Random *k*-Labelsets: An Ensemble Method for Multilabel Classification

Grigorios Tsoumakas and Ioannis Vlahavas

Department of Informatics,
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
{greg,vlahavas}@csd.auth.gr

Abstract. This paper proposes an ensemble method for multilabel classification. The RAndom *k*-labELsets (RAKEL) algorithm constructs each member of the ensemble by considering a small random subset of labels and learning a single-label classifier for the prediction of each element in the powerset of this subset. In this way, the proposed algorithm aims to take into account label correlations using single-label classifiers that are applied on subtasks with manageable number of labels and adequate number of examples per label. Experimental results on common multilabel domains involving protein, document and scene classification show that better performance can be achieved compared to popular multilabel classification approaches.

1 Introduction

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label λ from a set of disjoint labels L, $|L| > 1$. If $|L| = 2$, then the learning task is called *binary* classification (or *filtering* in the case of textual and web data), while if $|L| > 2$, then it is called *multi-class* classification. In *multilabel* classification, the examples are associated with a set of labels $Y \subseteq L$.

Multilabel classification is a challenging research problem that emerges in several modern applications such as music categorization [1], protein function classification [2,3,4,5] and semantic classification of images [6,7]. In the past, multilabel classification has mainly engaged the attention of researchers working on text categorization [8,9,10], as each member of a document collection usually belongs to more than one semantic category.

Multilabel classification methods can be categorized into two different groups [11]: i) *problem transformation* methods, and ii) *algorithm adaptation* methods. The first group of methods are algorithm independent. They transform the multilabel classification task into one or more single-label classification, regression or label ranking [12] tasks. The second group of methods extend specific learning algorithms in order to handle multilabel data directly. There exist multilabel extensions of decision tree [2], support vector machine [13,14], neural network

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 406–417, 2007.
© Springer-Verlag Berlin Heidelberg 2007

[15,5], Bayesian [9], lazy learning [16] and boosting [10] learning algorithms. This paper focuses on the former group of methods.

The most widely-used problem transformation method considers the prediction of each label as an independent binary classification task. It learns one binary classifier $h_\lambda : X \rightarrow \{\neg\lambda, \lambda\}$ for each different label $\lambda \in L$. It transforms the original data set into $|L|$ data sets D_λ that contain all examples of the original data set, labeled as λ if the labels of the original example contained λ and as $\neg\lambda$ otherwise. It is the same solution used in order to deal with a multi-class problem using a binary classifier, commonly referred to as one-against-all or one-versus-rest. Following [12], we will refer to this method as Binary Relevance (BR) learning, a name popular within the Information Retrieval community. BR is criticized for not considering correlations among the labels.

A less common problem transformation method considers each different subset of L as a single label. It so learns one single-label classifier $h : X \rightarrow P(L)$, where $P(L)$ is the powerset of L, containing all possible label subsets. We will refer to this method as Label Powerset (LP) learning. LP has the advantage of taking label correlations into account, but suffers from the large number of label subsets, the majority of which are associated with very few examples.

This paper proposes an approach that constructs an ensemble of LP classifiers. Each LP classifier is trained using a different small random subset of the set of labels. The proposed approach, dubbed RAKEL (RAndom k-labELsets), aims at taking into account label correlations and at the same time avoiding the aforementioned problems of LP. Ensemble combination is accomplished by thresholding the average zero-one decisions of each model per considered label. The paper investigates the issue of selecting appropriate parameters (subset size, number of models, threshold) for RAKEL through an experimental study on three domains concerning protein, image and document classification. Results of performance comparison against the BR and LP methods are in favor of the proposed approach.

A secondary contribution of this paper is a unified presentation of existing evaluation measures for multilabel classification, including their categorization into example-based and label-based measures. The categorization goes further discussing micro and macro averaging operations for any label-based measure.

The remainder of this paper is organized as follows: Section 2 introduces the proposed approach and Section 3 presents the categorization of evaluation measures. Section 4 gives the setup of the experimental study and Section 5 discusses the results. Finally, Section 6 concludes and points to future work.

2 The RAKEL Algorithm

We first define the concept of *k-labelsets* and introduce notation that is subsequently used. Let $L = \{\lambda_i\}$, $i = 1..|L|$ be the set of labels in a multilabel classification domain. A set $Y \subseteq L$ with $k = |Y|$ is called k-labelset. We will use the term L^k to denote the set of all distinct k-labelsets on L. The size of L^k is given by the binomial coefficient: $|L^k| = \binom{|L|}{k}$.

The RAKEL (RAndom k-LabELsets) algorithm iteratively constructs an ensemble of m Label Powerset (LP) classifiers. At each iteration, $i = 1..m$, it randomly selects a k-labelset, Y_i, from L^k without replacement. It then learns an LP classifier $h_i : X \rightarrow P(Y_i)$. The pseudocode of the ensemble production phase is given in Figure 1.

```
Input: Number of models $m$, size of labelset $k$, set of labels $L$, training set $D$
Output: An ensemble of LP classifiers $h_i$ and corresponding $k$-labelsets $Y_i$
$R \leftarrow L^k$;
for $i \leftarrow 1$ to $\min(m, |L^k|)$ do
    $Y_i \leftarrow$ a $k$-labelset randomly selected from $R$;
    train an LP classifier $h_i : X \rightarrow P(Y_i)$ on $D$;
    $R \leftarrow R \setminus \{Y_i\}$;
```

Fig. 1. The ensemble production phase of RAKEL

The number of iterations (m) is a user-specified parameter with acceptable values ranging from 1 to $|L^k|$. The size of the labelsets (k) is another user-specified parameter with meaningful values ranging from 2 to $|L| - 1$. For $k = 1$ and $m = |L|$ we get the binary classifier ensemble of the Binary Relevance (BR) method, while for $k = |L|$ (and consequently $m = 1$) we get the single-label classifier of the LP method. We hypothesize that using labelsets of small size and an adequate number of iterations, RAKEL will manage to model label correlations effectively. The experimental study in Section 5 provides evidence in support of this hypothesis and guidelines on selecting appropriate values for k and m.

For the multilabel classification of a new instance x, each model h_i provides binary decisions $h_i(x, \lambda_j)$ for each label λ_j in the corresponding k-labelset Y_i. Subsequently, RAKEL calculates the average decision for each label λ_j in L and outputs a final positive decision if the average is greater than a user-specified threshold t. An intuitive value for t is 0.5, but RAKEL performs well across a wide range of t values as it shown by the experimental results. The pseudocode of the ensemble production phase is given in Figure 2.

2.1 Computational Complexity

If the complexity of the single-label base classifier is $O(g(|C|, |D|, |A|))$ for a dataset with $|C|$ class values, $|D|$ examples and $|A|$ predictive attributes, then the computational complexity of RAKEL is $O(mg(2^k, |D|, |A|))$. The complexity is linear with respect to the number of models m, as in most ensemble methods, and it further depends on the complexity of the base classifier.

One important thing to note is the high number of class values (2^k) that each LP classifier of RAKEL must learn. This may become an important hindrance of the proposed algorithm, especially if the base classifier has quadratic or greater

```
Input: new instance $x$, ensemble of LP classifiers $h_i$, corresponding set of
       $k$-labelsets $Y_i$, set of labels $L$
Output: multilabel classification vector $Result$
for $j \leftarrow 1$ to $|L|$ do
    $Sum_j \leftarrow 0$;
    $Votes_j \leftarrow 0$;
for $i \leftarrow 1$ to $m$ do
    forall labels $\lambda_j \in Y_i$ do
        $Sum_j \leftarrow Sum_j + h_i(x, \lambda_j)$;
        $Votes_j \leftarrow Votes_j + 1$;
for $j \leftarrow 1$ to $|L|$ do
    $Avg_j \leftarrow Sum_j / Votes_j$;
    if $Avg_j > t$ then
        $Result_j \leftarrow 1$ ;
    else $Result_j \leftarrow 0$ ;
```

Fig. 2. The ensemble combination phase of RAKEL

complexity with respect to the number of class values, as in the case of support vector machine classifiers. In practice however, the actual number of class values is never 2^k, because LP can simply consider the label subsets that appear in the training data. The number of these subsets is typically significantly smaller than 2^k. See for example the number of label subsets for the multilabel datasets considered in the experimental study of this paper (Section 4, Table 1).

3 Evaluation Measures

Multilabel classification requires different evaluation measures than those used in traditional single-label classification. Several measures have been proposed in the past for the evaluation of multilabel classifiers. Some of them are calculated based on the average differences of the actual and the predicted sets of labels over all test examples. Others decompose the evaluation process into separate evaluations for each label, which they subsequently average over all labels. We call the former *example-based* and the latter *label-based* evaluation measures.

3.1 Example-Based

Let D be a multilabel evaluation data set, consisting of $|D|$ multilabel examples (x_i, Y_i), $i = 1..|D|, Y_i \subseteq L$. Let h be a multilabel classifier and $Z_i = h(x_i)$ be the set of labels predicted by h for example x_i.

Schapire and Singer [10] consider the Hamming Loss, which is defined as:

$$HammingLoss(h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \triangle Z_i|}{|L|}$$

where $\triangle$ stands for the symmetric difference of two sets, which is the set-theoretic equivalent of the exclusive disjunction (XOR operation) in Boolean logic.

Classification Accuracy [17] or Subset Accuracy [18] is defined as follows:

$$\mathit{ClassificationAccuracy}(h,D)=\frac{1}{|D|}\sum_{i=1}^{|D|}I(Z_i=Y_i)$$

where I(true)=1 and I(false)=0. This is a very strict evaluation measure as it requires the predicted set of labels to be an exact match of the true set of labels.

The following measures are used in [14]:

$$\mathit{Precision}(h,D)=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i\cap Z_i|}{|Z_i|}\qquad \mathit{Recall}(h,D)=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i\cap Z_i|}{|Y_i|}$$

$$F(h,D)=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{2|Y_i\cap Z_i|}{|Z_i|+|Y_i|}\qquad \mathit{Accuracy}(h,D)=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i\cap Z_i|}{|Y_i\cup Z_i|}$$

3.2 Label-Based

Any known measure for binary evaluation can be used here, such as accuracy, area under the ROC curve, precision and recall. The calculation of these measures for all labels can be achieved using two averaging operations, called *macro-averaging* and *micro-averaging* [8]. These operations are usually considered for averaging precision, recall and their harmonic mean (F-measure) in Information Retrieval tasks.

Consider a binary evaluation measure $M(tp,tn,fp,fn)$ that is calculated based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). Let tp_λ, fp_λ, tn_λ and fn_λ be the number of true positives, false positives, true negatives and false negatives after binary evaluation for a label λ. The macro-averaged and micro-averaged versions of M, are calculated as follows:

$$M_{macro}=\frac{1}{|L|}\sum_{\lambda=1}^{|L|}M\left(tp_\lambda,fp_\lambda,tn_\lambda,fn_\lambda\right)$$

$$M_{micro}=M\left(\sum_{\lambda=1}^{|L|}tp_\lambda,\sum_{\lambda=1}^{|L|}fp_\lambda,\sum_{\lambda=1}^{|L|}tn_\lambda,\sum_{\lambda=1}^{|L|}fn_\lambda\right)$$

Note that micro-averaging has the same result as macro-averaging for some measures, such as accuracy, while it differs for other measures, such as precision, recall and area under the ROC curve. Note also that the average (macro/micro) accuracy and Hamming loss sum up to 1, as Hamming loss is actually the average binary classification error.

4 Experimental Setup

4.1 Datasets

We experiment with 3 datasets from 3 different application domains: bioinformatics, semantic scene analysis and document categorization. The biological dataset *yeast* [13] is concerned with protein function classification. The image dataset *scene* [6] is concerned with semantic indexing of still scenes. The textual dataset *tmc2007* [19] concerns aviation safety reports. These and other multilabel datasets are available for download in Weka's ARFF format at: http://mlkd.csd.auth.gr/multilabel.html

Table 1 shows certain standard statistics of these datasets, such as the number of examples in the train and test sets, the number of numeric and discrete attributes and the number of labels, along with multilabel data statistics, such as the number of distinct label subsets, the label cardinality and the label density [11]. Label cardinality is the average number of labels per example, while label density is the same number divided by $|L|$.

Table 1. Standard and multilabel statistics for the data sets used in the experiments

	Examples		Attributes			Distinct	Label	Label
Dataset	Train	Test	Numeric	Discrete	Labels	Subsets	Cardinality	Density
scene	1211	1196	294	0	6	15	1.074	0.179
tmc2007	21519	7077	0	48981	22	1341	2.158	0.098
yeast	1500	917	103	0	14	198	4.327	0.302

Feature selection was applied on *tmc2007*, in order to reduce the computational cost of training. We used the χ^2 feature ranking method separately for each label in order to obtain a ranking of all features for that label. We then selected the top 500 features based on the their maximum rank over all labels. A similar approach was found to have high performance in previous experimental work on textual datasets [20].

4.2 Multilabel Methods

We compare RAKEL against the BR and LP methods. In all datasets we experiment with 9 different threshold values for RAKEL, ranging from 0.1 to 0.9 with a 0.1 step. We also experiment with a range of subset sizes and number of models, that differ depending on the dataset. We evaluate the performance of methods using a hold-out set. In particular, we use the original train and test set splits that come with the distributions of the datasets. Although we calculate most of the evaluation measures of Section 3, we only present results for Hamming loss and the micro-averaged F-measure, due to limited space. These two metrics are widely-used in the literature and indicative of the performance of multilabel classification methods.

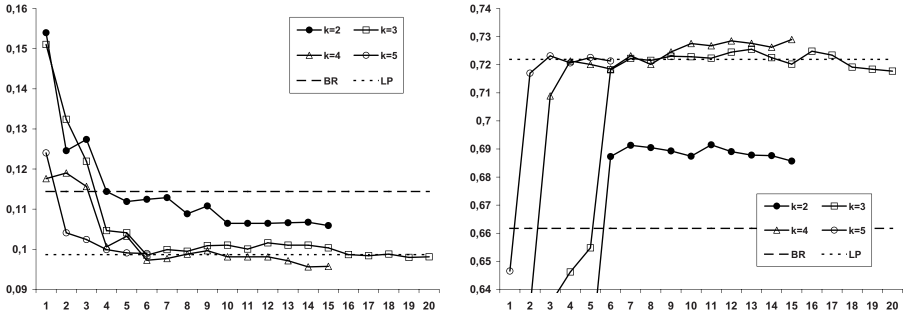
The BR, LP and RAKEL methods can utilize any learning algorithm for classifier training. Evaluating the performance of different algorithms was out of the scope of this paper. We selected the support vector machine (SVM) [21] for the experiments, based on its strong performance in a past study [11]. The SVM was trained with a linear kernel and the complexity constant C equal to 1. The one-against-one strategy is used for dealing with multi-class tasks.

We have implemented a package of Java classes for multilabel classification based on Weka [22]. The package includes implementations of BR, LP, RAKEL and other methods, an evaluation framework that supports the measures presented in Section 3 and code for the calculation of multilabel statistics. It has a command line interface similar to Weka but the full feature set is available only as an API. The package contains source code and a compiled library. Java v1.5 or better and Weka v3.5.5 is required to run the software, which is available for download at http://mlkd.csd.auth.gr/multilabel.html.

5 Results and Discussion

5.1 *Scene* Dataset

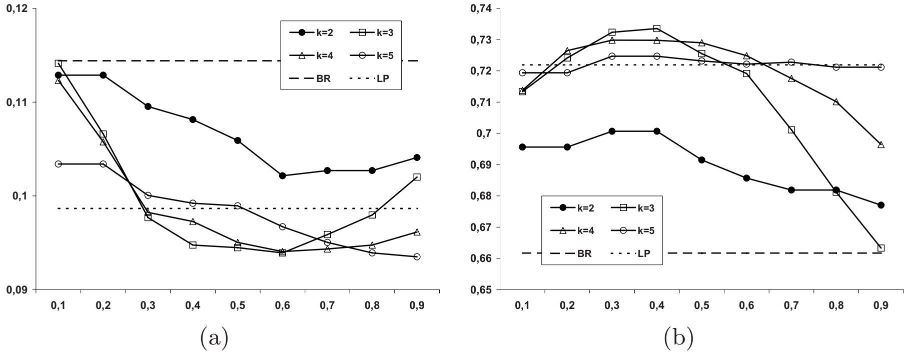
For the *scene* dataset we experiment with all meaningful values for k (2 to 5). We also build incrementally the ensemble with values for m ranging from 1 to $|L^k|$. Figures 3(a) and 3(b) show the Hamming loss and F-measure respectively (y-axis) of BR, LP and RAKEL for $t = 0.5$, with respect to the number of iterations m (x-axis).



(a) (b)

Fig. 3. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for t=0.5

We first notice that LP is better than the more popular BR in this dataset. The small number of labels (6) and label subsets (15) are factors that may contribute to this result. We also notice that for all values of k RAKEL has better performance than BR after the construction of a few models. For $k = 3$, RAKEL achieves the best results, which are better than LP for $m >= 10$. Better



(a) (b)

Fig. 4. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for optimal m

results than LP are also achieved for certain values of m in the cases where $k = 4$ and $k = 5$. These results show that for the default threshold value (0.5) the performance of RAKEL exceeds that of BR and LP for a range of subset sizes and iterations.

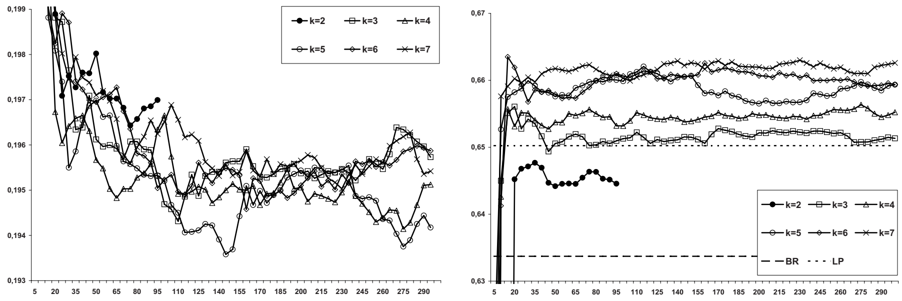
Figures 4(a) and 4(b) show the minimum Hamming loss and maximum F-Measure respectively (y-axis) for RAKEL across all iterations m, with respect to all values of t. The performance of BR and LP is given too. These figures show the best performance that can be achieved by RAKEL irrespectively of the number of models for the different threshold values.

We notice that low Hamming loss can be achieved for a range of t values for $k = 3$ and $k = 4$, with the best results being achieved for $t = 0.6$. The F-measure on the other hand seems to be favored by threshold values around 0.4. RAKEL can achieve higher F-measure than LP for $k = 3$ or $k = 4$ for threshold values ranging from 0.2 to 0.5.

5.2 *Yeast* Dataset

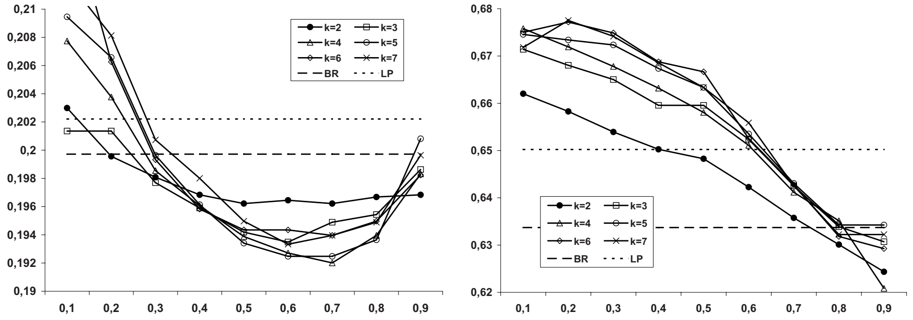
For the *yeast* dataset we experimented with k values from 2 to 7 (half of all labels). The number of iterations (m) were ranging from 1 to $min(|L^k|, 300)$. Similarly to the *scene* dataset, Figures 5(a) and 5(b) show the Hamming loss and F-measure respectively (y-axis) of BR, LP and RAKEL for $t = 0.5$, with respect to the number of iterations m (x-axis). For clarity of presentation, we grouped the values in batches of 5 models and calculated the average.

In Figure 5(a) we notice that the Hamming loss of BR and LP is not displayed, as their values (BR=0.1997 and LP=0.2022) are beyond the focus of the plot. RAKEL achieves better Hamming loss than BR and LP for all values of k after the first 20 models. Hamming loss has a decreasing trend up to around 150 models, while from then on it seems to have a slightly increasing trend. In Figure 5(b) we notice similar results for F-measure, but this time for $k > 3$. As a conclusion we can argue that RAKEL using the default threshold value (0.5) attains better performance than BR and LP for a wide range of k and m values.



(a) (b)

Fig. 5. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for t=0.5



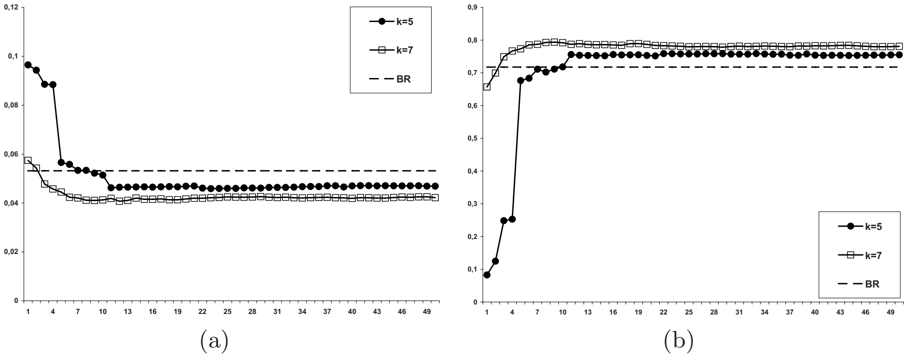
(a) (b)

Fig. 6. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for optimal m

In Figure 6(a), we notice that irrespectively of the subset size, RAKEL has lower Hamming loss than BR and LP for a range of t values (0.4 to 0.8). Regarding the different subset sizes, we notice high performance for $k = 4$ and $k = 5$ consistently for a range of t values (0.5 to 0.8). The lowest Hamming loss is achieved for k=4 and t=0.7. In Figure 6(b), we notice that similarly to the Hamming loss results, RAKEL has higher F-measure than BR and LP for a range of t values (0.1 to 0.6), but this time for $k > 2$. We also notice that compared to Hamming loss, the F-measure is favored by low threshold values. In fact, it seems that F-measure is linearly decreasing with t. The highest F-measure is achieved for $k = 7$ and $t = 0.2$. For $k > 4$ we notice consistently higher performance than for $k \leq 4$ for a range of t values (0.2 to 0.5).

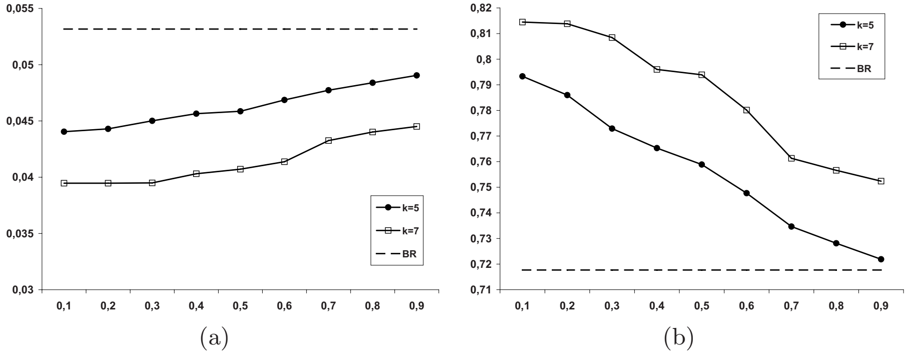
5.3 *Tmc2007* Dataset

For the *tmc2007* dataset we present preliminary experiments for $k = 5$, $k = 7$ and m ranging from 1 to 50. Similarly to the previous datasets, Figures 7(a) and 7(b) show the Hamming loss and F-measure respectively (y-axis) of BR and



(a) (b)

Fig. 7. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for t=0.5



(a) (b)

Fig. 8. Hamming loss (a) and F-measure (b) of BR, LP and RAKEL for optimal m

RAKEL for $t = 0.5$, with respect to the number of iterations m (x-axis). The performance of the full LP classifier was not computed, due to the high memory requirements and computational complexity that comes from the high number of distinct subsets and the quadratic complexity of SVM with respect to the classes.

In Figure 7, we notice that RAKEL achieves better Hamming loss and F-measure than BR for both values of k after the first 10 models. For $k = 7$ the results are better than for $k = 5$. Once more, we conclude that RAKEL using the default $t = 0.5$ value has better performance than BR for a wide range of m values and for both the two k values of the preliminary experiments.

In Figures 8, we notice that irrespectively of the subset size and the threshold value, RAKEL has better Hamming loss and F-measure than BR. Similarly to the *yeast* dataset, we notice that the F-measure is linearly decreasing with t. This behavior of the F-measure with respect to the threshold is consistent in all three datasets, so we can conclude that low t values lead to higher F measure. Similar behavior is noticed for Hamming loss in this dataset, which is linearly increasing with respect to t. This result is different from the previous datasets where large t values seemed to favor Hamming loss.

6 Conclusions and Future Work

This paper has presented a new ensemble method for multilabel classification that is based on random projections of the label space. We train an ensemble of Label Powerset (LP) classifiers in this work and show that higher performance can be achieved than the popular Binary Relevance (BR) method and the LP classifier on the full set of labels. We consider the novel idea of label space projection an important contribution, as it offers a framework for the development of new multilabel ensemble methods, using different multilabel classifiers than LP at the base level and heuristic projection approaches, instead of random.

The latter issue definitely deserves further investigation, as the random nature of RAKEL may be leading to the inclusion of models that affect the ensemble's performance in a negative way. To alleviate this problem, we plan as future work to couple RAKEL with an ensemble selection method [23] in order to select those models that will lead to increased performance.

Acknowledgements

The authors would like to thank Robert Friberg for his valuable contribution in the development of the Java software for multilabel classification. This work is partly funded by the Greek General Secretariat for Research and Technology, project Regional Innovation Pole of Central Macedonia.

References

1. Li, T., Ogihara, M.: Detecting emotion in music. In: Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA, pp. 239–240 (2003)
2. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
3. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 448–456. Springer, Heidelberg (2005)
4. Roth, V., Fischer, B.: Improved functional prediction of proteins by learning kernel combinations in multilabel settings. In: Proceeding of 2006 Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology (PMSB 2006), Tuusula, Finland (2006)
5. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18, 1338–1351 (2006)
6. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37, 1757–1771 (2004)

7. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York City, NY, USA, pp. 1719–1726. IEEE Computer Society Press, Los Alamitos (2006)
8. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval 1, 78–88 (1999)
9. McCallum, A.: Multi-label text classification with a mixture model trained by em. In: Proceedings of the AAAI' 99 Workshop on Text Learning (1999)
10. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning 39, 135–168 (2000)
11. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3, 1–13 (2007)
12. Brinker, K., Furnkranz, J., Hullermeier, E.: A unified model for multilabel classification and ranking. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI '06), Riva del Garda, Italy, pp. 489–493 (2006)
13. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Advances in Neural Information Processing Systems 14 (2002)
14. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
15. Crammer, K., Singer, Y.: A family of additive online algorithms for category ranking. Journal of Machine Learning Research 3, 1025–1058 (2003)
16. Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classification. In: Proceedings of the 1st IEEE International Conference on Granular Computing, pp. 718–721. IEEE Computer Society Press, Los Alamitos (2005)
17. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in Information Retrieval, pp. 274–281. ACM Press, New York (2005)
18. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 3005 ACM Conference on Information and Knowledge Management (CIKM '05), Bremen, Germany, pp. 195–200. ACM Press, New York (2005)
19. Srivastava, A., Zane-Ulman, B.: Discovering recurring anomalies in text reports regarding complex space systems. In: 2005 IEEE Aerospace Conference, IEEE Computer Society Press, Los Alamitos (2005)
20. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, pp. 659–661 (2002)
21. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)
23. Tsoumakas, G., Angelis, L., Vlahavas, I.: Selective fusion of heterogeneous classifiers. Intelligent Data Analysis 9, 511–525 (2005)