
Multi-Label Learning with Class-Based Features Using Extended Centroid-Based Classification Technique (CCBF)

P. R. Suganya Devi*, R. Baskaran and S. Abirami

Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai

Abstract

Real world applications, such as news feeds categorization deal with multi-label classification problem, where the objects are associated with multiple class labels and each object is represented by a single instance (feature vector). In this paper, a new algorithm adaptation method called centroid-based multi-label classification using class-based features (CCBF) algorithm has been proposed to tackle the multi-label classification problem. It includes class-based feature vectors generation and local label correlations exploitation. In the testing stage, centroid-based classification algorithm is extended for multi-label classification problem. Experiments on reuters multi-label dataset with 103 labels demonstrate the performance and efficiency of CCBF algorithm and the result is compared with those obtained using other multi-label classification algorithms. The CCBF algorithm obtains competitive F measures with respect to the most accurate algorithms.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Class-based features; Cure clustering algorithm; Document classification; Label correlations; Multi-label learning.

1. Introduction

Multi-label learning deals with objects having multiple class labels and each object is represented by one single instance. The task is to learn a model which can predict a set of possible labels for an unseen object. For example, given class labels Asia, N. America, S. America, Europe and Australia, a news article about U.S troops in Bosnia may be labeled with both N. America and Europe classes. Multi-label learning has been applied to a variety of domains, such as text classification, image annotation, video annotation, social network and music categorization into emotions, bioinformatics, etc.

A common approach to multi-label classification is problem transformation, in which a multi-label problem is transformed into one or more single-label problems. The alternative to problem transformation is algorithm adaptation which modifies an existing single-label classification algorithm for multi-label classification. The common strategy adopted by existing approaches is that all the class labels are discriminated based on identical feature representation of the object. Also, existing approaches consider global label correlations only. However, using identical feature representation is inadequate to discriminate different class labels, as different class labels in the label space may carry

*Corresponding author.
E-mail address: suganya.rvks5@gmail.com

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.047
specific characteristics of their own. Also, different examples may share different label correlations. The main goal of this work is to improve the classification accuracy by considering class-based features and local label correlations.

The rest of the paper is organized as follows. Section 2 presents existing work on multi-label learning and further relevant literature. Section 3 presents the proposed framework for multi-label learning. The experimental results are discussed in section 4. Section 5 is a brief conclusion of this work.

2. Related Work

In the past decades, many well-established methods have been proposed to solve multi-label learning problems in various domains. All these methods can be divided into two categories: Problem Transformation Methods (fitting data to algorithm) and Algorithm Adaptation Methods (fitting algorithm to data).

Problem Transformation Methods convert the multi-label problem into a set of binary classification problems. Binary relevance (BR) method\(^1\) takes each class label as an independent binary problem. Dependent binary relevance (DBR) learning\(^8\) combines properties of both, chaining and stacking. The limitation of these binary relevance methods is that the computational complexity is worst. EPS method\(^9\) is concentrated on the concept of treating sets of labels as single labels. It achieves better performance, and trains much faster than other multi-label methods. Label powerset (LP) method\(^2\) considers label correlation by combining the unique set of class labels. But it is usually unfeasible for practical application, because it generates a huge number of class labels.

Algorithm Adaptation Methods modify traditional single label learning algorithms for multi-label learning, which can handle multi-label data directly. Ricardo\(^12\) proposed a method composed of an online procedure. Documents are classified using statistics computed from labeled instances. The limitation is that context is not considered in the feature vector. ML-kNN\(^7\) is Multi-Label k Nearest Neighbor which is extended from the standard kNN algorithm. Tsoumakas\(^1\) implements BRkNN algorithm and compares different multi-label classification algorithms.

Tan\(^13\) proposed a novel batch-updated approach which takes advantage of errors to update the model by batch. But it can be applied only to classic train/test problems. Yu\(^15\) proposed two novel multi-label classification algorithms, called multi-label classification using rough sets (MLRS) and MLRS using local correlation (MLRS-LC). They achieve promising performance when compared with other multi-label learning algorithms. Ren\(^11\) introduced class-indexing-based term-weighting approach, in which the inverse class frequency (ICF) is incorporated to generate more informative terms. Vale\(^14\) proposed a class-based feature selection method. This method chooses the attributes that are important for a specific class. Qian\(^16\) proposed CURE-NS (CURE with new shrinking scheme), which uses CURE clustering algorithm and uses the difference of density values of the representative points to determine the direction and distance of shrinking.

3. CCBF Multi-Label Learning

This paper proposes a strategy to learn from multi-label data, where class-based features and local label correlations are exploited. Finally, centroid-based classifier is extended for multi-label classification problem. Class-based features are considered to benefit the discrimination of different class labels. They are generated by performing clustering on positive examples and on negative examples for each class label. Cluster centroids are building blocks for generating modified class-based feature vectors are computed. For performing clustering analysis, CURE clustering algorithm is used. Then class-based feature vectors, exclusively for multi-label classification, are generated for each data point. Local label correlations are considered to improve the performance. For this, possible label subsets are recognized and training documents are grouped to the corresponding label subset. Using the modified class-based feature vectors and labels of each data point, label space is partitioned. Then, prototypes for each partition will be computed. A map function is used to automatically accommodate the incoming document in a region of the partition using Euclidean metric. Then, centroid-based classifier for multi-label classification is used to output a set of labels, according to the region. The overall architecture is shown in the following Fig. 1.

Processes involved:

- Cure clustering analysis.
- Class-based feature vectors construction.
3.1 Cure clustering analysis

CURE (Clustering Using Representatives) is a data clustering algorithm. To avoid the problems with the shape of the clusters, CURE uses a hierarchical clustering algorithm. In CURE, \( n \) number of distant points of a cluster are selected and shrunk towards the centroid by a fraction \( \alpha \), which is equal to 0.2. Those points are used as representative points. The clusters with the closest representative points are merged at each step. So, it becomes less sensitive to outliers. This is summarized in the following Algorithm 1. This algorithm is implemented for each class label \( l_i \in L \). The output of this algorithm is \( k \) cluster centroids, which are the class-based features of the corresponding class label \( l_i \).

3.2 Class-based feature vectors construction

After performing clustering on positive and negative instances sets of each class label, new class-based feature vectors for each data point need to be generated. For this, Euclidean distances between the data point and each centroid (class-based features) in the positive instances set and also in the negative instances set of a class label are
computed. These distance values will form a new feature vector. For example, for the first document, the new feature vector generated using the class-based features of the first class label ‘\( l_1 \)’ is shown in the following equation 1.

\[
FV_1(1) = a_{1(1)}, a_{2(1)}, \ldots, a_{m(1)}.
\]

(1)

Similarly, distance values between the data point and class-based features of second class label are computed. These values will generate another feature vector, which is shown in the following equation 2.

\[
FV_1(2) = b_{1(2)}, b_{2(2)}, \ldots, b_{n(2)}.
\]

(2)

Then, the second feature vector needs to be appended with the first feature vector generated for the first class label.

\[
FV_1(1, 2) = FV_1(1), FV_1(2).
\]

(3)

i.e.,

\[
FV_1(1, 2) = a_{1(1)}, a_{2(1)}, \ldots, a_{m(1)}, b_{1(2)}, b_{2(2)}, \ldots, b_{n(2)}.
\]

(4)

Similarly, new feature vectors will be generated for all the class labels and they will be appended one by one to form a new class-based feature vector for a data point, exclusively for multi-label classification. The following equation 5 shows the final modified feature vector of a document (object), which uses class-based features of 103 class labels.

\[
FV_1(1, 2, \ldots, 103) = FV_1(1), FV_1(2), \ldots, FV_1(103).
\]

(5)

This is summarized in the following Algorithm 2.

3.3 Local label correlation estimation

Local label correlations are considered to improve the performance. Local label correlation estimation has been done by performing label space partitioning, and by computing prototypes for each partition. First, all the possible label subsets are identified from the training data. Then, the documents belonging to the same label subset are grouped and centroids of each group are computed. This is summarized in the Algorithm 3.

3.4 Map function and classification

The aim of the map function is to make documents to collide in the correct region. For this, we use centroids (prototypes) computed in the previous module. First, class-based feature vector of the incoming document is generated. Then similarity between the incoming document and each prototype is measured using Euclidean distance computation.
Distance \[= \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \] (6)

Here \( n \) is the number of dimensions, \( p \) is incoming document, \( q \) is a prototype. From this, the nearest prototype is selected. This prototype will give the index of the region, where the document collides. The documents assigned to the same region have the same nearest prototype (centroid) and they are similar. So, the centroid-based multi-label classifier outputs multiple labels based on the nearest prototype. It is done by extracting the labels assigned to the nearest prototype.

4. Experimental Results

4.1 Data sets

The reuters dataset\(^5\) is a benchmark dataset for text classification methods. In this paper, an existing subset of this dataset that contains 1500 news articles with 500 features, assigned into one or more out of 103 class labels has been used.

4.2 Evaluation metrics

In order to evaluate the performance of the proposed algorithm, a thorough experimental and performance study was conducted\(^4\) using reuters dataset. A tradeoff between precision and recall is unavoidable. For this reason, these scores are usually combined into a single performance measure, called the \( F_1 \) measure. In multi label classification, the goal is to obtain a good performance among all the possible class labels, including those with fewer samples. Two different methods are typically used to assess such multi-label performance: macro – and micro – averaging. In macro averaging, performance is measured by averaging the values of \( F_1 \) among the different labels.

\[
F_1^{\text{macro}} = n_i^{-1} \sum_j F_1^j, 
\] (7)

where \( F_1^j = 2p_jr_j/(p_j + r_j) \), \( n_i \) is number of class labels, Precision \( p_j = n_{j++}/(n_{j++} + n_{j+-}) \), \( n_{j++} \) is false positive, \( n_{j+-} \) is true positive, \( n_{j-+} \) is false negative, recall \( r_j = n_{j+++}/(n_{j+} + n_{j-++}) \). In micro averaging, the different types of errors are first computed as a whole, and then they are processed to compute micro \( F_1 \) measure.

\[
F_1^{\text{micro}} = 2n_{++}/(2n_{++} + n_{+-} + n_{-+}), 
\] (8)

where \( n_{+-} = \sum_j n_{j+-}, n_{++} = \sum_j n_{j++}, n_{-+} = \sum_j n_{j-+} \).

The cumulative performance achieved by the model is monitored by computing precision, recall, and \( F_1 \) measures at any desired time. To evaluate the multi-label classifiers, micro and macro averages are computed among all the 103 tags.
Table 1. Experimental results of different multi-label learning algorithms on the reuters dataset.

<table>
<thead>
<tr>
<th>Evaluation criterion</th>
<th>BR</th>
<th>ML-KNN</th>
<th>RAkEL</th>
<th>CCBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro Average</td>
<td>0.084</td>
<td>0.169</td>
<td>0.032</td>
<td>0.348</td>
</tr>
<tr>
<td>Micro Average</td>
<td>0.136</td>
<td>0.426</td>
<td>0.128</td>
<td>0.811</td>
</tr>
<tr>
<td>Hamming Loss</td>
<td>0.128</td>
<td>0.016</td>
<td>0.015</td>
<td>0.0052</td>
</tr>
</tbody>
</table>

Fig. 2. $F_1$ Macro average values of different algorithms.

of the Reuters corpus. Here ML-KNN and BR algorithms are compared with the proposed extended centroid-based multi-label classification using class-based features (CCBF).

The Hamming loss computes the percentage of relevant labels that are not predicted or irrelevant labels that are predicted.

$$\text{Hamming Loss}(x_i, y_i) = \frac{1}{|D|} \sum_{i=1}^{|D|} x \lor (x_i, y_i)$$

where $|L|$ is the number of labels, $|D|$ is the number of samples, $y_i$ is the ground truth, $x_i$ is the prediction.

4.3 Comparative studies

In this section, CCBF is compared against two simple problem transformation methods BR, RAkEL and the high performing algorithm adaptation method ML-KNN. The comparison of the multi-label learning methods was performed using the implementations in the MEKA environment.

- BR (Binary Relevance): It learns binary classifiers one for each different label. It gives third largest macro and micro $F$ measures values.
- ML-KNN (Multi-label $K$-Nearest Neighbor): It is an extension of the popular k-nearest neighbour algorithm. It gives second largest macro and micro $F$ measure values.
- RAkEL (RAndom $k$-labELsets): It is an ensemble method for multi-label classification. It gives the lowest macro and micro $F$ measures values.

Table 1 shows the experimental results of different multi-label learning algorithms on the reuters dataset. The Fig. 2 and 3 show that $F1$ macro and micro average values of the proposed CCBF algorithm are better than BR, ML-KNN and RAkEL.

Hamming loss is a loss function whose optimal value is zero. We have observed that the smaller the value of hamming loss, the better the performance obtains. Figure 4 shows that the proposed CCBF algorithm is 24 times efficient when compared to BR method, thrice as efficient when compared to ML-KNN method; twice when compared to RAkEL.
5. Conclusion

There have been many algorithms which determine the membership of each possible class label to an object based on an identical feature set. This algorithm constructs class-based features to generate class-based feature vector, so that it gives better performance than other multi-label classification algorithms. The major contributions of this work are class-based feature vectors generation and local label correlation exploitation. Exploiting local label correlation minimizes the hamming loss. Exploiting class-based features is effective compared to other feature manipulation mechanisms. Experiments on reuters multi-label dataset show that CCBF achieves highly competitive performance against other state-of-the-art multi-label learning algorithms.

References