Pattern Recognition Letters
journal homepage: www.elsevier.com

Interval prediction for graded multi-label classification

Gerardo Lastra[a], Oscar Luaces[b,**], Antonio Bahamonde[b]

[a]CERN, CH-1211 Geneva 23, Switzerland
[b]Artificial Intelligence Center, University of Oviedo at Gijón, 33204 Gijón, Spain

ARTICLE INFO

Article history:
Received —
Received in final form —
Accepted —
Available online —

Communicated by —

Keywords:
Graded multi-label classification
Nondeterministic classification
Interval classification

ABSTRACT

Multi-label was introduced as an extension of multi-class classification. The aim is to predict a set of classes (called labels in this context) instead of a single one, namely the set of relevant labels. If membership to the set of relevant labels is defined to a certain degree, the learning task is called graded multi-label classification. These learning tasks can be seen as a set of ordinal classifications. Hence, recommender systems can be considered as multi-label classification tasks. In this paper, we present a new type of nondeterministic learner that, for each instance, tries to predict at the same time the true grade for each label. When the classification is uncertain for a label, however, the hypotheses predict a set of consecutive grades, i.e., an interval. The goal is to keep the set of predicted grades as small as possible; while still containing the true grade. We shall see that these classifiers take advantage of the interrelations of labels. The result is that, with quite narrow intervals, it is possible to obtain dramatic improvements in the number of right predictions compared with those achieved by a state-of-the-art deterministic learner which always predicts only one grade for all labels.

1. Introduction

Multi-label classification (MLC) has recently received increasing attention from the Machine Learning community both as an application field and as an intellectual challenge. Given an instance, the aim in MLC is to *simultaneously* obtain a collection of binary classifications. In other words, each instance has a *set of labels* attached, the *relevant* labels, instead of a single one, as occurs in multi-class classification tasks.

Tsoumakas et al. have made a detailed presentation of multi-label classification and its applications (Tsoumakas and Katakis, 2007; Tsoumakas et al., 2010). These applications arise in different fields; for instance, in many text document, video, music or movie databases, items are tagged with several labels.

Cheng et al. (2010) extended MLC to consider situations in which a label is relevant to an instance to a *certain degree*. They call this extension *graded* multi-label classification (GMLC). The relevance of each label is represented by a fuzzy set instead of a (crisp) standard 0/1 membership relation. Thus, the set of degrees of relevance or membership are generalized from {0, 1} to a finite ordered set, M, typically represented by a subset of contiguous integers that can be read as linguistic variables.

Additionally, GMLC can be seen as a set of *ordinal classifications*. For each label, instead of a binary classification, GMLC defines a ranking of instances in the set M of degrees of membership. Let us recall that the aim of ordinal classification (sometimes called ordinal *regression*) is to find hypotheses able to predict classes or *ranks* that belong to a finite ordered set, like the set M of degrees of membership.

From this point of view, GMLC is a reasonable framework for handling *recommender systems*. The ratings of users over a collection of items can be considered as grades of membership of those items with respect to the set of *preferable* items. Thus,

**Corresponding author: Oscar Luaces
e-mail: gerardo.lastra@cern.ch (Gerardo Lastra), oluaces@uniovi.es (Oscar Luaces), abahamonde@uniovi.es (Antonio Bahamonde)

items would play the role of labels, while ratings are grades. In the experimental results reported at the end of the paper, we illustrate this application field of GMLC with some datasets built from *Jester*, an online joke recommender system (Goldberg et al., 2001).

On the other hand, the ordinal classifications involved in GMLC tasks can be extended to *nondeterministic* classifiers. In multi-class classification tasks, nondeterministic classifiers are able to predict one or more classes, while traditional (deterministic) classifiers predict only one. The central idea is that nondeterministic classifiers return more than one class when there are reasonable doubts about the right prediction, instead of risking a single prediction. These classifiers were introduced in Alonso et al. (2008); del Coz et al. (2009); Luaces et al. (2011) for multi-class and for ordinal classification tasks, although these approaches are not devised to deal with multi-label data.

In the context of GMLC, nondeterministic classifiers would predict intervals of grades for each label. We shall show that, with quite narrow intervals, the performance of predictors can be dramatically improved in terms of right predictions, while the size of the predicted intervals is forced to be as small as possible. For this purpose, we define the predictions as those with the best expected trade-off between accuracy and size in a sense that will be explained in Section 4. Formally, the multi-label classification of an instance $\boldsymbol{x}$ is defined as the output that optimizes the expected F_1 measure.

The paper is organized as follows. In the next section we present a motivating example of a GMLC task. We then introduce the formal framework for classical, graded and nondeterministic graded multi-label classification. The fifth section is devoted to reporting and discussing a number of experiments carried out to evaluate the proposals put forward in this paper. The last section summarizes some conclusions about the work presented here.

2. A Motivating Example

Let us consider BeLa-E, the dataset employed in Cheng et al. (2010), where graded multi-label classification was introduced. The origin of the data (Abele-Brehm and Stief, 2004) was the result of a poll conducted to find out the opinion of a sample of students about different aspects of their potential future jobs. Each student was asked to grade, on an ordered scale of 5 values, the degree of importance of properties of future jobs, including 'reputation', 'safety', 'high income' and 'friendly colleagues'. The poll records, for each student, the answers to 48 questions plus 2 additional items, the student's sex and age.

There are several reasons to reduce the number of questions in a poll like this one. First, in order to gain insight into the rationale behind the answers, it is useful to discover whether some answers can be deduced from others. Second, to increase the quality of the information gathered by the poll, if it is possible to reduce the number of questions, the students will be more willing to answer the questions while maintaining the necessary attention.

Thus, let us suppose that we want to learn to predict the opinion of students regarding a group of 10 issues according to the answers given to the remaining questions. Notice that, instead of concluding whether the label 'reputation' is relevant or not for a student, the purpose of the learning process is to predict the degree of relevance. This is the framework of graded multi-label classification tasks (Cheng et al., 2010). This is, in fact, a learning task that arises in many recommender systems.

It is a difficult task to learn the exact grade of each label for a number of different factors; we shall see this in detail in Section 5, where we report some experiments conducted with this dataset. The upshot of this situation is that the usefulness of such a learned hypothesis may be limited.

In this paper we explore a type of hypothesis allowed to predict more than one grade for each label in doubtful situations. The idea is to be able to predict, for instance, that a label is *not very* relevant, since the grade is 'very low' or 'low'.

To capture this approach, we need to extend the set of outputs from grades to intervals of grades. The use of intervals instead of arbitrary subsets is important, given that predictions must somehow incorporate the fact that grades are an ordered set.

On the other hand, we must establish a tradeoff between the proportion of true predictions and the size of the intervals. Obviously, an interval including all grades will contain the true one, but that is not useful. To accomplish this task, we shall use a function employed in information retrieval, the F_β presented in the next section.

Furthermore, we can opt for learning each issue separately or all together, trying to take advantage of the interdependence between issues. In Section 4, we prove that if we are willing to make a prediction with a fixed number of grades (joining all the labels together), then we must search for those intervals with the highest sum of probabilities. This implies that a joint strategy outperforms the attempt to optimize the predictions of each label separately.

The results in the poll dataset show that it is possible to dramatically increase the score of a state-of-the-art deterministic learner. The percentage of times that predictions include the true grade rise from 49.35% to 74.58%, while the average number of predicted grades per label (question in the poll) is only 1.64.

3. A Formal Framework for Graded and Nondeterministic multi-label Classification

Let L be a finite and non-empty set of labels $\{l_1, \ldots, l_{|L|}\}$, and let $\mathcal{X}$ be an input space. A multi-label classification task can be represented by a dataset

$$D = \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_{|D|}, Y_{|D|})\}$$

of pairs of instances $\boldsymbol{x} \in \mathcal{X}$ and subsets of labels $Y_{\boldsymbol{x}} \subset L$. The goal is to induce from D a hypothesis defined as follows.

Definition 1. *A* multi-label *hypothesis is a function h from the input space to the set of subsets (power set) of labels; in symbols,*

$$h : \mathcal{X} \longrightarrow \{0, 1\}^L. \tag{1}$$

The prediction $h(\boldsymbol{x})$ can be understood as the set of *relevant* labels retrieved for a *query* $\boldsymbol{x}$. There is a straightforward approach to inducing a multi-label hypothesis from a dataset D, the so-called *Binary Relevance* strategy. For each $l \in L$, induce a binary hypothesis $h_l : \mathcal{X} \longrightarrow \{0, 1\}$, and then define $h(\boldsymbol{x}) = \{l : h_l(\boldsymbol{x}) = 1\}$.

When the set $Y_{\boldsymbol{x}}$ of relevant labels is a *fuzzy* subset of L. That is, $Y_{\boldsymbol{x}}$ is defined by a *membership* function $L \to M$, where M is the discrete set of graded membership degrees, in our case, without any loss of generality, $M = \{0, 1, \ldots, m\}$. In these cases, the goal is to learn a graded multi-label hypothesis.

Definition 2. *A* graded *multi-label hypothesis Cheng et al. (2010) is a function h from the input space to the set of fuzzy subsets of L with membership degrees in M; in symbols,*

$$h : \mathcal{X} \longrightarrow M^L. \tag{2}$$

We shall now take a further step forward by extending M to the set of intervals of M.

Definition 3. *A* nondeterministic *graded multi-label hypothesis is a function h from the input space to the set of fuzzy subsets of L with membership degrees in the set of intervals of M (subsets of consecutive degrees); in symbols,*

$$h : \mathcal{X} \longrightarrow (\mathit{Intervals}(M))^L. \tag{3}$$

Alternatively, a graded multi-label hypothesis h, for each instance $\boldsymbol{x} \in \mathcal{X}$, defines a relation $h(\boldsymbol{x})$ from the set of labels L into the set of grades M. For the sake of coherence with the adjective nondeterministic, the hypotheses whose predictions have always one membership grade, Eq. (2), are called *deterministic*. In the deterministic case, $h(\boldsymbol{x})$ is a function: for each $l \in L$ the prediction is only one grade of M, $h(\boldsymbol{x})(l) \in M$. In general, in the nondeterministic case, we allow more than one grade to be assigned to $h(\boldsymbol{x})(l)$. However, in order to grasp the ordinal meaning of M, the set of grades must be an interval, $h(\boldsymbol{x})(l) \in \mathit{Intervals}(M)$. Thus, the relation $h(\boldsymbol{x})$ can be represented as

$$h(\boldsymbol{x}) = \{(l, g) : l \in L, g \in h(\boldsymbol{x})(l)\} \subset L \times M. \tag{4}$$

Or, alternatively, as a function that returns one interval for each label

$$h(\boldsymbol{x}) = (I_1, \ldots, I_{|L|}) \tag{5}$$

3.1. Loss and Score Functions

Loss and score functions for nondeterministic classifiers must take into account not only whether the true membership grades are included in the predicted intervals, but also the length of these intervals. In order to assess the performance of a graded hypothesis h (deterministic or nondeterministic), it is useful to consider its predictions as the relation in Eq. (4). Given an input instance $\boldsymbol{x} \in \mathcal{X}$, we have to compare the set of predictions $h(\boldsymbol{x}) \subset L \times M$ and a subset of *truly relevant graded* labels $Y \subset L \times M$. For this purpose, we can compute the following contingency matrix,

	Y	$(L \times M) \setminus Y$
$h(\boldsymbol{x})$	a	b
$(L \times M) \setminus h(\boldsymbol{x})$	c	d

(6)

in which each entry (a, b, c, d) is the number of elements of the intersection of the corresponding sets of the row and column. Notice, for instance, that in the binary case ($|M| = 2$), a is the number of relevant labels predicted by h for $\boldsymbol{x}$.

In the most general case, we have that $a + b$ is the number of predictions; i.e., the *size* of the prediction. Moreover, since all labels have exactly one degree of membership, including the lowest (0), which means that they are not relevant *at all*, $a + c$ is the number of labels. In symbols,

$$a + b = |h(\boldsymbol{x})| = |\{(l, g) : l \in L, g \in h(\boldsymbol{x})(l)\}|, \tag{7}$$

$$a + c = |Y| = |L|. \tag{8}$$

Notice that, in the deterministic case, the size of the prediction is equal to the number of labels. For general nondeterministic hypothesis, however, the size is bigger than $|L|$.

From another point of view, the predictions of a graded hypothesis can be considered as the answers to a *query* represented by an instance $\boldsymbol{x}$. Using this metaphor, we can extend the loss and score functions from *information retrieval* to graded multi-label hypotheses. We thus have the following definitions.

Definition 4. *The* Recall *in a* query *(i.e., an instance $\boldsymbol{x}$) is defined as the proportion of relevant labels Y included in $h(\boldsymbol{x})$:*

$$R(h(\boldsymbol{x}), Y) = \frac{a}{a + c}. \tag{9}$$

Definition 5. *The* Precision *is defined as the proportion of retrieved labels in $h(\boldsymbol{x})$ that are truly relevant:*

$$P(h(\boldsymbol{x}), Y) = \frac{a}{a + b}. \tag{10}$$

Finally, the tradeoff between *Recall* and *Precision* is formalized by

Definition 6. *The F_β ($\beta \geq 0$) is defined, in general, by*

$$F_\beta(h(\boldsymbol{x}), Y) = \frac{(1 + \beta^2)PR}{\beta^2 P + R} = \frac{(1 + \beta^2)a}{|h(\boldsymbol{x})| + \beta^2 |L|}. \tag{11}$$

The size of predictions coincides exactly with the number of labels ($|L| = |h(x)|$) for deterministic hypotheses; thus, *Recall*, *Precision*, and F_β have the same value: the proportion of successful grade predictions. However, these score functions take on a proper meaning in nondeterministic hypotheses, as the size of predictions $|h(\boldsymbol{x})|$ is, in general, greater than the number of labels.

To illustrate these concepts let's suppose a graded multi-label dataset where each example belongs to 4 labels with different degrees of membership (grades), ranging each one from 0 to 9. The true values for a given instance could be, for example, $Y = \{9, 3, 5, 2\}$. If a deterministic algorithm makes the prediction $h_{det}(\boldsymbol{x}) = \{9, 4, 5, 2\}$ for this instance, the corresponding contingency matrix is:

	Y	$(L \times M) \setminus Y$
$h_{det}(\boldsymbol{x})$	3	1
$(L \times M) \setminus h(\boldsymbol{x})$	1	35

where $L \times M$ represents all the possible grades (10 in this example) for all the labels (4 in this example). Thus, $a = 3$, since the

grades for the first, third and fourth labels have been correctly predicted by h_{det}.

A deterministic learner predicts just one grade for each label, so the length of a prediction, $|h_{det}(\boldsymbol{x})|$, is equal to the number of labels, $|L|$; Therefore, considering equations (7) and (8), we have that $b = c$ and thus, Recall, Precision and F_1 yield the same value for any deterministic prediction.

However, a nondeterministic learner predicts an interval of grades for each label which can contain several grades. Let's suppose a nondeterministic prediction

$$h(\boldsymbol{x}) = \{9, \{3, 4, 5\}, \{4, 5\}, \{5, 6, 7, 8\}\}.$$

that can be read as:

> *the grade for label 1 is* 9*, for label 2 it is in the interval [3,5] (i.e. it can be 3, 4 or 5), for label 3 it is in [4,5] and for label 4 it is in [5,8].*

The contingency matrix for this prediction/example is:

	Y	$(L \times M) \setminus Y$
$h(\boldsymbol{x})$	3	7
$(L \times M) \setminus h(\boldsymbol{x})$	1	29

which yields different values for Precision, Recall and F_1, since the length of a nondeterministic prediction, $|h(\boldsymbol{x})|$, is usually larger than the number of labels, $|L|$ and thus, $b \neq c$, as opposite to deterministic learners.

4. The Nondeterministic Graded Approach

We wish to define a hypothesis h for each instance $\boldsymbol{x}$, (Eq. 5), that optimizes a score function defined in terms of the entries a, b, c of the contingency matrix (6). In our case we are trying to optimize the F_1 measure. Therefore, we only need the number of correct grade predictions throughout the list of labels, a, and the length of the intervals.

$$h(\boldsymbol{x}) = \underset{(I_1,\ldots,I_{|L|})}{\operatorname{argmax}} \frac{2\sum_{a=1}^{|L|} a \Pr(a|(I_1,\ldots,I_{|L|}),\boldsymbol{x})}{\sum_{i=1}^{|L|}|I_i| + |L|}. \tag{12}$$

From hereon, we shall assume that we have learned an estimation of the posterior probabilities for each label $l \in L$ and each grade $g \in M$, given $\boldsymbol{x}$:

$$\Pr(l, g|\boldsymbol{x}), \forall l \in L, \forall g \in M. \tag{13}$$

To derive an algorithm to find optimum values for h predictions, let us generalize the case where we only have a single label. That is, $|L| = 1$, trying to optimize the F_1 measure, Alonso et al. (2008), if p is the posterior probability of an interval of grades I, Eq. (12) can be written as

$$h(\boldsymbol{x}) = \underset{I}{\operatorname{argmax}}\left(\frac{2p}{|I|+1}\right). \tag{14}$$

In the next proposition we shall generalize this formula to any number of labels. For this purpose, we need to assume the *independence* of the probabilities of labels.

Proposition 1 (Average number of correct classifications)**.** *If the posterior probabilities of labels are* independent*, the average number of correct classifications for $h(\boldsymbol{x}) = (I_1, \ldots, I_{|L|})$ is the sum of the posterior probabilities of the intervals. In symbols,*

$$\sum_{a=1}^{|L|} a \Pr(a|(I_1,\ldots,I_{|L|}),\boldsymbol{x}) = \sum_{i=1}^{|L|} \Pr(I_i|\boldsymbol{x}).$$

Proof. The proof can be made by induction on the number of labels. For only one label, the thesis of this proposition is trivial; see (14). Then, assuming the proposition proven for r labels, we now prove the equation for $r + 1$.

Since $\boldsymbol{x}$ was fixed, we get rid of it to facilitate the reading of the following formulae.

$$\begin{aligned}
&\sum_{a=1}^{r+1} a \Pr(a|(I_1,\ldots,I_{r+1})) = \\
&\quad= \sum_{a=1}^{r+1} a\Big[(1-\Pr(I_{r+1}))\Pr(a|(I_1,\ldots,I_r)) + \\
&\qquad + \Pr(I_{r+1})\Pr(a-1|(I_1,\ldots,I_r))\Big] = \\
&\quad= (1-\Pr(I_{r+1}))\sum_{a=1}^{r} a\Pr(a|(I_1,\ldots,I_r)) + \\
&\qquad + \Pr(I_{r+1})\sum_{a=1}^{r+1}(a-1)\Pr(a-1|(I_1,\ldots,I_r)) + \\
&\qquad + \Pr(I_{r+1})\sum_{a=1}^{r+1}\Pr(a-1|(I_1,\ldots,I_r)),
\end{aligned}$$

given that $\Pr(r+1|(I_1,\ldots,I_r)) = 0$. Moreover, since

$$\sum_{a=1}^{r+1}\Pr(a-1|(I_1,\ldots,I_r)) = \sum_{a=0}^{r}\Pr(a|(I_1,\ldots,I_r)) = 1,$$

applying the induction hypothesis, we finally have that

$$\begin{aligned}
&\sum_{a=1}^{r+1} a \Pr(a|(I_1,\ldots,I_{r+1})) = \\
&\quad= \Big[(1-Pr(I_{r+1})) + \Pr(I_{r+1})\Big]\sum_{a=1}^{r} a\Pr(a|(I_1,\ldots,I_r)) + \\
&\qquad + \Pr(I_{r+1}) = \sum_{i=1}^{r+1}\Pr(I_i).
\end{aligned}$$

□

Corollary 1 (Defining optimal F_1 predictions)**.** *If the posterior probabilities of labels are* independent*, the prediction for an input $\boldsymbol{x}$ with the maximum expected F_1 score is given by*

$$h(\boldsymbol{x}) = \underset{(I_1,\ldots,I_{|L|})}{\operatorname{argmax}} \frac{2\sum_{i=1}^{|L|}\Pr(I_i|\boldsymbol{x})}{\sum_{i=1}^{|L|}|I_i| + |L|}.$$

4.1. Searching for a Near Optimum

Once we have an estimation of the expected F_1 given an instance $\boldsymbol{x}$ and an output $(I_1, \ldots, I_{|L|})$, we need to search for the best set of intervals. According to Corollary 1, we only need to compute the scores obtained by the best intervals for each possible length. But notice that, for a given length of the prediction, there are many possibilities; moreover, it is not clear how to divide a prediction length between labels. Depending on the distribution of probabilities, the risk of error in labels may be different. This is hence the point at which we explicitly consider the set of labels at the same time. Although we assumed independence between label probabilities, we have to adopt a multi-label point of view when searching for the best combination of intervals.

Thus, given an input $\boldsymbol{x}$, let us first compute the matrix S with one column for each label in L, and one row for each grade in M, defined by

$$S(i, j) = (p^i_j, I^i_j),\ \forall i \in M, \forall j \in L, \tag{15}$$

where I^i_j is the interval of grades, of size i, with the highest posterior probability, p^i_j, for label l_j. These probabilities can be computed by means of a simple loop.

The prediction $h(\boldsymbol{x})$ is a combination of intervals, one from each column of the matrix S. Notice that we do not consider the possibility of abstention in any label: all labels will have a nonempty interval of grades. Therefore, the search space has $|M|^{|L|}$ possible combinations of intervals.

To avoid exponential complexity, we use a greedy breadth-first search. So, let us start assuming that the best combination of intervals is given by the first row of matrix S; i.e., by assuming that, for each label, the best prediction is the interval with just one grade: the one with the highest posterior probability. Then, the algorithm iteratively tries to replace one of the intervals by another with one more grade. To do so, the algorithm computes the highest increase in the sum of probabilities. The algorithm stops when no improvements can be reached after searching the columns of S.

In the worse case, the algorithm considers the optimization of each possible prediction length (from $|L|$ to $|M| \times |L|$). The optimization involves checking $|L|$ possibilities. Therefore, the complexity of our algorithm is

$$O\left(|L|\left(|M| \times |L| - |L|\right)\right) = O\left(|L|^2|M|\right).$$

Despite this search, the algorithm does not guarantee finding the optimum combination. The experiments reported in the next section show that the classifiers achieved using this algorithm outperform the *Binary Relevance* strategy, which would make predictions for each label separately.

5. Experimental Results

In this section we report the results of a set of experiments designed to evaluate the learners proposed in the paper. We compare the nondeterministic learners introduced in the preceding section with a state-of-the-art deterministic learner. After presenting these learners in detail, we describe the datasets used in the comparison discussed in the last subsection.

Table 1. Description of the datasets used in the experiments. Sources: † Cheng et al. (2010); Abele-Brehm and Stief (2004); ⋆ Goldberg et al. (2001)

Dataset	Instances	Attribs.	Range	Labels	Sources
BeLa-E	1930		5		†
10		40		10	
20		30		20	
Jester-1.1	7200	80	5, 10, 20	20	⋆
Jester-1.2	6916	80	5, 10, 20	20	⋆
Jester-2	3091	80	5, 10, 20	20	⋆

Table 2. Average F_1 scores (expressed as percentages) of all learners and average size of the predictions for nondeterministic learners

	F_1			$\lvert h\rvert$	
	$IBLR_{GML}$	BR_{nd}	GML_{nd}	BR_{nd}	GML_{nd}
BeLa-E					
10	49.35	56.22	56.65	1.85	1.64
20	47.99	54.75	55.21	1.90	1.70
Jester-1.1					
5	40.46	46.66	46.82	2.35	2.07
10	22.31	28.95	29.20	3.50	2.52
20	11.71	16.66	16.79	5.27	2.52
Jester-1.2					
5	41.25	46.73	46.86	2.35	2.05
10	22.96	29.13	29.43	3.46	2.48
20	12.10	16.81	16.88	5.19	2.51
Jester-2					
5	48.16	50.08	50.44	2.12	1.84
10	29.58	31.81	31.94	2.90	2.06
20	16.26	18.35	18.17	3.91	2.09

5.1. Learners Compared

As a deterministic learner, we used $IBLR_{GML}$, the graded version of *IBLR-ML* (Cheng and Hüllermeier, 2009) presented by Cheng et al. (2010). We employed the implementation provided by the authors through the library *Mulan*[1] (Tsoumakas et al., 2010, 2011). We wrote an interface using Matlab to ensure that cross-validations were carried out with the same splits of training and testing data.

On the nondeterministic side, we used *LibLinear* (Fan et al., 2008) to estimate posterior probabilities (Wu et al., 2004). We used a *Binary Relevance strategy (*BR_{nd}*)*, with the implementation provided by the authors of Alonso et al. (2008). The learner proposed in Section 4 shall be called GML_{nd}.

5.2. Datasets and Parameter Settings

We used 11 datasets to compare the performance of the different approaches. Table 1 reports the characteristics of these datasets. Their structure is quite similar: they are basically matrices of grades from an ordered set M of integers or real numbers. From an abstract point of view, all datasets can be seen

[1] http://mulan.sourceforge.net/

Table 3. Average *Recall* and *Precision* (expressed as percentages). Notice that for deterministic learners *Recall* and *Precision* are the same as F_1 scores; however, we repeat the values of Table 2 for ease of reference

	Recall			*Precision*	
	$IBLR_{GML}$	BR_{nd}	GML_{nd}	BR_{nd}	GML_{nd}
BeLa-E					
10	49.35	79.61	74.58	43.60	45.94
20	47.99	79.10	74.49	41.89	43.98
Jester-1.1					
5	40.46	78.00	71.99	33.40	34.85
10	22.31	65.08	51.81	18.69	20.54
20	11.71	52.45	29.88	9.98	11.87
Jester-1.2					
5	41.25	77.96	71.49	33.48	35.05
10	22.96	64.80	51.64	18.88	20.80
20	12.10	52.14	30.05	10.11	11.94
Jester-2					
5	48.16	77.13	70.95	37.52	39.64
10	29.58	61.17	48.75	21.84	24.26
20	16.26	44.95	28.64	11.79	13.76

as records from a recommender system. The rows gather the assessments of people regarding different items represented by the columns.

The first 2 datasets were built from BeLa-E (Cheng et al., 2010; Abele-Brehm and Stief, 2004) presented in Section 2. We built a matrix whose rows record the data for each student: sex, age, and the answers to the 48 questions about the degree of importance of properties of future jobs. From this matrix, 2 different datasets were generated following the scheme used in Cheng et al. (2010). In BeLa-E-10, we randomly selected 10 (respectively 20 in BeLa-E-20) columns from the set of 48 students' answers as the set of class labels, while all the remaining columns, including sex and age, were taken as predictive features.

The other datasets used were compiled from *Jester*, an online joke recommender system[2] (Goldberg et al., 2001). There are 3 different datasets, Jester-1.1, Jester-1.2, and Jester-2. The first two, Jester-1.*, collect anonymous continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users, collected between April 1999 and May 2003. Jester-1.1 (respectively Jester-1.2) gathers data from 24,983 (respectively 23,500) users who have rated 36 or more jokes. To avoid missing values, in both cases we considered the subset of users who have rated the whole collection of 100 jokes.

In the case of Jester-2, there are 150 jokes rated by 63,974 users, collected between November 2006 and May 2009. We selected the 100 jokes with the highest number of ratings, and then the users who have rated all of them.

From the resulting matrices, in all the Jester datasets we randomly separated subsets of 20 columns as class labels, while the remaining 80 columns were taken as predictive features. The continuous ratings of label columns were discretized in scales of 20, 10 and 5 values using a simple *equal length* procedure.

In all datasets, we used *LibLinear* to estimate the posterior probabilities needed by *nd* classifiers, with the default behavior of the learner as a logistic regressor. An *internal* grid search adjusted the C parameter selection from $\{10^i : i = -3, -2, -1, 0, 1\}$ using a 2-fold cross-validation repeated 3 times.

5.3. Comparisons

Following the experimental method of Cheng et al. (2010), each learner was evaluated on each dataset estimating different scores using a 10-fold cross-validation. These estimations were then averaged over a total number of 25 randomly generated datasets to avoid the influence of random splits in labels and predictive features.

Since graded multi-label classification can be seen from different points of view, we made different comparisons. First we compared the F_1 scores of deterministic and nondeterministic learners, since optimizing this measure was the aim of our proposal. To contrast deterministic and nondeterministic learners, we attached the average size of predictions to F_1 scores for nondeterministic learners. Table 2 shows these scores.

The nondeterministic multi-label GML_{nd} outperforms the other options in F_1. Moreover, the differences are significant. To compare the performance of the 3 learners considered, following García and Herrera (2008), we performed a Bergmann-Hommel procedure using the software provided in the paper. GML_{nd} is the best learner in all cases except on one occasion, BR_{nd} is the second best, while the deterministic $IBLR_{GML}$ comes third. The differences between every pair of learners are significant with $p < 0.02$.

We used a Wilcoxon two-sided signed rank test to compare the two nondeterministic options in all cases. The differences in F_1 between nondeterministic learners are slight, though systematic and significant ($p < 0.02$). These results provide statistical support to the claim that the optimization strategy of BR_{nd} is suboptimal with respect to that of GML_{nd}. On the other hand, the differences in the average size of predictions are bigger and significant with $p < 0.001$. In this case, BR_{nd} always predicts more grades than the multi-label option, GML_{nd}.

Note that the highest differences appear in datasets where the level of successful predictions are the lowest, in the Jester dataset with 20 degree options. The quality of posterior probabilities is lower in these datasets than in the others. In these cases, BR_{nd} tries to improve the performance by spending more predictions in each label. On the other hand, the multi-label approach somehow discovers that it is possible from a general perspective, using 2 or 3 predictions less, to achieve a similar or better F_1 performance. In fact, Jester-2 with 20 grades is the only dataset in which BR_{nd} achieves better F_1 than the multi-label GML_{nd}.

These results mean that the multi-label strategy is able to distribute the number of predictions between the labels better than BR_{nd}. The global point of view of multi-label outperforms the marginal perspective adopted by the binary relevance learner.

[2] Available at http://eigentaste.berkeley.edu/dataset/

To complete the information retrieval point of view, Table 3 shows the scores achieved in *Recall* and *Precision*. Remember that for deterministic learners *Recall* and *Precision* are the same as F_1 scores. The *Recall* in BR_{nd} is higher than in GML_{nd}, though in *Precision* the results are the opposite. The reason is that BR_{nd} needs more grades than GML_{nd}, therefore the right grade is more often included in its predictions (*Recall*), but the *density* of correct predictions (*Precision*) is lower. In both cases, the differences are significant (using a Wilcoxon two-sided signed rank test) with $p < 0.001$. Yet again the highest differences appear in datasets with 20 degrees.

In both nondeterministic classifiers, the scores in *Precision* are generally lower than those obtained by the deterministic $IBLR_{GML}$. This is a typical side effect of nondeterminism; see del Coz et al. (2009). To improve F_1 scores, *nd* classifiers increase the size of predictions, which worsens *Precision* scores.

6. Conclusions

We have presented graded multi-label hypotheses (Cheng et al., 2010) as a set of ordinal classifications. This allows us to consider recommender systems as a straightforward application field. Furthermore, we have introduced nondeterministic classifiers in this context.

For each instance, the learner proposed in this paper, GML_{nd}, needs the estimations of the posterior probabilities of each grade and label to compute the prediction with the best expected F_1. Since the search for the optimum has a huge search space, we propose a greedy algorithm that returns a near-optimum set of predictions. The complexity is $O(|L|^2 \cdot |M|)$, where $|L|$ is the number of labels and $|M|$ the number of grades.

The complexity is acceptable for a small number of labels, such as those used in the experiments reported in the previous section. If we had very large sets of labels, we could cluster them into small subsets using some similarity measure between labels.

The paper includes an experimental comparison with another nondeterministic (binary relevance) alternative and a deterministic state-of-the-art learner for GMLC tasks, $IBLR_{GML}$. The result is the consequence of a formal proof that establishes that the best option for a given amount of predictions is the one with highest sum of probabilities among all labels.

The role of nondeterministic learners can be illustrated noting that GML_{nd}, predicting around 2 grades on average, generally succeeds many more times than $IBLR_{GML}$, which only predicts one grade. The difference is quite important, around 25 percentage points on average; see the *Recall* scores in Table 3. Since an interval of size 2 is often a good approximation to a degree of membership, the improvement may be noteworthy in most practical applications.

The approach presented in this paper is related to a couple of papers previously published by our research group (del Coz et al., 2009; Quevedo et al., 2012). The proposal put forward in del Coz et al. (2009) is a method for extending multiclass classification to allow predictions with more than one class: nondeterministic classifiers. The contribution of Quevedo et al. (2012) is a method to learn multi-label using a thresholding strategy. The algorithm presented in this paper uses the idea of del Coz et al. (2009), extending it with new results to a more difficult setting: multi-label classification. It is additionally based on some ideas from Quevedo et al. (2012), although the extension to a new setting, graded multi-label classification, allows completely new results that have no sense if there are any graduation of the membership of labels. These new results comprise Propositions 1 and 2 and the algorithm described in Section 4.1.

Acknowledgments

The research reported here is supported in part under grant TIN2011-23558 from the Ministerio de Economía y Competitividad, Spain. We would also like to acknowledge Eyke Hüllermeier, who generously shared with us the dataset BeLa-E and the code of *IBLR*, and Grigorios Tsoumakas for making the Mulan library available.

References

Abele-Brehm, A., Stief, M., 2004. Die Prognose des Berufserfolgs von Hochschulabsolventinnen und-absolventen. Zeitschrift für Arbeits-und Organisationspsychologie A & O 48, 4–16.

Agarwal, A., Davis, J., Ward, T., 2001. Supporting ordinal four-state classification decisions using neural networks. Information Technology and Management 2, 5–26.

Alonso, J., del Coz, J.J., Díez, J., Luaces, O., Bahamonde, A., 2008. Learning to predict one or more ranks in ordinal regression tasks, Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), Springer. pp. 39–54.

Cardoso, J., da Costa, J., 2007. Learning to Classify Ordinal Data: The Data Replication Method. Journal of Machine Learning Research 8, 1393–1429.

Cheng, W., Dembczyński, K., Hüllermeier, E., 2010. Graded Multilabel Classification: The Ordinal Case, Proceedings of the 27^{th} International Conference on Machine Learning, (ICML), pp. 223–230.

Cheng, W., Hüllermeier, E., 2009. Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. Machine Learning 76, 211–225.

del Coz, J.J., Bayón, G.F., Díez, J., Luaces, O., Bahamonde, A., Sañudo, C., 2005. Trait selection for assessing beef meat quality using non-linear SVM, Advances in Neural Information Processing Systems 17 (NIPS '04), MIT Press, Cambridge, MA. pp. 321–328.

del Coz, J.J., Díez, J., Bahamonde, A., 2009. Learning nondeterministic classifiers. Journal of Machine Learning Research 10, 2273–2293.

Elisseeff, A., Weston, J., 2001. A kernel method for multi-labelled classification, Advances in Neural Information Processing Systems 14, MIT Press. pp. 681–687.

Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C., 2008. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9, 1871–1874.

García, S., Herrera, F., 2008. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694.

Goldberg, K., Roeder, T., Gupta, D., Perkins, C., 2001. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval 4, 133–151.

Joachims, T., 2006. Training linear SVMs in linear time, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM.

Lin, C.J., Weng, R.C., Keerthi, S.S., 2008. Trust Region Newton Method for Logistic Regression. Journal of Machine Learning Research 9, 627–650.

Luaces, O., Rodrigues, L., Meira, C., Bahamonde, A., 2011. Using nondeterministic learners to alert on coffee rust disease. Expert Systems with Applications 38, 14276–14283.

Montañés, E., Quevedo, J.R., del Coz, J.J., 2011. Aggregating independent and dependent models to learn multi-label classifiers, Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), pp. 484–500.

Quevedo, J.R., Luaces, O., Bahamonde, A., 2012. Multilabel classifiers with a probabilistic thresholding strategy. Pattern Recognition 45, 876–883.

Read, J., Pfahringer, B., Holmes, G., Frank, E., 2009. Classifier Chains for Multi-label Classification, Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), pp. 254–269.

Schapire, R., Singer, Y., 2000. Boostexter: A boosting-based system for text categorization. Machine learning 39, 135–168.

Shen, L., Joshi, A., 2005. Ranking and Reranking with Perceptron. Machine Learning 60, 73–96.

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

Tsoumakas, G., Katakis, I., Vlahavas, I., 2010. Mining Multilabel Data. In O. Maimon and L. Rokach (Ed.), Data Mining and Knowledge Discovery Handbook, Springer , 667–685.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I., 2011. Mulan: A Java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414.

Wu, T.F., Lin, C.J., Weng, R.C., 2004. Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research 5, 975–1005.

Yu, S., Yu, K., Tresp, V., Kriegel, H., 2006. Collaborative ordinal regression, Proceedings of the 23^{rd} International Conference on Machine Learning (ICML'06), pp. 1089–1096.

Zhang, M., Zhou, Z., 2007. ML-KNN: A Lazy Learning Approach to Multi-label Learning. Pattern Recognition 40, 2038–2048.