Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

# Clustering Based Multi-Label Classification for Image Annotation and Retrieval

Gulisong Nasierding
School of Engineering
Deakin University
Burwood, VIC 3125, Australia
gn@deakin.edu.au

Grigorios Tsoumakas
Department of Informatics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
greg@csd.auth.gr

Abbas Z. Kouzani
School of Engineering
Deakin University
Geelong, VIC 3217, Australia
kouzani@deakin.edu.au

***Abstract*—This paper presents a novel multi-label classification framework for domains with large numbers of labels. Automatic image annotation is such a domain, as the available semantic concepts are typically hundreds. The proposed framework comprises an initial clustering phase that breaks the original training set into several disjoint clusters of data. It then trains a multi-label classifier from the data of each cluster. Given a new test instance, the framework first finds the nearest cluster and then applies the corresponding model. Empirical results using two clustering algorithms, four multi-label classification algorithms and three image annotation data sets suggest that the proposed approach can improve the performance and reduce the training time of standard multi-label classification algorithms, particularly in the case of large number of labels.**

***Index Terms*—Clustering, multi-label classification, automatic image annotation.**

## I. Introduction

Image annotation is concerned with the task of assigning one or more semantic concepts to a given image. It is an important task that enables retrieving images from digital libraries based on the semantic concepts or keyword annotations of images. Manual image annotation is labor intensive. Therefore, there is a need for automatic image annotation (AIA).

Image data are typically multi-labeled, meaning that each image is annotated with more than a single concept. Most approaches treat such a learning task as multiple binary classification tasks. This way potential correlations among the labels are not taken into account. Recently however, several AIA approaches that attempt to take correlations into account have been developed [1], [2].

One important problem of methods that take correlations into account is the scalability with respect to a large-scale concept space [2]. A large number of labels may increase the computational cost of training and testing a multi-label classifier, and at the same time harm the predictive performance due to the skewed label distributions [3].

The above problem has stimulated us to propose a clustering based multi-label classification (CBMLC) framework. The proposed framework comprises an initial clustering phase that breaks the original training set into several disjoint clusters of data. It then trains a multi-label classifier from the data of each cluster. Given a new test instance, the framework first finds the nearest cluster and then applies the corresponding model. Empirical results using 2 clustering algorithms, 4 multi-label algorithms and 3 image annotation data sets suggest that the proposed approach can improve the performance and reduce the training time of standard multi-label algorithms, particularly in the case of large number of labels.

The rest of this paper is structured as follows. Section 2 presents background information on multi-label classification, including descriptions of algorithms used in the experimental setting. Section 3 reviews related work on automated image annotation, while Section 4 introduces the proposed approach. Section 5 describes the setting of the empirical study and Section 6 presents and discusses the results. Finally, section 7 concludes this work and points to future research directions.

## II. Multi-label Classification

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label $\lambda$ from a set of disjoint labels $L$, $|L| > 1$. In *multi-label* classification, the examples are associated with a set of labels $Y \subseteq L$. Multi-label classification algorithms can be categorized into 2 different groups [4]: i) *problem transformation* methods, and ii) *algorithm adaptation* methods. The first group includes methods that are algorithm independent. They transform the multi-label classification task into one or more single-label classification, regression or ranking tasks. The second group includes methods that extend specific learning algorithms in order to handle multi-label data directly.

We next present the methods used in the experimental part of this work. For their formal description, we use $L = \{\lambda_j : j = 1 \dots M\}$ to denote the finite set of labels in a multi-label learning task and $D = \{(\vec{x_i}, Y_i), i = 1 \dots N\}$ to denote a set of multi-label training examples, where $\vec{x_i}$ is the feature vector and $Y_i \subseteq L$ the set of labels of the $i$-th example.

Binary relevance (BR) is a popular problem transformation method that learns $M$ binary classifiers, one for each different label in $L$. For the classification of a new instance, BR outputs the union of the labels $\lambda_j$ that are positively predicted by the $M$ classifiers.

Label powerset (LP) is a simple problem transformation method, which 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. Given a new instance, the single-label classifier of LP outputs the most probable class, which

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

is actually a set of labels. The random $k$-labelsets (RA$k$EL) method [5] constructs an ensemble of LP classifiers. Each LP classifier is trained using a different small random subset of the set of labels. A ranking of the labels is produced by averaging the zero-one predictions of each model per considered label. Thresholding is then used to produce a classification as well.

BP-MLL [6] is an adaptation of the popular back-propagation algorithm for multi-label learning. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account. ML-$k$NN [7] extends the popular $k$ Nearest Neighbors ($k$NN) lazy learning algorithm using a Bayesian approach. It uses the maximum a posteriori principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the $k$ nearest neighbors.

## III. Related Work

Nowadays, a variety of automatic image annotation (AIA) approaches that use supervised machine learning algorithms are becoming increasingly popular [8], [9]. We particularly focus on several among these approaches that are based on classification [10], [11], [12], [1], [2]. The approaches of Coherent Language Model (CLM) with active learning [11] and Correlated Label Propagation (CLP) [1] consider the correlation of words among the semantic keyword annotations of images. A hierarchical boosting by incorporating concept ontology and multi-task learning model is addressed [10]; Particularly, to our interests, [12], [1], [13], [2] handled image annotation problem with effective multi-label classification methods such as multi-label boosting strategy which learning the correlation between image segments and the text tag in a set of training images [12]; SVM based multi-label active learning model is explored for image classification that eventually service to image annotation [13]. Kang et al. [1]pointed out that the critical problem remained in the existing approaches are how to exploit the correlations between class labels, i.e. multi-label learning problem. For this, a novel framework with correlated label propagation method is proposed [1]; more recently, a weighted k-nearest neighbor multi-label classification based image annotation algorithm [2] is explored which is inspiring, especially from the view of classifying multi-label images with large number of label set. Furthermore, multimedia data classifications for indexing and automated detection of the video concepts are explored as well [14], [15], [16].

## IV. Clustering Based Multi-Label Classification

The clustering-based multi-label classification (CBMLC) framework, which is proposed in this paper, comprises a clustering algorithm and a multi-label classification algorithm.

The training of CBMLC consists of 2 main steps: At the 1st step, CBMLC groups the training data into a user-specified number of clusters, $k$, using the clustering algorithm. The labels of the training data are not taken into consideration during this step. At the 2nd step, CBMLC uses the multi-label algorithm on the data of each cluster and produces $k$ multi-label classification models. For the classification of a new instance, CBMLC first finds the closest cluster of this instance, and then uses the corresponding multi-label model to classify it.

The clustering process splits the original training data into smaller parts of similar objects. We expect that similar objects are annotated with similar labels. We therefore hypothesize that each part will be annotated with much fewer labels compared to those used to annotate the complete training data set. This in turn is expected to improve the predictive performance of the corresponding multi-label models and reduce the training and classification time compared to using the same multi-label classification algorithm on the original data.

The instantiation of CBMLC to an actual algorithm requires the selection of specific clustering and multi-label classification algorithms. These could be selected based on the properties of the data at hand (e.g. dimensionality, discrete/numeric type of features, etc.) or application specific constraints (e.g. fast response time, fast training time, low space complexity etc.).

## V. Experimental Setup

### A. Data sets

We conducted experiments on three data sets containing multi-labeled multimedia objects. The first one, *eccv2002* [17], is a popular benchmark for image classification and annotation methods [8], [2], [1], [11]. It is based on 5000 Corel images, 4500 of which are used for training and the rest 500 for testing. The second one, *jmlr2003*, is produced from the first (001) subset of the data accompanying [18]. It is based on 6932 images, 5188 of which are used to create the training set and the rest 1744 to create the test set. The third one is based on the Mediamill Challenge data set [15]. It contains pre-computed low-level multimedia features from the 85 hours of international broadcast news video of the TRECVID 2005/2006 benchmark.

Table I shows the number of examples used for training and testing, the number of features and the number of labels for each data set. Note that the number of labels is typically large, ranging from 101 to 374.

| | examples | | | |
|---|---|---|---|---|
| name | train | test | features | labels |
| eccv2002 | 42379 | 4686 | 36 | 374 |
| jmlr2003 | 48859 | 16503 | 46 | 153 |
| mediamill | 30993 | 12914 | 120 | 101 |

TABLE I
Name, number of examples used for training and testing, number of features and number of labels for each data set used in the experiments

### B. Experimental Setting

We instantiate the clustering component of CBMLC using the $k$-means [19] and the expectation maximization [20] algorithms. We vary the number of clusters $k$ from 2 to 10.

We instantiate the multi-label classification component of CBMLC using the BR, RA$k$EL, ML$k$NN and BPMLL algorithms (see Section II). BR and RA$k$EL use the C4.5 decision tree learning algorithm for training the underlying single-label classifier. As recommended in [7], MLkNN is run with 10 nearest neighbors and a smoothing factor equal to 1. As recommended in [6], BPMLL is run with 0.05 learning rate, 100 epochs and the number of hidden units equal to 20% of the input units. We use the implementations of the multi-label classification algorithms from the open source Mulan library [5], which is built on top of the open source Weka library [21].

We evaluate all learning algorithms using the original train and test splits provided with the data sets. Given the large size of these data sets, such a single hold-out experiment should suffice to get accurate estimates of predictive performance. We calculate a variety of multi-label evaluation metrics that are available in the Mulan library [5], but present results using the micro $F_1$ measure only, which is a popular and good representative measure for multi-label classification.

We also keep a record of the training time of each algorithm, to give an empirical estimate of their computational complexity. All experiments were performed on a workstation equipped with a 2.5GHz processor and 8GB of memory.

## VI. Results and Discussion

In order to assess the benefit of the proposed framework, apart from providing absolute performance measurements, we also present graphs that plot the ratio of the performance of CBMLC over that of using directly the underlying multi-label classifier. For example, we compare the performance of CBMLC using BR compared to directly using BR.

### A. *Predictive Performance*

Table II shows the micro $F_1$ measure of the 4 multi-label classification algorithms in the 3 data sets. We notice that *eccv2002* and *jmlr2003* are quite difficult domains, while in *mediamill* the results are better. A different algorithm achieves the best results in each data set: RA$k$EL is the best in *eccv2002*, BPMLL in *jmlr2003* and ML$k$NN in *mediamill*. We will study the results of the CBMLC framework separately for each data set.

| data set | BR | RA$k$EL | ML$k$NN | BPMLL |
|---|---|---|---|---|
| eccv2002 | 0.1234 | 0.1349 | 0.0387 | 0.0186 |
| jmlr2003 | 0.1507 | 0.1507 | 0.0290 | 0.1661 |
| mediamill | 0.5055 | 0.5422 | 0.5451 | 0.5010 |

TABLE II
Micro $F_1$ of the 4 multi-label algorithms in the 3 data sets.

*1) Results on eccv2002:* Table III shows the micro $F_1$ measure for the different instantiations of CBMLC. As the gain of CBMLC for BPMLL is significantly larger than for the rest of the algorithms, we split the ratio graphs in two figures, one for BR, RA$k$EL and ML$k$NN (Figure 1), and one for BPMLL (Figure 2).

In terms of the clustering algorithms, we notice that using EM leads to better results than $k$-means for most number of clusters and all multi-label algorithms. In terms of the multi-label algorithms, we notice that BPMLL is significantly improved using both EM and $k$-means. This of course is in part due to the very low performance of BPMLL used by itself (0.0186). CBMLC improves BR and ML$k$NN from 5% to 20% for most number of clusters, when EM is used as the clusterer. The best overall result was achieved by CBMLC instantiated with EM ($k = 6$) and BR. When $k$-means is used there is no clear benefit for BR, while the performance of ML$k$NN degrades. Finally, RA$k$EL is not improved by CBMLC irrespectively of the clustering algorithm and number of clusters. This is explained from the fact that RA$k$EL directly deals with the problem of large number of labels, by utilizing labelsets of smaller size (in our case this size is only 3). Therefore it comes as no surprise that CBMLC does not improve it.

| clusterer | $k$ | BR | RA$k$EL | ML$k$NN | BPMLL |
|---|---|---|---|---|---|
| EM | 2 | 0.1248 | 0.1247 | 0.0417 | 0.1140 |
| $k$-means | 2 | 0.1248 | 0.1239 | 0.0427 | 0.1264 |
| EM | 3 | 0.1349 | 0.1250 | 0.0468 | 0.1039 |
| $k$-means | 3 | 0.1320 | 0.1286 | 0.0387 | 0.1109 |
| EM | 4 | 0.1310 | 0.1332 | 0.0425 | 0.1300 |
| $k$-means | 4 | 0.1189 | 0.1228 | 0.0334 | 0.1143 |
| EM | 5 | 0.1281 | 0.1359 | 0.0424 | 0.1287 |
| $k$-means | 5 | 0.1231 | 0.1224 | 0.0341 | 0.1020 |
| EM | 6 | 0.1372 | 0.1332 | 0.0434 | 0.1207 |
| $k$-means | 6 | 0.1196 | 0.1306 | 0.0340 | 0.1224 |
| EM | 7 | 0.1342 | 0.1252 | 0.0442 | 0.1323 |
| $k$-means | 7 | 0.1266 | 0.1291 | 0.0344 | 0.1232 |
| EM | 8 | 0.1356 | 0.1273 | 0.0414 | 0.1359 |
| $k$-means | 8 | 0.1263 | 0.1187 | 0.0342 | 0.1256 |
| EM | 9 | 0.1325 | 0.1289 | 0.0373 | 0.1348 |
| $k$-means | 9 | 0.1261 | 0.1250 | 0.0327 | 0.1294 |
| EM | 10 | 0.1336 | 0.1299 | 0.0420 | 0.1349 |
| $k$-means | 10 | 0.1198 | 0.1099 | 0.0306 | 0.1279 |

TABLE III
Micro $F_1$ for all instantiations of CBMLC in *eccv2002*.

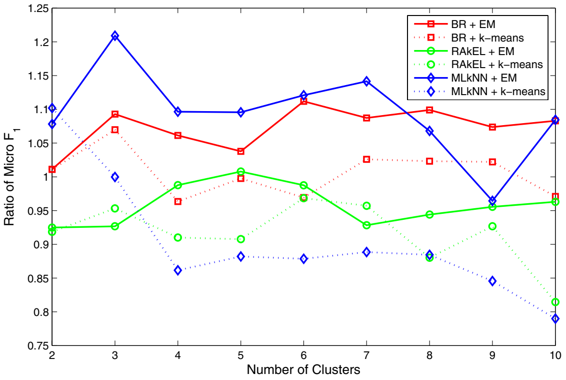

Fig. 1. Ratio of micro $F_1$ for BR, ML$k$NN and RA$k$EL using 2 to 10 clusters in *eccv2002*.

*2) Results on jmlr2003:* Table IV shows the micro $F_1$ measure for the different instantiations of CBMLC and Figure 3 depicts the ratio graph.

In terms of the clustering algorithms, we notice this time that using EM leads to better results than $k$-means for most number of clusters when using BPMLL and ML$k$NN only. For the other two multi-label algorithms there is no clear winning clustering algorithm, with $k$-means leading to some quite good results with RA$k$EL. In terms of the multi-label algorithms,

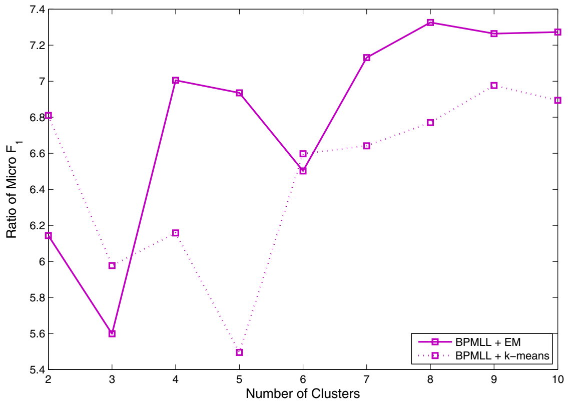

Fig. 2. Ratio of micro $F_1$ for BPMLL using 2 to 10 clusters in *eccv2002*.

| clusterer | $k$ | BR | RA$k$EL | ML$k$NN | BPMLL |
|---|---|---|---|---|---|
| EM | 2 | 0.1515 | 0.1513 | 0.0267 | 0.1863 |
| $k$-means | 2 | 0.1485 | 0.1560 | 0.0277 | 0.1797 |
| EM | 3 | 0.1541 | 0.1446 | 0.0296 | 0.1842 |
| $k$-means | 3 | 0.1571 | 0.1619 | 0.0263 | 0.1843 |
| EM | 4 | 0.1556 | 0.1559 | 0.0327 | 0.1842 |
| $k$-means | 4 | 0.1545 | 0.1665 | 0.0255 | 0.1808 |
| EM | 5 | 0.1523 | 0.1471 | 0.0303 | 0.1839 |
| $k$-means | 5 | 0.1510 | 0.1448 | 0.0274 | 0.1841 |
| EM | 6 | 0.1570 | 0.1489 | 0.0297 | 0.1928 |
| $k$-means | 6 | 0.1475 | 0.1511 | 0.0251 | 0.1807 |
| EM | 7 | 0.1549 | 0.1447 | 0.0319 | 0.1901 |
| $k$-means | 7 | 0.1571 | 0.1709 | 0.0257 | 0.1890 |
| EM | 8 | 0.1512 | 0.1504 | 0.0314 | 0.1942 |
| $k$-means | 8 | 0.1522 | 0.1457 | 0.0267 | 0.1862 |
| EM | 9 | 0.1550 | 0.1483 | 0.0299 | 0.1917 |
| $k$-means | 9 | 0.1491 | 0.1387 | 0.0235 | 0.1863 |
| EM | 10 | 0.1575 | 0.1531 | 0.0303 | 0.1945 |
| $k$-means | 10 | 0.1493 | 0.1387 | 0.0209 | 0.1839 |

TABLE IV
MICRO $F_1$ FOR ALL INSTANTIATIONS OF CBMLC IN *jmlr2003*.

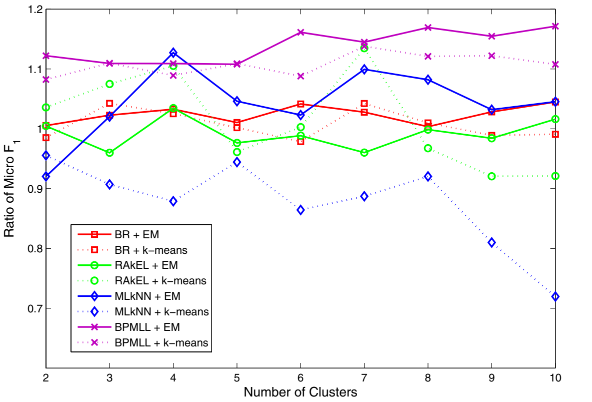

Fig. 3. Ratio of micro $F_1$ gain for all algorithms and number of clusters in *jmlr2003*.

again BPMLL is the algorithm improved most, but this time the improvement was less significant. This increase however is not negligible (more than 10%) and the fact that BPMLL was the best overall algorithm by itself makes this fact important. The best overall result is achieved by CBMLC instantiated with EM ($k = 10$) and BPMLL. In addition, all results for $k = 6$ to $k = 10$ of this instantiation of CBMLC are better than using BPMLL by itself.

ML$k$NN is improved by 2% to 12% from CBMLC using EM for $k > 2$. However, when $k$-means was used, the performance degraded, similarly to the previous data set. BR is slightly improved (up to 5%) from CBMLC using EM for all number of clusters, while when $k$-means is used, the performance is unstable: it decreases for certain values of $k$ and increases for others. The same instability is noticed for the instantiation of CBMLC using RA$k$EL and $k$-means. In this case the improvement increases up to 13% for $k$=7 but also drops by 8% for $k$=10. When EM is used instead the instability is reduced, similarly to BR, with the performance improving/dropping up to 4%.

*3) Results on Mediamill:* Similarly to the previous data set, Table V shows the micro $F_1$ measure for the different instantiations of CBMLC and Figure 4 depicts the ratio graph.

| clusterer | $k$ | BR | RA$k$EL | ML$k$NN | BPMLL |
|---|---|---|---|---|---|
| EM | 2 | 0.5060 | 0.5499 | 0.5423 | 0.5078 |
| $k$-means | 2 | 0.5068 | 0.5552 | 0.5396 | 0.5051 |
| EM | 3 | 0.5071 | 0.5545 | 0.5408 | 0.5114 |
| $k$-means | 3 | 0.5071 | 0.5624 | 0.5401 | 0.5136 |
| EM | 4 | 0.5010 | 0.5619 | 0.5356 | 0.5185 |
| $k$-means | 4 | 0.5081 | 0.5488 | 0.5410 | 0.5157 |
| EM | 5 | 0.5058 | 0.5490 | 0.5348 | 0.5229 |
| $k$-means | 5 | 0.5085 | 0.5484 | 0.5387 | 0.5205 |
| EM | 6 | 0.5014 | 0.5454 | 0.5370 | 0.5245 |
| $k$-means | 6 | 0.5001 | 0.5492 | 0.5369 | 0.5243 |
| EM | 7 | 0.5065 | 0.5521 | 0.5360 | 0.5269 |
| $k$-means | 7 | 0.5047 | 0.5585 | 0.5392 | 0.5209 |
| EM | 8 | 0.5016 | 0.5485 | 0.5386 | 0.5249 |
| $k$-means | 8 | 0.4992 | 0.5549 | 0.5389 | 0.5231 |
| EM | 9 | 0.5022 | 0.5575 | 0.5378 | 0.5254 |
| $k$-means | 9 | 0.5040 | 0.5544 | 0.5352 | 0.5251 |
| EM | 10 | 0.4988 | 0.5483 | 0.5373 | 0.5295 |
| $k$-means | 10 | 0.5015 | 0.5449 | 0.5308 | 0.5227 |

TABLE V
MICRO $F_1$ FOR ALL INSTANTIATIONS OF CBMLC IN *mediamill*.

In terms of the clustering algorithms, we notice this time that there is no clear winner between EM and $k$-means. In terms of the multi-label algorithms, once again BPMLL is the algorithm improved most, irrespectively of the clustering algorithm, and this time the improvement was even less significant (up to 6%). The performance of BR is unstable. Mostly it decreases but for some values of $k$ it increases. The amplitude of change is negligible (1%). This pattern is observed irrespectively of the clustering algorithm.

In contrast to the results obtained in the previous data sets, the performance of ML$k$NN is decreased irrespectively of the clustering algorithm for all values of $k$. CBMLC appears to be harmful to ML$k$NN in this data set. RA$k$EL is improved irrespectively of the clustering algorithm for all numbers of clusters. The improvement is small (up to 4%), but important, as it allows RA$k$EL to surpass the performance of ML$k$NN, which was the best performing algorithm in itself. The best overall result is achieved by CBMLC instantiated with $k$-means ($k = 3$) and RA$k$EL.

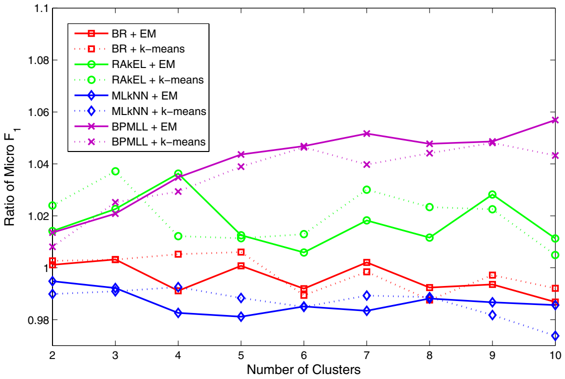

Fig. 4. Ratio of micro $F_1$ gain for all algorithms and number of clusters in *mediamill*.

*4) General Conclusions:* A general pattern observed across all data sets, was that CBMLC always improves BPMLL irrespectively of the clustering algorithm and the number of clusters. The improvement is proportional to the number of labels in the data set and approximately proportional to the number of clusters. For the rest of the algorithms there is no clear pattern. Instantiating CBMLC with BR or ML$k$NN and EM seems to produce good results for most number of clusters, especially when the number of labels is large. The opposite holds for RA$k$EL, which seems to be performing well with $k$-means, especially when the number of labels is small. Finally, EM seems to lead CBMLC to better results compared to $k$ means, especially for data sets with large number of labels.

### B. Training Time

Table VI shows the minutes of training time of the 4 algorithms in the 3 data sets. Being a lazy algorithm, ML$k$NN exhibits the lowest training time, which is mainly consumed in the calculation of prior probabilities. BPMLL is quite efficient as well, as it runs for 100 weight-updating epochs. BR trains a single decision tree for each label. As the number of trees is quite large, BR takes more time compared to training the single neural network of BPMLL. Finally, RA$k$EL is the most time-consuming algorithm, as it is an ensemble algorithm that builds as many models as twice the number of labels. The decision trees trained by RA$k$EL are not binary, but may deal with up to 8 classes (all combinations of $k = 3$ binary labels).

| data set | BR | RA$k$EL | ML$k$NN | BPMLL |
|---|---|---|---|---|
| eccv2002 | 43 | 192 | 8 | 30 |
| jmlr2003 | 47 | 182 | 9 | 13 |
| mediamill | 37 | 162 | 6 | 8 |

TABLE VI
MINUTES OF TRAINING TIME OF THE FOUR MULTI-LABEL CLASSIFICATION ALGORITHMS IN THE THREE DATA SETS.

Figures 5, 6 and 7 show the ratio of the training time of CBMLC over that of the underlying multi-label classification algorithm by itself for the *eccv2002*, *jmlr2003* and *mediamill* data sets respectively. As results exhibit approximately the same pattern in all data sets, we discuss them together.

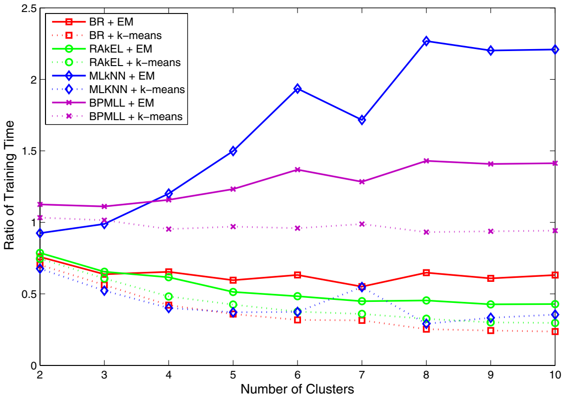

Fig. 5. Ratio of training time in *eccv2002* for all algorithms using 2 to 10 clusters.

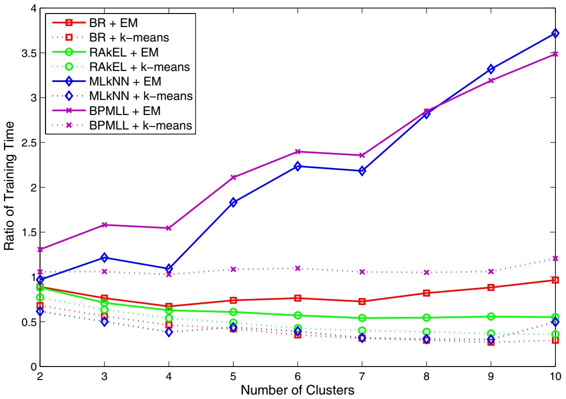

Fig. 6. Ratio of training time in *jmlr2003* for all algorithms using 2 to 10 clusters.

In terms of the clustering algorithm, we notice that using EM takes more time than using $k$-means. The single exception to this rule appears in the results of *mediamill*, where the instantiation of CBMLC with RA$k$EL takes more time with $k$-means than with EM for $k = 2$ (indiscernible in the graph). These results are in line with the fact that $k$-means is computationally less demanding than EM. However, this often comes at the expense of the predictive performance of CBMLC as we saw in the previous subsection.

In terms of the algorithms, we first notice that the training time of BPMLL and ML$k$NN is increased when instantiating CBMLC with EM. As these algorithms are quite fast already, the cost of clustering via EM dominates the total time, leading to the observed increase. The increase is amplified as the number of training data and labels in the data set decreases (e.g. *mediamill*).

When using $k$-means instead, we notice that there is no great overhead or reduction for BPMLL. This can be explained

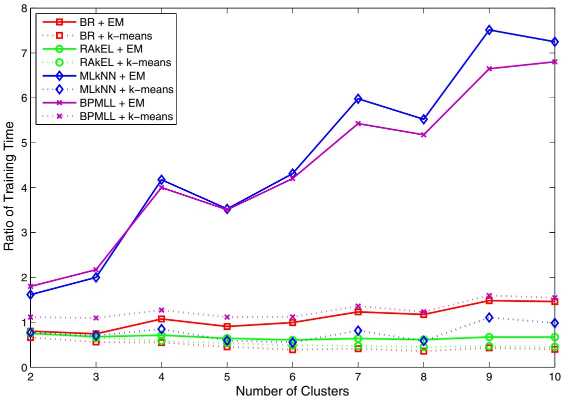

Fig. 7. Ratio of training time in *mediamill* for all algorithms using 2 to 10 clusters.

from the fact that reducing the number of labels through clustering, actually reduces only a small part of the weight-updating process that drives the main complexity of BPMLL. For ML$k$NN on the other hand we notice a reduction of up to 50 % (especially in the larger data sets) with respect to the number of clusters. It seems that the fewer labels in each cluster reduces the number of prior probabilities that need to be computed by ML$k$NN, improving its efficiency.

The training time of RA$k$EL is reduced via CBMLC for all numbers of clusters in all data sets. The same applies for BR, with the exception of *mediamill*, where the instantiation of CBMLC with EM leads to an increase of the training time for certain numbers of clusters. The larger the number of labels in a data set, the larger the probability that clustering will lead to clusters with reduced number of labels, and consequently the larger the gains in training time for BR.

## VII. Conclusions and Future Work

This paper presented a new multi-label classification framework, called CBMLC, that involves an initial clustering phase. The motivation for this framework was the large number of labels that is present in several multi-label classification domains, such as that of automatic image annotation. The framework is generic and can utilize existing clustering algorithms and multi-label classification algorithms. Experimental results with several instantiations of the framework showed that it can improve the predictive performance and reduce the training time compared to standard multi-label algorithms.

One issue worthy of further investigation is that of clustering. Experimental results showed that the clustering algorithm plays a crucial role in the overall performance of the framework. We therefore intent to experiment using additional high-performance clustering algorithms. Another issue worth examining is the scaling of the proposed approach to domains with thousands of labels. We expect that CBMLC will be highly valuable in such domains. Finally, we aim at applying CBMLC to image and image region annotations [8] and deliver a complete service for concept based image retrieval.

## References

[1] F. Kang, R. Jin, and R. Sukthankar, “Correlated label propagation with application to multi-label learning,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1719–1726.

[2] M. Wang, X. Zhou, and T. S. Chua, “Automatic image annotation via local multi-label classification,” in *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*. ACM, 2008, pp. 17–26.

[3] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Effective and efficient multilabel classification in domains with large number of labels,” in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008.

[4] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[5] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification,” in *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, Warsaw, Poland, September 17-21 2007, pp. 406–417.

[6] M.-L. Zhang and Z.-H. Zhou, “Multi-label neural networks with applications to functional genomics and text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[7] ——, “Ml-knn: A lazy learning approach to multi-label learning,” *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[8] C.-F. Tsai and C. Hung, “Automatically annotating images with keywords: A review of image annotation systems,” *Recent Patents on Computer Science*, vol. 1, no. 1, pp. 55–68, 2008.

[9] G. Nasierding and A. Z. Kouzani, “Image to text translation: A review,” in *Proceedings of the 4th International Conference on Humanized Systems*, Beijing, October 2008, pp. 378–383.

[10] J. Fan, Y. Gao, and H. Luo, “Hierarchical classification for automatic image annotation,” in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 2007, pp. 111–118.

[11] R. Jin, J. Y. Chai, and L. Si, “Effective automatic image annotation via a coherent language model and active learning,” in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. ACM Press, 2004, pp. 892–899.

[12] M. Johnson and R. Cipolla, “Improved image annotation and labelling through multi-label boosting,” in *Proceedings of 2005 British Machine Vision Conference (BMVC '05)*, 2005.

[13] X. Li, L. Wang, and E. Sung, “Multi-label svm active learning for image classification,” in *IEEE 2004 International Conference on Image Processing (ICIP '04)*, 2004, pp. 2207–2210.

[14] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, “Correlative multi-label video annotation,” in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 17–26.

[15] C. G. Snoek, M. Worring, J. C. van Gemert, J. Geusebroek, and A. W. Smeulders, “The challenge problem for automated detection of 101 semantic concepts in multimedia,” in *Proceedings of ACM Multimedia*, Santa Barbara, USA, October 2006, pp. 421–430.

[16] A. Dimou, G. Tsoumakas, V. Mezaris, and I. Kompatsiaris, I.and Vlahavas, “An empirical study of multi-label learning methods for video annotation,” in *7th International Workshop on Content-Based Multimedia Indexing, IEEE*, 2009.

[17] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in *7th European Conference on Computer Vision*, 2002, pp. (IV)97–112.

[18] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan, “Matching words and pictures,” *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[19] A. Jain and R. Dubes, *Algorithms for clustering data*. Englewood Cliffs: Prentice-Hall, 1988.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[21] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.