GENERALIZED *K*-LABELSET ENSEMBLE FOR MULTI-LABEL CLASSIFICATION

Hung-Yi Lo[1,2]*, Shou-De Lin*[2]*, Hsin-Min Wang*[1]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2]Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{hungyi, whm}@iis.sinica.edu.tw, sdlin@csie.ntu.edu.tw

ABSTRACT

Label powerset (LP) method is one category of multi-label learning algorithms. It reduces the multi-label classification problem to a multi-class classification problem by treating each distinct combination of labels in the training set as a different class. This paper proposes a basis expansion model for multi-label classification, where a basis function is a LP classifier trained on a random k-labelset. The expansion coefficients are learned to minimize the global error between the prediction and the multi-label ground truth. We derive an analytic solution to learn the coefficients efficiently. We have conducted experiments using several benchmark datasets and compared our method with other state-of-the-art multi-label learning methods. The results show that our method has better or competitive performance against other methods.

Index Terms— Multi-label classification, ensemble method, labelset.

1. INTRODUCTION

Multi-label classification has attracted a great deal of attention in recent years. In a conventional single-label classification task, given a set of K possible disjoint classes, each instance is associated with one and only one class. In multi-label classification, an instance could be associated with a set of labels jointly. For example, in a music tagging website, a song might be jointly tagged as "vocal", "slow", and "r&b". In a social bookmarking website, a website might be tagged as "handheld", "technology", and "mobile". In image classification tasks, an image may contain several concepts, such as "sea" and "sunset". Such prediction tasks are usually denoted as multi-label classification problems.

Label powerset (LP) [1] method is one category of multi-label learning algorithms. It reduces the multi-label classification problem to a *single-label multi-class* classification problem by treating each distinct combination of labels in the training set as a different class. Given a test instance, the multi-class LP classifier predicts the most probable class and can be transformed to a set of labels. Table 1 shows an example of multi-label dataset with transformed multi-class label based on LP. In contrast to the binary relevance approach, which loses the label dependency information while learning a binary classifier for each label independently, the LP method exploits conditional label dependency information by learning the joint label distribution [2]. However, when the number of labels increases, the number of potential classes increases proportionally, and each class will be associated with very few training instances. Moreover, LP can only predict labelsets observed in the training data.

Table 1. An Example of Multi-Label Dataset with Transformed Multi-Class Label

Instance	Label Set	Transformed Class
1	Rock, Guitar	1
2	Rock, Guitar, Drum	2
3	Rock, Guitar, Vocal	3
4	Country, Guitar	4
5	Rock, Guitar, Drum	2
6	R&B, Vocal	5
7	Country, Guitar	4
8	Vocal	6

In [3], a method called Random *k*-Labelsets (RA*k*EL) is proposed to overcome the drawback of the LP method. RA*k*EL randomly selects a number of label subsets from the original set of labels and uses the LP method to train the corresponding multi-label classifiers. The final prediction of RA*k*EL is made by voting of the LP classifiers in the ensemble. By using this method, each of the transformed multi-class classification problems are computationally simpler since the number of classes is reduced, and each class will be associated with more training instances. Previous research [4] states that "a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the individual classifiers are accurate and diverse". An accurate classifier is the one that has an error rate better than random guessing. In RA*k*EL, the property of classifiers diversity is achieved by randomly selecting label subsets. Experiment results have shown the improvement of RA*k*EL over LP.

This paper is inspired by another ensemble method, AdaBoost [5]. The decision function of AdaBoost is a weighted sum of base classifiers $f_t(\boldsymbol{x})$ of the following form: $F(\boldsymbol{x}) = \sum_t \alpha_t f_t(\boldsymbol{x})$. For each iteration t, AdaBoost algorithm maintains a weight vector D_t over training instances and uses the base learner to find a model $f_t(\boldsymbol{x})$ to minimize a weighted error according to D_t. The coefficient α_t controls the influence of each base classifier $f_t(\boldsymbol{x})$ to the decision function and is determined analytically by minimizing a function of training error upper bound. The derived function for calculating α_t shows that α_t is proportional to the predictive performance of $f_t(\boldsymbol{x})$ on the weighted error according to D_t.

Motivated by the success of AdaBoost learning procedure, this paper propose a *basis expansion* model [6] for multi-label classifica-

This work was partially supported by the Taiwan e-Learning and Digital Archives Program (TELDAP) sponsored by the National Science Council of Taiwan under Grant: NSC100-2631-H-001-013.

tion of the following form:

$$H(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m h_m(\boldsymbol{x}), \tag{1}$$

where the basis functions $h_m(\boldsymbol{x})$ are the LP classifiers trained on random *k*-labelsets. General basis expansion model treats the classifiers $h_m(\boldsymbol{x})$ as *dictionary functions* and tries to approximate the unknown target classifier using the linear combination of the dictionary functions. Intuitively, the basis expansion model is more flexible and has better chance to accurately approximate the target classifier. In our proposed method, the coefficients $\boldsymbol{\beta}$ are learned to minimize the *global error* between the prediction of $H(\boldsymbol{x})$ and the multi-label ground truth. Since each *k*-labelset is selected randomly, some of the LP classifiers inevitably have bad predictive performance and might be redundant. The coefficients β_t can be considered as reflecting the predictive performance or redundancy of the corresponding classifiers $h_m(\boldsymbol{x})$.

When using basis expansion model, controlling the complexity to avoid overfitting is very important. We propose a regularized objective function to overcome this problem. The proposed objective function is regularized by both the two-norm of $\boldsymbol{\beta}$ and a hypergraph Laplacian regularizer. The hypergraph captures the high-order relation among instances and multiple labels jointly. It has been successfully used in feature transformation [7] for multi-label classification and an SVM-based multi-label classifier [8]. We derive an analytic solution to learn the coefficients efficiently. The RA*k*EL [3] can be considered as a special case of our proposed method when the coefficients are assigned *uniformly*. We have conducted experiments using several multi-label benchmark datasets and compared our method with other state-of-the-art multi-label learning methods. The results show that our method has better or competitive performance against other methods.

The remainder of this paper is organized as follows. In Section 2 we present the Generalized *k*-Labelset Ensemble (GKL). Then we present and discuss the results of our experiments in Section 3. Finally, Section 4 contains some concluding remarks.

Algorithm 1 The training process of GKL

- **Input:** number of models M, size of labelset k, learning parameters B and C, set of labels $\mathcal{L}$, and the training set $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$
- **Output:** an ensemble of LP classifiers g_m, the corresponding *k*-labelsets $\mathcal{R}_m$ and coefficients β_m

1. Initialize $\mathcal{S} \leftarrow \mathcal{L}^k$
2. **for** $m \leftarrow 1$ **to** min(M,$|\mathcal{L}^k|$) **do**
   - $\mathcal{R}_m \leftarrow$ a *k*-labelset randomly selected from $\mathcal{S}$
   - train the LP classifier g_m based on $\mathcal{D}$ and $\mathcal{R}_m$
   - calculate a transformed prediction of g_m using (3)
   - $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{R}_m$
3. **end**
4. Learn $\boldsymbol{\beta}$ using (4)

Algorithm 2 The classification process of GKL

- **Input:** number of models M, a test sample $\boldsymbol{x}$, an ensemble of LP classifiers g_m, and the corresponding k-labelsets $\mathcal{R}_m$ and coefficients β_m
- **Output:** the multi-label classification vector $\boldsymbol{r} = (r_1, r_2, ..., r_K)$

1. **for** $j \leftarrow 1$ **to** K **do**
   (a) $r_j = 0$
   (b) **for each** g_m, **if** $j \in \mathcal{R}_m$ **do**
   - $r_j = r_j + \beta_m \cdot q(g_m(\boldsymbol{x}), j)$

   (c) **end**
2. **end**

2. GENERALIZED *K*-LABELSET ENSEMBLE

We first introduce the concept of multi-label classification. Let $\boldsymbol{x} \in \mathbb{R}^d$, which is a d-dimensional input space, and $\mathcal{Y} \subseteq \mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_K\}$, which is a finite set of K possible labels. To facilitate the discussion, hereafter, $\mathcal{Y}$ is represented by a vector $\boldsymbol{y} = (y_1, y_2, ..., y_K) \in \{1, -1\}^K$, in which $y_j = 1 \Leftrightarrow \lambda_j \in \mathcal{Y}, y_j = -1 \Leftrightarrow \lambda_j \notin \mathcal{Y}$. We denote the labels of the whole instances by $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$, where the i-th row of $\boldsymbol{Y}$ is $\boldsymbol{y}_i$. Given a training set $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$ that contains N samples, the goal of multi-label classification is to learn a classifier $H : \mathbb{R}^d \rightarrow 2^K$ such that $H(\boldsymbol{x})$ predicts which labels should be assigned to an unseen sample $\boldsymbol{x}$.

Algorithms 1 and 2 describe the training and classification processes of the proposed GKL, respectively. A *k*-labelset is a labelset $\mathcal{R} \subseteq \mathcal{L}$ with $|\mathcal{R}| = k$. In the training stage of GKL, the coefficients $\boldsymbol{\beta}$ are learned by solving a minimization problem formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \tfrac{1}{2}||\boldsymbol{Y} - \sum_{m=1}^{M} \beta_m Q_m||_F^2 + \tfrac{B}{2}||\boldsymbol{\beta}||_2^2 \\ & + \tfrac{C}{2}\text{trace}\left(\Big(\sum_{m=1}^{M} \beta_m Q_m \Big)^T \boldsymbol{L} \Big(\sum_{m=1}^{M} \beta_m Q_m \Big) \right), \end{aligned} \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm of a matrix, $\boldsymbol{L}$ is the hypergraph Laplacian, and $Q_m \in \mathbb{R}^{N \times K}$ is a transformed prediction of g_m which will be described in more detail below. The prediction of a multi-class LP classifier, g_m, for a sample $\boldsymbol{x}$ is denoted by $g_m(\boldsymbol{x}) \in \{1, 2, \ldots, V\}$. Note that V will be much smaller than 2^k if the data is sparse. The i, j-th element in Q_m is calculated by $q(g_m(\boldsymbol{x}_i), j)$, which is defined as:

$$q(g_m(\boldsymbol{x}_i), j) = \begin{cases} 1, & \text{if } j \in \mathcal{R}_m \text{ and } j \text{ is positive in } g_m(\boldsymbol{x}_i), \\ -1, & \text{if } j \in \mathcal{R}_m \text{ and } j \text{ is negative in } g_m(\boldsymbol{x}_i), \\ 0, & \text{if } j \notin \mathcal{R}_m. \end{cases} \tag{3}$$

For example, when $k = 2$, the classes 1, 2, 3, and 4 correspond to $(1, 1), (1, -1), (-1, 1)$, and $(-1, -1)$, respectively. If label j is not included in $\mathcal{R}_m$, $q(g_m(\boldsymbol{x}_i), j)$ is 0. If label j corresponds to the first label of $\mathcal{R}_m$, $q(1, j), q(2, j), q(3, j)$, and $q(4, j)$ will output scores $1, 1, -1$, and -1, respectively. We note that the function $q(g_m(\boldsymbol{x}), j)$ is used to generate the $h_m(\boldsymbol{x})$ in the final classifier (1) by gathering the predictions on all label j.

The first term in the objective function aims to minimize the global error between the prediction of $H(\boldsymbol{x})$, that is, $\sum_{m=1}^{M} \beta_m Q_m$, and the multi-label ground truth $\boldsymbol{Y}$. The second term is a two-norm regularization term of the coefficients $\boldsymbol{\beta}$. If the parameter B is larger, then we will obtain smoother coefficients $\boldsymbol{\beta}$. The third term is a hy-

pergraph regularization term. A hypergraph is a generalization of a graph, in which edges, called hyperedges, may connect with any positive number of vertices. The instances with their labels can be represented as one single hypergraph. The vertex is a data point and the hyperedge is a label that connects the instances associated with it. The hypergraph Laplacian $\boldsymbol{L}$ is a similarity measure of instances that calculates the random walk probability starting from one node (instance) until reaching another note on the hypergraph. According to the hypergraph Laplacian, two instances tend to have high similarity if they have large amount of overlapping labels. The intuition behind the hypergraph regularization term is that the prediction on two instances, that is, two rows in $\sum_{m=1}^{M} \beta_m Q_m$, should be similar if they have high similarity according to the hypergraph. This kind of hypergraph representation has been used for multi-label classification in other manners [7, 8].

After some calculus, we have derived the solution of (2) as follows:

$$\boldsymbol{\beta}^* = (\hat{Q}^T \hat{Q} + B\boldsymbol{I} + \frac{C}{2}(P + P^T))^{-1} \hat{Q} \hat{\boldsymbol{Y}} \tag{4}$$

where each column in $\hat{Q} \in \mathbb{R}^{(L \cdot N) \times M}$ is vectorized from Q_m by reshaping Q_m into $\mathbb{R}^{(L \cdot N)}$, $\hat{\boldsymbol{Y}} \in \mathbb{R}^{(L \cdot N)}$ is vectorized from $\boldsymbol{Y}$, $\boldsymbol{I}$ is the $M \times M$ identity matrix, and $P \in \mathbb{R}^{M \times M} = \sum_{j=1}^{L} \rho_j$ where ρ_j is computed as follows:

$$\rho_j = \begin{pmatrix} Q_{1,j}^T \boldsymbol{L} Q_{1,j} & \cdots & Q_{1,j}^T \boldsymbol{L} Q_{M,j} \\ \vdots & \ddots & \vdots \\ Q_{M,j}^T \boldsymbol{L} Q_{1,j} & \cdots & Q_{M,j}^T \boldsymbol{L} Q_{M,j} \end{pmatrix}, \tag{5}$$

where $Q_{m,j}$ is the j-th column vector in Q_m.

3. EXPERIMENTS

We compare the proposed methods with other multi-label classification algorithms on several data sets. In the following sections, we describe the experiment setup including the data sets, the compared algorithms, and the evaluation criteria, and then discuss the experiment results.

3.1. Experiment Setup

We conduct experiments on ten benchmark datasets belonging to different domains. The data sets include *scene*, *enron*, *cal500*, *majorminer* [9], *medical*, *bibtex*, and four versions of *delicious* (from *dlc1* to *dlc4*). More details on these data sets are available at the MULAN library website[1]. We use five popular evaluation metrics for multi-label classification: the Hamming loss, ranking loss, set error, one error, and average precision. Hamming loss calculates the percentage of labels whose relevance is predicted incorrectly. Ranking loss evaluates the average fraction of label pairs, that is, a positive label versus a negative label, that are not correctly ordered. Set error evaluates a multi-label prediction as a whole. It evaluates the percentage of predicted label sets that do not exactly match the true label sets. One error evaluates the number of times the top-ranked label is not relevant. For these four metrics, the smaller the result value is, the better our algorithm performs. Average precision evaluates that, for each relevant label, the percentage of relevant labels among all labels that ranked above it.

We compare the performance of GKL with that of four state-of-the-art multi-label learning algorithms: RA*k*EL, multi-label K-nearest neighbor (MLKNN), instance-based learning by logistic regression (IBLR), and backpropagation for multi-Label learning (BPMLL). These algorithms are implemented in the MULAN package. We use SVM to train the LP classifiers in GKL and RA*k*EL. The parameters k and M in GKL and RA*k*EL are selected using cross-validation. We perform three-fold cross-validation sixty times and calculate the mean and standard deviation of the results.

3.2. Experiment Results

The experimental results are summarized in Table 2. The numbers in parentheses represent the rank of the algorithms among the compared algorithms. We do not report the performance on *cal500* in terms of set error since none of the methods can achieve an error rate better than 1.0. The average rankings of our method on ten datasets using five different metrics are 1.4, 2.0, 1.4, 1.3, and 1.7. We can observe that RA*k*EL performs well in terms of Hamming loss, but not so well on the other metrics. We can observe that MLKNN performs better than GKL only in terms of ranking loss. Generally speaking, our proposed GKL achieves the best performance comparing with the other state-of-the-art methods.

4. CONCLUSION

This paper proposes a basis expansion model for multi-label classification, based on the label powerset method. We have proposed a novel objective function to learn the expansion coefficients and found an analytic solution to learn the coefficients efficiently. The experiment results show that the performance of LP-based ensemble method can be significantly improved by assigning different weights to the classifiers in the ensemble.

5. REFERENCES

[1] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*. 2010.

[2] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence in multi-label classification," in *MLD Workshop*, 2010.

[3] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *ECML*, 2007.

[4] T. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. 2000.

[5] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, 1999.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2001.

[7] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *KDD*, 2008.

[8] F. Wang, X. Wang, B. Shao, T. Li, and M. Ogihara, "Tag integrated multi-label music style classification with hypergraph," in *ISMIR*, 2009.

[9] Michael I. Mandel and Daniel P. W. Ellis, "A web-based game for collecting music metadata," *Journal of New Music Research*, vol. 37, no. 2, pp. 151–165, 2008.

[1] http://mulan.sourceforge.net/datasets.html

Table 2. Experiment Results

	GKL	RA*k*EL	MLKNN	IBLR	BPMLL
Hamming Loss					
scene	0.097±0.002 (3)	0.097±0.002 (2)	0.090±0.002 (1)	0.147±0.003 (4)	0.245±0.035 (5)
enron	0.049±0.000 (1)	0.049±0.000 (2)	0.053±0.000 (3)	0.108±0.001 (5)	0.069±0.001 (4)
cal500	0.164±0.001 (2)	0.163±0.001 (1)	0.277±0.001 (3)	0.363±0.003 (5)	0.308±0.002 (4)
majorminer	0.081±0.000 (1)	0.081±0.000 (2)	0.085±0.000 (3)	0.119±0.001 (5)	0.093±0.001 (4)
medical	0.010±0.000 (1)	0.010±0.000 (2)	0.016±0.000 (3)	0.492±0.017 (5)	0.028±0.000 (4)
bibtex	0.012±0.000 (2)	0.012±0.000 (1)	0.014±0.000 (3)	0.074±0.000 (5)	0.016±0.000 (4)
dlc1	0.039±0.000 (1)	0.039±0.000 (2)	0.040±0.000 (3)	0.054±0.001 (5)	0.051±0.000 (4)
dlc2	0.033±0.000 (1)	0.033±0.000 (2)	0.033±0.000 (3)	0.064±0.001 (5)	0.035±0.000 (4)
dlc3	0.031±0.000 (1)	0.031±0.000 (2)	0.032±0.000 (3)	0.077±0.001 (5)	0.034±0.000 (4)
dlc4	0.025±0.000 (1)	0.025±0.000 (2)	0.026±0.000 (3)	0.158±0.007 (5)	0.028±0.001 (4)
AvgRank	1.40	1.80	2.80	4.90	4.10
Ranking Loss					
scene	0.103±0.003 (2)	0.155±0.004 (3)	0.081±0.002 (1)	0.169±0.006 (4)	0.501±0.015 (5)
enron	0.090±0.001 (1)	0.247±0.003 (4)	0.095±0.001 (2)	0.306±0.004 (5)	0.118±0.001 (3)
cal500	0.153±0.001 (1)	0.225±0.002 (2)	0.248±0.002 (3)	0.383±0.004 (5)	0.296±0.002 (4)
majorminer	0.125±0.001 (1)	0.175±0.002 (3)	0.149±0.001 (2)	0.176±0.002 (4)	0.382±0.032 (5)
medical	0.055±0.004 (2)	0.110±0.007 (3)	0.046±0.002 (1)	0.119±0.006 (4)	0.290±0.015 (5)
bibtex	0.198±0.001 (2)	0.420±0.002 (5)	0.218±0.001 (3)	0.226±0.003 (4)	0.077±0.001 (1)
dlc1	0.178±0.001 (3)	0.647±0.002 (5)	0.157±0.000 (2)	0.140±0.002 (1)	0.202±0.001 (4)
dlc2	0.174±0.001 (3)	0.760±0.008 (5)	0.156±0.001 (1)	0.161±0.002 (2)	0.202±0.001 (4)
dlc3	0.219±0.001 (3)	0.775±0.003 (5)	0.196±0.001 (1)	0.214±0.000 (2)	0.250±0.002 (4)
dlc4	0.214±0.001 (2)	0.755±0.002 (5)	0.191±0.001 (1)	0.260±0.002 (4)	0.245±0.001 (3)
AvgRank	2.00	4.00	1.70	3.50	3.80
Set Error					
scene	0.274±0.005 (2)	0.285±0.006 (3)	0.256±0.006 (1)	0.412±0.010 (4)	0.834±0.018 (5)
enron	0.776±0.005 (1)	0.782±0.005 (2)	0.889±0.006 (3)	0.915±0.005 (4)	0.996±0.005 (5)
majorminer	0.908±0.003 (1)	0.911±0.003 (2)	0.936±0.003 (3)	0.970±0.003 (4)	0.991±0.003 (5)
medical	0.221±0.006 (1)	0.227±0.007 (2)	0.346±0.008 (4)	0.282±0.010 (3)	0.982±0.009 (5)
bibtex	0.727±0.004 (2)	0.727±0.004 (1)	0.878±0.002 (4)	0.922±0.005 (5)	0.834±0.003 (3)
dlc1	0.787±0.002 (3)	0.806±0.001 (4)	0.782±0.001 (1)	0.786±0.001 (2)	0.867±0.001 (5)
dlc2	0.850±0.001 (1)	0.885±0.004 (3)	0.851±0.001 (2)	0.904±0.004 (4)	0.946±0.001 (5)
dlc3	0.882±0.002 (1)	0.904±0.001 (3)	0.901±0.002 (2)	0.952±0.001 (5)	0.949±0.000 (4)
dlc4	0.878±0.002 (1)	0.896±0.001 (2)	0.898±0.001 (4)	0.896±0.003 (3)	0.970±0.000 (5)
AvgRank	1.44	2.44	2.67	3.78	4.67
One Error					
scene	0.254±0.005 (2)	0.257±0.005 (3)	0.235±0.005 (1)	0.384±0.010 (4)	0.812±0.022 (5)
enron	0.287±0.013 (2)	0.278±0.008 (1)	0.319±0.007 (3)	0.536±0.012 (5)	0.494±0.027 (4)
cal500	0.110±0.011 (2)	0.203±0.015 (4)	0.078±0.006 (1)	0.350±0.017 (5)	0.131±0.068 (3)
majorminer	0.369±0.006 (1)	0.374±0.006 (2)	0.452±0.007 (3)	0.723±0.010 (4)	0.835±0.068 (5)
medical	0.146±0.005 (1)	0.148±0.006 (2)	0.264±0.009 (4)	0.199±0.010 (3)	0.976±0.011 (5)
bibtex	0.389±0.001 (1)	0.389±0.001 (2)	0.603±0.004 (4)	0.793±0.008 (5)	0.541±0.007 (3)
dlc1	0.572±0.002 (1)	0.578±0.001 (3)	0.575±0.003 (2)	0.611±0.005 (4)	0.692±0.000 (5)
dlc2	0.597±0.002 (1)	0.710±0.017 (3)	0.616±0.003 (2)	0.764±0.007 (4)	0.769±0.009 (5)
dlc3	0.638±0.003 (1)	0.695±0.003 (3)	0.644±0.004 (2)	0.819±0.003 (5)	0.792±0.000 (4)
dlc4	0.576±0.001 (1)	0.643±0.001 (4)	0.623±0.004 (3)	0.619±0.010 (2)	0.807±0.000 (5)
AvgRank	1.30	2.70	2.50	4.10	4.40
Average precision					
scene	0.841±0.003 (3)	0.842±0.003 (2)	0.860±0.003 (1)	0.754±0.007 (4)	0.421±0.013 (5)
enron	0.666±0.004 (1)	0.646±0.004 (2)	0.620±0.003 (3)	0.399±0.005 (5)	0.505±0.005 (4)
cal500	0.613±0.003 (2)	0.603±0.003 (3)	0.635±0.003 (1)	0.493±0.004 (5)	0.577±0.006 (4)
majorminer	0.599±0.003 (1)	0.593±0.003 (2)	0.541±0.002 (3)	0.427±0.004 (4)	0.221±0.029 (5)
medical	0.861±0.004 (1)	0.855±0.005 (2)	0.794±0.006 (3)	0.792±0.009 (4)	0.122±0.010 (5)
bibtex	0.522±0.002 (1)	0.515±0.002 (2)	0.338±0.003 (4)	0.244±0.005 (5)	0.445±0.003 (3)
dlc1	0.475±0.000 (3)	0.416±0.001 (4)	0.495±0.001 (2)	0.498±0.003 (1)	0.393±0.000 (5)
dlc2	0.421±0.004 (2)	0.285±0.007 (5)	0.431±0.002 (1)	0.364±0.005 (3)	0.308±0.001 (4)
dlc3	0.357±0.001 (2)	0.278±0.001 (3)	0.359±0.003 (1)	0.275±0.001 (4)	0.253±0.001 (5)
dlc4	0.357±0.002 (1)	0.298±0.001 (3)	0.354±0.001 (2)	0.271±0.002 (4)	0.217±0.001 (5)
AvgRank	1.70	2.80	2.10	3.90	4.50