Evaluation of Distance Measures for Hierarchical Multi-Label Classification in Functional Genomics

Darko Aleksovski, Dragi Kocev, and Sašo Džeroski

Department of Knowledge Technologies, Jozef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
{Darko.Aleksovski, Dragi.Kocev, Saso.Dzeroski}@ijs.si

Abstract. Hierarchical multi-label classification (HMLC) is a variant of classification where instances may belong to multiple classes that are organized in a hierarchy. The approach we used is based on decision trees and is set in the predictive clustering trees framework (PCTs), which is implemented in the CLUS system. In this work, we are investigating how different distance measures for hierarchies influence the predictive performance of the PCTs. The distance measures that we consider include weghted Euclidean distance, Jaccard, SimGIC and ImageCLEF distance. We use datasets from the area of functional genomics to evaluate the performance of the PCTs with different distances. The datasets describe different functions of the genes in the genomes of two well-studied organisms: S. Cerevisiae and A. Thaliana. We use precision-recall curves as an evaluation metric for the predictive performance. The results from the Friedman test for statistical significance suggest that there is no statistical significance in the performance.

1 Introduction

Hierarchical multi-label classification (HMLC) is an extension of binary classification where an instance can be labeled with multiple classes that are organized in a hierarchy. Additionally, when an instance is assigned to some class it should also be assigned to all its superclasses. The main applications of HMLC are in the areas of gene function prediction [1, 2], text classification [3] and image classification [4].

There are two general approaches for solving the HMLC task: decomposing this task to simpler single-target tasks and solving them with basic classification approaches or using the hierarchical structure and trying to make predictions for the whole hierarchy. An example for the first approach is learning a binary classifier for each class and an example for the second approach is to learn a single model which predicts all the classes simultaneously. The second group of algorithms has some advantages over the first group [5–7]. First, they exploit the dependencies between the components and as a result have better predictive performance. Second, they are more efficient: it can easily happen that the number of components in the output is very large (e.g., hierarchies in functional genomics) in which case running a learning algorithm for each component is not feasible. Third, they produce a single model valid for the structure as a whole, as compared to the many models, each valid just for one given component: the single model is usually much more concise.

In this study, we focus on the latter approach: we learn a single Predictive Clustering Tree [6] to make a prediction for the complete hierarchy. The PCTs were extended to the HMLC task by Vens at al. [2], and they use weighted Euclidean distance as a distance measure between two hierarchies. Here, we consider three additional distance measures (Jaccard distance [13], SimGIC [8] and ImageCLEF [9]). We implemented the distance measures in the CLUS system and we evaluated them on several datasets from functional genomics.

The remainder of this paper is organized as follows: Section 2 describes the PCTs algorithm and the proposed distance measures and Section 3 presents the datasets that we used for evaluation. Section 4 gives the experimental design, while Section 5 presents the obtained results. Finally, conclusions and points for further work are presented in Section 6.

2 Methodology

2.1 Predictive Clustering Trees

The approach we use is based on decision trees and is set in the predictive clustering trees (PCTs) framework. This framework views a decision tree as a hierarchy of clusters, where the top node correspond to a cluster containing all data, which are recursively partitioned into smaller clusters while building the tree from top to bottom. The PCT framework is implemented in the CLUS system (available at `http://www.cs.kuleuven.be/~dtai/clus`).

PCTs can be constructed with a standard "top-down induction of decision trees" (TDIDT) algorithm. The heuristic that is used for selecting the tests is the reduction in variance caused by partitioning the instances. Maximizing the variance reduction maximizes cluster homogeneity and improves predictive performance. With appropriate instantiation of the variance and prototype function the PCTs can handle different types of data, e.g., multiple targets [11] or time series [12]. A detailed description of the PCT framework can be found in [6].

In the remainder of this sub-section, we explain how PCTs were instantiated for the HMLC task, namely we present the internal representation of the hierarchy, annotation of the examples, making a prediction and we give an example of PCT for HMLC. The hierarchy is represented as a 0/1 vector: if a given example is labeled with some label, then for that label the value in the vector is set to 1, otherwise it is set to 0. The annotation scheme is presented in Figure 1. The example is annotated with the following labels: B, B.1, C, D, D.2 and D.3. If an example belongs to a node, then it belongs also to all the node's parents.

The reduction of variance is calculated using the following equation:

$$Var(S) = \frac{\sum_i d(v_i, \bar{v})^2}{|S|} \tag{1}$$

where S denotes the set of examples over which the variance is calculated, $\bar{v}$ is the mean label, and v_i is a label of the example. The sum goes over all possible labels. The mean label is calculated as the mean of the vectors of the examples from S, in that node.

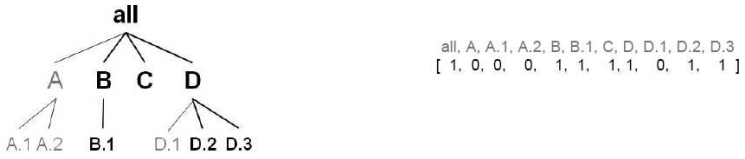

Fig. 1. A hierarchy (left) with an example annotated to it (subset of the hierarchy shown bold); the example's vector representation (right).

Different distance measures can be used in equation 1. In the original implementation from [2], the Euclidean distance is used for PCT induction. In this work, we use three other distances that can be used in the context of HMLC. We present and explain the distances in the next sub-sections.

The PCTs at every leaf of the tree contain probabilities (a probability vector) of an instance belonging to each class in the hierarchy. To obtain a prediction, a threshold is applied to the probability vector. If a given label has a bigger probability than the threshold, the example is annotated with that label and its parents. An example of a PCT for HMLC is presented in Figure 2. It looks like an ordinary decision tree, but in the leaves, instead of the majority class, it contains as prediction the annotation for the examples from that node. Note that for some of the leaves have prediction: "none". This is because no annotations could be assigned for the used threshold value (i.e., the probabilities for example belonging to the classes are lower than the specified threshold).

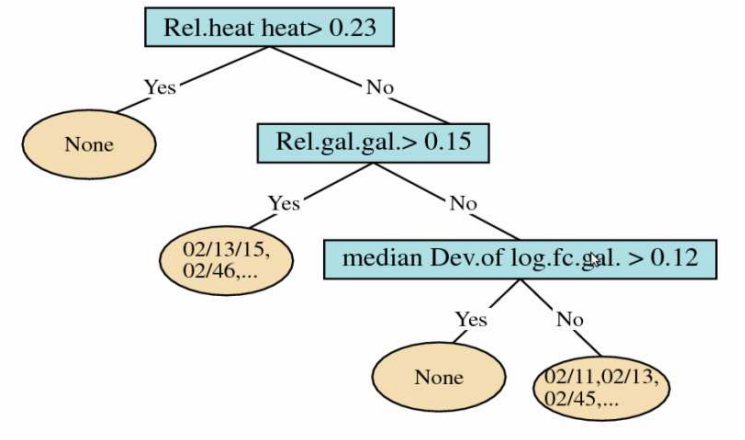

Fig. 2. An example PCT for HMLC, obtained with a given threshold, for the 'church' dataset with FunCat annotation.

2.2 Weighted Euclidean distance

The Euclidean distance is a well known distance measure. In order to include knowledge about the hierarchy, Vens et al. [2] have introduced a weighting scheme that depends on the depth of the node in the hierarchy. The weighted Euclidean distance can be calculated using the following equation:

$$d(v_1, v_2) = \sqrt{\sum_i w(c_i)(v_{1,i} - v_{2,i})^2} \tag{2}$$

where $v_{k,i}$ is the i'th component of the class vector v_k of an instance x_k. The function $w(c)$ is denoted as the weighting scheme and the default instantiation here is to apply a weight to each class label c according to the depth of this class in the hierarchy (e.g., $w(c) = w_0^{depth(c)}$ with $0 < w_0 < 1$). With this parameter, the user can control the influence of the top classes on the distance.

Let us consider two examples: (x_1, S_1) and (x_2, S_2), which are annotated with the hierarchy from Figure 1: S_1 = {B, B.1, C, D, D.2, D.3} and S_2 = {D, D.2, D.3}. Using the vector representation presented above, the weighted Euclidean distance is:

$$d(S_1, S_2) = \sqrt{w_0 + w_0^2 + w_0} \tag{3}$$

The weighting function described here is only one of the possible weighting schemes that can be used. Others weighting schemes are described in [2] and it is recommended to use weighting.

2.3 Jaccard distance

The Jaccard distance [13] (which can also be found in the literature as Union-intersection distance/score) can be calculated using the following equation:

$$d_{Jaccard}(v_1, v_2) = 1 - \frac{\sum_{c \in labels(v_1) \cap labels(v_2)} w(c)}{\sum_{c \in labels(v_1) \cup labels(v_2)} w(c)} \tag{4}$$

where v_1 and v_2 are class vectors, $labels(v)$ presents the elements from v, c is a class node from v_k. This distance actually is taking into account the ratio between the sum of the weights of the joint annotations and the sum of the weights of the annotations of both examples. As in the case of weighted Euclidean distance, we use the same exponential weighting scheme.

Let us consider the same example as for the weighted Euclidean distance. The Jaccard distance for the two examples (x_1, S_1) and (x_2, S_2) will be:

$$d(S_1, S_2) = 1 - \frac{w_0^0 + w_0 + w_0^2 + w_0^2}{w_0^0 + w_0 + w_0^2 + w_0 + w_0 + w_0^2 + w_0^2} = 1 - \frac{1 + w_0 + 2w_0^2}{1 + w3_0 + 3w_0^2} \tag{5}$$

2.4 SimGIC distance

The Similarity for Graph Information Content (SimGIC) distance [8] is similar to the Jaccard distance, but instead of summing the weights of the labels, it sums up their information content [2].

$$d_{SimGIC}(v_1, v_2) = 1 - \frac{\sum_{c \in labels(v_1) \cap labels(v_2)} IC(c)}{\sum_{c \in labels(v_1) \cup labels(v_2)} IC(c)} \tag{6}$$

The variables here are the same as for the Jaccard distance, and $IC(c)$ is the Information Content for a class node c, which is calculated as:

$$IC(c) = -logp(c) \tag{7}$$

Here $p(c)$ is the probability of usage of the label in the dataset, which is calculated as the frequency of the label in the dataset. Let us consider the example from the weighted Euclidean and Jaccard distance sub-sections. The SimGIC distance for the two examples (x_1, S_1) and (x_2, S_2) will be:

$$d(S_1, S_2) = 1 - \frac{-log(P(all)P(D)P(D.2)P(D.3))}{-log(P(all)P(B)P(B.1)P(C)P(D)P(D.2)P(D.3))} \tag{8}$$

The ImageCLEF distance is derived from the evaluation score of the ImageCLEF annotation task [9]. This distance can be calculated using the following formula:

$$d_{ImageCLEF}(v_1, v_2) = 1 - \frac{\sum_{c \in labels(v_1) \cap labels(v_2)} \frac{1}{siblings(c)+1} \frac{1}{depth(c)}}{\sum_{c \in labels(v_1) \cup labels(v_2)} \frac{1}{siblings(c)+1} \frac{1}{depth(c)}} \tag{9}$$

where $siblings(c)$ denotes the number of siblings of the class node c in the hierarchy and $depth(c)$ is the depth of the class node c (the root node is omitted in the calculations).

Let us consider the same example as for the weighted Euclidean distance. The ImageCLEF distance for the two examples (x_1, S_1) and (x_2, S_2) will be:

$$d(S_1, S_2) = 1 - \frac{\frac{1}{4}\frac{1}{1} + \frac{1}{3}\frac{1}{2} + \frac{1}{3}\frac{1}{2}}{\frac{1}{4}\frac{1}{1} + \frac{1}{1}\frac{1}{2} + \frac{1}{4}\frac{1}{1} + \frac{1}{4}\frac{1}{1} + \frac{1}{3}\frac{1}{2} + \frac{1}{3}\frac{1}{2}} = \frac{12}{19} \tag{10}$$

2.5 Adaptations of the distance measures for DAGs

The variance (equation 1) is computed using the distance between the class vectors, where a class c's weight $w(c)$ depends on its depth in the class hierarchy (e.g., $w(c) = w_0^{depth(c)}$ with $0 < w_0 < 1$). When the hierarchy structures the classes in the form of a directed acyclic graph (DAG), the depth of a class is not unique since it can have more than one path to a top-level class. An approach was chosen with rewriting the equation $w(c) = w_0^{depth(c)}$ to its recurrent form $w(c) = w_0 w(par(c))$, where $par(c)$ is

the parent class of c. Using the equation in this form along with an aggregation function (like sum, min, max, average) several alternatives are possible. In this work we chose to use the average as aggregation function, as recommended in [2]. So, the weighting scheme for DAGs can be defined as follows:

$$w(c) = w_0 avg_j w(par_j(c)) \tag{11}$$

3 Data Description

In this section, we describe the datasets that we used to evaluate the distance measures. We used sixteen datasets from the domain of functional genomics. The datasets represent different aspects of the genes in the genome of Saccharomyces Cerevisiae (12 of the datasets) and Arabidopsis Thaliana (4 of the datasets).

We consider two annotation schemes: FunCat [14] which is a tree-structured class hierarchy and the Gene Ontology (GO) [15], which forms a hierarchy using a directed acyclic graph: each term can have multiple parents (to be more precise, GO's "is-a" relationship between terms is used here).

The basic properties of the datasets are presented in Table 1. The number of examples in each dataset ranges from 1592 to 11763, the number of attributes from 27 to 19628, and the number of nodes in the hierarchy from 250 to 4125.

The datasets include different types of bioinformatic data. The 'pheno' dataset contains information about the phenotype; 'church' and 'eisen' contain data about the expression levels as measured with microarray chips. The 'scop' dataset contains the predicted SCOP class, while 'struc' has the predicted secondary structure. The protein pattern annotations are available in the 'interpro' datasets. Datasets 'spo', 'cellcycle', 'derisi', 'gasch1', 'gasch2' contain microarray data - expression levels of genes of the yeast genome. A more detailed description of the datasets can be found in [10, 2].

4 Experimental design

4.1 Evaluation measures

To measure the predictive performance of the algorithm with the different distance measures we will use Precision-Recall (PR) curves. These curves are obtained by plotting the precision and recall using different thresholds for the obtained probability vectors from the PCTs. Precision is the proportion of positive predictions that are correct, and recall is the proportion of positive examples that are correctly predicted positive. That is,

$$Prec = \frac{TP}{TP + FP} \qquad Prec = \frac{TP}{TP + FN} \tag{12}$$

with TP the number of true positives (correctly predicted positive examples), FP the number of false positives (positive predictions that are incorrect), and FN the number of false negatives (positive examples that are incorrectly predicted negative). Note that these measures ignore the number of correctly predicted negative examples.

Table 1. Basic properties of the used datasets.

	Dataset	Annotation	Number of instances	Number of discrete attributes	Number of continuous attributes	Number of nodes in hierarchy
Saccharomyces Cerevisiae	cellcycle	FunCat	3766	0	77	499
	church	FunCat	3764	1	26	499
	derisi	FunCat	3733	0	63	499
	eisen	FunCat	2425	0	79	461
	gasch1	FunCat	3733	0	173	499
	gasch2	FunCat	3788	0	52	499
	pheno	FunCat	1592	69	0	455
	spo	FunCat	3711	3	77	499
	struc	FunCat	3851	0	19628	499
	church	Gene Ontology	3764	1	26	4125
	eisen	Gene Ontology	2425	0	79	4125
	pheno	Gene Ontology	1592	69	0	3127
Arabidopsis Thaliana	interpro	FunCat	3719	2815	0	263
	scop	FunCat	3097	0	2003	250
	struc	FunCat	3719	14804	0	263
	interpro	Gene Ontology	11763	2815	0	629

The reason why Precision-Recall based evaluation is chosen in this context instead of the ROC analysis, which is more popular, was the following. In functional genomics datasets similar to the ones described and used here, typically only a few genes have been annotated to have a particular function (a particular class in the class hierarchy). This implies that one has to deal with a strongly skewed class distribution were the number of negative instances by far exceeds the number of positive ones [2]. There is a strong interest for correctly predicting the positive instances (that an instance has a given label), rather than the negative ones. ROC curves can present an overly optimistic view of the algorithm's performance (giving rise to a low false positive rate).

We use two approaches to calculate the AUPRC: area under the average PR curve and average area under the PR curve. The first approach uses averages of the precision and recall over all classes, thus obtaining a single curve ($AU(P\bar{R}C)$). The second approach constructs PR curve for each class, and returns the average area under the PR curves for all classes ($AU\bar{P}RC$). The two curves are able to catch different aspects of the performance of the distance measures. The first curve measure uses the information about the frequencies of the classes and the more frequent classes have bigger influence to the final score. On the other hand, the second measure is averaging the performance of each of the classes, i.e. each class has equal contribution to the final score.

4.2 Experimental methodology

The evaluation of the predictive performance was done using separate testing sets. The threshold value ranged from 0.0 to 1.0 step 0.05. The weight of the depth (w0) was set to 0.75, same as in [2]. Vens et al. in [2] conclude that the weighting parameter has no

strong effect on the performance (as compared to non-weighted it gives slightly better results when using Euclidean distance).

To prevent over-fitting, we used two pre-pruning methods: minimal number of examples in a leaf and F-test pruning.The minimal number of examples in a leaf is used as a stopping criterion in the PCT induction algorithm. In our experiments we set this value to 5 examples. The F-test pruning uses the F-test for statistical significance. The F-test is used by the algorithm to check whether the variance reduction is statistically significant at a given significance level. The algorithm takes as input a vector of significance levels and, by internal 10-fold cross-validation it selects one. In our experiments the used vector of significance levels was [0.001, 0.005, 0.01, 0.05, 0.1, 0.125].

5 Results

The performance results of the four different distance measures on the sixteen datasets are summarized in Table 2 and Table 3. As stated in the experimental design section, to evaluate the predictive performance we use the following two measures: the area under the average precision-recall curve and the average area under the PR curves. To check whether the difference in the performance using each of the four distances is statistically significant we used the corrected Friedman test (as recommended in [16]). The corrected Friedman test didn't detect any statistically significant differences in the performance in both cases (p ¡ 0.073 for the are under the average PR curve, and p ¡ 0.176 for the average area under the PR curves).

Table 2. Predictive performance of the algorithms estimated by the area under the average PR curve.

			$AU(\overline{PRC})$			
	Dataset	Annotation	Weighted Euclidean	Jaccard	Image CLEF	SimGIC
Saccharomyces Cerevisiae	cellcycle	FunCat	0.172	0.172	0.173	0.174
	church	FunCat	0.166	0.173	0.174	0.167
	derisi	FunCat	0.175	0.172	0.176	0.174
	eisen	FunCat	0.205	0.198	0.199	0.205
	gasch1	FunCat	0.195	0.182	0.186	0.182
	gasch2	FunCat	0.205	0.203	0.185	0.201
	pheno	FunCat	0.158	0.151	0.147	0.164
	spo	FunCat	0.186	0.182	0.185	0.181
	struc	FunCat	0.181	0.171	0.170	0.176
	church	GO	0.348	0.346	0.348	0.349
	eisen	GO	0.380	0.386	0.386	0.389
	pheno	GO	0.338	0.337	0.334	0.330
Arabidopsis Thaliana	interpro	FunCat	0.381	0.382	0.386	0.387
	scop	FunCat	0.517	0.483	0.492	0.510
	struc	FunCat	0.275	0.277	0.270	0.289
	interpro	GO	0.570	0.563	0.558	0.570

The ranking of the distances by the area under the average PR curve is as follows: the SimGIC distance has the best average rank, followed by the weighted Euclidean distance and the ImageCLEF distance. The Jaccard distance has the worst average rank. The situation is a bit different when average area under the PR curves is used for comparison: the weighted Euclidean distance has the best rank, followed by the SimGIC distance. Next are the ImageCLEF and Jaccard distance with equal rank.

Table 3. Predictive performance of the algorithms estimated by the average area under the PR curves.

			$\overline{AUPRC}$			
	Dataset	Annotation	Weighted Euclidean	Jaccard	Image CLEF	SimGIC
Saccharomyces Cerevisiae	cellcycle	FunCat	0.032	0.027	0.028	0.032
	church	FunCat	0.029	0.028	0.027	0.026
	derisi	FunCat	0.027	0.026	0.028	0.026
	eisen	FunCat	0.047	0.042	0.038	0.040
	gasch1	FunCat	0.036	0.040	0.030	0.038
	gasch2	FunCat	0.034	0.028	0.030	0.029
	pheno	FunCat	0.030	0.030	0.032	0.031
	spo	FunCat	0.030	0.031	0.030	0.032
	struc	FunCat	0.030	0.030	0.027	0.028
	church	GO	0.014	0.015	0.015	0.014
	eisen	GO	0.030	0.022	0.023	0.026
	pheno	GO	0.019	0.017	0.016	0.018
Arabidopsis Thaliana	interpro	FunCat	0.096	0.096	0.099	0.092
	scop	FunCat	0.163	0.139	0.150	0.159
	struc	FunCat	0.045	0.052	0.043	0.052
	interpro	GO	0.139	0.133	0.128	0.142

In Figure 3, we present the average PR curves obtained using the four distances and the 'pheno' dataset with FunCat annotation (Saccharomyces Cerevisiae). We can see that the PR curve for SimGIC is always above the PR-curves for the other distances. It thus clearly performs better than the other distances on this dataset.

6 Conclusions

In this work, we have reviewed and evaluated several distance measures that can be applied in the hierarchical multi-label classification task. In particular, we compared the weighted Euclidean distance, Jaccard distance, SimGIC distance and ImageCLEF distance. The distances were appropriate for hierarchies in the form of a tree, as well as hierarchies in the form of a directed acyclic graph.

We used separate testing sets to evaluate the influence of each distance measure on the learning process. The predictive performance was estimated with the area under the

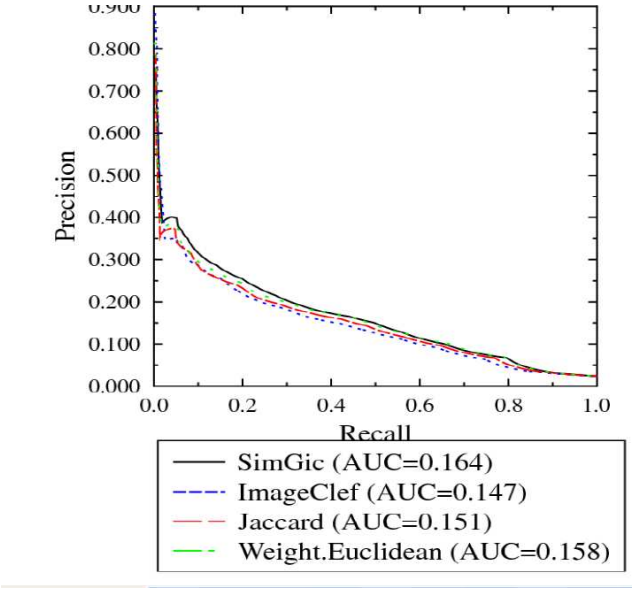

Fig. 3. An example PCT for HMLC, obtained with a given threshold, for the 'church' dataset with FunCat annotation.

average PR curve and the average area under the PR curves. The corrected Friedman test for statistical significance testing didn't detect difference in the performance. However, the SimGIC distance has the best average rank for the area under the average PR curve, while weighted Euclidean distance for the average area under the PR curves.

For future work, we plan to investigate the different weighting schemes. A distance can achieve better predictive performance if used with an appropriate weighting scheme. Also, we will conduct series of experiments on additional datasets from functional genomics and other domains, such as image annotation, text categorization etc.

Another line of further work is the use of ensembles form PCTs [17] to check whether the ensembles can increase the predictive performance and which distance is most suitable for ensemble learning.

Also we plan to investigate other evaluation measures of predictive performance adapted for HMLC [18], such as the hierarchical F-measure, hierarchical Precision, hierarchical Recall, average category similarity and other.

References

1. Barutcuoglu, Z., Schapire, R., Troyanskaya, O.: Hierarchical multi-label prediction of gene function. Bioinformatics **22** (2006) 830–836
2. Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning **73** (2008) 185–214

3. Rousu, J., Saunders, C., Szdemak, S., Shawe-Taylor, J.: Learning hierarchical multi-category text classification models. In: Proc. of the 22nd Int. Conf. on Machine Learning, Omnipress (2005) 745–752
4. Dimitrovski, I. Kocev, D., Loskovska, S., Džeroski, S.: Hierchical annotation of medical images. In: Proc. of the 11th International Multi-Conference Information Society IS 2008, 13. do 17. oktober 2008, Institut "Jozef Stefan", 2008, p. 174-181
5. Bakir, G., Hofmann, T., Scholkopf, B., Smola, A., Taskar, B., Vishwanathan, S. Predicting Structured Data. MIT Press, 2007
6. Blockeel, H., De Raedt, L., Ramon, J. Top-down induction of clustering trees. In: Proc. of the 15th ICML. (1998) 55-63
7. Zenko, B. Learning Predictive Clustering Rules, PhD Thesis, Faculty of Computer and Information Science, University of Ljubljana, 2007
8. Pesquita, C., Faria, D., Bastos, H., Falco, A.O., Couto, F.M. Evaluating GO-based semantic similarity measures In: Proceedings of the 10th Annual Bio-Ontologies Meeting (Bio-Ontologies 2007)
9. Image 2008 Medical Automatic Image Annotation Task http://www.imageclef.org/2008/medaat
10. A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. *Lecture Notes in Computer Science*, 2168:42–53, 2001.
11. Struyf, J., Džeroski, S. Constraint based induction of multi-objective regression trees, In: Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID'05, LNCS vol. 3933, pp. 222-233, 2006
12. Dzeroski, S., Gjorgjioski, V., Slavkov, I., Struyf, J. Analysis of Time Series Data with Predictive Clustering Trees, In: KDID06, LNCS vol. 4747, p. 63-80, 2007
13. Guo, X., Liu, R., Shriver, C.D., Hu, H. and Liebman, M.N. Assessing semantic similarity measures for the characterization of human regulatory pathways Bioinformatics, 22, 967-973
14. Mewes, H. W., Heumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S., Frishman, D. MIPS: a database for protein sequences and complete genomes. Nucl. Acids Research, 27, 1999, 44-48.
15. Ashburner, M. et al. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature Genetics, 25(1), 2000, 25-29.
16. Demsar J. Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 2006, 1-30
17. D.Kocev, C. Vens, J. Struyf, S. Džeroski. Ensembles of Multi-Objective Decision Trees, In: Proc. of the ECML 2007, LNAI vol. 4701, p. 624-631, 2007
18. Costa, E.P., Lorena, A.C., Carvalho, A.C.P.L.F., Freitas, A.A. A review of performance evaluation measures for hierarchical classifiers. In: Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop