A Structured Prediction Approach for Label Ranking

Anna Korba, Alexandre Garcia, Florence d'Alché-Buc
LTCI, Télécom ParisTech
Université Paris-Saclay
Paris, France
firstname.lastname@telecom-paristech.fr

Abstract

We propose to solve a label ranking problem as a structured output regression task. In this view, we adopt a least square surrogate loss approach that solves a supervised learning problem in two steps: a regression step in a well-chosen feature space and a pre-image (or decoding) step. We use specific feature maps/embeddings for ranking data, which convert any ranking/permutation into a vector representation. These embeddings are all well-tailored for our approach, either by resulting in consistent estimators, or by solving trivially the pre-image problem which is often the bottleneck in structured prediction. Their extension to the case of incomplete or partial rankings is also discussed. Finally, we provide empirical results on synthetic and real-world datasets showing the relevance of our method.

1 Introduction

Label ranking is a prediction task which aims at mapping input instances to a (total) order over a given set of labels indexed by $\{1, \dots, K\}$. This problem is motivated by applications where the output reflects some preferences, or order of relevance, among a set of objects. Hence there is an increasing number of practical applications of this problem in the machine learning litterature. In pattern recognition for instance (Geng and Luo, 2014), label ranking can be used to predict the different objects which are the more likely to appear in an image among a predefined set. Similarly, in sentiment analysis, (Wang et al., 2011) where the prediction of the emotions expressed in a document is cast as a label ranking problem over a set of possible affective expressions. In ad targeting, the prediction of preferences of a web user over ad categories (Djuric et al., 2014) can be also formalized as a label ranking problem, and the prediction as a ranking guarantees that each user is qualified into several categories, eliminating overexposure. Another application is metalearning, where the goal is to rank a set of algorithms according to their suitability based on the characteristics of a target dataset and learning problem (see Brazdil et al. (2003); Aiguzhinov et al. (2010)). Interestingly, the label ranking problem can also be seen as an extension of several supervised tasks, such as multiclass classification or multi-label ranking (see Dekel et al. (2004); Fürnkranz and Hüllermeier (2003)). Indeed for these tasks, a prediction can be obtained by postprocessing the output of a label ranking model in a suitable way. However, label ranking differs from other ranking problems, such as in information retrieval or recommender systems, where the goal is (generally) to predict a target variable under the form of a rating or a relevance score (Cao et al., 2007).

More formally, the goal of label ranking is to map a vector x lying in some feature space $\mathcal{X}$ to a ranking y lying in the space of rankings $\mathcal{Y}$. A ranking is an ordered list of items of the set $\{1, \dots, K\}$. These relations linking the components of the y objects induce a structure on the output space $\mathcal{Y}$. The label ranking task thus naturally enters the framework of structured output prediction for which an abundant litterature is available (Nowozin and Lampert, 2011). In this paper, we adopt the Surrogate Least Square Loss approach introduced in the context of output kernels (Cortes et al., 2005; Kadri et al., 2013; Brouard et al., 2016) and recently theoretically studied by Ciliberto et al.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

(2016) and Osokin et al. (2017) using Calibration theory (Steinwart and Christmann, 2008). This approach divides the learning task in two steps: the first one is a vector regression step in a Hilbert space where the outputs objects are represented through an embedding, and the second one solves a pre-image problem to retrieve an output object in the $\mathcal{Y}$ space. In this framework, the algorithmic complexity of the learning and prediction tasks as well as the generalization properties of the resulting predictor crucially rely on some properties of the embedding. In this work we study and discuss some embeddings dedicated to ranking data.

Our contribution is three folds: (1) we cast the label ranking problem into the structured prediction framework and propose embeddings dedicated to ranking representation, (2) for each embedding we propose a solution to the pre-image problem and study its algorithmic complexity and (3) we provide theoretical and empirical evidence for the relevance of our method.

The paper is organized as follows. In section 2, definitions and notations of objects considered through the paper are introduced, and section 3 is devoted to the statistical setting of the learning problem. section 4 describes at length the embeddings we propose and section 5 details the theoretical and computational advantages of our approach. Finally section 6 contains empirical results on benchmark datasets.

2 Preliminaries

2.1 Mathematical background and notations

Consider a set of items indexed by $\{1, \dots, K\}$, that we will denote $[\![K]\!]$. Rankings, i.e. ordered lists of items of $[\![K]\!]$, can be complete (i.e, involving all the items) or incomplete and for both cases, they can be without-ties (total order) or with-ties (weak order). A *full ranking* is a complete, and without-ties ranking of the items in $[\![K]\!]$. It can be seen as a permutation, i.e a bijection $\sigma : [\![K]\!] \to [\![K]\!]$, mapping each item i to its rank $\sigma(i)$. The rank of item i is thus $\sigma(i)$ and the item ranked at position j is $\sigma^{-1}(j)$. We say that i is preferred over j (denoted by $i \succ j$) according to σ if and only if i is ranked lower than j: $\sigma(i) < \sigma(j)$. The set of all permutations over K items is the symmetric group which we denote by $\mathfrak{S}_K$. A *partial ranking* is a complete ranking including ties, and is also referred as a weak order or bucket order in the litterature (see Kenkre et al. (2011)). This includes in particular the top-k rankings, that is to say partial rankings dividing items in two groups, the first one being the $k \leq K$ most relevant items and the second one including all the rest. These top-k rankings are given a lot of attention because of their relevance for modern applications, especially search engines or recommendation systems (see Ailon (2010)). An *incomplete ranking* is a strict order involving only a small subset of items, and includes as a particular case pairwise comparisons, another kind of ranking which is very relevant in large-scale settings when the number of items to be ranked is very large. We now introduce the main notations used through the paper. For any function f, $Im(f)$ denotes the image of f, and f^{-1} its inverse. The indicator function of any event $\mathcal{E}$ is denoted by $\mathbb{I}\{\mathcal{E}\}$. We will denote by $sign$ the function such that for any $x \in \mathbb{R}$, $sign(x) = \mathbb{I}\{x > 0\} - \mathbb{I}\{x < 0\}$. The notations $\|.\|$ and $|.|$ denote respectively the usual l_2 and l_1 norm in an Euclidean space. Finally, for any integers $a \leq b$, $[\![a, b]\!]$ denotes the set $\{a, a+1, \dots, b\}$, and for any finite set C, $\#C$ denotes its cardinality.

2.2 Related work

An overview of label ranking algorithms can be found in Vembu and Gärtner (2010), Zhou et al. (2014)), but we recall here the main contributions. One of the first proposed approaches, called *pairwise classification* (see Fürnkranz and Hüllermeier (2003)) transforms the label ranking problem into $K(K-1)/2$ binary classification problems. For each possible pair of labels $1 \leq i < j \leq K$, the authors learn a model m_{ij} that decides for any given example whether $i \succ j$ or $j \succ i$ holds. The model is trained with all examples for which either $i \succ j$ or $j \succ i$ is known (all examples for which nothing is known about this pair are ignored). At prediction time, an example is submitted to all $K(K-1)/2$ classifiers, and each prediction is interpreted as a vote for a label: if the classifier m_{ij} predicts $i \succ j$, this counts as a vote for label i. The labels are then ranked according to the number of votes. Another approach (see Dekel et al. (2004)) consists in learning for each label a linear utility function from which the ranking is deduced. Then, a large part of the dedicated literature was devoted to adapting classical partitioning methods such as k-nearest neighbors (see Zhang and Zhou (2007), Chiang et al. (2012)) or tree-based methods, in a parametric (Cheng et al. (2010), Cheng et al.

(2009), Aledo et al. (2017)) or a non-parametric way (see Cheng and Hüllermeier (2013), Yu et al. (2010), Zhou and Qiu (2016), Clémençon et al. (2017), Sá et al. (2017)). Finally, some approaches are rule-based (see Gurrieri et al. (2012), de Sá et al. (2018)). We will compare our numerical results with the best performances attained by these methods on a set of benchmark datasets of the label ranking problem in section 6.

3 Structured prediction for label ranking

3.1 Learning problem

Our goal is to learn a function $s : \mathcal{X} \to \mathcal{Y}$ between a feature space $\mathcal{X}$ and a structured output space $\mathcal{Y}$, that we set to be $\mathfrak{S}_K$ the space of full rankings over the set of items $[\![K]\!]$. The quality of a prediction $s(x)$ is measured using a loss function $\Delta : \mathfrak{S}_K \times \mathfrak{S}_K \to \mathbb{R}$, where $\Delta(s(x), \sigma)$ is the cost suffered by predicting $s(x)$ for the true output σ. We suppose that the input/output pairs (x, σ) come from some fixed distribution P on $\mathcal{X} \times \mathfrak{S}_K$. The label ranking problem is then defined as:

$$\text{minimize}_{s:\mathcal{X}\to\mathfrak{S}_K} \mathcal{E}(s), \quad \text{with} \quad \mathcal{E}(s) = \int_{\mathcal{X}\times\mathfrak{S}_K} \Delta(s(x), \sigma) dP(x, \sigma). \tag{1}$$

In this paper, we propose to study how to solve this problem and its empirical counterpart for a family of loss functions based on some ranking embedding $\phi : \mathfrak{S}_K \to \mathcal{F}$ that maps the permutations $\sigma \in \mathfrak{S}_K$ into a Hilbert space $\mathcal{F}$:

$$\Delta(\sigma, \sigma') = \|\phi(\sigma) - \phi(\sigma')\|^2_{\mathcal{F}}. \tag{2}$$

This loss presents two main advantages: first, there exists popular losses for ranking data that can take this form within a finite dimensional Hilbert Space $\mathcal{F}$, second, this choice benefits from the theoretical results on Surrogate Least Square problems for structured prediction using Calibration Theory of Ciliberto et al. (2016) and of works of Brouard et al. (2016) on Structured Output Prediction within vector-valued Reproducing Kernel Hilbert Spaces. These works approach Structured Output Prediction along a common angle by introducing a surrogate problem involving a function $g : \mathcal{X} \to \mathcal{F}$ (with values in $\mathcal{F}$) and a surrogate loss $L(g(x), \sigma)$ to be minimized instead of Eq. 1. The surrogate loss is said to be calibrated if a minimizer for the surrogate loss is always optimal for the true loss (Calauzenes et al., 2012). In the context of true risk minimization, the surrogate problem for our case writes as:

$$\text{minimize}_{\ g:\mathcal{X}\to\mathcal{F}} \mathcal{R}(g), \quad \text{with} \quad \mathcal{R}(g) = \int_{\mathcal{X}\times\mathfrak{S}_K} L(g(x), \phi(\sigma)) dP(x, \sigma). \tag{3}$$

with the following surrogate loss:

$$L(g(x), \phi(\sigma)) = \|g(x) - \phi(\sigma)\|^2_{\mathcal{F}}. \tag{4}$$

Problem of Eq. (3) is in general easier to optimize since g has values in $\mathcal{F}$ instead of the set of structured objects $\mathcal{Y}$, here $\mathfrak{S}_K$. The solution of (3), denoted as g^*, can be written for any $x \in \mathcal{X}$: $g^*(x) = \mathbb{E}[\phi(\sigma)|x]$. Eventually, a candidate $s(x)$ pre-image for $g^*(x)$ can then be obtained by solving:

$$s(x) = \operatorname*{argmin}_{\sigma\in\mathfrak{S}_K} L(g^*(x), \phi(\sigma)). \tag{5}$$

In the context of Empirical Risk Minimization, a training sample $\mathcal{S} = \{(x_i, \sigma_i), i = 1, \ldots, N\}$, with N i.i.d. copies of the random variable (x, σ) is available. The Surrogate Least Square approach for Label Ranking Prediction decomposes into two steps:

- Step 1: minimize a regularized empirical risk to provide an estimator of the minimizer of the regression problem in Eq. (3):

$$\text{minimize}_{\ g\in\mathcal{H}} \mathcal{R}_{\mathcal{S}}(g), \quad \text{with} \quad \mathcal{R}_{\mathcal{S}}(g) = \frac{1}{N}\sum_{i=1}^{N} L(g(x_i), \phi(\sigma_i)) + \Omega(g). \tag{6}$$

  with an appropriate choice of hypothesis space $\mathcal{H}$ and complexity term $\Omega(g)$. We denote by $\widehat{g}$ a solution of (6).

- Step 2: solve, for any x in $\mathcal{X}$, the pre-image problem that provides a prediction in the original space $\mathfrak{S}_K$:

$$\widehat{s}(x) = \underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmin}} \|\phi(\sigma) - \widehat{g}(x)\|_{\mathcal{F}}^2. \tag{7}$$

The pre-image operation can be written as $\widehat{s}(x) = d \circ \widehat{g}(x)$ with d the decoding function:

$$d(h) = \underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmin}} \|\phi(\sigma) - h\|_{\mathcal{F}}^2 \text{ for all } h \in \mathcal{F}, \tag{8}$$

applied on $\widehat{g}$ for any $x \in \mathcal{X}$.

This paper studies how to leverage the choice of the embedding ϕ to obtain a good compromise between computational complexity and theoretical guarantees. Typically, the pre-image problem on the discrete set $\mathfrak{S}_K$ (of cardinality $K!$) can be eased for appropriate choices of ϕ as we show in section 4, leading to efficient solutions. In the same time, one would like to benefit from theoretical guarantees and control the excess risk of the proposed predictor $\widehat{s}$.

In the following subsection we exhibit popular losses for ranking data that we will use for the label ranking problem.

3.2 Losses for ranking

We now present losses Δ on $\mathfrak{S}_K$ that we will consider for the label ranking task. A natural loss for full rankings, i.e. permutations in $\mathfrak{S}_K$, is a distance between permutations. Several distances on $\mathfrak{S}_K$ are widely used in the literature (Deza and Deza, 2009), one of the most popular being the *Kendall's τ distance*, which counts the number of pairwise disagreements between two permutations $\sigma, \sigma' \in \mathfrak{S}_K$:

$$\Delta_\tau(\sigma, \sigma') = \sum_{i<j} \mathbb{I}[(\sigma(i) - \sigma(j))(\sigma'(i) - \sigma'(j)) < 0]. \tag{9}$$

The maximal Kendall's τ distance is thus $K(K-1)/2$, the total number of pairs. Another well-spread distance between permutations is the *Hamming distance*, which counts the number of entries on which two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ disagree:

$$\Delta_H(\sigma, \sigma') = \sum_{i=1}^{K} \mathbb{I}[\sigma(i) \neq \sigma'(i)]. \tag{10}$$

The maximal Hamming distance is thus K, the number of labels or items.

The Kendall's τ distance is a natural discrepancy measure when permutations are interpreted as rankings and is thus the most widely used in the preference learning literature. In contrast, the Hamming distance is particularly used when permutations represent matching of bipartite graphs and is thus also very popular (see Fathony et al. (2018)). In the next section we show how these distances can be written as Eq. (2) for a well chosen embedding ϕ.

4 Output embeddings for rankings

In what follows, we study three embeddings tailored to represent full rankings/permutations in $\mathfrak{S}_K$ and discuss their properties in terms of link with the ranking distances Δ_τ and Δ_H, and in terms of algorithmic complexity for the pre-image problem (5) induced.

4.1 The Kemeny embedding

Motivated by the minimization of the Kendall's τ distance Δ_τ, we study the Kemeny embedding, previously introduced for the ranking aggregation problem (see Jiao et al. (2016)):

$$\begin{aligned} \phi_\tau \colon \mathfrak{S}_K &\to \mathbb{R}^{K(K-1)/2} \\ \sigma &\mapsto (\operatorname{sign}(\sigma(j) - \sigma(i)))_{1 \leq i < j \leq K} \ . \end{aligned}$$

which maps any permutation $\sigma \in \mathfrak{S}_K$ into $Im(\phi_\tau) \subsetneq \{-1, 1\}^{K(K-1)/2}$ (that we have embedded into the Hilbert space $(\mathbb{R}^{K(K-1)/2}, \langle ., . \rangle)$). One can show that the square of the euclidean distance

between the mappings of two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ recovers their Kendall's τ distance (proving at the same time that ϕ_τ is injective) up to a constant: $\|\phi_\tau(\sigma) - \phi_\tau(\sigma')\|^2 = 4\Delta_\tau(\sigma, \sigma')$. The Kemeny embedding then naturally appears to be a good candidate to build a surrogate loss related to Δ_τ. By noticing that ϕ_τ has a constant norm ($\forall \sigma \in \mathfrak{S}_K$, $\|\phi_\tau(\sigma)\| = \sqrt{K(K-1)/2}$), we can rewrite the pre-image problem (7) under the form:

$$\widehat{s}(x) = \underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmin}} -\langle \phi_\tau(\sigma), \widehat{g}(x) \rangle. \tag{11}$$

To compute (11), one can first solve an Integer Linear Program (ILP) to find $\widehat{\phi_\sigma} = \operatorname{argmin}_{\phi_\sigma \in Im(\phi_\tau)} -\langle \phi_\sigma, \widehat{g}(x) \rangle$, and then find the output object $\sigma = \phi_\tau^{-1}(\widehat{\phi_\sigma})$. The latter step, i.e. inverting ϕ_τ, can be performed in $\mathcal{O}(K^2)$ by means of the Copeland method (see Merlin and Saari (1997)), which ranks the items by their number of pairwise victories[1]. In contrast, the ILP problem is harder to solve since it involves a minimization over $Im(\phi_\tau)$, a set of structured vectors since their coordinates are strongly correlated by the *transitivity* property of rankings. Indeed, consider a vector $v \in Im(\phi_\tau)$, so $\exists \sigma \in \mathfrak{S}_K$ such that $v = \phi_\tau(\sigma)$. Then, for any $1 \leq i < j < k \leq K$, if its coordinates corresponding to the pairs (i, j) and (j, k) are equal to one (meaning that $\sigma(i) < \sigma(j)$ and $\sigma(j) < \sigma(k)$), then the coordinate corresponding to the pair (i, k) cannot contradict the others and must be set to one as well. Since $\phi_\sigma = (\phi_\sigma)_{i,j} \in Im(\phi_\tau)$ is only defined for $1 \leq i < j \leq K$, one cannot directly encode the transitivity constraints that take into account the components $(\phi_\sigma)_{i,j}$ with $j > i$. Thus to encode the transitivity constraint we introduce $\phi'_\sigma = (\phi'_\sigma)_{i,j} \in \mathbb{R}^{K(K-1)}$ defined by $(\phi'_\sigma)_{i,j} = (\phi_\sigma)_{i,j}$ if $1 \leq i < j \leq K$ and $(\phi'_\sigma)_{i,j} = -(\phi_\sigma)_{i,j}$ else, and write the ILP problem as follows:

$$\begin{aligned} &\widehat{\phi_\sigma} = \underset{\phi'_\sigma}{\operatorname{argmin}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi'_\sigma)_{i,j}, \\ &s.c. \quad \begin{cases} (\phi'_\sigma)_{i,j} \in \{-1, 1\} \quad \forall\, i, j \\ (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,i} = 0 \quad \forall\, i, j \\ -1 \leq (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,k} + (\phi'_\sigma)_{k,i} \leq 1 \quad \forall\, i, j, k \text{ s.t. } i \neq j \neq k. \end{cases} \end{aligned} \tag{12}$$

Such a problem is NP-Hard. In previous works (see Calauzenes et al. (2012); Ramaswamy et al. (2013)), the complexity of designing calibrated surrogate losses for the Kendall's τ distance had already been investigated. In particular, Calauzenes et al. (2012) proved that there exists no convex K-dimensional calibrated surrogate loss for Kendall's τ distance. As a consequence, optimizing this type of loss has an inherent computational cost. However, in practice, branch and bound based ILP solvers find the solution of (12) in a reasonable time for a reduced number of labels K. We discuss the computational implications of choosing the Kemeny embedding section 5.2. We now turn to the study of an embedding devoted to build a surrogate loss for the Hamming distance.

4.2 The Hamming embedding

Another well-spread embedding for permutations, that we will call the Hamming embedding, consists in mapping σ to its permutation matrix $\phi_H(\sigma)$:

$$\begin{aligned} \phi_H \colon \mathfrak{S}_K &\to \mathbb{R}^{K \times K} \\ \sigma &\mapsto (\mathbb{I}\{\sigma(i) = j\})_{1 \leq i,j \leq K}\ , \end{aligned}$$

where we have embedded the set of permutation matrices $Im(\phi_H) \subsetneq \{0, 1\}^{K \times K}$ into the Hilbert space $(\mathbb{R}^{K \times K}, \langle ., . \rangle)$ with $\langle ., . \rangle$ the Froebenius inner product. This embedding shares similar properties with the Kemeny embedding: first, it is also of constant (Froebenius) norm, since $\forall \sigma \in \mathfrak{S}_K,\ \|\phi_H(\sigma)\| = \sqrt{K}$. Then, the squared euclidean distance between the mappings of two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ recovers their Hamming distance (proving that ϕ_H is also injective): $\|\phi_H(\sigma) - \phi_H(\sigma')\|^2 = \Delta_H(\sigma, \sigma')$. Once again, the pre-image problem consists in solving the linear program:

$$\widehat{s}(x) = \underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmin}} -\langle \phi_H(\sigma), \widehat{g}(x) \rangle, \tag{13}$$

[1]Copeland method firstly affects a score s_i for item i as: $s_i = \sum_{j \neq i} \mathbb{I}\{\sigma(i) < \sigma(j)\}$ and then ranks the items by decreasing score.

which is, as for the Kemeny embedding previously, divided in a minimization step, i.e. find $\widehat{\phi_\sigma} = \operatorname{argmin}_{\phi_\sigma \in Im(\phi_H)} -\langle \phi_\sigma, g(x) \rangle$, and an inversion step, i.e. compute $\sigma = \phi_H^{-1}(\widehat{\phi_\sigma})$. The inversion step is of complexity $\mathcal{O}(K^2)$ since it involves scrolling through all the rows (items i) of the matrix $\widehat{\phi_\sigma}$ and all the columns (to find their positions $\sigma(i)$). The minimization step itself writes as the following problem:

$$\begin{aligned} &\widehat{\phi_\sigma} = \underset{\phi_\sigma}{\operatorname{argmax}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi_\sigma)_{i,j}, \\ &s.c \quad \begin{cases} (\phi_\sigma)_{i,j} \in \{0,1\} \quad \forall\, i,j \\ \sum_i (\phi_\sigma)_{i,j} = \sum_j (\phi_\sigma)_{i,j} = 1 \quad \forall\, i,j\ , \end{cases} \end{aligned} \tag{14}$$

which can be solved with the Hungarian algorithm (see Kuhn (1955)) in $\mathcal{O}(K^3)$ time. Now we turn to the study of an embedding which presents efficient algorithmic properties.

4.3 Lehmer code

A permutation $\sigma = (\sigma(1), \ldots, \sigma(K)) \in \mathfrak{S}_K$ may be uniquely represented via its Lehmer code (also called the inversion vector), i.e. a word of the form $c_\sigma \in \mathcal{C}_K \triangleq \{0\} \times [\![0,1]\!] \times [\![0,2]\!] \times \cdots \times [\![0,K-1]\!]$, where for $j = 1, \ldots, K$:

$$c_\sigma(j) = \#\{i \in [\![K]\!] : i < j, \sigma(i) > \sigma(j)\}. \tag{15}$$

The coordinate $c_\sigma(j)$ is thus the number of elements i with index smaller than j that are ranked higher than j in the permutation σ. By default, $c_\sigma(1) = 0$ and is typically omitted. For instance, we have:

e	1	2	3	4	5	6	7	8	9
σ	2	1	4	5	7	3	6	9	8
c_σ	0	1	0	0	0	3	1	0	1

It is well known that the Lehmer code is bijective, and that the encoding and decoding algorithms have linear complexity $\mathcal{O}(K)$ (see Mareš and Straka (2007), Myrvold and Ruskey (2001)). This embedding has been recently used for ranking aggregation of full or partial rankings (see Li et al. (2017)). Our idea is thus to consider the following Lehmer mapping for label ranking;

$$\begin{aligned} \phi_L \colon \mathfrak{S}_K &\to \mathbb{R}^K \\ \sigma &\mapsto (c_\sigma(i)))_{i=1,\ldots,K}\ , \end{aligned}$$

which maps any permutation $\sigma \in \mathfrak{S}_K$ into the space $\mathcal{C}_K$ (that we have embedded into the Hilbert space $(\mathbb{R}^K, \langle ., . \rangle)$). The loss function in the case of the Lehmer embedding is thus the following:

$$\Delta_L(\sigma, \sigma') = \|\phi_L(\sigma) - \phi_L(\sigma')\|^2, \tag{16}$$

which does not correspond to a known distance over permutations (Deza and Deza, 2009). Notice that $|\phi_L(\sigma)| = d_\tau(\sigma, e)$ where e is the identity permutation, a quantity which is also called the number of inversions of σ. Therefore, in contrast to the previous mappings, the norm $\|\phi_L(\sigma)\|$ is not constant for any $\sigma \in \mathfrak{S}_K$. Hence it is not possible to write the loss $\Delta_L(\sigma, \sigma')$ as $-\langle \phi_L(\sigma), \phi_L(\sigma') \rangle$[2].Moreover, this mapping is not distance preserving and it can be proven that $\frac{1}{K-1} \Delta_\tau(\sigma, \sigma') \leq |\phi_L(\sigma) - \phi_L(\sigma')| \leq \Delta_\tau(\sigma, \sigma')$ (see Wang et al. (2015)). However, the Lehmer embedding still enjoys great advantages. Firstly, its coordinates are decoupled, which will enable a trivial solving of the inverse image step (7). Indeed we can write explicitly its solution as:

$$\widehat{s}(x) = \underbrace{\phi_L^{-1} \circ d_L}_{d} \circ \widehat{g}(x) \quad \text{with} \quad \begin{aligned} d_L \colon \mathbb{R}^K &\to \mathcal{C}_K \\ (h_i)_{i=1,\ldots,K} &\mapsto (\underset{j \in [\![0, i-1]\!]}{\operatorname{argmin}} (h_i - j))_{i=1,\ldots,K}, \end{aligned} \tag{17}$$

where d is the decoding function defined in (8). Then, there may be repetitions in the coordinates of the Lehmer embedding, allowing for a compact representation of the vectors.

[2]The scalar product of two embeddings of two permutations $\phi_L(\sigma), \phi_L(\sigma')$ is not maximized for $\sigma = \sigma'$.

4.4 Extension to partial and incomplete rankings

In many real-world applications, one does not observe full rankings but only partial or incomplete rankings (see the definitions section 2.1). We now discuss to what extent the embeddings we propose for permutations can be adapted to this kind of rankings *as input data*. Firstly, the Kemeny embedding can be naturally extended to partial and incomplete rankings since it encodes *relative* information about the positions of the items. Indeed, we propose to map any partial ranking $\widetilde{\sigma}$ to the vector:

$$\phi(\widetilde{\sigma}) = (sign(\widetilde{\sigma}(i) - \widetilde{\sigma}(j))_{1 \leq i < j \leq K}, \tag{18}$$

where each coordinate can now take its value in $\{-1, 0, 1\}$ (instead of $\{-1, 1\}$ for full rankings). For any incomplete ranking $\bar{\sigma}$, we also propose to fill the missing entries (missing comparisons) in the embedding with zeros. This can be interpreted as setting the probability that $i \succ j$ to 1/2 for a missing comparison between (i, j). In contrast, the Hamming embedding, since it encodes the absolute positions of the items, is tricky to extend to map partial or incomplete rankings where this information is missing. Finally, the Lehmer embedding falls between the two latter embeddings. It also relies on an encoding of relative rankings and thus may be adapted to take into account the partial ranking information. Indeed, in Li et al. (2017), the authors propose a generalization of the Lehmer code for partial rankings. We recall that a tie in a ranking happens when $\#\{i \neq j, \sigma(i) = \sigma(j)\} > 0$. The generalized representation c' takes into account ties, so that for any partial ranking $\widetilde{\sigma}$:

$$c'_{\widetilde{\sigma}}(j) = \#\{i \in [\![K]\!] : i < j, \widetilde{\sigma}(i) \geq \widetilde{\sigma}(j)\}. \tag{19}$$

Clearly, $c'_{\widetilde{\sigma}}(j) \geq c_{\widetilde{\sigma}}(j)$ for all $j \in [\![K]\!]$. Given a partial ranking $\widetilde{\sigma}$, it is possible to break its ties to convert it in a permutation σ as follows: for $i, j \in [\![K]\!]^2$, if $\widetilde{\sigma}(i) = \widetilde{\sigma}(j)$ then $\sigma(i) = \sigma(j)$ iff $i < j$. The entries $j = 1, \ldots, K$ of the Lehmer codes of $\widetilde{\sigma}$ (see (20)) and σ (see (15)) then verify:

$$c'_{\widetilde{\sigma}}(j) = c_{\sigma}(j) + IN_j - 1 \quad , \quad c_{\widetilde{\sigma}}(j) = c_{\sigma}(j), \tag{20}$$

where $IN_j = \#\{i \leq j, \widetilde{\sigma}(i) = \widetilde{\sigma}(j)\}$. An example illustrating the extension of the Lehmer code to partial rankings is given in the Supplementary. However, computing each coordinate of the Lehmer code $c_{\sigma}(j)$ for any $j \in [\![K]\!]$ requires to sum over the $[\![K]\!]$ items. As an incomplete ranking do not involve the whole set of items, it is also tricky to extend the Lehmer code to map incomplete rankings.

Taking as input partial or incomplete rankings only modifies Step 1 of our method since it corresponds to the mapping step of the training data, and in Step 2 we still predict a full ranking. Extending our method to the task of predicting as output a partial or incomplete ranking raises several mathematical questions that we did not develop at length here because of space limitations. For instance, to predict partial rankings, a naive approach would consist in predicting a full ranking and then converting it to a partial ranking according to some threshold (i.e, keep the top-k items of the full ranking). A more formal extension of our method to make it able to predict directly partial rankings as outputs would require to optimize a metric tailored for this data and which could be written as in Eq. (2). A possibility for future work could be to consider the extension of the Kendall's τ distance with penalty parameter p for partial rankings proposed in Fagin et al. (2004).

5 Computational and theoretical analysis

5.1 Theoretical guarantees

In this section, we give some statistical guarantees for the estimators obtained by following the steps described in section 3. To this end, we build upon recent results in the framework of Surrogate Least Square by Ciliberto et al. (2016). Consider one of the embeddings ϕ on permutations presented in the previous section, which defines a loss Δ as in Eq. (2). Let $c_{\phi} = \max_{\sigma \in \mathfrak{S}_K} \|\phi(\sigma)\|$. We will denote by s^* a minimizer of the true risk (1), g^* a minimizer of the surrogate risk (3), and d a decoding function as (8)[3]. Given an estimator $\widehat{g}$ of g^* from Step 1, i.e. a minimizer of the empirical surrogate risk (6) we can then consider in Step 2 an estimator $\widehat{s} = d \circ \widehat{g}$. The following theorem reveals how the performance of the estimator $\widehat{s}$ we propose can be related to a solution s^* of (1) for the considered embeddings.

[3]Note that $d = \phi_L^{-1} \circ d_L$ for ϕ_L and is obtained as the composition of two steps for ϕ_τ and ϕ_H: solving an optimization problem and compute the inverse of the embedding.

Embedding	Step 1 (a)	Step 2 (b)
ϕ_τ	$\mathcal{O}(K^2N)$	NP-hard
ϕ_H	$\mathcal{O}(KN)$	$\mathcal{O}(K^3N)$
ϕ_L	$\mathcal{O}(KN)$	$\mathcal{O}(KN)$

Regressor	Step 1 (b)	Step 2 (a)
kNN	$\mathcal{O}(1)$	$\mathcal{O}(Nm)$
Ridge	$\mathcal{O}(N^3)$	$\mathcal{O}(Nm)$

Table 1: Embeddings and regressors complexities.

Theorem 1 *The excess risks of the proposed predictors are linked to the excess surrogate risks as:*

(i) For the loss (2) *defined by the Kemeny and Hamming embedding ϕ_τ and ϕ_H respectively:*

$$\mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(s^*) \leq c_\phi \sqrt{\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)}$$

with $c_{\phi_\tau} = \sqrt{\frac{K(K-1)}{2}}$ and $c_{\phi_H} = \sqrt{K}$.

(ii) For the loss (2) *defined by the Lehmer embedding ϕ_L:*

$$\mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(s^*) \leq \sqrt{\frac{K(K-1)}{2}} \sqrt{\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)} + \mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$$

The full proof is given in the Supplementary. Assertion (i) is a direct application of Theorem 2 in Ciliberto et al. (2016). In particular, it comes from a preliminary consistency result which shows that $\mathcal{E}(d \circ g^*) = \mathcal{E}(s^*)$ for both embeddings. Concerning the Lehmer embedding, it is not possible to apply their consistency results immediately; however a large part of the arguments of their proof is used to bound the estimation error for the surrogate risk, and we remain with an approximation error $\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$ resulting in Assertion (ii). In Remark 2 in the Supplementary, we give several insights about this approximation error. Firstly we show that it can be upper bounded by $2\sqrt{2}\sqrt{K(K-1)}\mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$. Then, we explain how this term results from using ϕ_L in the learning procedure. The Lehmer embedding thus have weaker statistical guarantees, but has the advantage of being more computationnally efficient, as we explain in the next subsection.

Notice that for Step 1, one can choose a consistent regressor with vector values $\widehat{g}$, i.e such that $\mathcal{R}(\widehat{g}) \to \mathcal{R}(g^*)$ when the number of training points tends to infinity. Examples of such methods that we use in our experiments to learn $\widehat{g}$, are the k-nearest neighbors (kNN) or kernel ridge regression (Micchelli and Pontil, 2005) methods whose consistency have been proved (see Chapter 5 in Devroye et al. (2013) and Caponnetto and De Vito (2007)). In this case the control of the excess of the surrogate risk $\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)$ implies the control of $\mathcal{E}(\widehat{s}) - \mathcal{E}(s^*)$ where $\widehat{s} = d \circ \widehat{g}$ by Theorem 1.

Remark 1 *We clarify that the consistency results of Theorem 1 are established for the task of predicting full rankings which is adressed in this paper. In the case of predicting partial or incomplete rankings, these results are not guaranteed to hold. Providing theoretical guarantees for this task is left for future work.*

5.2 Algorithmic complexity

We now discuss the algorithmic complexity of our approach. We recall that K is the number of items/labels whereas N is the number of samples in the dataset. For a given embedding ϕ, the total complexity of our approach for learning decomposes as follows. Step 1 in Section 3 can be decomposed in two steps: a preprocessing step (Step 1 (a)) consisting in mapping the training sample $\{(x_i, \sigma_i), i = 1, \ldots, N\}$ to $\{(x_i, \phi(\sigma_i)), i = 1, \ldots, N\}$, and a second step (Step 1 (b)) that consists in computing the estimator $\widehat{g}$ of the Least squares surrogate empirical minimization (6). Then, at prediction time, Step 2 Section 3 can also be decomposed in two steps: a first one consisting in mapping new inputs to a Hilbert space using $\widehat{g}$ (Step 2 (a)), and then solving the preimage problem (7) (Step 2 (b)). The complexity of a predictor corresponds to the worst complexity across all steps. The complexities resulting from the choice of an embedding and a regressor are summarized Table 1, where we denoted by m the dimension of the ranking embedded representations. The Lehmer embedding with kNN regressor thus provides the fastest theoretical complexity of $\mathcal{O}(KN)$ at the cost of weaker theoretical guarantees. The fastest methods previously proposed in the litterature typically involved a sorting procedure at prediction Cheng et al. (2010) leading to a $\mathcal{O}(NKlog(K))$ complexity. In the experimental section we compare our approach with the former (denoted as Cheng

PL), but also with the label wise decomposition approach in Cheng and Hüllermeier (2013) (Cheng LWD) involving a kNN regression followed by a projection on $\mathfrak{S}_K$ computed in $\mathcal{O}(K^3N)$, and the more recent Random Forest Label Ranking (Zhou RF) Zhou and Qiu (2016). In their analysis, if $d_{\mathcal{X}}$ is the size of input features and $D_{\max}$ the maximum depth of a tree, then RF have a complexity in $\mathcal{O}(D_{\max} d_{\mathcal{X}} K^2 N^2)$.

6 Numerical Experiments

Finally we evaluate the performance of our approach on standard benchmarks. We present the results obtained with two regressors : Kernel Ridge regression (Ridge) and k-Nearest Neighbors (kNN). Both regressors were trained with the three embeddings presented in Section 4. We adopt the same setting as Cheng et al. (2010) and report the results of our predictors in terms of mean Kendall's τ:

$$k_\tau = \frac{C - D}{K(K-1)/2} \quad \begin{cases} C : \text{number of concordant pairs between 2 rankings} \\ D : \text{number of discordant pairs between 2 rankings} \end{cases}, \tag{21}$$

from five repetitions of a ten-fold cross-validation (c.v.). Note that k_τ is an affine transformation of the Kendall's tau distance Δ_τ mapping on the $[-1, 1]$ interval. We also report the standard deviation of the resulting scores as in Cheng and Hüllermeier (2013). The parameters of our regressors were tuned in a five folds inner c.v. for each training set. We report our parameter grids in the supplementary materials.

Table 2: Mean Kendall's τ coefficient on benchmark datasets

	authorship	glass	iris	vehicle	vowel	wine
kNN Hamming	0.01±0.02	0.08±0.04	-0.15±0.13	-0.21±0.04	0.24±0.04	-0.36±0.04
kNN Kemeny	**0.94**±0.02	0.85±0.06	0.95±0.05	0.85±0.03	0.85±0.02	0.94±0.05
kNN Lehmer	0.93±0.02	0.85±0.05	0.95±0.04	0.84±0.03	0.78±0.03	0.94±0.06
ridge Hamming	-0.00±0.02	0.08±0.05	-0.10±0.13	-0.21±0.03	0.26±0.04	-0.36±0.03
ridge Lehmer	0.92±0.02	0.83±0.05	**0.97**±0.03	0.85±0.02	0.86±0.01	0.84±0.08
ridge Kemeny	**0.94**±0.02	0.86±0.06	**0.97**±0.05	**0.89**±0.03	**0.92**±0.01	0.94±0.05
Cheng PL	**0.94**±0.02	0.84±0.07	0.96±0.04	0.86±0.03	0.85±0.02	**0.95**±0.05
Cheng LWD	0.93±0.02	0.84±0.08	0.96±0.04	0.85±0.03	0.88±0.02	0.94±0.05
Zhou RF	0.91	**0.89**	**0.97**	0.86	0.87	**0.95**

The Kemeny and Lehmer embedding based approaches are competitive with the state of the art methods on these benchmarks datasets. The Hamming based methods give poor results in terms of k_τ but become the best choice when measuring the mean Hamming distance between predictions and ground truth (see Table 3 in the Supplementary). In contrast, the fact that the Lehmer embedding performs well for the optimization of the Kendall's τ distance highlights its practical relevance for label ranking. The Supplementary presents additional results (on additional datasets and results in terms of Hamming distance) which show that our method remains competitive with the state of the art. The code to reproduce our results is available: `https://github.com/akorba/Structured_Approach_Label_Ranking/`

7 Conclusion

This paper introduces a novel framework for label ranking, which is based on the theory of Surrogate Least Square problem for structured prediction. The structured prediction approach we propose comes along with theoretical guarantees and efficient algorithms, and its performance has been shown on real-world datasets. To go forward, extensions of our methodology to predict partial and incomplete rankings are to be investigated. In particular, the framework of prediction with abstention should be of interest.

References

Aiguzhinov, A., Soares, C., and Serra, A. P. (2010). A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In *International Conference on Discovery Science*, pages 16–26. Springer.

Ailon, N. (2010). Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*, 57(2):284–300.

Aledo, J. A., Gámez, J. A., and Molina, D. (2017). Tackling the supervised label ranking problem by bagging weak learners. *Information Fusion*, 35:38–50.

Brazdil, P. B., Soares, C., and Da Costa, J. P. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277.

Brouard, C., Szafranski, M., and d?Alché Buc, F. (2016). Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17(176):1–48.

Calauzenes, C., Usunier, N., and Gallinari, P. (2012). On the (non-) existence of convex, calibrated surrogate losses for ranking. In *Advances in Neural Information Processing Systems*, pages 197–205.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th Annual International Conference on Machine learning (ICML-07)*, pages 129–136. ACM.

Caponnetto, A. and De Vito, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368.

Cheng, W., Hühn, J., and Hüllermeier, E. (2009). Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML-09)*, pages 161–168. ACM.

Cheng, W. and Hüllermeier, E. (2013). A nearest neighbor approach to label ranking based on generalized labelwise loss minimization.

Cheng, W., Hüllermeier, E., and Dembczynski, K. J. (2010). Label ranking methods based on the plackett-luce model. In *Proceedings of the 27th Annual International Conference on Machine Learning (ICML-10)*, pages 215–222.

Chiang, T.-H., Lo, H.-Y., and Lin, S.-D. (2012). A ranking-based knn approach for multi-label classification. In *Asian Conference on Machine Learning*, pages 81–96.

Ciliberto, C., Rosasco, L., and Rudi, A. (2016). A consistent regularization approach for structured prediction. In *Advances in Neural Information Processing Systems*, pages 4412–4420.

Clémençon, S., Korba, A., and Sibony, E. (2017). Ranking median regression: Learning to order through local consensus. *arXiv preprint arXiv:1711.00070*.

Cortes, C., Mohri, M., and Weston, J. (2005). A general regression technique for learning transductions. In *Proceedings of the 22nd Annual International Conference on Machine learning (ICML-05)*, pages 153–160.

de Sá, C. R., Azevedo, P., Soares, C., Jorge, A. M., and Knobbe, A. (2018). Preference rules for label ranking: Mining patterns in multi-target relations. *Information Fusion*, 40:112–125.

Dekel, O., Singer, Y., and Manning, C. D. (2004). Log-linear models for label ranking. In *Advances in neural information processing systems*, pages 497–504.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Deza, M. and Deza, E. (2009). *Encyclopedia of Distances*. Springer.

Djuric, N., Grbovic, M., Radosavljevic, V., Bhamidipati, N., and Vucetic, S. (2014). Non-linear label ranking for large-scale prediction of long-term user interests. In *AAAI*, pages 1788–1794.

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2004). Comparing and aggregating rankings with ties. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58. ACM.

Fathony, R., Behpour, S., Zhang, X., and Ziebart, B. (2018). Efficient and consistent adversarial bipartite matching. In *International Conference on Machine Learning*, pages 1456–1465.

Fürnkranz, J. and Hüllermeier, E. (2003). Pairwise preference learning and ranking. In *European conference on machine learning*, pages 145–156. Springer.

Geng, X. and Luo, L. (2014). Multilabel ranking with inconsistent rankers. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3742–3747. IEEE.

Gurrieri, M., Siebert, X., Fortemps, P., Greco, S., and Słowiński, R. (2012). Label ranking: A new rule-based label ranking method. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 613–623. Springer.

Jiao, Y., Korba, A., and Sibony, E. (2016). Controlling the distance to a kemeny consensus without computing it. In *Proceedings of the 33rd Annual International Conference on Machine learning (ICML-16)*, pages 2971–2980.

Kadri, H., Ghavamzadeh, M., and Preux, P. (2013). A generalized kernel approach to structured output learning. In *Proceedings of the 30th Annual International Conference on Machine learning (ICML-13)*, pages 471–479.

Kamishima, T., Kazawa, H., and Akaho, S. (2010). A survey and empirical comparison of object ranking methods. In *Preference learning*, pages 181–201. Springer.

Kenkre, S., Khan, A., and Pandit, V. (2011). On discovering bucket orders from preference data. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 872–883. SIAM.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97.

Li, P., Mazumdar, A., and Milenkovic, O. (2017). Efficient rank aggregation via lehmer codes. *arXiv preprint arXiv:1701.09083*.

Mareš, M. and Straka, M. (2007). Linear-time ranking of permutations. In *European Symposium on Algorithms*, pages 187–193. Springer.

Merlin, V. R. and Saari, D. G. (1997). Copeland method ii: Manipulation, monotonicity, and paradoxes. *Journal of Economic Theory*, 72(1):148–172.

Micchelli, C. A. and Pontil, M. (2005). Learning the kernel function via regularization. *Journal of machine learning research*, 6(Jul):1099–1125.

Myrvold, W. and Ruskey, F. (2001). Ranking and unranking permutations in linear time. *Information Processing Letters*, 79(6):281–284.

Nowozin, S. and Lampert, C. H. (2011). Structured learning and prediction in computer vision. *Found. Trends. Comput. Graph. Vis.*, 6(3:8211;4):185–365.

Osokin, A., Bach, F. R., and Lacoste-Julien, S. (2017). On structured prediction theory with calibrated convex surrogate losses. In *Advances in Neural Information Processing Systems (NIPS) 2017*, pages 301–312.

Ramaswamy, H. G., Agarwal, S., and Tewari, A. (2013). Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *Advances in Neural Information Processing Systems*, pages 1475–1483.

Sá, C. R., Soares, C. M., Knobbe, A., and Cortez, P. (2017). Label ranking forests.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Springer.

Vembu, S. and Gärtner, T. (2010). Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer.

Wang, D., Mazumdar, A., and Wornell, G. W. (2015). Compression in the space of permutations. *IEEE Transactions on Information Theory*, 61(12):6417–6431.

Wang, Q., Wu, O., Hu, W., Yang, J., and Li, W. (2011). Ranking social emotions by learning listwise preference. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 164–168. IEEE.

Yu, P. L. H., Wan, W. M., and Lee, P. H. (2010). *Preference Learning*, chapter Decision tree modelling for ranking data, pages 83–106. Springer, New York.

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.

Zhou, Y., Liu, Y., Yang, J., He, X., and Liu, L. (2014). A taxonomy of label ranking algorithms. *JCP*, 9(3):557–565.

Zhou, Y. and Qiu, G. (2016). Random forest for label ranking. *arXiv preprint arXiv:1608.07710*.

8 Supplementary material

8.1 Proof of Theorem 1

We borrow the notations of Ciliberto et al. (2016) and recall their main result Theorem 2. They firstly exhibit the following assumption for a given loss Δ, see Assumption 1 therein:

Assumption 1. There exists a separable Hilbert space $\mathcal{F}$ with inner product $\langle ., . \rangle_{\mathcal{F}}$, a continuous embedding $\psi : \mathcal{Y} \to \mathcal{F}$ and a bounded linear operator $V : \mathcal{F} \to \mathcal{F}$, such that:

$$\Delta(y, y') = \langle \psi(y), V\psi(y') \rangle_{\mathcal{F}} \quad \forall y, y' \in \mathcal{Y} \tag{22}$$

Theorem 2 *Let $\Delta : \mathcal{Y} \to \mathcal{Y}$ satisfying Assumption 1 with $\mathcal{Y}$ a compact set. Then, for every measurable $g : \mathcal{X} \to \mathcal{F}$ and $d : \mathcal{F} \to \mathcal{Y}$ such that $\forall h \in \mathcal{F}$, $d(h) = \operatorname{argmin}_{y \in \mathcal{Y}} \langle \phi(y), h \rangle_{\mathcal{F}}$, the following holds:*

(i) Fisher Consistency: $\mathcal{E}(d \circ g^*) = \mathcal{E}(s^*)$

(ii) Comparison Inequality: $\mathcal{E}(d \circ g) - \mathcal{E}(s^*) \leq 2c_\Delta \sqrt{\mathcal{R}(g) - \mathcal{R}(g^*)}$

with $c_\Delta = \|V\| \max_{y \in \mathcal{Y}} \|\phi(y)\|$.

Notice that any discrete set $\mathcal{Y}$ is compact and $\phi : \mathcal{Y} \to \mathcal{F}$ is continuous. We now prove the two assertions of Theorem 1.

Proof of Assertion(i) in Theorem 1. Firstly, $\mathcal{Y} = \mathfrak{S}_K$ is finite. Then, for the Kemeny and Hamming embeddings, Δ satisfies Assumption 1 with $V = -id$ (where id denotes the identity operator) , and $\psi = \phi_K$ and $\psi = \phi_H$ respectively. Theorem 2 thus applies directly.

Proof of Assertion(ii) in Theorem 1. In the following proof, $\mathcal{Y}$ denotes $\mathfrak{S}_K$, ϕ denotes ϕ_L and $d = \phi_L^{-1} \circ d_L$ with d_L as defined in (17). Our goal is to control the excess risk $\mathcal{E}(s) - \mathcal{E}(s^*)$.

$$\begin{aligned}
\mathcal{E}(s) - \mathcal{E}(s^*) &= \mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(s^*) \\
&= \underbrace{\mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(d \circ g^*)}_{(A)} + \underbrace{\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*)}_{(B)}
\end{aligned}$$

Consider the first term (A).

$$\begin{aligned}
\mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(d \circ g^*) &= \int_{\mathcal{X} \times \mathcal{Y}} \Delta(d \circ \widehat{g}(x), \sigma) - \Delta(d \circ g^*(x), \sigma) dP(x, \sigma) \\
&= \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(d \circ \widehat{g}(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 dP(x, \sigma) \\
&= \underbrace{\int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x))\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x))\|_{\mathcal{F}}^2 dP(x)}_{(A1)} + \\
&\qquad \underbrace{2 \int_{\mathcal{X}} \langle \phi(d \circ g^*(x)) - \phi(d \circ \widehat{g}(x)), \int_{\mathcal{Y}} \phi(\sigma) dP(\sigma, x) \rangle dP(x)}_{(A2)}
\end{aligned}$$

The first term (A1) can be upper bounded as follows:

$$\begin{aligned}
\int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x))\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x))\|_{\mathcal{F}}^2 dP(x) &\leq \int_{\mathcal{X}} \langle \phi(d \circ \widehat{g}(x)) - \phi(d \circ g^*(x)), \phi(d \circ \widehat{g}(x)) + \phi(d \circ g^*(x)) \rangle_{\mathcal{F}} dP(x) \\
&\leq 2c_\Delta \int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x)) - \phi(d \circ g^*(x))\|_{\mathcal{F}} dP(x) \\
&\leq 2c_\Delta \sqrt{\int_{\mathcal{X}} \|d_L(\widehat{g}(x)) - d_L(g^*(x))\|_{\mathcal{F}}^2 dP(x)} \\
&\leq 2c_\Delta \sqrt{\int_{\mathcal{X}} \|g^*(x) - \widehat{g}(x)\|_{\mathcal{F}}^2 dP(x)} + \mathcal{O}(K\sqrt{K})
\end{aligned}$$

with $c_\Delta = \max_{\sigma \in \mathcal{Y}} \|\phi(\sigma)\|_{\mathcal{F}} = \sqrt{\frac{(K-1)(K-2)}{2}}$ and since $\|d_L(u) - d_L(v)\| \leq \|u - v\| + \sqrt{K}$. Since $\int_{\mathcal{X}} \|g^*(x) - \widehat{g}(x)\|_{\mathcal{F}}^2 dP(x) = \mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)$ (see Ciliberto et al. (2016)) we get the first term of Assertion (i).

For the second term (A2), we can actually follow the proof of Theorem 12 in Ciliberto et al. (2016) and we get:

$$\int_{\mathcal{X}} \langle \phi(d \circ g^*(x)) - \phi(d \circ \widehat{g}(x)), \int_{\mathcal{Y}} \phi(\sigma) dP(\sigma, x) \rangle dP(x) \leq 2c_\Delta \sqrt{\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)}$$

Consider the second term (2). By Lemma 8 in (Ciliberto et al., 2016), we have that:

$$g^*(x) = \int_{\mathcal{Y}} \phi(\sigma) dP(\sigma|x) \tag{23}$$

and then:

$$\begin{aligned}
\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) &= \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(d \circ g^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 - \|\phi(s^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 dP(x, \sigma) \\
&\leq \int_{\mathcal{X} \times \mathcal{Y}} \langle \phi(d \circ \widehat{g}(x)) - \phi(s^*(x)), \phi(d \circ \widehat{g}(x)) + \phi(s^*(x)) - 2\phi(\sigma) \rangle_{\mathcal{F}} dP(x, \sigma) \\
&\leq 4c_\Delta \int_{\mathcal{X}} \|\phi(d \circ g^*(x)) - \phi(s^*(x))\|_{\mathcal{F}} dP(x) \\
&\leq 4c_\Delta \int_{\mathcal{X}} \|d_L \circ g^*(x)) - d_L \circ \phi(s^*(x))\|_{\mathcal{F}} dP(x) \\
&\leq 4c_\Delta \int_{\mathcal{X}} \|g^*(x)) - \phi(s^*(x))\|_{\mathcal{F}} dP(x) + \mathcal{O}(K\sqrt{K})
\end{aligned}$$

where we used that $\phi(s^*(x)) \in \mathcal{C}_K$ so $d_L \circ \phi(s^*(x)) = \phi(s^*(x))$. Then we can plug (23) in the right term:

$$\begin{aligned}
\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) &\leq 4c_\Delta \int_{\mathcal{X}} \| \int_{\mathcal{Y}} \phi(\sigma) dP(\sigma|x) - \phi(s^*(x)) \|_{\mathcal{F}} dP(x) + \mathcal{O}(K\sqrt{K}) \\
&\leq 4c_\Delta \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(\sigma) - \phi(s^*(x))\|_{\mathcal{F}} dP(x) + \mathcal{O}(K\sqrt{K}) \\
&\leq 4c_\Delta \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})
\end{aligned}$$

Remark 2 *As proved in Theorem 19 in (Ciliberto et al., 2016), since the space of rankings $\mathcal{Y}$ is finite, Δ_L necessarily satisfies Assumption 1 with some continuous embedding ψ. If the approach we developped was relying on this ψ, we would have consistency for the minimizer g^* of the Lehmer loss* (16)*. However, the choice of ϕ_L is relevant because it yields a pre-image problem with low computational complexity.*

8.2 Lehmer embedding for partial rankings

An example, borrowed from (Li et al., 2017) illustrating the extension of the Lehmer code for partial rankings is the following:

e	1	2	3	4	5	6	7	8	9
$\widetilde{\sigma}$	1	1	2	2	3	1	2	3	3
σ	1	2	4	5	7	3	6	8	9
c_σ	0	0	0	0	0	3	1	0	0
IN	1	2	1	2	1	3	3	2	3
$c_{\widetilde{\sigma}}$	0	0	0	0	0	3	1	0	0
$c'_{\widetilde{\sigma}}$	0	1	0	1	0	5	3	1	2

where each row represents a step to encode a partial ranking.

8.3 Additional experimental results

Details concerning the parameter grids. We first recall our notations for vector valued kernel ridge regression. Let $\mathcal{H}_K$ be a vector-valued Reproducing Kernel Hilbert Space associated to an operator-valued kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathbb{R}^n)$. Solve:

$$\min_{g \in \mathcal{H}_K} \sum_{k=1}^{N} \|g(x_k) - \phi(\sigma_k)\|^2 + \lambda \|h\|_{\mathcal{H}_K}^2 \tag{24}$$

The solution of this problem is unique and admits an expansion: $\widehat{g}(.) = \sum_{i=1}^{N} K(x_i, .)c_i$ (see Micchelli and Pontil (2005)). Moreover, it has the following closed-form solution:

$$\widehat{g}(.) = \psi_x(.)(K_x + \lambda I_N)^{-1} Y_N \tag{25}$$

where K_x is the $N \times N$ block-matrix, with each block of the form $K(x_k, x_l)$, Y_N is the vector of all stacked vectors $\phi(\sigma_1), \ldots, \phi(\sigma_N)$, and ψ_x is the matrix composed of $[K(., x_1), \ldots, K(., x_N)]$. In all our experiments, we used a decomposable gaussian kernel $K(x, y) = \exp(-\gamma\|x - y\|^2)I_m$. The bandwith γ and the regularization parameter λ were chosen in the set $\{10^{-i}, 5 \cdot 10^{-i}\}$ for $i \in 0, \ldots, 5$ during the gridsearch cross-validation steps. For the k-Nearest Neighbors experiments, we used the euclidean distance and the neighborhood size was chosen in the set $\{1, 2, 3, 4, 5, 8, 10, 15, 20, 30, 50\}$.

Experimental results. We report additional results in terms of rescaled Hamming distance ($d_{H_K}(\sigma, \sigma') = \frac{d_H(\sigma,\sigma')}{K^2}$) on the datasets presented in the paper and in terms of Kendall's τ coefficient on other datasets. All the results have been obtained in the same experimental conditions: ten folds cross-validation are repeated five times with the parameters tuned in a five folds inner cross-validation. The results presented in Table 3 correspond to the mean normalized Hamming distance between the prediction and the ground truth (lower is better). Whereas Hamming based embeddings led to very low results on the task measured using the Kendall's τ coefficient, they outperform other embeddings for the Hamming distance minimization problem as expected.

Table 3: rescaled Hamming distance

	authorship	glass	iris	vehicle	vowel	wine
kNN Kemeny	0.05±0.01	0.07±0.02	0.04±0.03	0.08±0.01	0.07±0.01	0.04±0.03
kNN Lehmer	0.05±0.01	0.08±0.02	0.03±0.03	0.10±0.01	0.10±0.01	0.04±0.03
kNN Hamming	0.05±0.01	0.08±0.02	0.03±0.03	0.08±0.02	0.07±0.01	0.04±0.03
ridge Kemeny	0.06±0.01	0.08±0.03	0.04±0.03	0.08±0.01	0.08±0.01	0.04±0.03
ridge Lehmer	0.05±0.01	0.09±0.03	**0.02**±0.02	0.10±0.01	0.08±0.01	0.09±0.04
ridge Hamming	**0.04**±0.01	**0.06**±0.02	**0.02**±0.02	**0.07**±0.01	**0.05**±0.01	**0.04**±0.02

In Table (4), we show that Lehmer and Hamming based embeddings stay competitive on other standard benchmark datasets. The Ridge results have not been reported due to scalability issues as the number of inputs elements and the output space size grow.

Table 4: Kendall's τ coefficient on additional datasets

	bodyfat	calhousing	cpu-small	pendigits	segment	wisconsin	fried	sushi
kNN Lehmer	**0.23**±0.01	0.22±0.01	0.40±0.01	**0.94**±0.00	0.95±0.01	**0.49**±0.00	0.85±0.02	0.17±0.01
kNN Kemeny	**0.23**±0.06	0.33±0.01	**0.51**±0.00	**0.94**±0.00	0.95±0.01	**0.49**±0.04	0.89±0.00	0.31±0.01
Cheng PL	**0.23**	0.33	0.50	**0.94**	0.95	0.48	0.89	**0.32**
Zhou RF	0.185	**0.37**	**0.51**	**0.94**	**0.96**	0.48	**0.93**	–

On the sushi dataset Kamishima et al. (2010), we additionally tested our approach Ridge Kemeny which obtained the same results as Cheng PL (**0.32** Kendall's τ).