Cautious Label-Wise Ranking with Constraint Satisfaction

Yonatan-Carlos Carranza-Alarcon(✉), Soundouss Messoudi(✉), and Sébastien Destercke(✉)

HEUDIASYC - UMR CNRS 7253, Université de Technologie de Compiègne,
57 avenue de Landshut, 60203 Compiegne Cedex, France
{yonatan-carlos.carranza-alarcon,soundouss.messoudi,
sebastien.destercke}@hds.utc.fr
https://www.hds.utc.fr/

Abstract. Ranking problems are difficult to solve due to their combinatorial nature. One way to solve this issue is to adopt a decomposition scheme, splitting the initial difficult problem in many simpler problems. The predictions obtained from these simplified settings must then be combined into one single output, possibly resolving inconsistencies between the outputs. In this paper, we consider such an approach for the label ranking problem, where in addition we allow the predictive model to produce cautious inferences in the form of sets of rankings when it lacks information to produce reliable, precise predictions. More specifically, we propose to combine a rank-wise decomposition, in which every sub-problem becomes an ordinal classification one, with a constraint satisfaction problem (CSP) approach to verify the consistency of the predictions. Our experimental results indicate that our approach produces predictions with appropriately balanced reliability and precision, while remaining competitive with classical, precise approaches.

Keywords: Label ranking problem · Constraint satisfaction · Imprecise probabilities

1 Introduction

In recent years, machine learning problems with structured outputs received an increasing interest. These problems appear in a variety of fields, including biology [33], image analysis [23], natural language treatment [5], and so on.

In this paper, we look at *label ranking (LR)*, where one has to learn a mapping from instances to rankings (strict total order) defined over a finite, usually limited number of labels. Most solutions to this problem reduce its initial complexity, either by fitting a probabilistic model (Mallows, Plackett-Luce [7]) with few parameters, or through a decomposition scheme. For example, ranking by pairwise comparison (RPC) [24] transforms the initial problem into binary problems. Constraint classification and log-linear models [13], as well as SVM-based

M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1238, pp. 96–111, 2020.
https://doi.org/10.1007/978-3-030-50143-3_8

methods [30] learn, for each label, a (linear) utility function from which the ranking is deduced. Those latter approaches are close to other proposals [18] that perform a label-wise decomposition.

In ranking problems, it may also be interesting [9,18] to predict partial rather than complete rankings, abstaining to make a precise prediction in presence of too little information. Such predictions can be seen as extensions of the reject option [4] or of partial predictions [11]. They can prevent harmful decisions based on incorrect predictions, and have been applied for different decomposition schemes, be it pairwise [10] or label-wise [18], always producing cautious predictions in the form of partial order relations.

In this paper, we propose a new label ranking method, called LR-CSP, based on a label-wise decomposition where each sub-problem intends to predict a set of ranks. More precisely, we propose to learn for each label an imprecise ordinal regression model of its rank [19], and use these models to infer a set of possible ranks. To do this, we use imprecise probabilistic (IP) approaches are well tailored to make partial predictions [11] and represent potential lack of knowledge, by describing our uncertainty by means of a convex set of probability distributions $\mathscr{P}$ [31] rather than by a classical single precise probability distribution $\mathbb{P}$. An interesting point of our method, whose principle can be used with any set of probabilities, is that it does not require any modification of the underlying learning *imprecise* classifier, as long as the classifier can produce lower and upper bounds $[\underline{P}, \overline{P}]$ over binary classification problems.

We then use CSP techniques on the set of resulting predictions to check whether the prediction outputs are consistent with a global ranking (i.e. that each label can be assigned a different rank).

Section 2 introduces the problem and our notations. Section 3 shows how ranks can be predicted from imprecise probabilistic models and presents the proposed inference method based on robust optimization techniques. Section 4 discusses related work. Finally, Sect. 5 is devoted to experimental evaluation showing that our approach does reach a higher accuracy by allowing for partial outputs, and remains quite competitive with alternative approaches to the same learning problem.

2 Problem Setting

Multi-class problems consist in associating an instance $\mathbf{x}$ coming from an input space $\mathcal{X}$ to a single label of the output space $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ representing the possible classes. In label ranking, an instance $\mathbf{x}$ is no longer associated to a unique label of Λ but to an order relation[1] $\succ_{\mathbf{x}}$ over $\Lambda \times \Lambda$, or equivalently to a complete ranking over the labels in Λ. Hence, the output space is the set $\mathcal{L}(\Lambda)$ of complete rankings of Λ that contains $|\mathcal{L}(\Lambda)| = k!$ elements (i.e., the set of all permutations). Table 1 illustrates a label ranking data set example with $k = 3$.

[1] A complete, transitive, and asymmetric relation.

Table 1. An example of label ranking data set $\mathbb{D}$

X_1	X_2	X_3	X_4	Y
107.1	25	Blue	60	$\lambda_1 \succ \lambda_3 \succ \lambda_2$
−50	10	Red	40	$\lambda_2 \succ \lambda_3 \succ \lambda_1$
200.6	30	Blue	58	$\lambda_2 \succ \lambda_1 \succ \lambda_3$
107.1	5	Green	33	$\lambda_1 \succ \lambda_2 \succ \lambda_3$
...	...	...	...	...

We can identify a ranking $\succ_{\mathbf{x}}$ with a permutation $\sigma_{\mathbf{x}}$ on $\{1, \dots, k\}$ such that $\sigma_{\mathbf{x}}(i) < \sigma_{\mathbf{x}}(j)$ iff $\lambda_i \succ_{\mathbf{x}} \lambda_j$, as they are in one-to-one correspondence. $\sigma_{\mathbf{x}}(i)$ is the *rank* of label i in the order relation $\succ_{\mathbf{x}}$. As there is a one-to-one correspondence between permutations and complete rankings, we use the terms interchangeably.

Example 1. Consider the set $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ and the observation $\lambda_3 \succ \lambda_1 \succ \lambda_2$, then we have $\sigma_{\mathbf{x}}(1) = 2,\ \sigma_{\mathbf{x}}(2) = 3,\ \sigma_{\mathbf{x}}(3) = 1$.

The usual objective in label ranking is to use the training instances $\mathbb{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$ with $x_i \in \mathcal{X}$, $y_i \in \mathcal{L}(\Lambda)$ to learn a predictor, or a *ranker* $h : \mathcal{X} \rightarrow \mathcal{L}(\Lambda)$. While in theory this problem can be transformed into a multi-class problem where each ranking is a separate class, this is in practice undoable, as the number of classes would increase factorially with k. The most usual means to solve this issue is either to decompose the problem into many simpler ones, or to fit a parametric probability distribution over the ranks [7]. In this paper, we shall focus on a label-wise decomposition of the problem.

This rapid increase of $|\mathcal{L}(\Lambda)|$ also means that getting reliable, precise predictions of ranks is in practice very difficult as k increases. Hence it may be useful to allow the ranker to return partial but reliable predictions.

3 Label-Wise Decomposition: Learning and Predicting

This section details how we propose to reduce the initial ranking problem in a set of k label-wise problems, that we can then solve separately. The idea is the following: since a complete observation corresponds to each label being associated to a unique rank, we can learn a probabilistic model $p_i : K \rightarrow [0, 1]$ with $K = \{1, 2, \dots, k\}$ and where $p_{ij} := p_i(j)$ is interpreted as the probability $P(\sigma(i) = j)$ that label λ_i has rank j. Note that $\sum_j p_{ij} = 1$.

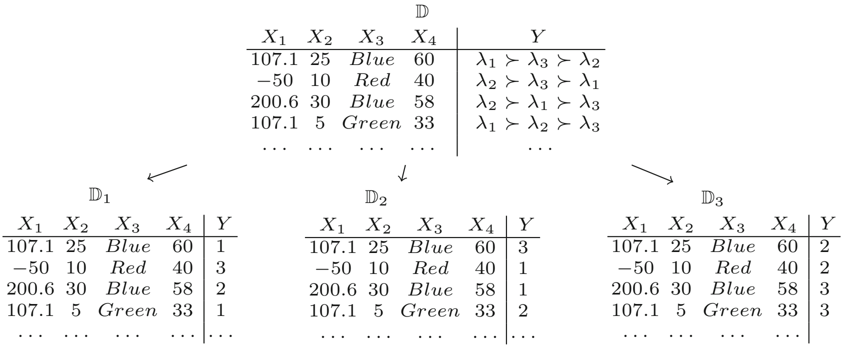
A first step is to decompose the original data set $\mathbb{D}$ into k data sets $\mathbb{D}_j = \{(\mathbf{x}_i, \sigma_{\mathbf{x}_i}(j)) \mid i = 1, \dots, n\},\ j = 1, \dots, k$. The decomposition is illustrated by Fig. 1. Estimating the probabilities p_{ij} for a label λ_i then comes down to solve an ordinal regression problem [27]. In such problems, the rank associated to a label is the one minimizing the expected cost $\mathbb{E}_{ij}$ of assigning label λ_i to rank j, that depends on p_{ij} and a distance $D : K \times K \rightarrow \mathbb{R}$ between ranks as follows:

$$\mathbb{E}_{ij} = \sum\nolimits_{\ell=1}^{k} D(j, k) p_{ik}. \tag{1}$$

Common choices for the distances are the L_1 and L_2 norms, corresponding to

$$D_1(j,k) = |j-k| \quad \text{and} \quad D_2(j,k) = (j-k)^2. \tag{2}$$

Other choices include for instance the pinball loss [29], that penalizes asymmetrically giving a higher or a lower rank than the actual one. An interest of those in the imprecise setting we will adopt next is that it produces predictions in the form of intervals, i.e., in the sense that $\{1,3\}$ cannot be a prediction but $\{1,2,3\}$ can. In this paper, we will focus on the L_1 loss, as it is the most commonly considered in ordinal classification problems[2].



$\mathbb{D}$

X_1	X_2	X_3	X_4	Y
107.1	25	*Blue*	60	$\lambda_1 \succ \lambda_3 \succ \lambda_2$
−50	10	*Red*	40	$\lambda_2 \succ \lambda_3 \succ \lambda_1$
200.6	30	*Blue*	58	$\lambda_2 \succ \lambda_1 \succ \lambda_3$
107.1	5	*Green*	33	$\lambda_1 \succ \lambda_2 \succ \lambda_3$
...	...	...	...	...

$\mathbb{D}_1$

X_1	X_2	X_3	X_4	Y
107.1	25	*Blue*	60	1
−50	10	*Red*	40	3
200.6	30	*Blue*	58	2
107.1	5	*Green*	33	1
...	...	...	...	...

$\mathbb{D}_2$

X_1	X_2	X_3	X_4	Y
107.1	25	*Blue*	60	3
−50	10	*Red*	40	1
200.6	30	*Blue*	58	1
107.1	5	*Green*	33	2
...	...	...	...	...

$\mathbb{D}_3$

X_1	X_2	X_3	X_4	Y
107.1	25	*Blue*	60	2
−50	10	*Red*	40	2
200.6	30	*Blue*	58	3
107.1	5	*Green*	33	3
...	...	...	...	

Fig. 1. Label-wise decomposition of rankings

3.1 Probability Set Model

Precise estimates for p_i issued from the finite data set $\mathbb{D}_k$ may be unreliable, especially if these estimates rely on little, noisy or incomplete data. Rather than relying on precise estimates in all cases, we propose to consider an imprecise probabilistic model, that is, to consider for each label λ_i a polytope (a convex set) $\mathscr{P}_i$ of possible probabilities. In our setting, a particularly interesting model are imprecise cumulative distributions [15], as they naturally encode the ordinal nature of rankings, and are a common choice in the precise setting [22]. They consist in providing bounds $\left[\underline{P}(A_\ell), \overline{P}(A_\ell)\right]$ on events $A_\ell = \{1, \ldots, \ell\}$ and to consider the resulting set

$$\mathscr{P}_i = \left\{ p_i : \underline{P}_i(A_\ell) \leq \sum\nolimits_{j=1}^{\ell} p_{ij} \leq \overline{P}_i(A_\ell), \sum\nolimits_{j \in K} p_{ij} = 1 \right\}. \tag{3}$$

We will denote by $\underline{F}_{ij} = \underline{P}_i(A_j)$ and $\overline{F}_{ij} = \overline{P}_i(A_j)$ the given bounds. Table 2 provides an example of a cumulative distribution that could be obtained in a ranking problem where $k = 5$ and for a label λ_i. For other kinds of sets $\mathscr{P}_i$ we could consider, see [17].

[2] The approach easily adapts to the other losses.

Table 2. Imprecise cumulative distribution for λ_i

Rank j	1	2	3	4	5
$\overline{F}_{ij}$	0.15	**0.55**	*0.70*	***0.95***	1
$\underline{F}_{ij}$	**0.10**	*0.30*	***0.45***	0.80	1

This approach requires to learn k different models, one for each label. This is to be compared with the RPC [24] approach, in which $^{k(k-1)}/_2$ models (one for each pair of labels) have to be learned. There is therefore a clear computational advantage for the current approach when k increases. It should also be noted that the two approaches rely on different models: while the label-wise decomposition uses learning methods issued from ordinal regression problems, the RPC approach usually uses learning methods issued from binary classification.

3.2 Rank-Wise Inferences

The classical means to compare two ranks as possible predictions, given the probability p_i, is to say that rank ℓ is preferable to rank m (denoted $\ell \succ m$) iff

$$\sum_{j=1}^{k} D_1(j,m)p_{ij} \geq \sum_{j=1}^{k} D_1(j,\ell)p_{ij} \tag{4}$$

That is if the expected cost (loss) of predicting m is higher than the expected cost of predicting ℓ. The final prediction is then the rank that is not dominated or preferred to any other (with typically a random choice when there is some indifference between the top ranks).

When precise probabilities p_i are replaced by probability sets $\mathscr{P}_i$, a classical extension[3] of this rule is to consider that rank ℓ is preferable to rank m iff it is so for every probability in $\mathscr{P}_i$, that is if

$$\inf_{p_i \in \mathscr{P}_i} \sum_{j=1}^{k} (D_1(j,m) - D_1(j,\ell))p_{ij} \tag{5}$$

is positive. Note that under this definition we may have simultaneously $m \not\succ \ell$ and $\ell \not\succ m$, therefore there may be multiple undominated, incomparable ranks, in which case the final prediction is a set-valued one.

In general, obtaining the set of predicted values requires to solve Eq. (5) at most a quadratic number of times (corresponding to each pairwise comparison). However, it has been shown [16, Prop. 1] that when considering D_1 as a cost function, the set of predicted values corresponds to the set of possible medians within $\mathscr{P}_i$, which is straightforward to compute if one uses the generalized p-box [15] as an uncertainty model. Namely, if $\underline{F}_i, \overline{F}_i$ are the cumulative distributions for label λ_i, then the predicted ranks under D_1 cost are

$$\hat{R}_i = \left\{ j \in K : \underline{F}_{i(j-1)} \leq 0.5 \leq \overline{F}_{ij}, \quad \underline{F}_{i(0)} = 0 \right\}, \tag{6}$$

[3] Also, known as maximality criterion [31].

a set that is always non-empty and straightforward to obtain. Looking back at Table 2, our prediction would have been $\hat{R} = \{2, 3, 4\}$, as these are the three possible median values.

As for the RPC approach (and its cautious versions [9]), the label-wise decomposition requires to aggregate all decomposed models into a single (partial) prediction. Indeed, focusing only on decomposed models $\mathscr{P}_i$, nothing forbids to predict the same rank for multiple labels. In the next section, we discuss cautious predictions in the form of sets of ranks, as well as how to resolve inconsistencies.

3.3 Global Inferences

Once we have retrieved the different set-valued predictions of ranks for each label, two important questions remain:

1. Are those predictions consistent with the constraint that each label should receive a distinct rank?
2. If so, can we reduce the obtained predictions by integrating the aforementioned constraint?

Example 2. To illustrate the issue, let us consider the case where we have four labels $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. Then the following predictions

$$\hat{R}_1 = \{1, 2\},\ \hat{R}_2 = \{1, 2\},\ \hat{R}_3 = \{1, 2\},\ \hat{R}_4 = \{3, 4\}$$

are inconsistent, simply because labels $\lambda_1, \lambda_2, \lambda_3$ cannot be given simultaneously a different rank (note that pair-wisely, they are not conflicting). On the contrary, the following predictions

$$\hat{R}_1 = \{1, 2\},\ \hat{R}_2 = \{1, 2, 3\},\ \hat{R}_3 = \{2\},\ \hat{R}_4 = \{1, 2, 3, 4\}$$

are consistent, and could also be reduced to the unique ranking

$$\hat{R}'_1 = \{1\},\ \hat{R}'_2 = \{3\},\ \hat{R}'_3 = \{2\},\ \hat{R}'_4 = \{4\},$$

as the strong constraint $\hat{R}_3 = \{2\}$ propagates to all other predictions by removing λ_2 from them, which results in a new strong constraint $\hat{R}^*_1 = \{1\}$ that also propagates to all other predictions. This redundancy elimination is repeated as new strong constraints emerge until we get the unique ranking above.

Such a problem is well known in Constraint Programming [12], where it corresponds to the *alldifferent* constraint. In the case where all rank predictions are intervals, that is a prediction $\hat{R}_i$ contains all values between $\min \hat{R}_i$ and $\max \hat{R}_i$, efficient algorithms using the fact that one can concentrate on bounds alone exist, that we can use to speed up computations [28].

4 Discussion of Related Approaches

As said in the introduction, one of our main goals in this paper is to introduce a label ranking method that allows the ranker to partially abstain when it has insufficient information, therefore producing a corresponding set of possible rankings. We discuss here the usefulness of such rank-wise partial prediction (mainly w.r.t. approaches producing partial orders), as well as some related works.

4.1 Partial Orders vs Imprecise Ranks

Most existing methods [9,10] that propose to make set-valued or cautious predictions in ranking problems consider partial orders as their final predictions, that is pairwise relations $\succ_{\mathbf{x}}$ that are transitive and asymmetric, but no longer necessarily complete. To do so, they often rely on decomposition approaches estimating preferences between each pairs of labels [24].

However, while a complete order can be equivalently described by the relation $\succ_{\mathbf{x}}$ or by the rank associated to each label, this is no longer true when one considers partial predictions. Indeed, consider for instance the case where the set of rankings over three labels $\{\lambda_1, \lambda_2, \lambda_3\}$ we would like to predict is $S = \{\lambda_1 \succ \lambda_2 \succ \lambda_3, \lambda_1 \prec \lambda_2 \prec \lambda_3\}$, which could correspond to an instance where λ_2 is a good compromise, and where the population is quite divided about λ_1 and λ_3 that represent more extreme options.

While the set S can be efficiently and exactly represented by providing sets of ranks for each item, none of the information it contains can be retained in a partial order. Indeed, the prediction $\hat{R}_1 = \{1, 3\}, \hat{R}_2 = \{2\}, \hat{R}_3 = \{1, 3\}$ perfectly represents S, while representing it by a partial order would result in the empty relation (since for all pairs i, j, we have $\lambda_i \succ \lambda_j$ and $\lambda_j \succ \lambda_i$ in the set S).

We could find an example that would disadvantage a rank-wise cautious prediction over one using partial orders, as one representation is not more general than the other[4]. Yet, our small example shows that considering both approaches makes sense, as one cannot encapsulate the other, and vice-versa.

4.2 Score-Based Approaches

In a recent literature survey [30], we can see that there are many score-based approaches, already been studied and compared in [24], such as constraint classification, log-linear models, etc. Such approaches learn, from the samples, a function h_j for each label λ_j that will predict a strength $h_j(\boldsymbol{x}^*)$ for a new instance. Labels are then ranked accordingly to their predicted strengths.

We will consider a typical example of such approaches, based on SVM, that we will call SVM label ranking (SVM-LR). Vembu and Gärtner [30] show that the SVM method [20] solving multi-label problems can be straightforwardly generalized to a label ranking problem. In contrast to our approach where each model is learned separately, SVM-LR fits all the functions at once, even if at prediction time they are evaluated independently. While this may account for label dependencies, this comes at a computational cost since we have to solve a quadratic optimization problem (i.e. the dual problem introduced in [20]) whose scale increases rapidly as the number of training samples and labels grows.

More precisely, the score functions $h_j(\boldsymbol{x}^*) = \langle \boldsymbol{w}_j \mid \boldsymbol{x}^* \rangle$ are scalar products between a weight vector $\boldsymbol{w}_j$ and $\boldsymbol{x}^*$. If α_{ijq} are coefficients that represent the

[4] In the sense that the family of subsets of ranking representable by one is not included in the other.

existence of either the preference $\lambda_q \succ_{\boldsymbol{x}_i} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}_i} \lambda_q$ of the instance $\boldsymbol{x}_i$, $\boldsymbol{w}_j$ can be obtained from the dual problem in [20, Sect. 5] as follows:

$$w_j = \frac{1}{2}\sum_{i=1}^{n}\left[\sum_{(j,q)\in E_i}\alpha_{ijq} - \sum_{(p,j)\in E_i}\alpha_{ipj}\right]\boldsymbol{x}_i \tag{7}$$

where α_{ipq} are the weighted target values to optimize into the dual problem. E_i contains all preferences, i.e. $\{(p,q)\in E_i \iff \lambda_p \succ \lambda_q\}$, of the training instance $\boldsymbol{x}_i$.

It may seem at first that such approaches, once made imprecise, could be closer to ours. Indeed, the obtained models h_i after training also provide label-wise information. However, if we were to turn these method imprecise and obtain imprecise scores $[\underline{h}_i, \overline{h}_i]$, the most natural way to build a partial prediction would be to consider that $\lambda_i \succ \lambda_j$ when $\underline{h}_i > \overline{h}_j$, that is when the score of λ_i would certainly be higher than the one of λ_j. Such a partial prediction would be an interval order and would again not encompass the same family of subsets of rankings, as it would constitute a restricted setting compared to the one allowing for prediction any partial order.

5 Experiments

This section describes our experiments made to test if our approach is (1) competitive with existing ones and if (2) the partial predictions indeed provide more reliable inferences by abstaining on badly predicted ranks.

5.1 Data Sets

The data sets used in the experiments come from the UCI machine learning repository [21] and the Statlog collection [25]. They are synthetic label ranking data sets built either from classification or regression problems. From each original data set, a transformed data set $(\mathbf{x}_i, y_i)$ with complete rankings was obtained by following the procedure described in [8]. A summary of the data sets used in the experiments is given in Table 3. We perform 10 × 10-fold cross-validation procedure on all the data sets (c.f. Table 3).

5.2 Completeness/Correctness Trade-Off

To answer the question whether our method correctly identifies on which label it is desirable to abstain or to deliver a set of possible rankings, it is necessary to measure two aspects: how accurate and how precise the predictions are. Indeed, a good balance should be sought between informativeness and reliability of the predictions. For this reason, and similarly to what was proposed in the pairwise setting [9], we use a completeness and a correctness measure to assess the quality of the predictions. Given the prediction $\hat{R} = \{\hat{R}_i,\ i = 1,\dots,k\}$, we propose as the completeness (CP) and correctness (CR) measure

$$CP(\hat{R}) = \frac{k^2 - \sum_{i=1}^{k}|\hat{R}_i|}{k^2 - k} \quad \text{and} \quad CR(\hat{R}) = 1 - \frac{\sum_{i=1}^{k}\min_{\hat{r}_i\in\hat{R}_i}|\hat{r}_i - r_i|}{0.5k^2} \tag{8}$$

Table 3. Experimental data sets

#	Data set	Type	#Inst	#Attributes	#Labels
a	authorship	classification	841	70	4
b	bodyfat	regression	252	7	7
c	calhousing	regression	20640	4	4
d	cpu-small	regression	8192	6	5
e	fried	regression	40768	9	5
f	glass	classification	214	9	6
g	housing	regression	506	6	6
h	iris	classification	150	4	3
i	pendigits	classification	10992	16	10
j	segment	classification	2310	18	7
k	stock	regression	950	5	5
l	vehicle	classification	846	18	4
m	vowel	classification	528	10	11
n	wine	classification	178	13	3

where CP is null if all $\hat{R}_i$ contains the k possible ranks and has value one if all $\hat{R}_i$ are reduced to singletons, whilst CR is equivalent to the Spearman Footrule when having a precise observation. Note that classical evaluation measures [36] used in an IP setting cannot be straightforwardly applied here, as they only extend the 0/1 loss and are not consistent with Spearman Footrule, and adapting cost-sensitive extensions [34] to the ranking setting would require some development.

5.3 Our Approach

As mentioned in Sect. 3, our proposal is to fit an *imprecise* ordinal regression model for every label-wise decomposition $\mathbb{D}_i$, in which the lower and upper bounds of the cumulative distribution $[\underline{F}_i, \overline{F}_i]$ must be estimated in order to predict the set of rankings (Eq. 6) of an unlabeled instance $\boldsymbol{x}^*$. In that regard, we propose to use an extension of Frank and Hall [22] method to imprecise probabilities, already studied in detail in [19].

Frank and Hall's method takes advantage of k ordered label values by transforming the original k-label ordinal problem to $k-1$ binary classification sub-problems. Each estimates of the probability[5] $P_i(A_\ell) := F_i(\ell)$ where $A_\ell = \{1, \ldots, \ell\} \subseteq K$ and the mapping $F_i : K \to [0, 1]$ can be seen as a discrete cumulative distribution. We simply propose to make these estimates imprecise and to use bounds

[5] For readability, we here drop the condition of a new instance in all probabilities, i.e. $P_i(A_\ell) := P_i(A_\ell|\boldsymbol{x}^*)$.

$$\underline{P}_i(A_j) := \underline{F}_i(j) \quad \text{and} \quad \overline{P}_i(A_j) := \overline{F}_i(j)$$

which is indeed a generalized p-box model [15], as defined in Eq. (3).

To estimate these bounds, we use the naive credal classifier (NCC)[6] [35], which extends the classical naive Bayes classifier (NBC), as a base classifier. This classifier imprecision level is controlled through a hyper-parameter $s \in \mathbb{R}$. Indeed, the higher s, the wider the intervals $[\underline{P}_i(A_j), \overline{P}_i(A_j)]$. For $s = 0$, we retrieve the classical NBC with precise predictions, and for $s >>> 0$, the NCC model will make vacuous predictions (i.e. all rankings for every label).

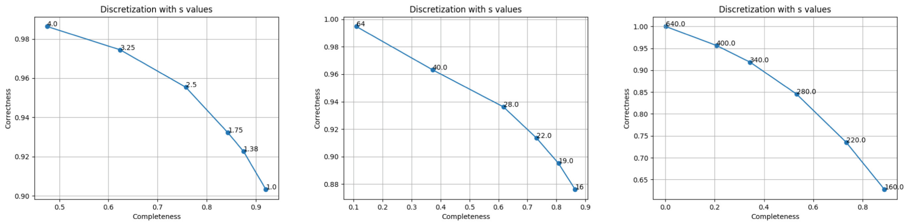
However, the imprecision induced by a peculiar value of s differs from a data set to another (as show the values in Fig. 2), and it is essential to have an adaptive way to quickly obtain two values:

- the value $s_{\min}$ corresponding to the value with an average completeness close to 1, making the corresponding classifier close to a precise one. This value is the one we will use to compare our approach to standard, precise ones;
- the value $s_{\max}$ corresponding to the value with an average correctness close to 1, and for which the made predictions are almost always right. The corresponding completeness gives an idea of how much we should abstain to get strong guarantees on the prediction, hence of how "hard" is a given data set.

To find those values, we proceed with the following idea: we start from an initial interval of values $[\underline{s}, \overline{s}]$, and from target intervals $[\underline{CP}, \overline{CP}]$ and $[\underline{CR}, \overline{CR}]$, typically $[0.95, 1]$ of average completeness and correctness. Note that in case of inconsistent predictions, $\hat{R}_i = \emptyset$ and the completeness is higher than 1 (in such case, we consider $CR = 0$). For $s_{\min}$, we will typically start from $\underline{s} = 0$ (for which $CP > 1$) and will consider a value $\overline{s}$ large enough for which $CP < 0.95$ (e.g., starting from $s = 2$ as advised in [32] and doubling s iteratively until $CP < 0.95$, as when s increases completeness decreases and correctness increases in average). We then proceed by dichotomy to find a value $s_{\min}$ for which average predictions are within interval $[\underline{CP}, \overline{CP}]$. We proceed similarly for $s_{\max}$.

With $s_{\min}$ and $s_{\max}$ found, a last issue to solve is how to get intermediate values of $s \in [s_{\min}, s_{\max}]$ in order to get an adaptive evolution of completeness/correctness, as in Fig. 2. This is done through a simple procedure: first, we start by calculating the completeness/correctness for the middle value between $s_{\min}$ and $s_{\max}$, that is for $(s_{\min}+s_{\max})/2$. We then compute the distance between all the pairs of completeness/correctness values obtained for consecutive s values, and add a new s point in the middle between the two points with the biggest Euclidean distance. We repeat the process until we get the number of s values requested, for which we provide completeness/correctness values.

[6] Bearing in mind that they can be replaced by any other imprecise classifiers, see [2,6].



(a) Glass data set

(b) Stock data set

(c) Calhousing data set

Fig. 2. Evolution of the hyper-parameter s on glass, stock and calhousing data sets.

The Fig. 2 shows that the boundary values of the hyper-parameter of imprecision s actually significantly depend on the data set. Our approach enables us to find the proper "optimal" value $s_{\min}$ for each data set, which can be small (as in glass where $s_{\min} = 1$) or big (as in calhousing where $s_{\min} = 160$).

Figure 2 is already sufficient to show that our abstention method is working as expected, as indeed correctness increases quickly when we allow abstention, that is when completeness decreases. Figure 2(a) shows that for some data sets, one can have an almost perfect correctness while not being totally vacuous (as correctness of almost 1 is reached for a completeness slightly below 0.5, for a value $s = 4$), while this may not be the case for other more difficult data sets such as calhousing, for which one has to choose a trade-off between completeness and correctness to avoid fully vacuous predictions. Yet, for all data sets (only three being shown for lack of space), we witness a regular increase of correctness.

5.4 Comparison with Other Methods

A remaining question is to know whether our approach is competitive with other state-of-art approaches. To do this, we compare the results obtained on test data sets (in a 10×10 fold cross validation) between the results we obtain for $s = s_{\min}$ and several methods. Those results are indeed the closest we can get to precise predictions in our setting. The methods to which we compare ourselves are the following:

- The ranking by pairwise comparisons (RPC), as implemented in [3];
- The Label ranking tree (LRT [8]), that adopt a local non-decomposed scheme;
- The SVM-LR approach that we already described in Sect. 4.2.

As the NCC deals with discrete attributes, we need to discretize continuous attributes in z intervals before training[7]. While z could be optimized, we use in this paper only two arbitrarily chosen levels of discretization $z = 5$ and $z = 6$ (i.e. LR-CSP-5 and LR-CSP-6 models) to compare our method against the others, for simplicity and because our goal is only to show competitiveness of our approach.

[7] Available in https://github.com/sdestercke/classifip.

As mentioned, we consider the comparison by picking the value $s_{\min}$. By fixing this hyper-parameter regulating the imprecision level of our approach, we then compare the correctness measure (8) with the Spearman Footrule loss obtained for RCP and LRT methods, and implemented into existing software [3]. For the SVM-LR, of which we did not find an online implementation, we used a Python package[8], which solves a quadratic problem with known solvers [1] for little data sets, or a Frank-Wolfe algorithm for bigger data sets. In fact, Frank-Wolfe's algorithm almost certainly guarantees the convergence to the global minimum for convex surfaces and to a local minimum for non-convex surfaces [26].

A last issue to solve is how to handle inconsistency predictions, ones in which the *alldifferent* constraint would not find a precise or partial solution but an empty one. Here, such predictions are ignored, and our results consider correctness and Spearman footrule on consistent solutions only, as dealing with inconsistent predictions will be the object of future works.

5.5 Results

The average performances and their ranks in parentheses obtained in terms of the correctness (CR) measure are shown in Table 4(a) and 4(b), with discretization 5 and 6 respectively applied to our proposal method LR-CSP.

A Friedman test [14] on the ranks yields p-values of 0.00006176 and 0.0001097 for LR-CSP-5 and LR-CSP-6, respectively, thus strongly suggesting performance differences between the algorithms. The Nemenyi post-hoc test (see Table 5) further indicates that LR-CSP-5 (and LR-CSP-6) is significantly better than SVM-LR. Our approach also remains competitive with LRT and RPC.

Finally, recall that our method is also quite fast to compute, thanks to the simultaneous use of decomposition (requiring to build k classifiers), and of probability sets and loss functions offering computational advantages that make the prediction step very efficient. Also, thanks to the fact that our predictions are intervals, i.e. sets of ranks without holes in them, we can use very efficient algorithms to treat the *alldifferent* constraints [28].

Note also that our proposal discretized at $z = 6$ intervals gets more accurate predictions (and also indicate a little drop in the *p-value* of all comparisons of Table 5) what can suggest us that an optimal value of $\hat{z}$ may improve the prediction performance (all that remains, of course, hypothetical).

[8] Available in https://pypi.org/project/svm-label-ranking/.

Table 4. Average correctness accuracies (%) compared to LR-CSP-5 (left) and LR-CSP-6 (right)

	LR-CSP-5	**LRT**	**RPC**	**SVM-LR**
a	94.19 ± 1.31 (1)	91.49 ± 0.31 (3)	93.25 ± 0.25 (2)	64.75 ± 0.41 (4)
b	53.30 ± 3.81 (1)	41.56 ± 1.17 (4)	52.05 ± 0.42 (2)	52.56 ± 0.39 (3)
c	61.46 ± 0.92 (1)	58.33 ± 0.28 (2)	51.76 ± 0.01 (3)	38.48 ± 0.02 (4)
d	68.66 ± 0.63 (1)	60.96 ± 0.24 (3)	62.10 ± 0.04 (2)	47.08 ± 0.85 (4)
e	99.34 ± 0.07 (1)	91.29 ± 0.08 (3)	99.92 ± 0.01 (2)	84.19 ± 2.63 (4)
f	90.92 ± 3.48 (3)	91.75 ± 0.50 (1)	91.05 ± 0.18 (2)	87.16 ± 0.34 (4)
g	79.18 ± 1.98 (2)	84.55 ± 0.51 (1)	74.54 ± 0.15 (3)	69.54 ± 0.35 (4)
h	96.86 ± 3.72 (1)	96.77 ± 0.60 (2)	93.16 ± 0.56 (3)	88.39 ± 0.41 (4)
i	91.55 ± 0.24 (3)	95.15 ± 0.05 (1)	94.12 ± 0.01 (2)	58.66 ± 2.71 (4)
j	90.37 ± 0.46 (3)	96.19 ± 0.09 (1)	94.56 ± 0.02 (2)	66.39 ± 3.00 (4)
k	86.75 ± 1.59 (2)	91.46 ± 0.34 (1)	82.59 ± 0.06 (3)	74.49 ± 0.20 (4)
l	84.81 ± 2.13 (3)	88.07 ± 0.40 (2)	89.28 ± 0.17 (1)	82.30 ± 0.96 (4)
m	86.32 ± 2.34 (1)	85.36 ± 0.97 (2)	74.32 ± 0.06 (3)	66.60 ± 1.23 (4)
n	97.98 ± 2.89 (1)	91.75 ± 0.88 (4)	94.55 ± 0.62 (2)	94.53 ± 0.50 (3)
avg.	84.41 ± 1.83(1.72)	83.19 ± 0.46(2.14)	81.95 ± 0.18(2.28)	69.65 ± 1.00(3.86)

	LR-CSP-6	**LRT**	**RPC**	**SVM-LR**
a	93.90 ± 0.69 (1)	91.53 ± 0.31 (3)	93.21 ± 0.23 (2)	64.42 ± 0.36 (4)
b	54.12 ± 3.73 (1)	41.70 ± 1.48 (4)	50.43 ± 0.39 (3)	51.10 ± 0.49 (2)
c	61.05 ± 0.80 (1)	58.37 ± 0.28 (2)	51.85 ± 0.02 (3)	38.45 ± 0.02 (4)
d	68.72 ± 1.42 (1)	60.76 ± 0.30 (3)	61.93 ± 0.04 (2)	46.71 ± 0.87 (4)
e	99.20 ± 0.07 (2)	91.26 ± 0.06 (3)	99.92 ± 0.01 (1)	84.18 ± 2.67 (4)
f	91.95 ± 2.90 (1)	91.59 ± 0.47 (2)	90.83 ± 0.24 (3)	85.68 ± 0.33 (4)
g	79.21 ± 3.37 (2)	85.09 ± 0.46 (1)	74.86 ± 0.16 (3)	70.16 ± 0.46 (4)
h	99.36 ± 1.28 (1)	97.16 ± 0.55 (2)	92.75 ± 0.58 (3)	87.39 ± 0.37 (4)
i	91.31 ± 0.14 (3)	95.14 ± 0.05 (1)	94.12 ± 0.01 (2)	58.75 ± 2.71 (4)
j	91.20 ± 0.85 (3)	96.11 ± 0.10 (1)	94.52 ± 0.03 (2)	66.25 ± 3.05 (4)
k	88.63 ± 1.53 (2)	91.64 ± 0.27 (1)	82.23 ± 0.08 (3)	75.20 ± 0.17 (4)
l	85.29 ± 1.91 (3)	88.03 ± 0.44 (2)	89.24 ± 0.14 (1)	81.93 ± 1.00 (4)
m	88.23 ± 1.00 (1)	84.40 ± 0.62 (2)	72.88 ± 0.06 (3)	65.41 ± 1.21 (4)
n	98.20 ± 1.19 (1)	91.80 ± 0.87 (4)	94.58 ± 0.61 (2)	94.56 ± 0.50 (3)
avg.	85.03 ± 1.49(1.64)	83.18 ± 0.45(2.21)	81.67 ± 0.19(2.36)	69.30 ± 1.02(3.79)

Table 5. Nemenyi post-hoc test: null hypothesis H_0 and p-value

#	H_0	LRT	RPC	SVM-LR
1	LR-CSP-5 =	0.8161	0.6452	**0.000066**
2	LR-CSP-6 =	0.6450	0.4600	**0.000066**

6 Conclusion and Perspectives

In this paper, we have proposed a method to make partial predictions in label ranking, using a label-wise decomposition as well as a new kind of partial predictions in terms of possible ranks. The experiments on synthetic data sets show that our proposed model (LR-CSP) produces reliable and cautious predictions and performs close to or even outperforms the existing alternative models.

This is quite encouraging, as we left a lot of room for optimization, e.g., in the base classifiers or in the discretization. However, while our method extends straightforwardly to partially observed rankings in training data when those are top-k rankings (considering for instance the rank of all remaining labels as $k+1$), it may be trickier to apply it to pairwise rankings, another popular way to get such data. Some of our future works will focus on that.

Acknowledgments. This work was carried out in the framework of the Labex MS2T and PreServe projects, funded by the French Government, through the National Agency for Research (Reference ANR-11-IDEX-0004-02 and ANR-18-CE23-0008).

References

1. Andersen, M., Dahl, J., Vandenberghe, L.: CVXOPT: a python package for convex optimization (2013). http://abel.ee.ucla.edu/cvxopt
2. Augustin, T., Coolen, F., de Cooman, G., Troffaes, M.: Introduction to Imprecise Probabilities. Wiley, Chichester (2014)
3. Balz, A., Senge, R.: WEKA-LR: a label ranking extension for weka (2011). https://cs.uni-paderborn.de/de/is/research/research-projects/software/weka-lr-a-label-ranking-extension-for-weka/
4. Bartlett, P., Wegkamp, M.: Classification with a reject option using a hinge loss. J. Mach. Learn. Res. **9**, 1823–1840 (2008)
5. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. J. Mach. Learn. Res. Proc. Track **22**, 127–135 (2012)
6. Carranza-Alarcon, Y.C., Destercke, S.: Imprecise gaussian discriminant classification. In: International Symposium on Imprecise Probabilities: Theories and Applications, pp. 59–67 (2019)
7. Cheng, W., Dembczynski, K., Hüllermeier, E.: Label ranking methods based on the Plackett-Luce model. In: Proceedings of the 27th Annual International Conference on Machine Learning - ICML, pp. 215–222 (2010)
8. Cheng, W., Hühn, J., Hüllermeier, E.: Decision tree and instance-based learning for label ranking. In: Proceedings of the 26th Annual International Conference on Machine Learning - ICML 2009 (2009)

9. Cheng, W., Rademaker, M., De Baets, B., Hüllermeier, E.: Predicting partial orders: ranking with abstention. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6321, pp. 215–230. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15880-3_20
10. Cheng, W., Hüllermeier, E., Waegeman, W., Welker, V.: Label ranking with partial abstention based on thresholded probabilistic models. In: Advances in Neural Information Processing Systems, pp. 2501–2509 (2012)
11. Corani, G., Antonucci, A., Zaffalon, M.: Bayesian networks with imprecise probabilities: theory and application to classification. In: Holmes, D.E., Jain, L.C. (eds.) Data Mining: Foundations and Intelligent Paradigms, pp. 49–93. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-23166-7_4
12. Dechter, R.: Constraint Processing. Morgan Kaufmann, San Mateo (2003)
13. Dekel, O., Manning, C.D., Singer, Y.: Log-linear models for label ranking. In: Advances in Neural Information Processing Systems (2003)
14. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
15. Destercke, S., Dubois, D., Chojnacki, E.: Unifying practical uncertainty representations: I. Generalized p-boxes. Int. J. Approximate Reasoning **49**, 649–663 (2008)
16. Destercke, S.: On the median in imprecise ordinal problems. Ann. Oper. Res. **256**(2), 375–392 (2016). https://doi.org/10.1007/s10479-016-2253-x
17. Destercke, S., Dubois, D.: Special cases. In: Introduction to Imprecise Probabilities, pp. 79–92 (2014)
18. Destercke, S., Masson, M.H., Poss, M.: Cautious label ranking with label-wise decomposition. Eur. J. Oper. Res. **246**(3), 927–935 (2015)
19. Destercke, S., Yang, G.: Cautious ordinal classification by binary decomposition. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 323–337. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_21
20. Elisseeff, A., Weston, J.: Kernel methods for multi-labelled classification and categorical regression problems. In: Advances in Neural Information Processing Systems, pp. 681–687. MIT Press, Cambridge (2002)
21. Frank, A., Asuncion, A.: UCI machine learning repository (2010). http://archive.ics.uci.edu/ml
22. Frank, E., Hall, M.: A simple approach to ordinal classification. In: De Raedt, L., Flach, P. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44795-4_13
23. Geng, X.: Multilabel ranking with inconsistent rankers. In: Proceedings of CVPR 2014 (2014)
24. Hüllermeier, E., Furnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artif. Intell. **172**, 1897–1916 (2008)
25. King, R., Feng, C., Sutherland, A.: StatLog: comparison of classification algorithms on large real-world problems. App. Artif. Intell. **9**(3), 289–333 (1995)
26. Lacoste-Julien, S.: Convergence rate of Frank-Wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345 (2016)
27. Li, L., Lin, H.T.: Ordinal regression by extended binary classification. In: Advances in Neural Information Processing Systems, pp. 865–872 (2007)
28. López-Ortiz, A., Quimper, C.G., Tromp, J., Van Beek, P.: A fast and simple algorithm for bounds consistency of the all different constraint. In: IJCAI, vol. 3, pp. 245–250 (2003)
29. Steinwart, I., Christmann, A., et al.: Estimating conditional quantiles with the help of the pinball loss. Bernoulli **17**(1), 211–225 (2011)

30. Vembu, S., Gärtner, T.: Label ranking algorithms: a survey. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 45–64. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14125-6_3
31. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, New York (1991)
32. Walley, P.: Inferences from multinomial data: learning about a bag of marbles. J. Roy. Stat. Soc.: Ser. B (Methodol.) **58**(1), 3–34 (1996)
33. Weskamp, N., Hullermeier, E., Kuhn, D., Klebe, G.: Multiple graph alignment for the structural analysis of protein active sites. IEEE/ACM Trans. Comput. Biol. Bioinf. **4**(2), 310–320 (2007)
34. Yang, G., Destercke, S., Masson, M.H.: The costs of indeterminacy: how to determine them? IEEE Trans. Cybern. **47**(12), 4316–4327 (2016)
35. Zaffalon, M.: The naive credal classifier. J. Stat. Plann. Infer. **105**(1), 5–21 (2002)
36. Zaffalon, M., Corani, G., Mauá, D.: Evaluating credal classifiers by utility-discounted predictive accuracy. Int. J. Approximate Reasoning **53**(8), 1282–1301 (2012)