Graphical Feature Selection for Multilabel Classification Tasks

Gerardo Lastra, Oscar Luaces, Jose R. Quevedo, and Antonio Bahamonde

Artificial Intelligence Center. University of Oviedo at Gijón, Asturias, Spain
www.aic.uniovi.es

Abstract. Multilabel was introduced as an extension of multi-class classification to cope with complex learning tasks in different application fields as text categorization, video o music tagging or bio-medical labeling of gene functions or diseases. The aim is to predict a set of classes (called labels in this context) instead of a single one. In this paper we deal with the problem of feature selection in multilabel classification. We use a graphical model to represent the relationships among labels and features. The topology of the graph can be characterized in terms of relevance in the sense used in feature selection tasks. In this framework, we compare two strategies implemented with different multilabel learners. The strategy that considers simultaneously the set of all labels outperforms the method that considers each label separately.

1 Introduction

Many complex classification tasks share that each instance can be assigned with more than one class or label instead of a single one. These tasks are called *multilabel* to emphasize the multiplicity of labels. This is the case of text categorization where items have to be tagged for future retrieval; frequently, news or other kind of documents should be annotated with more than one label according to different points of view. Other application fields include semantic annotation of images and video, functional genomics, music categorization into emotions and directed marketing. Tsoumakas *et al.* in [11, 12] have made a detailed presentation of multilabel classification and their applications.

From a computational perspective, the aim of multilabel classification is to obtain *simultaneously* a collection of binary classifications; the positive classes are referred to as *labels*, the so-called *relevant* labels of the instances. A number of strategies to tackle multilabel classification tasks have been published. Basically, they can be divided in two groups [11, 12].

Strategies in the first group try to *transform* the learning tasks into a set of single-label (binary or multiclass) classification tasks. *Binary Relevance (BR)* is the most simple, but very effective, transformation strategy. Each label is classified as relevant or irrelevant without any relation with the other labels.

On the other hand, *proper multilabel* strategies try to take advantage of correlation or interdependence between labels. The presence or absence of a label in the set assigned to an instance may be conditioned not only by the feature values of the instance, but also by the values of the remaining labels.

Feature selection is an important issue in machine learning in general. In multilabel, accordingly to [12], most feature selection tasks have been addressed by extending the techniques available for single-label classification using the bridge provided by multilabel transformations. Thus, when the BR strategy is used, it is straightforward to employ a feature subset selection on each binary classification task, and then combining somehow the results [16]. In [10], the authors present a feature selection strategy based on the transformation called *label powerset*.

Kong *et al.* [6] presented a multilabel selection method in the special case where instances are graphs so that the selection has to find subgraphs. Finally, in [18] feature selection is performed using a combination of principal component analysis with a genetic algorithm.

We propose to extend a well known filter devised for multiclass classification tasks, *FCBF (Fast Correlation-Based Filter)* [17]. This filter computes the relation between features and the target class using a non-linear correlation measure, the *Symmetrical Uncertainty (SU)*. For this reason we have to assume that all feature values are discrete.

The core idea of the method proposed here is to represent the relationships between the variables involved (features and labels) in a multilabel classification task by means of a graph computed in two stages. First, we build the matrix of SU scores for all pairs of variables. Then, we compute the spanning tree of the complete undirected graph where the nodes are the variables, and the edges are weighted by SU scores.

Clearly, the inspiration underlying this apporach is rooted in seminal papers such as [2]; in this case a spanning tree was used to factorize a probability distribution from a Bayesian point of view. In [14], this kind of graphs were used for multi-dimensional classifications. Our approach, however, is not based on Bayesian networks. We prove that the spanning tree links can be characterized in terms of relevance (in the sense of feature selection) and redundance of features and labels.

The paper is organized as follows. In the next section we present the formal framework for multilabel classification including the definition of scores and loss functions devised to measure the performance of classifiers. Then we present the graphical model that relates features and labels. The fourth section is devoted to report and discuss a number of experiments carried out to evaluate the proposals of the paper. The last section summarizes some conclusions about the work presented here.

2 Formal Framework for Multilabel Classification

A formal presentation of a multilabel classification learning task can be given as follows. Let L be a finite and non-empty set of labels $\{l_1, \ldots, l_{|L|}\}$, and let $\mathcal{X}$ be an input space. A multilabel classification task can be represented by a dataset

$$D = \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_{|D|}, Y_{|D|})\} \tag{1}$$

of pairs of instances $\boldsymbol{x}_i \in \mathcal{X}$ and subsets of labels $Y_i \subset L$. The goal is to induce from D a hypothesis defined as follows.

Definition 1. *A* multilabel *hypothesis is a function h from the input space to the set of subsets (power set) of labels $\mathscr{P}(L)$; in symbols,*

$$h : \mathcal{X} \longrightarrow \mathscr{P}(L) = \{0,1\}^L. \tag{2}$$

Given a multilabel classification task D, there is a straightforward approach to induce a multilabel hypothesis from a dataset D, the so-called *Binary Relevance* strategy. For each $l \in L$, this approach induces a binary hypothesis

$$h_l : \mathcal{X} \longrightarrow \{0,1\}, \tag{3}$$

and then its predictions are defined as

$$h(\boldsymbol{x}) = \{l : h_l(\boldsymbol{x}) = 1\}.$$

In any case, the prediction $h(\boldsymbol{x})$ of a multilabel hypothesis can be understood as the set of *relevant* labels retrieved for a *query* $\boldsymbol{x}$. Thus, multilabel classification can be seen as a kind of Information Retrieval task for each instance; in this case the labels play the role of documents. Performance in Information Retrieval is compared using different measures in order to consider different perspectives. The most frequently used measures are *Recall* (proportion of all relevant documents (labels) that are found by a search) and *Precision* (proportion of retrieved documents (labels) that are relevant). The harmonic average of the two amounts is used to capture the goodness of a hypothesis in a single measure. In the weighted case, the measure is called F_β. The idea is to measure a tradeoff between *Recall* and *Precision*.

For further reference, let us recall the formal definitions of these measures. Thus, for a prediction of a multilabel hypothesis $h(\boldsymbol{x})$, and a subset of *truly relevant* labels $Y \subset L$, we can compute the following contingency matrix,

$$\begin{array}{r|cc} & Y & L \setminus Y \\ \hline h(\boldsymbol{x}) & a & b \\ L \setminus h(\boldsymbol{x}) & c & d \end{array} \tag{4}$$

in which each entry (a, b, c, d) is the number of labels of the intersection of the corresponding sets of the row and column. Notice for instance, that a is the number of relevant labels in Y predicted by h for $\boldsymbol{x}$.

According to the matrix, (Eq. 4), we thus have the following definitions.

Definition 2. *The* Recall *in a* query *(i.e. an instance $\boldsymbol{x}$) is defined as the proportion of relevant labels Y included in $h(\boldsymbol{x})$:*

$$R(h(\boldsymbol{x}), Y) = \frac{a}{a+c} = \frac{|h(\boldsymbol{x}) \cap Y|}{|Y|}. \tag{5}$$

Definition 3. *The* Precision *is defined as the proportion of retrieved labels in $h(\boldsymbol{x})$ that are relevant Y:*

$$P(h(\boldsymbol{x}), Y) = \frac{a}{a+b} = \frac{|h(\boldsymbol{x}) \cap Y|}{|h(\boldsymbol{x})|}. \tag{6}$$

Finally, the tradeoff is formalized by

Definition 4. *The F_β is defined, in general, by*

$$F_\beta(h(\boldsymbol{x}), Y) = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)a}{(1+\beta^2)a + b + \beta^2 c}. \tag{7}$$

The most frequently used F-measure is F_1. For ease of reference, let us state the formula of F_1 for a multilabel classifier h and a pair $(\boldsymbol{x}, Y)$:

$$F_1(h(\boldsymbol{x}), Y) = \frac{2|h(\boldsymbol{x}) \cap Y|}{|Y| + |h(\boldsymbol{x})|}. \tag{8}$$

These measures are not *proper loss* functions in the sense that high scores mean good performance. Thus, for instance, to obtain a loss function from F_β scores it is necessary to compute the complementary $(1 - F_\beta)$. In any case, when we try to optimize F_β, we mean to improve the performance according to this measure; that is, to maximize F_β or to minimize $1 - F_\beta$.

So far, we have presented functions able to evaluate the performance of a hypothesis on one instance $\boldsymbol{x}$. To extend these functions to a test set, we shall use the so-called *microaverage* extension of these score functions. For further reference, let

$$D' = \big\{(\boldsymbol{x}_i, Y_i) : i = 1, \dots, |D'|\big\}$$

be a multilabel dataset used for testing. Moreover, for ease of reading, we have expressed the microaverage of F_1 as percentages in the experiments reported at the end of the paper. Thus, the formulas for a hypothesis h are the following:

$$F_1(h, D') = \frac{100}{|D'|} \sum_{i=1}^{|D'|} \frac{2|h(\boldsymbol{x}'_i) \cap Y'_i|}{|Y'_i| + |h(\boldsymbol{x}'_i)|}. \tag{9}$$

Additionally, to avoid cumbersome notation, we have overloaded the meaning of the symbol F_1 for the microaverage extensions.

3 Modeling the Relationships of Labels and Attributes

In this section we introduce a graphical representation of the relevance relationship between labels and the attributes or features used to describe input instances; in this paper we shall use attribute and feature as synonyms. To make a formal presentation, throughout this section, let D be a multilabel classification task (Eq. 1) with instances $\boldsymbol{x} \in \mathcal{X}$, and labels in L.

If $\mathcal{X}$ can be represented by vectors of dimension $|\mathcal{X}|$, D can be seeing as a matrix $\mathbb{M}$ given by

$$\mathbb{M} = [\mathbb{X} \quad \mathbb{L}] \tag{10}$$

where $\mathbb{X}$ and $\mathbb{L}$ are matrices with $|D|$ rows (one for each training example), and $|\mathcal{X}|$ and $|L|$ columns respectively. The first matrix, $\mathbb{X}$, collects the input instance descriptions; while columns represent attributes (or features). As we said in the Introduction, we assume that the entries of matrix $\mathbb{X}$ are discrete values. On the other hand, the matrix $\mathbb{L}$ has Boolean values: $\mathbb{L}[i, j] = 1$ if and only if the i-th example of D has the label $l_j \in L$.

In this paper we extend the filter FCBF (Fast Correlation-Based Filter) introduced in [17] to multilabel classification tasks. Since this filter was devised for dealing with multiclass classification tasks, we need to involve the whole set of labels. From a formal point of view, FCBF deals with a matrix $\mathbb{X}$ and just one column of matrix $\mathbb{L}$. Thus, we are going to review the selection method of FCBF using the matrix $\mathbb{M}$ that collects all labels at the same time.

Given a single class and a collection of predictive attributes or features, the filter FCBF proceeds in two steps: relevance and redundancy analysis, in this order. For both steps the filter uses the so-called *symmetrical uncertainty*, a normalized version of the mutual information. Let us now rewrite the formulation of this measure applied to the columns of the matrix $\mathbb{M}$. It is based on a nonlinear correlation, the *entropy*, a measure of the uncertainty that is defined for a column m_j of the matrix as follows

$$H(m_j) = -\sum_{i=1}^{|D|} \Pr(m_j^i) \log_2(\Pr(m_j^i)). \tag{11}$$

Additionally, the entropy of a column m_j after observing the values of another column m_k is defined as

$$H(m_j|m_k) = -\sum_{r=1}^{|D|} \Pr(m_k^r) \sum_{s=1}^{|D|} \Pr(m_j^s|m_k^r) \log_2(\Pr(m_j^s|m_k^r)), \tag{12}$$

where $\Pr(m_k^r)$ denotes the prior probabilities for all possible values of column m_k; and $\Pr(m_j^s|m_k^r)$ denotes the posterior probabilities of m_j.

In a similar way, it is possible to define $H(m_j, m_k)$ using in (Eq. 11) the joint probability distribution.

The *information gain* (IG) of m_j given m_k, also known as the Kullback-Leibler divergence, is defined as the difference between the prior and posterior entropy to the observed values of m_j. In symbols,

$$IG(m_j|m_k) = H(m_j) - H(m_j|m_k) = H(m_j) + H(m_k) - H(m_j, m_k). \tag{13}$$

The information gain is a symmetrical measure. To ensure a range of values in $[0, 1]$, FCBF uses a normalized version, the *symmetrical uncertainty (SU)* defined as follows

$$SU(m_j, m_k) = 2\left[\frac{IG(m_j|m_k)}{H(m_j) + H(m_k)}\right] = 2\left[1 - \frac{H(m_j, m_k)}{H(m_j) + H(m_k)}\right]. \tag{14}$$

To return the list of relevant variables for a single variable, FCBF first removes those attributes whose SU is lower or equal than a given threshold. Then, FCBF orders the remaining attributes in descending order of their SU with the class, and applies an iterative process to eliminate redundancy. This process is based on approximate Markov blankets; in the multilabel context, this concept can be formulated as follows.

Definition 5. *(Approximate Markov Blanket) Given three different columns m, m_i and m_j in $\mathbb{M}$, m_j forms an approximate Markov blanket for m_i if and only if*

$$SU(m_j, m) \geq SU(m_i, m) \wedge SU(m_i, m_j) \geq SU(m_i, m). \tag{15}$$

Notice that the aim of this definition in [17] is to mark the feature m_i as *redundant* with m_j when the goal is to predict the values of m. To avoid tie situations that would require random choices, we exclude the equalities of (Eq. 15). In other words, we assume that all SU values are different. Hence, for further reference, we make the following definitions.

Definition 6. *(Redundancy) The column m_i is redundant with m_j for predicting m if and only if*

$$SU(m_j, m) > SU(m_i, m) \wedge SU(m_i, m_j) > SU(m_i, m). \tag{16}$$

Once we have reviewed the core of FCBF, to extend it to multilabel classification tasks, we start computing the Symmetrical Uncertainty (SU) for all pairs of columns of matrix $\mathbb{M}$.

Definition 7. *(Symmetrical Uncertainty Matrix) Given a multilabel classification task D, with labels in L, the $\mathbb{SU}$ matrix is formed by the symmetrical uncertainty of all columns of $\mathbb{M}$ (Eq. 10),*

$$\mathbb{SU} = [SU(m, m') : m, m' \in \mathit{columns}].$$

This matrix represents a weighted undirected graph in which the set of vertices is the set of *columns*; that is, the set of attributes of $\mathcal{X}$ and labels in L. To *order* this graph, we now compute the spanning tree with maximum SU values.

Definition 8. *(Maximum Spanning Tree) MST is the maximum spanning tree of the $\mathbb{SU}$ matrix.*

Figure 1 shows one MST for an hypothetical dataset. Our aim now is to explain the meaning of this tree in terms of relevance of the attributes and labels. The general idea is to compare the topology of the MST with the results of applying the filter FCBF considering each column as the category and the others as predictors.

To compute the MST we may use, for instance, Kruskal's algorithm [7]. The edges are ordered from the highest to the lowest SU values. Then, starting from an empty MST, the algorithm iteratively adds one edge to the MST at each step, provided that it does not form a cycle in the tree. We shall see that this basic building step can be interpreted in terms of redundancy. First, however, we state some propositions to establish the ideas presented here.

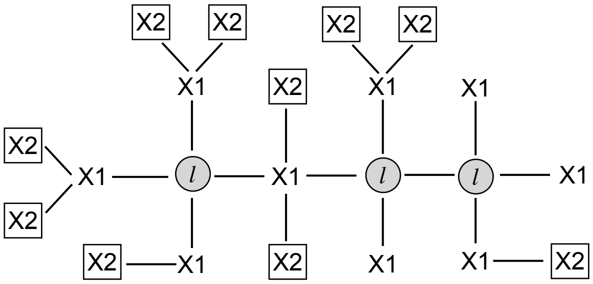

Fig. 1. Maximum Spanning Tree of an hypothetical multilabel task, see definition 8. Nodes marked with l stand for *labels*. Nodes with an Xi represent attributes: $X1$ are the attributes at distance 1 from labels, and $X2$ are attributes at distance 2

Proposition 1. *If m and m' are two adjacent nodes in the MST defined in (Def. 8), then m' is relevant for m using the filter FCBF.*

Proof. If we assume that there is another label m'' that removes m' from the list of relevant nodes for m, then m' will be redundant with m'' for m. In symbols,

$$SU(m'', m) > SU(m', m) \wedge SU(m'', m') > SU(m', m). \tag{17}$$

In this case, however, the link between m and m' could not be included in the MST. Therefore, there cannot exist such a node m'', and so m' is relevant for m according to the filter FCBF.

Proposition 2. *If m is adjacent to m', and this label is adjacent to m'' ($m'' \neq m$) in the MST, then m'' is redundant with m' for m.*

Proof. Consider the triangle of vertices m, m', m'' in the complete graph represented by the SU matrix (Def. 7). Since the edge $m - m''$ is not included in the MST, we have that

$$SU(m', m) > SU(m'', m) \wedge SU(m'', m') > SU(m'', m).$$

Thus, m'' is redundant with m' for m according to (Def. 6).

The conclusion is that, given a column m in $\mathbb{M}$, its adjacent labels in the MST are relevant for it. Moreover, if m, m', m'' is a path in the MST, m'' is relevant for m', which is in turn relevant for m. Hence, m'' is redundant with m' for m. However, sometimes redundant information helps classifiers to increase their performance, thus we may heuristically select some redundant items in order to achieve better performance. In our case, this heuristic is implemented fixing in the graph the distance to targets from predictors; this distance will be called *level of proximity*.

3.1 Multilabel Ranker

Taking into account the previous results, in order to select a subset of features to predict the labels, given a dataset D, we can fix a level of proximity k, and then our proposal is the following:

- Compute the Symmetrical Uncertainty Matrix $\mathbb{SU}$ of attributes and labels.
- Compute the Maximum Spanning Tree (MST).
- Select the attribute nodes whose distance to any label is smaller than or equal to k, see Figure 1.

Notice that this method produces a ranking of features; thus, we call it *Multilabel Feature Ranker (MLfR)*. In fact, increasing the level of proximity (k) we obtain a sequel of features in decreasing order of usefulness to predict a set of labels. However, the features are returned in *chunks* (in *quanta*) instead of one by one. In general, there are more than one feature at a given distance from the set of labels.

4 Experimental Results

In this section we report a number of experiments conducted to test the multilabel feature ranker MLfR in two and complementary dimensions: the classification performance and the quality of the ranking.

First we check the capacity of MLfR to optimize a performance score like F_1 (Eq. 9). For this purpose, with each training data we built the MST (Def. 8) and then we selected the best k (Section 3.1) using an internal (in the training set) 2-fold cross validation repeated 5 times [4]. The range of k values included $\{1, 2, \ldots, 20\}$ and k_{50}, k_{75} and k_{100}, where k_t is the smallest k value that ensures that the $t\%$ of all features are selected. Since the aim was to compare strategies for feature selection, the number of discretization bins were constant in all cases.

To obtain a multilabel learner with this selection scheme we used two state of the art multilabel *base* learners. The first one is IBLR-ML [1]. We used the implementation provided by the authors in the library *Mulan* [13, 12], which is built on the top of *Weka* [15]. We wrote an interface with MatLab. The second base learner used was the *Ensemble of Classifier Chains* (ECC) [9] in the version described in [3]; for this reason we called it *ECC**. The implementation was made in MatLab using the *BR* built with *LibLinear* [8, 5] with the default parameters: a logistic regression learner with regularization parameter $C = 1$.

On the other hand, to test the quality of the ranking of features produced by MLfR, we computed the number of features selected by the first chunk ($k = 1$) and the F_1 achieved with those features. To summarize in one number the quality of the first chunk of the ranking, we computed the *contribution* of each feature to the F_1 as follows,

$$contribution = \frac{F_1(K = 1)}{\#features(K = 1)}. \tag{18}$$

To compare the results obtained by MLfR, we computed the F_1 scores achieved by the base learners without performing any selection at all. Additionally, to compare the quality of the rankings we wanted to contrast the multilabel ranking with a purely binary ranker; that is, a ranker that considers labels one by one. To make a fair comparison we implemented a *binary relevance* version of MLfR as follows. For each label $l \in L$, we computed MLfR considering only that label. The set of features at distance k from l obtained in this way, $MLfR_l(k)$, were joined together for all labels to get a chunk of level k in the so-called *Binary Relevance Feature Ranker (BRfR).*

$$BRfR(k) = \bigcup_{l \in L} MLfR_l(k) \tag{19}$$

The comparison presented here was carried out using 8 datasets previously used in experiments reported in other papers about multilabel classification. Table 1 shows a summarized description of these datasets including references to their sources. Attributes with continuous values have been discretized in 10 bins using a *same frequency* procedure. The comparison was performed using a simple hold-out method. We used the split of datasets in training and testing sets provided by the sources of the data, when available. The size of the splits are also shown in Table 1. Other details about the datasets, included preprocessing, can be found following the references provided in the table.

Table 1. The datasets used in the experiments, associated statistics, and references to the sources of the data

	#Instances						
	train	test	total	#fea.	$\vert L \vert$	Cardinality	Source
enron	1123	579	1702	1001	53	3.38	[13]
genbase	463	199	662	1185	27	1.25	[13]
medical	333	645	978	1449	45	1.25	[13]
slashdot	2500	1282	3782	1079	22	1.18	[9]
emotions	391	202	593	72	6	1.87	[13]
reuters	5000	2119	7119	243	7	1.24	[1, 19, 20]
scene	1211	1196	2407	294	6	1.07	[13]
yeast	1500	917	2417	103	14	4.24	[13]

The scores achieved in F_1 by the classifiers compared are shown in Table 2. To make statistical comparisons we considered together the scores obtained with all base learners, since the objective was to compare different selection strategies and not base learner scores.

Thus, we observe that although the scores obtained by selectors are higher in average than those achieved without any selection, the differences are not significant using a paired, two-sided, Wilcoxon signed rank test. Also, there are no significant differences between the scores obtained with the two selection approaches. On the other hand, both selectors reduce considerably the number

Table 2. Number of features and F_1 scores achieved in test data when the aim in grid search (for selectors) was to optimize F_1

		Base		MLfR		BRfR	
	dataset	#fea.	F_1	#fea.	F_1	#fea.	F_1
IBLR-ML	enron	1001	41.78	26	49.79	49	48.25
	genbase	1186	99.00	46	99.15	40	98.31
	medical	1449	47.33	88	68.24	85	70.41
	slashdot	1079	15.80	46	26.04	48	26.27
	emotions	72	64.41	72	64.41	72	64.41
	reuters	243	74.38	189	74.78	151	76.78
	scene	294	70.29	294	70.29	294	70.29
	yeast	103	61.72	103	61.72	103	61.72
ECC*	enron	1001	53.49	920	52.82	1001	53.49
	genbase	1186	99.41	68	98.31	65	98.31
	medical	1449	61.62	88	69.81	85	69.14
	slashdot	1079	37.54	1079	37.54	1079	37.54
	emotions	72	60.59	38	60.10	37	59.37
	reuters	243	76.87	207	78.02	204	78.07
	scene	294	56.29	294	56.29	294	56.29
	yeast	103	59.51	30	58.02	38	59.03

Table 3. F_1 and number of features selected in the first chunk ($k = 1$) for MLfR and BRfR. The scores achieved when no selection is performed is included for comparison. Additionally, for each ranker we computed the contribution of each feature to the F_1 score (see Eq. 18)

		Base		MLfR			BRfR		
	dataset	#fea.	F_1	#fea.	F_1	contri.	#fea.	F_1	contri.
IBLR-ML	enron	1001	41.78	26	49.79	1.92	49	48.25	0.98
	genbase	1186	99.00	46	99.41	2.16	40	99.15	2.48
	medical	1449	47.33	88	68.24	0.78	85	70.41	0.83
	slashdot	1079	15.80	46	26.04	0.57	48	26.27	0.55
	emotions	72	64.41	1	33.04	33.04	5	52.44	10.49
	reuters	243	74.38	44	74.13	1.68	45	73.93	1.64
	scene	294	70.29	3	24.58	8.19	6	38.35	6.39
	yeast	103	61.72	2	51.58	25.79	9	54.65	6.07
ECC*	enron	1001	53.49	26	42.58	1.64	49	51.77	1.06
	genbase	1186	99.41	46	97.81	2.13	40	98.31	2.46
	medical	1449	61.62	88	69.81	0.79	85	69.14	0.81
	slashdot	1079	37.54	46	25.72	0.56	48	25.40	0.53
	emotions	72	60.59	1	36.42	36.42	5	49.97	9.99
	reuters	243	76.87	44	74.95	1.70	45	74.91	1.66
	scene	294	56.29	3	13.85	4.62	6	25.98	4.33
	yeast	103	59.51	2	54.33	27.16	9	56.57	6.29

of features used for classification. The differences are not significant between selectors.

It could be expected important reductions of the number of features and the error rates for high dimensional problems such as Enron or Slashdot. But the scores shown in Table 2 for *ECC** report null or insignificant reductions; the reason can be found in the poor quality of the classifiers, both datasets provide the smallest F_1 scores for this learner.

The statistically significant differences appear when we check the quality of the first chunk. Thus, the contribution of each of the features obtained with MLfR for $k = 1$ is significantly higher than that of the features returned by BRfR in the same conditions. In this sense we conclude that the ranking learned from a multilabel point of view is better than the ranking obtained considering each label separately.

5 Conclusions

We have presented an algorithm to learn a ranking of the features involved in a multilabel classification task. It is an extension of the FCBF (Fast Correlation-Based Filter) [17], and it uses a graphical representation of features and labels. The method so obtained, MLfR (multilabel feature ranker), was compared with a version that considers each label separately, in the same way as BR (Binary Relevance) learns a multilabel classifier. We experimentally tested that the multilabel version achieves significantly better results than the BR release when testing the quality of the rankings.

Moreover, the graph built by MLfR provides a valuable representation of the correlation and interdependence between labels and features. We proved formally that the topology of the graph can be read in terms of relevancy and redundance of the features and labels.

6 Acknowledgements

The research reported here is supported in part under grant TIN2008-06247 from the MICINN (Ministerio de Ciencia e Innovación, Spain). We would also like to acknowledge all those people who generously shared the datasets and software used in this paper.

References

1. Cheng, W., Hüllermeier, E.: Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. Machine Learning 76(2), 211–225 (2009)
2. Chow, C., Liu, C.: Approximating Discrete Probability Distributions with Dependence Trees. IEEE Transactions on Information Theory 14(3), 462–467 (1968)
3. Dembczyński, K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. Proceedings of the 27th International Conference on Machine Learning (ICML) (2010)

4. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10(7), 1895–1923 (1998)
5. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
6. Kong, X., Yu, P.: Multi-label feature selection for graph classification. In: 2010 IEEE International Conference on Data Mining (ICDM'10). pp. 274–283. IEEE (2010)
7. Kruskal Jr, J.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society 7(1), 48–50 (1956)
8. Lin, C.J., Weng, R.C., Keerthi, S.S.: Trust Region Newton Method for Logistic Regression. Journal of Machine Learning Research 9, 627–650 (2008)
9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD). pp. 254–269 (2009)
10. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA. vol. 2008 (2008)
11. Tsoumakas, G., Katakis, I.: Multi Label Classification: An Overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)
12. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multilabel Data. In O. Maimon and L. Rokach (Ed.), Data Mining and Knowledge Discovery Handbook, Springer (2010)
13. Tsoumakas, G., Vilcek, J., Spyromitros, L.: Mulan: A Java Library for Multi-Label Learning. http://mulan.sourceforge.net/
14. Van Der Gaag, L., De Waal, P.: Multi-dimensional Bayesian Network Classifiers. In: Proceedings of the Third European Workshop in Probabilistic Graphical Models, Prague. pp. 107–114 (2006)
15. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Pub (2005)
16. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proceedings of the International Conference on Machine Learning, (ICML'97),. pp. 412–420. Citeseer (1997)
17. Yu, L., Liu, H.: Efficient Feature Selection via Analysis of Relevance and Redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
18. Zhang, M.L., Peña, J.M., Robles, V.: Feature selection for multi-label naive bayes classification. Inf. Sci. 179, 3218–3229 (September 2009)
19. Zhang, M.L., Zhou, Z.: M3MIML: A Maximum Margin Method for Multi-instance Multi-label Learning. In: Eighth IEEE International Conference on Data Mining, 2008. ICDM'08. pp. 688–697 (2008)
20. Zhou, Z.: Learning And Mining from DatA (LAMDA). http://lamda.nju.edu.cn/data.ashx