Improving Multi-label Classifiers via Label Reduction with Association Rules

Francisco Charte[1], Antonio Rivera[1], María José del Jesus[1], and Francisco Herrera[2]

[1] Dep. of Computer Science, University of Jaén, Jaén, Spain
[2] Dep. of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
{fcharte,arivera,mjjesus}@ujaen.es, herrera@decsai.ugr.es
http://simidat.ujaen.es, http://sci2s.ugr.es

Abstract. Multi-label classification is a generalization of well known problems, such as binary or multi-class classification, in a way that each processed instance is associated not with a class (*label*) but with a subset of these. In recent years different techniques have appeared which, through the transformation of the data or the adaptation of classic algorithms, aim to provide a solution to this relatively recent type of classification problem.

This paper presents a new transformation technique for multi-label classification based on the use of association rules aimed at the reduction of the label space to deal with this problem.

Keywords: Multi-label Classification, Data Transformation, Dimensionality Reduction, Association Rules.

1 Introduction

The increasing volume of documents of all kinds on the web, especially texts and images, has originated the demand to properly classify them into non mutually exclusive categories. The availability of large databases in areas such as genetics has generated the need to analyze them in order to obtain useful knowledge, as well. These and other similar circumstances have made the interest in solving this problem, called multi-label classification [8], grow in recent years.

While this is a technique originated toward the field of document categorization [17], it is also applicable to other tasks of similar interest such as map labeling [6], medical diagnosis [5] or Bioinformatics [4].

To address this new problem it has been necessary to develop new classification methods, and also to define specific measures for analyzing the characteristics of multi-label datasets and evaluating the outputs of the classifiers. In recent years, much research has addressed the multi-label classification mainly through two ways [2]: 1) reducing the problem to a set of binary or multi-class classifiers via data preprocessing techniques and 2) through adaptation of existing algorithms to provide them with the capacity to deal simultaneously with multiple labels/classes.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 188–199, 2012.

However, few researchers have addressed the problem of high dimensionality in the label space, very common in multi-label datasets, and its influence on the results and the performance of the classifiers. Some multi-label datasets have hundreds or even thousands of different labels and, sometimes, their number exceeds the amount of features and often also the number of instances. There is, therefore, a problem of high dimensionality in a new space: the label space.

In this paper a new transformation method based on association rules, *Label Reduction with Association Rules* (LRwAR), is presented. An association rule mining algorithm is applied over the label space, obtaining a set of association rules. The ones that reach a minimum confidence level are used to preprocess the dataset, reducing label cardinality as well as the total number of labels in the dataset. This subset of rules is applied, after having executed the multi-label classifier, in order to retrieve the relevant labels in a postprocessing phase.

The rest of this paper is organized as follows. Section 2 presents the fundamentals of multi-label classification and the different approaches that have been addressed in the literature. Section 3 describes our proposal, aimed at improving both performance and results of multi-label classifiers. It also briefly introduces the concept of association rules and association rule mining. The experimental study performed and the results achieved are described in Section 4. Finally, Section 5 summarizes our conclusions.

2 Multi-label Classification

In Machine Learning, and particularly in supervised learning, one of the most important applications is classification. Using a set of labeled samples (dataset) the process starts by obtaining a model that is capable of labeling new samples not seen during the learning phase. Traditionally, classifiers are designed with datasets in which each sample is associated with one and only one class or label, which is the target value to obtain once the model has been built. L being the set of labels applied to the instances of a dataset, Xi all samples belonging to a particular class and l_a and l_b the indexes of two labels, the following premises must be met:

$$L = \{l_1, l_2, ..., l_k\}, |L| = k > 1. \tag{1}$$

$$X_{l_a} \cap X_{l_b} = \oslash, \forall l_a, l_b, a \neq b. \tag{2}$$

The premise in Equation 1 indicates that the set L must have at least two classes, since otherwise all samples would be associated with the same class. Equation 2 states that instances regarding their class are disjoint subsets or, put another way, that each sample corresponds to a single class.

When $|L| = 2$ then the classification is binary and the classes are usually identified as `true` or `false`. A clear example is found in the classifiers used to process the electronic mailing separated into two categories: `spam` and `non spam`. $|L| > 2$ specifies a multi-class classifier. Whereas a binary classifier would provide a positive or negative output, a multi-class classifier returns a value that, after the appropriate interpretation, will determine the class. In any case each instance belongs to only one of these classes.

Addressing a multi-label classification problem [8] Equation 2, which states that the instances subsets must be disjointed with respect to the class they belong to, is not satisfied, i.e. a sample can belong to more than one class. As indicated in Equation 3 the classifier, however, once trained will facilitate a set Y, subset of L, with the labels associated with each test instance. The premise in Equation 2 leads to the expression in Equation 4, telling us that subsets of instances are not necessarily disjointed with respect to their class.

$$Y = f(x_i), Y \subseteq L. \tag{3}$$

$$\neg\forall l_a, l_b, X_{l_a} \cap X_{l_b} = \oslash. \tag{4}$$

While in traditional classification the goal is to associate each sample to a class between $|L|$ possible classes, so that the range of possible output values is limited by the number of existing classes, in a multi-label classifier there are $2^{|L|}$ different possible values as output: it could be any combination of labels in L. A multi-label classifier, as stated in [8], generates its prediction in one of two ways: with a binary partition of the label set or with a label ranking.

The traditional classification algorithms based on trees, neural networks, support vector machines and instances, are designed to provide a single value as output: the class which the processed sample belongs to. These algorithms cannot be used directly, as such, to tackle a problem of multi-label classification. The literature [8] proposes two different ways to deal with this problem:

- To transform the dataset, making it possible to use the known classification algorithms, such as training a binary classifier for each label.
- To adapt a traditional classification algorithm, adding the ability to deal with the fact that each sample can be associated with multiple labels.

Each approach brings benefits but also has disadvantages that it is necessary to know in order to choose the best option.

2.1 The Data Transformation Approach

While many data transformation based methods have been proposed (a complete taxonomy can be found in [2]) Binary Relevance (BR) and Label Powerset (LP) are the two most important. These methods are algorithm independent and also known as *problem transformation methods*.

Introduced in [19] with the name *ensemble of binary classifiers*, the BR method divides the multi-label dataset into multiple binary datasets. An independent binary classifier will be trained for each dataset. There will be as many binary classifiers as different labels the original dataset had. Test samples will go through each binary classifier and finally, by the union of all the binary predictions, the multi-label prediction will be obtained. It is a simple method that allows us to use any underlying binary classification algorithm. Its main drawback is that it dismisses the relationship between labels, information potentially useful for improving the results in classification. It also implies a linear increase in execution time by the total number of labels in the dataset.

The LP method was introduced in [14] under the name *MODEL-n*. The final n highlights the fact that each distinct combination of labels in the data becomes a *new* class. Thus the original multi-label dataset becomes a classical one in which each data instance is associated with only one class, allowing the use of any multi-class classification algorithm. Unlike BR, the LP method takes into account the relationship between the labels by generating a new class for each different combination. The main problem with this method is that the number of combinations of labels is $2^{|L|}$, so the amount of classes could become intractable.

2.2 The Method Adaptation Approach

The transformation methods described above allow us to addressing the problem of multi-label classification using algorithms that are not designed for the specificities of the task. Faced with this choice, the focus of the algorithm adaptation approach aims to modify existing algorithms so that they can deal with multi-label samples, without requiring any preprocessing. In recent years the number of proposals published in this regard has increased strikingly. So here we just list the more remarkable ones.

In [4] the authors modified the C4.5 algorithm in order to classify genes according to their function, which can be multiple. They made two changes: each leaf of the tree stores not a class but a set of them, and the original entropy measure is adapted to take into consideration the fact that the samples are multi-label. Another tree-based algorithm is proposed in [7]. This begins with an ADT (Alternate Decision Tree) and incorporates an internal decomposition of the multi-label samples by OVA (One-vs-All) technique.

There are several adaptations of instance based algorithms. The most notable are ML-kNN[15] and IBLR-ML[20], the latter being a variation of the first. ML-kNN produces a ranking of labels from the closest neighbours of the instance to classify and basically operates as a binary classifier for each label, regardless of the relationships between them. IBLR-ML addresses this deficiency by considering the labels of nearest neighbours as additional input features to the classifier.

The first adaptation of a neural network to multi-label classification was BP-MLL[16], a perceptron with back-propagation learning which introduces a modified error function that takes into account the multi-label nature of the samples. Another proposal in this field is the ML-RBF[18] algorithm for designing multi-label RBFNs. The number of neurons in the hidden layer is calculated according to the number of labels in the dataset, obtaining their centers by the K-means algorithm. The training, which adjusts the weights of the connections between neurons, is performed with the SVD (Singular Value Decomposition) method and an adapted error function.

There are also proposals based on Support Vector Machine, such Rank-SVM[3], classifier chains[13] and ensembles of classifiers[9] and even those based on ant colonies, like MuLAM[1]. The number of publications related to the adaptation of algorithms for multi-label classification is constantly growing.

2.3 New Measures and Evaluation Metrics

The peculiarities of the problem faced require the use of new measures, firstly, to characterize the multi-label datasets, and secondly to facilitate the evaluation of the new algorithms.

In the first group there is *label cardinality* (Equation 5), defined as the average number of labels per sample in the dataset D, and *label density* (Equation 6), defined as label cardinality divided by $|L|$ (the total number of labels), providing a measure independent of the absolute number of labels in the dataset. These metrics offer information about multi-label datasets useful for fine-tuning the operation of the algorithms, as discussed later.

$$Card(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |Y_i|. \tag{5}$$

$$Dens(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|}. \tag{6}$$

In terms of measures that assess the quality of predictions, they can be grouped 1) by operating on a bipartition of the labels or on a ranking of these, and 2) according to the calculation method: averaging by instance (example based) or label (label based). There are more than a dozen different measures that are exposed in detail in [8]. One of the most widely used, and taken as reference in this work, is *Hamming Loss* (Equation 7), where Y_i is the set of predicted labels, Z_i the set of real labels and the operator Δ represents the symmetric difference.

$$HammingLoss = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta Z_i|}{|L|}. \tag{7}$$

3 LRwAR: Label Reduction with Association Rules

This section is devoted to describing LRwAR. Section 3.1 describes the problem of high dimensionality in the label space in multi-label datasets. Section 3.2 shows our approach to tackling this problem with a novel method of data transformation based on association rules.

3.1 The Problem of Dimensionality in the Label Space

Label cardinality (previously defined) is a factor that plays an important role in the performance of classifiers, affecting both their efficiency and effectiveness. By using transformation techniques like those described in Section 2, with the BR method a higher cardinality implies more binary classifiers must be used, so the computational complexity increases. The number of label combinations also increases, which affects the LP method and generally results in a reduction in

classification accuracy. Therefore it seems logical to think that if there were a way to reduce the cardinality when there is a problem of high dimensionality in the label space, the classifier performance could improve globally.

Algorithms adapted to the multi-label problem are also affected by the problem of high dimensionality in the label space. It is important to keep in mind that many of these algorithms are based on an internal data transformation similar to those described above, of which the most usual is based on training multiple binary classifiers and combining their results. Also, the predictive models obtained are much more complex with a larger number of labels and high cardinality datasets. With the reduction of cardinality these models will be simpler, more efficient and more accurate.

This paper proposes a methodology based on the transformation of data for multi-label classification which uses association rules. Association rules[12] describe co-occurrences between elements that provide knowledge about the underlying system to a set of data and can be interpreted as implications, so that the presence of certain elements implies the occurrence of others. The process of extracting association rules from a database or set of transactions is known as association rule mining, and there are many algorithms for doing this. In [12] there is a review of the most common ones, many of them based on the best known: the Apriori algorithm. As stated in [11], FP-Growth is a better association rule mining algorithm for working with large transaction databases.

Our interest in the association rules is related to their use as a tool for hiding to the multi-label classifier labels that can be inferred, with a certain confidence level, from the appearance of other labels. For this we used the so-called *support-confidence framework* defined in the literature.

3.2 LRwAR: A Method for Reducing Dimensionality in the Label Space

Label Reduction with Association Rules (LRwAR) is an original method that, through the application of preprocessing and postprocessing stages and based on the extraction of association rules, can reduce the cardinality of multi-label datasets. This reduction improves the performance and results obtained by the underlying classifiers. Operating exclusively in that label space (ignoring all other attributes), we assume that each label is an *item* and that the labels assigned to each instance of the dataset form a *transaction*. The objective is to obtain, with the FP-Growth algorithm, rules that allow the inference of certain labels from the presence of others with a certain level of confidence. The inferred labels can be hidden from the multi-label classifier. Hidden labels are added to each instance after the classifier's prediction in a postprocessing phase, simply by applying the previously-obtained association rules.

The number of rules obtained from a multi-label dataset is a parameter that depends not on the total number of different labels it contains, but on label cardinality. If that measure is small, close to 1, transactions are mostly formed by 1 or 2 items and it will be difficult to obtain valid rules.

The application of each obtained rule involves removing one or more labels from the dataset. It should be noted that apart from reducing the total number of labels in the dataset the most affected parameter, significantly reduced, will be label cardinality. This will help to reduce both execution time and complexity of models generated and, in theory, should also improve classification results.

Method Description. The proposed transformation method, including preprocessing and postprocessing stages, is described in the following pseudocode:

```
1. X = Dataset to process
2. T =   # Label set per instance (transactions)
3. For each instance Xi in X
4.   Li = labels of Xi
5.   T = T ∪ Li
6. R = FPGrowth(T) # Rules set ordered by confidence
7. MR = SubC(R)    # Get better rules in R
8. C = LabelsInConsequent(MR)
9. X = X - C   # Hide labels in C from original dataset
10. DTra = TrainingPartition(X)
11. DTst = TestPartition(X)
12. Clas = ObtainClassifier(DTra)
13. For each instance Xi in DTst
14.   Pi = Clas(Xi) # Obtain classifier's prediction
15.   Pi = Pi ∪ ApplyRules(MR) # Label inference
16. Evaluate(P)
```

The FPGrowth algorithm (line 6) returns a set of association rules ordered by a certain quality measure, in our case confidence. From this set the method obtains a subset (line 7) with the best rules, those that achieve a minimum value in this measure.

Once the classifier `Clas` is obtained, and when the test partition processing starts, the postprocessing step comes into the picture. This, as shown in line 15, consists of the application to the prediction of the classifier for each test sample the rules used when the training was realized, inferring the labels necessary to add. Finally, given `P` the total set of predictions for the test samples, the calculation of measures of quality assessment is carried out.

Method Implementation. The method described above has been implemented in Java and integrated with the MULAN[10] software, so that the process of partitioning the dataset, classifier training and test outcome rests entirely on MULAN.

The preprocessing stage creates a new multi-label dataset, having deleted the labels that can be inferred from chosen rules. That reduced dataset is given as input to MULAN. In order to integrate the postprocessing step a specialized class has been derived from MULAN's `Evaluator` class, adding labels inferred

from the rules after the classifier's prediction but before evaluation measures calculation.

MULAN stores for each processed sample the label's associated bipartition and also the measure associated with each of them that will serve to build the label ranking. LRwAR delivers to MULAN an updated bipartition while post-processing, in the application of association rules phase, including rule confidence as the measure for ranking generation.

4 Experimental Framework, Results and Analysis

This section describes the experimental study conducted for this paper. Datasets, algorithms and parameters are detailed in Section 4.1. Results are shown and discussed in Section 4.2.

4.1 Experimental Framework

In order to empirically test the proposed method we selected a set of MULAN[10] repository's datasets with mixed characteristics: many vs. few labels, high vs. low cardinality, etc. Table 1 indicates the name of each dataset and its main characteristics in instance: total number and number of distinct instances (those which have different values in at least one attribute), attribute: number of nominal and numeric attributes, and label spaces.

Table 1. Datasets used in experimentation and their characteristics

Dataset	Instances		Attributes		Labels		
Name	Number	Distinct	Nominal	Numeric	Total	Cardinality	Density
emotions	593	27	0	72	6	1.869	0.311
CAL500	502	502	0	68	174	26.044	0.150
Corel5k	5000	3175	599	0	374	3.522	0.009
enron	1702	753	1001	0	53	3.378	0.064
yeast	2417	198	0	103	14	4.237	0.303

The execution of the FP-Growth algorithm on the labels of each dataset, in order to extract association rules, was performed with the following parameters: support was set to 0.025 and minimum confidence at 0.5.

As for multi-label classification algorithms used for this study, we wanted to have a representation of methods based on both data transformation and method adaptation approaches. It should be noted that the transformation methods do their work after the preprocessing phase described previously, operating on a reduced dataset having already eliminated the labels inferred from the rules. The methods chosen, all implemented in MULAN, were:

- **BR-J48/LP-J48:** Performs a BR or LP transformation, respectively, and uses the classification algorithm C4.5 (called J48 in Weka/MULAN) as underlying algorithm for binary classification.

- **MLkNN:** This algorithm was proposed in [15], it is a k nearest neighborhood instance based method.
- **IBLR-ML:** An improved version of the previous method, presented in [20]. Combines instance based classification and logistic regression.
- **BP-MLL:** This algorithm was proposed in [16]. It is a back-propagation neural network based method.

Each run of the proposed method, once the set of rules to apply has been obtained and the dataset preprocessed, is repeated 5 times for each classification method with the instances of the dataset distributed randomly at the beginning of each repetition. Cross-validation is used with the usual configuration at 10 partitions, so there is a total of 5 repetitions x 10 partitions = 50 runs of the underlying algorithm for each dataset. The results are averages of those 50 runs, obtaining for every one the HammingLoss (HL) measure described in Section 2.

4.2 Results and Analysis

Table 2 shows the results obtained with transformation methods: LP and BR, whereas Table 3 shows the results from methods adapted to multi-label classification. In both tables the first column of every algorithm corresponds to the measure (`HL` metric) of the base method, the second ($LRwAR_1$) is the result obtained by applying all the obtained rules and the third ($LRwAR_2$) after the application of only the most confident association rule. The best results are highlighted in bold.

Table 2. Results from transformation methods

Algorithm	LP-J48			BR-J48		
Dataset	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$
emotions	0.2777	0.2770	**0.2476**	**0.2474**	0.2793	0.2584
CAL500	0.1996	**0.1991**	0.1999	0.1615	**0.1607**	0.1618
Corel5k	0.0168	**0.0161**	0.0167	0.0098	**0.0097**	0.0098
enron	0.0716	0.0761	**0.0715**	**0.0508**	0.0581	0.0533
yeast	0.2780	0.2808	0.2780	**0.2454**	0.2508	0.2468

Table 3. Results from adapted methods

Algorithm	MLkNN			IBLR-ML			BP-MLL		
Dataset	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$
emotions	0.1951	0.2299	**0.1904**	0.1883	0.2205	**0.1872**	0.2061	0.2633	**0.1999**
CAL500	0.1388	**0.1387**	0.1389	0.2307	**0.2300**	0.2315	**0.2478**	0.2509	0.2506
Corel5k	0.0094	**0.0093**	0.0094	0.0225	**0.0219**	0.0229	0.8315	0.7735	**0.7631**
enron	**0.0524**	0.0573	0.0543	**0.0557**	0.0610	0.0573	**0.3831**	0.6424	0.5443
yeast	0.1933	0.1933	**0.1931**	0.1934	**0.0092**	0.1928	0.2258	0.2264	**0.2244**

From the analysis of these results it can be observed that the proposed method obtains 17 best results against 7 cases with no improvements. Our proposal outperforms all the base methods except for BR-J48.

If we analyze the datasets from their characteristics, Table 1, we appreciate that for datasets with the largest number of labels and higher cardinalities, such as `CAL500` (174 labels) and `Corel5k` (374 labels), the label reduction applying all the obtained rules achieves the best results. By contrast, for datasets with a low number of labels, such as `emotions` and `yeast`, our method improves the results by applying only the most confident rule.

It seems logical to conclude, therefore, that the elimination of the maximum number of labels (applying all the obtained rules) and reduction of label cardinality, has a positive influence when the cardinality and total number of labels is large, whereas the use of only one rule is more appropriate for datasets with fewer labels and a minor label cardinality.

It should be noted that these results, which maintain or even improve the evaluation measures, are obtained in a shorter run time and generate simpler models than the original, since they work with a reduced version of the datasets. Is a fact that becomes clear in Table 4, which shows the average execution times in seconds for every algorithm-dataset combination. Our proposal improves in all cases except for the Corel5k-MLkNN combination. It must be highlighted that MLkNN is a lazy sort algorithm that does not generate a prediction model, most of its run time is used to calculate distances between the input variables of the sample data space, a space that is not modified by our proposal.

Table 4. Average run time (seconds)

Algorithm	LP-J48		BR-J48		MLkNN		IBLR-ML		BP-MLL	
Dataset	Base	LRwAR	Base	LRwAR	Base	LRwAR	Base	LRwAR	Base	LRwAR
emotions	6.93	**1.87**	4.49	**1.43**	3.50	**3.27**	5.59	**4.36**	7.88	**4.49**
CAL500	71.34	**68.58**	14.97	**14.41**	4.39	**3.79**	316.38	**283.53**	277.79	**260.95**
Corel5k	5333.12	**4442.04**	448.52	**445.82**	**1244.7**	1333.03	17870.88	**16648.92**	4815.05	**4004.86**
enron	2174.22	**1859.17**	125.81	**125.11**	44.07	**43.80**	150.84	**122.91**	1553.37	**1501.86**
yeast	144.99	**114.01**	107.22	**86.15**	112.62	**101.59**	138.62	**121.6**	66.19	**56.9**

5 Conclusions and Future Work

In this work we have presented a novel transformation method designed to reduce the number of labels in a multi-label dataset, as well as the label cardinality, by the use of association rules. This approach can be used with any underlying multi-label classification algorithm, allowing classifier training in less time, resulting in simpler models and, in many cases, improving evaluation measures.

Experimentation and results (Section 4.2) lead us to conclude that it is a useful approach to reducing label cardinality of multi-label datasets. Adjusting the optimal number of association rules to apply, depending on the characteristics of the processed dataset, needs a deeper analysis. Both the definition of the proposed method and the experimentation are a first approach to dimensionality reduction in the label space, an alternative studied poorly until now.

Acknowledgments. Supported by the Spanish Ministry of Education under the F.P.U. National Program (Ref. AP2010-0068), the Spanish Ministry of Science and Technology under the Project TIN2008-06681-C06-02, FEDER founds, and the Andalusian Research Plan TIC-3928, FEDER founds.

References

1. Chan, A., Freitas, A.A.: A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 27–34 (2006)
2. de Carvalho, A., Freitas, A.: A Tutorial on Multi-label Classification Techniques. Foundations of Computational Intelligence 5, 177–195 (2009)
3. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Neural Information Processing Systems, 681–687 (2001)
4. Clare, A.J., King, R.D.: Knowledge Discovery in Multi-Label Phenotype Data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, p. 42. Springer, Heidelberg (2001)
5. Karalic, A., Pirnat, V.: Significance level based multiple tree classification. Informatica 15(5) (1991)
6. Zhu, B., Poon, C.K.: Efficient Approximation Algorithms for Multi-label Map Labeling. In: Proceedings of the 10th International Symposium on Algorithms and Computation, pp. 143–152 (1999)
7. Comité, F., Gilleron, R., Tommasi, M.: Learning Multi-label Alternating Decision Trees from Texts and Data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003)
8. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
9. Tsoumakas, G., Vlahavas, I.P.: Random k -Labelsets: An Ensemble Method for Multilabel Classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
10. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: MULAN: A Java Library for Multi-Label Learning. Journal of Machine Learning Research, 2411–2414 (2011)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of the 2000 ACM-SIGMOD International Conference on Management of Data, vol. 29(2), pp. 1–12 (2000)
12. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining - a general survey and comparison. ACM SIGKDD Explorations Newsletter 2(1), 58–64 (2000)
13. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS Part I, vol. 5782, pp. 254–269. Springer, Heidelberg (2009)
14. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition 37, 1757–1771 (2004)
15. Zhang, M., Zhou, Z.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40, 2038–2048 (2007)

16. Zhang, M., Zhou, Z.: Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering 18, 1338–1351 (2006)
17. Zhang, M., Zhou, Z.: A k-nearest neighbor based algorithm for multi-label. In: Proceedings of the 1st IEEE International Conference on Granular Computing, pp. 718–721 (2005)
18. Zhang, M.: Ml-rbf: RBF Neural Networks for Multi-Label Learning. Neural Processing Letters 29, 61–74 (2009)
19. 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)
20. Cheng, W., Hüllermeier, E.: Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. Machine Learning 76, 211–225 (2009)