東海大学紀要情報通信学部
Vol.10,No.2,2017,pp.1-8

Paper

Performance Evaluation of Combined Approach of Topic Model and LMS Estimation for Multi-label Classification

by

ALAMRI Salah Aziz *[1] and Iwao FUJINO *[2]

(received on Sep. 4, 2017 & accepted on Jan. 11, 2018)

Abstract

In this paper, we will give a performance evaluation to a two-stage approach for multi-label classification. At first, we will provide a simple description to our two-stage approach combining topic model and LMS estimation for providing multiple labels to text documents and other kinds of feature data. The first stage of the approach applies unsupervised learning with topic model to obtain a topic distribution for given instances, while the second stage performs supervised learning using the results of the first stage as features. Then we will show some experiment results to evaluate the performance of this combined approach using several typical evaluation open data set for multi-label classification. The results of these experiments confirm that our approach works effectively as expected.

Keywords: *multi-label classification, topic model, latent Dirichlet allocation, least mean square estimation, MLKNN (multi-label k nearest neighbor).*

1. Introduction

In recent years, because of the great advancement of digital and internet technology, tremendous text documents, digital images and various evaluations of products and services can be widely accumulated from the general public. In order to manage such digital information and provide them on web search engines, it becomes necessary to classify them into specific categories or associate them with specific tags. However and for practical problems, each instance could belong to several different classes simultaneously. In contrast with single label classification, this kind of multi-label classification has not reached, to the best of our knowledge, efficient solution so far, not only in the meaning of theoretical viewpoint but also in the meaning of practical method of utilization.

Over the past few years, the multi-label classification problem has attracted increasing attention of researchers involved in machine learning. While early approaches were specially designed for multi-label classification, McCallum proposed an approach using a mixture-based model [1], Schapire and Singer suggested an approach so-called Boostexter [2], while Elisseeff and Weston introduced a ranking algorithm based on a Support Vector Machine (SVM) [3]. Designed as extensions of the k nearest neighbor (kNN) method, multi-label classification algorithms have been also proposed [4] and [5]. However, these kNN based methods are within the range can be considered as lazy learning approach, in which the classifier will not learn from training data set until the test instance is given. In order to derive a label vector for each new given instance, the distances between the new given instance and all training instances must be recalculated and resorted from the beginning. In the case of a large number of training instances or features, this leads to a very expensive calculation cost. There are two key points that should be improved. The first key point is the lazy approach, i.e., one cannot make any preparation in advance even if the training data set is available. The second key point is the feature dimensionality of the instance. In order to speed up the classification processing, it might be necessary to introduce a dimensionality reducing stage, which can provide new features with efficient information for classification whereas its dimensionality is very small comparing with that of the original data.

Following the considerations mentioned above, we developed a two-stage approach for multi-label classification. The first stage of the approach applies unsupervised learning with topic model as a preparation stage. Here topic model is implemented by the LDA (Latent Dirichlet Allocation) [6] method, which gives a topic distribution expression for each document. The second stage performs supervised learning using the results of the first stage as features, while supervised learning we used is least mean square estimation. Our research is based on the following assumption. A topic should be independent from all others under ideal conditions, so that the topic distribution can be a feature vector for each instance. This gives the advantage of offering a great reduction of the feature dimensionality. Furthermore, the label vector is derived

*1 Graduate School of Information and Telecommunication Engineering, Course of Information and Telecommunication Engineering, Master's Program
*2 School of Information and Telecommunication Engineering, Department of Communication and Network Engineering, Professor

by a linear transform from its topic distribution. This leads us to calculate the optimum solution under the least mean square condition. Therefore, the optimum solution of label vector estimation under the least mean square condition is finally derived. Lastly, we introduced a method for deriving the most appropriate decisions from the LMS estimation results. In order to evaluate the proposed methods, we conducted several experiments using typical evaluation data sets of multi-label classification. The results of these experiments confirm that our approach works effectively as expected.

The remainder of this paper is organized as follows. Section 2 surveys a series of related works on multi-label classification. Section 3 describes the basic considerations and methodologies applied by our study. We also derive the optimum solution of the LMS estimation of the label vector from topic distributions of LDA and a training label data. Last, the paper describes a ranking-based decision rule for determining the proper labels from estimation results. In section 4, the suggested method is applied to several typical data sets and gives some discussions while comparing the experiment results of this data set. Finally section 5 concludes the paper and describes a few directions for future research.

2. Related Works

MLKNN is a binary relevance learner initially proposed by Zhang and Zhou [4]. It learns to build a classifier according to each label taken separately. It inferences each label according to a maximum a posteriori principle and calculates the posteriori probability by counting the number of instance with specific label in the *k*-nearest neighbors. For each given instance, at first identify its *k*-nearest neighbors in the training set. Then, the conditional probability is applied by counting the instances with similar labels. Next, according to the Bayes theorem, the posteriori probability is derived. This paper showed some experimental results on Yeast data set, nature scene data set and Yahoo data set, which confirmed their method outperformed conventional algorithms.

The IBLR (Instance-Based Logistic Regression) algorithm, introduced by Cheng and Hüllermeier [7], combines instance-based learning and logistic regression techniques, where they include the statistics of *k*-nearest neighbors as features in the logistic regression. Considering label information of neighbored examples as features of a query instance, the idea of IBLR is to reduce instance-based learning formally to logistic regression. This approach allows one to capture interdependencies between class labels for multi-label classification. According to the reported experimental results, the proposed approach improves predictive accuracy in terms of several evaluation criteria.

The NBML algorithm proposed by Wei and colleagues [8] is designed by adapting single label Naive Bayesian classifier to multi-label classification. A two-step feature selection strategy is first applied, and aims to satisfy the assumption of conditional independency given by Naive Bayes classification theory. While first deriving the posteriori probability for each label, the average value of posteriori probability of all labels is used as a threshold to determine which label the instance belongs to. This paper showed some experimental results on Yahoo data set, which gives highly competitive performance with several famous algorithms.

3. Basic Considerations and Methodology

This section describes the basic considerations and methodology of our study. At first the basic main principles of our two-stage approach are introduced. Next the principles behind the derivation of an optimum solution for LMS estimation of multi-label classification are developed. Finally, concrete decision rules for determining the label from the result of LMS estimation are presented.

3.1 Basic Considerations

As mentioned above, in the case of massive training instances and a large number of features contained in each instance, MLKNN and other kNN based methods cause very expensive calculation cost. As opposed to lazy learning, we adopt eager learning in our study, i.e. we invest our effort on training the classifier. We think a two-stage structure of unsupervised learning and supervised learning will benefit to solve this problem. For details, our study is performed based on the following basic considerations.

(1) Introducing unsupervised learning LDA algorithm as a preprocessing procedure, so that we can obtain mutually independent features and reduce the feature dimensionality from the number words (which is several ten thousands in usual) to the number of topics (which is a few dozen or about one hundred in usual).

(2) After the preprocessing procedure, because the topic is independent from each other under ideal condition, i.e., topics will make a vector space and a multi-label of an instance can be thought a vector of this vector space or subspace. Therefore, the LMS estimation is applied to get the optimum solution for the transform matrix.

(3) In order to deride the appropriate labels from the estimated label probability of a given instance, we identify a series decision rules with regards with the average and standard

deviation of the label length in training data set, maximum value, mean value and standard deviation of estimated probability.

3.2 Optimum Solution of the Estimated Label Vector

Topic model is a probabilistic generative model applied to text document. An implementation of the topic model called LDA has been proposed by Blei et al. [6]. A graphical model representation of LDA is shown in Fig.1, where θ is a topic proportion on documents, and ϕ is a word distribution on words. Also α is a hyper parameter for generating θ and β is a hyper parameter for generating ϕ. Furthermore *z* is a topic matrix and *w* is a word count matrix for each document and each word.

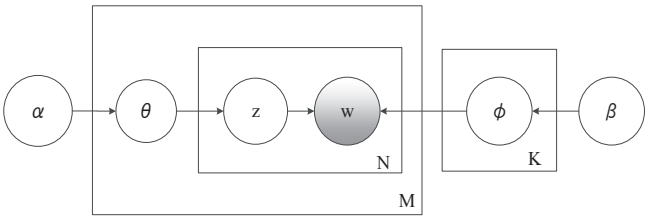

Fig. 1 Graphical model representation of LDA

The results of the LDA processing give a topic proportion vector for each document as follows.

$$\boldsymbol{\theta}_d = \begin{pmatrix} \theta_{d1} \\ \theta_{d2} \\ \vdots \\ \theta_{dK} \end{pmatrix} \ (d = 1,2,\dots,M) \tag{1}$$

where *M* denotes the number of documents and *K* denotes the number of topics. Here let us suppose that we can transform the topic proportion vector $\boldsymbol{\theta}_d$ to label probability vector $\hat{\boldsymbol{L}}_d$ by introducing a weight matrix $\boldsymbol{W}$ as follows.

$$\hat{\boldsymbol{L}}_d = \boldsymbol{W}\boldsymbol{\theta}_d \ (d = 1,2,\dots,M) \tag{2}$$

where $\hat{\boldsymbol{L}}_{\boldsymbol{d}}$ and $\boldsymbol{W}$ can be expressed as follows.

$$\hat{\boldsymbol{L}}_{\boldsymbol{d}} = \begin{pmatrix} p_{d1} \\ p_{d2} \\ \vdots \\ p_{dL} \end{pmatrix} \ (d = 1,2,\dots,M) \tag{3}$$

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1K} \\ w_{21} & w_{22} & \cdots & w_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ w_{L1} & w_{L2} & \cdots & w_{LK} \end{pmatrix} \tag{4}$$

where *L* denotes the number of labels. In order to determine the weight matrix, from least mean square (LMS) criterion, the evaluation function can be defined as follows.

$$J = \mathrm{E}\left[\sum_{n=1}^{L} e_n^2\right] = E[\boldsymbol{e}^T\boldsymbol{e}] \tag{5}$$

where $\boldsymbol{e} = \boldsymbol{L}_d - \hat{\boldsymbol{L}}_d$ is the error vector between true label vector $\boldsymbol{L}_d$ and estimated label vector $\hat{\boldsymbol{L}}_d$. After derivation of the matrix equation, let us derive the evaluation function to the following expression.

$$J = E[\boldsymbol{L}_d^T\boldsymbol{L}_{\boldsymbol{d}}] - 2E[\boldsymbol{L}_d^T\boldsymbol{W}\boldsymbol{\theta}_d] + E[\boldsymbol{\theta}_d^T\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{\theta}_d] \tag{6}$$

Then by differentiating this evaluation function J by weight matrix $\boldsymbol{W}$, the following expression is obtained.

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{W}} &= -2E[\boldsymbol{L}_d\boldsymbol{\theta}_d^T] + 2\boldsymbol{W}E[\boldsymbol{\theta}_d\boldsymbol{\theta}_d^T] \\ &= -2R_{L\theta} + 2\boldsymbol{W}R_{\theta\theta} \end{aligned} \tag{7}$$

Then from $\frac{\partial J}{\partial \boldsymbol{W}} = 0$, we can get the optimum solution for the weight matrix $\boldsymbol{W}^*$ as follows.

$$\boldsymbol{W}^* = R_{L\theta}R_{\theta\theta}^{-1} \tag{8}$$

Finally, we assume that this transform relation topic proportion vector and label probability vector will not change with time, so for a given topic proportion vector $\boldsymbol{\theta}_t$ from test set, we can estimate label vector $\hat{\boldsymbol{L}}_t$ according to the following expression.

$$\hat{\boldsymbol{L}}_t = R_{L\theta}R_{\theta\theta}^{-1}\boldsymbol{\theta}_t \tag{9}$$

$$y_l = \begin{cases} True & p(l=1|\boldsymbol{D}) > p(l=0|\boldsymbol{D}) \\ False & otherwise \end{cases} \tag{10}$$

3.3 Label Decision Rule from the Optimum Solution of Estimated Label Vector

In the case of bipartition decision, the result for each label is decided according to the following decision rule.

However, in practical cases, it may be assumed that there is one label at least for each instance, but sometimes bipartition decision rule gives no *True* label in the result at all. To deal with this problem, a method using the mean value of all labels as a threshold has been reported [8]. By extending this method, we define a few rules to take a final decision for each label. From an optimum solution of estimated label vector $\hat{\boldsymbol{L}}$, we can sort its elements in descending order shown as follows.

$$p_{l_1} > p_{l_2} > \cdots > p_{l_f} > \cdots > p_{l_L}$$

and then select the label $l_1 \sim l_f$ as true label if the following two rules are satisfied.

(1) Suppose μ_t mean value of label length and σ_t is the standard deviation of label length in training data set. The first rule is given as follows.

$$f \leq int(\mu_t + 3\sigma_t) \tag{11}$$

(2) Suppose ρ is the maximum value, μ is the mean value and σ is the standard deviation of probability p_{l_i} $(i = 1,2, \dots L)$. The second rule is given as follows.

$$y_{l_i} = \begin{cases} True & p_{l_i} > \max(\rho - 3\sigma, \mu) \\ False & otherwise \end{cases} \tag{12}$$

4. Experiments

In this section, at first we describe the data sets, several performance measures and computer environment for our experiment. Then we show the statistics summary of each data set, At last we report the experimental results of the MLKNN algorithm as a baseline and the experimental results of our proposed approach with the number of topics as a parameter. About the number of topics, we think it should be bigger than the number of labels and much smaller than the number of features.

4.1 Data Set for Confirmation Experiments

In order to evaluate our proposed approach, several experiments have been conducted using typical data sets. The baseline for these experiments is MLKNN algorithm [4]. Firstly, as a confirmation experiment, we selected the Genbase data set and Yeast data set. The Genbase dataset is provided by S. Diplaris, G. Tsoumakas, P. Mitkas and I. Vlahavas [9], which is formed for classifying proteins to structural families according to its functions. The Yeast dataset is provided by A. Elisseeff and J. Weston, which is formed for classifying gene to functional classes [3]. However Genbase is a kind of easy level data set and Yeast is a kind of difficult level data set for multi-label classification problem. After that as a confirmation experiment for practical use, we selected the Yahoo data set. This data set is provided by N. Ueda, K. Saito [10], which is formed for classifying Web page to a number of different categories. All these data sets are downloaded from the Mulan website[+1].

4.2 Measures Retained for Describing the Data Set

Suppose that the data set we used for experiments is $D(x_i, Y(x_i))$, x_i denotes a feature and $Y(x_i)$ is a label with regards to x_i. Also suppose the data set has *T* instances and *L* labels. To describe the basic conditions of the data set for experiments, we use these parameters and defined as follows.

(1) Label cardinality [11]: label cardinality is the average number of labels for each document. Its definition is shown as follows.

$$LC = \frac{1}{T}\sum_{i=1}^{T}|Y(x_i)| \tag{13}$$

(2) Label density [11]: label density is the average number of average labels for each document. Its definition is shown as follows.

$$LD = \frac{1}{T}\sum_{i=1}^{T}\frac{|Y(x_i)|}{L} \tag{14}$$

(3) μ_{MC}: μ_{MC} denotes the mean of maximum value in cross correlation coefficient between the labels of different document. Its definition is shown as follows.

$$\mu_{MC} = \text{mean}\Big(\max\big(corr\big(Y(x_i), Y\big(x_j\big), (j = 1, \dots L, j \neq i)\big), (i = 1, \dots L)\Big) \tag{15}$$

4.3 Performance Evaluation Measures

In order to evaluate a given multi-label classification test set with *T* instances and *L* labels, the evaluation metrics used are shown as follows.

(1) To focus on the performance with regard to each label, we use the F-measure to evaluate the results. The definition of precision [11], recall [11] and F-measure are given as follows.

$$Precision = \frac{1}{T}\sum_{i=1}^{T}\frac{|Y(x_i) \cap Z(x_i)|}{|Z(x_i)|} \tag{16}$$

$$Recall = \frac{1}{T}\sum_{i=1}^{T}\frac{|Y(x_i) \cap Z(x_i)|}{|Y(x_i)|} \tag{17}$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{18}$$

(2) To focus the performance with regard to each instance, we use hamming loss to evaluate the results. For a binary multi-label

+1 http://mulan.sourceforge.net/datasets-mlc.html

classifier h, hamming loss is an average value of how many times an instance-label pair is misclassified, which definition is shown as follows [5], where Δ stands for the symmetric difference of two sets.

$$HammingLoss(h) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{L}|H(x_i)\Delta Y(x_i)| \qquad (19)$$

(3) To focus the performance with regards to ranking approach, we use one error to evaluate the results. For a ranking based classifier h, one error is an average value of how many times the top-ranked is not in the set of true labels of the instances, which definition is shown as follows [5].

$$OneError(f) = \frac{1}{T}\sum_{i=1}^{T} I\left(argmax_{\lambda \ni L} f(x_i, \lambda) \notin Y(x_i)\right) \qquad (20)$$

4.4 Computer Environment

The computer environment of our experiment is shown as follows:

Hardware: HP ZBook 17 Mobile Workstation (CPU: Intel Core i7-4800MQ, Memory: 32GB, HDD: 480GB)

Software: OS: Ubuntu 14.04 LTS (64bit), Programing language: Python 2.7, Software package: numpy 1.9.1, scipy 0.13.3.

4.5 Confirmation Experiments

In order to evaluate our approach, we made a computer program according to our proposed method and we also made a computer program according to the MLKNN algorithm [4] as a baseline of evaluation. These experiments aim to confirm our computer programs for the experiments. The datasets we used are Genbase and Yeast. The statistics of both data sets are shown in Table 1.

The results of the multi-label classification experiment for the Genbase data set are shown in Table 2, where the number of topics for our approach is from 20 to 120. From the results for the MLKNN algorithm, we can confirm that the results of hamming-loss and one-error show almost same value as reported in [5] and [7]. The results show that both MLKNN and our proposed method achieved satisfactory figures for each performance measure. Also our proposed method shows best value of F-measure where the number of topics t=100. About the value of label cardinality, MLKNN is 1.15 which is smaller than the one of the original data set, whereas the label cardinality of our approaches are 1.23~1.27, which is close to the value of the original data set. With respect to calculation time, although the value of our approach changes with the number of topics, it is still very small compared to that of MLKNN.

The results of multi-label classification experiment for the Yeast data set are shown in Table 3, where the number of topics for our approach is from 5 to 25. From the results for the MLKNN algorithm, we can confirm that the results of hamming-loss and one-error show almost same value as reported in [4], [5] and [7]. The results show that the value of the performance measures of the proposed method is slightly poor comparing with that of MLKNN. For this data set, because the number of features is 103 and the number of labels is 14, so there is no proper value for topic number for our proposed method according to the rule to choose the topic number mentioned above. The improper topic number assigned may affect the independence between topics and this causes the slightly poor result. Also, as seen in the results of Genbase data set, MLKNN tends to give smaller label cardinality whereas our approach tends to give higher label cardinality compared to the one of the original data set. With respect to calculation time, our approaches show from 21.1% to 36.6% increased than MLKNN.

4.6 Confirmation Experiments with Yahoo Web Page Categorization Data Sets

For confirmation experiments to classify a practical multi-label text, a Yahoo web page data set has been selected. We retained a data set for automatic web page categorization introduced by Ueda and Saito in 2003 [10]. The statistics for each category for the Yahoo data set is shown in Table 4. From a bag-of-words description of documents, topic model can give a topic distribution for each document and a word distribution for each topic. Because the number of topics is quite lower than the number of words, by introducing the topic model at the preprocessing stage, we can reduce the dimensionality of feature significantly. For implementing MLKNN algorithm for Yahoo data set, we preprocessed each data set to extract words with highest 2% document frequency based on the report of Yang and Pedersen [12], although we think this may lead to some all zero feature instances. But as the MLKNN binary bipartition algorithm does not always give label results, i.e., the results for all labels are possible *False* for some instances, we introduced a ranking-based decision rule as mentioned in section 3.3, which is denoted as MLKNN+Ranking in the tables of the experimental results. Also, and for all of experiments, we used all instances in the training set to train the classifier and all instances in the test set to evaluate the performance of the trained classifier. The experimental results of F-measure, hamming-loss and one-error are shown in Table 5, 6, 7, respectively, where the number of topics for our approach is from 10 to 90. From these results, one can see that, for each performance

measure, as an average of all individual data set, our proposed approach shows at least 25% improvement than MLKNN+Ranking. For details, as shown in Table 5, all the 11 data sets using our proposed approach show better F-measure performance, with 68.2% improvement on maximum and 25.7% improvement on average. Then as shown in Table 6, 10 out of 11 data sets using our proposed approach show better hamming-loss performance, with 40.6% improvement on maximum and 25.3% on average. Also as shown in Table 7, all the 11 data sets using our proposed approach show better one-error performance, with 46.6% improvement on maximum and 30.7% on average. Also from the data of Table 5, a line graph of F-measure vs. the number of topics for each individual data set is shown in Fig.2. This graph shows that the higher value of the number of topics the better performance we have mostly, but this increase becomes gradually slow while the increasing of the number of topics.

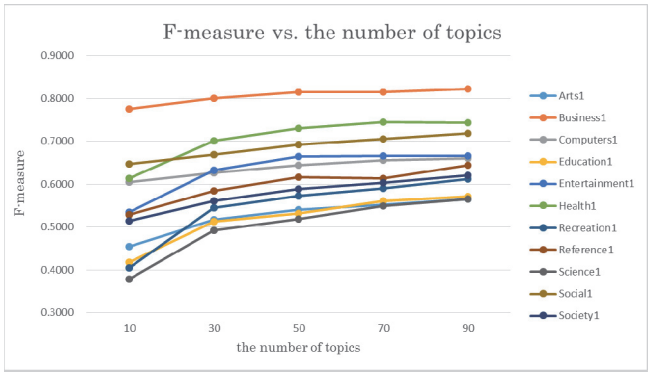

Fig. 2 F-measure vs. the number of topics

5. Conclusions

This paper presents a two-stage approach for multi-label classification. The first stage is an unsupervised learning stage to implement mutually independent features and also dimensionality reducing by introducing topic model, while the second stage is supervised learning to perform a topic-label transform according to a least mean square estimation. The experimental results on public evaluation data set show that our proposed approach is suitable for the case with big amount training instances and a large number of features. However the experimental results also show that our proposed approach is not suitable for the case with a small number of features like Yeast data set. For Yahoo web page categorization data set, we achieved remarkable performance improvement comparing with MLKNN algorithm in all of F-measure, hamming-loss and one-error measures. It also appears that calculation time can be decreased for the case of a large number of features.

Although we have achieved many progresses, the current performances are still not good enough for practical use. For practical problems, an important factor for improving performance is the correlation between labels. In future work, we would like to explore a method to remove the cross correlation between labels by introducing matrix factorization. By its result we expect to adopt Naive Bayesian approach to multi-label classification under ideal condition and this will bring us more performance improvements with relatively small calculation cost.

References

[1] A. K. McCallum: Multi-label Text Classification with a Mixture Model Trained by EM, Proceedings of AAAI'99 Workshop on Text Learning, Orlando FL (1999)

[2] R. E. Schapire and Y. Singer: Bootexter: A Boosting-based System for Text Categorization, Machine Learning, vol.39(2-3), pp.135-168 (2000)

[3] A. Elisseeff and J. Weston: A Kernel Method for Multi-labelled Classification, In T. G. Dietterich, S. Becker and Z. Ghahramani (eds) Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, pp.681-687 (2001)

[4] M. Zhang and Z. Zhou: ML-KNN: A Lazy Learn Approach to Multi-label Learning. Pattern Recognition, vol.40(7), pp.2038-2048 (2007)

[5] T. Chiang, H. Lo and S. Lin: A Ranking-based KNN Approach for Multi-label Classification, JMLR: Workshop and Conference Proceedings, vol.25, pp.81-96 (2012)

[6] D. M. Blei, A.Y. Ng and M. I. Jordan: Latent Dirichlet Allocation, Journal of Machine Learning Research, vol.3, pp.993-1022 (2003)

[7] W. Cheng and E. Hüllermeier: Combining Instance-based Learning and Logistic Regression for Multilabel Classification, Machine Learning, vol.76(2-3), pp.211-225 (2009)

[8] Z. Wei, H. Zhang, Z. Zhang, W. Li and D. Miao: A Naive Bayesian Multi-label Classification Algorithm with Application to Visualize Text Search Results, International Journal of Advanced Intelligence, vol.3(2), pp. 173-188 (2011)

[9] S. Diplaris, G. Tsoumakas, P. Mitkas and I. Vlahavas: Protein Classification with Multiple Algorithms, Proc. 10th Panhellenic Conference on Informatics (PCI 2005), pp. 448-456 (2005).

[10] N. Ueda and K. Saito: Parametric Mixture Models for Multi-label Text, in: S. Becker, S. Thrun and K. Obermayer (eds.) Advances in Neural Information Processing System 15, MIT Press, Cambridge, MA, pp.721-728 (2003)

[11] G. Tsoumakas and I. Katakis: Multi-Label Classification: An Overview, International Journal of Data Warehousing and Mining, vol.3(3), pp.1-13 (2007)

[12] Y. Yang and J. O. Pedersen: A Comparative Study on Feature Selection in Text Categorization, Proceedings of the International Conference on Machine Learning, pp.412-420 (1997)

Table 1 Statistics of Genbase and Yeast data sets

Data set	Instances in training set	Instances in test set	Features	Labels	Label Cardinality	Label Density	Mean of max. crosscorr.
genbase	463	199	1186	27	1.252	0.0463	0.4436
yeast	1500	917	103	14	4.237	0.3026	0.6149

Table 2 Experimental results for the Genbase data sets

Algorithm	Precision	Recall	F-measure	Hamming Loss	One Error	Cardinality	Time(s)
MLKNN(k=10)	**0.9899**	0.9669	0.9782	**0.0035**	0.0100	1.15	148
LDA(t=20)+LMS	0.9497	0.9618	0.9557	0.0080	0.0301	1.23	12
LDA(t=40)+LMS	0.9547	0.9585	0.9566	0.0085	0.0201	1.22	18
LDA(t=60)+LMS	0.9581	**0.9819**	0.9699	0.0057	**0.0050**	1.27	25
LDA(t=80)+LMS	0.9648	0.9769	0.9708	0.0059	0.0201	1.23	30
LDA(t=100)+LMS	0.9790	0.9778	**0.9784**	0.0046	0.0251	1.23	40
LDA(t=120)+LMS	0.9526	0.9698	0.9746	0.0048	0.0150	1.23	45

Table 3 Experimental results for the Yeast data sets

Algorithm	Precision	Recall	F-measure	Hamming Loss	One Error	Cardinality	Time(s)
MLKNN(k=10)	**0.7322**	0.5491	**0.6275**	**0.1980**	0.2846	3.14	142
LDA(t=5)+LMS	0.4932	**0.7631**	0.5992	0.3100	**0.2497**	6.42	172
LDA(t=10)+LMS	0.4920	0.7581	0.5967	0.3127	0.2508	6.37	178
LDA(t=15)+LMS	0.4930	0.7415	0.5926	0.3127	0.2617	6.24	193
LDA(t=20)+LMS	0.4911	0.7517	0.5940	0.3141	0.2595	6.34	189
LDA(t=25)+LMS	0.4973	0.7314	0.5920	0.3084	0.2671	6.13	194

Table 4 Statistics of the Yahoo data set

Data set	Instances in training set	Instances in test set	Features	Labels	Label cardinality	Label density	Mean of max. cross-corr.
Arts1	3712	3772	23146	26	1.6539	0.06361	0.2159
Business1	5710	5504	21924	30	1.5989	0.05329	0.4028
Computers1	6270	6174	34096	33	1.5072	0.04567	0.3085
Education1	6030	6000	27534	33	1.4631	0.04433	0.1419
Entertainment1	6556	6374	32001	21	1.4137	0.06732	0.1998
Health1	4557	4648	30605	32	1.6441	0.05137	0.2915
Recreation1	6471	6357	30324	22	1.4289	0.06495	0.2214
Reference1	4027	4000	39679	33	1.1744	0.03558	0.1102
Science1	3214	3214	37187	40	1.4497	0.03624	0.2875
Social1	6037	6074	52350	39	1.2792	0.0328	0.2729
Society1	7273	7239	31802	27	1.6704	0.06186	0.1929

Table 5 F-measure results for the Yahoo multi-label data set

Data Set	MLKNN(k=10) +Ranking	LDA+LMS t=10	LDA+LMS t=30	LDA+LMS t=50	LDA+LMS t=70	LDA+LMS t=90
Arts1	0.3908	0.4550	0.5180	0.5403	0.5520	**0.5650**
Business1	0.7905	0.7753	0.8012	0.8149	0.8154	**0.8219**
Computers1	0.5845	0.6056	0.6270	0.6447	0.6562	**0.6602**
Education1	0.4505	0.4178	0.5133	0.5323	0.5616	**0.5709**
Entertainment1	0.4997	0.5343	0.6332	0.6650	0.6665	**0.6668**
Health1	0.5782	0.6139	0.7006	0.7308	**0.7445**	0.7432
Recreation1	0.3638	0.4056	0.5450	0.5735	0.5899	**0.6121**
Reference1	0.5346	0.5295	0.5838	0.6159	0.6133	**0.6441**
Science1	0.3658	0.3787	0.4923	0.5181	0.5497	**0.5656**
Social1	0.6405	0.6467	0.6698	0.6928	0.7062	**0.7188**
Society1	0.5194	0.5146	0.5608	0.5883	0.6037	**0.6214**
average	0.5198	0.5343	0.6041	0.6288	0.6417	**0.6536**

Table 6 Hamming-loss results for the Yahoo multi-label data set

Data Set	MLKNN(k=10) +Ranking	LDA+LMS t=10	LDA+LMS t=30	LDA+LMS t=50	LDA+LMS t=70	LDA+LMS t=90
Arts1	0.1283	0.1058	0.0996	0.0954	0.0922	**0.0910**
Business1	0.0293	0.0294	0.0274	0.0260	0.0264	**0.0256**
Computers1	**0.0501**	0.0588	0.0557	0.0523	0.0518	0.0518
Education1	0.0723	0.0758	0.0624	0.0605	0.0568	**0.0564**
Entertainment1	0.1268	0.1133	0.0899	**0.0839**	0.0859	0.0867
Health1	0.0633	0.0585	0.0453	0.0415	**0.0393**	0.0398
Recreation1	0.1565	0.1440	0.1060	0.1018	0.0958	**0.0929**
Reference1	0.0373	0.0444	0.0385	0.0357	0.0364	**0.0339**
Science1	0.0649	0.0639	0.0503	0.0499	0.0470	**0.0456**
Social1	0.0317	0.0334	0.0322	0.0305	0.0293	**0.0286**
Society1	0.0752	0.0877	0.0848	0.0794	0.0782	**0.0717**
average	0.0760	0.0741	0.0629	0.0597	0.0581	**0.0567**

Table 7 One-error results for Yahoo multi-label data set

Data Set	MLKNN(k=10) +Ranking	LDA+LMS t=10	LDA+LMS t=30	LDA+LMS t=50	LDA+LMS t=70	LDA+LMS t=90
Arts1	0.6219	0.5501	0.4844	0.4509	0.4337	**0.4072**
Business1	0.1226	0.1337	0.1182	0.1121	0.1115	**0.1099**
Computers1	0.3994	0.3809	0.3735	0.3506	0.3436	**0.3346**
Education1	0.5575	0.6218	0.4945	0.4733	0.4406	**0.4256**
Entertainment1	0.5381	0.5070	0.3652	0.3239	0.3264	**0.3214**
Health1	0.4225	0.3704	0.2747	0.2409	0.2181	**0.2252**
Recreation1	0.6683	0.6138	0.4366	0.4056	0.3795	**0.3706**
Reference1	0.4580	0.4767	0.4247	0.3905	0.3812	**0.3655**
Science1	0.6505	0.6552	0.5211	0.4863	0.4508	**0.4306**
Social1	0.3547	0.3646	0.3379	0.3052	0.2892	**0.2820**
Society1	0.4438	0.4772	0.4272	0.4032	0.3751	**0.3548**
average	0.4761	0.4683	0.3871	0.3584	0.3409	**0.3298**