Graded Multilabel Classification by Pairwise Comparisons

Technical Report TUD-KE-2014-01

Christian Brinker, Eneldo Loza Mencía, Johannes Fürnkranz

Knowledge Engineering Group,
Technische Universität Darmstadt
brinker@stud.tu-darmstadt.de, {eneldo,juffi}@ke.tu-darmstadt.de

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Knowledge
Engineering

Abstract

The task in multilabel classification is to predict for a given set of labels whether each individual label should be attached to an instance or not. Graded multilabel classification generalizes this setting by allowing to specify for each label a degree of membership on an ordinal scale. This setting can be frequently found in practice, for example when movies or books are assessed on a one-to-five star rating in multiple categories. In this paper, we propose to reformulate the problem in terms of preferences between the labels and their scales, which then be tackled by learning from pairwise comparisons. We present three different approaches which make use of this decomposition and show on three datasets that we are able to outperform baseline approaches. In particular, we show that our solution, which is able to model pairwise preferences across multiple scales, outperforms a straight-forward approach which considers the problem as a set of independent ordinal regression tasks.

Contents

1 Introduction

Multilabel Classification (MLC), the task of learning to assign multiple labels to a single data item, has received a lot of attention in the recent machine learning literature (Tsoumakas et al., 2010) because it has many real-world applications such as tagging of messages in blogs, annotating images, or assigning keywords to scientific papers. However, often it does not suffice to only predict whether a label is present or not, but instead we need to predict a degree or grade of membership to a particular category or label. Cheng, Dembczyński, and Hüllermeier (2010) introduced this task as *Graded Multilabel Classification* (GMLC). For example, TV guides often rate a movie on a scale from one to five stars in several different categories such as 'fun', 'action', 'sex', or 'suspense', as is shown in Table 1.1. The additional information in the form of grades of memberships in contrast to simple binary assignments of genres can be very useful and appreciable information for a user choosing her individual TV program. Another application is the prediction of answers from questionnaires, where a common setting is to ask the probands to answer a series of questions and to respond on a graded scale of agreement, frequency, importance, quality or likelihood.

Although superficially similar, this task differs from a classical recommendation task (Jannach et al., 2010). While in both cases one essentially needs to make ordinal predictions that correspond to ratings, in recommender systems the training information is a sparsely populated rating matrix and the task is to predict (some of) the missing values. In contrast, the training information for GLMC is a complete matrix where each of the objects in the lines is characterized with a set of features (e.g., features that characterize the respective movie), and the task is to predict the entries for a new line, given the features that correspond to this new entry.

Similar to the *binary relevance* (BR) approach to MLC, a straight-forward solution to GMLC is to transform the task into n separate ordinal classification problems, one for each category. In the example of Table 1.1, we would get four tasks, one for predicting the 'fun' rating, one for 'action', and two more for 'sex' and 'suspense'. However, when separating these tasks into several independent ordinal classification tasks, the inter-dependencies and correlations of the labels cannot be utilized, in much the same way as label dependencies cannot be tackled with the BR approach to MLC (Dembczynski et al., 2012). For this reason, Cheng, Dembczyński, and Hüllermeier (2010) proposed several techniques for tackling this problem without losing the dependency between the categories, and showed that this leads to better classification results.

In this technical report, we assume an inherent preference structure between the labels in combination with their grade of membership, and propose pairwise preference learning as a suitable technique to exploit this structure. To this end, we generalize calibrated label ranking, a technique for tackling multilabel classification in a pairwise fashion (Fürnkranz et al., 2008), to the case where we have mul-

Table 1.1: Example of ratings of some movies according to the German TV guide TVSpielfilm.de

Movie title	'fun'	'action'	'sex'	'suspense'
The other guys	⋆⋆⋆	⋆⋆		
A few good men		⋆		⋆⋆⋆
Once upon a time in the west		⋆⋆⋆	⋆	⋆⋆⋆
Dirty dancing		⋆	⋆	

tipartite instead of bipartite preference information. In particular, we show how the use of a calibration label, which indicates the separation between relevant and irrelevant labels in the predicted ranking, can be generalized to multiple such labels. As a result, we investigate and experimentally compare three different variations of this principled approach.

We start the technical report with a brief recapitulation of ordinal classification and multilabel classification (Section 2). In Section 3, we formally define graded multilabel classification, and recapitulate previous approaches. Section 4 introduces our reformulation of this approach in a preference-based setting, resulting in three different variants. We then experimentally compare our approaches with the approaches presented by Cheng et al. (2010) (Section 5), and draw some conclusions in Section 6.

2 Preliminaries

We represent an instance or object as a vector $\mathbf{x}$ in a feature space $\mathbb{X}$. Each instance can be associated with a point $y_\mathbf{x}$ in the target space $\mathbb{Y}$. A training set is a finite set of tuples $(\mathbf{x}, y_\mathbf{x}) \in \mathbb{X} \times \mathbb{Y}$ drawn independently from an unknown probability distribution on $\mathbb{X} \times \mathbb{Y}$. The goal is to learn a classifier $H : \mathbb{X} \to \mathbb{Y}$ which correctly predicts the true $y_\mathbf{x}$ for a given $\mathbf{x}$. We will denote the prediction of H with a circumflex, i.e. $\hat{y} = H(\mathbf{x})$. Depending on the form of $\mathbb{Y}$ we face different problems and assumptions and may consider different learning strategies. In the simplest case, binary classification, we have $\mathbb{Y} = \{0, 1\}$. The two problems described below, ordinal classification and multilabel classification, generalize this problem by extending the value space from binary to an ordinal scale and by adding several binary value spaces. The combination of both generalizations will be introduced in in Section 3.

2.1 Ordinal Classification

Ordinal classification, or ordinal regression, denotes the problem of learning a mapping from an instance space $\mathbb{X}$ to a discrete and ordered finite space $\mathbb{Y} = \mathbb{M} = \{\mu_1, \ldots, \mu_m\}$ with an inner structure $\mu_1 \prec \mu_2 \prec \ldots \prec \mu_m$, where $\prec$ denotes a relation inducing a total order. In contrast to a numeric regression problem, we do not assume a linear or additive scale. Consider e.g. the ordinal structure *small* $\prec$ *medium* $\prec$ *large*. A hidden characterization for the three grades in terms of absolute amount of meters or intervals may exist, but cannot be generally assumed. The scale is, e.g., valid and plausible for categories like *shoes* and *furniture*, but may greatly differ in absolute terms between both product classes. Moreover, the difference between two values cannot be determined by subtracting their levels, i.e., the difference between *small* and *medium* does not have to be equivalent to the difference between *medium* and *large*.

One straight-forward solution to ordinal classification is to ignore the structure on $\mathbb{M}$ and solve the problem as a standard multiclass problem, e.g. using one-against-all decomposition (see Sec. 3). A more simple yet effective decomposition strategy was proposed by Frank and Hall (2001): the original problem is decomposed into $n-1$ independent binary subproblems, each of which contains all instances with a class value $\prec \mu_i$ as positive examples and all others as negative examples. The probabilistic estimations of the base classifiers are then combined into a distribution $P(\mu_i) = P(\prec \mu_{i+i}) - P(\prec \mu_i)$ over the possible class grades.

2.2 Multilabel Classification

Multilabel classification (Tsoumakas et al., 2010; Zhang and Zhou, 2013) refers to the task of learning a function H that maps instances $\mathbf{x} \in \mathbb{X}$ to subsets $P_\mathbf{x} \subseteq \mathscr{L}$, where $\mathscr{L}$ is a finite set of predefined labels $\{\lambda_1, \ldots, \lambda_n\}$. An alternative representation is to consider the label space as $\mathbb{Y} = \{0, 1\}^n$ and to represent $P_\mathbf{x}$ as a vector $\mathbf{y}_\mathbf{x} = (y^1_\mathbf{x}, \ldots, y^n_\mathbf{x})$ where y_i is 1 if $\lambda_i \in P_\mathbf{x}$ and 0 otherwise. The labels in $P_\mathbf{x}$ are usually said to be *relevant*, *present* or *positive*, whereas $\mathscr{L}$. $N_\mathbf{x} = \mathbb{L} \backslash P_\mathbf{x}$ is called the set of *irrelevant*, *absent* or *negative* labels. Thus, in contrast to multi-class classification, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance.

The most straight-forward approach to solving multilabel problems is to decompose them into several independent binary subproblems, one for each label. Thus, each of the base classifiers $H_i : \mathbb{X} \to \{0, 1\}$ trained on training instances $(\mathbf{x}, y^i_\mathbf{x})$ tries to predict the relevance of one label λ_i, which is why it is frequently referred to as *binary relevance* decomposition (BR). The overall function H is obtained

by simple combination: $H(\mathbf{x}) = (H_1(\mathbf{x}), \ldots, H_n(\mathbf{x}))$. Its main drawback is, obviously, that dependencies between labels are completely ignored although it is generally considered that exploiting these dependencies is a crucial issue in multilabel classification. The pairwise decomposition approach, on which we will focus in this work, tries to alleviate this problem by modeling the pairwise relation between relevant and irrelevant labels as preferences. We will return to this in Section 4.

3 Graded Multilabel Classification

In graded multilabel classification (Cheng et al., 2010), each label λ in the set of relevant labels $P_{\mathbf{x}}$ of instance $\mathbf{x} \in \mathbb{X}$ is no longer only relevant or not ($\mathbb{M} = \{0, 1\}$), but has output values $\mathbb{M} = \{\mu_1, \ldots, \mu_m\}$ with an ordered scale $\mu_1 \prec \mu_2 \prec \ldots \prec \mu_m$ as in ordered classification. It is assumed that the same ordinal scale is used for all labels, i.e. $\mathbb{Y} = \{\mu_1, \ldots, \mu_m\}^n$. This is a strong restriction but is motivated on real applications such as those sketched in the introduction. On the other hand, this assumption induces a (limited) comparability between the grades of the different labels which cannot be assumed in the more general setting of multi-target ordinal regression. Moreover, we assume that the grade μ_1 describes the complete absence of a label and μ_m its full presence. Thus, in case of $m = 2$, this setting reduces to multilabel classification.

Following (Cheng et al., 2010), we define the auxiliary membership function $L_{\mathbf{x}} : \mathbb{L} \to \mathbb{M}$ as $L_{\mathbf{x}}(\lambda_i) = y^i_{\mathbf{x}}$ which returns the grade of a specific label and instance. Moreover, let $P'^i_{\mathbf{x}} = \{\lambda \mid \mu_i = L_{\mathbf{x}}(\lambda)\}$ be the set of labels that have exactly grade μ_i, and $P^i_{\mathbf{x}} = \{\lambda \mid \mu_i \preceq L_{\mathbf{x}}(\lambda)\}$ be the set of labels that are at least as relevant as grade μ_i. The latter set allows to model the assumption that if a label has a membership degree of μ_i, it also has all grades $\mu_j \prec \mu_i$ associated to it. Thus, since μ_1 is the lowest possible grade, it follows that $P^1_{\mathbf{x}} = \mathbb{L}$.

Cheng et al. (2010) introduce three straight-forward *reduction* schemes in order to decompose the original problem into a set of well-known and solvable subproblems. In the following, we briefly recapitulate these approaches. Figure 3.1 illustrates these reductions on an example where we have four possible labels $\mathbb{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$, each with a scale $\mu_1 \prec \mu_2 \prec \mu_3 \prec \mu_4$. Shown is a case of an example $\mathbf{x}$ for which the labels are increasingly relevant, i.e., where $\forall i : L_{\mathbf{x}}(\lambda_i) = \mu_i$, $i = 1 \ldots 4$.

3.1 Vertical Reduction

In the vertical reduction, the original problem of learning $H : \mathbb{X} \to \mathbb{M}^n$ is reduced to n ordinal classification problems of learning $[H]_{\lambda_1}, \ldots, [H]_{\lambda_n}$, $[H]_{\lambda_i} : \mathbb{X} \to \mathbb{M}$, one for each label $\lambda_1, \ldots, \lambda_n$ (cf. Figure 3.1 (left)). The aggregation of the individual predictions is trivially given by $H(\mathbf{x}) = ([H]_{\lambda_1}(\mathbf{x}), \ldots, [H]_{\lambda_n}(\mathbf{x}))$. Obviously, the individual ordered classifiers are not able to model interdependencies and correlations between the different labels, which is the main disadvantage of this approach.

3.2 Horizontal Reduction

In contrast, the horizontal reduction transforms the original problem into $m = |\mathbb{M}|$ multilabel classification problems. For each grade μ_i, $i = 1 \ldots m$ we learn a classifier $[H]^i : \mathbb{X} \to \mathscr{P}(\mathscr{L})$ using $(\mathbf{x}, P^i_{\mathbf{x}})$ as training information. Note that due to $P^1_{\mathbf{x}} = \mathbb{L}$ we can ignore grade μ_1.

Also note that the classifiers $[H]^i$ are, in principle, not independent from each other, since if some label is relevant to some grade μ_i, it is also relevant to all grades $\mu_j \prec \mu_i$. More specifically, it holds that $P^j_{\mathbf{x}} \subseteq P^i_{\mathbf{x}}$ for $\mu_j \prec \mu_i$. This obviously leads to an additional challenge during the aggregation of the individual predictions, since although this dependency will be reflected in the training data, it cannot be guaranteed that $[H]^j(\mathbf{x}) = \hat{P}^j_{\mathbf{x}} \subseteq [H]_i(\mathbf{x}) = \hat{P}^i_{\mathbf{x}}$, $\mu_j \prec \mu_i$ holds.

Cheng et al. attempt to address this problem by weighting the evidence for a higher grade higher than the evidence for a lower grade, and hence propose to resolve contradictions by taking for each

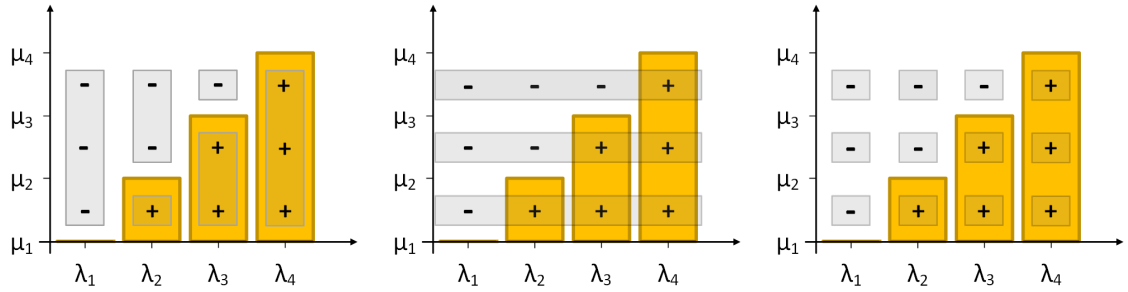


Figure 3.1: Different decompositions of graded multilabel classification: vertical (*left*), horizontal (*center*), and complete (*right*). The illustration shows the decompositions for a training instance for which label λ_1 has grade μ_1, λ_2 is with grade μ_2, λ_3 with μ_3 and finally λ_4 with μ_4.

label λ_i the maximum predicted grade $\max_{\prec}\{\mu_j \in \mathbb{M} \mid \lambda_i \in \hat{P}^j_{\mathbf{x}}\}$, where max is defined with respect to the total order relation $\prec$.

In a way, this problem is orthogonal to the above-mentioned problem of the independent predictions in the vertical reduction. On the other hand, unlike the vertical scheme, the horizontal reduction scheme conserves dependencies between labels because each multilabel subproblem allows to model the label dependencies at a certain degree of membership. This information can be taken into account by algorithms like IBLR-ML used in Cheng et al. (2010).

3.3 Complete Reduction

The complete reduction learns one single classifier $[H]_{\lambda_i\mu_j} : \mathbb{X} \to \{0,1\}$ for each of the $n \cdot m$ possible label–grade combinations using training information $(\mathbf{x}, \mathbb{I}(\mu_j \preceq y^i_{\mathbf{x}}))$ where $\mathbb{I}$ is the indicator function ($\mathbb{I}(x) = 1$ if x is true, and 0 otherwise).

This reduction can be seen as a combination of the previous two techniques: either we reduce the problem horizontally and then consider each label separately as a an ordinal problem with two ordinal classes, or we use vertical decomposition and solve each ordinal problem by learning to predict each grade. Thus, it corresponds to using binary relevance learning for solving the multilabel problems resulting from a horizontal decomposition or using the approach of Frank and Hall for the vertical reduction. However, the aggregation is different than in the latter approach, since we use the max aggregation already employed for the horizontal reduction. Note that in any case, dependencies between labels and/or grades cannot be exploited at all by the complete reduction.

3.4 Horizontal reduction with IBLR-ML

For completeness, we also briefly sketch the approach that was actually proposed by Cheng et al. (2010). Essentially, they propose to use a horizontal decomposition with a state-of-the-art multilabel classificaiton algorithm that combines instance-based learning with logistic regression (IBLR-ML), which allows them to exploit the interdependencies between the labels in the horizontal reduction setup (Cheng and Hüllermeier, 2009).

IBLR-ML counts the presence of each label in the vicinity of the k nearest neighbors of a test instance $\mathbf{x}$ (w.r.t. some distance metric) and uses these counts as input to a logistic regression learner which can then estimate the probabilities for each label. As the input to the learner is just the counts represented as a vector in $\mathbb{Z}$, we can view IBLR-ML as a stacking approach which uses the predictions of n k-NN classifiers as substitution of the original features and learns a logistic regression on top in order to

predict the correct labels. Other approaches that employ a similar stacking approach include Godbole and Sarawagi; Tsoumakas et al. (2009); Montañés et al. (2014). Best known are probably classifier chains, which only use a subset of predictions of the previous classifiers as additional features for subsequent classifiers (Read et al., 2011).

For the sake of comparability with the only existing results on graded multilabel classification and considering that the mentioned approaches are similar to each other, we only compare to IBLR-ML as representative approach which explicitly takes label dependencies into account and leave additional evaluations for further work.

4 Graded Multilabel Classification by Pairwise Comparison

Learning by pairwise decompositions is based on the idea of modeling preferences between labels (Hüllermeier et al., 2008). These preferences are either derived from the label structure (e.g. a hierarchy) or given for the training instances at hand, e.g. in the form of a total or partial, often multi-partite ranking. Moreover, pairwise decomposition implicitly takes label dependencies into account to some extent, since it explicitly models the cases of pairwise exclusions. We hence believe that pairwise decomposition is well suited to the setting of graded multilabel classification. In particular, we build upon *calibrated label ranking* (CLR), a pairwise approach to solving multilabel problems, which we describe in more detail in Section 4.1. Thereafter, we will introduce three different approaches for generalizing CLR to the graded case, which are all based on the idea of working with multiple calibration labels (Section 4.2).

4.1 Calibrated Label Ranking

The pairwise decomposition of multilabel problems interprets the training information given as bipartite rankings $N_{\mathbf{x}} \prec P_{\mathbf{x}}$, i.e., we can deduce explicit preference statements $\lambda_u \prec \lambda_v$ for all $\lambda_u \in N_{\mathbf{x}}, \lambda_v \in P_{\mathbf{x}}$. These preferences are learned by training classifiers $H_{uv} : \mathbf{x} \to \{0,1\}$ for each of the possible pairs of labels, $1 \leq u < v \leq n$. Hence, the problem is decomposed into $\frac{n(n-1)}{2}$ smaller binary sub-problems. For each pair of labels (λ_u, λ_v), only examples belonging to either λ_u or λ_v are used to train the corresponding classifier $H_{u,v}$. All other examples are ignored. More precisely, assuming $u < v$, an example is added to the training set for classifier $H_{u,v}$ if λ_u is a relevant label and λ_v is an irrelevant label or vice versa, i.e., if $(\lambda_u, \lambda_v) \in P_{\mathbf{x}} \times N_{\mathbf{x}}$ or $(\lambda_u, \lambda_v) \in N_{\mathbf{x}} \times P_{\mathbf{x}}$. Thus, training examples belonging to label λ_u will receive a training signal of 1, whereas training examples of label λ_v will be classified with 0.

During classification, the predictions of the $\frac{n(n-1)}{2}$ base classifiers $H_{u,v}$ are interpreted as *preference statements* that predict for a given example which of the two labels λ_u or λ_v is preferred. In order to convert these binary preferences into a label ranking, we use simple voting which interprets each binary preference as an (unweighted) full vote (0 or 1) for the preferred class. Labels are then ranked according to the number of received votes after the evaluation of all base classifiers.

To convert the resulting ranking of labels into a multilabel prediction, we use the *calibrated label ranking* (CLR) approach (Fürnkranz et al., 2008). This technique avoids the need for learning a threshold function for separating relevant from irrelevant labels, which is often performed as a post-processing phase after computing a ranking of all possible classes. The key idea is to introduce an artificial *calibration label* $v = \lambda_0$, which represents the split-point between relevant and irrelevant labels. Thus, it v assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over v (cf. Figure 4.1).

During prediction, the virtual label is naturally embedded in the label ranking and is treated like any other label. The position of the virtual label in the predicted ranking then denotes a natural cutting point for dividing the label ranking into two sets.[1]

The pairwise learning method is often regarded as superior to binary relevance (or one-against-all, respectively) because it profits from simpler decision boundaries in the sub-problems (Fürnkranz, 2002; Fürnkranz et al., 2008). The reason is that each of the pairwise classifiers contains fewer examples.

[1] We break ties in the final counting in favor of the virtual label.

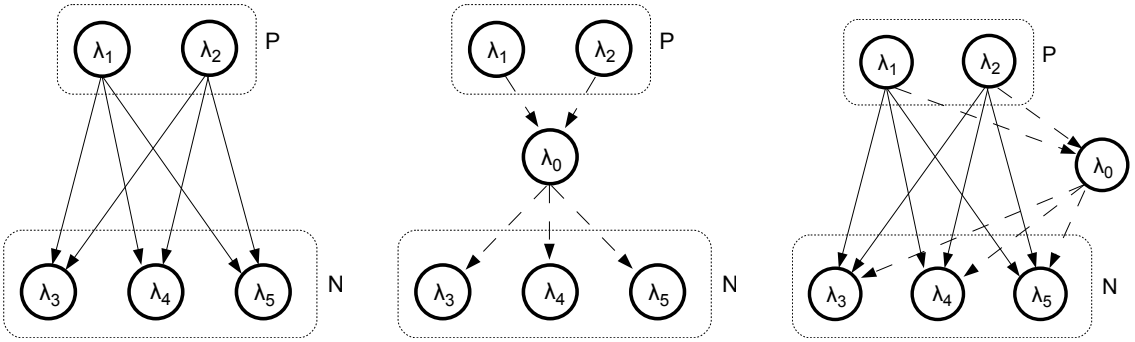


Figure 4.1: Preferences in calibrated label ranking: on the left, we see all preferences between the relevant labels $P_{\mathbf{x}} = \{\lambda_1, \lambda_2\}$ and the irrelevant labels $N_{\mathbf{x}} = \{\lambda_3, \lambda_4, \lambda_5\}$, the center graph shows the position of the virtual label $v = \lambda_0$, and the right graph shows all generated preferences (the union of the previous two graphs).

More precisely, each original training example occurs in all of the n BR classifiers, whereas it only occurs in $|P_{\mathbf{x}}|(n-|P_{\mathbf{x}}|)$ of the quadratic number of pairwise classifiers, with $|P_{\mathbf{x}}|$ being usually rather small (< 5). The fact that these examples are distributed over a larger number of different classifiers makes the pairwise approach particularly attractive for expensive classifiers like SVMs, because a smaller problem size (in terms of training examples) goes typically hand in hand with an increase of the space where a separating hyperplane can be found. Thus it is very likely for a sub-problem to have a larger margin than the full problem. Because of the same reason, it has also been shown that the complexity for training an ensemble of pairwise classifiers is comparable to the complexity of training a BR ensemble. In fact, the algorithm is practical for problems with several thousands of labels (Fürnkranz, 2002; Loza Mencía and Fürnkranz, 2010). Although we have to evaluate a quadratic number of classifiers in order to predict a full ranking, the prediction phase can also be considerably sped up in cases where we only need a small number of relevant labels (Loza Mencía et al., 2010).

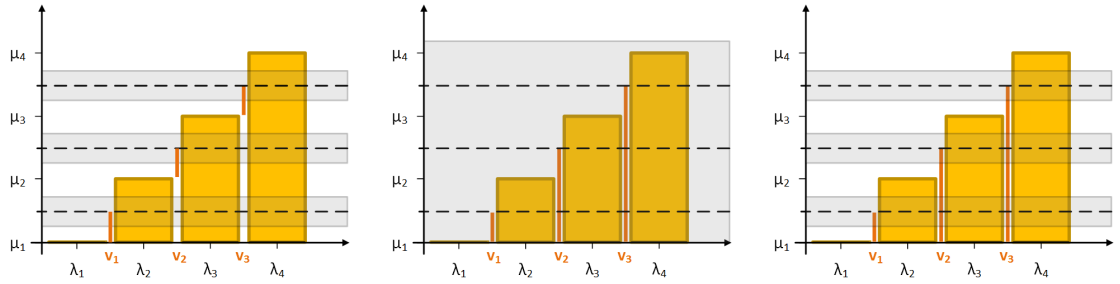
4.2 Multiple Calibration Labels

The key idea of the proposed pairwise approach to graded multilabel classification is to generalize calibrated label ranking to the case of multiple calibration labels $\mathbb{V} = \{v_1, \ldots, v_{m-1}\}$, where each label represents an intermediate grade v_i between the original grades μ_i and μ_{i+1}. Hence, we obtain $\mathbb{M}_v = \mathbb{M} \cup \mathbb{V}$ with the inner structure

$$\mu_1 \prec v_1 \prec \mu_2 \prec v_2 \prec \mu_3 \prec \ldots \prec v_{m-1} \prec \mu_m$$

As a consequence, we obtain an extended set of labels $\mathbb{L} \cup \mathbb{V}$. Note that we use $\mathbb{V}$ to denote both, labels and grades, which conveniently emphasizes the fixed mapping between grade and label v_i, i.e. it generally holds $L(v_i) = v_i$.

Furthermore, in order to cover the case that some training instances may be ignored by certain pairwise classifiers, we introduce the projection function $[p]_{rp} : \mathbf{x} \to \{0, 1, \varnothing\}$ which indicates to use a training example $\mathbf{x}$ either as positive (1), negative (0) example or not at all ($\varnothing$) for the given decomposition rp. Let us further also assume that the pairwise base classifiers are symmetric, i.e. $[H]_{\lambda_u,\lambda_v} = 1 - [H]_{\lambda_v,\lambda_u}$.



Generated preferences

$\lambda_1 \prec v_1 \prec \lambda_2, \lambda_3, \lambda_4$
$\lambda_1, \lambda_2 \prec v_2 \prec \lambda_3, \lambda_4$
$\lambda_1, \lambda_2, \lambda_3 \prec v_3 \prec \lambda_4$

$\lambda_1 \prec v_1 \prec \lambda_2 \prec v_2 \prec \lambda_3 \prec v_3 \prec \lambda_4$

$\lambda_1 \prec v_1 \prec \lambda_2, \lambda_3, \lambda_4, v_2, v_3$
$v_1, \lambda_1, \lambda_2 \prec v_2 \prec \lambda_3, \lambda_4, v_3$
$v_1, v_2, \lambda_1, \lambda_2, \lambda_3 \prec v_3 \prec \lambda_4$

General case ($i = 1 \dots m-1$)

$\bigcup_{j=1}^{i} P'^{j}_{\mathbf{x}} \prec \{v_i\} \prec \bigcup_{j=i+1}^{m} P'^{j}_{\mathbf{x}}$

(a) Horizontal CLR

$P'^{i}_{\mathbf{x}} \prec \{v_i\} \prec P'^{i+1}_{\mathbf{x}}$

(b) Full CLR

$\{v_1 \dots v_{i-1}\} \cup \bigcup_{j=1}^{i} P'^{j}_{\mathbf{x}} \prec \{v_i\}$
$\{v_i\} \prec \bigcup_{j=i+1}^{m} P'^{j}_{\mathbf{x}} \cup \{v_{i+1} \dots v_{m-1}\}$

(c) Joined CLR

Figure 4.2: The three different approaches for a pairwise decomposition of a graded multilabel problem, showing also exemplarily the generated preferences and the general case ($i = 1 \dots m-1$).

4.3 Horizontal Calibrated Label Ranking

The first, simple approach to generalize calibrated label ranking to the graded case is to use the horizontal decomposition as described in Section 3.2, and to solve each of the resulting multilabel problems with CLR. Thus, in order to learn each $[H]^i$, we choose grade v_i as our cutting point, i.e. we only differentiate between grades greater or smaller than v_i. Translated to CLR, v_i becomes the calibrating label and $\cup_{v_i \prec \mu_j} P'^{j}_{\mathbf{x}}$ and $\cup_{\mu_i \prec v_j} P'^{j}_{\mathbf{x}}$ our positive and negative set of labels, respectively, as is illustrated in Figure 4.2(a).

More precisely, we train each $[H]^i_{\lambda_u,\lambda_v}$, $\lambda_u \neq \lambda_v$, $\lambda_u, \lambda_v \in \mathbb{L} \cup \{v_i\}$ using training examples $(\mathbf{x}, [p]^i_{\lambda_u,\lambda_v}(\mathbf{x}))$ given by

$$[p]^i_{\lambda_u,\lambda_v}(\mathbf{x}) = \begin{cases} 1 & \text{if } [L]^i_{\mathbf{x}}(\lambda_v) \prec [L]^i_{\mathbf{x}}(\lambda_u) \\ 0 & \text{if } [L]^i_{\mathbf{x}}(\lambda_u) \prec [L]^i_{\mathbf{x}}(\lambda_v) \\ \varnothing & \text{if } [L]^i_{\mathbf{x}}(\lambda_u) = [L]^i_{\mathbf{x}}(\lambda_v) \end{cases} \tag{4.1}$$

and

$$[L]^i(\lambda_u) = \begin{cases} \mu_i & \text{if } \lambda_u \prec v_u \\ \mu_{i+1} & \text{if } v_u \prec \lambda_u \end{cases} \tag{4.2}$$

For making a prediction for a test instance $\mathbf{x}$, the votes $h_{\mathbf{x}}(\lambda_u) = \sum_{\lambda_u \neq \lambda_v} [H]^i_{\lambda_u,\lambda_v}(\mathbf{x})$ are summed up for each label $\lambda_u \in \mathbb{L} \cup \{v_i\}$, and λ_u is predicted as relevant if $h_{\mathbf{x}}(\lambda_u) > h_{\mathbf{x}}(\lambda_{v_i})$. The final graded prediction is obtained by using the maximum predicted score for each label, as described in Section 3.2.

4.4 Full Calibrated Label Ranking

The idea of the full calibrated label ranking approach is to consider the targets in a GMLC problem as a multipartite ranking. We therefore transform $\mathbf{y}$ into the multipartite ranking $P'^1_{\mathbf{x}} \prec P'^2_{\mathbf{x}} \ldots \prec P'^m_{\mathbf{x}}$ (cf. Figure 4.2(b)). Enriched by the virtual labels we eventually obtain

$$P'^1_{\mathbf{x}} \prec \{v_1\} \prec P'^2_{\mathbf{x}} \ldots \prec \{v_{m-1}\} \prec P'^m_{\mathbf{x}}$$

Obviously, for $m = 2$, this reduces to calibrated label ranking with $P_{\mathbf{x}} = P'^1_{\mathbf{x}}$ and $N_{\mathbf{x}} = P'^2_{\mathbf{x}}$.

The projection function for base classifiers $[H]_{\lambda_u,\lambda_v}$, $\lambda_u \neq \lambda_v$, $\lambda_u, \lambda_v \in \mathbb{L} \cup \mathbb{V}$ only slightly changes in comparison to (4.1), namely into

$$[p]_{\lambda_u,\lambda_v}(\mathbf{x}) = \begin{cases} 1 & \text{if } L(\lambda_v) \prec L(\lambda_u) \\ 0 & \text{if } L(\lambda_u) \prec L(\lambda_v) \\ \varnothing & \text{if } L(\lambda_u) = L(\lambda_v) \end{cases} \tag{4.3}$$

Note that in contrast to the horizontal decomposition in Sec. 4.3 we can sum up the votes across the grades, obtaining one global ranking over all labels and grades. After querying all $(n+m-1)(n+m-2)/2$ base classifiers, we then predict $\hat{y}^j = \arg\max_{\mu_i} h_{\mathbf{x}}(\lambda_j) > h_{\mathbf{x}}(\lambda_{v_i})$ for λ_j.

A possible disadvantage of this approach is that the algorithm is prone to producing many ties in the ranking since $n+m-1$ labels have to be ordered on a scale of 0 to $n+m-2$ obtainable votes. This can potentially be remedied using a different voting function like weighted voting. However, we observed that predicting accurate and comparable scores such as confidences or probabilities is not a trivial task. Hence, 0-1 voting is more robust and makes the fewest assumptions on the base classifiers. We will restrict ourselves to this approach in this technical report. Another, related problem is that *preference intensities* are not considered, i.e., the difference between the grades of two compared labels is ignored, for training as well as during prediction. The joined CLR approach, described in the next section, provides a solution to this.

4.5 Joined Calibrated Label Ranking

On the one hand, Full CLR is not able to capture different degrees of preference intensities since the preference between two labels λ_u, λ_v is only obtained in a binary way. On the other hand, we recall that in the horizontal approach we learn each discriminating classifier $[H]^i_{\lambda_u,\lambda_v}$ exactly $m-1$ times, once for every grade transition. In fact, the number of classifiers λ_u vs. λ_v which use a training instance $\mathbf{x}$ depends on the difference between the grades of the labels, more precisely, it is exactly $|y^u_{\mathbf{x}} - y^v_{\mathbf{x}}|$. We can hence expect that the difference in the number of votes between both labels correlates with the difference in the true grades. A solution, which would take such predictions with varying intensity into account, is to compute a common, joint ranking across degrees and labels, i.e. to compute $s(\lambda_u) = \sum_{\mu_i} \sum_{\lambda_v \neq \lambda_u} [H]^i_{\lambda_u,\lambda_v}$ for all $\lambda_u, \lambda_v \in \mathbb{L} \cup \mathbb{V}$. Although this would possibly produce a good ranking over the labels in $\mathbb{L}$. Unfortunately, it cannot be expected to provide a good ranking over the virtual labels, because each of the virtual labels only appears in one horizontal sub-problem and can therefore only obtain at most n votes. In contrast, each of the real labels can obtain up to $n(m-1)$ votes.

Joined CLR solves this problem by generalizing the horizontal decomposition introduced above, so that all virtual labels are always used in all horizontal sub-problems. More precisely it decomposes the initial problem into $m-1$ bi-partite (three-partite if we count the virtual label) ranking problems with one main calibrating label v_i on each grade transition. In this regard, Joined CLR is equivalent to Horizontal Calibrated Label Ranking and all pairwise base classifiers learned by Horizontal CLR are also learned in exactly the same manner by Joined CLR. On the other hand, as shown in Figure 4.2(c), Joined CLR also adds all remaining virtual labels $v_j \neq v_i$ into these bi-partite ranking problems allowing

them to accumulate the necessary voting mass. The resulting problem remains bi-partite, since we map all grades to μ_i and μ_i+1 as in Horizontal CLR. Using a simplified informal representation, this basically means that in addition to the comparisons

$$\mu_1, \ldots, \mu_i \prec \nu_i \prec \mu_{i+1}, \ldots, \mu_{m-1}$$

each horizontal subproblems is enriched with the following preferences:

$$\mu_1, \ldots, \mu_i \prec \nu_{i+1}, \ldots, \nu_{m-1}$$

$$\nu_1, \ldots, \nu_{i-1} \prec \mu_{i+1}, \ldots, \mu_m$$

$$\nu_1, \ldots, \nu_{i-1} \prec \nu_i \prec \nu_{i+1}, \ldots, \nu_{m-1}$$

More formally, we learn classifiers $[H]^i_{\lambda_u,\lambda_v}$ using $[p]$ and $[L]$ from Eq. (4.1) and Eq. (4.2), but in this case for each $\lambda_u \neq \lambda_v,\ \lambda_u, \lambda_v \in \mathbb{L} \cup \mathbb{V}$. Note that the training signal between two virtual labels is always fixed. Hence, we can set $[H]^i_{\nu_u,\nu_v}(\mathbf{x}) = 0$ if $\nu_u \prec \nu_v$, 1 otherwise, for $\nu_u \neq \nu_v,\ \nu_u, \nu_v \in \mathbb{V}$.

During prediction, the votes for each label are aggregated across all grade transitions as proposed in the beginning of this subsection.

Note that fixing the predictions between virtual labels can introduce a bias since these predictions are always perfect, whereas the remaining predictions depend on the classification performance of a classifier trained on potentially noisy data. This problem can be alleviated e.g. by allowing different fixed values than 0 and 1 or by removing some comparisons. We are currently developing such methods and leave the investigation for further work.

5 Experiments

In this section, we describe the data and setup of the experiments, followed by the results.

5.1 Data & Experimental setup

An overview over the used datasets is given in Table 5.1. The BELA-E benchmark was used in previous work, whereas MOVIES and MEDICAL are two new real-world datasets. [1]

5.1.1 BeLa-E

Cheng et al. (2010) used a dataset obtained from a questionnaire (Abele-Brehm and Stief, 2004). This dataset called BELA-E consists of 1930 instances each representing a graduate student. Each instance has 50 attributes. Two attributes, age and sex, characterize the student, the remaining 48 attributes represent the actual questions to the students, which were on the importance of certain properties of their future jobs. Each of these answers has a grade from '1' (completely unimportant) to '5' (very important), so they share a common inner structure on $\mathbb{M}$. In view of the lack of a more comprehensive and informative characterization of the students, Cheng et al. (2010) decided to use a subset of the question answers as additional attributes for characterizing the students. Following the same setup, we generated 50 datasets by choosing randomly a subset of n questions as target labels. The remaining $50 - n$ attributes were used as features of the instances. We generated two kinds of datasets, for $n = 5$ and $n = 10$, respectively.

5.1.2 Movies

We collected a dataset from the German TV program guide www.TVSpielfilm.de which rates movies by assigning grades to the categories 'fun', 'action', 'sex', 'suspense' and 'sophistication' rather than giving an overall rating. Each category has grades from '0' to '3'. We interpret the grades as a mixture of degree of presence or relevance and degree of quality. The background is that a classic comedy film could be rated as 'not funny at all' by the editorial team. In total, we had data for 1967 movies. For characterizing them, we extracted the associated summary texts from www.imdb.org. Furthermore, we added the English title, the year, director's name, actors' names, characters' names, writers' names, runtime, country of origin, and language as text to the summary. The text was tokenized, stemmed with the Porter algorithm and common English stopwords were filtered. We computed then the TF-IDF values of the tokens on the respective training data of the 10-fold cross validation.

5.1.3 Medical

The MEDICAL dataset consists of 1953 free text radiology reports. They were collected for the CMC's 2007 Medical Natural Language Center (Pestian et al., 2007) and three expert companies were asked to annotate them with a set of ICD-9-CM disease/diagnosis classification codes. In the original dataset for the multilabel classification competition, a document was assigned to a code if there was a consensus among at least two of the annotators on a specific code. In contrast, we generated a GMLC dataset by

[1] The datasets are available at http://www.ke.tu-darmstadt.de/resources/GMLC

Table 5.1: Overview of datasets used in the experiments. Shown are the total number of instances, attributes, unique labels n, different grades m, the average grade index and the frequency of the specific grades μ_i appearing in the label–instance mappings.

						distribution of grades $\mu_i, i =$				
Dataset	Instances	Attributes	Labels	Grades	Avg. Grade	1	2	3	4	5
BeLa-E n=5	1930	45	5	5	2.50	7.95	13.04	23.89	31.43	23.69
BeLa-E n=10	1930	40	10	5	2.50	7.95	13.04	23.89	31.43	23.69
Movies	1967	27002	5	4	0.72	50.26	31.13	15.18	3.43	–
Medical	1953	1602	204	4	0.02	99.08	0.31	0.24	0.37	–

considering the level of agreement as grade of assignment. We expect a more distinguished and useful automatic classification than by only using the merged mappings in the golds standard, however, this was not evaluated. Note that it lies in the nature of the problem that the assignments are very sparse in the sense that labels are very likely to be absent. The texts were processed as for Movies but we used the absolute term frequency in contrast to TF-IDF.

5.1.4 Experimental setup

Al proposed approaches, except the IBLR-ML(which we obtained from the authors), were implemented as part of the LPCforSOS framework, which is an extension of the Weka framework Hall et al. (2009)[2]. The code of the IBLR-ML was provided by the authors of Cheng et al. (2010). We used the J48 classifier of the Weka framework as binary base classifier, which is an implementation of the C4.5 decision tree learner algorithm (Quinlan, 1993). The complete reduction approach was implemented by using horizontal reduction with binary relevance decomposition (referred to as BR). We used the ordinal classification method of Frank and Hall (cf. Sec. 2.1) in the implementation of the Weka framework for the vertical reduction (F&H). On each of the datasets we obtained our results by averaging the evaluation measures on the test folds of a 10-fold cross validation. In addition, on the BeLa-E datasets, we averaged the results on different datasets. Although this is commonly not good practice, in this case the individual datasets are generated from the same original dataset and, as stated by Cheng et al., should be evenly distributed.

For calculating the rank losses for the complete reduction approaches (BR and F&H), the IBLR-ML and the horizontal calibrated label ranking (H-CLR), the predicted grade is used as the score.

5.2 Losses

For the GMLC problem, Cheng et al. generalized several common losses for multilabel classification. We will discuss the measures and their meaning in short. All losses are computed individually on the instances and averaged first on the test set and afterwards on the 10 test folds.

[2] See http://www.lpcforsos.sf.net and http://www.cs.waikato.ac.nz/ml/weka/

5.2.1 Hamming Loss

In the GMLC the *Hamming loss* can be generalized by measuring the original loss on the sub-tasks of either the horizontal or vertical reduction. Cheng et al. showed that both functions are equal to each other. For simplification the hamming loss is used in the vertical version

$$\textsc{HammLoss}\left(\hat{y}_{\mathbf{x}}, y_{\mathbf{x}}\right) = \frac{\sum_{i=1}^{n} AE\left(\hat{y}_{\mathbf{x}}^{i}, y_{\mathbf{x}}^{i}\right)}{(m-1)\cdot n}$$

with $AE : \mathbb{M} \times \mathbb{M} \to \mathbb{N}, AE\left(\mu_i, \mu_j\right) = |i-j|$. Thus, the hamming loss in graded multilabel classification denotes the mean deviation of the predicted label grades to the real ones. Even though the use of grade distances for the evaluation of ordinal predictions is questionable from a theoretical point of view (cf. Section 2.1), it is nevertheless a useful indicator of classification performance in such settings.

5.2.2 Vertical 0-1 Loss

The *vertical 0-1 loss* measures the percentage of labels with incorrectly assigned grades. Contrary to *Hamming loss*, this metric does not consider the size of the grade differences.

$$\textsc{Vert01}\left(\hat{y}_{\mathbf{x}}, y_{\mathbf{x}}\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(\hat{y}_{\mathbf{x}}^{i} \neq y_{\mathbf{x}}^{i}\right)$$

5.2.3 C-Index

The *C-index* (Gönen and Heller, 2005) is a generalization of the rank loss. To fit the graded case, it measures the pairwise ranking error between a pair of labels out of two different sets $P_{\mathbf{x}}^{\prime i}, P_{\mathbf{x}}^{\prime j}, i < j$. Essentially, the *C-index* counts the number of incorrectly ordered pairs of labels with different grade in the ranking.

$$\text{C-}\textsc{Index}\left(h_{\mathbf{x}}, P_{\mathbf{x}}^{\prime 1}, \ldots, P_{\mathbf{x}}^{\prime m}\right) = \frac{\sum\limits_{\mu_i < \mu_j} \sum\limits_{(\lambda,\lambda') \in P_{\mathbf{x}}^{\prime i} \times P_{\mathbf{x}}^{\prime j}} S\left(h_{\mathbf{x}}(\lambda), h_{\mathbf{x}}\left(\lambda'\right)\right)}{\sum\limits_{\mu_i < \mu_j} \left|P_{\mathbf{x}}^{\prime i} \times P_{\mathbf{x}}^{\prime j}\right|}$$

with $S(u,v) = \mathbb{I}(u > v) + \frac{1}{2}\mathbb{I}(u = v)$. For Full CLR and Joined CLR, we can take the number of votes for each label as ordering criterion. For the remaining ones we just use the predicted grade. Basically, this corresponds to the comparisons considered by the Full CLR approach.

5.2.4 One Error Rank Loss

This metric is the generalization of the *one error* loss for rankings in multilabel classification. In Cheng et al. (2010) this loss is generalized to measure if the highest ranked label has the highest possible grade μ_m. The drawback of this version is that if an instance out of the test set has no label with a relevance of the highest possible grade, the *one error* cannot be zero, even if the classification of the instance is completely correct. To solve this problem we propose a changed version of the *one error* comparing the real grade of the highest ranked label with the highest grade of all labels of an instance.

$$\textsc{OneErr}\left(\hat{y}_{\mathbf{x}}^{i}, y_{\mathbf{x}}^{i}\right) = \frac{1}{m-1} AE\left(\max_{1 \leq i \leq n} \hat{y}_{\mathbf{x}}^{i}, \max_{1 \leq j \leq n} y_{\mathbf{x}}^{j}\right)$$

5.2.5 Optimistic Hamming Loss

Under some circumstances, CLR tends to under- or overestimate the correct position of the virtual label. In order to be independent of such an effect, we follow the idea of Fürnkranz et al. (2008) and propose to evaluate the ranking performance by *cheating* on the positioning of the virtual label: we place the cutting points in hindsight so that the distribution of grades corresponds to the real one. In a way this allows us to compute bi-partitioning metrics even when the underlying algorithm can only predict rankings. Furthermore, it allows us to compute the *regret* of using a specific cutting technique.

We generalize this method to the GMLC and the multi-partite ranking case, respectively. Therefore, we define the cheated partitioning $\hat{P}''^1_{\mathbf{x}}, \hat{P}''^2_{\mathbf{x}}, \ldots$ over a ranking such that $|\hat{P}''^i_{\mathbf{x}}| = |P'^i_{\mathbf{x}}|$ and $s_{\mathbf{x}}(\lambda_u) \leq s_{\mathbf{x}}(\lambda_v)$ if $\lambda_u \in P'^i_{\mathbf{x}}, \lambda_v \in P'^j_{\mathbf{x}}, \mu_i \prec \mu_j$. Given the corresponding prediction (in the form of $\hat{\mathbf{y}}''$), we thus obtain the optimistic Hamming loss as

$$\textsc{OptHammLoss} = \textsc{HammLoss}\left(\hat{\mathbf{y}}''_{\mathbf{x}}, \mathbf{y}_{\mathbf{x}}\right)$$

5.3 Results

The experimental results are summarized in Table 5.2. The first observation is that BR, i.e., the complete reduction using horizontal and vertical cuts, is usually outperformed by the pairwise approaches, even for Hamming loss. Moreover, BR is always outperformed by F&H, even though both classifiers are trained equally. The difference is due to the different aggregation strategies of the predictions of the binary classifiers (see Sec. 3.1 and 3.2), and obviously, the more sophisticated approach by Frank and Hall pays off for these datasets.

The next observation is that the approach using IBLR-ML shows even worse results than BR. This is surprising, since it does not correspond to the results reported by Cheng et al. (2010), where BR is beaten by IBLR-ML, although we used the code provided by the authors. A reason might be that the 50 sub-datasets are obviously not exactly equal due to the random initialization. Furthermore, we used a different base learner for BR which explains the differences for this algorithm, but not the ones for IBLR-ML, which was used exactly the same way as in Cheng et al. (2010).

Still, our results for C-Index and one error seem more reasonable to us since IBLR-ML uses the same overestimating aggregation as BR. Horinzontal CLR also uses this aggregation but pairwise classification is an ensemble method and thereby is more robust to noise predictions of single classifiers. So the IBLR-ML approach is probably more prone to error propagation.

Interestingly, the approach using vertical reduction (F&H) seems to perform quite competitive w.r.t. other approaches, especially for Hamming and vertical 0-1 loss. This may show that preserving and focusing on the information about the grades (vertical) is more important for GMLC than considering the relations between the labels at each grade (horizontal). On the other hand, Horizontal CLR outperforms F&H specifically on exactly these both losses (except for MEDICAL, where they perform equally). On the BELA-E datasets, all approaches are pairwise statistically significantly different with $\alpha = 0.01$ (sign test).

So the graded multilabel classification seems not to have so much gain towards solving several independent ordinal classification problems towards these losses. This maybe is grounded in the used datasets but is visible in all of them. Looking at the ranking of the labels and thereby the C-Index and the one error the approach cannot compete with the calibrated label ranking approaches. So the interdependencies of the labels seems to have its main impact to these measures.

The results of the different calibrated label ranking approaches show a high correspondence to their inner structure. The Full CLR shows the highest Hamming and vertical 0-1 loss among the approaches. When looking at its Optimistic Hamming loss and the quite good C-Index and one error, this seems to be clearly just a problem of the correct positioning of the virtual labels due to the narrowness and thus

ties in the rankings (see 4.4). The Joined CLR shows a similar behavior. In particular, we can observe that Joined CLR has the best results among the approach for all three ranking losses, except on the MEDICAL dataset. The somewhat worse results on the medical dataset suggest that the Joined CLR has problems on datasets with many labels being assigned too extreme low or high grades (see Tab. 5.1).

As already mentioned, Horizontal CLR outperforms all other approaches w.r.t. Hamming and vertical 0-1 loss. This is very likely due to the easier positioning of the single calibrating label, especially in comparison to Full CLR but also to Joined CLR. On the other hand, Horizontal CLR reveals its disadvantages regarding the prediction of good rankings. It is the worst approach compared to the other pairwise methods w.r.t. C-Index and one error. It seems very obvious that the aggregation strategy of selecting the highest seen grade for each label, also used by BR and IBLR-ML and proposed by Cheng et al., is not advantageous w.r.t. ranking quality.

In summary, the pairwise approaches generally outperform all other approaches on the used ranking losses. Especially the full and joined decomposition provide a clear advantage when good rankings of labels are important. On the other hand, if we desire good predictions for each label independently (hence for each ordinal problem separately), then Horizontal CLR is the most appropriate method among all evaluated techniques in our experiments.

These two main results make us confident that learning by pairwise comparisons has a natural access to the inner structure of GMLC problems. Moreover, it was shown that pairwise learning provides a flexible adaptation to different objectives by adjusting decomposition and aggregation. The very low optimistic Hamming losses of the CLR approaches additionally promise an even better result of the CLR algorithms through finding a better way of positioning the virtual labels into the global ranking.

Table 5.2: Results of the three pairwise graded multilabel algorithms in comparison to IBLR-ML and two benchmarks. In addition to the results of the five different loss functions in terms of percentage ($\times 100$), we show the standard deviation for BELA-E and the average rank of each algorithm on the particular dataset in parenthesis.

Dataset	Evaluation Measure	IBLR-ML	BR	F&H	Full CLR	Joined CLR	Horizontal CLR
BELA-E $n = 5$	Hamming Loss	27.23 (4)± 4.51	28.07 (5)± 2.62	16.08 (2)± 1.65	33.97 (6)± 5.79	17.96 (3)± 1.31	**15.77 (1)**± 1.53
	Optimistic Hamming Loss	–	–	–	11.00 (2)± 1.70	**9.62 (1)**± 1.45	–
	Vertical 0-1 Loss	69.39 (5)± 5.39	61.27 (3)± 4.19	51.97 (2)± 3.68	73.44 (6)± 7.58	61.82 (4)± 3.61	**51.90 (1)**± 3.52
	C-Index	49.55 (6)± 8.44	32.63 (5)± 3.19	24.34 (4)± 4.25	20.38 (2)± 4.13	**18.16 (1)**± 3.68	23.88 (3)± 4.11
	One Error Loss	27.80 (6)± 7.46	12.89 (5)± 3.20	11.35 (4)± 2.45	8.50 (2)± 2.25	**7.19 (1)**± 1.82	11.06 (3)± 2.31
BELA-E $n = 10$	Hamming Loss	27.27 (4)± 3.83	27.77 (5)± 1.83	16.04 (2)± 1.04	35.44 (6)± 3.70	17.92 (3)± 0.87	**15.13 (1)**± 0.95
	Optimistic Hamming Loss	–	–	–	12.70 (2)± 0.94	**12.03 (1)**± 0.91	–
	Vertical 0-1 Loss	69.95 (5)± 4.16	61.17 (3)± 2.69	51.97 (2)± 2.23	75.11 (6)± 4.47	61.76 (4)± 0.87	**50.45 (1)**± 2.15
	C-Index	50.37 (6)± 6.98	32.85 (5)± 3.45	24.14 (4)± 2.68	18.57 (2)± 2.27	**17.58 (1)**± 2.14	22.78 (3)± 2.53
	One Error Loss	34.47 (6)± 9.23	17.03 (5)± 4.38	12.92 (4)± 2.53	8.19 (2)± 1.67	**7.77 (1)**± 1.28	11.56 (3)± 1.93
MOVIES	Hamming Loss	32.33 (5)	21.94 (3)	18.95 (2)	76.51 (6)	25.32 (4)	**17.73 (1)**
	Optimistic Hamming Loss	–	–	–	9.58 (2)	**8.98 (1)**	–
	Vertical 0-1 Loss	67.34 (5)	50.85 (3)	47.86 (2)	96.50 (6)	67.16 (4)	**44.70 (1)**
	C-Index	33.98 (6)	30.86 (5)	23.12 (4)	15.43 (2)	**14.74 (1)**	21.40 (3)
	One Error Loss	15.43 (5)	18.43 (6)	14.24 (4)	9.30 (2)	**7.75 (1)**	12.21 (3)
MEDICAL	Hamming Loss	1.30 (3)	0.31 (2)	**0.26 (1)**	3.00 (4)	10.34 (5)	**0.26 (1)**
	Optimistic Hamming Loss	–	–	–	**0.23 (1)**	0.31 (2)	–
	Vertical 0-1 Loss	2.07 (3)	0.62 (2)	**0.60 (1)**	3.81 (4)	21.87 (5)	**0.60 (1)**
	C-Index	49.96 (6)	18.40 (5)	10.73 (3)	**3.27 (1)**	5.20 (2)	12.06 (4)
	One Error Loss	90.89 (6)	20.93 (5)	11.76 (3)	**10.44 (1)**	10.65 (2)	12.71 (4)

6 Conclusions

In this work, we introduced pairwise comparisons for representing and learning graded multilabel classification (GMLC) problems, which are a combination of ordinal and multilabel classification problems, where each instance is associated with several different grades of relevance to multiple categories at the same time. To be able to solve such problems by learning from pairwise comparisons we generalized Calibrated Label Ranking to the case of multiple calibration labels. We presented three different generalizations of CLR to graded multi-label classification, and experimentally compared them to previous work by Cheng et al. (2010) on three different datasets. In these experiments, our approaches achieved the best results in all measured losses during the experiments.

Nevertheless, we believe that we have not yet fully exploited the information that is inherent in GMLC problems. In particular, we believe that pairwise comparisons have the capacity to achieve even better results by improving the way the predicted ranking is separated into grades. In future work, we plan to investigate alternative aggregation strategies to the horizontal reduction, the use of different voting strategies like weighted voting, as well as novel approaches for introducing the virtual labels into the label rankings.

Acknowledgements

This research has been partially funded by the German Science Foundation (DFG). We would like to thank Weiwei Cheng and Eyke Hüllermeier for fruitful discussions and making their data and algorithms available.

Bibliography

A. E. Abele-Brehm and M. Stief. Die Prognose des Berufserfolgs von Hochschulabsolventinnen und -absolventen. *Zeitschrift für Arbeits- und Organisationspsychologie*, 48(1):4–16, 2004. ISSN 0033-2992; 0932-4089. 15

W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009. doi: 10.1007/s10994-009-5127-5. 8

W. Cheng, K. Dembczyński, and E. Hüllermeier. Graded multilabel classification: The ordinal case. In *Proceedings of the 27th International Conference on Machine Learning*, pages 223–230, 2010. 3, 4, 7, 8, 15, 16, 17, 18, 19, 21

K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012. 3

E. Frank and M. Hall. A simple approach to ordinal classification. In *Proceedings of the 12th European Conference on Machine Learning (ECML-01)*, pages 145–156, Freiburg, Germany, 2001. Springer-Verlag. 5, 8, 16

J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002. 10, 11

J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, June 2008. doi: 10.1007/s10994-008-5064-8. 3, 10, 18

S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery in Databases (PAKDD-04*. 9

M. Gönen and G. Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005. 17

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, Nov. 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. 16

E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008. 10

D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, Cambridge, UK, 2010. ISBN 978-0-521-49336-9. 3

E. Loza Mencía and J. Fürnkranz. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts – Where the Language of Law Meets the Law of Language*, pages 192–215. 1 edition, 2010. 11

E. Loza Mencía, S.-H. Park, and J. Fürnkranz. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing*, 73(7-9):1164 – 1176, 2010. ISSN 0925-2312. doi: 10.1016/j.neucom.2009.11.024. 11

E. Montañés, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier. Dependent binary relevance models for multi-label classification. *Pattern Recognition*, 47(3):1494 – 1508, 2014. ISSN 0031-3203. doi: 10.1016/j.patcog.2013.09.029. 9

J. Pestian, C. Brew, P. Matykiewicz, D. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007 at ACL 2007*, 2007. 15

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993. 16

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011. ISSN 0885-6125. doi: 10.1007/s10994-011-5256-5. 9

G. Tsoumakas, A. Dimou, E. Spyromitros Xioufis, V. Mezaris, I. Kompatsiaris, and I. P. Vlahavas. Correlation based pruning of stacked binary relevance models for multi-label learning. In *Proc 1st Int Workshop on Learning from Multi-Label Data (MLD'09)*, pages 101–116, 2009. 9

G. Tsoumakas, I. Katakis, and I. P. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2010. ISBN 978-0-387-09823-4. doi: 10.1007/978-0-387-09823-4_34. 3, 5

M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 99, 2013. ISSN 1041-4347. doi: http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.39. PrePrints. 5