# Ranking Median Regression: Learning to Order through Local Consensus

Anna Korba* Stéphan Clémençon* Eric Sibony†

* Telecom ParisTech, † Shift Technology

Universitá degli studi di Genova
June 26 2018

# Outline

1. Introduction to Ranking Data
2. Ranking Regression
3. Background on Ranking Aggregation/Medians
4. Risk Minimization for Ranking (Median) Regression
5. Algorithms - Local Median Methods
6. Ongoing work - Structured prediction methods
7. Conclusion

# Outline

# Ranking Data

Set of items $[\![n]\!] := \{1, \ldots, n\}$

## Definition (Ranking)

A ranking is a strict partial order $\prec$ over $[\![n]\!]$, *i.e.* a binary relation satisfying the following properties:

**Irreflexivity** For all $i \in [\![n]\!]$, $i \not\prec i$

**Transitivity** **For all i, j, k $\in [\![n]\!]$, if i $\prec$ j and j $\prec$ k then i $\prec$ k**

**Asymmetry** For all $i, j \in [\![n]\!]$, if $i \prec j$ then $j \not\prec i$

# Common types of rankings

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

- **Full ranking.** All the items are ranked, without ties

$$a_1 \succ a_2 \succ \cdots \succ a_n$$

- **Partial ranking.** All the items are ranked, with ties (”buckets”)

$$a_{1,1}, \dots, a_{1,n_1} \succ \cdots \succ a_{r,1}, \dots, a_{r,n_r} \quad \text{with} \quad \sum_{i=1}^{r} n_i = n$$

  $\Rightarrow$ includes **Top-k ranking**: $a_1, \dots, a_k \succ$ the rest

- **Incomplete ranking.** Only a subset of items are ranked, without ties

$$a_1 \succ \cdots \succ a_k \quad \text{with} \quad k < n$$

  $\Rightarrow$ includes **Pairwise comparison**: $a_1 \succ a_2$

# Ranking data arise in a lot of applications

## Traditional applications

- **Elections**: $[\![n]\!]$= a set of candidates
  $\rightarrow$ A voter ranks a set of candidates
- **Competitions**: $[\![n]\!]$= a set of players
  $\rightarrow$ Results of a race
- **Surveys**: $[\![n]\!]$= political goals
  $\rightarrow$ A citizen ranks according to its priorities

## Modern applications

- **E-commerce**: $[\![n]\!]$= items of a catalog
  $\rightarrow$ A user expresses its preferences (see "implicit feedback")
- **Search engines**: $[\![n]\!]$= web-pages
  $\rightarrow$ A search engine ranks by relevance for a given query

# Detailed example: analysis of full rankings

Consider:

- A set of $n$ items: $[\![n]\!] = \{1, \dots, n\}$ (Ex: $\{1, 2, 3, 4\}$)
- A full ranking: $a_1 \succ a_2 \succ \cdots \succ a_n$ (Ex: $2 \succ 1 \succ 3 \succ 4$)
- Also seen as the permutation $\sigma$ that maps an item to its rank:

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

Ex: $\sigma(2) = 1, \sigma(1) = 2, \cdots \Rightarrow \sigma = 2134$

- $\mathfrak{S}_n$: set of permutations of $[\![n]\!]$, the symmetric group.
  Ex: $\mathfrak{S}_4 = 1234, 1324, 1423, \dots, 4321$

**Probabilistic Modeling.** The dataset is a collection of random permutations drawn IID from a probability distribution $P$ over $\mathfrak{S}_n$:

$$\mathcal{D}_N = (\Sigma_1, \dots, \Sigma_N) \in \mathfrak{S}_n^N \qquad \text{with} \qquad \Sigma_i \sim P$$

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

## Challenges

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

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

## Challenges

- A random permutation $\Sigma \in \mathfrak{S}_n$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(n)) \in \mathbb{R}^n$... but
  The random variables $\Sigma(1), \ldots, \Sigma(n)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!
  $\Rightarrow$No natural notion of variance for $\Sigma$

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

## Challenges

- A random permutation $\Sigma \in \mathfrak{S}_n$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(n)) \in \mathbb{R}^n$... but
  The random variables $\Sigma(1), \ldots, \Sigma(n)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!
  $\Rightarrow$No natural notion of variance for $\Sigma$
- The set of permutations $\mathfrak{S}_n$ is finite... but

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \dots, \Sigma_N)$?

## Challenges

- A random permutation $\Sigma \in \mathfrak{S}_n$ can be seen as a random vector $(\Sigma(1), \dots, \Sigma(n)) \in \mathbb{R}^n$... but
  The random variables $\Sigma(1), \dots, \Sigma(n)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!
  $\Rightarrow$No natural notion of variance for $\Sigma$
- The set of permutations $\mathfrak{S}_n$ is finite... but
  Exploding cardinality: $|\mathfrak{S}_n| = n!$
  $\Rightarrow$ Few statistical relevance

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

## Challenges

- A random permutation $\Sigma \in \mathfrak{S}_n$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(n)) \in \mathbb{R}^n$... but
  The random variables $\Sigma(1), \ldots, \Sigma(n)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!
  $\Rightarrow$No natural notion of variance for $\Sigma$
- The set of permutations $\mathfrak{S}_n$ is finite... but
  Exploding cardinality: $|\mathfrak{S}_n| = n!$
  $\Rightarrow$ Few statistical relevance
- Apply a method from p.d.f. estimation (e.g. kernel density estimation)... but

# Detailed example: analysis of full rankings

How to analyze a dataset of permutations $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$?

## Challenges

- A random permutation $\Sigma \in \mathfrak{S}_n$ can be seen as a random vector $(\Sigma(1), \ldots, \Sigma(n)) \in \mathbb{R}^n$... but
  The random variables $\Sigma(1), \ldots, \Sigma(n)$ are highly dependent and the sum $\Sigma + \Sigma'$ is not a random permutation!
  $\Rightarrow$No natural notion of variance for $\Sigma$
- The set of permutations $\mathfrak{S}_n$ is finite... but
  Exploding cardinality: $|\mathfrak{S}_n| = n!$
  $\Rightarrow$ Few statistical relevance
- Apply a method from p.d.f. estimation (e.g. kernel density estimation)... but
  No canonical ordering of the rankings!

# Main approaches

**"Parametric" approach**

- Fit a predefined generative model on the data
- Analyze the data through that model

**"Nonparametric" approach**

- Choose a structure on $\mathfrak{S}_n$
- Analyze the data with respect to that structure

# Parametric Approach - Example of Models

- **Mallows model** [Mallows, 1957]
  Parameterized by a central ranking $\sigma_0 \in \mathfrak{S}_n$ and a dispersion parameter $\gamma \in \mathbb{R}^+$

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

# Parametric Approach - Example of Models

- **Mallows model** [Mallows, 1957]
  Parameterized by a central ranking $\sigma_0 \in \mathfrak{S}_n$ and a dispersion parameter $\gamma \in \mathbb{R}^+$

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

- **Plackett-Luce model** [Luce, 1959], [Plackett, 1975]
  Each item $i$ is parameterized by $w_i$ with $w_i \in \mathbb{R}^+$:

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

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

# Nonparametric approaches - Examples 1

- Harmonic analysis
  - Fourier analysis [Clémençon et al., 2011], [Kondor and Barbosa, 2010]

$$\hat{h}_\lambda = \sum_{\sigma \in \mathfrak{S}_n} h(\sigma)\rho_\lambda(\sigma) \text{ où } \rho_\lambda(\sigma) \in \mathbb{C}^{d_\lambda \times d_\lambda} \text{ for all } \lambda \vdash n.$$

  - Multiresolution analysis for incomplete rankings [Sibony et al., 2015]

# Nonparametric approaches - Examples 1

- Harmonic analysis
  - Fourier analysis [Clémençon et al., 2011], [Kondor and Barbosa, 2010]

$$\hat{h}_\lambda = \sum_{\sigma \in \mathfrak{S}_n} h(\sigma)\rho_\lambda(\sigma) \text{ où } \rho_\lambda(\sigma) \in \mathbb{C}^{d_\lambda \times d_\lambda} \text{ for all } \lambda \vdash n.$$

  - Multiresolution analysis for incomplete rankings [Sibony et al., 2015]
- Embeddings of permutations
  - Permutation matrices [Plis et al., 2011]

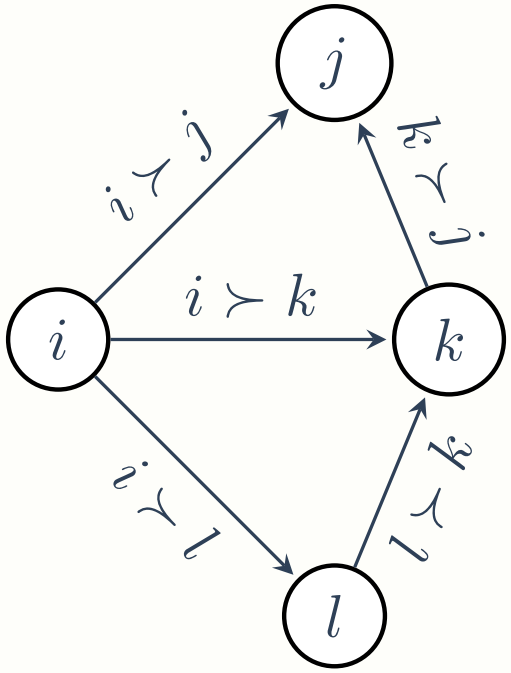
$$\mathfrak{S}_n \to \mathbb{R}^{n \times n}, \quad \sigma \mapsto P_\sigma \quad \text{with } P_\sigma(i,j) = \mathbb{I}\{\sigma(i) = j\}$$

  - Kemeny embedding [Jiao et al., 2016]

$$\mathfrak{S}_n \to \mathbb{R}^{n(n-1)/2}, \quad \sigma \mapsto \phi_\sigma \quad \text{with } \phi_\sigma = \begin{pmatrix} \vdots \\ sign(\sigma(i) - \sigma(j)) \\ \vdots \end{pmatrix}_{i<j}$$

# Nonparametric approaches - Examples 2

Modeling of pairwise comparisons as a graph:




- HodgeRank exploits the topology of the graph
  [Jiang et al., 2011]
- Approximation of pairwise comparison matrices
  [Shah and Wainwright, 2015]

# Some ranking problems

Perform some task on a dataset of $N$ rankings $\mathcal{D}_N = (\Sigma_1, \ldots, \Sigma_N)$.

## Examples

- **Top-1 recovery:** Find the "most preferred" item in $\mathcal{D}_N$
  e.g. Output of an election
- **Aggregation:** Find a full ranking that "best summarizes" $\mathcal{D}_N$
  e.g. Ranking of a competition
- **Clustering:** Split $\mathcal{D}_N$ into clusters
  e.g. Segment customers based on their answers to a survey
- **Prediction:** Predict a ranking given some information
  e.g. In a recommendation setting

# Outline

# Ranking Regression

**Problem**: Given a vector $X$ (e.g, the characteristics of an individual), the goal is to predict (her preferences) as a random permutation $\Sigma$ in $\mathfrak{S}_n$.

# Ranking Regression

**Problem**: Given a vector $X$ (e.g, the characteristics of an individual), the goal is to predict (her preferences) as a random permutation $\Sigma$ in $\mathfrak{S}_n$.

**Example:** $n$=4 items (fruits)

# Related Work

- Has been referred to as **label ranking** in the literature [Tsoumakas et al., 2009], [Vembu and Gärtner, 2010]
- Related to multiclass and multilabel classification
- A lot of 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., 2009], [Cheng et al., 2010]

# Related Work

- Has been referred to as **label ranking** in the literature [Tsoumakas et al., 2009], [Vembu and Gärtner, 2010]
- Related to multiclass and multilabel classification
- A lot of 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., 2009], [Cheng et al., 2010]

$\Rightarrow$ We develop an approach free of any parametric assumptions (**local learning**) relying on results and framework developped in [Korba et al., 2017] for **ranking aggregation**.

# Problem and Setting

Suppose we observe $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ i.i.d. copies of the pair $(X,\ \Sigma)$, where

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

*Ex: Users $i$ with characteristics $X_i$ order items by preference resulting in $\Sigma_i$.*

**Goal**: Learn a predictive ranking rule :

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

which given a random vector $X$, predicts the permutation $\Sigma$ on the $n$ items.

# Objective

**Performance**: Measured by the risk:

$$\mathcal{R}(s) = \mathbb{E}_{X \sim \mu, \Sigma \sim P_X} \left[ d_\tau \left( s(X), \Sigma \right) \right]$$

# Objective

**Performance**: Measured by the risk:

$$\mathcal{R}(s) = \mathbb{E}_{X \sim \mu, \Sigma \sim P_X} \left[ d_\tau \left( s(X), \Sigma \right) \right]$$

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

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

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

# Piecewise Constant Ranking Rules

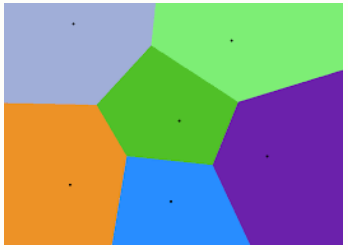
**Our approach**: build *piecewise constant* ranking rules, i.e:
Ranking rules that are constant on each cell of a partition of $\mathcal{X}$ built from the training data $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$.

# Piecewise Constant Ranking Rules

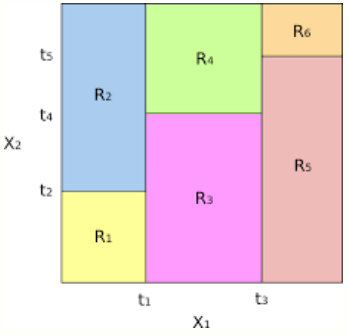
**Our approach**: build *piecewise constant* ranking rules, i.e:
Ranking rules that are constant on each cell of a partition of $\mathcal{X}$ built from the training data $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$.

Two methods are investigated:

- k-nearest neighbor (Voronoi partitioning)

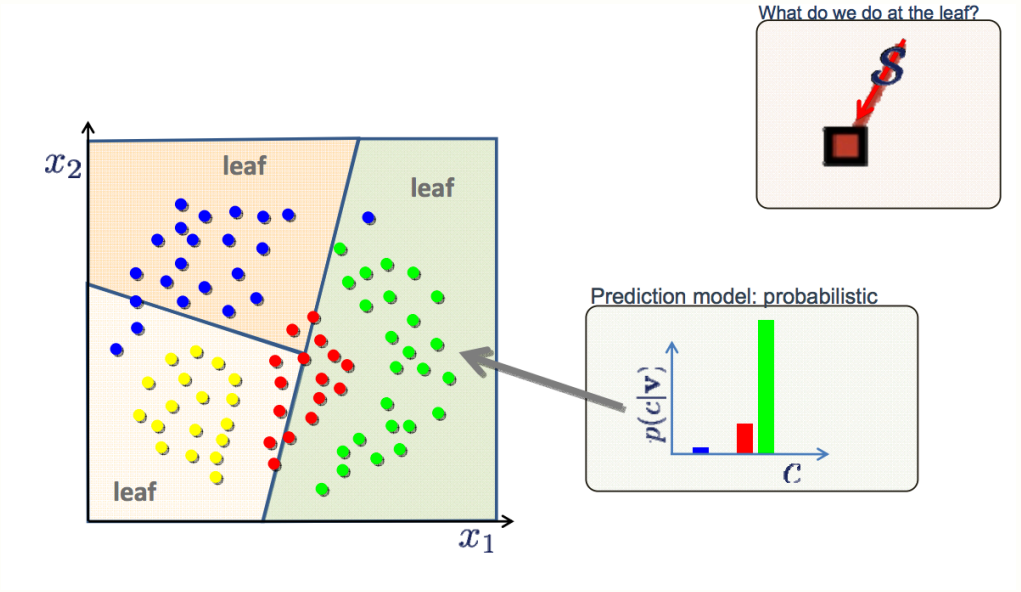


- decision tree (Recursive partitioning)



# Compute Local Labels/Medians

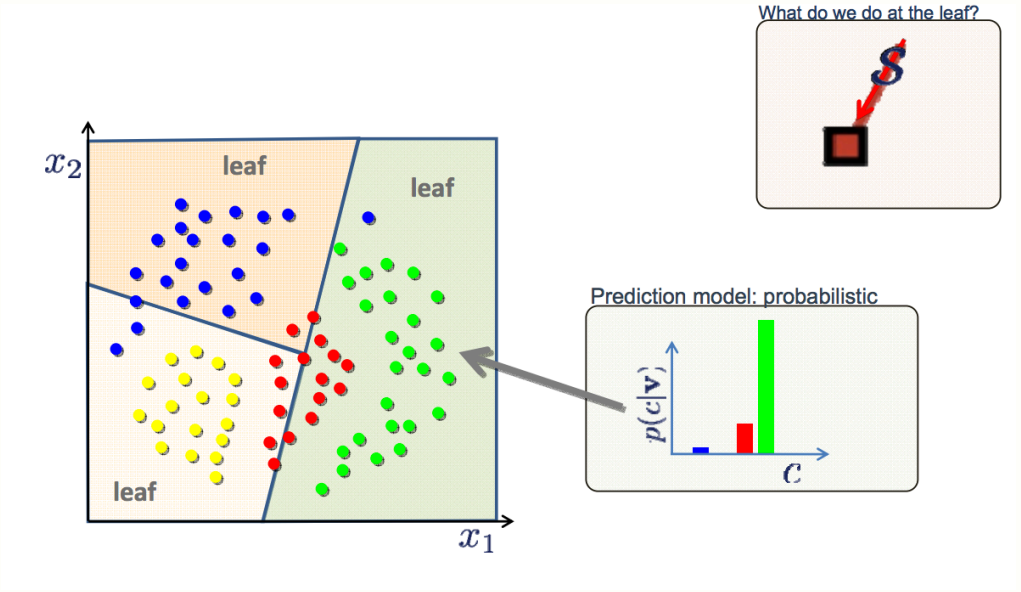
For classification, the label of a cell (ex: a leaf) is the **majority** label among the training data which fall in this cell.




4 classes: green, red, blue, yellow $\rightarrow$ green will be the label for the right cell.

# Compute Local Labels/Medians

For classification, the label of a cell (ex: a leaf) is the **majority** label among the training data which fall in this cell.




4 classes: green, red, blue, yellow $\rightarrow$ green will be the label for the right cell.

**Problem:** Our labels are *permutations* $\sigma$:

For a cell $\mathcal{C}$, if $\Sigma_1, \ldots, \Sigma_N \in \mathcal{C}$, how do we aggregate them into a final label $\sigma^*$?

$\Longrightarrow$ Ranking aggregation problem.

# Outline

# Ranking Aggregation - Methods

Suppose we have a dataset of rankings/permutations $\mathcal{D}_N = (\sigma_1, \ldots, \sigma_N) \in \mathfrak{S}_n^N$. We want to find a global order ("consensus") $\sigma^*$ on the $n$ items that best represents the dataset.

# Ranking Aggregation - Methods

Suppose we have a dataset of rankings/permutations $\mathcal{D}_N = (\sigma_1, \dots, \sigma_N) \in \mathfrak{S}_n^N$. We want to find a global order ("consensus") $\sigma^*$ on the $n$ items that best represents the dataset.

## Kemeny's rule (1959) - Optimization pb

$$\text{Solve} \quad \sigma^* = \underset{\sigma \in \mathfrak{S}_n}{\text{argmin}} \sum_{k=1}^{N} d(\sigma, \sigma_k)$$

# Ranking Aggregation - Methods

Suppose we have a dataset of rankings/permutations $\mathcal{D}_N = (\sigma_1, \dots, \sigma_N) \in \mathfrak{S}_n^N$. We want to find a global order ("consensus") $\sigma^*$ on the $n$ items that best represents the dataset.

## Kemeny's rule (1959) - Optimization pb

$$\text{Solve} \quad \sigma^* = \underset{\sigma \in \mathfrak{S}_n}{\text{argmin}} \sum_{k=1}^{N} d(\sigma, \sigma_k)$$

**Problem:** NP-hard.

# Ranking Aggregation - Methods

Suppose we have a dataset of rankings/permutations $\mathcal{D}_N = (\sigma_1, \dots, \sigma_N) \in \mathfrak{S}_n^N$. We want to find a global order ("consensus") $\sigma^*$ on the $n$ items that best represents the dataset.

## Kemeny's rule (1959) - Optimization pb

$$\text{Solve} \quad \sigma^* = \underset{\sigma \in \mathfrak{S}_n}{\operatorname{argmin}} \sum_{k=1}^{N} d(\sigma, \sigma_k)$$

**Problem:** NP-hard.

## Copeland method - Scoring method

Sort the items $i$ according to their Copeland score $s_C$:

$$s_C(i) = \frac{1}{N} \sum_{k=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{n} \mathbb{I}[\sigma_k(i) < \sigma_k(j)]$$

which counts the number of pairwise victories of item $i$ over the other items $j \neq i \Rightarrow \mathcal{O}(n^2 N)$ complexity.

# Statistical Ranking Aggregation [Korba et al., 2017]

## Probabilistic Modeling

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

where $P$ distribution on $\mathfrak{S}_n$.

# Statistical Ranking Aggregation [Korba et al., 2017]

## Probabilistic Modeling

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

where $P$ distribution on $\mathfrak{S}_n$.

## Definition

A **Kemeny median** of $P$ is solution of:

$$\sigma_P^* = \underset{\sigma \in \mathfrak{S}_n}{\operatorname{argmin}} L_P(\sigma), \tag{1}$$

where $L_P(\sigma) = \mathbb{E}_{\Sigma \sim P}[d(\sigma, \Sigma)]$ is **the risk** of $\sigma$.

**Question:** Can we exhibit some conditions on $P$ so that solving (1) is tractable?

# Exact Solutions [Korba et al., 2017]

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

# Exact Solutions [Korba et al., 2017]

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

Strict Stochastic Transitivity (**SST**): $(p_{i,j} \neq 1/2 \ \forall i, j)$

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

# Exact Solutions [Korba et al., 2017]

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

Strict Stochastic Transitivity (**SST**): $(p_{i,j} \neq 1/2 \ \forall i, j)$

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

Low-Noise/**NA**$(h)$ for $h > 0$ ([Audibert and Tsybakov, 2007]):

$$\min_{i<j} |p_{i,j} - 1/2| \geq h.$$

# Exact Solutions [Korba et al., 2017]

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

Strict Stochastic Transitivity (**SST**): $(p_{i,j} \neq 1/2 \ \forall i, j)$

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

Low-Noise/**NA**$(h)$ for $h > 0$ ([Audibert and Tsybakov, 2007]):

$$\min_{i<j} |p_{i,j} - 1/2| \geq h.$$

## Our result

Suppose $P$ satisfies **SST and NA**$(h)$ for a given $h > 0$. Then with overwhelming probability $1 - \frac{n(n-1)}{4} e^{-\alpha_h N}$:

# Exact Solutions [Korba et al., 2017]

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

Strict Stochastic Transitivity (**SST**): $(p_{i,j} \neq 1/2 \ \forall i, j)$

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

Low-Noise/**NA**$(h)$ for $h > 0$ ([Audibert and Tsybakov, 2007]):

$$\min_{i<j} |p_{i,j} - 1/2| \geq h.$$

## Our result

Suppose $P$ satisfies **SST and NA**$(h)$ for a given $h > 0$. Then with overwhelming probability $1 - \frac{n(n-1)}{4} e^{-\alpha_h N}$:

$\widehat{P}$ also verifies **SST**...

# Exact Solutions [Korba et al., 2017]

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

Strict Stochastic Transitivity (**SST**): $(p_{i,j} \neq 1/2 \ \forall i, j)$

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

Low-Noise/**NA**$(h)$ for $h > 0$ ([Audibert and Tsybakov, 2007]):

$$\min_{i<j} |p_{i,j} - 1/2| \geq h.$$

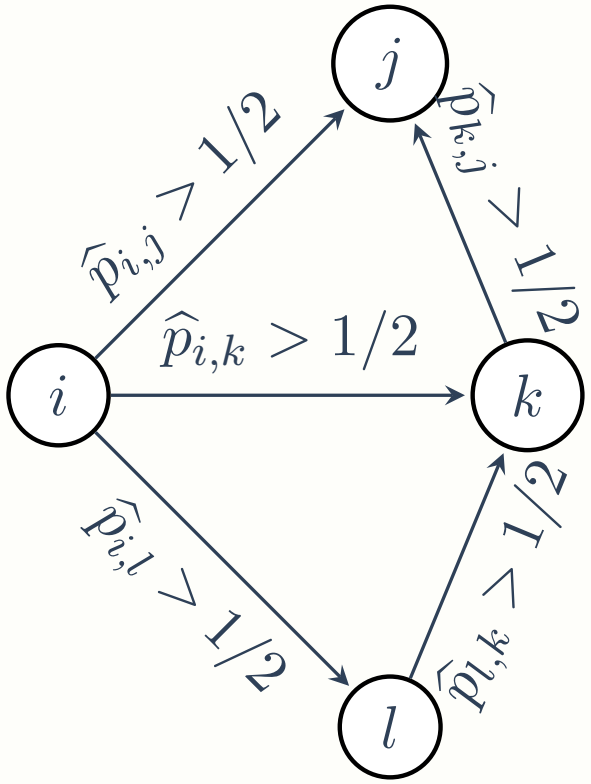
## Our result

Suppose $P$ satisfies **SST and NA**$(h)$ for a given $h > 0$. Then with overwhelming probability $1 - \frac{n(n-1)}{4} e^{-\alpha_h N}$:

$\widehat{P}$ also verifies **SST**...and the Kemeny median of $P$ is given by the empirical Copeland ranking:

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

# Graph of pairwise probabilities




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

$\Rightarrow$ sort the $i$'s by increasing input degree

# Outline

# Our Problem - Ranking Regression

**Goal**: Learn a predictive ranking rule :

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

which given a random vector $X$, predicts the permutation $\Sigma$ on the $n$ items.

**Performance**: Measured by the risk:

$$\begin{aligned} \mathcal{R}(s) &= \mathbb{E}_{X\sim\mu,\Sigma\sim P_X}\left[d_\tau\left(s(X),\Sigma\right)\right] \\ &= \mathbb{E}_{X\sim\mu}\left[\mathbb{E}_{\Sigma\sim P_X}\left[d_\tau\left(s(X),\Sigma\right)\right]\right] \\ &= \mathbb{E}_{X\sim\mu}\left[L_{P_X}(s(X))\right] \end{aligned}$$

# Our Problem - Ranking Regression

**Goal**: Learn a predictive ranking rule :

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

which given a random vector $X$, predicts the permutation $\Sigma$ on the $n$ items.

**Performance**: Measured by the risk:

$$\begin{aligned} \mathcal{R}(s) &= \mathbb{E}_{X\sim\mu,\Sigma\sim P_X}\left[d_\tau\left(s(X),\Sigma\right)\right] \\ &= \mathbb{E}_{X\sim\mu}\left[\mathbb{E}_{\Sigma\sim P_X}\left[d_\tau\left(s(X),\Sigma\right)\right]\right] \\ &= \mathbb{E}_{X\sim\mu}\left[L_{P_X}(s(X))\right] \end{aligned}$$

$\Rightarrow$ Ranking regression is an extension of ranking aggregation.

# Optimal Elements and Relaxation

## Assumption

For $X \in \mathcal{X}$, $P_X$ is **SST**: $\Rightarrow \sigma^*_{P_X} = \text{argmin}_{\sigma \in \mathfrak{S}_n} L_{P_X}(\sigma)$ is **unique**.

# Optimal Elements and Relaxation

## Assumption

For $X \in \mathcal{X}$, $P_X$ is **SST**: $\Rightarrow \sigma^*_{P_X} = \text{argmin}_{\sigma \in \mathfrak{S}_n} L_{P_X}(\sigma)$ is **unique**.

## Optimal elements

The predictors $s^*$ minimizing $\mathcal{R}(s)$ are the ones that maps any point $X \in \mathcal{X}$ to the **conditional** Kemeny median of $P_X$:

$$s^* = \underset{s \in \mathcal{S}}{\text{argmin}}\, \mathcal{R}(s) \quad \Leftrightarrow \quad s^*(X) = \sigma^*_{P_X}$$

# Optimal Elements and Relaxation

## Assumption

For $X \in \mathcal{X}$, $P_X$ is **SST**: $\Rightarrow \sigma^*_{P_X} = \text{argmin}_{\sigma \in \mathfrak{S}_n} L_{P_X}(\sigma)$ is **unique**.

## Optimal elements

The predictors $s^*$ minimizing $\mathcal{R}(s)$ are the ones that maps any point $X \in \mathcal{X}$ to the **conditional** Kemeny median of $P_X$:

$$s^* = \underset{s \in \mathcal{S}}{\text{argmin}}\, \mathcal{R}(s) \quad \Leftrightarrow \quad s^*(X) = \sigma^*_{P_X}$$

To minimize the risk $\mathcal{R}(s)$ approximately:

$$\sigma^*_{P_X} \text{ for any } X \Longrightarrow \sigma^*_{P_{\mathcal{C}}} \text{ for any } X \in \mathcal{C}$$

where $P_{\mathcal{C}}(\sigma) = \mathbb{P}[\Sigma = \sigma | X \in \mathcal{C}]$.

$\Rightarrow$ We develop Local consensus methods.

# Statistical Framework- ERM

Optimize a statistical version of the theoretical risk based on the training data $(X_k, \Sigma_k)$'s:

$$\min_{s \in \mathcal{S}} \widehat{\mathcal{R}}_N(s) = \frac{1}{N} \sum_{k=1}^{N} d_\tau(s(X_k),\ \Sigma_k)$$

where $\mathcal{S}$ is the set of measurable mappings.

# Statistical Framework- ERM

Optimize a statistical version of the theoretical risk based on the training data $(X_k, \Sigma_k)$'s:

$$\min_{s \in \mathcal{S}} \widehat{\mathcal{R}}_N(s) = \frac{1}{N} \sum_{k=1}^{N} d_\tau(s(X_k),\ \Sigma_k)$$

where $\mathcal{S}$ is the set of measurable mappings.

$\Rightarrow$ We consider a subset $\mathcal{S}_\mathcal{P} \subset \mathcal{S}$:

- rich enough so that $\inf_{s \in \mathcal{S}_\mathcal{P}} \mathcal{R}(s) - \inf_{s \in \mathcal{S}} \mathcal{R}(s)$ is "small"
- ideally appropriate for greedy optimization.

$\Rightarrow$ $\mathcal{S}_\mathcal{P}$= space of piecewise constant ranking rules

# Our results

## Rates of convergence

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

## Approximation Error

Suppose that:

There 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:

$$\mathcal{R}(s^*_{\mathcal{P}}) - \mathcal{R}(s^*) \leq M.\delta_{\mathcal{P}}$$

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

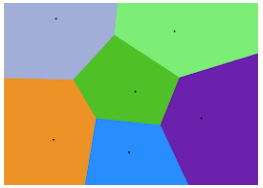
# Outline

# Partitioning Methods

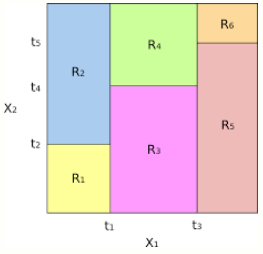
**Goal:** Generate partitions $\mathcal{P}_N$ from the training data $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$.

Two methods are investigated:

- k-nearest neighbor (Voronoi partitioning)



- decision tree (Recursive partitioning)

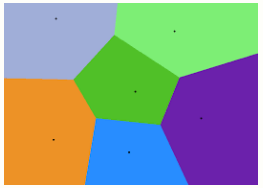


# Partitioning Methods

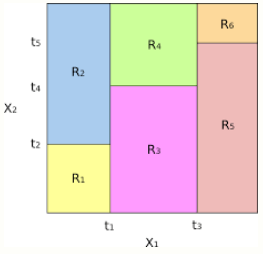
**Goal:** Generate partitions $\mathcal{P}_N$ from the training data $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$.

Two methods are investigated:

- k-nearest neighbor (Voronoi partitioning)



- decision tree (Recursive partitioning)




For $\mathcal{C} \in \mathcal{P}_N$, consider its empirical distribution:

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

# Final Labels in Practice

- If $\widehat{P}_{\mathcal{C}}$ is SST, compute $\sigma^*_{\widehat{P}}$ with Copeland method based on the $\widehat{p}_{i,j}$'s

# Final Labels in Practice

- If $\widehat{P}_{\mathcal{C}}$ is SST, compute $\sigma^*_{\widehat{P}}$ with Copeland method based on the $\widehat{p}_{i,j}$'s
- Else, compute $\widetilde{\sigma}^*_{\widehat{P}}$ with empirical Borda count ([Jiang et al., 2011])

$$\widetilde{\sigma}^*_{\widehat{P}}(i) = \frac{1}{N}\sum_{k=1}^{N} \Sigma_k(i) \quad \text{for } 1 \leq i \leq n$$

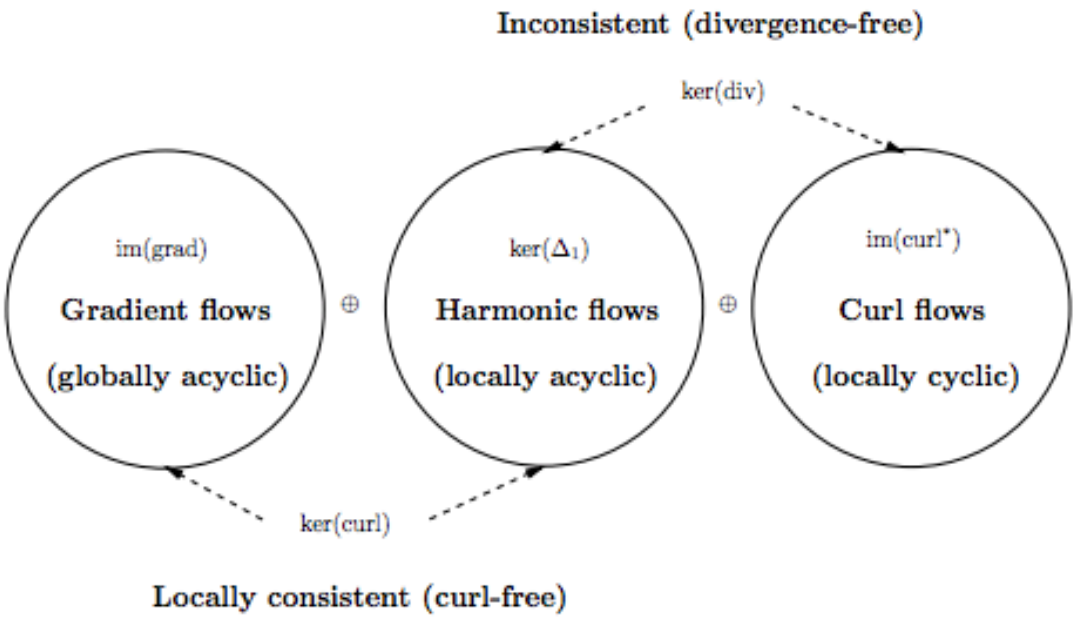



FIGURE 2. Hodge/Helmholtz decomposition of pairwise rankings

# K-Nearest Neigbors Algorithm

1. Fix $k \in \{1, \ldots, N\}$ and a query point $x \in \mathcal{X}$
2. Sort $(X_1, \Sigma_1), \ldots, (X_N, \Sigma_N)$ by increasing order of the distance to $x$ : $\|X_{(1,N)} - x\| \leq \ldots \leq \|X_{(N,N)} - x\|$
3. Consider next the empirical distribution calculated using the $k$ training points closest to $x$

$$\widehat{P}(x) = \frac{1}{k} \sum_{l=1}^{k} \delta_{\Sigma_{(l,N)}}$$

and compute the pseudo-empirical Kemeny median, yielding the $k$-NN prediction at $x$:

$$s_{k,N}(x) \overset{def}{=} \widetilde{\sigma}^*_{\widehat{P}(x)}.$$

$\Rightarrow$ We recover the classical bound $\mathcal{R}(s_{k,N}) - \mathcal{R}^* = \mathcal{O}(\frac{1}{\sqrt{k}} + \frac{k}{N})$

# Decision Tree

Split recursively the feature space $\mathcal{X}$ to minimize some impurity criterion.

Analog to Gini criterion in multiclassification: m classes, $f_i$ proportion of class $i \rightarrow I_G(\mathcal{C}) = \sum_{i=1}^{m} f_i(\mathcal{C})(1 - f_i(\mathcal{C}))$

# Decision Tree

Split recursively the feature space $\mathcal{X}$ to minimize some impurity criterion.

Analog to Gini criterion in multiclassification: m classes, $f_i$ proportion of class $i \to I_G(\mathcal{C}) = \sum_{i=1}^{m} f_i(\mathcal{C})(1 - f_i(\mathcal{C}))$

Here, for a cell $\mathcal{C} \in \mathcal{P}_N$:

- Impurity [Alvo and Philip, 2014]:

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

(ordering $n$ elements can be seen as $\binom{n}{2}$ classification tasks) which is tractable and satisfies the double inequality

$$\widehat{\gamma}_{\widehat{P}_{\mathcal{C}}} \leq \min_{\sigma \in \mathfrak{S}_n} L_{\widehat{P}_{\mathcal{C}}}(\sigma) \leq 2\widehat{\gamma}_{\widehat{P}_{\mathcal{C}}}.$$

- Terminal value : Compute the pseudo-empirical median of a cell $\mathcal{C}$:

$$s_{\mathcal{C}}(x) \overset{def}{=} \widetilde{\sigma}^*_{\widehat{P}_{\mathcal{C}}}.$$

# Simulated Data

- We generate two explanatory variables, varying their nature (numerical, categorical) $\Rightarrow$ Setting 1/2/3
- We generate a partition of the feature space
- On each cell of the partition, a dataset of full rankings is generated, varying the distribution (constant, Mallows with $\neq$ dispersion): $D_0/D_1/D_2$

| $D_i$ | Setting 1 | | | Setting 2 | | | Setting 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | n=3 | n=5 | n=8 | n=3 | n=5 | n=8 | n=3 | n=5 | n=8 |
| $D_0$ | 0.0698* | 0.1290* | 0.2670* | 0.0173* | 0.0405* | 0.110* | 0.0112* | 0.0372* | 0.0862* |
| | 0.0473** | 0.136** | 0.324** | 0.0568** | 0.145** | 0.2695** | 0.099** | 0.1331** | 0.2188** |
| | (0.578) | (1.147) | (2.347) | (0.596) | (1.475) | (3.223) | (0.5012) | (1.104) | (2.332) |
| $D_1$ | 0.3475 * | 0.569* | 0.9405 * | 0.306* | 0.494* | 0.784* | 0.289* | 0.457* | 0.668* |
| | 0.307** | 0.529** | 0.921** | 0.308** | 0.536** | 0.862** | 0.3374** | 0.5714** | 0.8544** |
| | (0.719) | (1.349) | (2.606) | (0.727) | (1.634) | (3.424) | (0.5254) | (1.138) | (2.287) |
| $D_2$ | 0.8656* | 1.522* | 2.503* | 0.8305 * | 1.447 * | 2.359* | 0.8105* | 1.437* | 2.189* |
| | 0.7228** | 1.322** | 2.226** | 0.723** | 1.3305** | 2.163** | 0.7312** | 1.3237** | 2.252** |
| | (0.981) | (1.865) | (3.443) | (1.014) | (2.0945) | (4.086) | (0.8504) | (1.709) | (3.005) |

Table: Empirical risk averaged on 50 trials on simulated data.

(): Clustering +PL, *: K-NN, **: Decision Tree

# Outline

# Structured prediction approach

**Goal**: Learn a predictive ranking rule : $s : \mathcal{X} \to \mathfrak{S}_n$
The ranking regression/label ranking problem is then defined as:

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

# Structured prediction approach

**Goal**: Learn a predictive ranking rule : $s : \mathcal{X} \to \mathfrak{S}_n$
The ranking regression/label ranking problem is then defined as:

$$\min_{s:\mathcal{X}\to\mathfrak{S}_n} \mathcal{R}(s), \text{ with } \mathcal{R}(s) = \mathbb{E}_{X\sim\mu,\Sigma\sim P_X}\left[\Delta\left(s(X),\Sigma\right)\right]$$

Consider a family of loss functions based on some ranking embedding $\phi : \mathfrak{S}_n \to \mathcal{F}$ that maps the permutations $\sigma \in \mathfrak{S}_n$ into a Hilbert space $\mathcal{F}$:

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

Motivation:

- Kendall's tau and Hamming distances can be written with Kemeny and Permutation matrices embeddings respectively

# Structured prediction approach

$$\min_{s\,:\,\mathcal{X}\to\mathfrak{S}_n} \mathcal{R}(s), \quad \text{with } \mathcal{R}(s) = \mathbb{E}_{X\sim\mu,\Sigma\sim P_X}\left[\Delta\left(s(X),\Sigma\right)\right] \quad (2)$$

and

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

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

- **Step 1 - Surrogate problem**: Solve an empirical version of (2) by replacing $\Delta$ with:

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

$\Longrightarrow \widehat{g} : \mathcal{X} \to \mathcal{F}$

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

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

# Ranking Embeddings

[Ciliberto et al., 2016] have proven consistency results under some assumptions on the loss $\Delta$/the mapping $\phi$, which apply to:

- Kendall's $\tau$ distance:

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

$$\rightarrow \phi(\sigma) = \begin{pmatrix} \vdots \\ sign(\sigma(i) - \sigma(j)) \\ \vdots \end{pmatrix}_{1 \leq i < j \leq n} \in \mathbb{R}^{n(n-1)/2}$$

- Hamming distance:

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

$$\rightarrow \phi(\sigma) = (\mathbb{I}\{\sigma(i) = j\})_{i,j=1,\ldots,n} \in \mathbb{R}^{n \times n}$$

- **consistency holds, but still the pre-image problem is hard**

# Structured prediction results

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

| | authorship | glass | iris | vehicle | vowel | wine |
|---|---|---|---|---|---|---|
| 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** |

# Outline

# Conclusion

Ranking data presents great and interesting challenges:

- Most of the maths from euclidean spaces cannot be applied
- But our intuitions still hold
- Based on our results on ranking aggregation, we develop a novel approach to ranking regression/label ranking
- Our contributions: theoretical results for this problem and new algorithms

## Openings:

- How to extend to incomplete rankings (+with ties)?


Alvo, M. and Philip, L. (2014).
Decision tree models for ranking data.
In *Statistical Methods for Ranking Data*, pages 199–222. Springer.


Audibert, J.-Y. and Tsybakov, A. (2007).
Fast learning rates for plug-in classifiers.
*Annals of statistics*, 35(2):608–633.


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., Dembczyński, K., and Hüllermeier, E. (2010).
Label ranking methods based on the Plackett-Luce model.
In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 215–222.


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


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.


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., Gaudel, R., and Jakubowicz, J. (2011).
On clustering rank data in the fourier domain.
In *ECML*.


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 *Proceeding of ICML 2016*.


Kondor, R. and Barbosa, M. S. (2010).
Ranking with kernels in Fourier space.
In *Proceedings of COLT'10*, pages 451–463.


Korba, A., Clémençon, S., and Sibony, E. (2017).
A learning theory of ranking aggregation.
In *Proceeding of AISTATS 2017*.


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


Mallows, C. L. (1957).
Non-null ranking models.

*Biometrika*, 44(1-2):114–130.


Plackett, R. L. (1975).
The analysis of permutations.
*Applied Statistics*, 2(24):193–202.


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


Shah, N. B. and Wainwright, M. J. (2015).
Simple, robust and optimal ranking from pairwise comparisons.
*arXiv preprint arXiv:1512.08949*.


Sibony, E., Clémençon, S., and Jakubowicz, J. (2015).
MRA-based statistical learning from incomplete rankings.
In *Proceeding of ICML*.


Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009).

Mining multi-label data.
In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

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

# US General Social Survey

Participants were asked to rank 5 aspects about a job: ”high income”, ”no danger of being fired”, ”short working hours”, ”chances for advancement”, ”work important and gives a feeling of accomplishment”.

- 18544 samples collected between 1973 and 2014.
- 8 individual attributes are considered: sex, race, birth cohort, highest educational degree attained, family income, marital status, number of children, household size
- plus 3 attributes of work conditions: working status, employment status, and occupation.

Results:
Risk of decision tree: 2,763 $\rightarrow$ Splitting variables:

1) occupation 2) race 3) degree