Multi-Label Sampling based on Label Imbalance Rate and Neighborhood Distribution

Zhen Zhang

Data Science and Advanced Computing, University of Reading, Reading Whiteknights House, Reading RG6 6UR

zhen.zhang@student.reading.ac.uk

Abstract. Existing multi-label classification algorithms often assume that label distribution in the training set is balanced, but practical datasets frequently exhibit significant label imbalance. This imbalance affects the learning and generalization performance of the classifiers. To address the problem of label imbalance in multi-label classification, this paper proposes a new synthetic oversampling algorithm, named Multi-Label Synthetic Oversampling based on Label Imbalance Rate and Neighborhood Distribution (MLSIN). This algorithm synthesizes new samples by considering both the imbalance rate of labels and the distribution of samples in their neighborhood, aiming to improve the classifier's performance on minority labels. The rest of this chapter first introduces the evaluation metrics for multi-label classification effectiveness. Then it defines and computes the degree of label imbalance, describes the calculation of imbalance weights, and proposes a sample type correction penalty strategy, detailing the algorithm's process for selecting base and auxiliary samples., and validates the proposed method on public datasets and summarizes the experimental results.

Keywords: Multi-Label classification, Class imbalance, Imbalance Rate, Heuristic sampling

1. Introduction

Multi-label learning, which assigns multiple labels to a single sample, is prevalent in various fields such as text [1, 2], image [3-5], video [6, 7], and bioinformatics [7]. Regarding the class imbalance problems, the accuracy of the classifier on minorities can be severely impaired by class imbalance. Nevertheless, accurate classification of minority classes is essential because minorities have more significant information. Existing methods to handle class imbalance are divided into two groups. One is called algorithm adaptation which makes the classifier skewed to the minority, the other is data preprocessing, including oversampling and undersampling. Compared with algorithm adaptation, sampling methods have a wider scope of application because they do not rely on the choice of the classifier [8-10].

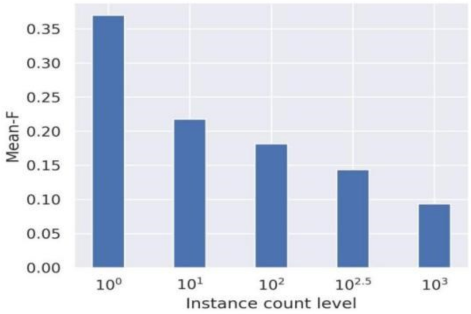

Figure 1. The impact of label imbalance on multi label classification

Figure 1 shows the evaluation results of Macro-F under different label imbalances in a multi-label dataset. This study found that the higher the imbalance level, the lower the value of Macro-F. Using neighborhood information has proven effective in multi-label contexts, particularly in oversampling strategies that mitigate overfitting and noise. This paper introduces a new sampling method that leverages label imbalance rates and neighborhood distribution. Comprehensive comparisons and parameter analyses of this method are presented in subsequent sections.

2. Related Work

2.1. Multi-label classification

Multi-label classification (MLC) assigns multiple labels to each instance within a dataset, Let $\mathrm{D} = \{\mathrm{x_i}, \mathrm{Y_i} | 1 \leq \mathrm{i} \leq \mathrm{n}\}$. For sample $\mathrm{x_i}$, $\mathrm{Y_i}$ represents its associated labels as a binary sequence, where “$\mathrm{Y_{ij}}$=1” indicates presence and “$\mathrm{Y_{ij}}$=0” absence. The objective of MLC is to learn a function mapping $\mathrm{x_i} \rightarrow \mathrm{Y_i}$ for each instance.

MLC algorithms can be categorized by their utilization of label correlation: First-order methods like Binary Relevance (BR) [11] and Multi-label k nearest neighbors (MLkNN) [12] treat labels independently. Second-order approaches, exemplified by Calibrated Label Ranking (CLR) [13], account for pairwise label correlations. High-order methods, such as RAndom k-labELsets (RAkEL) and Classifier Chain (CC) [14-15], consider more complex label dependencies. While first-order methods offer simplicity, higher-order techniques provide a nuanced understanding of label relationships.

2.2. Measure imbalance

There are three different measures are proposed to estimate class imbalance between labels for multi-label data [16-18]. In Eq.(1), the Imbalance ratio per Label (IRLbl) is proposed to be calculated by mining the relationships between the labels.

$$\boldsymbol{IRLbl = c_{max}^1 / c_j^1} \qquad (\boldsymbol{j \in 1, 2, 3 \ldots\ldots q}) \tag{1}$$

As can be seen in Eq.(2), MeanIR is obtained by averaging the IRLbl about all labels. This study defines that a label with IRLbl >MeanIR is the minority label but IRLbl <MeanIR is the majority label.

$$\boldsymbol{MeanIR = \frac{1}{q} \sum_{j=1}^{q} IRLbl_j} \tag{2}$$

The coefficient of variation of IRLbl (CVIR) is a measure that quantifies the variation of IRLbl, reflecting the similarity of the degree of class imbalance across all labels.

$$\boldsymbol{CVIR = \frac{1}{MeanIR} \sqrt{\sum_{j=1}^{q} \frac{(IRLbl_j - MeanIR)^2}{1-q}}} \tag{3}$$

the high CVIR indicates that there are significant differences in the level of imbalance among the labels, while a low value indicates that the labels experience similar levels of imbalance.

3. Proposed method

MLSIN first calculates a matrix $C \in \mathbb{R}^{n*n}$ that stores the proportion of majority class values in the distribution of sample neighborhood labels, this matrix records the proportion of k nearest neighbors of each sample that are not equal to their labels, where k is a hyper-parameters [19]. The calculation method is to first find the k nearest neighbor samples of each sample x_i, for each label l_j, and then calculate the proportion of kNN of sample x_i that are not equal to their labels. The calculation method is as shown in Eq.(4):

$$\boldsymbol{C_{ij}} = \frac{\boldsymbol{1}}{\boldsymbol{k}} \sum\nolimits_{\mathbf{x_i \in kNN_{x_i}}} [\![\mathbf{Y_{ij}} \neq \mathbf{Y_{kj}}]\!] \tag{4}$$

where the notation $[\![\Theta]\!]$ represents the indicator function, which yields 1 if $[\![\Theta]\!]$ is true and 0 otherwise.

Next, MLSIN calculate the sample weight w_i for each sample in the dataset. The larger the sample weight, the easier it is to selected as the base sample during the sampling process. The specific steps of calculation are to calculate the values of the training sample x_i in matrix C. The specific steps of calculation are to sum the values of the training x_i in matrix C to obtain the weight w_i of a single sample, which can represent the difficulty of correctly predicting the minority labels associated with the sample. To simplify the formula, assuming the minority class value in the label is 1, the calculation formula is shown in Eq.(5):

$$\boldsymbol{w_i} = \sum\nolimits_{\boldsymbol{i=1}}^{\boldsymbol{q}} \boldsymbol{C_{ij}} [\![\mathbf{Y_{ij}} = \mathbf{1}]\!] \tag{5}$$

The sample type matrix T divides minority class samples into four categories based on their C_{ij} values, including intra-class (IC), boundary (BD), out of class (OB), and outlier (OT).

IC: $0 < C_{ij} < 0.3$, the intra class samples are located in areas with dense minority class samples.

BD: $0.3 \leq C_{ij} < 0.5$, boundary samples are located at the decision boundary, usually between minority and majority classes.

OB: $0.5 \leq C_{ij} < 0.7$, outliers are closer to the majority class than boundary samples. At the sample outside the boundary. When selected for generating new samples, the label values of the new samples will be directly set to minority classes.

OT: $0.7 \leq C_{ij} < 1$, outliers are surrounded by majority class samples.

In addition, the type of majority class samples is defined as majority (MJ). Let T={IC, BD, OB, OT, MJ} be the matrix that stores the distribution of neighborhood labels, and T_{ij} be the type value of Y_{ij}. After determining the sample type matrix T, the algorithm will make a correction to the sample type. For all OB samples, if one of its k nearest neighbor samples is of type IC or BD, the type of that sample will be corrected to BD.

The MLSIN algorithm presented in this text is a synthetic sampling method developed from the MLSOL algorithm. It takes into account both the label imbalance rate and neighborhood distribution. The pseudocode shown in Table 1 outlines the algorithm's process. Initially, it calculates the label imbalance rate and the imbalance weight as well as a matrix C representing the proportion of majority class values in the neighborhood label distribution. These variables are used to compute the sample neighborhood weight and type matrix T, which will be utilized for selecting base samples and generating new ones. The algorithm then iteratively selects a base and auxiliary sample for the generation of new instances and adds them to the dataset. The process concludes when the number of generated samples reaches a predetermined threshold, ending the oversampling method.

Table 1. Pseudocode of MLSIN

```
Input: Multi-label Dataset: D, Sampling Rate: p, the nearest neighbor k
Output: Sampled Multi-label Dataset: D’

1   Calculate the number of new samples that need to be generated Num=n*p
2   Find k nearest neighbors for each sample
3   Calculate matrix C based on Eq.(4)
4   Define the type matrix T based on Eq.(5)
4   Copy Dataset D’← D
6   While Num>0:
        Select samples in D based on the value of w
        Select reference samples from neighboring samples based on the value of w
        Generate $x_g$ through linear interpolation and use T to generate $Y_g$
        $D’=D’∪(x_g,Y_g)$
7   Return D’
```

4. Experiments and Analysis

This study selected 12 MLDs that are widely used in different fields such as text, image, and bioinformatics, all data sets will be downloaded in MULAN. A simple feature selection approach is applied that retains the top 20% (Corel5k), top 10% (BibTex, medical, Enron) and the top 1% (rcv1subset), method is compared with two heuristic neighbor-based sampling methods (MLSMOTE, MLSOL) and two random sampling methods (MLROS, MLRUS), Especially, this experiment analyzed three label sets assigned methods (Union, Ranking, Intersection) about MLSMOTE [20, 21]. The sampling ratio p is selected from {0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7} for parametric sampling methods (MLROS, MLSOL, and MLRUS). What’s more, four standard multi-label learning algorithms (MLkNN, BR, CC, RAkEL) are used to display the universality of our methods. Regarding the BR, CC, and RAkEL, Decision-Tree is also used as the based classifier. Based on a rule-of-thumb, k=10 as the neighbor is set for MLkNN, k=3 and n=2q for RAkEL, and k=5 as the neighbor for MLSMOTE.

This paper chooses the multi-label evaluation metrics including Macro-averaged AUCROC (area under the receiver operating characteristic curve) to evaluate the performance of previous methods and the method proposed. Then, 5*2 cross-validation will be applied. Finally, to maintain the sample distribution of MLD, multi-label iterative stratification is applied to divide MLD into the training set and the test set.

The summary provided indicates that the MLSIN method outperforms other methods in AUC, showcasing significant improvement (Table 2). It tops average rankings across classifiers, suggesting wide applicability. Neighbor-based methods like MLSOL and MLSMOTE with Ranking also do well, but not as much as MLSIN. Conversely, MLRUS, due to its random undersampling nature, may eliminate crucial samples, making it less effective. The performance of MLSIN with MLkNN as a base classifier is noteworthy, and it is most effective in addressing imbalance within labels, essential for precision and recall in classification. However, in algorithms like RAkEL and CC, which consider label correlation, MLSIN's impact is less pronounced compared to MLkNN.

5. Conclusion

In the experiments, different settings for one parameter are used, while keeping others unchanged at the setting. The application of MLSIN improves the performance of the classifier. that proves to generate some samples will aid classifier learning from the training set. However, as the sampling ratio continues to increase, the performance of the classifier will decrease, because excess samples may distort the original class distributions, and then add to the difficulty of classification. In particular, for the base classifier RAkEL, which transforms the label subsets into classes. When the sampling ratio exceeds 0.1, the evaluation measures decrease significantly, mainly because RAkEL is sensitive to the change in the label set of samples. Otherwise, MLkNN is the best-performing classifier, and under different sampling ratios, the three evaluation measures perform relatively stable without significant decline. MLSIN obtains information from natural neighbors to generate samples, which can provide more guidance for neighbor-based MLkNN.

Research on existing multi-label resampling methods for handling imbalances reveals that current approaches often overlook the variations in label imbalance rates. Therefore, by integrating label imbalance rates with neighborhood distribution for sample selection, the sampling intensity for labels with a higher degree of imbalance is increased, allowing them to have a greater probability of being chosen during sample selection. A penalty strategy is introduced when classifying types of minority label samples, allowing for more cautious type correction and enhancing the information of minority class samples on the decision boundary. These methods effectively reduce the degree of imbalance in multi-label datasets, while also enhancing the classifier's ability to learn from and classify minority class labels and decision boundaries, thereby improving classification performance.

References

[1] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining Multi-label Data [C]// Data Mining and Knowledge Discovery Handbook, 2009: 667-685.

[2] Zhang M. L., Zhou Z. H. A review on multi-label learning algorithms [J]. IEEE transactions on knowledge and data engineering, 2013, 26(8): 1819-1837.

[3] Tsoumakas G., Katakis I. Multi-label classification: An overview [J]. International Journal of Data Warehousing and Mining (IJDWM), 2007, 3(3): 1-13..

[4] Buczak A. L., Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection [J]. IEEE Communications Surveys & Tutorials, 2015, 18(2): 1153-1176.

[5] Zhou F., Huang S., Xing Y. Deep semantic dictionary learning for multi-label image classification [C] Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(4): 3572-3580.

[6] Harding S. M., Benci J. L., Irianto J., et al. Mitotic progression following DNA damage enables pattern recognition within micronuclei [J]. Nature, 2017, 548(7668): 466-470.

[7] Zhu X., Li J., Ren J., et al. Dynamic ensemble learning for multi-label classification [J]. Information Sciences, 2023, 623: 94-111.

[8] B. Wu, E.H. Zhong, A. Horner, Q. Yang, Music emotion recognition by multi-label multi-layer multi-instance multi-view learning [C]// Proceedings of the 22nd ACM International Conference on Multimedia ACM, 2014: 117-126.

[9] Rastogi R., Kumar S. Discriminatory label-specific weights for multi-label learning with missing labels [J]. Neural Processing Letters, 2023, 55(2): 1397-1431.

[10] Chen Ming-Syan, Han Jiawei, Yu P.S. Data mining: An Overview from a Database Perspective [J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6): 866-883.

[11] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification [J]. Pattern Recognition, 2004, 37(9): 1757-1771.

[12] Zhang M. L., Li Y. K., Yang H, et al. Towards class-imbalance aware multi-label learning [J]. IEEE Transactions on Cybernetics, 2020, 52(6): 4459-4471.

[13] Tarekegn A. N., Giacobini M., Michalak K. A review of methods for imbalanced multi-label classification [J]. Pattern Recognition, 2021, 118: 107965.

[14] Mollas I., Chrysopoulou Z., Karlos S., et al. ETHOS: a multi-label hate speech detection dataset [J]. Complex & Intelligent Systems, 2022, 8(6): 4663-4678.

[15] Charte F., Rivera A. J., del Jesus M. J., et al. Addressing imbalance in multilabel classification: Measures and random resampling algorithms [J]. Neurocomputing, 2015, 163: 3-16.

[16] Charte F., Rivera A. J., del Jesus M. J., et al. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation [J]. Knowledge-Based Systems, 2015, 89: 385-397.

[17] Chawla N. V., Bowyer K. W., Hall L. O., et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[18] Pereira R. M., Costa Y. M. G., Silla Jr C. N. MLTL: A multi-label approach for the Tomek Link undersampling algorithm [J]. Neurocomputing, 2020, 383: 95-105.

[19] Charte F., Rivera A., del Jesus M. J., et al. Resampling multilabel datasets by decoupling highly imbalanced labels [C]// Hybrid Artificial Intelligent Systems: 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings 10. Springer International Publishing, 2015: 489-501.

[20] Liu B., Blekas K., Tsoumakas G. Multi-label sampling based on local label imbalance [J]. Pattern Recognition, 2022, 122: 108294.

[21] Zhang K., Mao Z., Cao P., et al. Label correlation guided borderline oversampling for imbalanced multi-label data learning [J]. Knowledge-Based Systems, 2023, 279: 110938.

Appendix:

Table 2. M-AUCROC of comparison approaches in different classifier

M-AUCROC	Default	Union	Ranking	Intersect	MLSOL	MLROS	MLRUS	MLSIN
emotions	0.6762(7)	0.6791(4)	0.6823(2)	0.6403(8)	0.6790(5)	0.6778(6)	0.6819(3)	**0.6826(1)**
flags	0.6209(3)	0.6226(2)	0.6157(7)	0.6030(8)	0.6190(4)	0.6181(5)	0.6115(6)	**0.6234(1)**
scene	0.7495(3)	0.7479(6)	0.7449(8)	0.7444(7)	0.7499(2)	0.7494(4)	0.7487(5)	**0.7503(1)**
yeast	0.5623(4)	0.5607(8)	0.5630(3)	0.5610(7)	**0.5643(1)**	0.5620(5)	0.5619(6)	0.5643(2)
Corel5k	0.5211(5)	0.5214(3)	0.5203(7)	0.5206(6)	**0.5220(1)**	0.5213(4)	0.5201(8)	0.5218(2)
medical	0.7083(4)	0.7075(5)	0.7088(3)	0.7098(2)	0.7043(6)	0.7073(7)	0.6967(8)	**0.7130(1)**
enron	0.5798(3)	0.5797(4)	0.5775(8)	0.5778(7)	0.5790(6)	0.5802(2)	0.5792(5)	**0.5812(1)**
rcv1subset1	0.5925(7)	0.5946(4)	0.5952(2)	0.5929(5)	0.5950(3)	0.5927(6)	0.5876(8)	**0.5968(1)**
rcv1subset2	0.5875(4)	0.5887(3)	0.5870(6)	0.5864(7)	**0.5905(1)**	0.5872(5)	0.5849(8)	0.5893(2)
rcv1subset3	0.5863(7)	0.5891(4)	**0.5924(1)**	0.5879(6)	0.5903(2)	0.5879(5)	0.5859(8)	0.5901(3)
cal500	0.5069(5)	0.5086(3)	**0.5107(1)**	0.5055(8)	0.5081(4)	0.5065(6)	0.5064(7)	0.5095(2)
bibtex	0.5697(3)	0.5712(2)	0.5680(6)	0.5609(8)	0.5691(4)	0.5691(5)	0.5660(7)	**0.5739(1)**
Ave-Ranking	4.58(5)	4.00(3)	4.50(4)	6.58(7)	3.25(2)	5.00(6)	6.58(7)	**1.50(1)**
Wilcoxon	+	+	+	+	+	+	+	
M-AUCROC	Default	Union	Ranking	Intersect	MLSOL	MLROS	MLRUS	MLSIN
emotions	0.7069(3)	0.7070(2)	0.7064(4)	0.6940(8)	0.7033(7)	0.7064(5)	0.7040(6)	**0.7073(1)**

Table 2. (continued).

flags	0.5671(7)	0.5724(3)	0.5682(5)	0.5687(4)	**0.5748(1)**	0.5673(6)	0.5652(8)	0.5733(2)
scene	0.9254(7)	0.9261(3)	0.9258(5)	0.9258(4)	0.9264(2)	0.9256(6)	0.9247(8)	**0.9277(1)**
yeast	0.6704(3)	0.6693(7)	**0.6734(1)**	0.6703(4)	0.6698(5)	0.6674(8)	0.6698(6)	0.6711(2)
Corel5k	0.5315(4)	0.5314(5)	0.5315(3)	0.5317(2)	0.5297(7)	0.5308(6)	0.5295(8)	**0.5407(1)**
medical	0.7924(4)	0.7903(5)	0.7903(6)	0.7903(7)	0.7948(2)	0.7894(8)	0.7929(3)	**0.7956(1)**
enron	0.6328(7)	0.6339(4)	0.6347(2)	0.6344(3)	0.6335(5)	0.6333(6)	0.6304(8)	**0.6350(1)**
rcv1subset1	0.6850(6)	0.6900(3)	0.6887(4)	0.6882(5)	**0.6969(1)**	0.6832(7)	0.6749(8)	0.6920(2)
rcv1subset2	0.6859(6)	0.6874(5)	0.6889(3)	0.6876(4)	**0.7023(1)**	0.6841(7)	0.6801(8)	0.6904(2)
rcv1subset3	0.6914(5)	0.6946(4)	0.6949(3)	0.6887(7)	0.6953(2)	0.6899(6)	0.6843(8)	**0.6968(1)**
cal500	0.5265(4)	0.5270(3)	0.5258(6)	0.5216(8)	0.5271(2)	0.5259(5)	0.5240(7)	**0.5272(1)**
bibtex	0.6733(3)	0.6786(2)	0.6610(7)	0.6541(8)	0.6704(4)	0.6662(6)	0.6703(5)	**0.6801(1)**
Ave-Ranking	4.91(5)	3.83(3)	4.08(4)	5.33(6)	3.25(2)	6.33(7)	6.91(8)	**1.33(1)**
Wilcoxon	+	+	+	+	+	+	+	
M-AUCPR	Default	Union	Ranking	Intersect	MLSOL	MLROS	MLRUS	MLSIN
emotions	0.6783(4)	0.6772(6)	0.6817(2)	0.6415(8)	0.6777(5)	**0.6846(1)**	0.6746(7)	0.6816(3)
flags	0.6208(3)	0.6183(6)	0.6138(7)	0.5824(8)	0.6193(4)	0.6229(2)	0.6192(5)	**0.6252(1)**
scene	0.7485(6)	0.7502(3)	0.7402(7)	0.7391(8)	0.7495(4)	0.7489(5)	**0.7520(1)**	0.7508(2)
yeast	0.5618(3)	0.5567(8)	0.5603(5)	0.5578(7)	0.5604(4)	0.5620(2)	0.5589(6)	**0.5621(1)**
Corel5k	0.5107(5)	0.5106(7)	0.5108(3)	0.5102(8)	0.5110(2)	0.5107(4)	0.5107(6)	**0.5120(1)**
medical	0.7084(4)	0.7086(3)	0.7081(5)	0.7087(2)	0.7061(7)	0.7081(6)	0.7058(8)	**0.7103(1)**
enron	0.5736(5)	0.5738(4)	0.5740(3)	0.5712(8)	0.5726(6)	0.5748(2)	0.5713(7)	**0.5801(1)**
rcv1subset1	0.5916(6)	0.5919(3)	0.5918(4)	0.5917(5)	0.5919(2)	0.5898(7)	0.5892(8)	**0.5925(1)**
rcv1subset2	0.5846(8)	0.5859(4)	0.5870(2)	0.5846(7)	0.5872(1)	0.5856(6)	0.5859(5)	0.5861(3)
rcv1subset3	0.5854(7)	0.5872(3)	0.5857(8)	0.5849(8)	**0.5878(1)**	0.5857(5)	0.5865(4)	0.5859(2)
cal500	0.5051(7)	0.5069(4)	**0.5086(1)**	0.5049(8)	0.5063(3)	0.5053(6)	0.5056(5)	0.5076(2)
bibtex	0.5672(3)	**0.5727(1)**	0.5661(6)	0.5596(8)	0.5669(4)	0.5667(5)	0.5660(7)	0.5675(2)

Table 2. (continued).

Ave-Ranking	5.08(5)	4.00(2)	4.41(4)	6.92(7)	3.58(2)	4.25(3)	5.75(6)	**1.67(1)**
Wilcoxon	+	+	+	+	+	+	+	
M-AUCPR	Default	Union	Ranking	Intersect	MLSOL	MLROS	MLRUS	MLSIN
emotions	0.6721(6)	0.6763(4)	0.6778(2)	0.6404(8)	0.6750(5)	0.6771(3)	0.6681(7)	**0.6798(1)**
flags	0.6105(7)	0.6128(5)	0.6128(4)	0.5855(8)	**0.6218(1)**	0.6170(2)	0.6123(6)	0.6147(3)
scene	0.7399(7)	0.7402(6)	0.7434(5)	0.7467(4)	0.7482(2)	0.7472(3)	0.7397(8)	**0.7502(1)**
yeast	0.5585(6)	0.5588(4)	0.5607(1)	0.5587(5)	0.5590(3)	0.5578(7)	0.5571(8)	0.5607(2)
Corel5k	0.5196(8)	0.5213(2)	0.5203(5)	0.5207(4)	0.5215(1)	0.5202(6)	0.5197(7)	0.5211(3)
medical	0.7084(5)	0.7110(2)	0.7090(4)	0.7061(8)	0.7080(6)	0.7101(3)	0.7064(7)	**0.7118(1)**
enron	0.5771(6)	0.5774(5)	0.5769(7)	0.5775(4)	0.5785(2)	0.5783(3)	0.5767(8)	**0.5811(1)**
rcv1subset1	0.5873(7)	0.5881(5)	0.5898(3)	0.5880(6)	0.5900(2)	0.5883(4)	0.5861(9)	**0.5908(1)**
rcv1subset2	0.5832(7)	0.5841(3)	0.5839(4)	0.5834(5)	**0.5867(1)**	0.5834(6)	0.5830(8)	0.5845(2)
rcv1subset3	0.5828(7)	0.5865(2)	0.5849(4)	0.5837(5)	0.5857(3)	0.5833(6)	0.5809(8)	**0.5866(1)**
cal500	0.5075(6)	0.5082(4)	**0.5095(1)**	0.5053(8)	0.5085(3)	0.5081(5)	0.5058(7)	0.5086(2)
bibtex	0.5681(4)	0.5652(7)	0.5660(5)	0.5592(8)	0.5683(3)	0.5683(2)	0.5653(6)	**0.5692(1)**
Ave-Ranking	6.33(7)	4.08(4)	3.75(3)	6.08(6)	2.67(2)	4.25(5)	7.41(8)	**1.59(1)**
Wilcoxon	+	+	+	+	+	+	+	