Multi-Label Classification: An Overview

Grigorios Tsoumakas, Ioannis Katakis

Dept. of Informatics, Aristotle University of Thessaloniki, 54124, Greece

ABSTRACT

Nowadays, multi-label classification methods are increasingly required by modern applications, such as protein function classification, music categorization and semantic scene classification. This paper introduces the task of multi-label classification, organizes the sparse related literature into a structured presentation and performs comparative experimental results of certain multi-label classification methods. It also contributes the definition of concepts for the quantification of the multi-label nature of a data set.

INTRODUCTION

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

In *multi-label* classification, the examples are associated with a set of labels $Y \subseteq L$. In the past, multi-label classification was mainly motivated by the tasks of text categorization and medical diagnosis. Text documents usually belong to more than one conceptual class. For example, a newspaper article concerning the reactions of the Christian church to the release of the *Da Vinci Code* film can be classified into both of the categories *Society\Religion* and *Arts\Movies*. Similarly in medical diagnosis, a patient may be suffering for example from diabetes and prostate cancer at the same time.

Nowadays, we notice that multi-label classification methods are increasingly required by modern applications, such as protein function classification (Zhang & Zincir-Heywood, 2005), music

categorization (Li & Ogihara, 2003) and semantic scene classification (Boutell et al., 2004). In semantic scene classification, a photograph can belong to more than one conceptual class, such as *sunsets* and *beaches* at the same time. Similarly, in music categorization a song may belong to more than one genre. For example, several hit songs of the popular rock band *Scorpions* can be characterized as both *rock* and *ballad*.

This paper aims to serve as a starting point and reference for researchers interested in multi-label classification. The main contributions are: a) a structured presentation of the sparse literature on multi-label classification methods with comments on their relative strengths and weaknesses and when possible the abstraction of specific methods to more general and thus more useful schemata, b) the introduction of an undocumented multi-label method, c) the definition of a concept for the quantification of the multi-label nature of a data set, d) preliminary comparative experimental results about the performance of certain multi-label methods.

The rest of the paper is organized as follows. The next section discusses tasks that are related to multi-label classification. Subsequently follows an organized presentation of multi-label classification methods. The next section introduces the concept of label density and presents the metrics that have been proposed in the past for the evaluation of multi-label classifiers. The following section presents the comparative experiments and discusses the results, while the concluding section summarizes this work and points to future research directions.

RELATED TASKS

A task that also belongs to the general family of supervised learning and is very relevant to multi-label classification is that of *ranking*. In ranking the task is to order a set of labels L, so that the topmost labels are more related with the new instance. There exist a number of multi-label classification methods that learn a ranking function from multi-label data. However, a ranking of labels requires post-processing in order to give a set of labels, which is the proper output of a multi-label classifier.

In certain classification problems the labels belong to a *hierarchical structure*. The *dmoz* open directory for example (http://www.dmoz.org/), maintains a hierarchy of conceptual classes for the categorization of web pages. A web page may be labelled using one or more of those classes, which can belong to different levels of the hierarchy. The top level of the MIPS (Munich Information Centre for Protein Sequences) hierarchy (http://mips.gsf.de/) consists of classes such as: *Metabolism*, *Energy*, *Transcription* and *Protein Synthesis*. Each of these classes is then subdivided into more specific classes, and these are in turn subdivided, and then again subdivided, so the hierarchy is up to 4 levels deep (Clare & King, 2001). When the labels in a data set belong to a hierarchical structure then we call the task *hierarchical classification*. If each example is labelled with more than one node of the hierarchical structure, then the task is called *hierarchical multi-label classification*. In this paper we focus on flat (non-hierarchical) multi-label classification methods.

Jin and Ghahramani (2002) call *multiple-label problems*, the semi-supervised classification problems where each example is associated with more than one classes, but only one of those classes is the true class of the example. This task is not that common in real-world applications as the one we are studying.

Multiple-instance learning is a variation of supervised learning, where the task is to learn a concept given positive and negative bags of instances (Maron & Lozano-Perez, 1997). Each bag may contain many instances, but a bag is labelled positive even if only one of the instances in it falls within the concept. A bag is labelled negative only if all the instances in it are negative.

MULTI-LABEL CLASSIFICATION METHODS

We can group the existing methods for multi-label classification into two main categories: a) *problem transformation methods*, and b) *algorithm adaptation methods*. We call problem transformation methods, those methods that transform the multi-label classification problem either into one or more single-label classification or regression problems, for both of which there exists a huge bibliography

of learning algorithms. We call algorithm adaptation methods, those methods that extend specific learning algorithms in order to handle multi-label data directly.

Problem transformation methods

To exemplify these methods we will use the data set of Table 1. It consists of four examples (documents in this case) that belong to one or more of four classes: *Sports*, *Religion*, *Science*, and *Politics*.

Table 1: Example of a multi-label data set

Ex.	Sports	Religion	Science	Politics
1	X			X
2			X	X
3	X			
4		X	X	

There exist two straightforward problem transformation methods that force the learning problem into traditional single-label classification (Boutell et al., 2004). The first one (dubbed PT1) subjectively or randomly selects one of the multiple labels of each multi-label instance and discards the rest, while the second one (dubbed PT2) simply discards every multi-label instance from the multi-label data set. Table 2 and Table 3 show the transformed data set using methods PT1 and PT2 respectively. These two problem transformation methods discard a lot of the information content of the original multi-label data set and are therefore not considered further in this work.

Table 2: Transformed data set using PT1

Ex.	Sports	Religion	Science	Politics
1	X			
2				X
3	X			
4		X		

Table 3: Transformed data set using PT2

Ex.	Sports	Religion	Science	Politics
3	X			

The third problem transformation method that we will mention (dubbed PT3), considers each different set of labels that exist in the multi-label data set as a single label. It so learns one single-label classifier $H: X \rightarrow \mathrm{P}(L)$, where P(L) is the power set of L. Table 4 shows the result of transforming the data set of Table 1 using this method. One of the negative aspects of PT3 is that it may lead to data sets with a large number of classes and few examples per class. PT3 has been used in the past in (Boutell et al., 2004; Diplaris et al., 2005).

Table 4: Transformed data set using PT3

Ex.	Sports	(Sports ∧ Politics)	(Science ∧ Politics)	(Science ∧ Religion)
1		X		
2			X	
3	X			
4				X

The most common problem transformation method (dubbed PT4) learns $|L|$ binary classifiers $H_l: X \rightarrow \{l, \neg l\}$, one for each different label l in L. It transforms the original data set into $|L|$ data sets D_l that contain all examples of the original data set, labelled as l if the labels of the original example contained l and as $\neg l$ otherwise. It is the same solution used in order to deal with a single-label multi-class problem using a binary classifier.

For the classification of a new instance x this method outputs as a set of labels the union of the labels that are output by the $|L|$ classifiers:

$$H_{PT4}(x) = \bigcup_{l \in L} \{l\} : H_l(x) = l$$

Figure 1 shows the four data sets that are constructed by PT4 when applied to the data set of Table 1. PT4 has been used in the past in (Boutell et al., 2004; Goncalves & Quaresma, 2003; Lauser & Hotho, 2003; Li & Ogihara, 2003).

Figure 1: The four data sets that are constructed by PT4

Ex.	Sports	¬Sports
1	X	
2		X
3	X	
4		X

(a)

Ex.	Politics	¬Politics
1	X	
2	X	
3		X
4		X

(b)

Ex.	Religion	¬Religion
1	X	
2	X	
3		X
4		X

(c)

Ex.	Science	¬Science
1		X
2	X	
3		X
4	X	

(d)

A straightforward, yet undocumented, problem transformation method is the following (dubbed PT5): Firstly, it decomposes each example (x, Y) into $|Y|$ examples (x, l) for all $l \in Y$. Then it learns one single-label *coverage-based* classifier from the transformed data set. Distribution classifiers are those classifiers that can output a distribution of certainty degrees (or probabilities) for all labels in L. Finally it post-processes this distribution to output a set of labels. One simple way to achieve this is to output those labels for which the certainty degree is greater than a specific threshold (e.g. 0.5). A more complex way is to output those labels for which the certainty degree is greater than a percentage (e.g. 70%) of the highest certainty degree. Table 5 shows the result of transforming the data set of Table 1 using this method.

Table 5: Transformed data set using PT5

Ex.	Class
1	Sports
1	Politics
2	Science
2	Politics
3	Sports
4	Religion
4	Science

Algorithm adaptation methods

Clare and King (2001) adapted the C4.5 algorithm for multi-label data. They modified the formula of entropy calculation as follows:

$$entropy(S) = -\sum_{i=1}^{N} \left(p(c_i) \log p(c_i) + q(c_i) \log q(c_i) \right)$$

where $p(c_i)$ = relative frequency of class c_i and $q(c_i) = 1 - p(c_i)$. They also allowed multiple labels in the leaves of the tree.

Adaboost.MH and Adaboost.MR (Schapire & Singer, 2000) are two extensions of AdaBoost (Freund & Schapire, 1997) for multi-label classification. They both apply AdaBoost on weak classifiers of the form $H: X \times L \rightarrow R$. In AdaBoost.MH if the sign of the output of the weak classifiers is positive for a new example x and a label l then we consider that this example can be labelled with l, while if it's negative then this example is not labelled with l. In AdaBoost.MR the output of the weak classifiers is considered for ranking each of the labels in L.

Although these two algorithms are adaptations of a specific learning approach, we notice that at their core, they actually use a problem transformation (dubbed PT6): Each example (x, Y) is decomposed into $|L|$ examples (x, l, $Y[l]$), for all $l \in L$, where $Y[l] = 1$ if $l \in Y$, and $[l] = -1$ otherwise. Table 6 shows the result of transforming the data set of Table 1 using this method.

ML-*k*NN (Zhang & Zhou, 2005) is an adaptation of the *k*NN lazy learning algorithm for multi-label data. Actually this method follows the paradigm of PT4. In essence, ML-*k*NN uses the *k*NN algorithm independently for each label l: It finds the k nearest examples to the test instance and considers those that are labelled at least with l as positive and the rest as negative. What mainly differentiates this method from the application of the original *k*NN algorithm to the transformed problem using PT4 is the use of prior probabilities. ML-*k*NN has also the capability of producing a ranking of the labels as an output.

Table 6: Transformed data set using PT6

Ex.	*l*	*Y*[*l*]
1	Sports	1
1	Religion	-1
1	Science	-1
1	Politics	1
2	Sports	-1
2	Religion	-1
2	Science	1
2	Politics	1
3	Sports	1
3	Religion	-1
3	Science	-1
3	Politics	-1
4	Sports	-1
4	Religion	1
4	Science	1
4	Politics	-1

Luo and Zincir-Heywood (2005) present two systems for multi-label document classification, which are also based on the *k*NN classifier. The main contribution of their work is on the pre-processing stage for the effective representation of documents. For the classification of a new instance, the systems initially find the *k* nearest examples. Then for every appearance of each label in each of these examples, they increase a corresponding counter for that label. Finally they output the *N* labels with the largest counts. *N* is chosen based on the number of labels of the instance. This is an inappropriate strategy for real-world use, where the number of labels of a new instance is unknown.

McCallum (1999) defines a probabilistic generative model according to which, each label generates different words. Based on this model a multi-label document is produced by a mixture of the word distributions of its labels. The parameters of the model are learned by maximum a posteriori estimation from labelled training documents, using Expectation Maximization to calculate which labels were both the mixture weights and the word distributions for each label. Given a new document the label set that is most likely is selected with Bayes rule. This approach for the classification of a

new document actually follows the paradigm of PT3, where each different set of labels is considered independently as a new class.

Elisseeff and Weston (2002) present a ranking algorithm for multi-label classification. Their algorithm follows the philosophy of SVMs: it is a linear model that tries to minimize a cost function while maintaining a large margin. The cost function they use is ranking loss, which is defined as the average fraction of pairs of labels that are ordered incorrectly. However, as stated earlier, the disadvantage of a ranking algorithm is that it does not output a set of labels.

Godbole and Sarawagi (2004) present two improvements for the Support Vector Machine (SVM) classifier in conjunction with the PT4 method for multi-label classification. The first improvement could easily be abstracted in order to be used with any classification algorithm and could thus be considered an extension to PT4. The main idea is to extend the original data set with $|L|$ extra features containing the predictions of each binary classifier. Then a second round of training $|L|$ new binary classifiers takes place, this time using the extended data sets. For the classification of a new example, the binary classifiers of the first round are initially used and their output is appended to the features of the example to form a meta-example. This meta-example is then classified by the binary classifiers of the second round. Through this extension the approach takes into consideration the potential dependencies among the different labels. Note here that this improvement is actually a specialized case of applying Stacking (Wolpert, 1992) (a method for the combination of multiple classifiers) on top of PT4.

The second improvement of (Godbole & Sarawagi, 2004) is SVM-specific and concerns the margin of SVMs in multi-label classification problems. They improve the margin by a) removing very similar negative training instances which are within a threshold distance from the learnt hyperplane, and b) removing negative training instances of a complete class if it is very similar to the positive class, based on a confusion matrix that is estimated using any fast and moderately accurate classifier on a

held out validation set. Note here that the second approach for margin improvement is actually SVM independent. Therefore, it could also be used as an extension to PT4.

MMAC (Thabtah, Cowling & Peng, 2004) is an algorithm that follows the paradigm of *associative classification*, which deals with the construction of classification rule sets using association rule mining. MMAC learns an initial set of classification rules through association rule mining, removes the examples associated with this rule set and recursively learns a new rule set from the remaining examples until no further frequent items are left. These multiple rule sets might contain rules with similar preconditions but different labels on the right hand side. Such rules are merged into a single multi-label rule. The labels are ranked according to the support of the corresponding individual rules.

ISSUES

How much multi-label is a data set?

Not all data sets are equally multi-label. In some applications the number of labels of each example is small compared to $|L|$, while in others it is large. This could be a parameter that influences the performance of the different multi-label methods. We here introduce the concepts of *label cardinality* and *label density* of a data set. Let D be a multi-label data set consisting of $|D|$ multi-label examples (x_i, Y_i), $i = 1..|D|$.

Definition 1: Label cardinality of D is the average number of labels of the examples in D:

$$\mathrm{LC}(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}|Y_i|$$

Definition 2: Label density of D is the average number of labels of the examples in D divided by $|L|$:

$$\mathrm{LD}(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i|}{|L|}$$

Label cardinality is independent of the number of labels $|L|$ in the classification problem, and is used to quantify the number of alternative labels that characterize the examples of a multi-label training data set. Label density takes into consideration the number of labels in the classification problem. Two data sets with the same label cardinality but with a great difference in the number of labels (different label density) might not exhibit the same properties and cause different behaviour to the multi-label classification methods. The two metrics are related to each other: $\mathrm{LC}(D) = |L|\ \mathrm{LD}(D)$.

Evaluation metrics

Multi-label classification requires different metrics than those used in traditional single-label classification. This section presents the various metrics that have been proposed in the literature. Let D be a multi-label evaluation data set, consisting of $|D|$ multi-label examples (x_i, Y_i), $i = 1..|D|$, $Y_i \subseteq L$. Let H be a multi-label classifier and $Z_i = H(x_i)$ be the set of labels predicted by H for example x_i.

Schapire and Singer (2000) consider the *Hamming Loss*, which is defined as:

$$\mathrm{HammingLoss}(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|}$$

Where Δ stands for the symmetric difference of two sets and corresponds to the XOR operation in Boolean logic.

The following metrics are used in (Godbole & Sarawagi, 2004) for the evaluation of H on D:

$$\mathrm{Accuracy}(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

$$\mathrm{Precision}(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Z_i|}$$

$$\mathrm{Recall}(H, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i|}$$

Boutell et al. (2004) give a more generalized version of the above accuracy using a parameter $\alpha \geq 0$, called forgiveness rate:

$$\text{Accuracy}(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \left(\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \right)^{\alpha}$$

This parameter is used in order to control the forgiveness of errors that are made in predicting labels. They also give an even more generalized version of the accuracy by introducing two additional parameters in order to allow different costs for false positives and true negatives. These two general measures of accuracy are too complex, due to the additional parameters, but could be useful in certain applications.

EXPERIMENTAL COMPARISON OF PT METHODS

We implemented the PT3, PT4 and PT6 methods in Java, within the framework of the WEKA (Witten & Frank, 1998) library of machine learning algorithms, and made the software publicly available at the following URL (mlkd.csd.auth.gr/multilabel.html). We experimented with the three PT methods in conjunction with the following classifier learning algorithms: kNN (Aha, Kibler & Albert), C4.5 (Quinlan, 1993), Naive Bayes (John & Langley, 1995) and SMO (Platt, 1998). For performance evaluation, we used the HammingLoss, Accuracy, Precision and Recall metrics that were presented in the previous section.

We experimented on the following multi-label data sets: *genbase* (Diplaris et al., 2005) and *yeast* (Elisseeff & Weston, 2002) are biological data sets that are concerned with protein function classification and gene function classification respectively. The *scene* data set (Boutell et al., 2004) contains data related to a scene classification problem. These data sets were retrieved from the site of the Support Vector Classification library LIBSVM (Chang & Lin, 2001), and transformed to a specific format that is suitable for our software, based on the ARFF file format of the WEKA library. The transformed data sets are also available at the aforementioned URL.

The details of the data sets, such as the number of examples, the number of numeric and discrete attributes the number of classes and their label density are given in Table 7. We notice that *genbase* (LD=0.05) and *scene* (LD=0.18) are quite sparse multi-label data sets with less than 1.5 labels per example on average. The *yeast* dataset on the other hand is denser (LD=0.30) with more than 4 labels per example on average.

Table 7: Examples, numeric and discrete attributes, labels and LD of datasets

Data set	Examples		Attributes		Labels	Label Density	Label Cardinality
	Train	Test	Numeric	Discrete			
genbase	463	199	0	1185	27	0.05	1.35
yeast	1500	917	103	0	14	0.30	4.25
scene	1211	1196	294	0	6	0.18	1.08

Table 8 presents analytical results on the three data sets. We will first discuss the results in terms of accuracy. The combination of the PT3 method together with the SMO learning algorithm gives the best results in each of the three data sets. In addition the PT3 method has the highest mean accuracy for all learning algorithms in each of the three data sets, followed by PT4 and then by PT6. This means that it is the best method independently of learning algorithm in each of the three data sets. This is an interesting result, given that the PT3 method is not so popular in the literature compared to PT4.

We will now discuss the results in terms of Hamming loss. In *genbase* the best results are obtained with PT4 in combination with either *k*NN or SMO. In *yeast* the best results are obtained again with PT4 in combination with SMO, while in *scene* the best results are obtained with PT3 in conjunction with SMO. Independently of the algorithm used, PT3 is the best method in *scene*, PT4 in *genbase* and PT6 in *yeast*.

One noteworthy result is that PT6 does not perform well in combination with SMO for the *scene* and *genbase* data sets. Note that these two data sets are quite sparse as LD(*scene*)=0.18 and LD(*genbase*)=0.05. This means that after the transformation, the class attribute will have a large

number of examples with a value of -1. It seems that in these cases SMO learns to predict always -1. This leads to zero accuracy, precision and recall, while Hamming loss becomes equal to the label density of the data set.

Table 8: Resuls on the three data sets

genbase												
	PT3				**PT4**				**PT6**			
Metric	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**
HamLoss	0,004	0,046	0,057	0,001	0,000	0,001	0,035	0,000	0,025	0,002	0,103	0,046
Accuracy	0,964	0,984	0,340	0,993	0,989	0,987	0,273	0,991	0,543	0,984	0,019	0,000
Recall	0,990	0,995	0,347	1,000	0,997	0,995	0,276	0,997	0,548	0,994	0,020	0,000
Precision	0,964	0,984	0,340	0,993	0,992	0,992	0,273	0,993	0,543	0,990	0,117	0,000
yeast												
	PT3				**PT4**				**PT6**			
Metric	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**
HamLoss	0,229	0,286	0,243	0,206	0,243	0,259	0,301	0,200	0,208	0,259	0,261	0,233
Accuracy	0,495	0,399	0,464	0,530	0,479	0,423	0,421	0,502	0,514	0,423	0,329	0,337
Recall	0,628	0,528	0,608	0,672	0,601	0,593	0,531	0,711	0,665	0,593	0,604	0,748
Precision	0,596	0,529	0,575	0,615	0,596	0,561	0,610	0,579	0,623	0,561	0,407	0,337
scene												
	PT3				**PT4**				**PT6**			
Metric	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**	***k*NN**	**C4.5**	**NB**	**SMO**
HamLoss	0,113	0,148	0,139	0,100	0,125	0,139	0,247	0,114	0,147	0,139	0,357	0,181
Accuracy	0,668	0,572	0,603	0,704	0,637	0,513	0,435	0,571	0,242	0,513	0,076	0,000
Recall	0,703	0,598	0,631	0,737	0,669	0,534	0,443	0,596	0,253	0,534	0,078	0,000
Precision	0,668	0,584	0,633	0,713	0,651	0,611	0,816	0,628	0,243	0,611	0,317	0,000

CONCLUSIONS AND FUTURE WORK

This work was involved with the task of multi-label classification: It introduced the problem, gave an organized presentation of the methods that exist in the literature and provided comparative experimental results for some of these methods. To the best of our knowledge, there is no other review paper on the interesting and upcoming task of multi-label classification.

In the future we intend to perform a finer-grained categorization of the different multi-label classification methods and perform more extensive experiments with more data sets and methods. We also intend to perform a comparative experimental study of problem adaptation methods.

REFERENCES

Aha, D.W., Kibler, D., & Albert, M.K. (1991), 'Instance-based learning algorithms', Machine Learning, vol. 6, no. 1, pp. 37-66 .

Boutell, M.R., Luo, J., Shen, X. & Brown, C.M. (2004), 'Learning multi-label scene classification', Pattern Recognition, vol. 37, no. 9, pp. 1757-71.

Chang, C.-C., & Lin, C.-J. (2004), 'LIBSVM : a library for support vector machines', Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Clare, A. & King, R.D. (2001), 'Knowledge Discovery in Multi-Label Phenotype Data', paper presented to Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany.

Diplaris, S., Tsoumakas, G., Mitkas, P. & Vlahavas, I. (2005), 'Protein Classification with Multiple Algorithms', paper presented to Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece, November.

Elisseeff, A. & Weston, J. (2002), 'A kernel method for multi-labelled classification', paper presented to Advances in Neural Information Processing Systems 14.

Freund, Y. & Schapire, R.E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-39.

Godbole, S. & Sarawagi, S. (2004), 'Discriminative Methods for Multi-labeled Classification', paper presented to Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004).

Goncalves, T. & Quaresma, P. (2003), 'A Preliminary Approach to the Multilabel Classification Problem of Portuguese Juridical Documents', paper presented to Proceedings of the 11th Portuguese Conference on Artificial Intelligence (EPIA '03).

Jin, R. & Ghahramani, Z. (2002), 'Learning with Multiple Labels', paper presented to Proceedings of Neural Information Processing Systems 2002 (NIPS 2002), Vancoucer, Canada.

John, G. & Langley, P. (1995), 'Estimating continuous distributions in bayesian classifiers', paper presented to Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Vancoucer, Canada

Lauser, B. & Hotho, A. (2003), 'Automatic multi-label subject indexing in a multilingual environment', paper presented to Proceedings of the 7th European Conference in Research and Advanced Technology for Digital Libraries (ECDL 2003).

Lewis, D.D., Tony, Y. Y., Rose, G. & Li, F. (2004). 'RCV1: A new benchmark collection for text categorization research', Journal of Machine Learning Research, Vol 5, pp 361-397.

Li, T. & Ogihara, M. (2003), 'Detecting emotion in music', paper presented to Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA.

Luo, X. & Zincir-Heywood, A.N. (2005), 'Evaluation of Two Systems on Multi-class Multi-label Document Classification', paper presented to Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems.

Maron, O. & Lozano-Perez, T. (1997), 'A framework for Multiple-Instance learning', paper presented to Proceedings of Neural Information Processing Systems 1997 (NIPS 1997).

McCallum, A. (1999), 'Multi-label text classification with a mixture model trained by EM', paper presented to Proceedings of the AAAI' 99 Workshop on Text Learning.

Platt, J. (1998), 'Fast training of support vector machines using sequential minimal optimization', In B. Scholkopf, B., Burges, C., & Smola, A., Advances in Kernel Methods - Support Vector Learning, MIT Press.

Quinlan, J.R. (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann.

Schapire, R.E. & Singer, Y. (2000), 'Boostexter: a boosting-based system for text categorization', Machine Learning, vol. 39, no. 2/3, pp. 135-68.

Thabtah, F.A., Cowling, P. & Peng, Y. (2004), 'MMAC: A New Multi-class, Multi-label Associative Classification Approach', paper presented to Proceedings of the 4th IEEE International Conference on Data Mining, ICDM '04.

Witten, I.H. & Frank, E. (1999), 'Data Mining: Practical machine learning tools with Java implementations', Morgan Kaufmann.

Wolpert, D.H. (1992), 'Stacked Generalization', Neural Networks, vol. 5, pp. 241-59.

Zhang, M.-L. & Zhou, Z.-H. (2005), 'A k-Nearest Neighbor Based Algorithm for Multi-label Classification', paper presented to Proceedings of the 1st IEEE International Conference on Granular Computing.