation to convert a continuous language into a list of separate words and then convert it into a vector of a certain dimension. Generally, multi-label learning algorithms can be divided into two categories, problem transformation methods and adapted algorithms. This work will use customer's comments about some hotels as a training data set, which contains labels for all aspects of the hotel evaluation, aiming to analyze and compare the performance of various multi-label learning algorithms on Chinese text classification. The experiment involves three basic methods of problem transformation methods: Support Vector Machine, Random Forest, k-Nearest-Neighbor; and one adapted algorithm of Convolutional Neural Network. The experimental results show that the Support Vector Machine has better performance.

Keywords: Multi-label classification, Chinese text classification, problem transformation, adapted algorithms.

1 Introduction

The classification problem has always been a very important content in machine learning, and it has received extensive attention from research institutes and related industries. Today, with the growth of Internet sites, the amount of data in social media and users' fast-growing networks is growing at a daily rate. Since these data are usually unstructured, it is necessary to manage and organize these data through an accurate text classification system. This paper will classify and compare the reviews of the catering industry through different multi-label classification methods.

There are two main implications for multi-label classification of Chinese commentary information. First of all, it can help the relevant departments to monitor the public opinion and guide the public opinion to positive aspects according to the results. This is of great significance for early detection of network hot events and early warning of public opinion. In addition, it provides information to help consumers make decisions when they choose to buy goods online. When the volume of comments is large, the consumer can quickly

[1] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

[2] School of Information Technology, Deakin University, Victoria, Australia.

[*] Corresponding Author: Leiming Yan. Email: yan_leiming@163.com.

JNM. doi:10.32604/jnm.2019.06238

learn about the item by reading the tag of the comment. Multi-label classification of comment information brings a lot of convenience to consumers.

There are many commonly used machine learning algorithms in traditional text categorization, for example, Support Vector Machine [Tong and Koller (2001)], Naïve Bayes [Tang, Kay and He (2016)], and Random Forest [Wu, Ye, Zhang et al. (2014)]. These algorithms that enable text categorization simply assign a text to a category. However, in real life, it often happens that text can belong to more than one category. For example, movies under the "Actions" category can also appear under "Love", "Suspense" and other categories. Since each text may belong to multiple categories, the traditional single-label text classification algorithm cannot be directly used to solve the multi-label text classification problem, which makes solving the multi-label classification problem a challenge.

Many related studies have offered solutions to the problem of multi-label classification and applied them to various fields. The multi-label classification framework has important applications in the field of medical diagnosis. The multi-label feature selection method [Fang, Cai, Sun et al. (2018)] can improve the accuracy and reduce the false negative rate compared with the traditional feature selection method.

Algorithms for solving multi-label text classification problems are generally divided into two types, Algorithm adaptation and problem transformation methods. In the method of problem transformation, many learning methods have been introduced in the literature. For example, Binary Relevance [Boutell, Luo, Shen et al. (2004)], Label Powerset [Tsoumakas and Vlahavas (2007)]. In the algorithm adaptation method, many literatures propose to extend the prior art single label learning algorithm. For example, AdaBoost.MH [Schapire and Singer (2000)] and RFBoost [Al-Salemi, Noah and Ab Aziz (2016)] are multi-label Boosting algorithms, extending from AdaBoost [Freund and Schapire, 1997)]. IBLRML [Cheng and Hüllermeier, 2009)], MLkNN [Zhang and Zhou (2007)] and BRkNN [Spyromitros, Tsoumakas and Vlahavas (2008)] are multi-label classification algorithms extended from KNN.

The remainder of this paper is organized as follows: The second part introduces the work of two mainstream multi-label classification algorithms. The third section briefly describes the multi-label classification method used for experimental evaluation. The fourth part introduces and describes the collected data sets, the preprocessing of the data, the evaluation criteria of the experiment, the experimental environment and the analysis of the experimental results. The fifth part introduces the contribution and results of the paper and some future directions.

2 Related work

Algorithm adaptation and problem transformation methods are two main solutions to multi-label classification. Some algorithms which are originally designed for single-label classification have been adapted to multi-label problems, such as ML-KNN [Zhang and Zhou (2005)]. On the other hand, we can transform the original multi-label problems into one or several single-label problems so that we can use existing single-label learning methods to suit the needs.

2.1 Multi-label learning algorithms

Binary Relevance (BR) is a well-known approach which is based on the assumption that the labels are independent and it trains different models for different labels. Godbole et al. [Godbole and Sarawagi (2004)] created a two-stage classification process by stacking BR classification outputs along with the full original attribute space. We call the modified method as Meta-BR (MBR). MBR can take the correlations between labels into consideration. However, MBR is likely to increase additional iterations during the training as it uses meta classifier. It is reported that overweighting positive examples in BR models can solve the issue of class-label imbalance which is caused by label sparsity [Ráez, López and Steinberger (2004)]. The authors also mentioned that ignoring the rare labels and considering trimming BR based on performance may improve classification speed. It is also reported that a framework aimed at extracting shared subspace can obtain the correlation factor [Ji, Tang, Yu et al. (2008)]. Yet it will lead to a higher computational complexity. Similar to MBR, the classifier chains model (CC) [Cheng, Llermeier and Dembczynski (2010)] can also include the correlations but only demands a single iteration.

Ensembles of classifier chains (ECC) [Cheng and Hüllermeier (2009)] trained m CC models, where each model is allocated a random chain ordering. However, with the increase of the numbers of labels, the amount of calculation is too large to be feasible. New strategies like Compressed Sensing (CS) [Hsu, Kakade, Langford et al. (2009)] were raised, which assumes sparsity in the label set and encodes labels with a small amount of linear random projectors.

Another model called RAkEL system [Tsoumakas and Vlahavas (2007)] is proved to be more accurate than BR. RAkEL system uses random subsets of k labels to train m LC models. With appropriate values of m and k, RAkEL system can achieve a high accuracy.

2.2 Algorithm adaptation approaches

The simplest and frequently used strategy is to extend the single-label learning algorithms to solve multi-label tasks. AdaBoost MH and AdaBoost.MR proposed adaptation-based multi-label algorithms by Schapire et al. [Schapire and Singer (1999)]. These are extended from the well-known boosting algorithm AdaBoost [Freund and Schapire (1997)]. AdaBoost.MH is used to minimize training Hamming loss. AdaBoost.MR is used to generate assumptions based on ranking tags and placing the correct tags at the top of the ranking. The experimental results of a study conducted by Schapire et al. [Schapire and Singer (2000)] show that AdaBoost.MH performs better than AdaBoost.MR. There is another method to solve the multi-label classification problem, the problem transformation approaches which still depend on single-label classifiers. The large number of single label classifiers makes it difficult to determine which transformation method is the latest technology for multi-label classification. In this respect, the adaptive multi-label learning algorithm can be a good choice because the single-label algorithm is suitable for solving the multi-label problem directly.

3 Multi-label learning methods

This section introduces main types of multi-label learning methods which can be divided into two categories, problem transformation approaches and adapted algorithms.

3.1 Problem transformation approaches

This part describes useful algorithms that can transform multi-label problems into single-label problems.

3.1.1 Binary correlation

As one of the most straightforward algorithms, Binary Relevance (BR) [Boutell, Luo, Shen et al. (2004)] is widely used in multi-label classification. The whole task is divided into several binary tasks, each one is trained individually for each label, instances which not relevant to the known label will be considered as negative instances. Later, the unions of the individual binary classifiers with positive tags for an instance becomes the multiple labels of that instance.

3.1.2 Classifier chains

However, BR does not take the labels' correlations into consideration. The Classifier Chains (CC) method [Read, Pfahringer, Holmes et al. (2011)] trains several binary classifiers which are subsequently linked randomly as a chain. Each classifier will incorporate the labels which are predicted by the previous classifiers as additional information to classify a given unseen sample. There also exist disadvantages to this solution in that as the chain is linked randomly, it will decrease the accuracy of the classifier.

3.1.3 Label powerset

Random forests (RF) or **random decision forests** is a classification algorithm based on Bagging [Breiman (1996)] which operates by constructing a multitude of decision trees at training time. RF changes the original setting of decision tree by splitting each node using the best among a subset of predictors randomly chosen at there.

Support Vector Machine (SVM) [Vapnik (2013)] is a set of supervised learning methods used for classification, regression and outliers detection. Its simplest form, the linear SVM forms a hyperplane that separates a set of positive examples from negative ones with maximum margin. SVM performs well in high dimensional spaces and can still be effective in the circumstance where the number of dimensions is greater than the number of samples.

k-Nearest-Neighbors (kNN) Aha et al. [Aha, Kibler, Albert et al. (1991)] is an instance-based learning algorithm that only stores the training instances instead of attempting to construct a general internal model. Classification is computed from a simple majority vote of the nearest neighbors of each point. The similarity is defined according to a distance metric between two data points. One of the most widely used method to calculate similarity is the Euclidean distance:

$$\mathrm{d}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

3.2 Adapted algorithms

The following are the evaluated multi-label learning algorithms that are adapted from well-known single-label learning algorithms:

3.2.1 Multi-label k-Near

MLKNN [Zhang and Zhou (2007)] is one of the most well-known multi-label algorithms. The traditional k-Nearest Neighbor (KNN) is one of the most basic and simple algorithms in machine learning. The idea is very straightforward: If a sample has most of the k nearest distances' votes in the feature space, the sample belongs to that category. In MLKNN, the traditional KNN algorithm is adapted to make it suitable for multi-label learning. The Maximum A Posteriori (MAP) principle is used to determine the set of labels for a given instance based on the prior probability and posterior probability of the frequency of each tag within the KNN.

3.2.2 Instance-based logistic regression multi-label

The learning of IBLR-ML [Cheng and Hüllermeier (2009)] is adapted from the traditional multi-class algorithm KNN. In IBLR-ML, instance-based learning (IBL) [Aha, Kibler, Albert et al. (1991)] is combined with logistic regression. It allows the use of tags of neighbor examples as an additional attribute in a logistic regression scheme to capture interdependencies between class tags to give an estimate of multiple tags for a given instance.

3.2.3 Binary correlation KNN

Binary correlation KNN (BRKNN) [Spyromitros, Tsoumakas and Vlahavas (2008)] is another multi-label classifier for the KNN algorithm. BRKNN includes the BRKNN-a and BRKNN-b methods. The BRKNN-a method is for the BRKNN prediction process, which generates an empty class set for a new test case for predication. At this moment, the algorithm selects the largest class label based on the distribution of each class of the standard value, and sets its value to be positive. The class label set of each sample to be predicted contains at least one class label value. The BRKNN-b method is for the BRKNN prediction process, a new test sample needs to be given the prediction of classes. Its specific method is to calculate the K-nearest neighbors in the training data set for each example, and count the average class number of the K neighbors as the number of class element of the test sample that needs to be predicted.

3.2.4 RFBoost

AdaBoost.MH [Schapire and Singer (1999)] is a multi-label boosting algorithm that extends from the famous AdaBoost [Freund and Schapire (1997)]. The AdaBoost method is an iterative algorithm that adds a new weak classifier to each round until a predetermined sufficiently small error rate is reached. As a lifting algorithm, AdaBoost.MH works by iteratively constructing a weak classifier of a set of decision tree stubs. The final classifier is then constructed as a combination of selected weak classifiers. A disadvantage of AdaBoost.MH is that the computation time is linear with the number of the training features [Al-Salemi, Ab Aziz and Noah (2015a, b)].

Al-Salemi et al. [Al-Salemi, Noah and Ab Aziz (2016)] proposed an accelerated version of AdaBoost.MH, named "RFBoost". In RFBoost, functional rankings are used to rank training functions. Then, in each promotion round, only a few top ranked features are selected to build the weak classifier. This strategy makes RFBoost faster and more accurate than AdaBoost.MH [Al-Salemi, Ayob and Noah (2018)].

3.2.5 ML_DCCNN

Convolutional Neural Network (CNN) is a tool commonly used in natural language processing. A new model was proposed [Xiong, Shen, Wang et al. (2018)] for learning exercise vectors, and combined with CBOW models and CNNs to establish a new deep learning model. The experimental results show that the semantic relativity and accuracy of the new model in the segment vector space is better than the CBOW model. Sentiment classification of online reviews using CNN has proven to be effective and feasible [Zhang, Wang, Li et al. (2018)].

ML_DCCNN [Yu, Wang and Wu (2018)] is a multi-label classification model based on convolutional neural network. This model uses the powerful feature extraction ability of convolutional neural networks to automatically learn the characteristics that can describe the nature of data. ML_DCCNN uses the migration learning method to reduce the training time of the model, and at the same time improve the fully connected layer of the convolutional neural network, propose a two-channel neuron, and reduce the parameter amount of the fully connected layer. Compared with the traditional multi-label classification algorithm and the existing deep learning-based multi-label classification model, ML_DCCNN maintains high classification accuracy and effectively improves classification efficiency.

4 Experiments, results and dataset

This section elaborates the process of the experiment. The dataset and evaluation metrics were first introduced. Then compared the advantages and disadvantages of different methods and analyzed the results based on the experiments.

4.1 Dataset

4.1.1 Dataset collection and preparation

In the experiment, we used a dataset of user reviews for fine-grained sentiment analysis from the catering industry, containing 335K public user reviews from Dianping.com. The dataset builds a two-layer labeling system according to the granularity, which contains 6 categories and 20 fine-grained elements.

4.1.2 Data description

This dataset includes a two-layer labeling system. The first layer is the coarse- grained evaluation object, such as service and location. The second layer is the fine-grained emotion object, such as waiter’s attitude and wait time in service category.

Every element has four sentimental types: positive, Neutral, Negative and Not mentioned, which are labelled as 1, 0, -1 and-2. The dataset is divided into three parts: Training set, Verification set and Test set.

Table 1: The specific description of dataset

The first layer	The second layer
Location	traffic convenience
	distance from business district
	easy to find
Service	wait time
	waiter’s attitude
	parking convenience
	serving speed
Price	price level
	cost-effective
	discount
Environment	decoration
	noise
	space
	cleaness
Dish	portion
	taste
	look
	recommendation
Others	overall experience
	willing to consume again

Table 2: Summarization of dataset

Type	**Amount of comments**
Training set	105,000
Verification set	5,000
Test set	15,000

4.2 Evaluation metrics

$$\text{Precision p} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall r} = \frac{TP}{TP+FN} \tag{3}$$

$$F_1 = \frac{2\cdot p\cdot r}{p+r} = \frac{2\cdot TP}{2\cdot TP+FP+FN} \tag{4}$$

For L categories fc1; c2; . . .; cLg, there are two commonly used measures-micro-averaged F1 and macro-averaged:

$$\text{Micro} - \text{averaged } F_{1micro} = \frac{2\cdot\sum_{j=1}^{L} TP_{c_j}}{2\cdot\sum_{j=1}^{L} TP_{c_j}+\sum_{j=1}^{L} FP_{c_j}+\sum_{j=1}^{L} FN_{c_j}} \tag{5}$$

$$\text{Macro} - \text{averaged } F_{1marco} = \frac{1}{L}\sum_{j=1}^{L}\frac{2TP_{c_j}}{2TP_{c_j}+FP_{c_j}+FN_{c_j}} \tag{6}$$

Suppose that the real label classification be $y \in \{0,1\}^{n_{samples}\times n_{labels}}$, and the classifier prediction situation is $\hat{f} \in \mathbb{R}^{n_{samples}\times n_{labels}}$ and $rank_{ij} = \left|\{k: \hat{f}_{ik} \geq \hat{f}_{ij}\}\right|$.

$$D_{Hamming}(\hat{y_i}, y_i) = \frac{1}{L}\sum_{j=1}^{L} 1\left(\widehat{y_{ij}} \neq y_{ij}\right) \tag{7}$$

$$coverage(y, \hat{f}) = \frac{1}{n_{samples}}\sum_{i=1}^{n_{samples}} \max_{j:y_{ij}=1} rank_{ij} \tag{8}$$

$$LRAP(y, \hat{f}) = \frac{1}{n_{samples}}\sum_{i=1}^{n_{samples}} \frac{1}{|y_i|}\sum_{j:y_{ij}=1}\frac{|\mathcal{L}_{ij}|}{rank_{ij}} \tag{9}$$

$$ranking(y, \hat{f}) = \frac{1}{n_{samples}}\sum_{i=1}^{n_{samples}} \frac{1}{|y_i|(n_{labels}-|y_i|)}\left|\mathcal{L}_{ij}\right| \tag{10}$$

Table 3: The contingency matrix of text-category assignment

	Actual category	**Positive**	**Negative**
Predicted category	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

4.3 Results and discussion

In this part we discuss the results by each model in the experiment. The result metrics of different methods' experiments are listed in Tab. 4.

Table 4: The best-obtained evaluation metrics of all methods

Classifier	Hamming loss	Ranking loss	Micro-F1	Macro-F1	Coverage-error	Label_ranking_average _precision_score
LP-SVM	0.284100	0.356550	0.539919	0.202236	2.515200	**0.820856**
LP-RF	0.387950	0.494383	0.468015	**0.361421**	2.685800	0.693422
LP-KNN	0.348750	0.380600	**0.555308**	0.358996	**2.469000**	0.738467
CC-SVM	**0.282800**	**0.355033**	0.541058	0.200024	2.513800	0.821967
CC-RF	0.387950	0.494383	0.468015	0.361421	2.685800	0.693422
CC-KNN	0.341050	0.484017	0.456451	0.283083	2.674600	0.717317
MLPCNN	0.331400	0.364050	0.553309	0.351751	2.470800	0.765094

4.3.1 Loss measure

It illustrates the misclassification of the sample on the labels, which means that the corresponding true label does not appear in the predicted label list. The lower loss means the better performance of the classifier.

Tab. 4 shows that the classification performance of the evaluated methods measured by hamming loss and ranking loss. Among all the classifiers, CC-SVM performed best especially when using poly as its core function. Meanwhile, MLP came after SVM and outperformed KNN. And RF seems to be the worst way to handle this kind of dataset.

Comparing the performance of these methods, SVM, generally, achieved the best performance when handling these data.

4.3.2 F1 measure

Only referring to Precision and Recall does not satisfy the evaluation of the performance of the classifier in different scenarios. For example, when the sample size is very small but the accuracy of the classifier is high, Precision will be high and Recall will be low.

Therefore, the reconciliation mean F1-score of Recall and Precision is proposed. It is considered that Recall and Precision are equally important. Compared with using Precision and Recall alone, F1-score can measure classification performance more reliably. The F1-score is divided into Micro F1-score and Macro F1-score. According to the experimental results in Tab. 4 and Fig. 1, it can be clearly seen that LP-KNN obtained the highest Micro F1-score and LP-RF had the worst performance. In the comparison of Macro F1-score, LP-RF achieved the best performance and CC-SVM performed the worst.

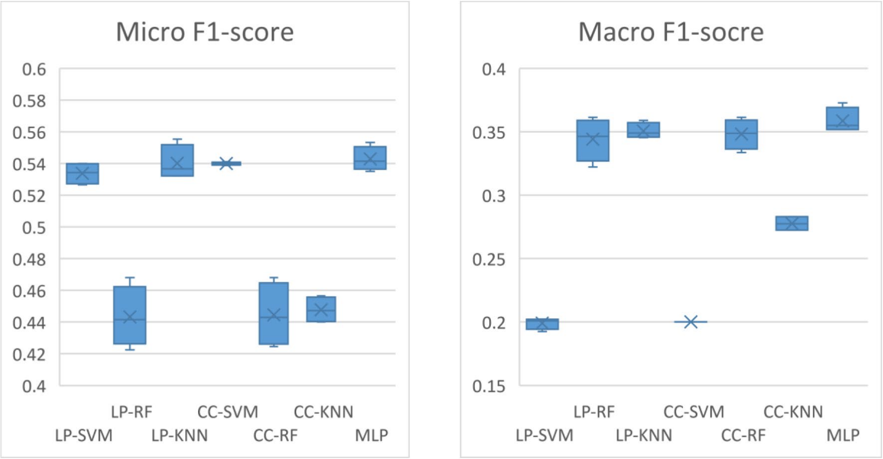

Figure 1: Boxplots of macro and micro F1 scores

4.3.3 Coverage-error measure

The data sets used in the experiments were all generated by a sample survey, which resulted in coverage errors due to the inability to cover each individual in the parent population at the time of the survey. In theory, if there is no difference between the characteristics of the group of individuals not covered and the other individuals in the mother's body, there is no coverage error even if there is a blind spot. However, coverage errors will occur when there are significant differences between covered and uncovered individuals.

According to the experimental results in Tab. 4, it can be clearly seen that LP-KNN obtained the best performance in Converge-error and LP-RF obtained the worst performance.

5 Conclusion

This work compared the classification performance of several common multi-label classification methods in restaurant evaluation. Common multi-label classification algorithms used in this paper are Support Vector Machines, k-Nearest-Neighbors, Convolutional Neural Networks, and Random Forests. The evaluation criteria used can be divided into three categories: loss measure, F1 measure, and coverage error measure.

The experimental result shows that the accuracy of SVM method classification is higher than other methods for the evaluation of such text sentiment classification in restaurants. The RF method in this experiment is not applicable to the classification herein. In future research, we will use more multi-label methods. We will also pay more attention to the preprocessing stage, such as using different stemming algorithms, feature weighting schemes, and feature reduction metrics.

Acknowledgement: This work is supported by the NSFC (Grant Nos. 61772281, 61703212, 61602254); Jiangsu Province Natural Science Foundation [grant number BK2160968]; the Priority Academic Program Development of Jiangsu Higher Edu-cation Institutions (PAPD) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

References

Aha, D. W.; Kibler, D.; Albert, M. K. (1991): Instance-based learning algorithms. *Machine Learning*, vol. 6, no. 1, pp. 37-66.

Al-Salemi, B.; Ayob, M.; Noah, S. A. M. (2018): Feature ranking for enhancing boosting-based multi-label text categorization. *Expert Systems with Applications*, vol. 113, pp. 531-543.

Al-Salemi, B.; Aziz, M. J. A.; Noah, S. A. (2015b): LDA-AdaBoost.MH: accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization. *Journal of Information Science*, vol. 41, no. 1, pp. 27-40.

Al-Salemi, B.; Noah, M. S.; Ab Aziz, M. J. (2016): RFBoost: an improved multi-label boosting algorithm and its application to text categorisation. *KnowledgeBased Systems*, vol. 103(Supplement C), pp. 104-117.

Arturo, M. R.; Luís, A. U. L.; Steinberger, R. (2004): Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. *Lecture Notes in Computer Science*, vol. 3230, pp. 1-12.

Breiman, L. (2001): Random forests. *Machine Learning*, vol. 45, pp. 5-32.

Boutell, M. R.; Luo, J.; Shen, X.; Brown, C. M. (2004): Learning multi-label scene classification. pattern Recognit. *Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771.

Cheng, W.; Eyke, H. (2009): Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, vol. 76, vol. 2-3, pp. 211-225.

Cheng, W.; Hüllermeier, E. (2009): Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, vol. 76, no. 2-3, pp. 211-225.

Cheng, W.; Llermeier, E. H.; Dembczynski, K. (2010): Bayes optimal multilabel classification via probabilistic classifier chains. *International Conference on Machine Learning.*

Fang, S.; Cai, Z.; Sun, W.; Liu, A.; Liu, F. et al. (2018): Feature selection method based on class discriminative degree for intelligent medical diagnosis. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 419-433.

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*, vol. 55, no. 1, pp. 119-139.

Godbole, S.; Sarawagi, S. (2004): Discriminative methods for multi-labeled classification. *Advances in Knowledge Discovery and Data Mining*, pp. 22-30.

Hsu, D.; Kakade, S. M.; Langford, J.; Zhang, T. (2009): Multi-label prediction via compressed sensing. *International Conference on Neural Information Processing Systems*, pp. 1-9.

Ji, S.; Tang, L.; Yu, S.; Ye, J. (2008): Extracting shared subspace for multi-label classification. *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Ráez, A. M.; López, L. A. U.; Steinberger, R. (2004): Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. *EsTAL: 4th International Conference on Advances in Natural Language Processing*, pp. 1-12.

Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. (2011): Classifier chains for multi-label classification. *Machine Learning*, vol. 85, no. 3, pp. 333-359.

Schapire, R. E.; Singer, Y. (1999): Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, vol. 37, no. 3, pp. 297-336.

Schapire, R. E.; Singer, Y. (2000): Boostexter: a boosting-based system for text categorization. *Machine Learning*, vol. 39, no. 2-3, pp. 135-168.

Spyromitros, E.; Tsoumakas, G.; Vlahavas, I. (2008): An empirical study of lazy multilabel classification algorithms. *Artificial Intelligence: Theories, Models and Applications*, pp. 401-406.

Tsoumakas, G.; Vlahavas, I. (2007): Random k-Labelsets: an ensemble method for multilabel classification. *European Conference on Machine Learning*, pp. 406-417.

Vapnik, V. (1995): *The Nature of Statistical Learning Theory.* Springer.

Xiong, Z.; Shen, Q.; Wang, Y.; Zhu, C. (2018): Paragraph vector representation based on word to vector and CNN learning. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 213-227.

Yu, Y.; Wang, L.; Wu, X. (2018): A multi-label classification algorithm based on improved convolutional neural network. *CAAI Transactions on Intelligent Systems*.

Zhang, M. L.; Zhou, Z. H. (2005): A *K*-nearest neighbor based algorithm for multi-label classification. *IEEE International Conference on Granular Computing*, vol. 2.

Zhang, M. L.; Zhou, Z. H. (2007): ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048.

Zhang, Y.; Wang, Q.; Li, Y.; Wu, X. (2018): Sentiment classification based on piecewise pooling convolutional neural network. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 285-297.