Noname manuscript No.
(will be inserted by the editor)

Local and Global Feature Selection for Multilabel Classification with Binary Relevance

An Empirical Comparison on Flat and Hierarchical Problems

André Melo · Heiko Paulheim

Received: date / Accepted: date

Abstract Multilabel classification has become increasingly important for various use cases. Amongst the existing multilabel classification methods, problem transformation approaches, such as Binary Relevance, Pruned Problem Transformation, and Classifier Chains, are among the most popular, since they break a global multilabel classification problem into a set of smaller binary or multiclass classification problems, which are well understood and extensively researched. Transformation methods enable the use of two different feature selection approaches: *local*, where the selection is performed independently for each of the transformed problems, and *global*, where the selection is performed on the original dataset, meaning that all local classifiers work on the same set of features. While global methods have been widely researched, local methods have received little attention so far. In this paper, we compare those two strategies on one of the most straight forward transformation approaches, i.e., Binary Relevance. We empirically compare their performance on various flat and hierarchical multilabel datasets of different application domains. We show that local outperforms global feature selection in terms of classification accuracy, without drawbacks in runtime performance.

Keywords multilabel classification · transformation methods · local feature selection · global feature selection · binary relevance

1 Introduction

Multilabel classification denotes a classification problem where a single instance cannot be assigned only one, but multiple classes. It has gradually attracted significant attention from various communities and has been widely applied to diverse

A. Melo
B6 26 (Room C 1.06), 68161 Mannheim
E-mail: andre@informatik.uni-mannheim.de

H. Paulheim
B6 26 (Room C 1.09), 68161 Mannheim
Tel.: +49 621 181 2646
E-mail: heiko@informatik.uni-mannheim.de

problems from automatic annotation for multimedia contents to bioinformatics, web page classification, information retrieval, tag recommendation, type prediction in knowledge bases, and many others.

There are two general families of multilabel classification algorithms: (1) *adaptations* of single-label machine learning algorithms which deal with multilabel data directly, and (2) *transformation methods*, which decompose the multilabel problem into a set of simpler learning problems, usually binary classification. Transformation methods have been widely used and allow standard binary classifiers to be used on the transformed problems, which are independent from each other and can be easily parallelized.

Binary Relevance (BR) is one of the simplest and most popular transformation methods. Its main drawback is that it does not consider dependencies between labels, which can be modeled with more complex transformation methods such as Classifier Chains (CC), Label Power-sets (LP), or Pruned Problem Transformation (PPT). On the other hand, taking those dependencies into account results in higher computational complexity, and scalability requirements might prohibit the use of more sophisticated methods which model dependencies. However, it has has been shown that in many cases, BR can yield predictive performance as good as more complex methods depending on the characteristics of the data [32]. Hence, in this paper, we focus on BR as a transformation method.

Feature selection is an important part of machine learning, allowing the reduction of training time, as well as the improvement of predictive quality [13,23,35]. When using a transformation approach for multilabel classification, the selection of features can be performed locally or globally. In the global approach, the feature selection is performed only once on the original dataset, and the set of selected features is the same for all local transformed datsets. In the local approach, the selection is performed separately for each local classifier on its correspondent transformed dataset, which means that different local classifiers may work on different sets of class-specific features specialized for each transformed problem. This can be particularly interesting for some datasets where the features which are relevant for predicting one class might not be relevant for another.

In hierarchical multilabel classification, where the labels are structured in a hierarchy, the use of the hierarchies can also influence the predictive performance and runtime of the local and global feature selection processes [50]. In this paper, we examine the difference between local and global feature selection not only with flat multilabel data, but also on hierarchical multilabel classification problems. So far, the global feature selection approach has been widely studied in the literature [16,52,65], while the local approach, although already considered in the context of multiclass classification [30], has received little attention and has not been systematically evaluated in the multilabel classification problem. The idea of *generating* class-specific features for the problem transformation method has been considered by LIFT [63], however, there are no works considering class-specific feature *selection* and performing a comparison with the global approach in terms of predictive performance and runtime on transformation based classifiers.

In this paper, we conduct an empirical comparison between the local and global feature selection approaches. We show that when using transformation methods, the local approach results in a better overall predictive performance with similar runtime. To the best of our knowledge, although local and global methods have

been discussed in the literature, this is the first systematic empirical evaluation and comparison of the two approaches.

The rest of this paper is structured as follows. Section 2 gives an introduction to multi-label classification, feature selection, and the evaluation metrics used in this paper. Section 3 introduces related works, parts of which are also used for comparison. Section 4 discusses the use of local feature selection with transformation methods, and section 5 presents the empirical results. We close with a discussion and an outlook on future work.

2 Background

In this section, we present the background work relevant to the understanding of this paper. We define the flat and hierarchical multilabel classification problems and briefly present various state-of-the-art methods, as well as evaluation measures used in the experiments.

2.1 Multilabel Classification

In the multi*label* classification problem, there are multiple classes and, contrary to the multi*class* classification problem, instances are allowed to have more than one class. We define the set of classes as $C = \{c_1, ..., c_{|C|}\}$, and we represent the multilabel of an instance x with a binary vector $y = (y_1, ..., y_{|C|}) \in \{0, 1\}^{|C|}$.

Multilabel classification approaches can be divided into transformation and adaptation methods. These will be discussed in more details later in this section. A comprehensive review on multilabel classification algorithms is given in Zhang et al. [64].

Some of the existing multilabel classification approaches are variations of standard binary classification algorithms, which have been adapted to the multilabel task without requiring problem transformations. This includes, e.g., *AdaboostMH* [49], *MLkNN* [66], and MLC4.5 [11], which are the respective multilabel versions of Adaboost, k-Nearest Neighbors, and C4.5 (a decision tree algorithm):

- AdaBoostMH is an adaptation of AdaBoost with its weak-learning conditions being based on a one-against-all reduction to binary, which was originally designed to use weak-hypotheses that return a prediction for every example and every label.
- The retrieval of the multilabel k-nearest neighbors is the same as in the traditional k-NN algorithm. The main difference is the determination of the label set of a test example, where the prior and posterior probabilities of each label within the k-nearest neighbors are considered and the Bayesian rule is use to derive the predicted set of labels.
- The multi-label C4.5 algorithm (ML-C4.5) adapted the original C4.5 by modifying the formula for calculating entropy to consider distributions over all labels, which is equivalent to the sums of the entropies for each individual class label, and allowing multiple labels in the leaves of the trees.

Adaptation methods usually learn a single model capable of predicting all classes. In comparison, transformation methods will have as many models as transformed datasets. Therefore, generally speaking, transformation methods require more memory space than adaptation methods.

Transformation methods decompose the multilabel problem into a set of binary classification problems. There are mainly three categories of transformation methods according to Madjarov et al. [32]: binary relevance, label power-set, and pair-wise methods.

The most popular method is called *Binary Relevance* [55] (BR), which trains a binary classifier for each class (against the others), inherently assuming independence between the classes. *Classifier Chains* [45] (CC) arrange the local classifiers in a chain where the outcome of a classifier is used as a feature on the next classifiers in the chain, allowing some dependency between labels to be modeled.

In the label power-set category, every different combination of labels occurring in the training data is considered an individual class and the transformed classification problem is multiclass. The potentially high number of label combinations poses scalability challenges, as the number of label combinations can grow exponentially with the number of labels resulting in up to $2^{|C|}$ transformed labels. This can significantly increase the complexity of the problem, making it prohibitive for datasets with a high number of classes.

Pruned Problem Transformation (PPT) [42], in order to reduce the complexity of the LP approach, selects only the transformed labels that occur more than a minimum number of times. *HOMER* [57] (acronym for Hierarchy Of Multi-labEl leaRners) generates a hierarchy for the labels, with meta-labels being non-leaf nodes representing the union of a subset of labels. A multilabel classifier is then trained for every non-leaf node in the hierarchy having the node's children as labels.

Ensemble methods use multiple adaptation and problem transformation methods as base classifiers and combine the output of the different trained models to make a prediction. *Random k-Labelsets (RAKeL)* [56] generates random sets of labels and trains a label power-set classifier for each randomly generated subset. *Ensemble of Pruned Sets* (EPS) [44] uses pruning to reduce computational complexity as well as example duplication method in order to reduce the error rate. *Ensemble of Classifier Chains* (ECC) [45] has classifier chains as base classifiers, the output of each base classifier is summed, and a threshold is applied to select the labels predicted.

An extensive experimental comparison of multilabel classifiers including adaptation transformation and ensemble methods is reported in [32]. The authors recommend the use of four different classifiers, three of which are transformation methods, i.e., HOMER, BR, and CC.

2.2 Hierarchical Multilabel Classification

The hierarchical multilabel classification problem is similar to the multilabel classification problem, but the classes C are additionally arranged in a hierarchy G. The labels of an instance should be consistent with G, i.e., if an instance belongs to a non-root class then it must also belong to its ancestors (i.e., $\forall c_i \sqsubseteq c_j$, if $y_i = 1$ then $y_j = 1$). The class hierarchy can be of two types: a tree, which allows nodes

to have a single parent only, and a directed acyclic graph (DAG) which allows nodes to have multiple parents.

Two important aspects of hierarchical multilabel classification approaches are whether they allow partial path predictions, i.e., mandatory vs. non-mandatory leaf node prediction, and whether they guarantee to output labels that are consistent with hierarchy or not.

As for multilabel classification, there are mainly two types of hierarchical multilabel classification approaches: local and global classifiers. The main difference is that the former is basically a transformation method which breaks down the classification problem into smaller and simpler problems exploiting the class hierarchy, while the latter considers the problem as a whole, learning a single more complex model.

The local hierarchical classification algorithms share a similar top-down approach in their prediction phase, where the classifier first predicts its first-level (most generic) classes of an instance, then it uses each predicted class to reduce the choices of classes to be predicted at the second level (the children of the classes predicted at the first level), and so on, recursively, until the most specific prediction is made. According to the hierarchical classification survey from Silla et al. [50], there are mainly two standard ways of using the local information: a *local classifier per node* and a *local classifier per parent node*. The *local classifier per level* approach, where one local multilabel classifier is trained for every level of the hierarchy, is also mentioned in the survey from Silla et al. [50]. However, very few works consider this approach since it suffers from inherent consistency problems: LCL has a single classifier per level and the children of nodes classified as false might also be classified as true. In contrast, LCN and LCPN guarantee consistency: in the prediction phase, they are applied top down, only predicting labels for lower levels in the hierarchy if their parent(s) have been predicted.

Local Classifier Per Node (LCN): The local classifier per node approach consists of training one binary classifier for each node of the class hierarchy. Similarly to Binary Relevance, each local binary classifier predicts whether an instance belongs to the class associated with the node or not. There are two main ways to define the training set of of the local binary classifiers. When training a local binary classifier for one label at hand, there are two strategies for selecting the negative examples for the local classifier: the *all* approaches, which uses all instances with all other labels as negative examples, and *siblings*, which only uses instances of the label's siblings in the hierarchy, reducing the size of the transformed datasets for classes in the lower levels of the hierarchy. A comparison of different negative example selection approaches is made in Eisner et al. [18] and Fagni et al. [20]. The results indicate that both approaches have roughly the same predictive performance, however, *siblings* is more scalable than *all*, as it reduces the average size of the local training sets.

Local Classifier Per Parent Node (LCPN): In this approach, a local multilabel classifier is learned for every parent (i.e., non-leaf) node in the hierarchy. The labels are the direct child nodes, and the training instances are those which belong to the parent node class. Depending on the choice of the local multilabel classifier, it is possible to model dependencies between sibling nodes. LCPN with Binary Relevance as a base multilabel classifier is equivalent to LCN using the siblings negative example selection policy.

In contrast to local classifier approaches, the global classifier approach (also known as *big bang approach*) learns one single classification model built from the training set, taking into account the class hierarchy as a whole during a single run of the classification algorithm. When used during the prediction phase, each instance is classified by the induced model, a process that can assign classes at potentially every level of the hierarchy to the instance.

One example for a global approach based on the Rocchio classifier is used in Labrou [29]. Some global methods do not guarantee consistency with the hierarchy and therefore need a post processing step in order to ensure consistency [24,25].

Clus-HMC [60] is a version of the previous method featuring predictive clustering trees [4] to generate a label hierarchy, which is not necessarily existent at first. Dimitrovski et at. [14] use ensemble approaches with Clus-HMC and report that the use of ensembles results in increased predictive performance. Otero et al. [37] propose a global hierarchical Ant-Miner classification method which is able to handle DAG hierarchies.

Further details about hierarchical multilabel classification methods can be found in the survey by Silla et al. [50], where an extensive comparison of classifiers, evaluation measures, as well as negative example selection policies is presented.

2.3 Evaluation Measures

In order to evaluate the predictive performance of the transformation based multilabel classifiers with different feature selection approaches, we use some popular measures recommended in the literature [50,32] for our experiments. The μP, μR and μF [24] are the micro-averaged measures of precision, recall and F-measure per class. By using the micro average, each class is weighted according to the frequency it occurs in the test data. The macro-average measures mF, mR and mF are the average with uniform weights for the classes. Equations 1 and 2 show the definition of these measures, where tp_i, fp_i and fn_i denote respectively the number of true positives, false positives and false negatives of the class c_i.

$$\mu P = \frac{\sum_{i=1}^{|C|} tp_i}{\sum_{i=1}^{|C|} tp_i + fp_i} \qquad \mu R = \frac{\sum_{i=1}^{|C|} tp_i}{\sum_{i=1}^{|C|} tp_i + fn_i} \qquad \mu F = 2\frac{\mu P \times \mu R}{\mu P + \mu R} \tag{1}$$

$$mP = \frac{1}{|C|}\sum_{i=1}^{|C|} \frac{tp_i}{tp_i + fp_i} \qquad mR = \frac{1}{|C|}\sum_{i=1}^{|C|} \frac{tp_i}{tp_i + fn_i} \qquad mF = 2\frac{mP \times mR}{mP + mR} \tag{2}$$

Equation 3 shows the Hamming loss (l_h) for one instance. We denote the true label vector of an instance as y, and the predicted vector as $\hat{y}$, with $y_i = 1$ if the instance is of class c_i, $y_i = 0$ otherwise. Hamming loss reports how many times on average, a class label is incorrectly predicted, i.e., the number of false positives and false negatives, normalized over total number of classes and total number of examples.

$$l_h(\hat{y}, y) = \sum_{i=1}^{|C|} 1_{\hat{y}_i \neq y_i} \tag{3}$$

Equation 4 shows the hierarchical loss (hl_H) [10], which is a hierarchical multilabel classification measure that extends hamming loss to account for any existing underlying hierarchical structure of the labels.

$$hl_H(\hat{y}, y) = \sum_{i=1}^{|C|} 1_{\hat{y}_i \neq y_i} \max_{\{j | c_i \sqsubseteq c_j\}} 1_{\hat{y}_j = y_j} \tag{4}$$

The idea of hierarchical loss is based on the notion that, whenever a classifier makes a mistake at a given node in a hierarchy, no further loss should be counted for any mistake in its descendants, therefore ignoring all wrong predictions for nodes which are descendants of a wrongly predicted node.

Costa et al. [12] review evaluation measure for hierarchical classification, and the authors conclude that there is no consensus on what measure should be used, and none of them have been widely adopted, while most works use the standard flat measures. Brucker et al. [7] make an empirical comparison of hierarchical and flat multilabel classification evaluation measures and search for relations between them. The authors report that hierarchical measures improve the quality assessment for hierarchical classification over flat measures, at the same time they state that flat and hierarchical measures agree on whether a classification result is good or not.

Cerri et al. [9] experimentally analyze methods for multilabel classification which are based on decision trees. Various evaluation measures are investigated in terms of consistency, discriminancy and indifferency. The authors suggest the use of hF, hP and hR as evaluation measures. These are equivalent to μF, μP and μR with all observed and predicted labels consistent with the class hierarchy as described in Section 2.2, i.e., if one instance is assigned a non-root class, it must also be assigned all the ancestors of the given class.

Kosmopulos et al. [27] make a detailed study of the problems of hierarchical classification evaluation measures. The authors categorize the evaluation measures into pair-based, which uses graph distance measures to compare predicted and true labels, and set-based, which use hierarchical relations to augment the sets of predicted and true labels (includes hF, hP, hR). Their results indicate that set-based measure F_{LCA} (F-measure with lowest common ancestor), which is highly correlated with hF, with the main difference being in DAG cases, which we do no consider in this paper.

For hierarchical classification we choose to use hF, because it is widely recommended in the literature. Since in our hierarchical classification experiments the labels are always consistent with the class hierarchy, which makes hF equivalent to μF, we choose to call it μF in our paper. In order to emphasize the performance on less frequent classes we also use the its macro-averaged version mF.

It is important to note that in this paper we do not use AUC and ROC, although these measures would be relevant in our evaluation. In order to use AUC and ROC in the multilabel context, it is necessary that all the classes share the same confidence threshold, however, in the transformation methods we allow

local classifiers to have their own confidence threshold values as they are trained independently from the other local classifiers.

2.4 Feature Selection Methods

Traditional feature selection methods can be divided into filter, wrapper, and hybrid methods [13,35]. Filter methods rank the features based on some relevance measure, and select the k highest ranked features according to that measure. It is the simplest and most scalable of the methods. Two of the most popular relevance measures used are mutual information (MI), which is equivalent to information gain, and chi-squared (χ^2). Other filter techniques include, for instance, relief [23], which is based on separation capabilities of randomly selected instances, and ensembles of different measures [48,36], where different ranking measures are used, and the feature ranks are combined. This approach has been extended for the hierarchical multilabel classification problem by Slavkov et al. [51].

In wrapper methods, a classifier is repetitively invoked and evaluated with different feature subsets. Exhaustive search is a method where all possible combinations of features are tested, and the one combination which yields the best performance is selected. This, of course, is utterly expensive and feasible only on datasets with small number of features. A popular wrapper method is the greedy forward search, where features are incrementally selected one at time in a greedy way, considering dependencies between features and reducing redundancy. It uses heuristics in order to reduce the search space, and therefore does not guarantee that the selected set of features is optimal.

Hybrid methods, as the name suggests, combine characteristics of both filter and wrapper methods. Huda et al. [21] include the filter's feature ranking score into the wrapper stage to speed up the search process. Zhu et al. [67] incorporate a filter ranking method into the memetic evolutionary algorithm accelerating the search and identifying core feature subsets.

3 Related Work

The quantitative effect of local feature selection has not been empirically researched, but some approaches follow similar ideas. Label specific sets of features has been exploited by the Label Specific Features (LIFT) [63]. The label specific features are generated by clustering the set of negative and positive instances (N_k and P_k) into m_k clusters each, and for each instance $2m_k$-dimensional features are generated by computing the distance from the instance to each of the centroids. Qu et al. [41] use the concept of local feature selection, and propose a relevance measure based on the density of negative and positive instances. It requires the distances between all pair of instances to be calculated, therefore the complexity grows with the number of instances squared, making it prohibitive for larger datesets.

Doquire and Verleysen [16] perform a comparison between multidimensional mutual information (MI) and chi-squared (χ^2) using greedy forward search strategy with problem transformation method (PPT). A nearest neighbors based MI estimator is used combined with a simple greedy forward search strategy to achieve

feature selection. Their experimental results indicate that the MI based approach has an advantage over the method based on the χ^2 statistic. Particularly, the proposed approach generally leads to an increase in performance both for the Hamming loss and the accuracy compared with the case where no feature selection is considered.

Zhang et al. [65] compare the performance of feature selection strategies based on principal component analysis (PCA) and genetic algorithms (GA) on MLNB (an adaptation of Naïve Bayes for the mutilabel problem), and on Naïve Bayes with binary relevance transformation. Their experiments indicate that the feature selection methods studied lead to an improved performance of the Naïve Bayes based classifiers. The authors raise scalability issues concerning the complexity of GA feature selection and its applicability on larger data with higher dimensional feature spaces. It is important to point out that the PCA requires the features to be numerical, therefore alternative methods or additional preprocessing steps should be considered for handling nominal attributes.

On the multi-class classification problem, de Lannoy et al. [30] study the local feature selection approach on the one-vs-all and one-vs-one approaches. They focus on correcting a potential bias caused by the fact that the binary classifiers are trained on different feature sets. On the hierarchical multilabel classification problem, Kosmopoulos et al. [26] uses a more scalable version of PCA for dimensionality reduction on sibling nodes at higher levels of the hierarchy which are the most expensive local classifiers because of the high number of instances. They perform experiments on the large-scale hierarchical text classification challenge data [38] (LSHTC[1]).

In this paper, we focus on the comparison of local and global feature selection approaches on transformation methods for multilabel classification, in particular binary relevance. To that end, we apply similar feature selection techniques locally and globally, evaluate the performance of transformation based methods with different base classifiers, and compare the results for flat and hierarchical multilabel classification datasets in terms of predictive performance and scalability.

4 Feature Selection on Transformed Multilabel Classification

When transforming a multilabel classification problem into a set of binary problems, there are two possible ways of performing feature selection: global feature selection, where all local binary classifiers are trained on the same set of globally selected features, and local feature selection, where the feature selection is performed separately for each local binary classifier.

4.1 Global Feature Selection

The global approach selects a set of features which is shared by all the local classifiers. This approach, in contrary to the local approach, can also be applied on adaptation methods. This may be one reason why the global approach has generally received more attention in the literature.

[1] `https://www.kaggle.com/c/lshtc`

Many global feature selection methods involve the computation of relevance measures individually for every class, followed by the aggregation of the class specific values. Spolaor and Tsoumakas [52] compare aggregation methods for the binary relevance approach. The approach consists of computing the relevance of each feature for every class individually, exactly like in the binary classification problem. Afterwards the global relevance of a given feature is computed by aggregating the relevance values of the feature for all classes.

The feature selection methods evaluated in the study were Mean, Min, Max, Round-Robin, and Rand-Robin. In Mean, the aggregated value is simply the average relevance of the feature over all classes, in Min the aggregation value is the lowest relevance and in Max the greatest. The features are then ranked by their aggregated values, and those with highest aggregated relevance are selected. Round-robin selects the best feature in the ranking related to each label in turn, while Rand-Robin selects the best feature for a label randomly chosen with probability inversely proportional to its frequency. Each feature taken in turn is removed from the rankings so that they cannot be selected again. The motivation for the Rand and Round-Robin approaches is to reduce the bias to selection of features more relevant to frequent classes to the detriment of less frequent ones.

The authors evaluate these approaches with both Chi-squared and Bi-normal Separation as relevance measures. The experimental results indicate the Max and Mean aggregation methods with chi-squared measure were the best performers.

Other feature selection methods adapts the traditional feature relevance for binary classification to consider all classes at the same time. Doquire and Verleysen [16] compare the multidimensional mutual information (MI), computed with Kozachenko-Leonenko estimation [28], and chi-squared (χ^2) relevance measures on a problem transformed with PPT and greedy forward search strategy for feature selection. The authors report a better performance of the MI based approach in comparison with the method based on the χ^2 statistic in terms of both hamming loss and accuracy. The use of greedy forward selection ensures dependencies between features are considered, however, that also affects scalability, which is an important aspect in our setting.

4.2 Local Feature Selection

The local feature selection approach can be applied on any transformation based multilabel classifier. The idea of local feature selection is to perform the selection for every local transformed dataset separately, i.e., different local classifiers may work on different sets of features, which are specialized for each subproblem. Given a dataset D with features set F, transformed labels $t \in T$, where t is a binary class attribute and T is the set of transformed labels, and transformed datasets $\{D_t | t \in T\}$, where D_t is a transformed dataset with binary class t. We define the set of locally selected features for each label as $F_t \subseteq F$ for each transformed dataset D_t, and F_{local} as the collection of subsets $\{F_t | t \in T\}$.

In order to demonstrate the practical difference between local and global feature selection approaches, we show as an example the top 10 features selected for the Enron dataset with the filter method and information gain. The dataset is a

Rank	Global	Local(B.B2)	Local(C.C6)	Local(C.C10)	Local(B.B9)
1	prices	subject	california	attorney	http
2	california	enron	commission	confidential	kaminski
3	price	pmto	price	received	forward
4	power	forwarded	diego	committee	email
5	utilities	steven	prices	include	issue
6	generators	kean	plants	continue	www
7	plants	original	customers	policy	ferc
8	federal	cc	generators	williams	meeting
9	electricity	message	market	50	mailto
10	davis	na	electricity	27	drive

Table 1: Enron features ranked by information gain in descending order

subset of Enron email corpus[2], labeled with a set of categories. It is based on a collection of email messages exchanged by Enron's board of directors that were categorized into 53 topic categories, such as company strategy, humour, and legal advice. The example from Table 1 shows the set of globally selected features and the sets of locally selected features for the labels B.B2 (Forwarded email(s) including replies), C.C6 (California energy crisis / California politics), C.C10 (legal advice), B.B9 (pointers to url). It is particularly noteworthy that the most relevant local features are very different, in fact, the sets of top-10 most relevant local features for the classes B.B2, C.C6, C.C10 and B.B9 are completely disjoint.

The Enron dataset is an example of a dataset where the local feature selection approach clearly outperforms the global approach. Different classes require very different sets of features for the local classifiers, and especially for small numbers of selected features, the difference in the performance is very significant. The predictive performance results for classifiers with the local and global feature selection approaches can be seen in the experiments section in Figure 1.

4.3 Analysis of Local Feature Sets

In this paper, we define F as the set of all features in a dataset, and $F_{\text{global}} \subseteq F$ as the set of globally selected features. In our setting, all the local feature sets $F_t, t \in T$, as well as F_{global} have the same number of features k.

In order to measure the differences between the set of globally selected features and the locally selected features we need to define measures which compare a set with a collection of set. We propose two similarity measures. The first measure (D_1), c.f. Equation 5, is the average Jaccard index of the set of global features F_{global} with each set of local features F_t. The second measure (D_2), c.f. Equation 6, is the ratio between the number of selected features k and the size of the union of all local feature sets $|\bigcup_{t \in T} F_t|$ and it indicates how similar with each other the local feature sets are.

The D_1 measure is bounded by the interval $[0, 1]$, while the D_2 measure is bounded by the interval $[1/|C|, 1]$. For D_1, a value of 1 means that the local sets are exactly equal to the global set, while a value close to 0 means that local feature sets and the global feature set have small intersections. For D_2, a value of 1 means

[2] `http://bailando.sims.berkeley.edu/enron_email.html`

that all the local sets are the same, while a value of $1/|C|$ means that the local features are completely disjoint.

$$D_1(F_{\text{global}}, F_{\text{local}}) = \frac{1}{|C|} \sum_{t \in T} \frac{|F_{\text{global}} \cap F_t|}{|F_{\text{global}} \cup F_t|} \tag{5}$$

$$D_2(F_{\text{local}}) = \frac{k}{|\bigcup_{t \in T} F_t|} \tag{6}$$

5 Experiments

In our experiments we make an extensive comparison of global and local feature selection approaches on standard flat and hierarchical multilabel datasets, as well as large-scale hierarchical multilabel datasets for type prediction on knowledge bases. We use binary relevance (BR) for flat multi-label classification, and its counterpart (i.e., local classifier per node, LCN) for hierarchical multi-label classification. We also run experiments with different popular binary classifiers as local classifier, and different feature relevance measures.

We vary the number of features to be selected, perform the feature selection with equivalent techniques using the local and global approaches, then run the same transformation classifiers on the different sets of selected features and compare their predictive performance, as well as runtime. We restrict the transformation approaches to binary relevance, which is a simple, popular, and generally well performing method, as discussed above. Other transformation methods might transform the labels in different ways, but at the end one classifier will be trained for each transformed label similarly to binary relevance. Therefore, for the goal of comparing the local and global feature selection approaches, the binary relevance method is sufficient.

The choice of the local binary classifier is an important factor in the evaluation, as the feature selection can have different impacts on different classifiers. In this paper we choose four different popular binary classifiers available in Weka 3.7.10: Naïve Bayes, Decision Tree (J48), K-Nearest Neighbors (IBk), Support Vector Machine (LibSVM), and AdaboostM1 (with decision stump).

A comparison against popular adaptation methods and the transformation methods with the adapted methods as local classifier is also performed. We select three popular adaptation method classifiers for our comparison: Multilabel k-nearest neighbors (MLkNN) and Multilabel C4.5 decision tree (MLC4.5). We compare them with Binary Relevance having their adapted binary classifiers (respectively k-NN, C4.5 and AdaboostM1) as local classifiers.

For our experiments, we use MULAN 1.5, which is an open-source Java library for learning from multilabel datasets based on WEKA. It includes a variety of state-of-the-art multilabel classification algorithms, and offers the global feature selection methods for binary relevance with Mean, Min and Max aggregation approaches. It also contains multilabel evaluation measures described in Section 2.3.

5.1 Datasets

We perform our experiments on popular flat and hierarchical multilabel classification datasets commonly used in performance benchmarking of multilabel classifiers. Furthermore, we created some additional benchmark datasets for hierarchical classification, which we extracted from large-scale cross-domain Linked Open Data sets, such as Wikidata, DBpedia, YAGO and NELL [34], as well as the smaller domain-specific datasets AIFB portal[3] and Mutagenesis.

For creating the benchmark datasets for hierarchical classification, we randomly sampled 10 000 instances from the larger Linked Open Data sets. There, instances usually come with types, which form a hierarchy in an ontology. Hence, predicting the type of an instance is a typical hierarchical classification task.[4] By doing the sampling randomly, we ensure that we do not introduce any bias which could be beneficial for one or the other method.

5.1.1 Multilabel Datasets

The flat multilabel datasets we use were obtained from MULAN datasets repository[5]. The datasets used include bibtex [22], birds [6], cal500 [58], corel5k [17], emotions [54], enron [44], genbase [15], imdb [43] medical [40], scene [5] and yeast [19], rcv1 subsets [31], the Yahoo datasets Arts1, Business1, Computers1, Education1, Entertainment1, Heath1, Science1, Social1, Society1 [59], delicious [57], tmc2007 [53], and slashdot[6]. Table 2 show statistics about the aforementioned datasets, where we state the number of labels, instances, features, cardinality (the average number of labels an instance has) and label dependency, which indicates the proportion of label pairs which are dependent (we consider dependent those pairs which fail the χ^2 test of independence, and the computation was performed using MULAN's UnconditionalChiSquareIdentifier class).

5.1.2 Hierarchical Multilabel Datasets

For the hierarchical mutilabel experiments we use the datasets from the biological domain which are available at the Clus datasets page[7]: cellcycle, church, derisi, eisen, gasch2, pheno and struc. These datasets are available in clus format and were converted to the Mulan format for our experiments using the converter existent in the Mulan library.

Additionally, we use datasets from Linked Open Data [2] extracted for the type prediction task, which is another example of hierarchical multilabel classification problem. The types, which are the labels to be predicted, are organized in a hierarchy, and various features can be extracted from the knowledge base graph and textual description of entities which are often available. The datasets we use are

[3] `http://www.aifb.kit.edu/web/Web_Science_und_Wissensmanagement/Portal`

[4] Note that this approach is only for generating benchmarks for hierarchical classification, and for comparing approaches to each other. However, we cannot transfer the results trivially to make a statement about how well the approaches work for the actual type prediction task in the original datasets.

[5] `http://mulan.sourceforge.net/datasets-mlc.html`

[6] `http://meka.sourceforge.net/#datasets`

[7] `https://dtai.cs.kuleuven.be/clus/hmcdatasets/`

Table 2: Statistics about the flat datasets used

Dataset	Instances	Labels	Features	Cardinality	Label Dep
Arts1	7484	26	23146	1.654	0.338
bibtex	7395	159	1836	2.402	0.111
birds	645	19	260	1.014	0.123
Business1	11214	30	21924	1.599	0.230
CAL500	502	174	68	26.044	0.192
Computers1	12444	33	34096	1.507	0.364
Corel5k	5000	374	499	3.522	0.030
delicious	16105	983	500	19.02	0.116
Education1	12030	33	27534	1.463	0.216
emotions	593	6	72	1.868	0.934
enron	1702	53	1001	3.378	0.141
Entertainment1	12730	21	32001	1.414	0.338
flags	194	7	19	3.392	0.381
genbase	662	27	1186	1.252	0.157
Health1	9205	32	30605	1.644	0.192
imdb	120919	28	1001	2	0.487
mediamill	43907	101	120	4.376	0.213
medical	978	45	1449	1.245	0.040
rcv1subset1	6000	101	47236	2.88	0.202
rcv1subset2	6000	101	47236	2.634	0.179
rcv1subset3	6000	101	47236	2.614	0.183
rcv1subset4	6000	101	47236	2.484	0.163
rcv1subset5	6000	101	47236	2.642	0.170
Recreation1	12828	22	30324	1.429	0.455
Reference1	8027	33	39679	1.174	0.169
scene	2407	6	294	1.074	0.934
Science1	6428	40	37187	1.45	0.196
slashdot	3782	22	1079	1.181	0.273
Social1	12111	39	52350	1.279	0.186
Society1	14512	22	49060	1.67	0.402
tmc2007	28596	22	49060	2.158	0.641
yeast	2417	14	103	4.237	0.670

NELL [8], Wikidata [61], DBpedia [3], and YAGO [33], AIFB, and Mutagenesis [47] As features we use binary attributes which indicate the existence of ingoing and outgoing properties [39, 46] for the first four datasets, and qualified relations [39], which indicate the existence of pairs of outgoing relations and object type, as well as pairs of ingoing relation and subject type.

For DBpedia, the types assigned to instances are only single-path, so that the hierarchical classification problem is not multi-label . On the other hand, relation features extracted from YAGO are too scarce to be meaningful. Therefore, we combine the two datasets by using types from YAGO and features from DBpedia. Since there are 384 174 types in YAGO, we select the top-474 most frequent types. We chose 474 because it was the original number of classes in DBpedia. We also use Wikidata, from wich we select the top-474 most frequent types, similarly to what was done to the YAGO types. The NELL dataset (08m.690) we used has only 10.3% of its originally 1 168 998 instances, since the properties are very sparse, and 89.7% of the instances have no features or only have the property *haswikipediaurl* which provides no information gain. The AIFB portal dataset describes the AIFB research institute in terms of its staff, research group, and publications. The data is an export of the AIFB website and contains around 270 thousand triples. The

Table 3: Statistics about the hierarchical datasets used

Dataset	Instances	Labels	Features	Card.	Fanout	Depth	Label Dep
aifb	27100	57	825	2.189	14.25	2.038	0.083
cellcycle	3757	498	78	8.716	2.846	3.709	0.286
church	3755	498	28	8.702	2.846	3.709	0.286
dbpedia-yago	2886305	474	1946	10.894	1.773	8.050	0.347
derisi	3725	498	64	8.759	2.846	3.709	0.286
eisen	2424	460	80	9.202	2.788	3.685	0.284
gasch2	3779	498	53	8.689	2.846	3.709	0.287
mutagensis	14157	86	132	2.398	7.167	2.486	0.041
NELL	120720	264	505	4.608	5.739	4.298	0.052
pheno	1591	454	70	9.175	2.751	3.682	0.236
struc	3838	498	19629	8.674	2.846	3.709	0.289
Wikidata	19254100	474	1866	2.007	5.386	1.948	0.003

type hierarchy is originally a wide and shallow tree with average fanout 14.25 and average depth 2.04. The MUTAG dataset is distributed as an example dataset for the DL-Learner toolkit[8]. It contains information about 340 complex molecules that are potentially carcinogenic, which is given by the isMutagenic property. The molecules can be classified as "mutagenic" or "not mutagenic", and the main entity types atoms bonds and compounds which define the molecules.

These hierarchical knowledge bases have a directed acyclic graph (DAG) as hierarchy. Since the MULAN framework only support trees, we have to simplify the problem and convert the DAGs into trees. It is important to mention that in the evaluation we ignore the original DAGs and consider only the converted trees. Therefore, for the nodes which have multiple parents, we retain only the subsumption relation with the parent class which is most frequent, i.e. the one with most instances, and remove the other edges. Alternatively, one could replicate the subtree of a node with multiple parents, leaving each replica with a single parent node. This, however, may significantly increase the number of classes, and most importantly generate consistency problems in case a classifier produces different predictions for the subtree replicas, which is a problem we do not address in this paper. Although supporting DAGs instead of converting them to trees can improve results [1], and the aforementioned conversion is an interesting approach to be considered, in this paper we restrict ourselves to the first conversion approach because of its simplicity.

Table 3 shows the same statistics from the Table 2 of flat datasets, plus average fanout, and the average depth of nodes. The average fanout is computed as the average number of children over all non-leaf nodes, and the average depth is computed over all the nodes in the hierarchy. For the average depth, we define depth of root nodes as one and the depth of non-root nodes as the depth of its parent plus one. The labels dependency is calculated similarly to the flat datasets case, however, instead of considering all possible label pairs, we consider only pairs of labels which are siblings.

[8] `http://dl-learner.org`

5.2 Scalability

In our experiments, we use the *filter* method for feature selection because it is a simple, popular, and highly scalable method. As discussed in Section 3, the filter method basically consists of the computation of relevance measures for every feature w.r.t. every label, and the ranking of these features by their relevance value.

For non-hierarchical multilabel classification problems, the computation of relevance measures on the local and global approach have the sample complexity of $O(|C| * |F| * |D|)$, where $|C|$ is the number of classes, $|F|$ the number of features (assumed to be binary) and $|D|$ the number of instances. This is because the relevance measure needs to be computed for every pair of a label and a feature, on all instances in the dataset. In the local case, this is done for a binary class on each transformed dataset ($|C|$ in the case of binary relevance). In the global case with aggregation method, the same computation has to be performed before aggregating the relevance values of each class into global relevance values, while in the multidimensional case, instead of 2-dimensional distributions, distributions over the $|C|$ needs to be computed. Therefore, in any of the cases mentioned before, the computation time grows linearly with the number of classes. The major difference is that the local approach has the disadvantage of having to sort the features by their relevance measure values $|C|$ times, while in the global approach it is performed only once. However, this does not change the overall complexity of the whole feature selection process: the additional effort of sorting the features is $|F| * log|F|$, and for almost every dataset, $log|F| < |D|$ holds (otherwise, the dataset would be very degenerate, having a few orders of magnitude more features than instances).

For hierarchical classification with siblings examples selection policy, the local feature selection is assumed to scale better since, for labels deeper in the hierarchy, the binary relevance measures are calculated only on a subset of the data. Typically, the number of labels in the lower levels of the hierarchy is higher, and the lower the level of the label node, the smaller is the subset of instances. Assuming that the labels hierarchy has a fanout b and the instances have a single path only, the complexity for computing the relevance measures in the local approach would be $O(b * log_b(|C|) * |D|)$, since the average transformed dataset size would be $|D| * (b * log_b(|C|))/|C|$ instead of $|D|$. The average size of the transformed datasets also increases with the cardinality of the multilabel dataset. However, for simplicity and because the cardinality is normally low in most real datasets, we ignore this factor when calculating the average size.

It is important to notice that the use of feature selection can reduce the overall runtime, as the multilabel classifier benefits from the dimensionality reduction. However, depending on the local classifier used and the feature selection method, the cost of performing the feature selection may be higher than the benefit of caused by the dimensionality reduction. Therefore, when employing a simple and highly scalable local classifier, such as Naïve Bayes, the overall runtime for some datasets without feature selection may be lower than with feature selection.

5.3 Results

Figure 1 shows a comparison between the local and global feature selection approaches for different numbers of selected features. The results reported were obtained with J48 as local classifier, the mean aggregation approach for the global feature selection, and ranking based feature selection with information gain as relevance measure. The reported runtime consists of the sum of feature selection and training time. The results reported in this section were obtained with 5-fold cross-validation. Because of space constraints, we show diagrams for only nine of the evaluated datasets[9].

The plots show that the local feature selection approach performs consistently better than the global approach, with a similar runtime for flat multilabel classification, and with significantly lower runtime for hierarchical. The difference in runtime for the hierarchical case is due to the siblings negative examples selection policy, as discussed above.

The difference in the micro F_1-measure is notably higher for smaller sets of selected features, where the average Jaccard index between locally and globally selected features is lower. Genbase is an example for a dataset which does not significantly profit from the local feature selection. Its average Jaccard index show that the locally selected features are not very different from the globally selected ones, converging very rapidly to a average Jaccard index of 1. Only for small numbers of selected features, the local approach shows an improvement in comparison to the global approach.

Tables 4 and 6 show a summary of the results for the datasets used in the experiments. We computed curves as in Figure 1 for all the datasets, local classifiers, i.e., micro/macro average F-measure graphed against the ratio of features selected. Instead of showing the plots as in Figure 1, we report the ratio of the area under the curves of local and global feature selection for the the *micro-averaged* F_1*-measure* ($R_{\mu F_1}$) and *macro-averaged* F_1*-measure* (R_{mF_1}) . Values greater than 1 for both $R_{\mu F_1}$ and R_{mF_1} means that the local approach outperforms the global approach. We report the normalized area under the curve of the D_1 and D_2 measures. The closer to 1, the more the local feature sets are similar to the global set. The last row in the tables (AVG) show the average value of the measures over all the datasets evaluated.

The averages show that that the vast majority of the reported ratios are larger than one, indicating that the local feature selection performs better overall. Depending on the method used as local classifier, the impact of the local approach can vary. Adaboost is the method which benefits the most in both flat and hierarchical datasets, while kNN benefits the least. When comparing the ratios of micro and macro-averaged F_1-measure ($R_{\mu F_1}$ and R_{mF_1}), the ratio of macro-averaged F_1-measure is higher than that of micro-averaged. This reveals that on flat datasets, the less frequent classes benefit from the local approach more than the more frequent classes. This can be explained by the fact that global feature selection approaches prefer features which are relevant to the more frequent classes, which means that, in general, the set of global features is more relevant to frequent classes than to infrequent ones.

[9] The complete set of plots can be found at `http://data.dws.informatik.uni-mannheim.de/hmctp/plots/report.pdf`

			J48		NaiveBayes		AdaBoost		kNN		LibSVM	
Dataset	D_1	D_2	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}
Arts1	0.339	0.237	1.553	1.736	1.135	1.539	1.343	1.549	1.488	1.585	1.572	1.630
bibtex	0.196	0.225	1.159	1.465	1.203	1.321	1.122	1.337	1.057	1.091	1.164	1.773
birds	0.308	0.284	0.996	1.143	1.208	1.174	1.156	1.307	0.996	0.971	1.634	1.218
Business1	0.265	0.238	1.027	1.423	0.998	1.435	1.007	1.306	1.034	1.359	1.023	1.480
CAL500	0.278	0.250	1.008	1.030	1.022	1.025	1.002	1.018	1.026	1.036	0.988	0.977
Computers1	0.242	0.233	1.110	1.448	1.192	1.593	1.079	1.701	1.088	1.287	1.109	1.606
Corel5k	0.168	0.225	2.223	1.054	1.848	1.358	2.942	1.015	1.548	1.093	1.109	1.006
delicious	0.146	0.227	1.736	1.312	1.352	1.543	2.524	1.474	1.297	0.948	2.099	1.518
Education1	0.285	0.245	1.491	1.666	1.062	1.464	1.297	1.422	1.418	1.650	1.579	1.780
emotions	0.386	0.365	1.022	1.044	1.008	1.011	1.030	1.053	1.021	1.023	0.953	0.937
enron	0.162	0.230	1.267	1.326	1.176	1.285	1.161	1.189	1.222	1.228	1.274	1.177
Entertainment1	0.357	0.256	1.754	1.732	1.161	1.539	1.836	1.770	1.512	1.440	1.811	1.744
flags	0.484	0.440	0.996	0.960	1.003	1.000	0.991	0.974	0.996	0.986	0.984	0.965
genbase	0.766	0.729	1.003	1.029	1.008	1.046	1.005	1.026	1.005	1.030	1.004	1.029
Health1	0.439	0.254	1.101	1.313	1.134	1.307	0.978	1.202	1.084	1.315	1.081	1.249
imdb	0.197	0.255	0.989	1.015	1.189	1.205	1.000	1.000	0.974	0.987	0.936	0.896
mediamill	0.117	0.238	1.026	1.063	0.999	1.149	1.025	1.045	1.013	1.013	1.028	1.042
medical	0.435	0.242	1.007	1.042	1.076	1.078	1.016	1.042	0.975	1.032	1.005	1.025
rcv1subset1	0.362	0.229	1.445	1.564	1.485	1.688	1.353	1.467	1.237	1.378	1.138	1.011
rcv1subset2	0.346	0.229	1.213	1.579	1.592	1.814	1.278	1.511	1.198	1.532	1.217	1.031
rcv1subset3	0.339	0.228	1.216	1.478	1.685	1.763	1.291	1.417	1.227	1.476	1.190	1.010
rcv1subset4	0.379	0.229	1.245	1.455	1.562	1.752	1.385	1.436	1.210	1.439	1.452	1.126
rcv1subset5	0.344	0.228	1.197	1.562	1.482	1.778	1.324	1.494	1.216	1.110	1.542	1.052
Recreation1	0.263	0.242	1.409	1.614	1.609	1.804	1.358	1.656	1.359	1.487	1.428	1.697
Reference1	0.464	0.242	1.167	1.210	1.246	1.142	1.095	1.095	1.159	1.285	1.141	1.185
scene	0.258	0.343	1.258	1.293	1.145	1.135	1.366	1.491	1.158	1.153	1.518	1.549
Science1	0.347	0.231	1.616	1.717	1.157	1.753	1.605	1.605	1.611	1.914	1.744	1.621
slashdot	0.381	0.239	1.548	1.289	1.551	1.469	1.213	1.129	1.408	1.319	1.726	1.356
Social1	0.334	0.245	1.022	1.759	1.239	1.666	0.974	1.480	1.029	1.582	1.035	1.765
Society1	0.301	0.251	1.404	2.307	1.313	1.579	1.504	2.007	1.341	1.837	1.374	2.168
tmc2007	0.188	0.241	1.157	1.757	1.145	1.819	1.153	1.843	1.155	1.546	1.157	1.970
yeast	0.164	0.267	1.021	1.056	1.017	1.040	1.030	1.085	0.997	1.003	1.052	1.185
AVG	0.314	0.269	1.262	1.389	1.250	1.415	1.295	1.348	1.189	1.285	1.283	1.337

Table 4: Comparison of local and global feature selection with *mean* aggregation on flat multilabel datasets

On the hierarchical datasets, however, the $R_{\mu F_1}$ is in general greater than R_{mF_1}, indicating that more frequent classes benefit more strongly from the local feature selection. The same fact that global feature selection approaches prefer features relevant to more frequent classes should apply in the hierarchical case as well. One possible explanation is that less frequent classes are at lower levels of the hierarchy, and errors for these classes can be caused by classification errors in the ascendant classes, since we used the top-down approach in this paper. That means the improvement in the local classifier of a leaf-node class, for example, is limited to the cases where the predictions of all the local classifiers of ascendant classes are correct.

Figures 2 and 3 show a comparison between adaptation methods and transformation methods using their correspondent adapted binary classifiers locally. We do that for MLC4.5 and MLkNN, which are shown respectively in the first, second and third column of plots. Similar to what was done in Figure 1, we vary the number of selected features with the adaptation methods using global feature

			J48		NaiveBayes		AdaBoost		kNN		LibSVM	
Dataset	D_1	D_2	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}
aifb	0.401	0.383	1.024	1.165	1.187	1.135	1.025	1.234	1.024	1.180	1.023	1.213
cellcycle	0.171	0.224	1.174	1.004	1.092	1.013	1.338	1.006	1.022	1.006	1.501	1.002
church	0.475	0.279	1.416	1.001	1.654	1.004	1.658	1.003	1.034	1.042	1.651	1.002
dbpedia-yago	0.271	0.299	1.010	1.146	1.005	1.210	1.004	1.130	1.014	0.901	0.952	0.336
derisi	0.128	0.228	0.992	1.001	1.095	1.011	1.013	1.002	1.026	0.978	0.609	0.998
eisen	0.085	0.225	1.091	1.001	1.098	0.997	1.178	0.994	1.007	1.012	1.030	1.001
gasch2	0.232	0.224	1.049	0.998	1.014	1.018	1.034	1.002	0.987	1.041	1.045	0.988
mutagenesis	0.203	0.333	1.194	1.130	1.201	1.127	1.190	1.118	1.196	1.139	0.781	1.021
NELL	0.228	0.248	1.022	1.085	1.019	1.036	1.000	1.023	1.019	1.092	1.021	1.040
pheno	0.326	0.256	1.177	1.003	1.094	1.002	1.709	1.007	1.055	1.010	1.017	1.000
struc	0.198	0.224	1.032	0.970	1.088	0.892	1.075	0.998	1.012	0.976	1.581	1.002
wikidata	0.215	0.389	1.004	1.027	1.022	1.150	1.004	1.023	1.006	1.033	1.003	1.009
AVG	0.244	0.276	1.099	1.044	1.131	1.050	1.186	1.045	1.033	1.034	1.101	0.967

Table 5: Comparison of local and global feature selection with *mean* aggregation on hierarchical multilabel datasets

selection since the local approach is not applicable. We also tried to compare to the AdaboostMH implementation in MULAN, however, the classifier did not seem to work properly, and the results remained constant over all the different sets of selected features used.

While we can observe that for transformation methods, local feature selection is generally favorable over global feature selection, the results are not conclusive about whether the adaptation or transformation methods have a better general performance. For some cases, such as the Corel5k dataset and decision tree classifiers, the adaptation method clearly outperforms the transformation method with local feature selection. On the other hand, for the same classifiers on the Enron dataset, the transformation method with local feature selection clearly outperforms MLC4.5. In order to draw any conclusions, a study dedicated to the comparison between adaptation methods and transformation methods with local feature selection would be required, which, however, is out of scope of this paper.

It is noteworthy that in this paper, we use binary relevance (BR) as transformation method for flat multilabel classification, and local classifier per node (LCN) for hierarchical multilabel classification. In the former all classes are assumed to be mutually independent, and latter all sibling nodes are assumed to be mutually independent. That means on datasets where labels have dependencies, the transformation methods evaluated are in disadvantage when compared with the adaptation methods which can model more label dependencies.

5.4 Statistical Analysis

After running all the experiments presented in this section, we need to test the significance of the results. For that we perform the Wilcoxon signed-rank test [62] comparing micro-averaged and macro-averaged F_1-measure of local and global feature selection approaches with transformation methods. Figure 4 shows the p-value of the Wilcoxon test over the 44 datasets reported (flat and hierarchical) for the five different local classifiers. This is done for different portions of the features

			J48		NaiveBayes		AdaBoost		kNN		LibSVM	
Dataset	D_1	D_2	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}
Arts1	0.339	0.237	1.525	1.690	1.176	1.543	1.315	1.450	1.478	1.600	1.537	1.574
bibtex	0.167	0.225	1.119	1.426	1.084	1.247	1.087	1.257	0.987	1.066	1.127	1.670
birds	0.282	0.284	1.048	1.209	1.346	1.343	1.199	1.317	1.022	1.038	1.654	1.270
Business1	0.265	0.238	1.029	1.444	0.994	1.408	1.003	1.240	1.041	1.452	1.021	1.463
CAL500	0.250	0.250	1.010	1.022	1.024	1.042	1.005	1.006	1.014	1.002	0.987	0.957
Computers1	0.242	0.233	1.242	1.824	1.431	1.794	1.178	1.644	1.249	1.660	1.244	2.051
Corel5k	0.113	0.225	1.606	1.033	1.820	1.554	1.130	1.004	1.357	1.053	2.972	1.002
delicious	0.146	0.227	1.872	1.297	1.570	1.642	1.895	1.134	1.650	1.273	1.989	1.295
Education1	0.285	0.245	1.583	1.649	0.853	1.291	1.431	1.461	1.569	1.874	1.622	1.844
emotions	0.380	0.365	1.032	1.047	1.007	1.012	1.040	1.067	1.022	1.023	0.953	0.936
enron	0.150	0.230	1.192	1.280	1.147	1.331	1.082	1.146	1.162	1.232	1.273	1.155
Entertainment1	0.357	0.256	1.703	1.729	1.088	1.472	1.851	1.906	1.492	1.463	1.737	1.728
flags	0.452	0.440	0.998	0.971	1.002	1.005	0.991	0.982	0.989	0.976	0.982	0.968
genbase	0.764	0.729	1.005	1.024	1.005	1.031	1.005	1.022	1.004	1.027	1.004	1.029
Health1	0.439	0.254	1.072	1.298	1.129	1.237	0.959	1.187	1.044	1.338	1.059	1.237
imdb	0.183	0.255	1.045	1.057	1.286	1.293	1.000	1.000	1.020	1.038	0.927	0.879
mediamill	0.144	0.228	1.032	0.866	0.968	0.958	1.007	0.450	0.995	1.045	1.028	1.042
medical	0.300	0.242	0.985	1.009	1.006	0.960	1.008	1.026	0.929	0.976	0.993	1.006
rcv1subset1	0.362	0.229	1.535	1.603	1.893	1.999	1.556	1.513	1.440	1.563	1.005	1.001
rcv1subset2	0.346	0.229	1.268	1.614	2.027	2.180	1.335	1.515	1.333	1.671	1.003	0.974
rcv1subset3	0.339	0.228	1.229	1.502	2.028	2.089	1.300	1.408	1.361	1.654	0.972	0.965
rcv1subset4	0.379	0.229	1.220	1.430	1.831	1.985	1.299	1.385	1.326	1.559	1.064	1.050
rcv1subset5	0.344	0.228	1.140	1.503	1.822	1.947	1.313	1.487	1.284	1.560	1.052	0.998
Recreation1	0.263	0.242	1.460	1.828	1.548	1.925	1.362	1.670	1.422	1.724	1.456	1.810
Reference1	0.464	0.242	1.164	1.177	1.276	1.132	1.085	1.034	1.138	1.207	1.132	1.137
scene	0.244	0.343	1.146	1.178	1.094	1.104	1.174	1.274	1.091	1.093	1.207	1.289
Science1	0.347	0.231	1.452	1.681	1.222	1.660	1.424	1.515	1.437	1.777	1.469	1.504
slashdot	0.381	0.239	1.576	1.293	1.533	1.470	1.214	1.125	1.394	1.295	1.706	1.342
Social1	0.334	0.245	1.041	1.965	1.291	1.736	0.959	1.515	1.058	1.850	1.044	2.000
Society1	0.301	0.251	1.456	2.207	1.245	1.540	1.475	1.873	1.392	1.814	1.448	2.064
tmc2007	0.188	0.241	1.161	1.676	1.133	1.799	1.152	1.749	1.160	1.580	1.145	1.748
yeast	0.168	0.267	1.020	1.059	1.009	1.049	1.038	1.123	0.999	1.003	1.047	1.166
AVG	0.304	0.269	1.249	1.393	1.309	1.462	1.215	1.296	1.214	1.359	1.277	1.317

Table 6: Comparison of local and global feature selection with *max* aggregation on flat multilabel datasets

selected, which is represented as percentage of total number of features in the horizontal axis.

For p-values under 0.05 the difference between local and global feature selection methods is statistically significant according to the Wilcoxon Test. We can observe that, for all local classifiers, the difference is highly significant with a p-value far below the 0.05 line for smaller numbers of features. In particular, for Naïve Bayes, this difference is the most significant amongst the evaluated local classifiers. Adaboost, LibSVM, and J48 also benefit significantly from the local feature selection approach, while IBk profits the least.

The results indicate that for all classifiers on a small set of selected features, the difference is the most significant, while for larger portions of selected features the significance is slightly lower. With that, we can confirm that when performing feature selection on transformation methods, the local approach is a better choice than the global method, especially when the portion of selected features is small.

			J48		NaiveBayes		AdaBoost		kNN		LibSVM	
Dataset	D_1	D_2	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}	$R_{\mu F_1}$	R_{mF_1}
aifb	0.346	0.383	1.024	1.166	1.187	1.135	1.025	1.235	1.024	1.180	1.023	1.213
cellcycle	0.163	0.224	1.115	1.002	1.087	1.002	1.385	1.006	1.008	1.005	1.141	1.000
church	0.344	0.279	1.031	0.999	1.226	1.010	1.068	1.000	0.845	1.049	1.141	1.00
dbpedia-yago	0.259	0.299	1.011	1.160	0.998	1.231	1.002	1.121	1.014	0.898	0.953	0.331
derisi	0.151	0.228	1.164	1.002	1.126	1.010	1.098	1.000	1.031	0.990	0.899	1.002
eisen	0.086	0.225	1.117	0.999	1.137	1.001	1.268	0.995	1.019	1.013	1.090	1.003
gasch2	0.192	0.224	1.077	0.997	1.036	1.025	1.079	1.003	1.007	1.034	1.037	0.987
mutagenesis	0.150	0.333	1.194	1.130	1.201	1.127	1.191	1.118	1.196	1.139	0.782	1.022
nell	0.198	0.244	1.021	1.072	1.017	1.027	1.000	1.024	1.019	1.077	1.018	1.034
pheno	0.242	0.256	1.150	1.002	1.306	0.995	1.229	1.003	1.169	1.005	1.202	1.000
struc	0.187	0.224	1.082	0.962	1.053	0.840	1.045	0.998	1.046	0.986	1.260	1.001
wikidata	0.206	0.389	1.009	1.027	1.019	1.148	1.004	1.020	1.011	1.031	1.005	1.009
AVG	0.210	0.276	1.083	1.043	1.116	1.046	1.116	1.044	1.032	1.034	1.045	0.967

Table 7: Comparison of local and global feature selection with *max* aggregation on. popular hierarchical multilabel datasets

	J48	Naive Bayes	AdaBoost	kNN	LibSVM	AVG
corr(D_1,μF_1)	−0.5364	−0.4791	−0.5348	−0.5273	−0.3909	−0.4937
corr(D_2,μF_1)	−0.5425	−0.4880	−0.5264	−0.5405	−0.4120	−0.5019
corr(D_1,mF_1)	−0.6069	−0.5356	−0.5769	−0.5190	−0.4580	−0.5393
corr(D_2,mF_1)	−0.6341	−0.5427	−0.5849	−0.4982	−0.4905	−0.5501
corr(D_1,$hamm$)	0.3808	0.2178	0.3576	0.3778	0.4476	0.3563
corr(D_2,$hamm$)	0.4119	0.2209	0.3457	0.4299	0.4955	0.3808

Table 8: Correlations between D_1 and D_2 measures and the ratio between global and local feature selection approaches for different evaluation measures

We also point out that the choice of local classifier can influence the benefits of local over global feature selection methods.

We also perform the same test on flat and hierarchical datasets separately with results in Figures 5 and 6 respectively. Overall the results are roughly similar with kNN and LibSVM having the higher p-values in both the flat and hierarchical datasets. Overall, because the number of datasets considered is smaller, the p-values are higher. Especially for the hierarchical case, the number of datasets is 12, which is too small for the Wilcoxon test, which typically requires a minimum number of 20.

Another investigation we made is how strong the correlation between the variance of local feature selection and the performance gain over global selection is. The hpyothesis is that for problems where there are strong differences in the best local feature sets, local feature selection will lead to a more significant performance improvement. In order to test that hypothesis, we measure the correlation of D_1 and D_2, which capture the variety of features in the different local sets, with the ratio between the performance measures of classifiers with local and global feature selection approaches. For a given dataset and local classifier we compute the values of D_1, D_2, and the values of μF_1, mF_1 and $hamm$ with local feature selection divided by the correspondent measure values with global feature selection for the different numbers of selected features k. Then the correlation is calculated for every dataset and with the different local classifiers evaluated in this paper. Table 8

shows the correlation values across all the datasets with different local classifiers, and, in the last column, the correlation over all classifiers and datasets.

For μF_1 and mF_1, ratios greater than one mean that the local approach performed better, for $hamm$ ratios less than one mean that the local approach performed better. For both measures D_1 and D_2, the smaller the value, the more distinct the local feature sets are from the global feature sets. Therefore, the negative values of corr(D_1,μF_1), corr(D_1,mF_1), corr(D_2,μF_1) and corr(D_2,mF_1), and the positive values of corr(D_1,$hamm$) and corr(D_2,$hamm$) show that the more local and global feature sets differ, the better the local approach will be in comparison to the global approach.

In all cases, the correlations are significant, which confirms the original hypothesis. Furthermore, we have observed that the correlation between D_1 and D_2 is 0.957, i.e., both measures are essentially very similar in measuring the variety of the local feature selection.

6 Conclusion and Future Work

In this paper, we have presented an experimental comparison of global and local feature selection methods on transformation methods for multilabel classification with flat and hierarchical labels. Although transformation methods are very popular, and allow the feature selection to be performed locally on each transformed dataset, this alternative has not been very extensively explored in the literature.

We performed an extensive evaluation of local and global feature selection approaches on transformation based multilabel classifiers. The results indicate that the local approach performs consistently better than the global approach in terms of predictive performance. Both approaches have similar runtimes and scalability on flat multilabel classification, and for hierarchically structured labels, the local approach scales better than the global approach. Based on these results, the local feature selection approach is considered superior to the global approach when using transformation methods for multilabel classification.

When comparing the local feature selection approach with transformation methods to adaptation methods with global approaches using a global feature selection method, the results are not generally conclusive. However, for many of the datasets, the local approach also performs better in that case.

So far, we have only considered binary relevance as a transformation technique. Performing a similar comparison with other transformation methods, such as classifier chains, would be interesting for future work. Furthermore, it would be interesting if it is possible to compile a better global feature set for adaptation methods, using local feature selection on a number of transformed problems.

Another possibility which the use of transformation brings is the use of different local classifiers for different transformed subproblems. The idea is similar to Vapnik's locality principle, however, the locality is in the class level instead of input space level.

Acknowledgements The work presented in this paper has been partly supported by the Ministry of Science, Research and the Arts Baden-Württemberg in the project SyKo^{2}W^2 (Synthesis of Completion and Correction of Knowledge Graphs on the Web).

References

1. Bi, W., Kwok, J.T.: Multi-label classification on tree- and dag-structured hierarchies. In: L. Getoor, T. Scheffer (eds.) Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 17–24. ACM, New York, NY, USA (2011). URL http://www.icml-2011.org/papers/10_icmlpaper.pdf
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International journal on semantic web and information systems **5**(3), 1–22 (2009)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Web Semantics **7**(3), 154–165 (2009)
4. Blockeel, H., Raedt, L.D., Ramong, J.: Top-down induction of clustering trees. In: In Proceedings of the 15th International Conference on Machine Learning, pp. 55–63. Morgan Kaufmann (1998)
5. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition **37**(9), 1757 – 1771 (2004). DOI DOI:10.1016/j.patcog.2004.03.009. URL http://www.sciencedirect.com/science/article/B6V14-4CF14JX-1/2/a17089f241a1d23f218e55d2c8d9f763
6. Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S., Hadley, A., Betts, M., Fern, X., Irvine, J., Neal, L., Thomas, A., Fodor, G., Tsoumakas, G., Ng, H.W., Nguyen, T.N.T., Huttunen, H., Ruusuvuori, P., Manninen, T., Diment, A., Virtanen, T., Marzat, J., Defretin, J., Callender, D., Hurlburt, C., Larrey, K., Milakov, M.: The 9th annual mlsp competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In: Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on, pp. 1–8 (2013). DOI 10.1109/MLSP.2013.6661934
7. Brucker, F., Benites, F., Sapozhnikova, E.: An Empirical Comparison of Flat and Hierarchical Performance Measures for Multi-Label Classification with Hierarchy Extraction, pp. 579–589. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). DOI 10.1007/978-3-642-23851-2_59. URL http://dx.doi.org/10.1007/978-3-642-23851-2_59
8. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr, E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proceedings of the third ACM international conference on Web search and data mining, pp. 101–110. ACM (2010)
9. Cerri, R., Pappa, G.L., de Leon Ferreira de Carvalho, A.C.P., Freitas, A.A.: An extensive evaluation of decision tree-based hierarchical multilabel classification methods and performance measures. Computational Intelligence **31**(1), 1–46 (2015). DOI 10.1111/coin.12011. URL http://dx.doi.org/10.1111/coin.12011
10. Cesa-bianchi, N., Zaniboni, L., Collins, M.: Incremental algorithms for hierarchical classification. In: Journal of Machine Learning Research, pp. 31–54. MIT Press (2004)
11. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD'01, pp. 42–53 (2001)
12. Costa, E., Lorena, A., Carvalho, A., Freitas, A.: A review of performance evaluation measures for hierarchical classifiers. In: C. Drummond, W. Elazmeh, N. Japkowicz, S. Macskassy (eds.) Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop, AAAI Technical Report WS-07-05, pp. 182–196. AAAI Press (2007). URL http://www.cs.kent.ac.uk/pubs/2007/2611
13. Dash, M., Liu, H.: Feature selection for classification. Intelligent data analysis **1**(3), 131–156 (1997)
14. Dimitrovski, I., Kocev, D., Loskovska, S., Dzeroski, S.: Hierarchical annotation of medical images. Pattern Recognition **44**(10-11), 2436–2449 (2011). URL http://dblp.uni-trier.de/db/journals/pr/pr44.html#DimitrovskiKLD11
15. Diplaris, S., Tsoumakas, G., Mitkas, P.A., Vlahavas, I.P.: Protein classification with multiple algorithms. In: P. Bozanis, E.N. Houstis (eds.) Panhellenic Conference on Informatics, *Lecture Notes in Computer Science*, vol. 3746, pp. 448–456. Springer (2005). URL http://dblp.uni-trier.de/db/conf/pci/pci2005.html#DiplarisTMV05
16. Doquire, G., Verleysen, M.: Feature selection for multi-label classification problems. In: J. Cabestany, I. Rojas, G.J. Caparrós (eds.) Advances in Computational Intelligence - 11th International Work-Conference on Artificial Neural Networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings, Part I, *Lecture Notes in Computer Science*, vol. 6691, pp. 9–16. Springer (2011). DOI 10.1007/978-3-642-21501-8_2

17. Duygulu, P., Barnard, K., Freitas, J.F.G.d., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings of the 7th European Conference on Computer Vision-Part IV, ECCV '02, pp. 97–112. Springer-Verlag, London, UK, UK (2002). URL http://dl.acm.org/citation.cfm?id=645318.649254
18. Eisner, R., Poulin, B., Szafron, D., Lu, P., Greiner, R.: Improving protein function prediction using the hierarchical structure of the gene ontology. In: Proc. IEEE CIBCB (2005)
19. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: T.G. Dietterich, S. Becker, Z. Ghahramani (eds.) Advances in Neural Information Processing Systems 14 (NIPS-01), pp. 681–687 (2002)
20. Fagni, T., Sebastiani, F.: On the selection of negative examples for hierarchical text categorization. In: In Proceedings of The 3rd Language Technology Conference, pp. 24–28 (2007)
21. Huda, S., Yearwood, J., Stranieri, A.: Hybrid wrapper-filter approaches for input feature selection using maximum relevance-minimum redundancy and artificial neural network input gain measurement approximation (annigma). In: Proceedings of the Thirty-Fourth Australasian Computer Science Conference - Volume 113, ACSC '11, pp. 43–52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2011). URL http://dl.acm.org/citation.cfm?id=2459296.2459302
22. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: In: Proceedings of the ECML/PKDD-08 Workshop on Discovery Challenge (2008)
23. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'92, pp. 129–134. AAAI Press (1992). URL http://dl.acm.org/citation.cfm?id=1867135.1867155
24. Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: in Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05 (2005)
25. Kiritchenko, S., Matwin, S., Nock, R., Famili, A.F.: Learning and evaluation in the presence of class hierarchies: Application to text categorization. In: Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence, AI'06, pp. 395–406. Springer-Verlag, Berlin, Heidelberg (2006). DOI 10.1007/11766247_34. URL http://dx.doi.org/10.1007/11766247_34
26. Kosmopoulos, A., Paliouras, G., Androutsopoulos, I.: The effect of dimensionality reduction on large scale hierarchical classification. In: Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings, pp. 160–171 (2014). DOI 10.1007/978-3-319-11382-1_16. URL http://dx.doi.org/10.1007/978-3-319-11382-1_16
27. Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., Androutsopoulos, I.: Evaluation measures for hierarchical classification: a unified view and novel approaches. Data Mining and Knowledge Discovery **29**(3), 820–865 (2015). DOI 10.1007/s10618-014-0382-x. URL http://dx.doi.org/10.1007/s10618-014-0382-x
28. Kozachenko, L.F., Leonenko, N.N.: Sample estimate of the entropy of a random vector. Probl. Inf. Transm. **23**(1-2), 95–101 (1987)
29. Labrou, Y.K.: Yahoo as an Ontology - using Yahoo categories to describe documents,. In: Proceedings of the 1999 ACM Conference on Information and Knowledge Management (CIKM'99) (1999)
30. de Lannoy, G., Franois, D., Verleysen, M.: Class-specific feature selection for one-against-all multiclass svms. In: ESANN (2011). URL http://dblp.uni-trier.de/db/conf/esann/esann2011.html#LannoyFV11
31. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004). URL http://dl.acm.org/citation.cfm?id=1005332.1005345
32. Madjarov, G., Kocev, D., Gjorgjevikj, D., Deroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recogn. **45**(9), 3084–3104 (2012)
33. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: A Knowledge Base from Multilingual Wikipedias. In: Conference on Innovative Data Systems Research (2015)
34. Melo, A., Paulheim, H., Völker, J.: Type prediction in rdf knowledge bases using hierarchical multilabel classification. In: 6th International Conference on Web-Intelligence, Mining and Semantics (WIMS) (2016)

35. Molina, L.C., Belanche, L., Nebot, À.: Feature selection algorithms: A survey and experimental evaluation. In: International Conference on Data Mining (ICDM), pp. 306–313. IEEE (2002)
36. Opitz, D.W.: Feature selection for ensembles. In: In Proceedings of 16th National Conference on Artificial Intelligence (AAAI, pp. 379–384. Press (1999)
37. Otero, F.E., Freitas, A.A., Johnson, C.G.: A hierarchical classification ant colony algorithm for predicting gene ontology terms. In: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, EvoBIO '09, pp. 68–79. Springer-Verlag, Berlin, Heidelberg (2009). DOI 10.1007/978-3-642-01184-9_7. URL http://dx.doi.org/10.1007/978-3-642-01184-9_7
38. Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.R., Galinari, P.: Lshtc: A benchmark for large-scale text classification. CoRR **abs/1503.08581** (2015). URL http://arxiv.org/abs/1503.08581
39. Paulheim, H., Fürnkranz, J.: Unsupervised Generation of Data Mining Features from Linked Open Data. In: International Conference on Web Intelligence, Mining, and Semantics (WIMS'12) (2012)
40. Pestian, J.P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07, pp. 97–104. Association for Computational Linguistics, Stroudsburg, PA, USA (2007). URL http://dl.acm.org/citation.cfm?id=1572392.1572411
41. Qu, H., Zhang, S., Liu, H., Zhao, J.: A multi-label classification algorithm based on label-specific features. Wuhan University Journal of Natural Sciences **16**(6), 520–524 (2011). DOI 10.1007/s11859-011-0791-2. URL http://dx.doi.org/10.1007/s11859-011-0791-2
42. Read, J.: A pruned problem transformation method for multi-label classification. In: In: Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS, pp. 143–150 (2008)
43. Read, J., Bifet, A., Holmes, G., Pfahringer, B.: Scalable and efficient multi-label classification for evolving data streams. Machine Learning **88**(1-2), 243–272 (2012). DOI 10.1007/s10994-012-5279-6. URL http://dx.doi.org/10.1007/s10994-012-5279-6
44. Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets. In: ICDM, pp. 995–1000. IEEE Computer Society (2008). URL http://dblp.uni-trier.de/db/conf/icdm/icdm2008.html#ReadPH08
45. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, ECML PKDD'09, pp. 254–269 (2009)
46. Ristoski, P., Paulheim, H.: A comparison of propositionalization strategies for creating features from linked open data. In: LD4KD (2014)
47. Ristoski, P., de Vries, G.K.D., Paulheim, H.: A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In: International Semantic Web Conference. Springer (2016)
48. Saeys, Y., Abeel, T., Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Part II, ECML PKDD '08, pp. 313–325. Springer-Verlag, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-87481-2_21. URL http://dx.doi.org/10.1007/978-3-540-87481-2_21
49. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning **39**(2/3), 135–168 (2000)
50. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Discov. **22**(1-2), 31–72 (2011). DOI 10.1007/s10618-010-0175-9
51. Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S., Dzeroski, S.: Relieff for hierarchical multi-label classification. In: A. Appice, M. Ceci, C. Loglisci, G. Manco, E. Masciari, Z.W. Ras (eds.) New Frontiers in Mining Complex Patterns - Second International Workshop, NFMCP 2013, Held in Conjunction with ECML-PKDD 2013, Prague, Czech Republic, September 27, 2013, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 8399, pp. 148–161. Springer (2013). DOI 10.1007/978-3-319-08407-7_10. URL http://dx.doi.org/10.1007/978-3-319-08407-7_10
52. Spolaôr, N., Tsoumakas, G.: Evaluating feature selection methods for multi-label text classication. In: A.N. Ngomo, G. Paliouras (eds.) Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop

of Conference and Labs of the Evaluation Forum 2013 (CLEF 2013) , Valencia, Spain, September 27th, 2013., *CEUR Workshop Proceedings*, vol. 1094. CEUR-WS.org (2013)
53. Srivastava, A., Zane-Ulman, B.: Discovering recurring anomalies in text reports regarding complex space systems. In: Proceedings of the 2005 IEEE Aerospace Conference (2005)
54. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.P.: Multi-label classification of music into emotions. In: J.P. Bello, E. Chew, D. Turnbull (eds.) ISMIR, pp. 325–330 (2008). URL http://dblp.uni-trier.de/db/conf/ismir/ismir2008.html#TrohidisTKV08
55. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Int J Data Warehousing and Mining **2007**, 1–13 (2007)
56. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering (1). DOI 10.1109/TKDE.2010.164
57. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08) (2008)
58. Turnbull, D., Barrington, L., Torres, D.A., Lanckriet, G.R.G.: Semantic annotation and retrieval of music and sound effects. IEEE Transactions on Audio, Speech and Language Processing **16**(2), 467–476 (2008). URL http://dblp.uni-trier.de/db/journals/taslp/taslp16.html#TurnbullBTL08
59. Ueda, N., Saito, K.: Parametric mixture models for multi-labeled text. In: S. Becker, S. Thrun, K. Obermayer (eds.) Advances in Neural Information Processing Systems 15, pp. 737–744. MIT Press (2003). URL http://papers.nips.cc/paper/2244-parametric-mixture-models-for-multi-labeled-text.pdf
60. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Mach. Learn. **73**(2), 185–214 (2008)
61. Vrandečić, D., Krötzsch, M.: Wikidata: a Free Collaborative Knowledge Base. Communications of the ACM **57**(10), 78–85 (2014)
62. Wilcoxon, F.: Individual Comparisons by Ranking Methods. Biometrics Bulletin **1**(6), 80–83 (1945). DOI 10.2307/3001968. URL http://dx.doi.org/10.2307/3001968
63. Zhang, M., Wu, L.: Lift: Multi-label learning with label-specific features. IEEE Trans. Pattern Anal. Mach. Intell. **37**(1), 107–120 (2015). DOI 10.1109/TPAMI.2014.2339815. URL http://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2339815
64. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014). DOI 10.1109/TKDE.2013.39. URL http://dx.doi.org/10.1109/TKDE.2013.39
65. Zhang, M.L., Peña, J.M., Robles, V.: Feature selection for multi-label naive bayes classification. Inf. Sci. **179**(19), 3218–3229 (2009). DOI 10.1016/j.ins.2009.06.010. URL http://dx.doi.org/10.1016/j.ins.2009.06.010
66. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
67. Zhu, Z., Ong, Y.S., Dash, M.: Wrapper-filter feature selection algorithm using a memetic framework. IEEE Transactions on Systems, Man, and Cybernetics, Part B **37**(1), 70–76 (2007). URL http://dblp.uni-trier.de/db/journals/tsmc/tsmcb37.html#ZhuOD07

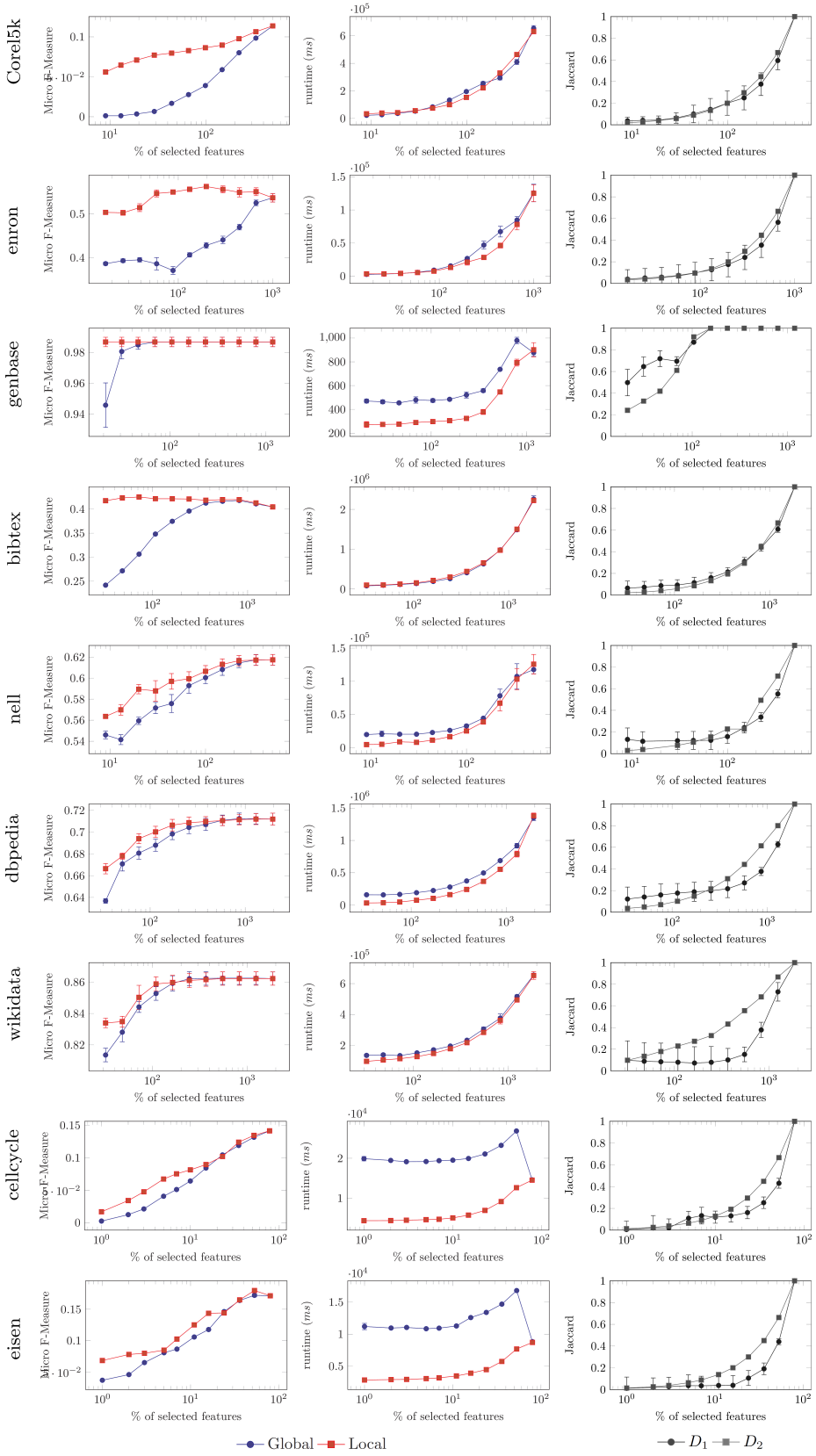


Fig. 1: Global vs local feature selection comparison with J48 and measures D_1 and D_2.

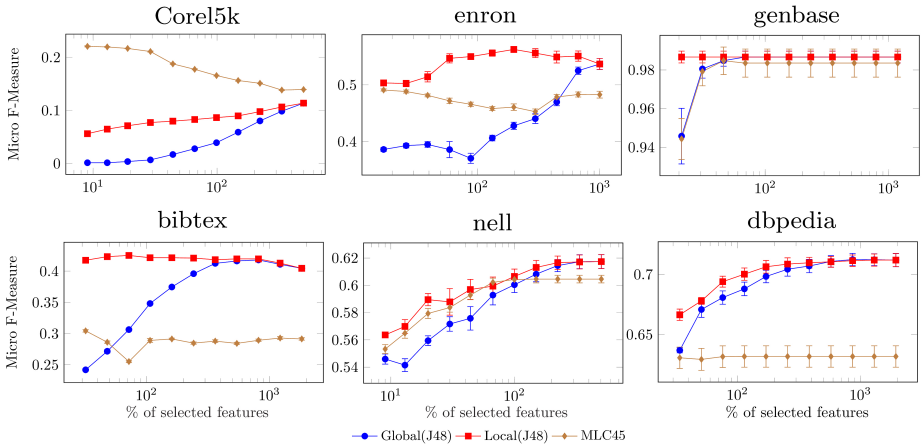


Fig. 2: Comparison of MLC4.5 with adaptation methods

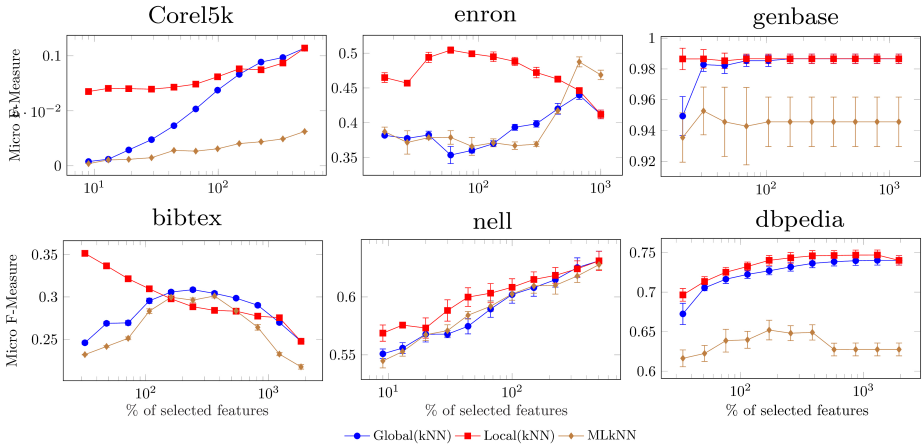
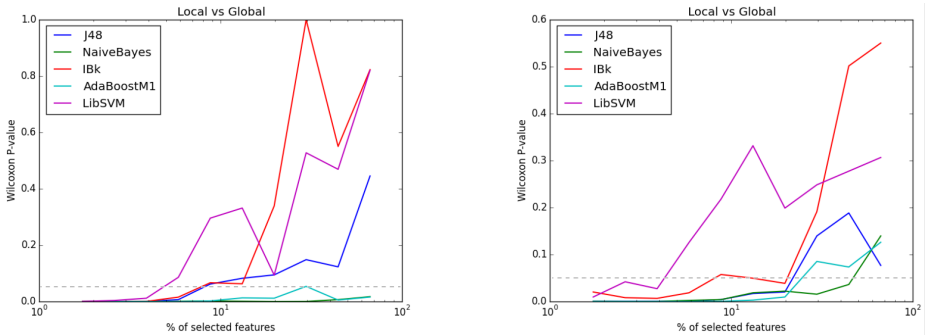
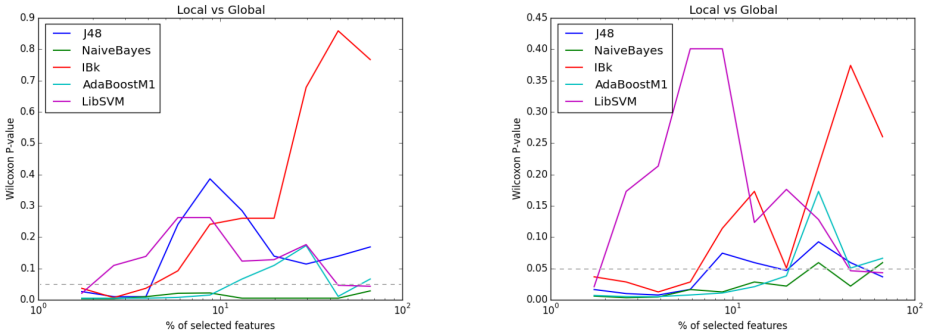


Fig. 3: Comparison of MLkNN with adaptation methods



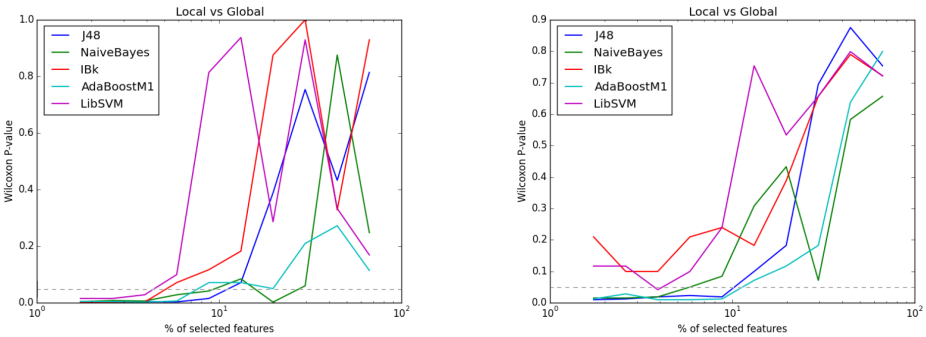
(a) Micro-averagedF-Measure (b) Macro-averagedF-Measure

Fig. 4: Wilcoxon test on flat and hierarchical multilabel datasets



(a) Micro-averagedF-Measure (b) Macro-averagedF-Measure

Fig. 5: Wilcoxon test on flat multilabel datasets



(a) Micro-averagedF-Measure (b) Macro-averagedF-Measure

Fig. 6: Wilcoxon test on hierarchical multilabel datasets