Towards Multilabel Rule Learning

Eneldo Loza Mencía and Frederik Janssen
Knowledge Engineering Group, TU Darmstadt
Darmstadt, Germany
eneldo@ke.tu-darmstadt.de, janssen@ke.tu-darmstadt.de

Abstract

In this position paper, we provide first insights into possible schemes to utilize rule learning algorithms to solve the task of multilabel classification. The main idea is to exploit specific properties of symbolic rule representations to build models that consist of high-quality multilabel rules. To this end, novel ideas which rely on the adaptation of conventional inductive rule learners to multilabel data are presented. Their expected advantages and disadvantages, opportunities and limitations are reviewed and discussed.

1 Introduction

Rule learning has a very long history and is a well-known problem in the machine learning community. Over the years many different algorithms to learn a set of rules were introduced. The main advantage of rule-based classifiers are interpretable models as rules can be easily comprehended by humans. Also, the structure of a rule offers the calculation of overlapping of rules, more specific, and more general relations. Thus, the rule set can be easily modified as opposed to most statistical models such as SVMs or neural networks. However, most rule learning algorithms are prone to multi-class classification.

On the other hand, many problems involve assigning more than one single class to an object. These so-called multilabel problems can be often found when text is classified into topics or tagged with keywords, but there are also many examples from other media such as the recognition of music instruments or emotions in audio recordings or the classification of scenes in images and from the domain of biology and gene function classification.

It is widely accepted that one major issue in learning from multilabel data is the exploitation of label dependencies. Learning approaches may greatly benefit from considering label correlations, and we believe that rule induction algorithms provide a good base for this. Firstly, label dependencies can directly be modeled and expressed in form of rules. Secondly, such rules are directly interpretable and comprehensible for humans. Even if complex and long rules are generated, the implication between classes can be estimated more easily than with other approaches by focusing on the part of the rules considering the classes.

In this paper, we present current work in progress and perspectives towards multilabel rule learning. Relatively little work exist regarding rule learners taking into account the popularity of multilabel classification. An overview of related work shows the current possibilities and limitations of such approaches. The challenges in rule induction and multilabel learning are reviewed and two general directions are proposed and discussed.

2 Related Work

Many rule-based approaches to multilabel learning rely on association rules. This is an obvious choice as this kind of rules is capable of having more than one condition in the head of the rule. However, as the goal of all classification algorithms is to assign classes to examples, usually Classification Association Rules (CARs) are used, instead of regular association rules that are induced in an unsupervised fashion. Often, these single-label association rules are introduced as a first step and then are combined to yield multilabel association rules or are used to directly predict the labels of a given test instance. The latter works by using all single-label association rules that cover the example and predict all labels that are in the head of these rules. However, in this case, the model does not consist of multilabel association rules.

The literature shows only a few approaches to multilabel rule learning. Most of them utilize association rule learning to induce the set of rules. As mentioned above, often the capability of the algorithms to handle multilabel data does not stem from the representation of the model (i.e., by using multilabel rules) but is reached by employing certain classification schemes. The approach of Arunadevi and Rajamani (2011) operates on spatial data. Single-label association rules are learned by using a multi-objective genetic algorithm. Then, the rules are sorted by a weighted combination of support, confidence and J-measure, and the final classifiers is produced according to this ranking.

In the same manner as Arunadevi and Rajamani (2011), Ávila *et al.* (2010) use a genetic algorithm to induce the single-label association rules. However, they use a decision list for classification of single labels. The multilabel prediction is also built by using a combination of all covering rules of the different rule sets. They also account for a good distribution of the labels by using a token-based recalculation of the fitness values of each rule.

Li *et al.* (2008) also learn single-label association rules and the test data is labeled by setting exactly those labels that have a probability greater than 0.5 in the covering rules.

Another method that can be applied to tackle multilabel data are the so-called multilabel alternating decision trees (De Comité *et al.*, 2003). The idea is to adapt boosting techniques to multilabel classification. As a result, the algorithm yields rules that have only one decision (similar to decision stumps) and that predict confidence values for each label.

A different idea is to change the model representation to make it suitable for multilabel data. Consequently, the rule representation has to be generalized to multilabel, i.e., a label vector instead of a single value in the head of the rules. In the work of Allamanis *et al.* (2013), such a generalized rule format is introduced. Interestingly, the proposed rules also allow for postponing the classification by offering a "*don't care*"-value. As there may be cases where the rule is not confident enough or simply when no rule covers the example such a value may be beneficial. In this work, a Michigan-style Learning Classifier System (LCS) is used in combination with a genetic algorithm. The classification is done by using a weighted voting scheme (the fitness of the rules is used as weight) as many multilabel rules may cover the example.

Another algorithm that also finds multilabel rules is *MMAC* (Thabtah *et al.*, 2004). The idea here is to use a multi-class, multilabel associative classification approach by not only generating from all frequent itemsets the rules that pass the confidence threshold but also include the second best rules and so on. These single-label association rules then are merged to create multilabel rules. The algorithm proceeds by deriving the frequent itemsets, generating the association rules, removing the covered instances, and repeat these steps on the remaining instances. Hence, rules that have the same conditions in the body then can be merged by using their single-label classes in the multilabel vector in the head of the rule. In this manner it is possible to create a total ranking of all labels for each test instance.

Another associate multilabel rule learner with several possible labels in the head of the rules was developed by Thabtah *et al.* (2006). These labels are found in the whole training set, while the multilabel lazy associative approach of Veloso *et al.* (2007) generates the rules from the neighborhood of a test instance during prediction. The advantage then is that fewer training instances are used to compute the coverage statistics which is beneficial when small disjuncts are a problem as they are often predicted wrong due to whole training set statistics. Another important aspect mentioned in this work is that essentially one assumes dependencies between the labels. Otherwise, multilabel data can be simply solved by decomposing it into single-label datasets by using schemes such as binary relevance. Surprisingly, Veloso *et al.* (2007) was the only work that mentioned this problem. Their solution is simple as they use the prediction of a first iteration as additional attribute in the dataset for a second iteration. This lasts as long as labels remain unused in the attribute section of the dataset.

In summary, most of the relevant work is based on classification association rules (CARs). Often, evolutionary algorithms are used to derive a high-quality rule set. Label dependencies are not tackled explicitly though they might be taken into account by algorithm-specific properties.

3 Multilabel Classification

Multilabel classification refers to the task of learning a function that maps instances $\mathbf{x} = (x_1, \ldots, x_m) \in \mathcal{X}$ to label subsets or label vectors $\mathbf{y} = (y_1, \ldots, y_n) \subset \{0,1\}^n$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$, $n = |\mathcal{L}|$ is a finite set of predefined labels and where each label attribute y_i corresponds to the absence (0) or presence (1) of label λ_i . Thus, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance.

Potentially, there are 2^n different allowed allocations of $\mathbf{y}$, which is a dramatic growth compared to the n possible states in the multiclass setting. This, and especially the resulting correlations and dependencies between the labels in $\mathcal{L}$, make the multilabel setting particularly challenging and interesting compared to the classical field of binary and multiclass classification.

From a probabilistic point of view, one of the main differences between multilabel and binary or multiclass classification are the possible dependencies in the label output space. In binary and multiclass problems the only observable probabilistic dependence is between the input variables, i.e. the attributes x_j, and the label variables y_i. A learning algorithm tries to learn exactly this dependence in form of a classifier function h. In fact, if a classifier provides a score or confidence for its prediction $\hat{\mathbf{y}}$, this is often regarded as an approximation of $P(\mathbf{y} = \hat{\mathbf{y}} \mid \mathbf{x})$, i.e. the probability that $\hat{\mathbf{y}}$ is true given a document $\mathbf{x}$.

As mentioned above, we may additionally observe dependencies between labels in multilabel classification. I.e. we may observe that the occurrence or absence of single labels under certain circumstances *correlate* with each other. From the early beginning of multilabel classification, there have been attempts to exploit these types of *label correlations* (cf. e.g. McCallum, 1999; Ghamrawi and McCallum, 2005; Zhu *et al.*, 2005). A middle way is followed by Read *et al.* (2009) and Dembczyński *et al.* (2010a) and their (probabilistic) classifier chains by stacking the underlying binary relevance classifiers with the predictions of the previous ones. However, only recently Dembczyński *et al.* (2010b) provided a clarification and formalization of label dependence in multilabel classifications. Following their argumentation, one must distinguish between unconditional and conditional label dependence. Roughly speaking, *unconditional dependence* or independence of labels does not depend on a specific given input instance (the condition) while *conditional dependence* does. An example may illustrate this.

Suppose a label space indicating topics from news articles, and suppose further that λ_u is the topic *politics* and λ_v corresponds to *foreign affairs*. Especially if the topics are organized in a hierarchy and λ_v is a sub-topic of λ_u, there will obviously be a dependency between both labels. We will hence observe y_u with a different probability $P(y_u = 1) < 1$ as if y_v was also observed, since then it holds $P(y_u = 1|y_v = 1) = 1$. The probability $P(y_v = 1|y_u = 1)$ of seeing an article about *foreign affairs* on a page in the politics section will in turn be also much higher than by just randomly opening the newspaper, which corresponds to $P(y_v = 1)$. These probabilities are *unconditional* since they do not depend on a particular document. Suppose now that a news article is about the *Euro crisis*. The *conditional* probabilities $P(\lambda_u = 1|\mathbf{x})$, $P(\lambda_v = 1|\mathbf{x})$ and $P(y_v = 1|y_u = 1, \mathbf{x})$ would likely increase and hence be different from the unconditional ones. However, if an article was about the *cardiovascular problems of Ötzi*, we would observe that both labels are *conditionally independent*, since (very likely) $P(y_u = a|y_v = b, \mathbf{x}) = P(y_u|\mathbf{x}) = 1 - a$ for all $a, b \in \{0, 1\}$ and interchanged u and v.

4 Inductive Rule Learning

Inductive rule learning is researched very well. Over the years the community has introduced a bunch of algorithms that are still in use (cf., *Ripper* (Cohen, 1995) as one of the popular examples). However, most multilabel rule learning algorithms rely on association rule mining (cf., Section 2), the combination of inductive rule learners and multilabel data is yet to be evaluated.

A rule learning algorithm has a set of rules as result. These rules are of the form

$$body \rightarrow head$$

where the body consists of a number of conditions (attribute-value tests) and, in the regular case, the head has only one single condition of the form $y_i = 0$ or 1. However, multilabel rules may have several of such conditions.

Most inductive rule learning algorithms for classification employ a separate-and-conquer (SeCo) strategy (Fürnkranz, 1999). Its basic idea is to find a rule that covers a part of the example space that is not explained by any learned rule yet (the conquer step). The possible candidates are evaluated according to a quality function (heuristic) defined on statistics of covered positive and negative examples. After such a rule is found, it is added to the current set of rules, and all examples that are covered by this rule are removed from the data set (the separate step). Then, the next rule is searched on the remaining examples. This procedure is repeated until no more (positive) examples are left. In order to prevent overfitting, the two constraints that all examples have to be covered (*completeness*) and that no negative example has to be covered in the binary case (*consistency*) can be relaxed so that some positive examples may remain uncovered and/or some negative examples may be covered by the set of rules.

Obviously, there are some shortcomings when the SeCo strategy should be employed on multilabel data. First of all, there is no direct and intuitive notion of positive and negative examples (cf. also Section 5) This affects the computation of the heuristics for selecting the candidate conditions.

Secondly, a SeCo algorithm is usually learned in order to subsequently cover the examples of each possible class (ordered one-against-all). This is obviously not longer possible in the multilabel setting, since an example may belong to different classes. Hence different decomposition approaches and stopping criterions have to applied in the multilabel settings.

5 Multilabel Rule Learning

The predominant approach in multilabel classification is *binary relevance* learning Tsoumakas and Katakis (cf. e.g. 2007). It tackles a multilabel problem by learning one classifier for each class, using all objects of this class as positive examples and all other objects as negative examples. There exists hence a strong connection to concept learning, which is dedicated to infer a model or description of a target concept from specific examples of it (see e.g. Domingos, 1997, Sec. 2.2). When several target concepts are possible or given for the same set of instances, we formally have a multilabel problem.

The problem of this approach is that each label is considered independently of each other, and as we have seen by the example given before, this can lead to loss of useful information for classification. This problem is commonly shared by all approaches mentioned in Section 2 which can contain only one condition, i.e. one label in the head of a rule.

5.1 Labelsets Approach

A rule induction approach which may consider several conditions in the head seem hence more appropriate for the multilabel setting. A possible simple solution is to use the label powerset transformation (Tsoumakas and Katakis, 2007), which decomposes the initial problem into a multiclass problem with $\{P_{\mathbf{x}} \mid \mathbf{x} \in \text{training set}\} \subseteq 2^{\mathcal{L}}\}$ as possible classes. This problem can then be processed with common rule induction algorithms, which will thus produce rules with several labels in the head.

In general, we can state that this approach is able to consider conditional dependency between labels of high order when using a separate and conquer approach, since rules are learned locally on subsets of the instances. However, an obvious disadvantage is that we only can only predict label relations and combinations which were seen in the training data. Hence, no new relationships can be discovered, and we may miss the correct labelsets in unknown test data.

We propose to modify the SeCo iteration as explained in the following: Firstly, we learn so-called multiclass decision lists, which allows to use different heads in the rules of the the decision list. If we limit ourselves to labelsets seen in the training data, this corresponds to using label powerset transformation with a multiclass decision list learner, with the mentioned shortcomings. In addition, the evaluation for each possible labelset can be very expensive ($\mathcal{O}(\min(2^n, m))$ in contrast to $\mathcal{O}(n)$). Hence, we propose a greedy approach. It starts by evaluating the current added condition with respect to all labels independently in order to determine the best covered label. If we add an additional label to our head, we can only stay the same or get worse, since the number of covered examples remain the same and the number of covered *positives*, for which the head applies, can not increase. Hence, we can safely prune great part of the label combinations as soon as the heuristic becomes worse.

Several aspects of this approach have to be analyzed. Firstly, it is not clear whether the greedy refinement step leads to mostly single label heads. Secondly, an interesting issue is the effect of allowing negative predictions, i.e. heads of the type $y_i = 0$. This is somehow contrary to the notion of concept learning, where we are interested in finding convenient representations of *the concept*, but it is in line with the label symmetry assumption of binary relevance and many other multilabel approaches. And thirdly, it has to be analyzed if this approach is indeed effective in predicting labelsets which could not be observed in the training set.

5.2 Chaining and Bootstrapping

An effective approach for exploiting conditional label dependencies showed to be classifier chains (Read *et al.*, 2009). Classifier chains (CC) make use of stacking the previous binary relevance predictions in order to implement the chain rule in probability theory $P(y_1, \ldots, y_n) = P(y_n \mid y_1, \ldots, y_{n-1})$, since they learn the binary classifiers h_i with training examples of the form $(x_1, \ldots, y_1, \ldots, y_{i-1})$ (Dembczyński *et al.*, 2010a). One drawback of CC is the predetermined, fixed order of the classifiers (and hence the labels) in the chain, which makes it impossible to learn dependencies in the contrary direction.

Thus, we propose to use a bootstrapping approach in order to benefit from the chaining rule effect but also in order to overcome the main disadvantage of CC, the fixed order. As we will see, our version of bootstrapping is particularly adequate for rule induction.

Like in binary relevance, we learn one theory for each label, but we expand our training instances by the label information of the other labels, i.e. the training examples vectors for learning label y_i is $(x_1, \ldots, y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$. Hence, we obtain theories with label attributes in the body, like in CC. The prediction for a test instance begins with empty label attributes, which means that they are set to *unknown*. Here we benefit from the natural support for such attribute states (*missing*, *don't care*, etc.) of symbolic approaches. Hence, in the first iteration, only rules apply which do not include any label attribute in the body. These rules were generated during the training process because there was enough local evidence and support for such a decision, which is only based on the instance attributes. This would be hardly reasonably and justifiable if we were using approaches like SVMs, which are in general not excluded from being used in similar bootstrapping settings. The prediction is then used in the next iteration to set the corresponding label attribute for the other classifiers. However, if no appropriate rule was found we prefer to absent from classifying instead of applying the default rule (predicting the majority class) so that the attribute may be filled up in consequent iterations. Again, rule induction algorithm naturally provide this option.

A deadlock may of course occur if no rules apply at all. We are currently investigating this issue also with respect to using different heuristics, but the overall preliminary results are very promising.

Nevertheless, the next natural step is to skip the binary relevance decomposition and to (virtually) apply bootstrapping directly in the SeCo training phase, hence to learn one single theory with rules with label conditions in the body.

6 Conclusions

This work deals with the challenges and chances of using rule induction in multilabel learning. We have presented two main perspectives. The first one addresses the fact that multilabel learning has to deal with sets of classes rather than single classes. The second one addresses the problem of label dependencies by using bootstrapping. In essence, both issues are solved by extending the formulation of the head and the body of a rule with additional conditions on the labels. First experiments with the bootstrapping approach make us confident about the potential of multilabel rule induction. However, we are still at the beginning of implementing all the presented ideas.

Moreover, many other aspects have still to be addressed: The right selection of the heuristic was already a complex issue in traditional rule induction and has to be reviewed for multilabel learning. Also, unordered and multiclass decision lists gain new relevance. And of course, a combination of both approaches, leading to global rules describing multilabel data, is also worth to be investigated.

References

M. Allamanis, F. Tzima, and P. Mitkas. Effective Rule-Based Multi-label Classification with Learning Classifier Systems. In *Adaptive and Natural Computing Algorithms, 11th International Conference, ICANNGA 2013*, pages 466–476, 2013.

J. Arunadevi and V. Rajamani. An evolutionary multi label classification using associative rule mining for spatial preferences. *IJCA Special Issue on Artificial Intelligence Techniques - Novel Approaches and Practical Applications*, (3):28–37, 2011.

J. Ávila, E. Galindo, and S. Ventura. Evolving Multi-label Classification Rules with Gene Expression Programming: A Preliminary Study. In *Hybrid Artificial Intelligence Systems*, volume 6077, pages 9–16. Springer, 2010.

W. W. Cohen. Fast Effective Rule Induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML-95)*, pages 115–123, 1995.

F. De Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision trees from texts and data. In *Proceedings of the 3rd international conference on Machine learning and data mining in pattern recognition*, MLDM'03, pages 35–49, Berlin, Heidelberg, 2003. Springer-Verlag.

K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 279–286. Omnipress, June 2010.

K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. On label dependence in multi-label classification. In *Proceedings of the ICML-10 Workshop on Learning from Multi-Label Data*, pages 5–12, June 2010.

P. Domingos. A Unified Approach to Concept Learning. Dissertation, University of California, Irvine, 1997.

J. Fürnkranz. Separate-and-Conquer Rule Learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

N. Ghamrawi and A. McCallum. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200. ACM, 2005.

B. Li, H. Li, M. Wu, and P. Li. Multi-label Classification based on Association Rules with Application to Scene Classification. In *Proceedings of the 2008 The 9th International Conference for Young Computer Scientists*, pages 36–41. IEEE Computer Society, 2008.

A. K. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*, 1999.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier Chains for Multi-label Classification. In *Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.

F. Thabtah, P. Cowling, and Y. Peng. MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 217–224. IEEE Computer Society, 2004.

F. Thabtah, P. Cowling, and Y. Peng. Multiple labels associative classification. *Knowledge and Information Systems*, 9(1):109–129, 2006.

G. Tsoumakas and I. Katakis. Multi Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

A. Veloso, W. Meira, M. A. Gonçalves, and M. Zaki. Multi-label lazy associative classification. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 605–612. Springer, 2007.

S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281. ACM, 2005.