Two families of methods for label ranking

Anna Korba [1]

Joint work with Florence d'Alché-Buc [2], Stephan Clémençon [2], Alexandre Garcia [2], Eric Sibony [3]

[1] Gatsby Unit, CSML, University College London

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

[3] Shift Technology, Paris

MAGNET seminar, INRIA Lille - 7th November 2019

Outline

Outline

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 $[\![K]\!] := \{1, \ldots, K\}$.

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 ($\succ$):

 :  $\succ$  $\succ$  $\succ$ 

 :  $\succ$ 

 :  $\succ$  $\succ$ 

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, biology...)

Analysis of full rankings

Set of items $[\![K]\!] := \{1, \ldots, K\}$. 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**: $a_1, \ldots, a_k \succ$ the rest, **Pairwise comparisons**: $a_1 \succ a_2$...

Analysis of full rankings

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

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

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

Analysis of full rankings

Set of items $[\![K]\!] := \{1, \ldots, K\}$. 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**: $a_1, \ldots, a_k \succ$ 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:

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

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

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

Rankings: a big, structured space

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?

Rankings: a big, structured space

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?

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

Rankings: a big, structured space

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?

- The set of permutations $\mathfrak{S}_K$ is finite...
  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$

Rankings: a big, structured space

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?

- The set of permutations $\mathfrak{S}_K$ is finite...
  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$
  $\Rightarrow$ Little statistical relevance

Rankings: a big, structured space

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

$$\mathcal{D}_N = (\Sigma_1, \Sigma_2, \dots, \Sigma_N) \in \mathfrak{S}_K^N$$

How to analyze it?

- The set of permutations $\mathfrak{S}_K$ is finite...
  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$
  $\Rightarrow$ Little statistical relevance

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

Rankings: a big, structured space

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?

- The set of permutations $\mathfrak{S}_K$ is finite...
  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$
  $\Rightarrow$ Little statistical relevance

- 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!

Rankings: a big, structured space

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?

- The set of permutations $\mathfrak{S}_K$ is finite...
  but it has exploding cardinality: $|\mathfrak{S}_K| = K!$
  $\Rightarrow$ Little statistical relevance

- 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

Main approaches 1 - Parametric

- Choose a predefined generative model on the data and analyze the data through that model

Main approaches 1 - Parametric

- Choose a predefined generative model on the data and analyze the data through that model
  - 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
  - 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
  1. Modeling of pairwise comparisons ([Jiang et al., 2011, Rajkumar and Agarwal, 2014, Shah and Wainwright, 2017])
  2. Kernel methods [Jiao and Vert, 2015]...
- Our setting: we exploit these structures to develop methods for label ranking data
- We also rely on results on a fundamental problem: **ranking aggregation**.

Ranking aggregation

Consider a dataset of N rankings/permutations of $[\![K]\!]$:

$$\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N) \in \mathfrak{S}_K^N$$

Rank. agg. aims at finding a global order (*consensus*) on the K items that best represent the dataset.

Ranking aggregation

Consider a dataset of N rankings/permutations of $[\![K]\!]$:

$$\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N) \in \mathfrak{S}_K^N$$

Rank. agg. aims at finding a global order (*consensus*) on the K items that best represent the dataset.

Kemeny's rule [Kemeny, 1959]

$$\text{Solve} \quad \sigma^*_{\mathcal{D}_N} = \operatorname*{argmin}_{\sigma \in \mathfrak{S}_K} \sum_{n=1}^{N} d(\sigma, \Sigma_n)$$

Ranking aggregation

Consider a dataset of N rankings/permutations of $[\![K]\!]$:

$$\mathcal{D}_N = (\Sigma_1, \dots, \Sigma_N) \in \mathfrak{S}_K^N$$

Rank. agg. aims at finding a global order (*consensus*) on the K items that best represent the dataset.

Kemeny's rule [Kemeny, 1959]

$$\text{Solve} \quad \sigma^*_{\mathcal{D}_N} = \underset{\sigma \in \mathfrak{S}_K}{\text{argmin}} \sum_{n=1}^{N} d(\sigma, \Sigma_n)$$

where d is the Kendall's τ distance, i.e. for $\sigma, \sigma' \in \mathfrak{S}_K$:

$$d_\tau(\sigma, \sigma') = \sum_{1 \leq i < j \leq K} \mathbb{I}\{(\sigma(i) - \sigma(j))(\sigma'(i) - \sigma'(j)) < 0\},$$

Ex: σ= 1234, σ'= 2413 $\Rightarrow d_\tau(\sigma, \sigma') = 3$ (disagree on (12),(14),(34)).

Tractable (Kemeny) ranking aggregation

- Natural loss when rankings represent preferences
- Kemeny's rule is the "canonical" way to aggregate rankings
- **Pb:** it is NP-hard in general even for $N = 4$ ([Dwork et al., 2001])

Tractable (Kemeny) ranking aggregation

- Natural loss when rankings represent preferences
- Kemeny's rule is the "canonical" way to aggregate rankings
- **Pb:** it is NP-hard in general even for $N = 4$ ([Dwork et al., 2001])

Probabilistic Modeling

$$\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N) \qquad \text{with} \qquad \Sigma_n \sim P$$

where P distribution on $\mathfrak{S}_K$. In [Korba et al., 2017], we exhibit some conditions on P so that solving (true) Kemeny ranking aggregation:

$$\sigma_P^* = \operatorname*{argmin}_{\sigma \in \mathfrak{S}_K} \mathbb{E}_{\Sigma \sim P}[d(\sigma, \Sigma)]$$

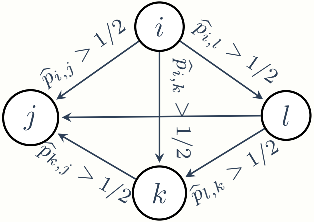
is tractable.

Let $p_{i,j} = \mathbb{P}[\Sigma(i) < \Sigma(j)]$ prob. that item $i \succ j$. Suppose:

- Strict Stochastic Transitivity (SST):
  $(p_{i,j} \neq 1/2)$ & $(p_{i,j} > 1/2$ and $p_{j,k} > 1/2 \Rightarrow p_{i,k} > 1/2.)$
- Low-noise: $\min_{i<j} |p_{i,j} - 1/2| \geq h.$

Let $p_{i,j} = \mathbb{P}[\Sigma(i) < \Sigma(j)]$ prob. that item $i \succ j$. Suppose:

- Strict Stochastic Transitivity (SST):
  $(p_{i,j} \neq 1/2)$ & $(p_{i,j} > 1/2$ and $p_{j,k} > 1/2 \Rightarrow p_{i,k} > 1/2.)$
- Low-noise: $\min_{i<j} |p_{i,j} - 1/2| \geq h.$



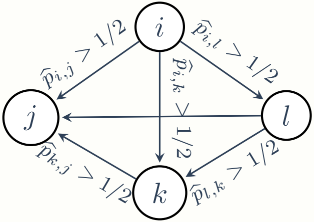
$\Rightarrow \widehat{P}$ will verify SST

$\Rightarrow$ Sort vertices by increasing input degree:

d(i)=0,d(l)=1,d(k)=2, d(j)=3

Let $p_{i,j} = \mathbb{P}[\Sigma(i) < \Sigma(j)]$ prob. that item $i \succ j$. Suppose:

- Strict Stochastic Transitivity (SST):
  $(p_{i,j} \neq 1/2)$ & $(p_{i,j} > 1/2$ and $p_{j,k} > 1/2 \Rightarrow p_{i,k} > 1/2.)$
- Low-noise: $\min_{i<j} |p_{i,j} - 1/2| \geq h$.



$\Rightarrow \widehat{P}$ will verify SST
$\Rightarrow$ Sort vertices by increasing input degree:
d(i)=0,d(l)=1,d(k)=2, d(j)=3

Theorem: The Kemeny median of P **is unique** and given by the empirical Copeland ranking (complexity: $\mathcal{O}(K^2N)$):

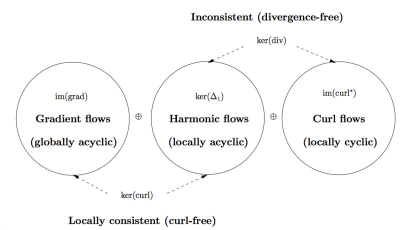
$$\text{for each } 1 \leq i \leq K, \quad \sigma^*_P(i) = 1 + \sum_{j \neq i} \mathbb{I}\{\widehat{p}_{i,j} < \frac{1}{2}\}$$

(with overwhelming probability $1 - \frac{K(K-1)}{4} e^{-\alpha_h N}$, $\alpha_h = \frac{1}{2} \log\left(1/(1-4h^2)\right)$)

What if $\widehat{P}$ does not satisfy SST (Strict Stochastic Transitivity)?

- We propose to compute an approximation $\widetilde{\sigma}^*_{\widehat{P}}$ with empirical Borda count

$$\widetilde{\sigma}^*_{\widehat{P}}(i) = \sigma^*_{proj_{im(grad)}(\widehat{P})}(i) = \frac{1}{N}\sum_{n=1}^{N} \Sigma_n(i) \quad \text{for } 1 \leq i \leq K$$



Hodge decomposition of pairwise rankings ([Jiang et al., 2011])

- (Remark:) Borda $\neq$ Kemeny unless

$$p_{i,j} > 1/2 \text{ and } p_{j,k} > 1/2 \Rightarrow p_{i,k} > max(p_{i,j}, p_{j,k})$$

Outline

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

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

Goal: Learn a predictive ranking rule :

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

which given a random X, predicts the permutation $s(X)$ on $[\![K]\!]$.

 $\rightarrow$ 

$\succ$  $\succ$  $\succ$ 

Example: targeted advertising domain

Related Work

- Other applications:
  - document categorization/sentiment analysis: rank a set of topics or emotions by relevance for a given document
  - meta learning: rank a set of algorithms according to their suitability for a new dataset
- Can be seen as an extension of multiclass and multilabel classification (postprocess a label ranking prediction in a suitable way)
- Most previous approaches rely on parametric modelling [Cheng and Hüllermeier, 2009], [Cheng et al., 2010]

Related Work

- Other applications:
  - document categorization/sentiment analysis: rank a set of topics or emotions by relevance for a given document
  - meta learning: rank a set of algorithms according to their suitability for a new dataset
- Can be seen as an extension of multiclass and multilabel classification (postprocess a label ranking prediction in a suitable way)
- Most previous approaches rely on parametric modelling [Cheng and Hüllermeier, 2009], [Cheng et al., 2010]

We develop two families of non-parametric methods:

1. **Partitioning methods** relying on results obtained for **ranking aggregation**.
2. **Structured prediction methods**, exploiting the geometry of well-chosen **feature maps** for rankings.

Outline

Motivation: Label ranking as an extension of ranking aggregation

Suppose:

- $X \sim \mu$, where μ is a distribution on some feature space $\mathcal{X}$
- $\Sigma \sim P_X$, where P_X (on $\mathfrak{S}_K$) is the conditional probability distribution : $P_X(\sigma) = \mathbb{P}[\Sigma = \sigma | X]$

Performance of s: Measured by the risk:

$$\mathcal{R}(s) = \mathbb{E}_{X \sim \mu} \underbrace{\mathbb{E}_{\Sigma \sim P_X} \left[d_\tau \left(s(X), \Sigma \right) \right]}_{\substack{\text{ranking aggregation risk,} \\ \text{minimized if } s(X) = \sigma^*_{P_X}}}$$

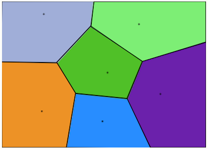
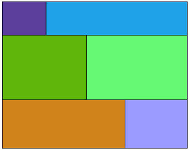
Assumption

For $X \in \mathcal{X}$, P_X is **SST**: $\Rightarrow \sigma^*_{P_X}$ is unique (and given by Copeland)

Idea: Relax within a region $\mathcal{C}$ and compute $\sigma^*_{P_\mathcal{C}}$ for $P_\mathcal{C}(\sigma) = \mathbb{P}[\Sigma = \sigma | X \in \mathcal{C}]$.

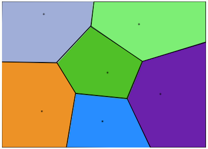
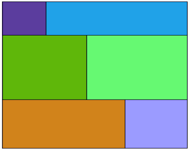
Partitioning Methods

Two methods are investigated:

K-nearest neighbors (Voronoi partitioning)	*Decision tree* (Recursive partition)
	

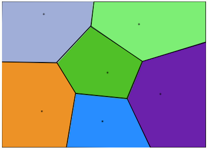
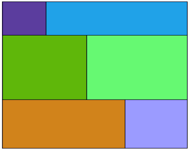
Partitioning Methods

Two methods are investigated:

K-nearest neighbors (Voronoi partitioning)	*Decision tree* (Recursive partition)
	

Partitioning Methods

Two methods are investigated:

K-nearest neighbors (Voronoi partitioning)	*Decision tree* (Recursive partition)
	

Consider the empirical distribution of rankings in $\mathcal{C}$:

$$\widehat{P}_{\mathcal{C}} = \frac{1}{|k : X_k \in \mathcal{C}|} \sum_{k : X_k \in \mathcal{C}} \delta_{\Sigma_k}$$

and solve:

$$\sigma^*_{\widehat{P}_{\mathcal{C}}} = \operatorname*{argmin}_{\sigma \in \mathfrak{S}_K} \mathbb{E}_{\Sigma \sim \widehat{P}_{\mathcal{C}}} [d_\tau(\sigma, \Sigma)]$$

$\Longrightarrow$ compute with Copeland method if $\widehat{P}_{\mathcal{C}}$ is **SST**, Borda otherwise

Partition the feature space: ex. of the decision tree

Split recursively the feature space by minimizing some impurity criterion.

Recall Gini criterion in multiclassification, if m is the nb of classes, and $f_i(\mathcal{C})$ proportion of class i in cell $\mathcal{C}$:

$$I_G(\mathcal{C}) = \sum_{i=1}^{m} f_i(\mathcal{C})(1 - f_i(\mathcal{C}))$$

Partition the feature space: ex. of the decision tree

Split recursively the feature space by minimizing some impurity criterion.

Recall Gini criterion in multiclassification, if m is the nb of classes, and $f_i(\mathcal{C})$ proportion of class i in cell $\mathcal{C}$:

$$I_G(\mathcal{C}) = \sum_{i=1}^{m} f_i(\mathcal{C})(1 - f_i(\mathcal{C}))$$

Here, for a cell $\mathcal{C}$ [Alvo and Yu, 2014]:

$$\gamma(\mathcal{C}) = \frac{1}{2} \sum_{1 \leq i < j \leq K} \widehat{p}_{i,j}(\mathcal{C}) \left(1 - \widehat{p}_{i,j}(\mathcal{C})\right)$$

which is tractable and satisfies the double inequality

$$\gamma(\mathcal{C}) \leq \min_{\sigma \in \mathfrak{S}_K} \mathbb{E}_{\Sigma \sim \widehat{P}_{\mathcal{C}}}[d(\sigma, \Sigma)] \leq 2\gamma(\mathcal{C})$$

Idea: ordering K elements can be seen as $\binom{K}{2}$ classification tasks.

Main results of [Clémençon et al., 2018]

Approximation error: ”partitioning methods approximate well”
Suppose that $\exists M < \infty$ such that:
$\forall (x, x') \in \mathcal{X}^2, \;\; \sum_{i<j} |p_{i,j}(x) - p_{i,j}(x')| \leq M \cdot ||x - x'||$, then

$$\inf_{\substack{s \in \text{piec. cst. on } \mathcal{P} \\ \text{equal to } \sigma^*_{P_\mathcal{C}} \text{ on } \mathcal{C}}} \mathcal{R}(s) - \mathcal{R}(s^*) \leq M.\delta_\mathcal{P}$$

where $\delta_\mathcal{P}$ is the max. diameter of $\mathcal{P}$'s cells.

Rates. Let $\widehat{s}_N$ a minimizer of the empirical risk over $\{$piec. cst. on $\mathcal{P}\}$. Excess of risk $\mathcal{R}(\widehat{s}_N) - \mathcal{R}(s^*)$?

- classical rates $\mathcal{O}(1/\sqrt{N})$ for ERM.
- fast rates $\mathcal{O}(1/N)$ under a ”uniform” Low-Noise **NA**(h):

$$\inf_{x \in \mathcal{X}} \min_{i<j} |p_{i,j}(x) - 1/2| \geq h.$$

Extensions and Limitations

- Could be extended to the setting where one only observes pairwise comparisons
- However, the SST assumption on P_X may be strict
- Also, we only work with Kendall's tau distance as a loss

Outline

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:
  $\Delta_\tau(\sigma, \sigma') = \sum_{1 \leq i < j \leq K} \mathbb{I}[(\sigma(i) - \sigma(j))(\sigma'(i) - \sigma'(j)) < 0]$
  $\rightarrow$ Intuitive when rankings represent preferences
- **Hamming**: $\Delta_H(\sigma, \sigma') = \sum_{i=1}^{K} \mathbb{I}[\sigma(i) \neq \sigma'(i)]$.
  $\rightarrow$ Popular when rankings represent matchings/assignments

Structured prediction for label ranking

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

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

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

$$\boxed{\Delta(\sigma, \sigma') = \|\phi(\sigma) - \phi(\sigma')\|_{\mathcal{F}}^2.} \qquad (1)$$

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. $\mathbb{R}^m$).

Motivation: There exist ϕ_τ, ϕ_H such that Δ_τ and Δ_H write as (1).

Structured prediction - surrogate problem

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

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

Structured prediction - surrogate problem

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

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

Idea: Introduce a surrogate problem:

$$\min_{g\,:\,\mathcal{X}\to\mathcal{F}} \mathcal{L}(g), \qquad \text{with} \qquad \mathcal{L}(g) = \mathbb{E}\left[\|g(X) - \phi(\Sigma)\|_{\mathcal{F}}^2\right] \tag{3}$$

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

Let s^* be a minimizer of (2) and g^* a minimizer of (3).

Structured prediction - surrogate problem

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

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

Idea: Introduce a surrogate problem:

$$\min_{g\,:\,\mathcal{X}\to\mathcal{F}}\mathcal{L}(g),\qquad \text{with}\qquad \mathcal{L}(g)=\mathbb{E}\left[\|g(X)-\phi(\Sigma)\|_{\mathcal{F}}^2\right] \tag{3}$$

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

Let s^* be a minimizer of (2) and g^* a minimizer of (3).
Consistency if: $\mathcal{R}(d\circ g^*)=\mathcal{R}(s^*)$**.**

Structured prediction - surrogate problem

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

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

Idea: Introduce a surrogate problem:

$$\min_{g\,:\,\mathcal{X}\to\mathcal{F}} \mathcal{L}(g), \quad \text{with} \quad \mathcal{L}(g) = \mathbb{E}\left[\|g(X) - \phi(\Sigma)\|_{\mathcal{F}}^2\right] \quad (3)$$

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

Let s^* be a minimizer of (2) and g^* a minimizer of (3).

Consistency if: $\mathcal{R}(d \circ g^*) = \mathcal{R}(s^*)$**.**

$\Longrightarrow$ approach structured prediction in **two steps**:
(see [Ciliberto et al., 2016, Brouard et al., 2016])

Structured Prediction Approach

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

$$\begin{tikzcd} \mathcal{X} \arrow[r, "\widehat{s}"] \arrow[d, "\widehat{g}"'] & \mathfrak{S}_K \\ \mathcal{F} \arrow[ur, "d\,(\phi^{-1})"'] & \end{tikzcd}$$

- **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

Structured Prediction Approach

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

$$\begin{tikzcd} \mathcal{X} \arrow[r, "\widehat{s}"] \arrow[d, "\widehat{g}"'] & \mathfrak{S}_K \\ \mathcal{F} \arrow[ur, "d\,(\phi^{-1})"'] & \end{tikzcd}$$

- **Step 1 (Regression)**: Learn $\widehat{g} : \mathcal{X} \to \mathcal{F}$
  - Step 1 (a): map $\mathcal{D}_N = (X_1, \Sigma_1), \dots, (X_N, \Sigma_N)$ to
    $\mathcal{D}'_N = (X_1, \phi(\Sigma_1)), \dots, (X_N, \phi(\Sigma_N))$ 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 $\widehat{s}(x) = \text{argmin}_{\sigma \in \mathfrak{S}_K} \|\phi(\sigma) - \widehat{g}(x)\|^2_{\mathcal{F}}$

Structured Prediction Approach

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

$$\begin{tikzcd} \mathcal{X} \arrow[r, "\widehat{s}"] \arrow[d, "\widehat{g}"'] & \mathfrak{S}_K \\ \mathcal{F} \arrow[ur, "d\,(\phi^{-1})"'] & \end{tikzcd}$$

- **Step 1 (Regression)**: Learn $\widehat{g} : \mathcal{X} \to \mathcal{F}$
  - Step 1 (a): map $\mathcal{D}_N = (X_1, \Sigma_1), \dots, (X_N, \Sigma_N)$ to $\mathcal{D}'_N = (X_1, \phi(\Sigma_1)), \dots, (X_N, \phi(\Sigma_N))$ 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 $\widehat{s}(x) = \text{argmin}_{\sigma \in \mathfrak{S}_K} \|\phi(\sigma) - \widehat{g}(x)\|^2_{\mathcal{F}}$

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)

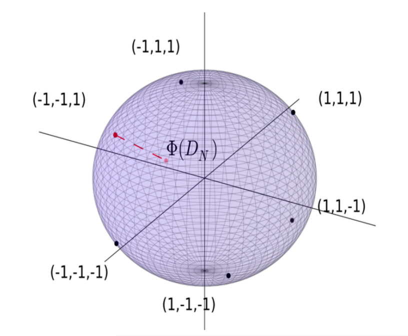
We now study 3 embeddings and their properties.

Ranking embeddings - Kemeny

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

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

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



(Recovers Kendall's tau distance d_τ)

Ranking embeddings - Hamming and Lehmer

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

(recovers Hamming distance d_H)

Ranking embeddings - Hamming and Lehmer

- 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: $\sigma = 132 \Longrightarrow \phi_H(\sigma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
(recovers Hamming distance d_H)

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

Complexity 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 **Kemeny and Hamming**, $\|\phi(\sigma)\| = C$ for any σ, so 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 :

$$\phi_\sigma^* = \underset{\phi_\sigma \in Im(\phi)}{\operatorname{argmax}} \langle \phi_\sigma, \widehat{g}(x) \rangle_{\mathcal{F}}$$

   $\Longrightarrow$ **NP-hard for Kemeny,** $\mathcal{O}(K^3)$ **for Hamming with the Hungarian algorithm.**

2. Invert the embedding: $\sigma = \phi^{-1}(\phi_\sigma^*)$

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 $j = 1, \ldots, K$:

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

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 $j = 1, \ldots, K$:

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

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

problem:

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

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	$\mathcal{O}(K^2 N)$	NP-hard
ϕ_H	$\mathcal{O}(KN)$	$\mathcal{O}(K^3 N)$
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**

Cheng PL ($\mathcal{O}(K \log(K)N)$) [Cheng et al., 2010] , Cheng LWD ($\mathcal{O}(K^3N)$) [Cheng and Hüllermeier, 2013], Zhou RF ($\mathcal{O}(K^2N^2)$) [Zhou and Qiu, 2016]

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

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: $a_1 \succ \cdots \succ a_k$ with $k < K$

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

Extension to partial and incomplete rankings

Different types of rankings:

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

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

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

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

[Fagin et al., 2004] propose an extension of Kendall's tau on partial rankings, which can be written as $\Delta(\sigma, \sigma') = \|\phi(\sigma) - \phi(\sigma')\|^2_{\mathcal{F}}$, but the consistency will be lost.

Conclusion

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

Alvo, M. and Yu, P. L. H. (2014).
Statistical Methods for Ranking Data.
Springer.

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. 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 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., Korba, A., and Sibony, E. (2018).
Ranking median regression: Learning to order through local consensus.
International Conference on Algorithmic Learning Theory (ALT).

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

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001).
Rank aggregation methods for the web.

In *Proceedings of the 10th International Conference on World Wide Web*, pages 613–622. ACM.

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2004).
Comparing and aggregating rankings with ties.
In *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58. ACM.

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.

Kemeny, J. G. (1959).
Mathematics without numbers.
Daedalus, 88:571–591.

Korba, A., Clémençon, S., and Sibony, E. (2017).
A learning theory of ranking aggregation.
In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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).

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.

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

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

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.

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

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

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.

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

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

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

Pre-image for the Hamming embedding

Enforce the constraints of Hamming representations

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

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

$\Longrightarrow$ Bipartite graph matching problem.

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