A Tutorial on Multi-label Classification Techniques

Chapter · July 2009

DOI: 10.1007/978-3-642-01536-6_8

2 authors:

Andre de Carvalho
University of São Paulo
319 PUBLICATIONS **2,795** CITATIONS
SEE PROFILE

Alex Freitas
University of Kent
263 PUBLICATIONS **8,873** CITATIONS
SEE PROFILE

A Tutorial on Multi-Label Classification Techniques

André C P L F de Carvalho[1] and Alex A. Freitas[2]

[1] Department of Computer Science, University of São Paulo, São Carlos, SP, Brazil, andre@icmc.usp.br, http://www.icmc.usp.br/~andre

[2] Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK, A.A.Freitas@kent.ac.uk, http://www.cs.kent.ac.uk/~aaf

Summary. *Most classification problems associate a single class to each example or instance. However, there are many classification tasks where each instance can be associated with one or more classes. This group of problems represents an area known as multi-label classification. One typical example of multi-label classification problems is the classification of documents, where each document can be assigned to more than one class. This tutorial presents the most frequently used techniques to deal with these problems in a pedagogical manner, with examples illustrating the main techniques and proposing a taxonomy of multi-label techniques that highlights the similarities and differences between these techniques.*

1 Introduction

Machine Learning (ML) is a sub-area of Artificial Intelligence (AI) concerned with the induction of a model through a learning process. A particular area of ML, named Inductive Learning, consists of techniques that induce these models by using a set of previously known instances or examples, called training instances. After the model has been induced, it can then be applied to new, previously unseen, data.

ML models have been applied to a wide range of tasks. These tasks can be broadly divided into five main classes: Association, Classification, Regression, Clustering and Optimization tasks. This paper is concerned with classification tasks, which can be formally defined as:

> *Given a set of training examples composed of pairs $\{x_i, y_i\}$, find a function f(x) that maps each attribute vector x_i to its associated class y_i, i = 1, 2, ..., n, where n is the total number of training examples.*

Once it has been trained, the classification model can have its predictive accuracy estimated by applying it to a set of new, previously unknown, examples. Its accuracy measure for these new instances estimates the generalization ability (predictive accuracy) of the classification model induced.

Classification problems can be categorized according to the number of class labels that can be assigned to a particular input instance. The most common approach is to have mutually exclusive classes. For example, suppose a document classification problem where each document should be classified according to the language it was written. If a document could be written in just one idiom and the possible

idioms were Chinese, English, French, German, Portuguese and Spanish, each document would be classified in one and only one of these six classes. In this case, each input instance is assigned to only one of the possible classes. This is known as single-label classification. Most of the classification problems investigated in ML are single-label classification problems.

However, there is a large number of relevant problems where each instance can be simultaneously associated with more than one class. These problems, where the classes are not disjoint, are known as multi-label classification problems.

The majority of the works on multi-label classification started as an attempt to deal with ambiguities found in document classification problems [51]. In a document categorization problem, each document may simultaneously belong to more than one topic or label. For example, a document can be classified as belonging to Computer Science, Physics and Application, another document can be assigned to the areas of Biology and Theory and a third can be a Mathematics document related to an Application in Physics. This problem would then have at least six classes or labels (Computer Science, Physics, Application, Biology, Theory and Mathematics). Even now, text classification is the main application area of multi-label classification techniques [21][24][26][28][29][30][36][42][48][50]. However, relevant works can also be found in areas like bioinformatics [11] [51], [14], medical diagnosis [25], scene classification [4][37] and map labeling [53].

Different approaches have been proposed in the literature for dealing with multi-label problems. One of them combines single-label classifiers to deal with the multi-label classification task. A second approach modifies single-label classifiers, by the adaptation of their internal mechanisms, to allow their use in multi-label problems. A third group proposes new algorithms specifically designed to deal with multi-label problems.

This text is organized as follows. In the next section, the main methods found in the literature for dealing with multi-label classification are organized and described. First, the authors define the structure and main characteristics of multi-label problems, without worrying about the learning algorithms used. Later, the authors discuss some algorithm specific approaches. Section 3 discusses how new instances can be classified in a multi-label environment. Section 4 has the final considerations and main conclusions of this work.

2. Categorizing Multi-Label Classification Problems

In a trained classifier, a probability can be associated with each one of the existent classes and then be used for the classification of a new example. Thus, if the problem has N classes, a probability p_i, $1 \leq i \leq N$, where $0 \leq p_i \leq 1$, is assigned to each class. If the system is trained for single-label classification, there is a restriction that $\sum p_i = 1$. For a multi-label problem, this restriction is not adopted.

According to [13], binary classification, multi-class classification and ordinal regression problems can be seen as special cases of multi-label problems where the number of labels assigned to each instance is equal to 1.

Some of the solutions to multi-label problems are restricted to binary classification. However, the largest number of methods found in the literature are used for are multi-class problems. For multi-class problems, the main focus of this text, the original multi-label problem is converted to one or more single-label problems.

In order to illustrate the different methods found in the literature for multi-label classification problems, these methods are organized in a hierarchical structure in Figure 1.

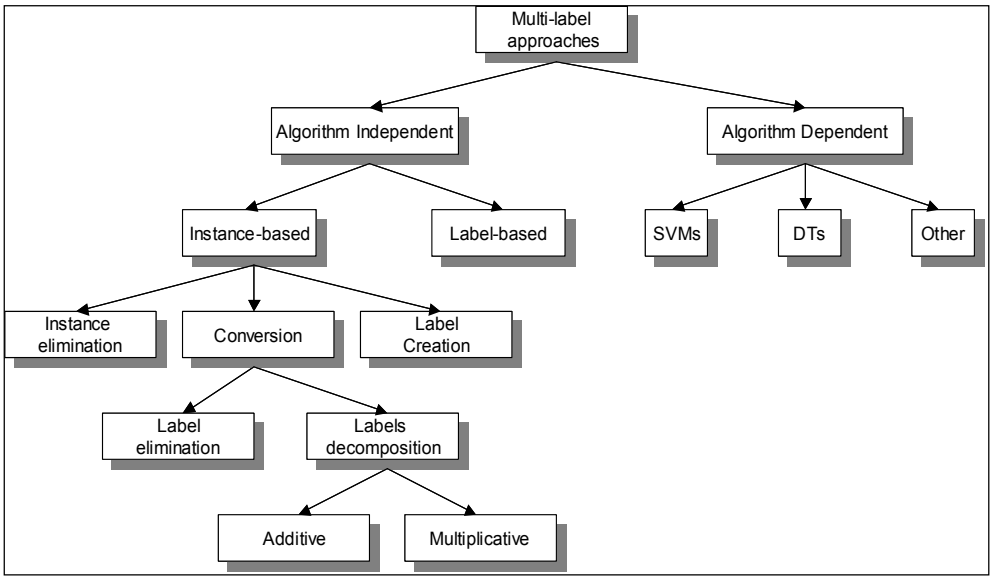

Fig. 1. Methods used in Multi-Label Classification Problems

According to Figure 1, the existing methods can be divided into two main approaches: algorithm independent and algorithm dependent. The next sections describe the main characteristics of the methods belonging to each approach.

2.1 Algorithm Independent Approach

The algorithm independent multi-label methods can be used with any learning algorithm. In this approach, a multi-label classification problem is usually dealt with by transforming the original problem into a set of single-label problems. This transformation can be based on either the class labels, named label-based, or the instances, named instance-based.

Label-based transformation

In the label-based transformation, N classifiers, where N is the number of classes, are used in the multi-label problem. Each classifier is associated with one of the classes and trained to solve a binary classification problem, its class against the others. For this reason, this approach is also known as the binary approach or cross-training [4]. Any classifier can be used for binary classification. Many popular classifiers can deal only with binary classification problems.

As an example of the use of this approach, suppose a multi-label problem, illustrated by Figure 2, with 3 classes or labels. Since one classifier should be associated with each class, 3 classifiers would be trained. The multi-label problem with 3 classes is then divided into 3 binary problems, one for each class. The i^{th} classifier would be trained to classify examples from the i^{th} class as positive and examples from the remaining classes as negative. Therefore, each classifier would

be specialized for a particular class. After the classifiers are trained, whenever a previously unknown example is presented, the classes whose classifier produced a positive label are assigned to it [1].

One of the first works in multi-label classification was due to [25]. In this work, the authors investigated the use of Decision Trees, DTs, in multi-label problems. They proposed a tree-based model, named MULTI-α, which divides the original multi-label problem into N single-label sub-problems, where N is the number of classes. Thus, for each class C_i, $1 \leq i \leq N$, it generates a decision tree using the classes C_i and $\neg C_i$. The outputs above a threshold value are assumed to be correct. The set of outputs produced by the individual classifiers provide the system's final decision. The method was evaluated in a medical diagnosis problem. It is possible to see that this method is similar to one of the algorithm-independent methods, the Label-based transformation.

Similarities can be found between the label-based transformation and the one-against-all approach employed for multi-class problems [23]. However, the one-against–all approach is employed to allow the solution of problems with more than two classes using binary classifiers. Another difference is that, in a multi-class problem, each example is assigned to only one class.

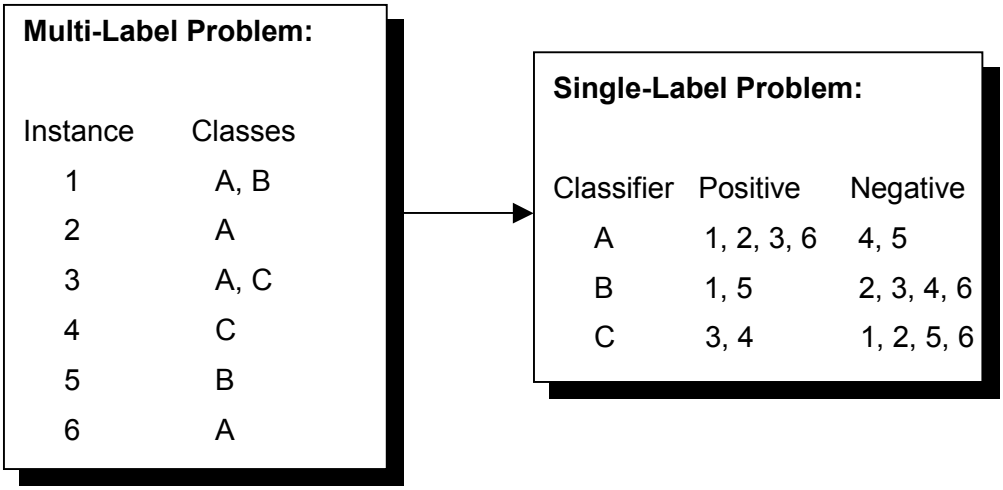

Fig. 2. Label-based transformation

This approach assumes that the labels of an instance are independent among themselves, which is not always true. By ignoring the possible correlation between labels, this approach may lead to poor generalization ability.

A label-based transformation is reversible, since it is possible to recover the original multi-label problem from the new single-label problem. It requires N classifiers, where N is the number of classes.

Instance-based transformation

In the transformation based on instances, named instance-based, the set of labels associated to each instance is redefined in order to convert the original multi-label problem into one or more single-label problems. In this redefinition, one or more classification problems can be produced. Different from label based

transformations, which produce only binary classification problems, instance based transformations may produce both binary and multi-class classification problems.

Three different groups of strategies have been proposed in the literature for instance-based transformation:

- Elimination of multi-label instances;
- Creation of new single-labels using the existent multi-labels, here named conversion
- Conversion of multi-label instances into single-label instances:
  - Simplification;
  - Decomposition:
    - Additive;
    - Multiplicative.

Instance elimination is the simplest, but probably the least effective instance-based strategy. It does not solve the original multi-label problem. The elimination of those instances with more than one label will change the current problem into another, much simpler problem, possibly not as relevant as the previous one. An example of the use of this approach is shown in the Figure 3. According to this figure, the multi-label instances, 1 and 3, are eliminated in order to transform the original multi-label problem into a single-label problem. The negative aspect of this approach is that the instances eliminated can represent relevant information to characterize the problem domain.

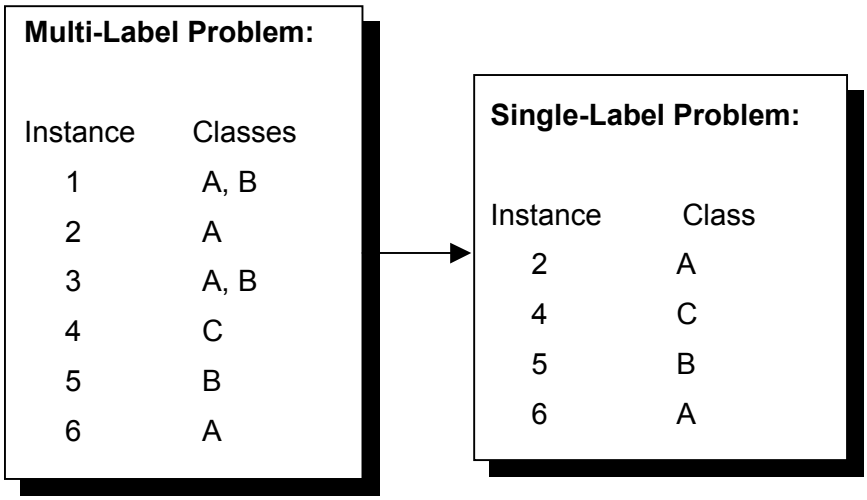

Fig. 3. Elimination of instances with more than one label

For protein classification, for example, many proteins have more than one function. How would the user be able to predict the other functions? The elimination of these proteins from the data set would significantly reduce the significance of the model induced by the classifier. Since it is not possible to find out, in the new single-label problem, which instances were eliminated, this method is irreversible. It does not change the number of required classifiers.

There are other methods reported in the literature that, although classifier independent, aim to improve the performance by pre-processing the data set rather than naively eliminating all multi-label instances. In [20], the authors propose the removal of the instances close to the decision hyperplane and the elimination of the instances in the confusing classes. The confusing classes are defined using the confusion matrix.

When label creation is adopted, each possible combination of more than one class is converted to a new single class (label). The combination of the original classes can largely increase the number of classes and result in some classes with very few instances. This problem becomes increasingly worse as the number of possible labels for each instance increases. Figure 4 illustrates this approach.

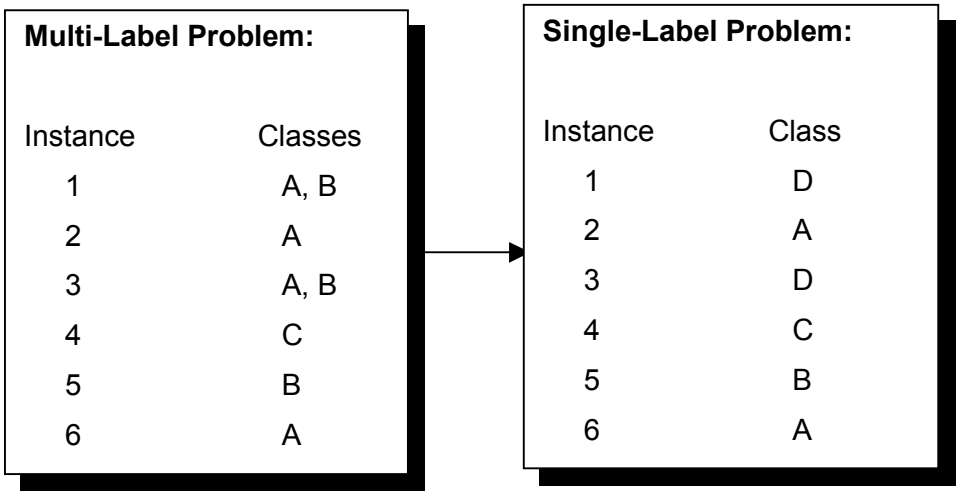

Fig. 4. Creation of new classes

It can be easily observed in this figure that the labels of the two multi-label instances, 1 and 3, which were A and B in both cases, were combined to create a new label, D.

The labels associated with each instance in the original multi-label problem are not lost in the creation of the new labels for the single-label problem. The number of classifiers is the same in both problems if a multi-class classifier is used. However, if a binary classifier is used, the number of classifiers required increases, by comparison with the original multi-label problem.

For the case of label conversion, there are two variations. The first variation transforms each multi-label instance into a single label instance. It is named label simplification. In the second variation, named label decomposition, each multi-label instance is decomposed into a set of single-label instances.

When transforming a multi-label instance into a single-label one, if the instance has more than one label, one of its labels is selected. The other labels are just eliminated. Two alternatives can be followed for the label selection. This procedure can either use a deterministic criterion, selecting from the labels associated with the instance the most likely to be true, or randomly select one of the labels. Figure 5 shows an example of this approach.

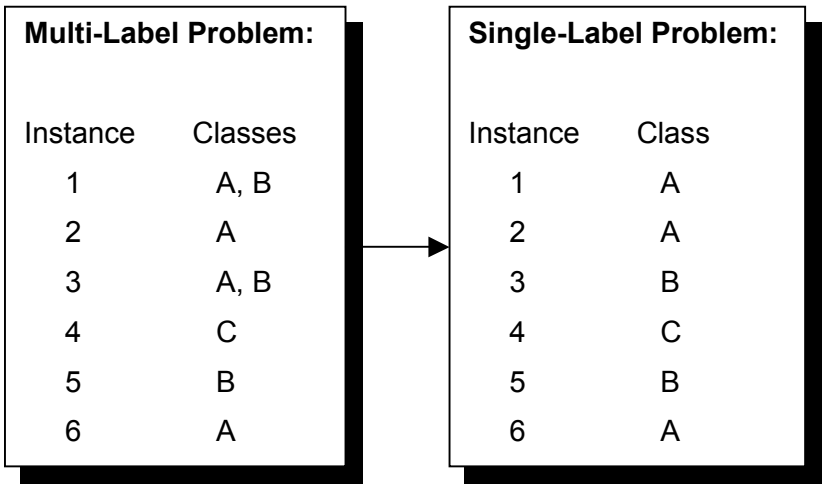

Fig. 5. Transform by label elimination of a multi-label problem into a set of single-label problems

It is easy to see in this example the simplification of the two multi-label instances, 1 and 3, by randomly selecting one of the labels, in both cases A and B, associated with each of them. As a result, the label A was randomly assigned to the instance 1 and the label B was randomly associated with the instance 2.

The selection of one of the labels will over-simplify the problem. Suppose that the classification problem involves the functional classification of a protein. A protein with more than one function would be classified as having just one of the functions, thus ignoring possibly relevant information.

If the deterministic criterion is adopted, it is possible to return to the original multi-label problem from the new single-label problem. If the random criterion is chosen, this return is not possible. The same number of classifiers is generally used in the multi-label and the single-label problems.

In the decomposition approach, the original multi-label problem with N classes and M instances is divided into K sets of single-label problems. The value of K varies from 1, when no instance has more than one label, to $(N-1)^M$, if all the instances have N-1 labels. Two alternatives can be employed for this approach: the additive method and the multiplicative method.

In the additive method, for each instance, each of the possible labels is considered to be the positive class in sequence. Therefore, the number of classifiers is given by $\sum_t (l_i-1)$, where l_i is the number of labels in the i^{th} instance. Thus, if the labels A, B and C appear in the multi-label instances, when the classifier for the class A is trained, all the multi-label instances that have the label A become single-label instances for the class A. The same happens for the other labels. This method was proposed in [37] and is named cross-training.

The number of classifiers, K, is equal to the number of labels that belong to at least one multi-label instance. This method allows the recovery of the original multi-label problem from the new single-label problems. Figure 6 illustrates this method. For this situation, the number of classifiers is given by 1 + 1 = 2.

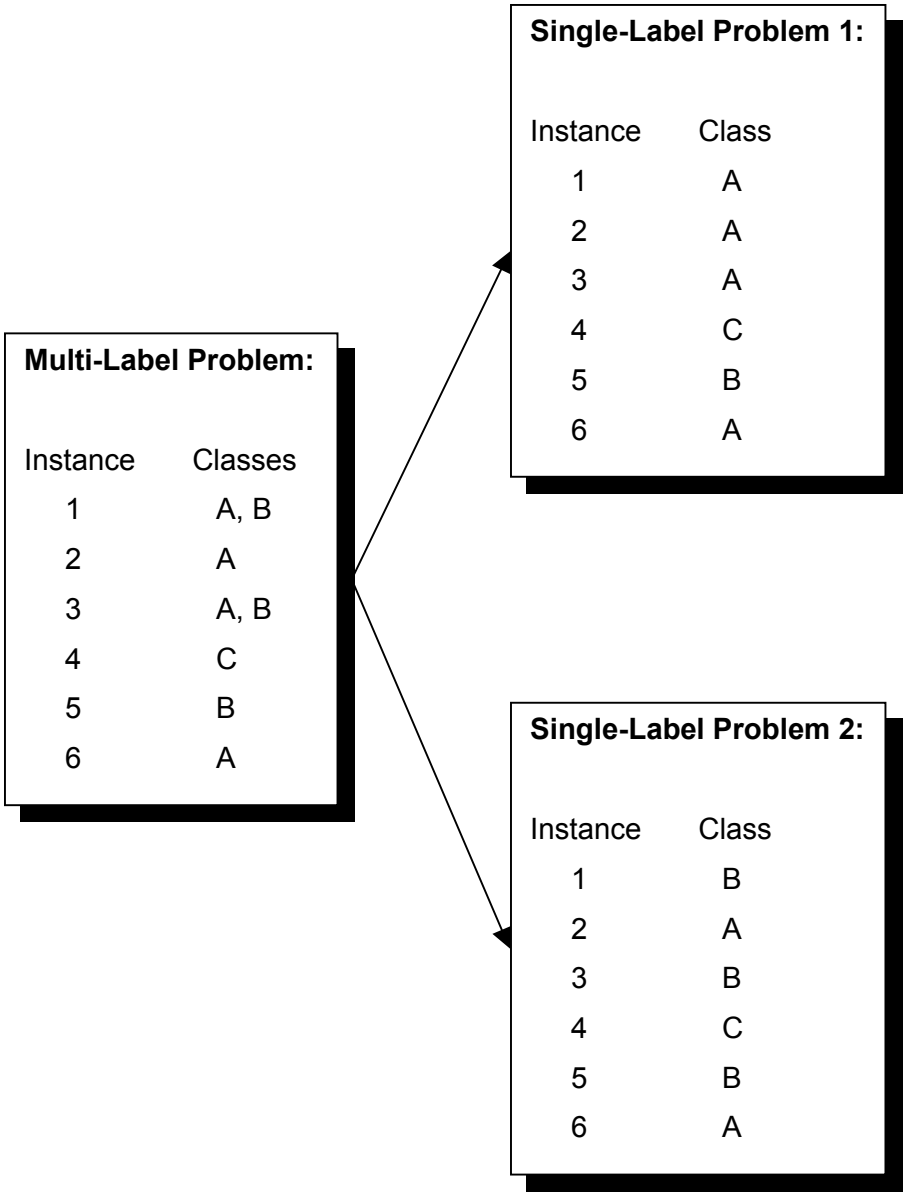

Fig. 6. Transform by decomposition of a multi-label problem into a set of single-label problems

For classifiers based on density estimation, this method may favor the multi-label instances [37]. The authors argue that the multi-label instances are likely to be closer to the decision boundaries, making the use of SVMs very suitable. They also say that if the proportion of multi-label samples is too high, a sampling technique can be employed to use a subset of them for each classifier.

The second decomposition method, here named multiplicative, is similar to another approach employed to divide multi-class problems into a set of binary problems, the one-against-one approach [23]. In this case, a combination of all the possible single-label classifiers is used.

The number of classifiers is given by $\prod l_i$, which is the product of the number of labels for each instance. Figure 7 illustrates this approach. In this case, the number

of classifiers would be equal to 2x1x2x1x1x1 = 4. This method is clearly not scalable, since the number of classifiers grows exponentially with the number of labels in the instances. It is easy to see that the previous additive method produces a subset of the single-label problems generated by this method. This method is reversible, allowing the restitution of the original multi-label problem.

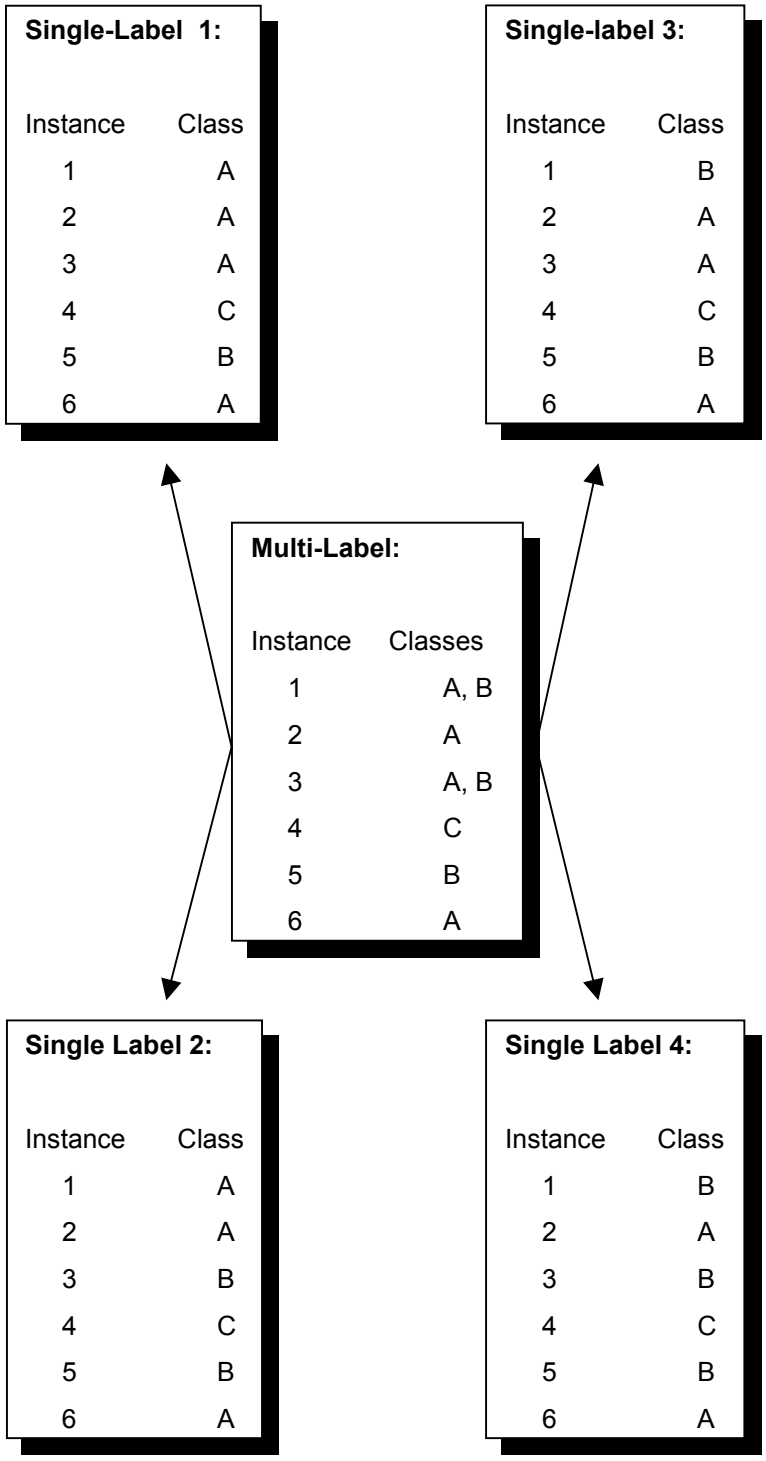

Fig. 7. Decomposition of a multi-label problem into a set of single-label problems according to the multiplicative method

Although the multiplicative decomposition method minimizes the deficiencies of those previous approaches where labels were combined or eliminated, the former, like the label creation method, does not take into account the interactions/correlations that can exist between the labels of a particular instance.

As seen in this section, different methods have been proposed in the literature for the algorithm independent approach. Table 1 summarizes the main characteristics of these methods.

Table 1. Summary of the algorithm-independent methods

Transformation Approach	Transformation Reversibility	Number of classifiers	Number of instances
Label-based	Yes	L	Same
Instance Elimination	No	Same	Reduced
Label Creation	Yes	Same	Same
Label Elimination	depends on the elimination criterion	Same	Same
Label Decomp. Add.	Yes	$\sum (l_i-1)$	Increased
Label Decomp. Mult.	Yes	$\prod l_i$	Increased

According to this table, where L represents the Number of labels and l_i the number of labels in the i^{th} instance, the methods differ, mainly, in the reversibility, number of classifiers used and size of the data set after the transformation.

In [20], the authors change the input instances, represented by feature vectors, in order to explore the co-occurrence of relationships among the classes. They do so by expanding a feature set, adding a new feature for each label. Next, the algorithm-dependent methods are introduced. In [19], dependencies between the different labels are explored through a collective approach. It does so by learning parameters for each possible pair of labels.

2.2 Algorithm Dependent Approach

As the name of this approach suggests, the methods following this approach have been proposed to specific algorithms. The advantage of this approach is that, by concentrating on a particular algorithm, the method may present a better performance in difficult real-world problems than the algorithm independent approaches.

Decision Trees

An extension of the alternating decision tree learning algorithm [17] for multi-label classification is also proposed in [12]. The alternating decision tree learning algorithm induces Alternating Decision Trees, a generalization of DTs. Its inductive principle is based on boosting. The proposed multi-label version is based on AdaBoost [16] and ADTBoost [17] This multi-class algorithm extends ADTs by

decomposing multi-class problems using the one-against-all approach.

In another work with DTs [11], the authors modify the C4.5 algorithm [32] for the classification of genes according to their function. A gene of the yeast *S. cerevisiae* may simultaneously belong to more than one class. Thus, this is a typical multi-label problem. The C4.5 algorithm uses a measure of entropy to define the tree nodes. This measure was originally defined for single-label problems. The authors modified the formulae in order to allow its use in multi-label problems. Another modification was the use of leaves of the tree to represent a set of class labels. When the leaf reached in the classification of an instance contains a set of classes, a separate rule is produced for each class. The authors claim that they could also have produced rules that predict a set of classes and improve the comprehensibility of the rules generated.

Support Vector Machines

Several of the recent works in multi-label classification employ Support Vector Machines (SVMs) [46]. SVMs are Large Margin Classifiers (LMC) [2] that minimize the ranking loss. LMC are ML techniques that place the decision frontier in a position that maximizes the distance between itself and the patterns belonging to each class.

The binary decomposition approach for multi-label problems has been partially studied in [33]. In this work, the authors investigate the use of SVMs for the multi-label classification of gene functional categories. The authors used a heterogeneous data set, generated by the combination of two data sets: gene expression data and phylogenetic profiles. According to the authors, this combination provided a more accurate picture of overlapping subsets of the gene functional classes. As a result, it leads to a better classification performance. They also observe that this improvement is not uniformly distributed among the different classes, thus the combination should only be tried if there is evidence of its benefits.

In [13], a similar method based on SVMs is proposed by the authors. In this paper, the authors also propose a new feature selection method for multi-label data sets. In another paper from the same authors, [14], they propose Rank-SVM, a linear model based on Kernel functions. As the name might suggest, this model follows the ranking approach and minimizes the ranking loss. For this model, the authors define a ranking system, which orders the labels according to their output value, and a predictor for the number of labels to be selected, named threshold-based method. This model is compared against a Binary-SVM model for multi-label classification and Boostexter using a bioinformatics data set, the Yeast data set. This data set contains the gene expression levels and phylogenetic profiles of selected genes. The target function is the prediction of the functional classes of a gene. In the experiments carried out, Rank-SVM outperformed the other two models.

One more method based on SVMs is proposed in [1]. When SVMs are employed for multi-label classification problems, the classification task is divided among several SVMs. The processing time is proportional to the number of kernel computations performed. The authors employed modified SVMs, which allows the simultaneous training of a set of SVM classifiers by using a single optimization procedure. In their approach, a single optimization procedure for the classifiers allows a shared use of the kernel matrix information among them. As a result, a

reduction in the learning complexity and training time are obtained, without loss in the classification performance. The performance of the proposed model was evaluated using a set of documents in a text mining task.

A set of SVMs was also adopted by [40], where a multiclass problem was decomposed into a set of binary problems using the one-against-all strategy. Experiments were performed using a data set of protein subcellular localization prediction. Kernel functions are also used in [31] [34] and [35].

Other Techniques

Zhang and Zhou propose in [51] a new multi-label learning algorithm based on K-NN, named ML-kNN. This model uses a lazy-learning approach. For each instance, the labels associated with the k-nearest neighbor instances are retrieved. A membership counting function is employed to count the number of neighbors associated with each label. The maximum a posteriori principle is used to define the label set for a new instance. The authors compare the performance of their algorithm against SVMs, ADTBoost.MH and BoosTexter. They use in the comparison the Hamming Loss, One-error, Coverage, Ranking Loss and Average Precision. In the experiments, they used the same Yeast gene functional data set used by [13]. In this data set, the maximum number of labels can be larger than 190. The results were very similar to those obtained by the other approaches.

Specific parametric mixture models are proposed by [44] [45] for multi-label and multi-class classification. The method was used for document classification using web pages. The experimental results were compared to several ML techniques, like Naive Bayes, K-NN and SVMs.

Two extensions of the Adaboost algorithm to enable their efficient use in multi-label problems are proposed and investigated in [38] [39]. The first extension is a modification of the evaluation of the prediction performance of the induced model by checking its ability to predict the correct set of labels for an input instance. The second extension changes the goal of the learning model to be the prediction of a ranking of labels for each input instance. The model is evaluated by its ability to correctly predict the high-ranking labels. These methods were evaluated using document classification data. Another work based on boosting was investigated in [49]. In this work, the author proposed an ensemble approach that is independent on the base classifier used. The proposed approach was applied to synthetic data and real multimedia data.

A multi-label learning algorithm based on class association rules is proposed in [41]. The algorithm, named multi-class multi-label associative classification (MMAC), is divided into three modules: rules generation, recursive learning and classification. Three measures for accuracy evaluation were also investigated in this work.

In [18], a maximal figure–of-merit (MfoM) learning algorithm initially proposed by three of the authors to binary classification is generalized for multi-label problems. The algorithm is experimentally compared with other ML algorithms in a text classification task.

In [22] the authors propose a new multi-label approach for dealing with data flows. For such, they use active learning in order to perform online multi label learning. Their approach is evaluated in a content-based video search application.

Finally, in [53], a classification algorithm based on entropy is used for information retrieval. In this work, the authors use their model to explore correlations among categories in multi-labelled documents.

3. Performance Evaluation

Finally, it is necessary to define how to evaluate the classification results. Different from single-label classification, where the classification of an instance is either correct or wrong, in multi-label tasks, the result can also be partially correct (or partially wrong). These would be the cases where the classifier correctly produces at least one of the correct labels but either misses one or more of the labels that should be assigned or includes one or more wrong labels in the list of assigned labels.

A few measures have been proposed and investigated to evaluate multi-label classifiers [25] [4] [37][41] [51]. The evaluation criteria can be based on either the multi-label classification made by a classifier, which uses the labels produced by the classifier for a given instance, or a ranking function, which uses the ranking position associated with each label by the classifier for a particular instance.

A similar division is proposed in [14], which divides the methods used to define the cost function into binary approach and ranking approach. In the binary approach, an output vector with the number of elements equal to the number of classes is used. Given an input vector, a sign function defines the value of each element of the output vector. Those elements with positive values are the labels for the input instance. Thus, a binary classifier can be used for each output element or class. Figure 8 illustrates this binary representation approach. It is interesting to notice that a neural network with three output nodes could be easily trained with a data set based on this representation.

Multi-Label Problem:		**Output vector:**		
Instance	Classes	A	B	C
1	A, B	1	1	0
2	A	1	0	0
3	A, B	1	1	0
4	C	0	0	1

Fig. 8. Binary representation for a multi-label problem

It is important to observe although a sign function is similar to a threshold function, being equal if the threshold value is zero, a heuristic can be followed to define the threshold value. For example, it can be adaptively defined or associated with the prior probability of the classes.

For classification-based evaluation, a common metric is the Hamming Loss. In the case of a binary encoding of the labels, like in Figure 8, the Hamming Loss measures the number of times a pair (instance, label) is misclassified. For such, it uses the average binary error. The smaller the Hamming loss, the better the

situation. The perfect situation occurs when its value is equal to 0.

In the ranking approach, it is assumed that the number of labels to be associated with the input instance, L, is previously known. When an input instance is presented to the multi-label classifier, the L labels with the highest output value are selected. An example of this system is the algorithm Boostexter [39].

For ranking-based evaluation, the metrics frequently employed in the literature are One-Error, Coverage and Average Precision. The One-Error measurement measures the number of times the label with the best rank computed by the classification algorithm is not in the set of correct labels of the input instance [37]. Another measurement, Coverage, says how far, on average, it is necessary to go down on the list of labels ordered by rank in order to include all the labels that should have been assigned to the input instance. The third method, Ranking Loss, calculates the average proportion of pairs that are not correctly ordered. The fourth metric, Average Precision, was originally proposed for Information Retrieval. It evaluates the average proportion of labels ranked above a particular desired label and that belong to the set of desired labels.

In [40], the performance was measured by two criteria, the prediction of the number of classes and the set of classes or labels associated with each test example.

4. Related Work and Discussion

Framework proposals for multi-label problems can be found in [4][43] and [52]. The first framework was presented by [4]. In this framework, the authors describe the initial training approaches to deal with multi-label classification and organize them into 4 major groups. They also discuss alternative testing criteria for the evaluation of multi-label classifiers and propose new evaluation metrics. The authors compared the different models and testing criteria using the evaluation measurements proposed in a scene classification problem. In [43], the authors also present experimental results comparing several multi-class methods.

Another work comparing different approaches for multi-label classification is presented in [27]. This paper includes a experimental comparison of six approaches using two data sets: bioinformatics and scene analysis. Several measures are used in this comparative work.

This paper advances the work in [4] [43] and [52] by proposing a new framework to categorize the methods proposed in the literature for multi-label classification and expanding the review of the current works in this area.

Several approaches for multi-label classification combine multi-label classification with hierarchical classification [11] [5] [6]. In hierarchical classification problems, the classes are disposed in a hierarchical structure. For this class of problems, the classes can be seen as nodes in either a tree-like or a direct acyclic graph (DAG) structure [15]. Several applications in text processing and bioinformatics combine these two issues [3][5] [6] [34] [35].

Recently, population-based meta-heuristics, like evolutionary computation [47] and ant colony optimization [10] have been used for multi-label classification problems.

Another promising work is the use of ranking with multi-label classification, where the classifier should predict not only the classes associated with an instance, but the order these classes are associated with the instance [7] [8] [9].

Finally, we believe that there will be a clear increase in the number of real multi-label classification problems and challenges, particularly in the area of bioinformatics, and this is therefore a promising research topic in Machine Learning.

Acknowledgement

The work was partly supported by the Brazilian Research Agencies CNPq and FAPESP.

References

1. Aiolli, F. & Sperduti, A. (2005) Multiclass Classification with Multi-Prototype Support Vector Machines. Journal of Machine Learning Research 6: 817-850
2. Barlett, P., Peter, B., Bartlett, J., Schölkopf, B., Schuurmans, D. & Smola, A. J. (2000) Advances in Large-Margin Classifiers. The MIT Press
3. Barutcuoglu, Z., Schapire, R. E. & Troyanskaya, O. G. (2006) Hierarchical multi-label prediction of gene function. Bioinformatics, 22:830-836, Oxford Press
4. Boutell, M., Shen, X., Luo, J., & Brown, C. (2003) Multi-label semantic scene classification. Technical Report, Department of Computer Science University of Rochester, USA
5. Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J. & Struyf, J. (2002) Hierarchical multiclassication, in: Proceedings of the ACM SIGKDD 2002 Workshop on Multi-Relational Data Mining (MRDM 2002), 2002, pp. 21-35, Edmonton, Canada
6. Blockeel, H., Schietgat, L., Struyf, J., Dzeroski, S. & Clare, A. (2006) Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics. 17th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD´06), pp. 18-29, Berlin, Germany.
7. Brinker, K., Fürnkranz, J. & Hüllermeier, E. (2006) A Unified Model for Multilabel Classification and Ranking. ECAI 2006, pp. 489-493
8. Brinker, K. & Hüllermeier, E. (2007) Case-Based Multilabel Ranking. IJCAI, pp. 702-707
9. Brinker, K. & Hüllermeier, E. (2007) Label Ranking in Case-Based Reasoning. ICCBR 2007, pp. 77-91
10. Chan A. & Freitas, A. A. (2006) A new ant colony algorithm for multi-label classification with applications in bioinfomatics. Genetic and Evolutionary Computation 2006 Conference (GECCO´06), pp. 27-34, Seattle, USA
11. Clare, A. J. & King, R. D. (2001) Knowledge discovery in multi-label phenotype data. In: The 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01) (Eds. L. De Raedt, A. Siebes) Lecture Notes in Artificial Intelligence, LNAI 2168 Sringer-Verlag, Heidelberg.
12. de Comite, F., Gilleron, R. & and Tommasi, M. (2003) Learning Multi-label Alternating Decision Trees from Texts and Data. Proceedings of the MLDM 2003, Lecture Notes of Computer Science, 2734, Springer-Verlag, pp 251-274

13. Elisseeff, A. & and Weston, J. (2001) Kernel methods for multi-labelled classification and categorical regression problems. Technical Report. BIOwulf Technologies
14. Elisseeff, A. & and Weston, J. (2001) A kernel method for multi-labelled classification. Neural Information processing Systems, NIPS 14, 2001.
15. Freitas A. A. & de Carvalho A. C. P. L. F. (2007) A Tutorial on Hierarchical Classification with Applications in Bioinformatics. In Research and Trends in Data Mining Technologies and Applications. Edited by: Taniar D. Idea Group; 2007:175-208.
16. Freund Y. & Schapire, R. E. (1995) A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory: Second European Conference, EuroCOLT '95, pp. 23–37, Springer-Verlag
17. Freund Y. & Mason, L. (1999) The alternating decision tree learning algorithm. In: Proceedings of the Sixteenth International Conference on Machine Learning, ICML, pp. 124-133
18. Gao, S. Wu, W., Lee C.-H. & Chua, T.-S. (2004) An MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization. Proceedings of the International Conference on Machine Learning (ICML '04), pp. 329-336, Banff, Canada
19. Ghamrawi, N. & McCallum, A. (2005) Collective Multi-Label Classification. Proceedings of the Fourteenth Conference on Information and Knowledge Management (CIKM), pp. 195-200
20. Godbole, S. & Sarawagi, S. (2004) Discriminative Methods for Multi-labeled Classification. Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD 2004, Lecture Notes in Computer Science 3056, pp. 22-30, Sydney, Australia
21. Gonçalves, T. & Quaresma, P. (2003) A preliminary approach to the multi-label classification problem of Portuguese juridical documents. 11th Portuguese Conference on Artificial Intelligence, Lecture Notes in Computer Science, LNAI 2902, pp. 435-444, Beja, Portugal
22. Hua, X. & Qi, G. (2008) Online multi-label active annotation: towards large-scale content-based video search. In Proceeding of the 16th ACM international Conference on Multimedia (Vancouver, British Columbia, Canada, October 26 - 31, 2008). MM '08. ACM, New York, NY, pp. 141-150
23. Hsu C.-W. & Lin C.-J. (2002) A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks, 13(2):415–425
24. Joachims, T. (1998) Text categorization with support vector machines: learning with many relevant features. In C. Nedellec and C. Rouveirol, editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, LNCS 1398, pages 137--142, Chemnitz, Germany, Springer Verlag
25. Karalič, A. & Pirnat, V. (1991) Significance level based multiple tree classification. Informatica, Vol. 15, No. 5, 12 pages
26. Lauser, B. & Hotho, A. (2003) Automatic multi-label subject indexing in a multilingual environment. In Proceedings of the 7th European Conference in Research and Advanced Technology for Digital Libraries, ECDL 2003, pp. 140-151, Ed. T. Koch and I. Solvberg, Lecture Notes of Artificial Intelligence, LNCS 2769
27. Li, T., Zhang, C. & Zhu, S. (2006) Empirical Studies on Multi-label Classification. In Proceedings of the 18th IEEE international Conference on

Tools with Artificial intelligence (November 13 - 15, 2006). ICTAI. IEEE Computer Society, Washington, DC, pp. 86-92
28. Luo, X. & Zincir-Heywood, A. N. (2005) Evaluation of Two Systems on Multi-class Multi-label Document Classification. Proceedings of the 15th ISMIS'2005, Springer Verlag LNAI 3488, pp. 161-169, New York, USA
29. McDonald, R., Crammer, K. Pereira, F. (2005) Flexible Text Segmentation with Structured Multilabel Classification. Proceedings of the Human Language Technology Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP, 2005), Vancouver, Canada
30. McCallum, A. (1999) Multi-label text classification with a mixture model trained by EM. AAAI'99 Workshop on Text Learning
31. Micchelli, C. A. & Pontil, M. (2005) Kernels for Multi--task Learning. Advances in Neural Information Processing Systems 17 (NIPS'04), pp. 921-928. Ed. L. K. Saul, Y. Weiss and L. Bottou. MIT Press, Cambridge, USA
32. Quinlan, J. R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
33. Pavlidis, P., Weston, J., Cai, J. & Grundy, W. (2001) Combining microarray expression data and phylogenetic profiles to learn functional categories using support vector machines, RECOMB, pp. 242-248
34. Rousu, J., Saunders, C., Szedmak, S. & Shawe-Taylor, J. (2005) Learning Hierarchical Multi-Category Text Classification Models. 22nd International Conference on Machine Learning (ICML'2005), pp. 745-752, Bonn, Germany
35. Rousu, J., Saunders, C., Szedmak, S. & Shawe-Taylor, J. (2006) Kernel-based Learning of Hierarchical Multilabel Classification Models Journal of Machine Learning Research Vol. 7, pp. 1601-1626
36. Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1-47
37. Shen, X., Boutell, M., Luo, J. & aBrown, C. (2003) Multi-label machine learning and its application to semantic scene classification. Storage and Retrieval Methods and Applications for Multimedia. Edited by Yeung, Minerva M.; Lienhart, Rainer W.; Li, Chung-Sheng. Proceedings of the SPIE, Volume 5307, pp. 188-199. 2003.
38. Schapire, R. E. & Singer, Y. (1999) Improved boosting algorithms using confidence-rated predictions. *Machine Learning,* Vol. 37 No. 3, pp. 297-336
39. Schapire, R. E. & Singer, Y. (2000) BoosTexter: A boosting-based system for text categorization. Machine Learning, Vol. 39, No. 2/3 pp. 135-168, 2000.
40. C.-Y. Su, C.-Y., Lo, A., Lin, C.-C., Chang, F. & Hsu W.-L. (2005) A Novel Approach for Prediction of Multi-Labeled Protein Subcellular Localization for Prokaryotic Bacteria. Computational Systems Bioinformatics Conference, CSB Workshops, pp. 79-82, Palo Alto, USA
41. Thabtah, F. A., Cowling, P. & Peng, Y. (2004) MMAC: A New Multi-Class, Multi-Label Associative Classification Approach, pp. 217-224, Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK
42. Tikk, D. & Biró, G. (2003) Experiments with multi-label text classifier on the Reuters collection. Proc. of the International Conference on Computational Cybernetics (ICCC 03), pp. 33–38, Siófok, Hungary
43. Tsoumakas, G. & Katakis, I. (2007) Multi-Label Classification: An Overview, International Journal of Data Warehousing and Mining, Vol. 3, No. 3, pp. 1-13
44. Ueda, N. & Saito, K. (2002) Parametric mixture models for multi-topic text,

Neural Information Processing Systems 15 (NIPS15), MIT Press, pp. 737-744
45. Ueda, N. & Saito, K. (2002) Single-shot detection of multi-category text using parametric mixture models. ACM SIG Knowledge Discovery and Data Mining (SIGKDD'2002), pp. 626-631
46. Vallim, R. M. M., Goldberg, D. E., Llorà, X., Duque, T.S.P.C. & A.C.P.L.F. Carvalho, A.C.P.L.F. (2008) A New Approach for Multi-label Classification Based on Default Hierarchies and Organizational Learning, IWLCS, Accepted for the 11th International Workshop on Learning Classifier Systems, part of the Genetic and Evolutionary Computation 2008 Conference (GECCO´08), Atlanta, Georgia, USA
47. Vapnik, V. N. (1995) The Nature of Statistical Learning Theory. Springer-Verlag
48. Xu, Y.-Y. Zhou, X.-Z. & Guo, Z.-W. (2002) Weak learning algorithm for multi-label multiclass text categorization, *International Conference on Machine Learning and Cybernetics, 2002. Proceedings.*vol.2, no., pp. 890-894
49. Yan, R., Tesic, J. & Smith, J. R. (2007) Model-shared subspace boosting for multi-label classification. In Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12 - 15, 2007). KDD '07. ACM, New York, NY, 834-843
50. Yu, K.Yu, S. & Tresp V. (2005) Multi-label informed latent semantic indexing Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 258-265
51. Zhang, M.-L. & Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. Proceedings of the 1st IEEE International Conference on Granular Computing (GrC'05), Beijing, China, 2005, pp.718-721
52. Zhou, Z. (2007) Mining Ambiguous Data with Multi-instance Multi-label Representation. In *Proceedings of the 3rd international Conference on Advanced Data Mining and Applications* (Harbin, China, August 06 - 08, 2007). R. Alhajj, H. Gao, X. Li, J. Li, and O. R. Zaïane, Eds. Lecture Notes In Artificial Intelligence, vol. 4632. Springer-Verlag, Berlin, Heidelberg, 1-1
53. Zhu, B. & Poon, C. K. (1999) Efficient Approximation Algorithms for Multi-label Map Labeling. Proc. 10th Int. Symp. Algorithms & Computation (ISAAC 1999), Lecture Notes in Computer Science, LNCS 1741, A. Aggarwal and C. Pandu Rangan, ed., Springer-Verlag, pp. 143–152
54. Zhu, S., Ji, X., Xu, W. & Gong, Y. (2005) Multi-labeled Classification Using Maximum Entropy Method. In Proceedings of Annual ACM Conference on Research and Development in Information Retrieval (SIGIR'05), pp. 274-281, Salvador, Brazil