K-Partition Ensemble Multi-label Classifier for Insurance Purchase Prediction

Jiayu Zhou, Yong Guo*, Yanqing Ye and Jiang Jiang
College of Systems Engineering, National University of Defense Technology, Changsha, China

*Corresponding author. Email: yguo@nudt.edu.cn

Abstract. Individual insurance purchase prediction can help effectively and accurately advertise to maximize sales. Most current insurance purchase prediction model only considers whether a customer will buy certain insurance. In this way, the prediction will be time-consuming with the rapid growth of the number of insurance products. In order to provide more effective and efficient marketing methods for insurance companies, this work proposes a K-Partition Ensemble Multi-Label classification model to predict the customer’s possible future insurance purchase. First, by transforming the insurance purchase prediction problem into a multi-label classification problem, the balance between features and labels of data division in the insurance purchasing dataset is explored. Second, a k-partition ensemble multi-label classification model is introduced, where each distinct label constitutes in the training set as a new category of a single-label classification task, and the random forest is used for multi-class classification. The empirical test is carried out using the Insurance Company Case data from CoIL Challenge 2000. We find prediction classifiers perform the best when the number of labels is around 20. Empirical evidence indicates that our model manages to improve substantially over other 3 classical multi-label classification algorithms with relatively little time, especially in domains with a large number of labels. The research results also provide a new idea and useful reference for the application in specific fields construction of data models based on the multi-label evaluation.

Keywords: insurance purchase prediction (IPP), multi-label classification, supervised learning, K-partition ensemble multi-label (KPEML) classification.

1. Introduction
To find more potential customers for increasing sales, Insurance Purchasing Prediction (IPP) is necessary for insurance companies. Individual IPP helps effectively and accurately advertise to constantly attract new customers buying different products. Many methods have been used in IPP so far. With various commercial issues generating a large amount of data in different activities, the superiority of data-driven analysis methods promotes the application of machine learning in insurance purchase prediction [1].

At present, many prediction methods only consider whether the user will buy certain insurance [2]. However, in the actual process of product promotion, the policy of identifying potential purchasers is to take out many kinds of products for analysis and select certain products suitable for customers. This requires the prediction of whether a client will purchase multiple products. To improve sales
efficiency, more products are supposed to be considered at once. Such a prediction problem is a multi-label classification problem.

The multi-label classification problem appears in many fields, such as video or image annotation [3, 4], multi-label text classification [5] and music genre classification [6]. The common challenges in solving this classification problem are 1) calculation efficiency, 2) prediction accuracy, and 3) evaluation reliability. As the number of labels increases, the number of possible label combinations grows exponentially, resulting in the overfitting of classifiers and much longer computation time. For example, for a dataset with 20 labels, the number of possible label combinations will exceed one million (i.e. $2^{20}$). The key problem of multi-label data prediction is its huge output space size. Specifically, for the multi-label classification problem of IPP, new challenges have emerged, including 1) difficulty in obtaining the product retaining data as features, 2) inaccurate predictions caused by fewer features, and 3) low calculation efficiency when considering multiple labels. What’s more, the number of predicted labels will lead to various results. To choose the proper number of features and labels, we transform the original insurance data into different datasets with different numbers of labels. We try to find the balance between the number of features and that of labels in multi-label classification for a more efficient and accurate model.

Currently, the main methods for multi-label classification are Problem Transformation (PT). PT aims to transform a multi-label classification problem into one multi-class classification or a set of single-label classification. It uses the existing single-label classification method to solve the problem [7]. PT includes Binary Relevance (BR) [8], Classifier Chain (CC) [9] and Label Powerset (LP) [10]. Among them, the BR and CC algorithm takes full advantage of high-performance traditional single-label classifiers. CC algorithm also regards the labels as features in turn according to the "Bayesian chain rules". However, these two methods will lead to higher computational costs when the label space is large. LP algorithm treats each unique label constitution existing in the multi-label training set as one label category of the new single-class classification. However, the multi-class classification cannot predict new label combinations and its accuracy is not high.

Selecting a classifier for a multi-label problem requires two considerations: performance and efficiency. This article comprehensively considers the above two points. Referring to the RAKEL [11] algorithm, we make the ensemble of RF and partition representation and proposes a K-Partition Ensemble Multi-Label (KPEML) algorithm for multi-label classification. Firstly, the problem is decomposed into sub-problems between labels to improve generalization quality. Secondly, the model considers each distinct combination of labels that exist in the training set as a different class value of a single-label classification task. And finally, a bagging method, random forest, is carried out for multi-class classification. In experiments, by comparing the KPEML algorithm with other algorithms, the results show that our proposal not only beats the compared methods over most measures on Insurance Company Case datasets but also enjoys relatively little calculation time.

2. Background

One of the improved bagging algorithms, Random Forest (RF) is a classifier using multiple decision trees to train and predict samples [12]. It greatly boosts the performance in the final model at the expense of a small increase in bias and some loss of interpretability [13]. In the RF model, S new datasets are obtained after randomly selecting S times from the original dataset. The decision tree algorithm is applied to each dataset to obtain S classifiers, then the category with the most votes in the classifier's voting results is selected as the final classification result. RF provides better fit for the estimation and validation sample in majority single-label classification problems [14, 15]. It has been widely used in practical applications due to its excellent classification results. That’s why we choose RF as our base classifier.

In this paper, we compare our model with three typical PT methods, BR, CC, and LP. Among them, the BR algorithm converts a multi-label dataset into L single-label datasets and carries out L classifiers separately. Each L classifier corresponds to an original label. Although it is designed to take full advantage of high-performance traditional single-label classifiers, it will lead to higher
computational costs when the label space is large. Considering the relationship between neighbor labels, CC regards the labels as features in turn. The n-th classification is performed taking into account the output of the last n-1 classifications. The subsequent binary classifiers in the chain are built upon the predictions of preceding ones according to the "Bayesian chain rules". It also converts a multi-label dataset into L single-label datasets which is the same as BR. LP treats each unique label constitution existing in the multi-label training set as the label category of the new single-label multi-class classification. However, the multi-class classification is challenged by a large number of labels and training examples. It cannot predict new label combinations and its accuracy is not high.

3. K-partition ensemble multi-label algorithm

We present the IPP problem as a multi-label model. In the d-dimensional feature space, given the input dataset \( X \subseteq \mathbb{R}^d \) including all customers’ related information, \( x = \{x_1, x_2, ..., x_d\} \in X \) represents a sample, where \( x_i \) represents the i-th dimension feature of sample \( x \). \( L = \{l_1, l_2, ..., l_q\} \) represents a label set with q possible class labels, which represents whether various insurances are purchased, and \( q \) is the number of labels. Thus, the multi-label data set can be defined as

\[
D = \{(x, Y)|x \in X, Y \subseteq L\}
\]

where \( Y = \{y_1, y_2, ..., y_m\} \) is the subset of labels corresponding to the sample \( x \), which represents the insurance group purchased by a customer.

3.1. K-partition with labels

As most of the labels are very infrequent in the samples, it is a good way to combine some labels together so that there are fewer new labels remaining. We can reduce the running time by transforming the multi-label problem into a multi-class classification problem as an LP algorithm. However, it cannot predict new label combinations out of training set, and the results of it are not satisfied with low generalization performance. To solve this problem, we divide the labels randomly into partitions first. Through partitioning, each partition is only \( 1/k \) of the original labels. Thus, it greatly decreases the number of possible label combinations.

In each partition, it considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task. We obtain the label combination as a new label category. Each classifier is trained using a different small random subset of the set of labels. Then we find all label combinations mode and outputs the most probable class, which actually represents a set of labels.

By dividing the labels into k partitions, it transforms the multi-label problem into an easier single-label learning problem to deal with. What’s more, the new single-label dataset is more balanced in its multi-class.

3.2. K-partition ensemble multi-label classification model

Comprehensively considering both efficiency and performance of the multi-label classifier, we make the ensemble of partition representation and RF algorithm and proposes K-Partition Ensemble Multi-Label (KPEML) classification algorithm. First of all, the model randomly divides a large set of labels into a number of small-sized label partitions. Secondly, for each of the label partitions, find unique label combinations in partition i, and treat them as the category of single-label multi-class classification. Finally, a multi-label classifier is trained and then the multi-class model is newly-built, by which the most similar label combination is found as its prediction. The overall procedures of KPEML are shown in Figure 1.
The overall process of k-partition and transforming the problem is as follows:

1. Dividing all of the labels into partitions randomly. Each partition represents a smaller multi-label sub-problem.
2. Training a sample of the base RF classifier for each label space subset in the partition.
3. A ranking of the labels is produced by averaging the 0-1 predictions of each model per considered label. Predicting the result with each of the sub-classifiers and returning the sum of their results.

The space complexity of the traditional single-label algorithm is $O(n^l)$, where $n$ denotes the number of samples and $l$ presents the number of labels. While the space complexity of the multi-label algorithm is $O(ln)$. We can see from the space complexity expressions that, as the number of labels increases, the complexity will become lower. The computational complexity of LP depends on the computational complexity of the underlying single-label classification algorithm concerning the number of examples and the number of classes. The computational complexity KPEML will be less than $k$ times that of LP.

**4. Experiment**

**4.1. Data description**

The dataset we use in the article comes from the CoIL 2000 Challenge with information about insurance company customers [16]. The data contains 86 variables, including product usage data and socio-demographic data. The last 43 variables represent whether a client buys certain insurance or not, which can be treated as labels.

Modeling the insurance purchase problem to machine learning problems. Each row in the dataset represents a sample of the client. Each column represents a customer’s personal information socio-demographic and purchase history feature. Choose several issues of insurance purchasing as labels, then the model can be treated as multi-label classification.

There are $q$ features directly related to insurance purchasing in the datasets $\{l_1, ..., l_q\}$. Firstly, we choose the last two features $\{l_{q-1}, l_q\}$ as labels and others as features. And during the next time, the last third feature is added into the label set $\{l_{q-2}, l_{q-1}, l_q\}$, and so on. The selected $m$ labels form a new collection $\{l_m, ..., l_q\}$. Whether to purchase other insurance or not is used as features of the datasets along with users’ personal information socio-demographic data. Then we divide each label set into $k$ partition. In each partition, the multi-label problem is transformed to a single-label multi-class classification model. RF is used as a classifier to predict the value of each label. The process described above is shown in Figure 2.
Figure 2. The process of data division.

In the dataset, we set the value to 1 as long as a client buys one insurance product, no matter how many he buys. We count the same 1 or 0 value in the last 43 columns. The mode similarity matrix of insurance purchase is shown in Figure 3. If we find the status of two labels always the same or the opposite, there may be a great possibility of suspecting a connection between them. It’s beneficial for LP to determine the multi-class category.

Figure 3. The mode similarity matrix of insurance purchase.

The color of labels heat-map is light universally, which suggests there may be some label combination patterns.

4.2. Evaluation and comparison of different dataset structure

The platform in this experiment is Mac OS, 2.3 GHz Intel Core, 8 GB CPU. The development environment is python 3.7. We use RF as the base classifiers of the BR, CC, and LP these three multi-label classifiers for fair. In the RF model, the number of trees, max-depth, min samples split is tested with the grid search method and all other parameters are left as default, such as max_features (default is auto), min_samples_split (default is 2) and min_samples_leaf (default is 1). In all 4 models, we set 100 decision trees in a forest and use the value of Entropy to calculate properties and select the most appropriate node. Each experiment will be performed 10 times and we calculate the average and standard deviation.

According to the basis of measurement, there are two major evaluation methods for multi-label classification. The example-based evaluation method calculates the average difference between the actual label set and the predicted label set. The label-based evaluation method divides the evaluation process into individual evaluations for each label and calculates their average. Among them, example-based evaluation includes subset Accuracy, hamming Loss, accuracy, precision, recall, and
The label-based evaluation includes accuracyMacro, accuracyMicro, precisionMacro, precisionMicro, recallMacro, recallMicro, fbetaMacro, and fbetaMicro. These standards measure performance in different aspects.

Confusion matrix is the basis of the evaluation variables. It counts the number of positive and negative sample data in each type of real and prediction results. TP (True Positive) indicates that a customer who has purchased an insurance product is predicted to buy. TN (True Negative) represents a customer who did not purchase an insurance product is predicted not to buy. FP (False Positive) indicates that a customer who did not buy an insurance product is predicted that he would buy. FN (False Negative) indicates that a customer who did not buy it is predicted that they. In specific, accuracy calculation can be modified as follows:

\[
\text{Accuracy}(TP_j, FP_j, TN_j, FN_j) = \frac{FP_j + TN_j}{TP_j + FP_j + TN_j + FN_j}
\] (1)

Hamming loss of the model can be modified as follows:

\[
hloss(h) = \frac{1}{p} \sum_{i=1}^{q} \frac{1}{q} |h(x_i)\Delta Y_i|
\] (2)

The precision of the model can be modified as follows:

\[
\text{Precision}(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FP_j}
\] (3)

Recall of the model can be modified as follows:

\[
\text{Recall}(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FN_j}
\] (4)

In the following experiments, reciprocal 2-43 features are taking respectively as labels for experiments. We performed ten experiments on each algorithm. Record the average measurements and standard deviations of 10 testing experiments for comparative studies. The 14 evaluation indicators of subset accuracy, hamming loss, accuracy, precision, recall, f1 score, accuracy Macro, accuracy Micro, precision Macro, precision Micro, recall Macro, recall Micro, f1 Macro and f1 Micro are used as the evaluation criteria for the experimental results. The experimental results of BR, CC, LP, and KPEML are shown respectively in Figure 4, Figure 5, Figure 6, and Figure 7.
The overall results of the four methods are roughly the same. All evaluation reached their peak when the number of labels is between 18 and 21. Then macro f1 score, macro recall, and macro precision started to decline. This is because the macro score is the average of n scores, which is heavily influenced by the small number of samples. When the number of labels is 26, the evaluation results of all evaluation indicators begin to decline except for hamming loss. We found that when the
number of multi-label is about 20, the insurance purchasing data set of the CoIL 2000 Challenge achieves the best results.

To compare the efficiency of the algorithms, the execution times of BR, CC, LP and KPEML algorithms are shown in Figure 8. It can be found that as the number of labels increases, the running time of algorithms is generally getting longer. Among the four algorithms, BR and CC have the longest running time. This is because BR is calculated as a single-label each time. CC has to add a label as a feature to the training each time. These two algorithms will calculate \( l \) times, where \( l \) is the number of labels. As a multi-class classification of problem transformation, LP has the shortest running time. For KPEML method considers label correlation and divide the labels, it takes a little longer time than LP. However, compared with BR and CC algorithms, KPEML takes less time.

![Figure 8. Running time of 4 algorithms.](image)

To compare the results of KPEML with BR, CC and LP classifiers, Table 1 shows the accuracy, hamming loss, precision, recall, and f1 score of four multi-label classification algorithm training with 20 labels. The smaller the value of the hamming loss the better. While the larger the value of other evaluation indexes the better. Among the five evaluation indicators, KPEML achieved the best value in all three indicators. The last column reports the average results ranking of 4 algorithms. KPEML results have the highest average ranking, with 1.4. BR is second, with an average rank of 2.2. However, BR runs the longest.

|                | accuracy    | h-loss | precision  | recall       | f1_score     | rank |
|----------------|-------------|--------|------------|--------------|--------------|------|
| BR             | 0.919±0.0012| 0.004  | 0.837±0.0008| 0.8569±0.0005| 0.843±0.0011 | 2    |
| CC             | 0.912±0.0009| 0.004  | 0.856±0.0007| 0.8402±0     | 0.848±0.0003 | 2.2  |
| LP             | 0.858±0.0031| 0.008  | 0.85±0.0007 | 0.8121±0.0011| 0.831±0.0007 | 3.8  |
| KPEML          | 0.915±0.0023| 0.004  | 0.857±0.0014| 0.8408±0.0014| 0.849±0.0014 | 1.4  |

4.3. Effect of k-partition size on experimental results

Based on the previous experiment, the result is the best when the number of labels is 20. The first 64 columns of the data set are sample features, and the last 20 columns are the predicted labels. We then conduct the experiment by changing the value of \( k \). The experimental accuracy and the running time are shown in Figure 9 and Figure 10, respectively. Through the experiments, we found that the smaller the k-partition, the higher the accuracy value that is finally predicted, but the longer it will take. When \( k = 5 \), the accuracy reaches a relatively high level with the relatively short time required, which is consistent with the selection of \( k = 5 \) in previous experiments.
5. Conclusions
Insurance purchase prediction is supposed to consider products as many as possible to improve sales efficiency. In order to better solve the insurance purchase problem, this paper proposes a k-partition ensemble multi-label classifier by transforming the problem into a multi-label classification problem. Besides, this paper explores the relationship between features and the number of labels in multi-label evaluation problems and solves the problem of dataset construction in actual insurance purchasing. Through the k-partition method and the ensemble with RF, the efficiency and prediction results of multi-label classification are further improved. The feasibility and effectiveness of the method are verified with actual cases with real insurance purchasing background. Intelligently solve the problem of minimizing the number of features and maximizing the number of labels, and provide a reference for the construction of data sets for practical applications in specific fields. In theory, we explore the balance between features and labels. In practice, we solve the problem of data analysis and multi-label application in practical insurance purchase prediction problems and find a classification method equipped with both highest efficiency and better results. It achieved relatively the high performance in 43 differently divided datasets. The running time is reduced and the calculation efficiency is greatly improved as well, which ultimately ensures the implementation of actual production and living tasks and the improvement of work efficiency.

Acknowledgment
This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 71671186 and 71571185.
Reference

[1] Rama Devi Burri, Ram Burri, Ramesh Reddy Bojja, Srinivasa Rao Buruga, Insurance Claim Analysis Using Machine Learning Algorithms. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN- 2278-3075, 8 (6S4), April 2019.

[2] Mater, A. T., Studiorum Universitá, Degli Studi, Di Bologna, et al. Comparison of Data Mining Techniques for Insurance Claim Prediction. (2011).

[3] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, et al. Correlative multi-label video annotation. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 2007, 17–26.

[4] M. Wang, X. Zhou, and T.-S. Chua. Automatic image annotation via local multi-label classification. In Proceedings of the 7th ACM International Conference on Image and Video Retrieval, Niagara Falls, Canada, 2008, 17–26.

[5] I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. In Proceedings of the ECML PKDD 2008 Discovery Challenge, Antwerp, Belgium, 2008, 75–83.

[6] C. Sanden and J. Z. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 2011, 705–714.

[7] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. International Journal of Data Warehousing and Mining, 3 (2007) 1–13.

[8] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, Learning multi-label scene classification. Pattern Recognition, 37 (2004), 1757–1771.

[9] J. Read, B. Pfahringer, G. Holmes, and E. Frank, Classifier chains for multi-label classification. Machine Learning, 85 (2011), 333-359.

[10] G. Tsoumakas and I. Vlahavas. Random k-labelsets: an ensemble method for multilabel classification. In Lecture Notes in Artificial Intelligence, 4701, 2007, 406–417.

[11] G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-Labelsets for Multi-Label Classification. IEEE Transactions on Knowledge and Data Engineering, 23(7) (2010), 1079–1089.

[12] Breiman, L. Random Forests. Machine Learning, 45, 5–32 (2001).

[13] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[14] Bart Larivière, Dirk Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications, 29 (2) (2005), 472-484.

[15] Gaspar Cano, Jose Garcia-Rodriguez, Alberto Garcia-Garcia, et al. Automatic selection of molecular descriptors using random forest: Application to drug discovery. Expert Systems with Applications, 72 (2017), 151-159.

[16] P. van der Putten and M. van Someren (eds). CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report, 2000-09. June 22, 2000.