2005

Collective Multi-Label Classification

Nadia Ghamrawi

University of Massachusetts - Amherst

Andrew McCallum

University of Massachusetts - Amherst

Follow this and additional works at: http://scholarworks.umass.edu/cs_faculty_pubs

Part of the Computer Sciences Commons


http://scholarworks.umass.edu/cs_faculty_pubs/190

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Computer Science Department Faculty Publication Series by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.
Collective Multi-Label Classification

Nadia Ghamrawi
University of Massachusetts Amherst
Amherst, Massachusetts, USA
ghamrawi@cs.umass.edu

Andrew McCallum
University of Massachusetts Amherst
Amherst, Massachusetts, USA
mccallum@cs.umass.edu

ABSTRACT
Common approaches to multi-label classification learn independent classifiers for each category, and employ ranking or thresholding schemes for classification. Because they do not exploit dependencies between labels, such techniques are only well-suited to problems in which categories are independent. However, in many domains labels are highly interdependent. This paper explores multi-label conditional random field (CRF) classification models that directly parameterize label co-occurrences in multi-label classification. Experiments show that the models outperform their single-label counterparts on standard text corpora. Even when multi-labels are sparse, the models improve subset classification error by as much as 40%.

Categories and Subject Descriptors
I.5.1 [Pattern Recognition]: Models—statistical, structural

General Terms
Design, Experimentation, Performance

Keywords
Classification, machine learning, multi-label, statistical learning, uncertainty

1. INTRODUCTION
Single-label classification assigns an object to exactly one class, when there are two or more classes. Multi-label classification is the task of assigning an object simultaneously to one or multiple classes.

The most common approach independently learns a binary classifier for each class, and then assigns to a test instance all of the class labels for which the corresponding classifier says “yes.” Experiments have shown that the classifiers such as Widrow-Hoff, k-nearest-neighbor, neural networks and linear least squares fit mapping are viable techniques for this approach [17], as are support vector machines [8]. Although some binary classifiers provide posterior probability over their binary answers, they need only have binary valued output.

Another approach requires a real-valued score for each class, suitable for ranking class labels, and then classifies an object into the classes that rank above a threshold. Schapire Singer Schapire99 [14] develop a boosting algorithm that gives rise to such a ranking. The model described by Crammer Singer Crammer02 [5] learns a prototype feature vector for each class, and a class rank is derived from the angle between its prototype and the document. The model in Gao et al. Gao04 [6] trains independent classifiers for each category that may share some parameters, and ranks each classification according to a confidence measure.

The above methods learn independent classifiers for each class. However, it is often the case that there are strong co-occurrence patterns and dependencies among the class labels. Explicitly leveraging these patterns may be advantageous. For example, the belief that a research article having the word sodium is likely to be labeled HEART DISEASE supports the belief that the document should also be given the label HYPERTENSION. A method that captures dependencies between class labels is likely to provide improved classification performance, particularly for more richly multi-labeled corpora than those used in experiments.

This paper presents two multi-label graphical models for classification that parameterize label co-occurrences. As in traditional classifiers, both models learn parameters associated with feature-label pairs. The Collective Multi-Label classifier (CML) also, jointly, learns parameters for each pair of labels. The Collective Multi-Label with Features classifier (CMLF) learns parameters for feature-label-triples—capturing the impact that an individual feature has on the co-occurrence probability of a pair of labels.

We present experiments using two data sets that, although sparsely multi-labeled, have become standard for multi-label classification experiments: the Reuters-21578 and OHSU-Med text corpora. CML and CMLF outperformed the binary models: they reduced error in subset accuracy by as much as 27%, reduced error in macro- and micro- averages by up to 9%, and had consistently better performance than their binary counterparts.

2. THREE MODELS FOR MULTI-LABEL CLASSIFICATION
Conditional probability models for classification offer a rich framework for parameterizing relationships between class labels and features, or characteristics, of objects. Furthermore, such models often outperform their generative counterparts.
2.2 Accounting for Multiple Labels

The single-label model above learns a distribution over labels. In a multi-label task, the model should learn a distribution over subsets of the set of labels \( \mathcal{Y} \), which are represented as bit vectors \( y \) of length \( |\mathcal{Y}| \).

In the most general form, given instance \( x \) and features \( f_k \),

\[
p(y|x) = \frac{1}{Z(x)} \exp \left( \sum_k \lambda_k f_k(x, y) \right),
\]

where \( Z(x) \) is the normalizing constant. All three CRF models capture the following enumeration over features in the learned distribution:

\[
k \in \{ \langle v_i, y_j \rangle : 1 \leq i \leq |V|, 1 \leq j \leq |\mathcal{Y}| \}.
\]

That is, all three models capture the dependency between each object feature and each label.

2.2.1 Binary Model

A common way to perform multi-label classification is with a binary classifier for each class. For each label \( y_b \), the binary model trains an independent binary classifier \( c_b \), partitioning training instances into positive (+) and negative (–) classes (Figure ??). The learned distribution \( p_b \) is as in Equation ??, except that

\[
k \in \{ \langle v_i, r_j \rangle : 1 \leq i \leq |V|, 0 \leq j \leq 1 \}
\]

since \( r_j \in \{+,-\} \). However, the distribution over multi-labelings, \( p(y|x) \) is as follows:

\[
p(y|x) = \prod_k p_b(y_b|x).
\]

This scheme attributes an object \( x \) to category labeled \( y_b \) if \( c_b \) classifies \( x \) positively. However, the classifications are treated independently.

Figure ?? depicts this model as a factor graph. The black squares (factors) represent the model parameters. For example, in Figure 1(a), the binary model maintains a parameter for each pair consisting of a label and a feature. Factor graphs are graphical models that depict the clique parameterizations. Inference in factor graphs is done in a way similar to inference in graphical models [10].

2.2.2 CML Model

In order to capture co-occurrence patterns among labels, this paper presents a conditional random field representing dependencies among the output variables.

In addition to having feature for each label-term pair, CML maintains features accounting for label co-occurrences. This model is depicted in Figure ??.

For object \( e \) and labels \( y \) and \( y' \), there are four features:

<table>
<thead>
<tr>
<th>feature</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y )</td>
<td>neither ( y' ) nor ( y ) labels ( e )</td>
<td>( y' ) but not ( y ) labels ( e )</td>
<td>( y ) but not ( y' ) labels ( e )</td>
<td>both ( y ) and ( y' ) label ( e )</td>
</tr>
</tbody>
</table>

For \( k' = \langle \text{WHEAT}, \text{GRAIN}, 2 \rangle \) and training document \( (x, y), f_k'(x, y) \) is 1 if \( x, y \) is labeled \( \text{GRAIN} \) but not \( \text{WHEAT} \), and 0 otherwise. A document has \( \binom{|\mathcal{Y}|}{2} \) such features.
The log likelihood $l$ each pair of labels and each feature. Parameterization involving pairs of labels, and figure (c) represents a parameterization for each label and each feature, together with clique parameterizations. In (a) each parameterization involves one label and one feature. Figure (b) represents an additional parameterization involving pairs of labels, and figure (c) represents a parameterization for each label and each feature, together with each pair of labels and each feature.

Figure 1: Factor graphs representing the multi-label models, where $y_i$ is a label and $x_i$ is a feature, and the black squares represent clique parameterizations. In (a) each parameterization involves one label and one feature. Figure (b) represents an additional parameterization involving pairs of labels, and figure (c) represents a parameterization for each label and each feature, together with each pair of labels and each feature.

The distribution $p(y|x)$ thus becomes

$$
\frac{1}{Z_A(x)} \exp \left( \sum_k \lambda_k f_k(x, y) + \sum_{k'} \lambda_{k'} f_{k'}(y) \right)
$$

where $Z_A(x)$ is the normalizing constant and

$$
\begin{align*}
&k \in \{(v_i, y_j) : 1 \leq i \leq |V|, 1 \leq j \leq |Y|\}, \\
&k' \in \{(y_i, y_j, q) : q \in \{0, 1, 2, 3\}, 1 \leq i, j \leq |Y|\}.
\end{align*}
$$

The log likelihood $l(\lambda|D)$ is similar to Equation ??:

$$
\sum_{d=1}^r \left( \sum_k \lambda_k f_k(x_d, y_d) + \sum_{k'} \lambda_{k'} f_{k'}(y_d) - \log Z_A(x_d) \right)
- \sum_k \frac{\lambda_k^2}{2\sigma^2} - \sum_{k'} \frac{\lambda_{k'}^2}{2\sigma^2}.
$$

(8)

The computation of the gradient is analogous to Equation ??: CML captures the label co-occurrences in the corpus independent of the object's feature values. Effectively, for each label set, it adds a bias that varies proportionally to the label set frequency in training data. The factor graph for this model is depicted in Figure ??.

### 2.2.3 CMLF Model

While CML parameterizes the dependencies between labels in general, these dependencies do not account for the presence of particular observational features (e.g., words). The tendency of labels to occur together in a multi-labeling is not independent of the appearance of the observational features. For instance, a text document belonging to the categories RICE and SOYBEAN might have increased likelihood of being correctly classified if the document has the word *cooking*, but decreased likelihood of belonging to ALTERNATIVE FUELS. The factor graph in Figure ?? reflects this dependency. The CMLF model maintains parameters that correspond to features for each $(term, label_1, label_2)$ triplet, capturing parameter values for (cooking, RICE, SOYBEAN), for example.

As with CML, CMLF defines feature parameters over the labels and words,

$$k \in \{(v_i, y_j) : 1 \leq i \leq |V|, 1 \leq j \leq |Y|\},$$

but also defines parameters over pairs of labels and words,

$$k' \in \{(v_i, y_j, y_{j'}) : 1 \leq i \leq |V|, 1 \leq j, j' \leq |Y|\},$$

for a total of $O(n^2|V|)$ parameters for $n$ labels. Note that CMLF maintains overlap in term occurrences: it has a feature for each pair consisting of a term and a label, as well as a feature for each triplet consisting of a term and two labels. The features enumerated by $k$ provide some shrinkage, and thus protection from overfitting [4].

The corresponding distribution that CMLF learns is

$$
\frac{1}{Z_A(x)} \exp \left( \sum_k \lambda_k f_k(x, y) + \sum_{k'} \lambda_{k'} f_{k'}(x, y) \right).
$$

(9)

The gradients of the log likelihood at $k$ and at $k'$ are the same as those of CML, except that $k'$ enumerates different features. CML has four features for each pair of labels, while CMLF has $|V|$ features. The factor graph for this model is depicted in Figure ??; note that for each observational feature, there is a parameter for each label, and also a parameter for each pair of labels.

Parameter estimation in these models is the same as for the single-label model: calculation of the value and gradient is straightforward, and BFGS is used to find the optimal parameters given the gradient of the log-likelihood. Note that neither multi-label model assumes that the label taxonomy has a complex structure, although extra parameters accounting for this could easily be added.

Table ?? shows the asymptotic complexity of training an instance. The binary technique is faster than the multi-label models in most cases, but performance of binary pruning depends on selection of the threshold, which determines the number of classes. In large datasets with many rarely occurring multi-labelings, binary pruning requires considerably less training time than supported inference, for comparable classification performance. Experiments suggest that the binary pruned inference technique is faster than supported inference. CMLF is linear with respect to CML, which is asymptotically simpler than the binary classifier method only if the multi-labelings are sparse. However, in practice binary classifiers are faster to train because they use fewer parameters in optimization.
Apart from providing a probability estimate for each label, exact inference using the collective models requires learning a probability distribution over all possible multi-labelings — that is, over all subsets of $\mathcal{Y}$. This method is intuitively appealing: it is easy to explain, and it is informative, since it offers a probability score for each combination of labels, regardless of the combination presence in the training data. However, since the number of subsets is exponential in the number of class labels, the problem is tractable only for about 3-12 classes. When the number of classes is larger, approximate inference methods may prune certain combinations of labels, and calculate the conditional distribution over the pruned set.

One method of pruning is to include only the label combinations that occur in training data—which we term the supported combinations. This method can sometimes be surprisingly effective. For the top 10 classes in Reuters-21578, only 0.6% of test instances belong to combinations of categories that do not occur in training data. For the entire ModApte split, the error due to supported inference is more significant: 4% of test instances have label combinations that do not occur in training data. When there are few classes and few such outliers, or when such rare combinations can be excluded, then supported inference is a very good solution.

An alternative approximate inference method is termed binary pruned inference, and represents a compromise between supported and exact inference. The model trains an independent binary classifier for each label. Then when classifying an object, exact inference considers only the labels having binary classifier probability scores above a certain threshold ($t$). Cross validation on training data is used to choose the threshold.

Binary pruned inference makes it possible to correctly classify test documents whose actual combinations do not occur in the training data. Furthermore, the method requires less training time than supported inference.

### 4. EXPERIMENTS

We present experiments with these multi-label classifiers on two standard multi-label data sets: Reuters-21578 and the ‘Heart Disease’ (HD) documents of OHSU-Med. The corpora differ in the noise level and length of documents. Both have simple label taxonomies: labels are not hierarchical, and each document has at least one label from the entire label set.

Except in the case of the $k'$ features of CML, features $f_i$ are represented by count of occurrences, in experiments presented here. Alternate representations include frequency of occurrences, for example.

![Image](image_url)

#### Table 1: Asymptotic per-instance training complexity, given $|V| = v$, $k$ labels, $s$ total label combinations of average size $a$ and $\tau$ labels ranking above threshold on average.

<table>
<thead>
<tr>
<th></th>
<th>binary</th>
<th>CML</th>
<th>CMLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>supported</td>
<td>$k^2v$</td>
<td>$s(ava + k^2)$</td>
<td>$sa^2v$</td>
</tr>
<tr>
<td>pruned</td>
<td>$k^2v$</td>
<td>$2'(rv + k^2)$</td>
<td>$2'r^2v$</td>
</tr>
</tbody>
</table>

#### 4.1 Corpora

The ModApte split of Reuters-21578, in which all labeled documents that occur before April 8, 1987 are used in training and other labeled documents are used in testing, is a popular benchmark for experiments. The ModApte documents consist of those documents labeled by the 90 classes which have at least one training and one testing instance, accounting for 94% of the corpus. Roughly 8.7% of these documents have multiple topic labels.

Experiments using corpus Reuters10 use only documents belonging to the 10 largest classes, which label 84% of the documents and form 39 distinct combinations of labels in the training data. Table ?? depicts the distribution of multi-label cardinalities in the ReutersAll test set, together with the label classification error rate of the binary classifiers.

The OHSU-Med [7] HD corpus, a popular dataset for text classification, is a collection of titles and abstracts of medical research journal articles from 1989-1991 corresponding to characterizations of the relevant heart conditions, such as “Heart Aneurysm” and “Myocarditis”. The HD-small documents belong to the 40 categories which label between 15 and 74 training documents, forming 106 combinations of labels in the training data. HD-big consists of documents belonging to the remaining 16 categories that each label 75 or more training documents.

#### 4.2 Results

Features are ranked according to their mutual information, so that the classifiers may select a proportion of features having the highest rank. Parameters that influence performance of the classifiers include proportion of features selected, Gaussian prior variance of the parameters, and in the case of binary pruning, the threshold for the binary classifiers. The classifiers are least sensitive to the Gaussian prior, and binary pruning is most sensitive to the threshold. Lower thresholds have higher classification cost but higher thresholds limit the performance of CML and CMLF to the performance of the binary classifiers.

In experiments presented in this paper, words occurring fewer than 5 times in all training documents are excluded from the vocabulary, and all classifiers assume a Gaussian prior variance of 1.0. Thresholds and feature proportions are learned using cross validation on training data. That is, the parameters that a given classifier uses are those which yield the best average performance, of the binary model and its multi-label counterpart, using a random partition of the training data into training and validation instances.

The results are compared using three metrics: F1 micro-average, F1 macro-average [17], and subset accuracy. The macro-average is the mean of the F1-scores of all the labels, thus attributing equal weights to each F1-score. The micro-average is the F1-score obtained from the summation of contingency matrices for all binary classifiers. The micro-average metric gives equal weight to all classifications, so that F1 scores of larger classes influence the metric more than F1 scores of smaller classes. F1-score reflects the harmonic mean of precision and recall. Subset accuracy is the proportion of documents with entirely correct bit vectors $y$.

#### 4.2.1 Reuters-21578

Even for the sparsely multi-labeled ReutersAll, CMLF reduces error in F1 averages by as much as 5%, and reduces error in subset classification by 16%. Table ?? depicts the results of experiments on ReutersAll using the ModApte split, as well as a comparison of

---

Footnote:

1 The mis-classification rate is the percent of times that the binary classifier incorrectly assigns one of the labels to an object, or fails to assign the correct label to an object.
the two inference methods. Supported inference experiments are more costly in time and space than binary pruning.

With ReutersAll, binary pruning generally performs better than supported inference. Furthermore CML and CMLF perform better than the best reported results.

The binary pruning technique resulted in 3% higher F1 micro-average and 23% higher macro-average than supported inference. The significant gain in macro-average suggests that binary pruning improves performance of smaller classes.

Collective classifiers perform better than the traditional binary model, supporting our contention that the classes are not independent, and that directly parameterizing these dependencies is advantageous.

4.2.2 OHSU-Med

HD is a noisier corpus than Reuters-21578, having topics that span a narrower semantic scope. As with Reuters-21578, CML and CMLF trump the traditional binary models. With thresholds chosen using cross validation on training data, CML and CMLF achieve better performance with supported inference than binary pruning.

In HD, typically more than half of the misclassifications in binary pruning are due to the pruning of positive classes. Thus on pruned instances, the F1 averages that the collective models achieve with supported inference are higher than the averages achieved using binary pruning.

Table ?? depicts performance of the five techniques on HD-small and HD-big. Compared to the traditional binary model, using supported inference, the collective classifiers improve subset accuracy by 20-40%, whereas with ReutersAll, this improvement is about 4%. (The collective models increase F1 averages by 5-9% for both HD corpora.) It is gratifying to see that on tasks with larger, more complex multi-labeled sets, our method provides even greater improvement.

The average improvement of CML and CMLF over binary classifiers is even greater across several trials using random test-train splits (of comparable proportions to those of Table ?? experiments) of the corpus. Experiments suggest that more innovative binary pruning models could improve performance considerably.

5. RELATED WORK

Some existing models indirectly leverage the multi-label dependencies that traditional methods do not. semantic scene classification, Boutell et al. Boutell03 [2] train a single-label classifier for each label, using all single-label documents and only the multi-label documents with that label. This approach indirectly leverages label co-occurrences, but it does not directly parameterize multi-label dependencies.

Expectation Maximization has been used to train a mixture model [11] for which the features of each document are produced by a mixture of word distributions for each class. [16] take a similar
approach in that each word in each category is generated from a multinomial distribution over vocabulary words. Both of these approaches are generative, and both leverage information about multiple class memberships for a given document implicitly by learning which classes generate which features.

Relational Markov Network models (RMNs) [15], are undirected graphical models like CML and CMLF. However, they perform single-label classification simultaneously of multiple documents, whereas CML and CMLF address the issue of multi-label classification of a single document. Furthermore, RMNs use the hyperlinks linking separate documents to capture dependencies between documents, but the model relies on the inherent sparseness of those dependencies, while CML and CMLF prove advantageous for densely multi-labeled corpora. RMNs use loopy belief propagation is used for estimating the gradient.

6. CONCLUSIONS AND FUTURE WORK

Multi-label classification is an important task in domains beyond text. In many real-world tasks, classes are not independent. CML and CMLF offer a framework for leveraging the dependencies between categories by including factors that capture label co-occurrences, whereas previous methods leverage category dependencies only indirectly, at best.

The success of conventional classification approaches depends on properties such as independence of classes and sparsity of multi-labeling. On varying corpora, over several metrics, the collective models outperform these methods.

Research related to multi-label classification involves automatically annotating biomedical abstracts with lists of genes that are mentioned in the documents. This is related to multi-label classification because each gene may have several synonyms, and a synonym may refer to several genes. More generally, in any domain in which subsets of unstructured interdependent outcomes are to be assigned, the CML and CMLF framework suggests a viable solution.

Future experiments may test the models in different domains and use corpora with varying noise characteristics, as well as domains in which features do not have uniform weight and type, including semantic scene classification.

Improved inference and pruning methods may be more tractable than exact and supported inference and allow greater flexibility than binary pruning.

A more general extension of CML and CMLF would parameterize larger factors, rather than pairs of labels, and incorporate schemes for learning which factors to include [12]. Enhanced models could also handle unlabeled data.

7. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by Air Force Office of Scientific Research contract #FA9550-04-C-0053 through subcontract #0250-1181 from Aiptima, Inc., in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES


