Multilabel Classification in Parallel Tasks

Eneldo Loza Mencía ENELDO@KE.TU-DARMSTADT.DE
Knowledge Engineering Group, TU Darmstadt, Hochschulstrasse 10, 64289 Darmstadt, Germany

Abstract

In real world multilabel problems, it is often the case that e.g. documents are simultaneously classified with labels from multiple domains, such as *genres* in addition to *topics*. In practice, each of these problems is solved independently without taking advantage of possible label correlations between domains. Following the multi-task learning setting, in which multiple similar tasks are learned in parallel, we propose a global learning approach that jointly considers all domains. It is empirically demonstrated in this work that this approach is effective despite its simplicity when using a multilabel learner that takes label correlations into account.

1. Introduction

The starting point of this work is the following exemplary scenario: Books in a library are typically cataloged according to different types or domains of associated characteristics, e.g. genre, language, topic, epoch, author, etc. This type of annotation of objects is a very natural and common approach not only in the cataloging of texts (Sec. 5) but also e.g. when indexing music (Pachet & Roy, 2009). Each of these mappings could be seen and treated as independent from each other. In reality, however, there may be dependencies between the different associated values from different domains. An author may write only in a specific language and focus exclusively on *crime fiction*. At the same time, crime fiction novels may often have *murder* as one of their topics, etc. Thus, if we consider to learn a model that automatically catalogs books in a library database as a text classification problem, for instance, it may be advantageous to consider all the parallel subproblems as a single large joint problem instead of tackling each subproblem separately.

Appearing in *Working Notes of the 2nd International Workshop on Learning from Multi-Label Data*, Haifa, Israel, 2010.

In principle this is the same idea as in multi-task learning. In multi-task learning, we have a set of related learning problems (tasks), i.e. problems that have a common shared representation of their objects. It has been shown that learning these tasks simultaneously and jointly outperforms the common approach of learning them separately (single-task learning) (cf. Sec. 2). The library example can be seen as a special multi-task learning scenario in which each categorization domain represents a separate task, and all tasks share the same representation of their objects (books have the same representation, e.g. the same bag of words, in every task).

Simultaneously, the approach of considering the whole task rather than each sub-task separately is in principle also the basic idea behind many multilabel classification algorithms. Instead of considering each label as a separate problem, as in the popular binary relevance (one-against-all) approach, most of the recent approaches try to implicitly or explicitly take into consideration existing label correlations in order to improve the predictive quality (cf. Sec. 2).

The approach that we propose is to consider the set of parallel multilabel tasks in the library as a single joint task, as in multi-task learning, and solve it with a conventional multilabel classification algorithm. Most of the recent and more sophisticated multilabel approaches may benefit from the parallel processing as they also benefit from the commonality in a conventional multilabel setting. We propose in this first work on the subject to use pairwise decomposition (aka. one-vs-one), which implicitly considers label relationships by learning preferences between pairs of labels (Loza Mencía & Fürnkranz, 2008a; Fürnkranz et al., 2008). Furthermore, recent advances in handling many (Loza Mencía et al., 2010), even thousands of classes (Loza Mencía & Fürnkranz, 2008b; Loza Mencía & Fürnkranz, 2010) despite the quadratic number of models enable us to address the considerably increased complexity when the subtasks are joined. For one of the datasets we additionally employed HOMER as a meta-algorithm, which

is also capable of processing large numbers of labels (Tsoumakas et al., 2008) and has shown to harmonize well with the pairwise approach (Tsoumakas et al., 2009).

2. Related work

Approaches that try to explicitly exploit label correlations include the early work of McCallum (1999), in which generative models for labelsets are generated as a mixture of topic based word distributions, the conditional random fields parameterized by label co-occurrences in (Ghamrawi & McCallum, 2005) and the label correlations conditioned maximum entropy method of (Zhu et al., 2005), among others. A middle way is followed by Read et al. (2009) and their classifier chains by stacking the underlying binary relevance classifiers with the predictions of the previous ones, and by Cheng & Hüllermeier (2009), whose k-NN approach stacks the appearances of labels in the neighborhood as new features.

However, the majority of the approaches implicitly consider correlations by optimizing a loss on the predicted ranking of the labels. MMP perceptrons (Crammer & Singer, 2003), Rank-SVM (Elisseeff & Weston, 2001), Structural SVMs (Tsochantaridis et al., 2005) and the BP-ML neural network algorithm (Zhang & Zhou, 2006) e.g. rely on this. The latter approach is conceptually very similar to the multi-task neural networks of Caruana (1997), as both train a common network with several outputs denoting the labels , i.e. task outcomes. This is a popular approach in multi-task learning, also applied to bayesian networks (Bakker & Heskes, 2003). Other techniques try to develop special kernel functions which model inter-task relations (Evgeniou et al., 2006), or use statistical Dirichlet processes for the bayesian modeling (Xue et al., 2007).

A common problem to the referenced multilabel methods is their scalability in terms of number of labels, a factor which significantly increases when the subtasks are joined. Existing large scale approaches rely on the binary relevance decomposition (Pouliquen et al., 2003; Montejo Ráez et al., 2004; Tang et al., 2009) or using one-class classifiers (Villalba & Cunningham, 2009). However, solving each sub label relevance problem in a separate way would not change anything in comparison to solving it as multiple single-task problems in our proposed setting, neither computationally nor predictively. A more complete overview of existing multilabel algorithms can be found in (Tsoumakas et al., 2010).

3. Preliminaries

In multi-task learning, there exist several associated tasks of the form $(\bar{\mathrm{x}}, y)$, with $\bar{\mathrm{x}}$ denoting a representation of an instance or example and y representing a state in the output space $\mathcal{Y}$. We represent an instance or object as a vector $\bar{\mathrm{x}} = (x_1, \ldots, x_a)$ in a feature space $\mathcal{X} \subseteq \mathbb{R}^a$. The feature spaces $\mathcal{X}^{(t)}$ of the different tasks $t = 1 \ldots k$ are supposed to be similar and share common features in multi-task learning. This is a precondition for the learning transfer: there has to be a link between the instances in task s and t for any link between the two output spaces to be recognized. In this work we assume that all tasks share the same input space $\mathcal{X} = \mathcal{X}^{(1)} = \ldots = \mathcal{X}^{(k)}$ and that there is a common training set $\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_m$ for all tasks $t = 1 \ldots k$. This restriction corresponds to the common problem setting described in Sec. 1. Each training example $\bar{\mathrm{x}}$ is hence associated with k outputs $y^{(1)}, \ldots, y^{(k)}$, with $y^{(t)} \in \mathcal{Y}^{(t)}$, $t = 1 \ldots\ k$. We will denote this setting as *parallel tasks* in this work, however we will occasionally use *multi-task* as a synonym.

Since we are dealing with multilabel data, $y^{(t)}$ denotes the set of relevant labels for instance $\bar{\mathrm{x}}$ in task t, where $y^{(t)}$ is a subset of the $n^{(t)}$ possible classes $\mathcal{L}^{(t)} = \{\lambda_1^{(t)}, \ldots, \lambda_{n^{(t)}}^{(t)}\}$ and $\mathcal{Y}^{(t)} = 2^{\mathcal{L}^{(t)}}$. The learned multilabel classifier is therefore a function $f^{(t)} : \mathcal{X} \to \mathcal{Y}^{(t)}$ with $\hat{y}^{(t)} = f^{(t)}(\bar{\mathrm{x}})$ as the relevant labels predicted for test document $\bar{\mathrm{x}}$. Multilabel classifiers commonly also predict a ranking $r^{(t)} : \mathcal{L}^{(t)} \to \{1 \ldots n^{(t)}\}$ on the labels, with $r^{(t)}(\lambda^{(t)})$ returning the position of class $\lambda^{(t)}$ in the relevance ranking.

4. Parallel task learning

In order to benefit from the parallel alignment of the sub-task the idea presented in this work is to simply join the different multilabel problems and treat them as a single large multilabel task. That means, we transform the problems into one global problem with the training set $\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_m$ and the training signals $y_1^*, \ldots, y_m^*$, with $y_i^* \subseteq \mathcal{L}^* = \cup_{1 \le t \le k} \mathcal{L}^{(t)}$, $i = 1 \ldots m$ and $\mathcal{Y}^* = 2^{\mathcal{L}^*}$. We define the $\mathcal{L}^{(t)}$ as being disjoint, i.e. $\mathcal{L}^{(s)} \cap \mathcal{L}^{(t)} = \emptyset$, $1 \le t < s \le k$. After training the global multilabel learner, we obtain the global model $f^* : \mathcal{X} \to \mathcal{Y}^*$, $\hat{y}^* = f^*(\bar{\mathrm{x}})$, which is then transformed back to the local classifiers $f^{(t)}(\bar{\mathrm{x}}) = \hat{y}^{(t)} = \hat{y}^* \cap \mathcal{L}^{(t)}$. The ranking function $r^{(t)}(\lambda^{(t)}) = |\{\lambda_u^{(t)} \in \mathcal{L}^{(t)} \mid r^*(\lambda_u^{(t)}) \le r^*(\lambda^{(t)})\}|$ is determined similarly.

As a convention, in the context of multilabel settings, we do not make any distinction in the notation of whether we are dealing with the global task or the

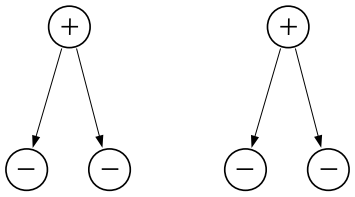

Figure 1. Pairwise training on two separate tasks (left and right): circles with a plus symbol represent the positive labels y an example training instance $\bar{\mathrm{x}}$ belongs to, circles with minus indicate the irrelevant labels $\overline{y}$ for $\bar{\mathrm{x}}$. The arrows represent the learned preferences, i.e. for which base learners λ_u vs. λ_v instance $\bar{\mathrm{x}}$ is a training example.

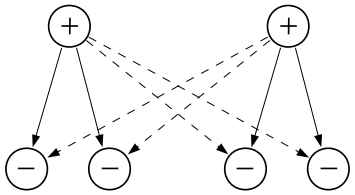

Figure 2. Pairwise training on the global problem by joining the two tasks from Fig. 1: the dotted arrows denote the additional learned relations for the current example $\bar{\mathrm{x}}$.

subtasks and we will therefore omit the superscript.

4.1. Pairwise classification for parallel tasks

In the pairwise decomposition method, one classifier is trained for each pair of classes, i.e., a problem with n different classes is decomposed into $\frac{n(n-1)}{2}$ smaller subproblems. For each pair of classes λ_u vs. λ_v, only examples belonging to either λ_u or λ_v are used to train the corresponding classifier $o_{u,v}$. In the multilabel case, an example is added to the training set for classifier $o_{u,v}$ if λ_u is a relevant class and λ_v is an irrelevant class or vice versa, i.e., $(\lambda_u, \lambda_v) \in y \times \overline{y}$ or, vice versa, $(\lambda_u, \lambda_v) \in \overline{y} \times y$, with $\overline{y} = \mathcal{L} \backslash y$ as negative labelset. The trained classifiers are shown as arrows in the simple example in Fig. 1. During classification, each base classifier is queried and the prediction is interpreted as a vote for one of its two classes. Labels are then ranked according to the number of received votes.

The pairwise binarization method is often regarded as superior to binary relevance (BR) since it benefits from simpler decision boundaries in the subproblems (Fürnkranz, 2002; Hsu & Lin, 2002; Loza Mencía & Fürnkranz, 2008a). Furthermore, we expect to further benefit from the following characteristics:

Firstly, pairwise classification implicitly exploits label correlations since the models are specifically trained to detect exclusion of labels. Remember that a base classifier $o_{u,v}$ is trained exactly with all the examples for which λ_u and λ_v are mutually exclusive. A positive prediction of $o_{u,v}$ could hence be interpreted as the implication $\lambda_u \in y \rightarrow \neg(\lambda_v \in y)$ holding on the current test instance. Since currently the base learners are not supposed to model something different than exclusion implications, we have to be cautious with this interpretation and therefore the estimations are currently just counted as simple votes and aggregated into a ranking. We plan to extend our approach in order to support extended expressiveness. In addition, incorporating (a priori) label constraints by incorporating them into the training process and by correcting predictions is being studied in ongoing work (Park & Fürnkranz, 2008).

Secondly, it was observed by Fürnkranz et al. (2008) that the additionally introduced virtual label (cf. below) and hence the additional learned preferences could slightly improve the predictions. We expect to benefit from this effect for the many additional connections when joining to a global model. Fig. 2 shows this on an example of two small parallel tasks. Assuming k parallel tasks of equal size n, the number of base classifiers increases by a factor of $\frac{kn(kn-1)\cdot 2}{k\cdot n(n-1)\cdot 2} = O(k)$.

To convert the resulting ranking of labels into a multilabel prediction, we use the *calibrated label ranking* (CLR) approach (Fürnkranz et al., 2008). The key idea is to introduce an artificial *calibration label* λ_0, which represents the split-point between relevant and irrelevant labels. Fig. 3 shows an example. As it turns out, the resulting n additional binary classifiers $\{\, o_{u,0} \,|\, u = 1 \ldots n\}$ are identical to the classifiers that are trained by the binary relevance approach. This holds also for the parallel task setting, as can be seen in Fig. 4 and 5, making the approach also easily applicable to this setting. [1]

QWeighted CLR reduces the classification costs from quadratic to roughly log-linear time in the number of labels (Loza Mencía et al., 2010) in exchange for predicting only a labelset. The training time in CLR is generally increased compared to BR by the factor of average labels per example. But since we are interested in the ranking performance in our evaluation, we rely on the full CLR.

For the text data, we use CLR together with the simple but fast perceptrons as base learners, leading to the (incrementally trainable) *Multilabel Pairwise Percep-*

[1] This approach may disadvantage smaller sub-tasks, however we evaluate mainly independently from the right tresholding and thus leave the analysis for future work.

trons (MLPP) algorithm and its dual variant *DMLPP* for large numbers of labels, as this combination has shown to be efficient as well as effective (Loza Mencía & Fürnkranz, 2008a;b; Loza Mencía & Fürnkranz, 2010).

4.2. HOMER with QCLR

Since the pairwise subproblems in one of the datasets in Sec. 5 are not linearly separable, we are not able to use the efficient pairwise perceptron approaches from Sec. 4.1.[2] And due to the high number of labels and the higher complexity of the common non-linear classifiers (both time and memory), plain CLR with a different base learner is also not viable so far.

We elude this problem by using *HOMER* as the metalearner for the CLR approach, which was specifically developed in order to handle multilabel problems with a large number of labels (Tsoumakas et al., 2008). The approach allows using base multilabel learners yet being sensible to the number of labels by decomposing the original problem into a tree of multilabel subproblems: a predetermined number of labels are joined to one metalabel, which is in turn one possible label in the parent multilabel subproblem. During prediction, the multilabel classifier at each inner node starting from the root is queried and the children nodes are visited for which the metalabel was predicted. The leaves represent the labels from the original problem.

For the decision, which labels to join to one metalabel at the inner node, a balanced k-means algorithm is employed that works on the similarity between the label vectors, i.e. the real labelsets of the examples. Hence, the approach prefers aggregating labels that are correlated.

HOMER is able to reduce training time in comparison with the binary relevance approach, since less training examples are used (in the inner nodes). The prediction costs are in general comparable or better (BR always has to evaluate all classifiers). Recently, HOMER was combined with CLR as its base multilabel learner, substantially reducing the training and test time and memory consumption in comparison to the plain CLR (Tsoumakas et al., 2009). It also helped to balance recall and precision since CLR becomes conservative in predicting with increasing number of labels. This combination also outperformed (in terms of predictive quality) or was comparable (computational costs) to the plain BR approach.

At first sight breaking up the problem into smaller ones may sound contradictory to our proposed approach of considering several sub tasks as a unique multilabel task. The reason for following the subdivision approach nevertheless is that we expect that we are still able to benefit from the additional possible inter-task label correlations, since the label clustering method in HOMER creates sub problems preserving as much label correlation information as possible.

[2] Unfortunately, the current implementation of DMLPP does not support kernels. We hope to add this soon.

Table 1. Statistics for *EUR-lex* and *rcv1*. Label *density* indicates the average number of labels per instance d relative to the total number of classes n, m denotes the number of documents, a the number of used features.

dataset	n	m	a	d	density
EUR-lex	4567	19348	5000	8.82	0.19 %
sm	201	”	”	2.21	1.11 %
dc	410	”	”	1.29	0.31 %
ev	3956	”	”	5.31	0.13 %
rcv1	103	804414	25000	3.24	3.15 %
ccat	34	”	”	1.44	4.24 %
ecat	26	”	”	0.41	1.58 %
gcat	33	”	”	0.70	2.12 %
mcat	10	”	”	0.69	6.90 %

5. Datasets

The *EUR-Lex* is a recent dataset containing 19,348 legislative documents from the European Union and is publicly available under `http://www.ke.tu-darmstadt.de/resources/eurlex/`. The documents are classified according to three different classification schemes: *subject matter* with 201 classes, *directory code* with 410 classes and *EUROVOC* with 3956 classes. For the processing of the text, we applied stop word removal and word stemming. The documents were then randomly split into 10 folds in order to perform cross validation. The 5,000 most frequent features on each of the training sets were selected and weighted with TF-IDF weights. The dataset was processed with DMLPP trained over two epochs.

The *HiFind* dataset contains 32,769 music titles annotated with 632 different labels (Pachet & Roy, 2009). The labels specify (mainly acoustic) characteristics of the categorized songs which can be divided into 17 distinct domains. Some of the sub tasks were intended to be single-label (binary or multiclass), however for all of them the number of distinct labelsets is greater than the number of classes. Following Tsoumakas et al. (2009), we trained HOMER with a cluster size of 7 (if possible) in combination with CLR and the J48 implementation of C4.5 (Witten & Frank, 2005) as base

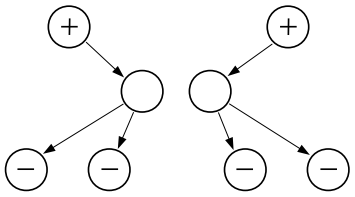

Figure 3. The two additional virtual labels $\lambda_0^{(s)}$ and $\lambda_0^{(t)}$ and additional preferences trained for the separate calibrations of the two tasks from Fig. 1. During prediction, the ranking of the obtained votes per label is split at the position of the virtual label.

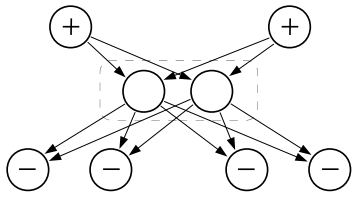

Figure 4. Calibration in the global task: both virtual labels $\lambda_0^{(s)}$ and $\lambda_0^{(t)}$ are always located at the same level between positive and negative labels y^* and $\overline{y}^*$, therefore the base classifiers to and from the two virtual classes are trained with the same examples.

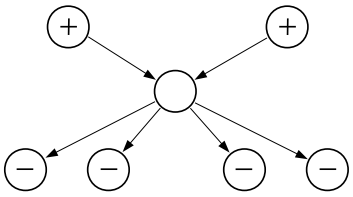

Figure 5. Calibration in the global task: the virtual labels $\lambda_0^{(s)}$ and $\lambda_0^{(t)}$ were merged to one unique calibration label λ_0^*. In all three cases in Fig. 3, 4 and 5 the same classifiers discriminating between the virtual labels and the remaining labels are learned.

classifiers (Sec. 4.2) on the first 16,452 examples and tested on the remaining 16,519.

As can be seen from the descriptions and Tables 1 and 2, the previous two real-world datasets fit perfectly to the illustrated library example in Sec. 1 and are hence prototypical for our parallel tasks setting. In addition, we simulated a multi-task setting on the Reuters datasets, for which we also recognized parallel dataset characteristics, although the corpus is normally only seen from the multilabel point of view.

The *Reuters Corpus Volume I (rcv1)* is one of the most widely used test collection for text categorization research. It contains 804,414 newswire documents, which we split into 535,987 training documents (all documents before and including April 26th, 1999) and 268,427 test documents (all documents after April 26th, 1999). A similar preprocessing as for the *EUR-Lex* data was used on the token files from Lewis et al. (2004). MLPP was applied on this dataset (one epoch). The 103 categories of the dataset are organized in a hierarchy with four main sub nodes: government/social (*gcat*), markets (*mcat*), economics (*ecat*) and corporate/industrial (*ccat*). We chose these four subsets as the domains of the tasks in the multi-task setting, although the classes therein are actually from the same type (topic categories) and it is therefore justified to treat them jointly from the beginning. However, a common binary benchmark dataset is based on this subdivision[3]. For the future we plan to use the additional associations contained in the corpus to 365 industry categories and 366 region categories, which have hardly received any attention yet in the literature.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#rcv1.binary`

6. Evaluation

We evaluate the effectiveness with label-based micro-averaged recall and precision and a multilabel version of accuracy. We follow the notation in Sec. 3 and define $\hat{\overline{y}}_i = \mathcal{L} \setminus \hat{y}_i$ as the set of predicted negative labels for a test instance $\bar{\mathrm{x}}_i$.

- *Precision* (PREC) computes the percentage of predicted labels that are relevant, *recall* (REC) computes the percentage of relevant labels that are predicted. F1 is the harmonic mean of both.

$$\text{PREC} = \frac{\sum_i |\hat{y}_i \cap y_i|}{\sum_i |\hat{y}_i|} \quad \text{REC} = \frac{\sum_i |\hat{y}_i \cap y_i|}{\sum_i |y_i|} \tag{1}$$

- The subset accuracy (ACC) denotes the percentage of perfectly predicted labelsets.

$$\text{ACC} = \frac{\sum_i I\left[\hat{y}_i = y_i\right]}{\sum_i 1}, \; I[x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Recall and precision allow a commensurate evaluation of an algorithm, in contrast to Hamming loss, which is usually used but unfortunately generally favors algorithms with high precision and low recall.

If a classifier was able to return rankings r on the labels, we computed the following loss measures as well.

- The ranking loss (RANK) returns the number of pairs of labels which are not correctly ordered, normalized by the total number of possible pairs.

$$\text{RANK} = \frac{|\{(\lambda, \lambda') \mid r(\lambda) > r(\lambda')\}|}{|y||\overline{y}|} \tag{3}$$

Table 2. Statistics for *HiFind*. The same notation is used as in Tab. 1.

dataset	n	m	a	d	density
HiFind	623	32971	98	37.3	5.98 %
character	37	”	”	3.97	10.7 %
country	27	”	”	0.98	3.64 %
dynamics	4	”	”	0.99	24.8 %
epoch	16	”	”	1.03	6.42 %
genre	31	”	”	2.65	8.53 %
instruments	100	”	”	5.09	5.09 %
language	16	”	”	1.01	6.30 %
metric	10	”	”	1.00	9.96 %
mood	59	”	”	5.27	8.94 %
period	2	”	”	0.004	0.24 %
popularity	3	”	”	0.97	32.5 %
rhythmics	10	”	”	1.17	11.7 %
setup	25	”	”	2.25	8.98 %
situation	74	”	”	5.26	7.11 %
style	158	”	”	1.21	0.77 %
tempo	8	”	”	0.99	12.4 %
variant	43	”	”	3.46	8.04 %

- Average Precision (AvgP) computes for each relevant label the percentage of relevant labels among all labels that are ranked before it, and averages these percentages over all relevant labels.

$$\text{AvgP} = \frac{1}{y}\sum_{\lambda\in y}\frac{|\{\lambda' \in y \mid r(\lambda') \leq r(\lambda)\}|}{r(\lambda)} \qquad (4)$$

  These two ranking measures are computed for each example and then averaged over all examples.

- For the idealistic F1($|y|$), we compute F1 as if exactly the right number of $|y|$ labels was returned (cf. Fürnkranz et al., 2008). Since the denominators in Eq. 1 coincide, we can interpret this measure as the example-based break-even point of precision and recall.

$$\text{F1}(|y|) = \frac{\sum_i |\{\lambda \in y_i \mid r_i(\lambda) \leq |y_i|\}|}{\sum_i |y_i|} \qquad (5)$$

6.1. Parallel tasks results

Tab. 3 shows the results on the *EUR-Lex* tasks. The first appreciable observation is that our parallel task (PT) approach considerably decreases recall and gains precision. This is due to the effect that calibration leads to cautious predictions when the number of labels is high. An intuitive explanation is that the higher the number of labels, the higher the number of votes to reach for the virtual label, the higher the probability that one of the base classifiers misses. We are currently investigating more robust alternative thresholding techniques specifically adapted to voting, that will hopefully be closer to the pseudo-F1($|y|$) measure. For this and for the remaining ranking based losses, the PT approach sometimes only slightly but always significantly outperforms the conventional method. The subset accuracy is again influenced by the conservative estimation of the calibration. Note also that we have to be very cautious when comparing the task-averaged measures in the last blocks since the tasks are indeed parallel, but the measures are nevertheless computed on different label domains. For this reason the non-parametric Wilcoxon signed-rank test is used were applicable.

Table 3. Results of DMLPP on the *EUR-Lex* dataset. First row: results on the global dataset. Next blocks: results trained on sub-tasks in 1^{st} row, trained on all parallel tasks (PT) in 2^{nd}. Last block: mean on all sub-tasks. Last row: #wins of PT model over local model. Bold entries show the winner, italics a significant difference on the cross validation (Wilcoxon signed-rank test, $p = 5\%$).

	Rec	Prec	Acc	Rank	AvgP	F1($\|y\|$)
EURlex	36.64	76.48	0.284	1.683	63.20	58.34
sm	***64.50***	75.48	33.29	0.874	83.26	75.25
PT	57.34	***85.36***	***36.42***	***0.851***	***84.45***	***76.96***
dc	***54.23***	77.11	***45.95***	0.844	81.05	71.42
PT	48.09	***83.94***	44.98	**0.840**	***82.20***	***73.25***
ev	**25.48**	66.63	**0.636**	2.325	53.35	48.59
PT	25.22	**67.10**	0.610	***2.307***	***53.47***	***48.71***
mean	**48.07**	73.07	26.63	1.348	72.55	65.09
PT	43.55	**78.80**	**27.34**	**1.333**	**73.37**	**66.31**
wins	0	**3**	1	**3**	**3**	**3**

Tab. 4 shows the averages on the 16 tasks of the *HiFind* dataset (task *period* was omitted since no classifier could be locally learned). We can observe the opposite behavior with respect to recall and precision using HOMER. These differences between recall and precision are more pronounced on the smaller tasks, which indicates that this might be related to the smaller proportion of number of labels to cluster size, since the smaller this proportion the greater precision and the smaller the recall in (Tsoumakas et al., 2009). Unfortunately, it is not possible to retrieve ranking losses for HOMER. Nevertheless, the gain in Rec for the globally trained model outweights the loss in Prec in terms of the less specific F1 and Acc. And more interestingly, this shows that it might be beneficial to join the parallel tasks although the base learner again breaks down the global task into smaller independent problems. For HOMER, this is probably due to the effective clustering of the generated subproblems so

Table 4. Results of HOMER on the *HiFind* dataset, only the means and number of wins are shown. Italic values indicate a statistically significant difference between the means on the 16 sub-tasks. Measures based on rankings are omitted since the used base classifier only predicts labelsets.

	Rec	Prec	Acc	F1
HiFind	56.51	51.95	0.012	54.13
mean	50.96	***53.72***	23.11	51.77
PT	***56.33***	51.82	**24.32**	***53.74***
wins	**16**	4	7	**13**

Table 5. Results on the *rcv1* dataset of MLPP. Each block shows the direct comparison between locally and globally learned model, as in Tab. 3.

	Rec	Prec	Acc	Rank	AvgP	F1($\lvert y\rvert$)
rcv1	80.32	83.89	49.78	0.526	93.35	87.22
ccat	**81.17**	76.27	66.16	0.549	97.37	88.71
PT	79.81	**80.85**	**69.54**	**0.528**	**97.42**	**89.00**
ecat	**70.09**	71.17	87.12	0.117	99.49	91.95
PT	68.90	**79.02**	**89.37**	**0.109**	**99.49**	**92.43**
gcat	**79.18**	81.67	83.57	0.158	99.10	91.36
PT	78.74	**84.79**	**85.07**	**0.149**	**99.13**	**91.78**
mcat	**89.12**	87.61	89.72	0.125	99.79	98.20
PT	88.39	**90.96**	**90.94**	**0.114**	**99.80**	**98.36**
mean	**79.89**	79.18	81.64	0.237	98.94	92.55
PT	78.96	**83.91**	**83.73**	**0.225**	**98.96**	**92.89**
wins	0	**4**	**4**	**4**	**4**	**4**

the information contained in the label correlations are preserved as much as possible. Reversely, this demonstrates the effectiveness of our parallel task setting since it shows the degree of additional information contained in the inter-domain correlations and that it can be effectively exploited. However, it would be interesting in this context to analyze the performance if the reverse way is followed, i.e. training on the local task and then aggregating it to a prediction for the global task. We leave this for future work.

The same conclusion is drawn from the results on the Reuters dataset in Tab. 5. Again, as on *EUR-Lex*, we can see the preference for high precision and lower recall of the global approach. However, the improvement on the remaining measures is clearer, even on the subset accuracy, though the differences in the nearly perfect AvgP results are almost not perceptible.

7. Conclusions

The starting point of this work was the recognition of a common characteristic of many real world problem, namely the mapping of the same object to concepts from several different domains. We referred to such problems as *parallel tasks* and evaluated the straightforward approach of joining the subtasks to one large global multilabel problem. As in the more general multi-task learning setting, we expected to benefit from the additional information obtained through the aggregation of the labelsets. We showed that multilabel algorithms which consider label correlations are able to effectively exploit the label correlations. In particular, the highly scalable and efficient pairwise perceptrons algorithms improved the quality of the predicted rankings. Perhaps more surprising and pleasing was the insight that HOMER allows also less scalable base learners to take advantage of the parallel task setting, though the used mechanism is to divide the original problem into smaller subproblems, which is in a certain sense directly opposed but actually compatible to the proposed approach.

This first evaluation of parallel tasks in multilabel leaves several possibilities for future work. The more explicit exploitation of label correlations in pairwise decomposition is an ongoing issue (cf. Sec. 4.1). Furthermore, different label correlation respecting algorithms could be compared. Actually, this setting could effectively be used in practice in order to analyze to which degree a multilabel algorithm takes label correlations into account. This particular property of multilabel algorithm makes it interesting to try to apply them on the more general multi-task learning setting, in which the objects in the tasks are not longer parallel but only similar. Of course, the opposite approach of incorporating ideas and mechanisms from multi-task learning is also very interesting.

References

Bakker, Bart and Heskes, Tom. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:2003, 2003.

Caruana, R. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997.

Cheng, Weiwei and Hüllermeier, Eyke. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

Crammer, Koby and Singer, Yoram. A Family of Additive Online Algorithms for Category Ranking. *Journal of Machine Learning Research*, 3(6):1025–1058, 2003.

Elisseeff, Andre and Weston, Jason. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, volume 14, pp. 681–687. MIT Press, 2001.

Evgeniou, T., Micchelli, C.A., and Pontil, M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615, 2006.

Fürnkranz, Johannes. Round Robin Classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

Fürnkranz, Johannes, Hüllermeier, Eyke, Loza Mencía, Eneldo, and Brinker, Klaus. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

Ghamrawi, Nadia and McCallum, Andrew Kachites. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 195–200, New York, NY, USA, 2005. ACM.

Hsu, Chih-Wei and Lin, Chih-Jen. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

Lewis, David D., Yang, Yiming, Rose, Tony G., and Li, Fan. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Loza Mencía, Eneldo and Fürnkranz, Johannes. Pairwise learning of multilabel classifications with perceptrons. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IJCNN 08)*, pp. 2900–2907, Hong Kong, 2008a.

Loza Mencía, Eneldo and Fürnkranz, Johannes. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Disocvery in Databases (ECML-PKDD-2008), Part II*, pp. 50–65, Antwerp, Belgium, 2008b. Springer-Verlag.

Loza Mencía, Eneldo and Fürnkranz, Johannes. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts*, volume 6036 of *Lecture Notes in Artificial Intelligence*, pp. 192–215. 1st edition, May 2010.

Loza Mencía, Eneldo, Park, Sang-Hyeun, and Fürnkranz, Johannes. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing*, 73(7-9):1164 – 1176, March 2010.

McCallum, Andrew Kachites. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*, 1999.

Montejo Ráez, Arturo, Ureña López, Luis Alfonso, and Steinberger, Ralf. Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In *Advances in Natural Language Processing, 4th International Conference (EsTAL 2004)*, volume 3230 of *Lecture Notes in Computer Science*, pp. 1–12. Springer, 2004.

Pachet, F. and Roy, P. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):335–343, 2009.

Park, Sang-Hyeun and Fürnkranz, Johannes. Multi-label classification with label constraints. In *Proceedings of the ECML/PKDD-08 Workshop on Preference Learning (PL-08)*, pp. 157–171, Antwerp, Belgium, 2008.

Pouliquen, Bruno, Steinberger, Ralf, and Ignat, Camelia. Automatic annotation of multilingual text collections with a conceptual thesaurus. In *Proceedings of the Workshop 'Ontologies and Information Extraction' at the Summer School 'The Semantic Web and Language Technology - Its Potential and Practicalities' (EUROLAN'2003), 28 July - 8 August 2003*, Bucharest, Romania, 2003.

Read, Jesse, Pfahringer, Bernhard, Holmes, Geoffrey, and Frank, Eibe. Classifier chains for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II*, volume 5782 of *Lecture Notes in Computer Science*, pp. 254–269. Springer, 2009.

Tang, Lei, Rajan, Suju, and Narayanan, Vijay K. Large scale multi-label classification via metalabeler. In *18th International World Wide Web Conference*, pp. 211–211, April 2009.

Tsochantaridis, Ioannis, Joachims, Thorsten, Hofmann, Thomas, and Altun, Yasemin. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

Tsoumakas, Grigorios, Katakis, Ioannis, and Vlahavas, Ioannis. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of the ECML/PKDD-08 Workshop on Mining Multidimensional Data (MMD-08)*, Antwerp, Belgium, 2008.

Tsoumakas, Grigorios, Loza Mencía, Eneldo, Katakis, Ioannis, Park, Sang-Hyeun, and Fürnkranz, Johannes. On the combination of two decompositive multilabel classification methods. In *Proceedings of the ECML/PKDD-09 Workshop on Preference Learning (PL-09)*, pp. 114–129, Bled, Slovenia, 2009.

Tsoumakas, Grigorios, Katakis, I., and Vlahavas, Ioannis P. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. 2 edition, 2010.

Villalba, Santiago D. and Cunningham, P. Using unsupervised classifiers for multilabel classification in open-class-set scenarios. In *Proceedings of the ECML/PKDD-09 Workshop on Learning from Multi-Label Data (MLD-09)*, pp. 146–160, Bled, Slovenia, 2009.

Witten, Ian H. and Frank, Eibe. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.

Xue, Ya, Liao, Xuejun, Carin, Lawrence, and Krishnapuram, Balaji. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:2007, 2007.

Zhang, Min-Ling and Zhou, Zhi-Hua. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18:1338–1351, 2006.

Zhu, Shenghuo, Ji, Xiang, Xu, Wei, and Gong, Yihong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 274–281, 2005.