A Structured Prediction Approach for Label Ranking

Anna Korba [1,2] Alexandre Garcia [1] Florence d'Alché-Buc [1]

[1] LTCI, Télécom ParisTech, Université Paris-Saclay

[2] Gatsby Unit, CSML, University College London

LAMSADE, Dauphine

15 Février 2019

Outline

1. Introduction to ranking data
2. Label ranking
3. Structured prediction for label ranking
4. Ranking embeddings
5. Computational and theoretical analysis
6. Openings and conclusion

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

Openings and conclusion

What is ranking data?

Consider a set of items $[\![K]\!] := \{1, \ldots, K\}$.

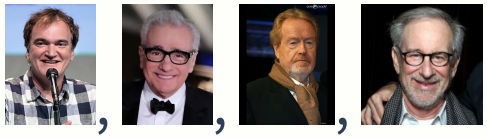
A ranking is an **ordered list** (of any size) **of items** of $[\![K]\!]$

What is ranking data?

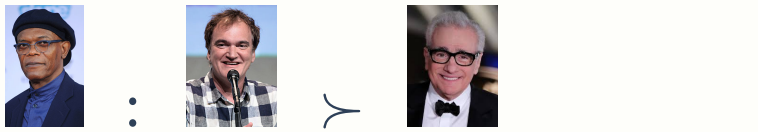
Consider a set of items $[\![K]\!] := \{1, \dots, K\}$.

A ranking is an **ordered list** (of any size) **of items** of $[\![K]\!]$

Example:

$[\![4]\!] := \{1, 2, 3, 4\}$=  .

Ask an actor to rank/order them by preference ($\succ$):

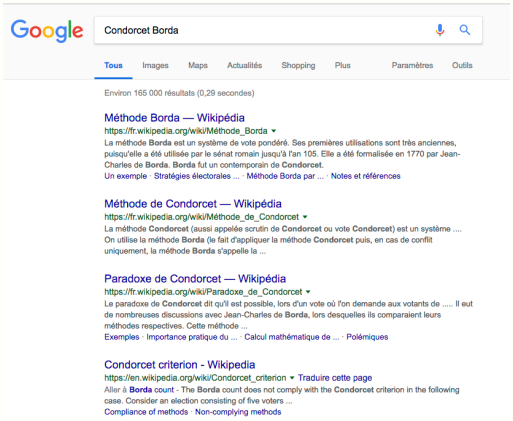


Many applications involve rankings/comparisons

- Modelling human preferences (elections, surveys, online implicit feedback)

$\Longrightarrow$ easier for an individual to rank than to rate

- Computer systems (search engines, recommendation systems)



- Other (competitions, biological data...)

Analysis of full rankings

Set of items $[\![K]\!] := \{1, \dots, K\}$. Ex: $\{1, 2, 3, 4\}$

- An individual expresses her preferences as a **full** ranking, i.e a strict order $\succ$ over the whole set $[\![K]\!]$:

$$a_1 \succ a_2 \succ \cdots \succ a_K$$

Other kind of rankings: **Top-k rankings**: $a_1, \dots, a_k \succ$ the rest, **Pairwise comparisons**: $a_1 \succ a_2$

Analysis of full rankings

Set of items $[\![K]\!] := \{1, \dots, K\}$. Ex: $\{1, 2, 3, 4\}$

- An individual expresses her preferences as a **full** ranking, i.e a strict order $\succ$ over the whole set $[\![K]\!]$:

$$a_1 \succ a_2 \succ \cdots \succ a_K$$

Other kind of rankings: **Top-k rankings**: $a_1, \dots, a_k \succ$ the rest, **Pairwise comparisons**: $a_1 \succ a_2$

A full ranking can be seen as the permutation σ that maps an item to its rank:

$$a_1 \succ \cdots \succ a_K \quad \Leftrightarrow \quad \sigma \in \mathfrak{S}_K \text{ such that } \sigma(a_i) = i$$

$$2 \succ 1 \succ 3 \succ 4 \quad \Leftrightarrow \quad \sigma = 2134 \quad (\sigma(2) = 1, \sigma(1) = 2, \dots)$$

Analysis of full rankings

Set of items $[\![K]\!] := \{1, \dots, K\}$. Ex: $\{1, 2, 3, 4\}$

- An individual expresses her preferences as a **full** ranking, i.e a strict order $\succ$ over the whole set $[\![K]\!]$:

$$a_1 \succ a_2 \succ \cdots \succ a_K$$

Other kind of rankings: **Top-k rankings**: $a_1, \dots, a_k \succ$ the rest, **Pairwise comparisons**: $a_1 \succ a_2$

A full ranking can be seen as the permutation σ that maps an item to its rank:

$$a_1 \succ \cdots \succ a_K \quad \Leftrightarrow \quad \sigma \in \mathfrak{S}_K \text{ such that } \sigma(a_i) = i$$

$$2 \succ 1 \succ 3 \succ 4 \quad \Leftrightarrow \quad \sigma = 2134 \quad (\sigma(2) = 1, \sigma(1) = 2, \dots)$$

Let $\mathfrak{S}_K$ be set of permutations of $[\![K]\!]$, the symmetric group.
Ex: $\mathfrak{S}_4 = 1234, 1324, 1423, \dots, 4321$

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

  but the random variables $\Sigma(1), \ldots, \Sigma(K)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

  but the random variables $\Sigma(1), \ldots, \Sigma(K)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!

  $\Rightarrow$ No natural notion of mean or variance for Σ

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

  but the random variables $\Sigma(1), \ldots, \Sigma(K)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!

  $\Rightarrow$ No natural notion of mean or variance for Σ

- The set of permutations $\mathfrak{S}_K$ is finite...

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

  but the random variables $\Sigma(1), \ldots, \Sigma(K)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!

  $\Rightarrow$ No natural notion of mean or variance for Σ

- The set of permutations $\mathfrak{S}_K$ is finite...

  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$

Analysis of ranking data: main challenges

Consider N individuals expressing their preferences on $[\![K]\!]$:
$\Longrightarrow$ results in a dataset of N rankings/permutations

$$\mathcal{D}_N = (\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- A random permutation $\Sigma \in \mathfrak{S}_K$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(K)) \in \mathbb{R}^K$...

  but the random variables $\Sigma(1), \ldots, \Sigma(K)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!

  $\Rightarrow$ No natural notion of mean or variance for Σ

- The set of permutations $\mathfrak{S}_K$ is finite...

  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$

  $\Rightarrow$ Little statistical relevance

Main approaches 1 - Parametric

- Choose a predefined generative model on the data and analyze the data through that model
  ([Lu and Boutilier, 2014, Zhao et al., 2016, Szörényi et al., 2015])

Main approaches 1 - Parametric

- Choose a predefined generative model on the data and analyze the data through that model
  ([Lu and Boutilier, 2014, Zhao et al., 2016, Szörényi et al., 2015])
  - Mallows [Mallows, 1957]
    Parameterized by a central ranking $\sigma_0 \in \mathfrak{S}_K$ and a dispersion parameter $\gamma \in \mathbb{R}^+$

    $$P(\sigma) = Ce^{-\gamma d(\sigma_0, \sigma)} \qquad \text{with } d \text{ a distance on } \mathfrak{S}_K.$$

  - Plackett-Luce [Luce, 1959]
    Each item i is parameterized by w_i with $w_i \in \mathbb{R}^+$:

    $$P(\sigma) = \prod_{i=1}^{K} \frac{w_{\sigma^{-1}(i)}}{\sum_{j=i}^{n} w_{\sigma^{-1}(j)}}$$

    Ex: $2 \succ 1 \succ 3 = \frac{w_2}{w_1+w_2+w_3} \frac{w_1}{w_1+w_3}$

Main approaches 1 - Parametric

- Choose a predefined generative model on the data and analyze the data through that model
  ([Lu and Boutilier, 2014, Zhao et al., 2016, Szörényi et al., 2015])
  - Mallows [Mallows, 1957]
    Parameterized by a central ranking $\sigma_0 \in \mathfrak{S}_K$ and a dispersion parameter $\gamma \in \mathbb{R}^+$

    $$P(\sigma) = Ce^{-\gamma d(\sigma_0, \sigma)} \qquad \text{with } d \text{ a distance on } \mathfrak{S}_K.$$

  - Plackett-Luce [Luce, 1959]
    Each item i is parameterized by w_i with $w_i \in \mathbb{R}^+$:

    $$P(\sigma) = \prod_{i=1}^{K} \frac{w_{\sigma^{-1}(i)}}{\sum_{j=i}^{n} w_{\sigma^{-1}(j)}}$$

    Ex: $2 \succ 1 \succ 3 = \frac{w_2}{w_1+w_2+w_3} \frac{w_1}{w_1+w_3}$

- may fail to hold on real data (see for instance [Davidson and Marschak, 1959, Tversky, 1972] on decision making)

Main approaches 2 -“Non Parametric”

- Choose a structure on $\mathfrak{S}_K$ and analyze the data with respect to that structure
  - Harmonic analysis ([Kondor and Barbosa, 2010, Clémençon et al., 2011, Sibony et al., 2015]
  - Kernel density smoothing [Sun et al., 2012]
  - Modeling of pairwise comparisons ([Jiang et al., 2011, Rajkumar and Agarwal, 2014, Shah and Wainwright, 2017])
  - Kernel methods [Jiao and Vert, 2015]...

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

Openings and conclusion

Label Ranking - A supervised learning problem

Now $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ i.i.d. copies of $(X,\ \Sigma)$

Label Ranking - A supervised learning problem

Now $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ i.i.d. copies of $(X,\ \Sigma)$

Ex: Users i with characteristics X_i and their produced rankings/preferences Σ_i.

Label Ranking - A supervised learning problem

Now $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ i.i.d. copies of $(X,\ \Sigma)$

Ex: Users i with characteristics X_i and their produced rankings/preferences Σ_i.

Goal: Learn a predictive ranking rule :

$$\begin{array}{ccccc} s & : & \mathcal{X} & \to & \mathfrak{S}_K \\ & & x & \mapsto & s(x) \end{array}$$

which given a random vector X, predicts the permutation Σ on the K items.

Example: targeted advertising domain

Risk minimization for label ranking

Goal: Learn a predictive ranking rule $s : \mathcal{X} \to \mathfrak{S}_K$ as:

$$\min_{s\,:\,\mathcal{X} \to \mathfrak{S}_K} \mathcal{R}(s), \quad \text{with } \mathcal{R}(s) = \mathbb{E}\left[\Delta\left(s(X), \Sigma\right)\right]$$

with Δ some loss function for rankings, e.g.:

- **Kendall's τ**:
  $\Delta_\tau(\sigma, \sigma') = \sum_{1 \leq i < j \leq K} \mathbb{I}[(\sigma(i) - \sigma(j))(\sigma'(i) - \sigma'(j)) < 0]$
  $\to$ Intuitive when rankings represent preferences

- **Hamming**: $\Delta_H(\sigma, \sigma') = \sum_{i=1}^{K} \mathbb{I}[\sigma(i) \neq \sigma'(i)]$.
  $\to$ Popular when rankings represent matchings/assignments

Related Work

- Can be seen as an extension of multiclass and multilabel classification
- Many applications, e.g : document categorization, meta-learning
  - rank a set of topics relevant for a given document
  - rank a set of algorithms according to their suitability for a new dataset, based on the characteristics of the dataset
- A lot of approaches rely on parametric modelling [Cheng and Hüllermeier, 2009], [Cheng et al., 2010]

Related Work

- Can be seen as an extension of multiclass and multilabel classification
- Many applications, e.g : document categorization, meta-learning
  - rank a set of topics relevant for a given document
  - rank a set of algorithms according to their suitability for a new dataset, based on the characteristics of the dataset
- A lot of approaches rely on parametric modelling [Cheng and Hüllermeier, 2009], [Cheng et al., 2010]

We develop an approach free of any parametric assumptions:
$\Longrightarrow$ relying on results and framework developped for **structured prediction**
$\Longrightarrow$ exploiting the geometry of well-chosen **feature maps** for rankings

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

Openings and conclusion

Structured prediction for label ranking

Goal: Learn a predictive ranking rule $s : \mathcal{X} \to \mathfrak{S}_K$ as:

$$\min_{s\,:\,\mathcal{X} \to \mathfrak{S}_K} \mathcal{R}(s), \quad \text{with } \mathcal{R}(s) = \mathbb{E}\left[\Delta\left(s(X), \Sigma\right)\right]$$

Main idea [Korba et al., 2018] : Consider a family of Δ loss functions:

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

with $\phi : \mathfrak{S}_K \to \mathcal{F}$ some ranking embedding, i.e. that maps the permutations $\sigma \in \mathfrak{S}_K$ into a Hilbert space $\mathcal{F}$ (e.g. $\mathbb{R}^d$ for $d \in \mathbb{N}$).

Motivation: There exist ϕ_τ, ϕ_H such that Δ_τ and Δ_H write as (**??**).

Structured prediction - surrogate problem

$$\min_{s\,:\,\mathcal{X}\to\mathfrak{S}_K} \mathcal{R}(s), \quad \text{with } \mathcal{R}(s) = \mathbb{E}\left[\|\phi(s(X)) - \phi(\Sigma)\|_{\mathcal{F}}^2\right] \qquad (1)$$

$\Rightarrow$ **Hard to optimize.**

Idea: Introduce a surrogate problem:

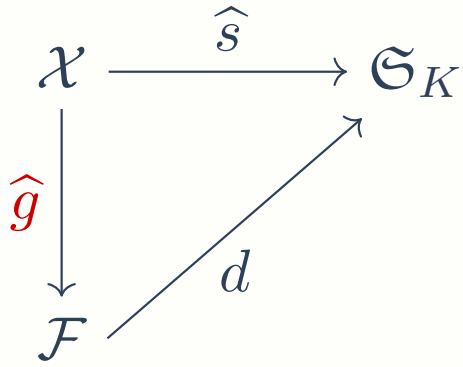
$$\min_{g\,:\,\mathcal{X}\to\mathcal{F}} \mathcal{L}(g), \quad \text{with} \quad \mathcal{L}(g) = \mathbb{E}\left[\|g(X) - \phi(\Sigma)\|_{\mathcal{F}}^2\right] \qquad (2)$$

$\Rightarrow$ **easier to optimize since g has values in $\mathcal{F}$**

Let s^* be a minimizer of (1) and g^* a minimizer of (2).

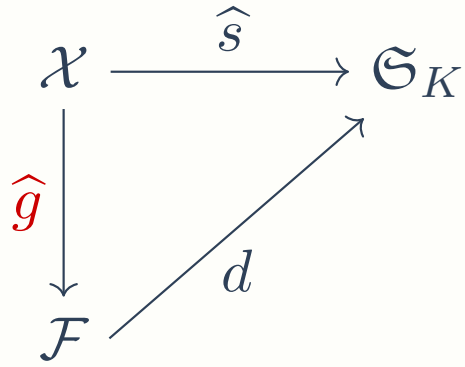
Structured Prediction Approach

We can thus approach structured prediction in **two steps**:
(see [Ciliberto et al., 2016, Brouard et al., 2016])



Structured Prediction Approach

We can thus approach structured prediction in **two steps**:
(see [Ciliberto et al., 2016, Brouard et al., 2016])



- **Step 1 (Regression)**: with any regression method (kNN, RF, Ridge regression...) $\Longrightarrow$ Output $\widehat{g} : \mathcal{X} \to \mathcal{F}$
- **Step 2 (Pre-image)**: for any $x \in \mathcal{X}$:

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

Consistency: $\mathcal{R}(d \circ g^*) = \mathcal{R}(s^*)$
$\Longrightarrow$ Choice of ϕ and regression method matter

Detailed method

Firstly pick a loss Δ ($\Leftrightarrow$ embedding ϕ)

- **Step 1 (Regression)**: Learn $\widehat{g} : \mathcal{X} \to \mathcal{F}$
  - Step 1 (a): map $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ to $\mathcal{D}'_N = (X_1, \phi(\Sigma_1)), \ldots, (X_N, \phi(\Sigma_N))$ where $\phi(\Sigma_i) \in \mathbb{R}^m$
  - Step 1 (b): Learn $\widehat{g}$ with any regressor

Detailed method

Firstly pick a loss Δ ($\Leftrightarrow$ embedding ϕ)

- **Step 1 (Regression)**: Learn $\widehat{g} : \mathcal{X} \to \mathcal{F}$
  - Step 1 (a): map $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ to $\mathcal{D}'_N = (X_1, \phi(\Sigma_1)), \ldots, (X_N, \phi(\Sigma_N))$ where $\phi(\Sigma_i) \in \mathbb{R}^m$
  - Step 1 (b): Learn $\widehat{g}$ with any regressor
- **Step 2 (Pre-image)**: $\forall x \in \mathcal{X}$:
  - Step 2 (a): Compute $\widehat{g}(x)$
  - Step 2 (b): Solve $\widehat{s}(x) = \text{argmin}_{\sigma \in \mathfrak{S}_K} \|\phi(\sigma) - \widehat{g}(x)\|^2_{\mathcal{F}}$

Detailed method

Firstly pick a loss Δ ($\Leftrightarrow$ embedding ϕ)

- **Step 1 (Regression)**: Learn $\widehat{g} : \mathcal{X} \to \mathcal{F}$
  - Step 1 (a): map $\mathcal{D}_N = (X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ to $\mathcal{D}'_N = (X_1, \phi(\Sigma_1)), \ldots, (X_N, \phi(\Sigma_N))$ where $\phi(\Sigma_i) \in \mathbb{R}^m$
  - Step 1 (b): Learn $\widehat{g}$ with any regressor
- **Step 2 (Pre-image)**: $\forall x \in \mathcal{X}$:
  - Step 2 (a): Compute $\widehat{g}(x)$
  - Step 2 (b): Solve $\widehat{s}(x) = \operatorname{argmin}_{\sigma \in \mathfrak{S}_K} \|\phi(\sigma) - \widehat{g}(x)\|^2_{\mathcal{F}}$

Choice of the embedding $\phi \Longrightarrow$ complexities of Step 1 (a) and 2 (b)
Choice of the regressor $\Longrightarrow$ complexities of Step 1 (b) and 2 (a)

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

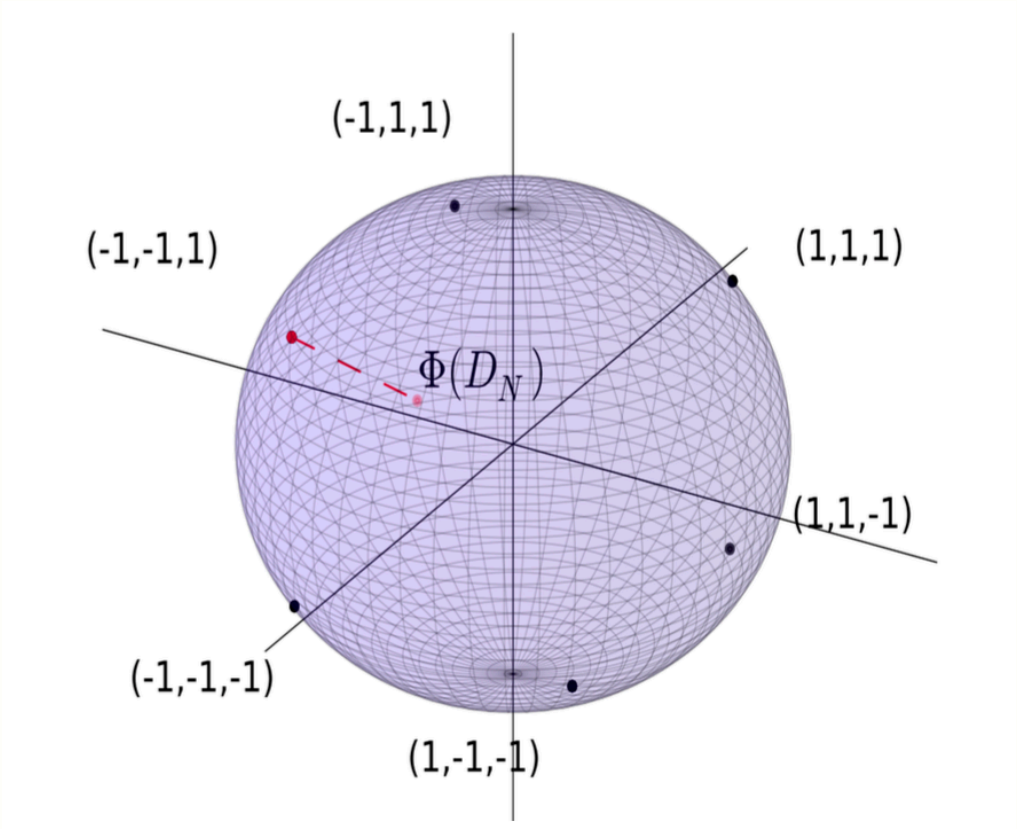
Openings and conclusion

Ranking embeddings proposed - 1

- Kemeny embedding ([Jiao and Vert, 2015, Jiao et al., 2016])

$$\phi_\tau : \mathfrak{S}_K \to \mathbb{R}^{K(K-1)/2}$$
$$\sigma \mapsto (\text{sign}(\sigma(j) - \sigma(i)))_{1 \le i < j \le K} \ .$$

Ex: $\sigma = 132 \Longrightarrow \phi_\tau(\sigma) = (1, 1, -1)$



Ranking embeddings proposed - 2

- Hamming embedding ([Plis et al., 2011])

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

Ex: $\sigma = 132 \Longrightarrow \phi_H(\sigma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

Ranking embeddings proposed - 2

- Hamming embedding ([Plis et al., 2011])

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

Ex: $\sigma = 132 \Longrightarrow \phi_H(\sigma) = \begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}$

- Lehmer embedding ([Li et al., 2017])

$$\begin{aligned}\phi_L : \mathfrak{S}_K &\to \mathbb{R}^{K}\\ \sigma &\mapsto (\#\{i : i<j, \sigma(i)>\sigma(j)\})_{j=1,\dots,K}\ ,\end{aligned}$$

"number of elements i with index smaller than j that are ranked higher than j in the permutation σ"

Ex: $\sigma = 132 \Longrightarrow \phi_L(\sigma) = (0,0,1)$
$\sigma = 321 \Longrightarrow \phi_L(\sigma) = (0,1,2)$
$Im(\phi_L) = \mathcal{C}_K = \{0\} \times [\![0,1]\!] \times [\![0,2]\!] \times \cdots \times [\![0,K-1]\!]$

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

Openings and conclusion

Computational analysis of the pre-image step - 2 (b)

Now suppose $\widehat{g}(x)$ is known (after the learning step).

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

For embeddings with constant norm ($\|\phi(\sigma)\| = C$ for any σ, e.g. Kemeny and Hamming), it can be rewritten:

$$\underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmax}} \langle \phi(\sigma), \widehat{g}(x) \rangle_{\mathcal{F}}$$

The solution comes in two steps:

1. Find the embedded object ϕ_σ in $Im(\phi) \subset \mathcal{F}$ which maximizes the linear program.
2. Invert the embedding to get σ

Pre-image for the Kemeny embedding

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 then the problem becomes.

$$\widehat{\phi_\sigma} = \underset{\phi_\sigma{}'}{\text{argmin}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi'_\sigma)_{i,j},$$

$$s.c. \quad \begin{cases} (\phi'_\sigma)_{i,j} \in \{-1, 1\} \quad \forall\, i, j \\ (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,i} = 0 \quad \forall\, i, j \\ -1 \leq (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,k} + (\phi'_\sigma)_{k,i} \leq 1 \quad \forall\, i, j, k \text{ s.t. } i \neq j \neq k. \end{cases}$$

Pre-image for the Kemeny embedding

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 then the problem becomes.

$$\widehat{\phi_\sigma} = \underset{\phi_\sigma{}'}{\text{argmin}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi'_\sigma)_{i,j},$$

$$s.c. \quad \begin{cases} (\phi'_\sigma)_{i,j} \in \{-1, 1\} \quad \forall\, i, j \\ (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,i} = 0 \quad \forall\, i, j \\ -1 \leq (\phi'_\sigma)_{i,j} + (\phi'_\sigma)_{j,k} + (\phi'_\sigma)_{k,i} \leq 1 \quad \forall\, i, j, k \text{ s.t. } i \neq j \neq k. \end{cases}$$

Minimal feedback Arc Set problem $\rightarrow$ NP-Hard

Pre-image for the Hamming embedding

Enforce the constraints of Hamming representations

$$\widehat{\phi_\sigma} = \underset{\phi_\sigma}{\text{argmax}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi_\sigma)_{i,j},$$

$$s.c \begin{cases} (\phi_\sigma)_{i,j} \in \{0,1\} \quad \forall\, i,j \\ \sum_i (\phi_\sigma)_{i,j} = \sum_j (\phi_\sigma)_{i,j} = 1 \quad \forall\, i,j \end{cases},$$

Pre-image for the Hamming embedding

Enforce the constraints of Hamming representations

$$\widehat{\phi_\sigma} = \underset{\phi_\sigma}{\text{argmax}} \sum_{1 \leq i,j \leq K} \widehat{g}(x)_{i,j} (\phi_\sigma)_{i,j},$$

$$s.c \begin{cases} (\phi_\sigma)_{i,j} \in \{0,1\} \quad \forall\, i,j \\ \sum_i (\phi_\sigma)_{i,j} = \sum_j (\phi_\sigma)_{i,j} = 1 \quad \forall\, i,j \end{cases},$$

$\Longrightarrow$ Bipartite graph matching problem.

Solved in $\mathcal{O}(K^3)$ with the Hungarian Algorithm.

Pre-image for the Lehmer Embedding

Recall: $\phi_L(\sigma) \in \mathcal{C}_K = \{0\} \times [\![0,1]\!] \times [\![0,2]\!] \times \cdots \times [\![0,K-1]\!]$, where for $j = 1, \ldots, K$:

$$\phi_L(\sigma)(j) = \#\{i : i < j, \sigma(i) > \sigma(j)\}$$

number of elements i with index smaller than j that are ranked higher than j in the permutation σ.

Pre-image for the Lehmer Embedding

Recall: $\phi_L(\sigma) \in \mathcal{C}_K = \{0\} \times [\![0,1]\!] \times [\![0,2]\!] \times \cdots \times [\![0,K-1]\!]$, where for $j = 1, \ldots, K$:

$$\phi_L(\sigma)(j) = \#\{i : i < j, \sigma(i) > \sigma(j)\}$$

number of elements i with index smaller than j that are ranked higher than j in the permutation σ.

The decoupled coordinates enable a trivial solving of the pre-image problem:

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

where d is the global decoding function.

Theoretical guarantees

For Kemeny and Hamming embedding:

- **consistency holds**: $\mathcal{R}(d \circ g^*) = \mathcal{R}(s^*)$ and:

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

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

- but the **pre-image step is hard** : NP-hard for Kemeny, $\mathcal{O}(K^3)$ for Hamming (K=number of labels)

Theoretical guarantees

For Kemeny and Hamming embedding:

- **consistency holds**: $\mathcal{R}(d \circ g^*) = \mathcal{R}(s^*)$ and:

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

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

- but the **pre-image step is hard** : NP-hard for Kemeny, $\mathcal{O}(K^3)$ for Hamming (K=number of labels)

In contrast, for the Lehmer embedding:

- we **lose consistency**:

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

- but the **pre-image step is simple**: $\mathcal{O}(K)$

Total complexity

Algorithmic analysis (for K objects to rank, N examples and m dimension of $\phi(\sigma)$)

ϕ	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	$\mathcal{O}(N^3)$	$\mathcal{O}(Nm)$

Embeddings and regressors complexities.

The Lehmer embedding with kNN regressor thus provides the fastest (linear) theoretical complexity of $\mathcal{O}(KN)$ at the cost of weaker theoretical guarantees.
And now in practice?

Structured prediction - Numerical results

Table: 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**

Kendall's τ coefficient corresponds to a rescaling of Kendall's tau distance d_τ between [-1,1] (so the closer from 1 is the better)

Outline

Introduction to ranking data

Label ranking

Structured prediction for label ranking

Ranking embeddings

Computational and theoretical analysis

Openings and conclusion

Extension to partial and incomplete rankings

Different types of rankings:

- Full: $a_1 \succ a_2 \succ \cdots \succ a_K$
- Partial: $a_1, .., a_{k_1} \succ \cdots \succ a_{k_{r-1}+1}, .., a_{k_r}$ with $\sum_{i=1}^{r} k_i = K$
- Incomplete: $a_1 \succ \cdots \succ a_k$ with $k < K$

Can we extend our approach to take as input these types of rankings?

Extension to partial and incomplete rankings

Different types of rankings:

- Full: $a_1 \succ a_2 \succ \cdots \succ a_K$
- Partial: $a_1, .., a_{k_1} \succ \cdots \succ a_{k_{r-1}+1}, .., a_{k_r}$ with $\sum_{i=1}^{r} k_i = K$
- Incomplete: $a_1 \succ \cdots \succ a_k \quad$ with $\quad k < K$

Can we extend our approach to take as input these types of rankings?

- Hamming: **absolute** information $\longrightarrow$ No
- Kemeny: **relative** information $\longrightarrow$ Yes
- Lehmer: **both** $\longrightarrow$ Yes for partial, no for incomplete

Extending our approach to *predict* other types of rankings is mathematically more challenging.

Conclusion

- Flexible framework to optimize various ranking losses
- Statistical and Algorithmic analysis: Optimizing 'good' losses has a price.
- Possible extensions to predict partial / incomplete ranking and improve scalability
- Code available: `https://github.com/akorba/Structured_Approach_Label_Ranking`


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.


Cheng, W. and Hüllermeier, E. (2009).
A new instance-based label ranking approach using the mallows model.
Advances in Neural Networks–ISNN 2009, pages 707–716.


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


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

Clémençon, S., Gaudel, R., and Jakubowicz, J. (2011).
Clustering rankings in the fourier domain.
In *Machine Learning and Knowledge Discovery in Databases*, pages 343–358. Springer.

Davidson, D. and Marschak, J. (1959).
Experimental tests of a stochastic decision theory.
Measurement: Definitions and theories, 17:274.

Jiang, X., Lim, L. H., Yao, Y., and Ye, Y. (2011).
Statistical ranking and combinatorial Hodge theory.
Mathematical Programming, 127(1):203–244.

Jiao, Y., Korba, A., and Sibony, E. (2016).
Controlling the distance to a kemeny consensus without computing it.
In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Jiao, Y. and Vert, J. (2015).

The kendall and mallows kernels for permutations.
In Blei, D. and Bach, F., editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1935–1944.


Kondor, R. and Barbosa, M. S. (2010).
Ranking with kernels in Fourier space.
In *The 23rd Conference on Learning Theory (COLT)*, pages 451–463.


Korba, A., Garcia, A., and Buc d'Alché, F. (2018).
A structured prediction approach for label ranking.
Advances in Neural Information Processing Systems (NIPS).


Li, P., Mazumdar, A., and Milenkovic, O. (2017).
Efficient rank aggregation via lehmer codes.
Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS).


Lu, T. and Boutilier, C. (2014).

Effective sampling and learning for mallows models with pairwise-preference data.
volume 15, pages 3963–4009.


Luce, R. D. (1959).
Individual Choice Behavior.
Wiley.


Mallows, C. L. (1957).
Non-null ranking models.
Biometrika, 44(1-2):114–130.


Plis, S., McCracken, S., Lane, T., and Calhoun, V. (2011).
Directional statistics on permutations.
In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 600–608.


Rajkumar, A. and Agarwal, S. (2014).
A statistical convergence perspective of algorithms for rank aggregation from pairwise data.

In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 118–126.


Shah, N. B. and Wainwright, M. J. (2017).
Simple, robust and optimal ranking from pairwise comparisons.
Journal of Machine Learning Research.


Sibony, E., Clemençon, S., and Jakubowicz, J. (2015).
Mra-based statistical learning from incomplete rankings.
In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1432–1441.


Sun, M., Lebanon, G., and Kidwell, P. (2012).
Estimating probabilities in recommendation systems.
Journal of the Royal Statistical Society: Series C (Applied Statistics), 61(3):471–492.


Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. (2015).

Online rank elicitation for plackett-luce: A dueling bandits approach.
In *Advances in Neural Information Processing Systems (NIPS)*, pages 604–612.


Tversky, A. (1972).
Elimination by aspects: A theory of choice.
Psychological review, 79(4):281.


Zhao, Z., Piech, P., and Xia, L. (2016).
Learning mixtures of plackett-luce models.
In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 2906–2914.