Evaluating Feature Selection Methods for Multi-Label Text Classification

Newton Spolaôr[1] and Grigorios Tsoumakas[2]

[1] Laboratory of Computational Intelligence
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil
[2] Machine Learning and Knowledge Discovery
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
newtonspolaor@gmail.com,greg@csd.auth.gr

Abstract. Multi-label text classification deals with problems in which each document is associated with a subset of categories. These documents often consist of a large number of words, which can hinder the performance of learning algorithms. Feature selection is a popular task to find representative words and remove unimportant ones, which could speed up learning and even improve learning performance. This work evaluates eight feature selection algorithms in text benchmark datasets. The best algorithms are subsequently compared with random feature selection and classifiers built using all features. Results agree with literature by finding that well-known approaches, such as maximum chi-squared scoring across all labels, are good choices to reduce text dimensionality while reaching competitive multi-label classification performance.

Keywords: problem transformation, binary relevance, round-robin, rand-robin, chi-squared, bi-normal separation

1 Introduction

Classical single-label learning deals with problems in which each dataset instance (or example) is described by a set of features and associated with only one label from a disjoint set of labels L. Single-label text classification (or text categorization), for example, learns from data in which each document has a unique category (topic) as label. If $L = 2$, this task is called binary text classification, and it is called multi-class text classification if $L > 2$.

Although a large amount of research has been carried out on single-label learning, the correspondent learning algorithms do not fit well into applications composed of instances annotated with subsets of labels from L. Even in some text categorization problems, each document is labeled with several topics simultaneously, such that the learning algorithm should tackle more than one label simultaneously to learn accordingly. Motivated by this scenario, multi-label learning algorithms have been developed [32,26].

Irrelevant and/or redundant features can hinder the performance of single-label and multi-label learning algorithms due to the "curse of dimensionality" [16]. Thus, the Feature Selection (FS) task is often applied before learning to find features which describe the dataset as well as, or even better than, the original set of features does, and remove the remaining ones. FS also speeds up learning algorithms and sometimes improves their performance [34].

Research on multi-label feature selection is still scarce. For example, many publications evaluate a number of FS algorithms in only a few multi-label datasets. This work contributes to reduce this gap by comparing 8 FS methods in 20 multi-label text classification datasets (9 from different sources and 11 from a web page). The methods combine 2 feature evaluation measures, Chi-squared (CS) and Bi-Normal Separation (BNS) [8], with 4 aggregation strategies to tackle multiple labels [9], some of them still unexplored for multi-label datasets. Results show that well-known approaches, such as considering the maximum CS score of each feature across all labels, led to some of the best classification models.

The rest of this work is organized as follows: Section 2 briefly presents multi-label learning, FS and related work. Section 3 describes the methods evaluated in Section 4, which is followed by the conclusion and future work in Section 5.

2 Background

This section describes basic notations and concepts related to multi-label learning and feature selection. Related work in multi-label feature selection for textual datasets is also considered.

2.1 Multi-label learning

Let D be a dataset composed of N examples $E_i = (\mathbf{x}_i, Y_i)$, $i = 1..N$. Each example (instance) E_i is associated with a feature vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ described by M features (attributes) X_j, $j = 1..M$, and a subset of labels $Y_i \subseteq L$, where $L = \{y_1, y_2, \ldots y_q\}$ is the set of q labels. Table 1 shows this representation. In this scenario, the multi-label learning task consists in generating a model H which, given an unseen instance $E = (\mathbf{x}, ?)$, is capable of accurately predicting its subset of labels Y, *i.e.*, $H(E) \rightarrow Y$.

Table 1. Multi-label data.

	X_1	X_2	$\ldots$	X_M	Y
E_1	x_{11}	x_{12}	$\ldots$	x_{1M}	Y_1
E_2	x_{21}	x_{22}	$\ldots$	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\ldots$	x_{NM}	Y_N

Multi-label learning methods can be organized into two main categories: algorithm adaptation and problem transformation [26]. The former one includes learning algorithms extended to deal with multi-label data directly, such as the Multi-label Naive Bayes algorithm [33]. On the other hand, the latter category consists of algorithm independent methods, as any state of the art single-label learning method can learn from each single-label problem generated by these methods. The *Binary Relevance* (BR) approach exemplifies this category by transforming a multi-label dataset into q single-label datasets, learning from each single-label problem separately and combining the results.

Furthermore, exploiting label dependence during learning can improve performance [4]. An alternative categorization organizes multi-label learning methods based on the order of label correlations taken into account [32]. First-order strategies ignore co-existence of other labels during learning, as BR does. Second-order strategies, exemplified by Calibrated Label [10], consider pairwise relations between labels. High-order strategies, such as Random k-labelsets [28], consider relations among more labels.

Although high-order strategies potentially model wider label correlations, they are usually computationally more demanding. In this work, the problem transformation/first-order strategy BR is used for classification.

2.2 Feature selection

FS for multi-label text datasets often applies single-label feature evaluation measures, *i.e.*, measures to score the quality of features, after using problem transformation approaches, such as BR [6,31]. Moreover, these measures usually follow the filter approach [16]. Unlike the wrapper and embedded approaches, filters remove irrelevant and/or redundant features regardless of the learning algorithm, which can save computational resources when working with large datasets. Both FS measures used in this work agree with these popular choices.

CS and BNS share the same notation [8]. Let tp, fp, fn and tn be the number of (feature) true positives, false positives, false negatives and true negatives in a binary dataset. In this scenario, tp counts when a feature and a label under evaluation co-occur, *i.e.*, both are positive, while fp counts the cases in which only the feature is positive. Defining the remaining notations is straightforward.

Chi-squared estimates the independence between the occurrence of a feature X_j and the occurrence of a label y_i, such that the higher the measure value, the more related X_j and y_i are. CS is defined by Equation 1, where $P_{pos} = (tp + fn)/(tp + fp + fn + tn)$, $P_{neg} = (fp + tn)/(tp + fp + fn + tn)$ and $t(count{,}expect) = (count - expect)^2/expect$.

$$\begin{aligned} CS(tp{,}fp{,}fn{,}tn) = {} & t\left(tp, (tp + fp)\, P_{pos}\right) + t\left(fn, (fn + tn)\, P_{pos}\right) + \\ & t\left(fp, (tp + fp)\, P_{neg}\right) + t\left(tn, (fn + tn)\, P_{neg}\right). \end{aligned} \tag{1}$$

Bi-Normal Separation measures the separation between two thresholds (positive and negative classes) in a Gaussian function. This measure models the occurrence of a feature X_j in the documents as a random Normal variable exceeding a hypothetical threshold, such that the frequency of X_j corresponds to the area under the curve past the threshold. As defined by Equation 2, the higher the difference between the thresholds, the better the feature X_j is. Let F^{-1} be the standard Normal distribution inverse cumulative probability function (z-score), $tpr = tp/(tp + fn)$ and $fpr = fp/(fp + tn)$.

$$BNS(tp,fp,fn,tn) = |F^{-1}(tpr) - F^{-1}(fpr)|. \tag{2}$$

2.3 Related work

FS has been an active research topic in supervised learning, with several related publications and comprehensive surveys [34,26,16]. Most of this research has been mainly proposed to support single-label classification, but there are also many publications on feature selection for multi-label text classification.

The systematic review process, a method to perform a wide, replicable and rigorous literature review, was carried out in [22] and recently updated to search for multi-label FS publications. Most of the methods found are filters, which are useful to save time/space when working with large textual datasets. Table 2 summarizes some publications on filter FS. As mentioned, many publications use only few datasets to evaluate their methods. Moreover, Information Gain (IG) and CS, two potentially related measures [31], are the most usual ones.

Table 2. Multi-label filter feature selection publications.

Paper	Number of datasets used	Feature evaluation measure
[31]	2	IG, CS, document frequency, term strength and mutual information
[24]	2	IG
[18]	1	CS
[15]	1	CS
[35]	1	IG, CS, correlation coefficient, odds ratio, odds ratio-square and signed IG
[20]	1	IG, CS and document frequency
[3]	2	IG, CS and orthogonal centroid feature selection
[19]	1	conditional mutual information
[21]	1	minimal redundancy maximal relevance
[28]	3	CS
[2]	1	mutual information
[17]	1	BNS
[29]	5	Hilbert-Schmidt independence criterion
[6]	6	IG
[13]	8	symmetrical uncertainty
[14]	3	mutual information
[23]	10	IG and ReliefF

3 Multi-label feature selection methods

After transforming a multi-label dataset into q binary datasets by BR and counting the number of (feature) true/false positives/negatives, any feature evaluation measure for binary data can be applied according to the macro-averaged approach [15], exemplified in [23,25,28,20].

In what follows, four aggregation strategies are described. Let tp_{y_i}, fp_{y_i}, tn_{y_i} and fn_{y_i} be the number of (feature) true/false positives/negatives for a label y_i, $i = 1..q$ and a feature X_j, $j = 1..M$.

The well-known Mean strategy (Mean) [31] averages the scores obtained after applying the measure fe in the binary dataset related to each label y_i (Equation 3). Max (Max), which returns the maximum score obtained across all labels (Equation 4), is also popular.

$$Mean(X_j) = \frac{1}{q}\sum_{i=1}^{q} fe\left(tp_{y_i}, fp_{y_i}, tn_{y_i}, fn_{y_i}\right). \tag{3}$$

$$Max(X_j) = \max_{i=1}^{q} fe\left(tp_{y_i}, fp_{y_i}, tn_{y_i}, fn_{y_i}\right). \tag{4}$$

A finding that some feature evaluation measures can be blinded by a surplus of strongly predictive features for frequent labels, while largely ignoring features needed to discriminate hard (low frequency) labels, motivated the proposal of Round-Robin (RoR) and Rand-Robin (RaR) [9]. After calculating q feature rankings, the former variation takes the best feature in the ranking related to each label in turn. On the other hand, the latter one takes the best feature for a label randomly chosen with probability inversely proportional to its frequency. Each feature taken in turn is removed from the q feature rankings.

Algorithm 3.1 suggests a generic implementation for all the strategies, which can be optimized according to each one for code optimization. In what follows, the main procedures and variables of this algorithm are described.

Textual data is often represented as sparse data, as not all features (words) will occur in every instance (document). This property is considered by the procedure $invertedIndexes$ (Line 2) for countings. As result, there are $M + q$ rows, such that each row consists in the inverted indexes linking a feature or a label to the instances they occur. Thus, a (feature) true positive is verified every time a feature and a label co-occur in the same instance (Line 7). Based on the number of inverted indexes, *i.e.*, the frequency of each feature or label, fp, fn and tn are easily set. Then the score calculated by a feature evaluation measure fe (Line 15) is set to the matrix of feature rankings FRM.

Algorithm 3.1 ends with the application of one of the strategies in Line 19. It should be emphasized that the Equations 3 and 4 would use the matrix FRM instead of reapplying the feature evaluation measure fe.

Algorithm 3.1 implementation can support parallelization and serialization, enabling user to save time/space in large datasets. First, the algorithm is split into several independent tasks, which is helpful to successful parallelization [11]. Second, serialization is considered to enable the algorithm to save a stream of bytes in the disk and load it back when necessary, releasing space in memory.

Algorithm 3.1 Generic implementation for the aggregation strategies

```
Input: Multi-label dataset $D$
Output: Feature ranking $FR$
Initialize $tp$ and $FRM$
$\{invertedFeatureIndexes, invertedLabelIndexes\} \leftarrow invertedIndexes(D)$
for each $label$ of $invertedLabelIndexes$ do
    for each $feature$ of $invertedFeatureIndexes$ do
        for each $invertedIndexF$ of $feature$ do
            for each $invertedIndexL$ of $label$ do
                if $invertedIndexF = invertedIndexL$ then
                    $tp \leftarrow tp + 1$
                end if
            end for
        end for
        $fp \leftarrow numberIndexes(feature) - tp$
        $fn \leftarrow numberIndexes(label) - tp$
        $tn \leftarrow N - (tp + fp + fn)$
        $FRM[label][feature] \leftarrow fe(tp, fp, fn, tn)$
        Reinitialize $tp$
    end for
end for
$FR \leftarrow aggregationStrategy(FRM)$
return $FR$
```

4 Experimental evaluation

In this work, 8 text FS methods (2 *feature evaluation measures* × 4 *strategies*) are applied in 20 benchmark datasets. The best methods are after compared with the classifiers built using All Features (AF) and using the features selected by Random Feature Selection (RFS) [8]. The RaR strategy and RFS were executed three times due to their stochasticity, and the correspondent Micro F-Measure values from RaR and RFS were averaged before calculating the average ranking.

Some implemented procedures use Weka [30] and LIBLINEAR [7] resources. All the reported classification results were obtained by Mulan [27], a framework for multi-label classification, using 10-fold cross-validation with paired folds.

4.1 Datasets and experimental setup

Table 3 shows, for each dataset, its name the number of instances (N), features (M) and labels (q), the label cardinality (LC), which is the average number of labels associated with each example, the label density (LD), which is a normalized version of LC divided by the total number of labels, and the number of distinct combinations (DC) of labels. The datasets were obtained from *Mulan*[3] and *Meka*[4] repositories.

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

[4] http://meka.sourceforge.net/#datasets

Table 3. Benchmark datasets used.

Name	N	M	q	LC	LD	DC
arts[3]	7484	23146	26	1.65	0.06	599
bibtex[3]	7395	1836	159	2.40	0.02	2856
bookmarks[3]	87856	2150	208	2.03	0.01	18716
business[3]	11214	21924	30	1.60	0.05	233
computers[3]	12444	34096	33	1.51	0.05	428
delicious[3]	16105	500	983	19.02	0.02	15806
education[3]	12030	27534	33	1.46	0.04	511
enron[3]	1702	1001	53	5.31	0.06	753
entertainment[3]	12730	32001	21	1.41	0.07	337
health[3]	9205	30605	32	1.64	0.05	335
medical[3]	978	1449	45	1.25	0.03	94
language log[4]	13929	1002	23	1.66	0.07	1147
rcv1v2 (subset1)[3]	6000	47236	101	2.88	0.03	1028
recreation[3]	12828	30324	22	1.43	0.06	530
reference[3]	8027	39679	33	1.17	0.04	275
science[3]	6428	37187	40	1.45	0.04	457
social[3]	12111	52350	39	1.28	0.03	361
society[3]	14512	31802	27	1.67	0.06	1054
slashdot[4]	3782	1079	22	1.18	0.05	156
tmc2007-500[3]	28596	500	22	2.21	0.10	1341

After applying each FS method in a dataset, the BR + Linear SVM (*BRLL*) method, efficient to classify large sparse datasets [7], was used. *BRLL* classifiers were built from data described by the best t features found by a FS method, in which $t = 10\%, 20\%, \ldots, 90\%$ of the number of features M. The learning algorithm was executed with SVM $C = 3$, tolerance of stopping criterion $e = 0.001$ and remaining parameters with default values[5].

All classification models built were evaluated according to *Micro F-Measure* [26]. This evaluation measure, defined by Equation 5, has values in the interval [0..1] and the higher its value, the better the multi-label classifier performance is. Let $T_{P_{y_i}}$, $F_{P_{y_i}}$, $T_{N_{y_i}}$ and $F_{N_{y_i}}$ be, respectively, the number of true/false positives/negatives for a label y_j from the set of labels L.

$$Micro\,F\text{-}Measure(H,D) = \frac{2\sum_{j=1}^{q} T_{P_{y_j}}}{2\sum_{j=1}^{q} T_{P_{y_j}} + \sum_{j=1}^{q} F_{P_{y_j}} + \sum_{j=1}^{q} F_{N_{y_j}}}. \quad (5)$$

4.2 Results and discussion

The micro F-measure of the 8 feature selection methods at the 9 percentages of selected features for each one of the 20 datasets are available in an online appendix[6]. Following the recommendations in [5], we will here compare different feature selection approaches at specific percentages of selected features based on their average rankings across all datasets.

We first discuss the relative performance of the 4 aggregation strategies (Max, Mean, RoR, RaR) for each feature evaluation measure (CS, BNS) separately.

[5] Solvers in LIBLINEAR are insensitive to C.

[6] http://tiny.cc/e0ke3w

Table 4 shows the mean and standard deviation of the ranking of the 4 aggregation strategies (columns) at the 9 different percentages of selected features (rows) across all datasets for BNS (left) and CS (right). The best (lowest) average ranking for each evaluation measure and percentage of selected features is highlighted in bold. Figure 1 presents the same values as graphs of the average ranking (y-axis) with respect to the percentage of selected features (x-axis) for BNS (left) and CS (right).

Table 4. Mean and standard deviation of the ranking of the 4 aggregation strategies (columns) at the 9 different percentages of selected features (rows) across all datasets for BNS (left) and CS (right).

t	MaxBNS	MeanBNS	RoRBNS	RaRBNS	MaxCS	MeanCS	RoRCS	RaRCS
10%	2.00 (0.7)	**1.95** (1.4)	3.15 (0.8)	2.90 (1.0)	2.50 (1.0)	**1.40** (0.8)	3.10 (0.9)	3.00 (1.0)
20%	2.10 (0.8)	**2.05** (1.4)	3.05 (1.0)	2.80 (1.0)	2.50 (0.9)	**1.65** (1.1)	2.80 (1.1)	3.05 (1.0)
30%	**2.10** (1.0)	**2.10** (1.4)	3.15 (0.9)	2.65 (0.9)	2.30 (1.0)	**1.90** (1.2)	2.80 (1.0)	3.00 (1.0)
40%	2.40 (1.1)	**2.15** (1.4)	3.10 (1.0)	2.35 (0.8)	2.25 (1.0)	**1.95** (1.2)	2.50 (1.0)	3.30 (1.0)
50%	2.40 (1.1)	**2.30** (1.5)	2.95 (1.1)	2.35 (0.7)	**2.00** (0.8)	2.20 (1.3)	2.80 (1.1)	3.00 (1.0)
60%	2.50 (1.1)	2.50 (1.3)	2.95 (0.9)	**2.05** (1.1)	**1.90** (1.1)	2.15 (1.0)	2.60 (0.9)	3.35 (1.0)
70%	2.65 (1.4)	2.68 (1.3)	2.48 (0.9)	**2.20** (0.8)	**2.00** (1.0)	2.20 (1.0)	2.80 (1.0)	3.00 (1.2)
80%	2.80 (1.1)	2.55 (1.4)	2.50 (1.2)	**2.15** (0.8)	**2.25** (1.2)	2.35 (1.0)	2.65 (1.2)	2.75 (1.1)
90%	2.90 (1.1)	2.40 (1.2)	**2.30** (1.1)	2.40 (1.1)	**2.10** (1.2)	2.70 (0.9)	2.65 (1.2)	2.55 (1.2)

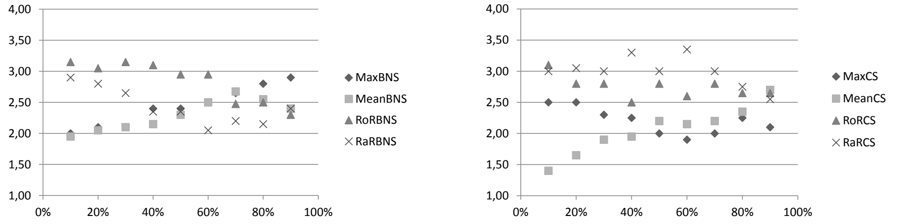

Fig. 1. Average rankings of the eight methods.

We notice that for both CS and BNS the Mean aggregation performs best when the percentage of features is low (up to 50% for BNS and 40% for CS). For larger percentages of features RaR (and RoR in the case of 90%) performs best for BNS, while Max performs best for CS. Recall that for each feature, Mean averages the scores across *all* labels, while Max, RoR and RaR are based on a *single* label. Therefore, at lower number of features, it is probably the case that

Max, RoR and RaR are not considering enough features for some of the labels in contrast to Mean. As the percentage of selected features increases, Max, RoR and Rar manage to select enough features for all labels and outweigh Mean, which is selecting features that work well for all labels *on average*.

The finding that Mean and Max lead to good classification models agrees with earlier publications which combine them with different feature evaluation measures, such as ReliefF and IG [23], CS [25,28,20] and mutual information [31].

We now focus on methods that in the previous comparison achieved the best average ranking for more than one percentage of selected features. These are: MeanBNS, RaRBNS, MaxCS and MeanCS. We will discuss the relative performance of these methods along with the baselines of using All Features (AF) and Random Feature Selection (RFS). Table 5 shows the corresponding average rankings and standard deviations. Note that the performance of AF is the same independently of the percentage of selected features, yet its relative ranking with respect to competing methods can and does differ.

Table 5. Mean and standard deviation of the ranking of MeanBNS, RaRBNS, MaxCS, MeanCS, RFS and AF (columns) at the 9 different percentages of selected features (rows) across all datasets.

t	MeanBNS	RaRBNS	MaxCS	MeanCS	RFS	AF ($t = 100\%$)
10%	3.60 (1.4)	4.15 (1.2)	3.10 (0.9)	**2.25** (1.1)	5.35 (1.1)	2.55 (2.2)
20%	3.60 (1.4)	4.05 (1.3)	2.75 (1.2)	**1.95** (1.0)	5.40 (1.1)	3.25 (2.0)
30%	3.35 (1.0)	4.05 (1.4)	2.45 (1.5)	**2.15** (1.1)	5.40 (1.1)	3.60 (1.9)
40%	3.30 (1.2)	3.80 (1.3)	**2.15** (1.2)	**2.15** (1.3)	5.45 (1.1)	4.15 (1.6)
50%	3.50 (1.4)	3.55 (1.2)	**2.00** (1.0)	**2.00** (1.1)	5.45 (1.1)	4.50 (1.4)
60%	3.55 (1.2)	3.20 (1.2)	**1.80** (1.1)	2.25 (1.1)	5.45 (1.1)	4.75 (1.3)
70%	3.25 (1.3)	2.85 (1.0)	**2.15** (1.2)	2.40 (1.4)	5.45 (1.1)	4.90 (1.1)
80%	3.35 (1.4)	2.75 (1.2)	**2.25** (1.3)	2.30 (1.1)	5.50 (1.0)	4.85 (1.2)
90%	3.38 (1.5)	2.90 (1.5)	**2.15** (1.5)	2.78 (1.0)	5.55 (0.8)	4.25 (1.5)

We notice that the best (lowest) average ranking is achieved by Max and Mean combined with CS. Besides obtaining better ranking than RFS, they also outperform AF, fulfilling the requirements of any reasonable feature selection method. CS, which behaves erratically for very small expected counts common in text classification, was found worse than BNS for single-label classification [8]. However, we here see that this scenario is reversed in the case of multi-label classification. The strategies Mean and Max seem to mitigate the disadvantage of CS. This could be because they consider more than one label, with different expected counts, when evaluating features.

We complete the comparison of the 8 feature selection methods by analyzing the similarity of the feature subsets that are selected by each method. In particular, we calculate a similarity index between each pair of methods [12]. This could be useful, for example, to identify diverse feature selection methods for constructing ensembles [1]. In this analysis, only one run of RaR is considered.

For the sake of saving space, Table 6 shows the similarity values averaged across all datasets at a specific percentage of features ($t = 50\%$), highlighting similarity values larger than 0.7 with bold typeface. Nevertheless, the patterns found also occur for other percentage of selected features.

Table 6. Similarity values of the feature subsets yielded by eight FS algorithms.

		MaxBNS	MeanBNS	RoRBNS	RaRBNS	MaxCS	MeanCS	RoRCS	RaRCS
BNS	Max		0.22	**0.75**	**0.72**	0.55	0.56	0.47	0.40
	Mean	0.22		0.15	0.17	0.33	0.42	0.23	0.14
	RoR	**0.75**	0.15		**0.82**	0.48	0.47	0.44	0.48
	RaR	**0.72**	0.17	**0.82**		0.47	0.49	0.44	0.40
CS	Max	0.55	0.33	0.48	0.47		**0.73**	**0.71**	0.53
	Mean	0.56	0.42	0.47	0.49	**0.73**		0.55	0.44
	RoR	0.47	0.23	0.44	0.44	**0.71**	0.55		**0.76**
	RaR	0.40	0.14	0.48	0.40	0.53	0.44	**0.76**	

We first notice that CS methods are quite different from BNS methods, as one would expect. Within BNS methods, we see that MeanBNS selects quite different feature subsets from the ones found by the other BNS methods, which in turn select relatively similar feature subsets. Within CS methods, we see 3 pairs of methods selecting similar features: RaR/RoR, Mean/Max and Mean/RoR.

5 Conclusion

This work evaluated 8 FS methods to support multi-label text classification in 20 benchmark datasets. They are based on 2 feature evaluation measures and 4 strategies to consider label information while evaluating features. The best methods from this group also highlighted in an experimental comparison with the classifiers built using all features and using features randomly selected.

The popular algorithms MeanCS and MaxCS, which respectively rank features according to the average or the maximum Chi-squared score across all labels, led to most of the best classifiers while using less features. The former was the best choice when the number of features was smaller. As the number of features increased, the latter yielded the best classifiers.

Future work will apply some of the best FS methods and their optimized implementation to rank features in large textual datasets. Furthermore, we plan to evaluate efficient FS methods which are able to consider label information in a higher level than the one considered by MeanCS and MaxCS [32].

Acknowledgment

This research was supported by the São Paulo Research Foundation (FAPESP), grant 2012/23906-2. The authors would like to thank Maria C. Monard, Huei D. Lee and Everton A. Cherman for their support in multi-label feature selection.

References

1. Cannas, L.M., Dessì, N., Pes, B.: Assessing similarity of feature selection techniques in high-dimensional domains. Pattern Recognition Letters 34(12), 1446–1453 (2013)
2. Chang, Y.C., Chen, S.M., Liau, C.J.: Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. Expert Systems with Applications 34(3), 1948–1953 (2008)
3. Chen, W., Yan, J., Zhang, B., Chen, Z., Yang, Q.: Document transformation for multi-label feature selection in text categorization. In: IEEE International Conference on Data Mining. pp. 451–456 (2007)
4. Dembczynski, K., Waegeman, W., Cheng, W., H?llermeier, E.: On label dependence and loss minimization in multi-label classification. Machine Learning 88, 5–45 (2012)
5. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
6. Dendamrongvit, S., Vateekul, P., Kubat, M.: Irrelevant attributes and imbalanced classes in multi-label text-categorization domains. Intelligent Data Analysis 15(6), 843–859 (2011)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
8. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3, 1289–1305 (2003)
9. Forman, G.: A pitfall and solution in multi-class feature selection for text classification. HPL-2004-86 (2004), Hewlett Packard
10. Fürnkranz, J., Hüllermeier, E., Mencía, E.L., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning 73(2), 133–153 (2008)
11. Kononenko, I., Robnik-Šikonja, M.: Weighting and local methods non-myopic feature quality evaluation with (R)ReliefF. In: Liu, H., Motoda, H. (eds.) Computational Methods of Feature Selection, pp. 169–191. Chapman & Hall/CRC (2008)
12. Kuncheva, L.I.: A stability index for feature selection. In: IASTED International Multi-Conference: artificial intelligence and applications. pp. 390–395 (2007)
13. Lastra, G., Luaces, O., Quevedo, J.R., Bahamonde, A.: Graphical feature selection for multilabel classification tasks. In: International Conference on Advances in Intelligent Data Analysis. pp. 246–257 (2011)
14. Lee, J., Kim, D.W.: Feature selection for multi-label classification using multivariate mutual information. Pattern Recognition Letters 34(3), 349–357 (2013)
15. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
16. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/CRC (2008)
17. Mayne, A., Perry, R.: Hierarchically classifying documents with multiple labels. In: IEEE Symposium on Computational Intelligence and Data Mining. pp. 133–139 (2009)
18. Nardiello, P., Sebastiani, F., Sperduti, A.: Discretizing continuous attributes in adaboost for text categorization. In: European Conference on Information Retrieval Research. pp. 320–334 (2003)

19. Novovičová, J., Somol, P., Haindl, M., Pudil, P.: Conditional mutual information based feature selection for classification task. In: Iberoamerican conference on Progress in pattern recognition, image analysis and applications. pp. 417–426 (2007)
20. Olsson, J.O.S., Oard, D.W.: Combining feature selectors for text classification. In: ACM International Conference on Information and Knowledge Management. pp. 798–799 (2006)
21. Saleh, S.N., El-Sonbaty, Y.: A feature selection algorithm with redundancy reduction for text classification. In: International Symposium on Computer and Information Sciences. pp. 1–6 (2007)
22. Spolaôr, N., Monard, M.C., Lee, H.D.: A systematic review to identify feature selection publications in multi-labeled data. ICMC Technical Report No 374. 31 pg. (2012), University of S?o Paulo
23. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: A comparison of multi-label feature selection methods using the problem transformation approach. Electronic Notes in Theoretical Computer Science 292, 135–151 (2013)
24. Toutanova, K., Chen, F., Popat, K., Hofmann, T.: Text classification in a hierarchical mixture model for small training sets. In: International Conference on Information and Knowledge Management. pp. 105–113 (2001)
25. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-label classification of music into emotions. In: International Conference on Music Information Retrieval. pp. 1–6 (2008)
26. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer (2010)
27. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414 (2011)
28. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: European Conference on Machine Learning. pp. 406–417 (2007)
29. Wang, B., Jia, Y., Han, Y., Han, W.: Effective feature selection on data with uncertain labels. In: IEEE International Conference on Data Engineering. pp. 1657–1662 (2009)
30. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2011)
31. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: International Conference on Machine Learning. pp. 412–420 (1997)
32. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms (in press). IEEE Transactions on Knowledge and Data Engineering PrePrints(PrePrints), 1–1 (2013)
33. Zhang, M.L., Peña, J.M., Robles, V.: Feature selection for multi-label Naive Bayes classification. Information Sciences 179, 3218–3229 (2009)
34. Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H.: Advancing feature selection research - ASU feature selection repository. Technical Report (2011), Arizona State University
35. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. SIGKDD Explorations Newsletter 6(1), 80–89 (2004)