Exact Inference for Multi-label Classification using Sparse Graphical Models

Yusuke Miyao[†] **Jun'ichi Tsujii**[†‡*]
[†]Department of Computer Science, University of Tokyo, Japan
[‡]School of Computer Science, University of Manchester, UK
[*]National Center for Text Mining, UK
{yusuke,tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper describes a parameter estimation method for multi-label classification that does not rely on approximate inference. It is known that multi-label classification involving label correlation features is intractable, because the graphical model for this problem is a complete graph. Our solution is to exploit the sparsity of features, and express a model structure for each object by using a sparse graph. We can thereby apply the junction tree algorithm, allowing for efficient exact inference on sparse graphs. Experiments on three data sets for text categorization demonstrated that our method increases the accuracy for text categorization with a reasonable cost.

1 Introduction

This paper describes an exact inference method for multi-label classification (Schapire and Singer, 2000; Ghamrawi and McCallum, 2005), into which label correlation features are incorporated. In general, directly solving this problem is computationally intractable, because the graphical model for this problem is a complete graph. Nevertheless, an important characteristic of this problem, in particular for text categorization, is that only a limited number of features are *active*; i.e., non-zero, for a given object x. This sparsity of features is a desirable characteristic, because we can remove the edges of the graphical model when no corresponding features are active. We can therefore expect that a graphical model for each object is a sparse graph. When a graph is sparse, we can apply the junction tree algorithm (Cowell et al., 1999), allowing for efficient exact inference on sparse graphs.

Our method is evaluated on three data sets for text categorization; one is from clinical texts, and the others are from newswire articles. We observe the trade-off between accuracy and training cost, while changing the number of label correlation features to be included.

2 Multi-label Classification

Given a set of labels, $L = \{l_1, \ldots, l_{|L|}\}$, multi-label classification is the task of assigning a subset $y \subseteq L$ to a document x. In the framework of statistical machine learning, this problem can be formulated as a problem of maximizing a scoring function η:

$$\hat{y} = \operatorname*{argmax}_y \eta(x, y) = \operatorname*{argmax}_y \eta(\mathbf{f}(x, y)). \quad (1)$$

As is usually the case in statistical machine learning, we represent a probabilistic event, $\langle x, y \rangle$, with a feature vector, $\mathbf{f}(x, y) = \langle f_1(x, y), \ldots, f_{|\mathbf{f}|}(x, y) \rangle$. In text categorization, most effective features represent a frequency of a word w in a document; i.e.,

$$f_{l,w}(x, y) = \begin{cases} c_x(w) & \text{if } l \in y, \\ 0 & \text{otherwise,} \end{cases}$$

where $c_x(w)$ is a frequency of w in x.

The most popular method for multi-label classification is to create $|L|$ binary classifiers, each of which determines whether or not to assign a single label (Yang and Pedersen, 1997). However, since the decision for each label is independent of the decision for other labels, this method cannot be sensitive to *label correlations*, or the tendency of label cooccurrences.

A recent research effort has been devoted to the modeling of label correlations. While a number of approaches have been proposed for dealing with label correlations (see Tsoumakas and

Coling 2008: Companion volume – Posters and Demonstrations, pages 63–66
Manchester, August 2008

Katakis (2007) for the comprehensive survey), the intuitively-appealing method is to incorporate features on two labels into the model (Ghamrawi and McCallum, 2005). The following *label correlation feature* indicates a cooccurrence of two labels and a word:

$$f_{l,l',w}(x,y) = \begin{cases} c_x(w) & \text{if } l, l' \in y, \\ 0 & \text{otherwise.} \end{cases}$$

3 A Method for Exact Inference

A critical difficulty encountered in the model with label correlation features is the computational cost for training and decoding. When features on every pair of labels are included in the model, its graphical model becomes a complete graph, which indicates that the exact inference for this model is NP-hard. However, not all edges are necessary in actual inference, because of the sparsity of features. That is, we can remove edges between l and l' when no corresponding features are active; i.e., $f_{l,l',w}(x,y) = 0$ for all w. In text categorization, when feature selection is performed, many edges can be removed because of this characteristic.

Therefore, our idea is to enjoy this sparsity of features. We construct a graphical model for each document, and put edges only when one or more features are active on the corresponding label pair. When a graph is sparse, we can apply a method for exact inference, such as the junction tree algorithm (Cowell et al., 1999). The junction tree algorithm is a generic algorithm for exact inference on any graphical model, and it allows for efficient inference on sparse graphs. The method converts a graph into a *junction tree*, which is a tree of cliques in the original graph. When we have a junction tree for each document, we can efficiently perform belief propagation in order to compute argmax in Equation (1), or the marginal probabilities of cliques and labels, necessary for the parameter estimation of machine learning classifiers, including perceptrons (Collins, 2002), and maximum entropy models (Berger et al., 1996). The computational complexity of the inference on junction trees is proportional to the exponential of *the tree width*, which is the maximum number of labels in a clique, minus one.

An essential idea of this method is that a graphical model is constructed for each document. Even when features are defined on all pairs of labels, active features for a specific document are limited. When combined with feature selection, this method greatly increases the sparsity of the resulting graphs, which is key to efficiency.

A weakness of this method comes from the assumption of feature sparseness. We are forced to apply feature selection, which is considered effective in text categorization, but not necessarily for other tasks. The design of features is also restricted in order to ensure the sparsity of features.

	# train	# test	# labels	card.
`cmc2007`	978	976	45	1.23
`reuters10`	6,490	2,545	10	1.10
`reuters90`	7,770	3,019	90	1.24

Table 1: Statistics of evaluation data sets

	κ	ν	c
`cmc2007`	1,000	10	0
`reuters10`	5,000	20	5
`reuters90`	5,000	80	5

Table 2: Parameters for evaluation data sets

4 Experiments

4.1 Experimental Settings

We evaluate our method for multi-label classification using three data sets for text categorization. Table 1 shows the statistics of these data. In this table, "card." denotes the average number of labels assigned to a document.

`cmc2007` is a data set used in the Computational Medicine Center (CMC) Challenge 2007 (Pestian et al., 2007)[1]. This challenge aimed at the assignment of ICD-9-CM codes, such as *cough* and *pneumonia*, to clinical free texts. It should be noted that this data is controlled, so that both training and test sets include the exact same label combinations, and the number of combinations is 90. This indicates that this task can be solved as a classification of 90 classes. However, since this is an unrealistic situation for actual applications, we do not rely on this characteristic in this work.

`reuters10` and `reuters90` are taken from the Reuters-21578 collection,[2] which is a popular benchmark for text categorization. This text collection consists of newswire articles, and each document is assigned topic categories, such as *grain* and *ship*. We split the data into training and test sets, according to the so-called *ModApte* split.

[1] Available at `http://www.computationalmedicine.org`

[2] Available at `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

cmc2007				
	BPM		ME	
γ	micro-F1	sub. acc.	micro-F1	sub. acc.
0	82.79	69.88	83.09	69.06
100	83.49	70.70	83.68	70.39
200	82.95	69.67	83.67	70.18
400	83.03	69.98	83.49	70.49
800	83.51	71.41	83.58	70.70
1600	83.10	70.49	83.56	71.00
3200	80.74	66.70	82.02	69.57
reuters10				
	BPM		ME	
γ	micro-F1	sub. acc.	micro-F1	sub. acc.
0	94.23	89.71	93.71	88.76
500	94.22	89.98	93.80	89.19
1000	94.43	90.37	94.07	89.55
2000	94.46	90.61	94.04	89.94
4000	94.12	90.26	94.12	89.98
8000	94.14	90.61	94.50	90.81
16000	93.92	90.29	94.30	90.88
reuters90				
	BPM		ME	
γ	micro-F1	sub. acc.	micro-F1	sub. acc.
0	84.07	77.91	86.83	79.50
500	84.96	78.27	86.89	79.66
1000	85.38	78.70	86.94	79.99
2000	85.73	79.79	86.55	79.93
4000	85.72	79.73	86.54	80.23
8000	85.90	80.19	86.77	80.39
16000	86.17	80.52	—	—

Table 3: Accuracy for `cmc2007`, `reuters10`, and `reuters90`

cmc2007			
γ	max. width	avg. width	time (sec.)
0	0	0.00	90
100	2	1.17	132
200	3	1.51	145
400	3	1.71	165
800	4	2.11	200
1600	5	2.93	427
3200	4	3.99	2280
reuters10			
γ	max. width	avg. width	time (sec.)
0	0	0.00	787
500	2	1.72	1378
1000	3	2.00	1752
2000	4	2.16	2594
4000	6	2.90	7183
8000	6	4.22	21555
16000	6	5.67	116535
reuters90			
γ	max. width	avg. width	time (sec.)
0	0	0.00	26172
500	5	1.74	28067
1000	6	2.24	38510
2000	6	3.22	42479
4000	8	3.68	60029
8000	14	4.56	153268
16000	17	6.39	—

Table 4: Tree width and training time for `cmc2007`, `reuters10`, and `reuters90`

From this data, we create two data sets. The first set, `reuters10`, is a subset of the *ModApte* split, to which the 10 largest categories are assigned. The other, `reuters90`, consists of documents that are labeled by 90 categories, having at least one document in each of the training and test sets.

In the following experiments, we run two machine learning classifiers: Bayes Point Machines (BPM) (Herbrich et al., 2001), and the maximum entropy model (ME) (Berger et al., 1996). For BPM, we run 100 averaged perceptrons (Collins, 2002) with 10 iterations for each. For ME, the orthant-wise quasi-Newton method (Andrew and Gao, 2007) is applied, with the hyper parameter for l_1 regularization fixed to 1.0.

We use word unigram features that represent the frequency of a particular word in a target document. We also use features that indicate the *non-existence* of a word, which we found effective in preliminary experiments; feature $f_{l,\bar{w}}(x, y)$ is 1 if $l \in y$ and w is not included in the document x. Words are stemmed and number expressions are normalized to a unique symbol. Words are not used if they are included in the stopword list (322 words), or they occur fewer than a threshold, c, in training data. We set $c = 5$ for `reuters10` and `reuters90`, following previous works (Ghamrawi and McCallum, 2005), while $c = 0$ for `cmc2007`, because the data is small.

These features are selected according to averaged mutual information (information gain), which is the most popular method in previous works (Yang and Pedersen, 1997; Ghamrawi and McCallum, 2005). For each label, features are sorted according to this score, and top-ranked features are included in the model. By preliminary experiments, we fixed parameters, κ for word unigram features and ν for non-existence features, for each data set, as shown in Table 2.

The same method is applied to the selection of label correlation features. In the following experiments, we observe the accuracy and training time by changing the threshold parameter γ for the selection of label correlation features.

4.2 Results

Table 3[3] shows microaveraged F-scores (micro-F1) and subset accuracies (sub. acc.) (Ghamrawi and McCallum, 2005) while varying γ, the num-

[3]The experiment with $\gamma = 16000$ for ME was not performed due to its cost (estimated time is approx. two weeks).

ber of label correlation features. In all data sets and with all classifiers, the accuracy is increased by incorporating label correlation features. The results also demonstrate that the accuracy saturates, or even decreases, with large γ. This indicates that the feature selection is necessary not only for obtaining efficiency, but also for higher accuracy.

Table 4 shows tree widths, and the time for the training of the ME models. As shown, the graphical model is represented effectively with sparse graphs, even when the number of label correlation features is increased. With these results, we can conclude that our method can model label correlations with a tractable cost.

The accuracy for `cmc2007` is significantly better than the results reported in Patrick et al. (2007) (micro-F1=81.1) in a similar setting, in which only word unigram features are used. Our best result is approaching the results of Crammer et al. (2007) (micro-F1=84.6), which exploits various linguistically motivated features. Numerous results have been reported for `reuters10`, and most of them report the microaveraged F-score around 91 to 94, while our best result is comparable to the state-of-the-art accuracy. For `reuters90`, Ghamrawi and McCallum (2005) achieved an improvement in the microaveraged F-score from 86.34 to 87.01, which is comparable to our result.

5 Conclusion

This paper described a method for the exact inference for multi-label classification with label correlation features. Experimental results on text categorization with the CMC challenge data and the Reuters-21578 text collection demonstrated that our method improves the accuracy for text categorization with a tractable cost. The availability of exact inference enables us to apply various machine learning methods not yet investigated in this paper, including support vector machines.

From the perspective of machine learning research, feature selection methods should be reconsidered. While we used a feature selection method that is widely accepted in text categorization research, it has no direct connection with machine learning models. Since feature selection methods motivated by the optimization criteria of machine learning models have been proposed (Riezler and Vasserman, 2004), we expect that the integration of our proposal with those methods will open up a new framework for multi-label classification.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan) and Grant-in-Aid for Young Scientists (MEXT, Japan).

References

Andrew, G. and J. Gao. 2007. Scalable training of l_1-regularized log-linear models. In *24th Annual International Conference on Machine Learning*.

Berger, A. L., S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *2002 Conference on Empirical Methods in Natural Language Processing*.

Cowell, R. G., A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. 1999. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York.

Crammer, K., M. Dredze, K. Ganchev, and P. P. Talukdar. 2007. Automatic code assignment to medical text. In *BioNLP 2007*, pages 129–136.

Ghamrawi, N. and A. McCallum. 2005. Collective multi-label classification. In *ACM 14th Conference on Information and Knowledge Management*.

Herbrich, R., T. Graepel, and C. Campbell. 2001. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279.

Patrick, J., Y. Zhang, and Y. Wang. 2007. Evaluating feature types for encoding clinical notes. In *10th Conference of the Pacific Association for Computational Linguistics*, pages 218–225.

Pestian, J. P., C. Brew, P. Matykiewicz, DJ Hovermale, N. Johnson, K. B. Cohen, and W. Duch. 2007. A shared task involving multi-label classification of clinical free text. In *BioNLP 2007*, pages 97–104.

Riezler, S. and A. Vasserman. 2004. Gradient feature testing and l_1 regularization for maximum entropy parsing. In *42nd Meeting of the Association for Computational Linguistics*.

Schapire, R. E. and Y. Singer. 2000. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Tsoumakas, G. and I. Katakis. 2007. Multi-label classification: an overview. *Journal of Data Warehousing and Mining*, 3(3):1–13.

Yang, Y. and J. O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *14th International Conference on Machine Learning*, pages 412–420.