Robust Ordinal Embedding from Contaminated Relative Comparisons

Ke Ma[1,2], Qianqian Xu[3], Xiaochun Cao[1]

[1] SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] VIPL, Institute of Computing Technology, Chinese Academy of Sciences

November 14, 2018

Ordinal Embedding

Ordinal Information

distance(A, B) < distance(A, C)

distance(A, B) < distance(A, D)

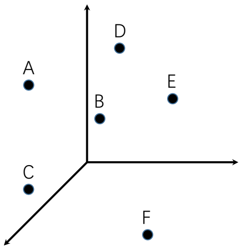
distance(B, C) > distance(B, D)

⋮

distance(D, A) > distance(D, E)

distance(B, C) < distance(B, F)

Metric Representation

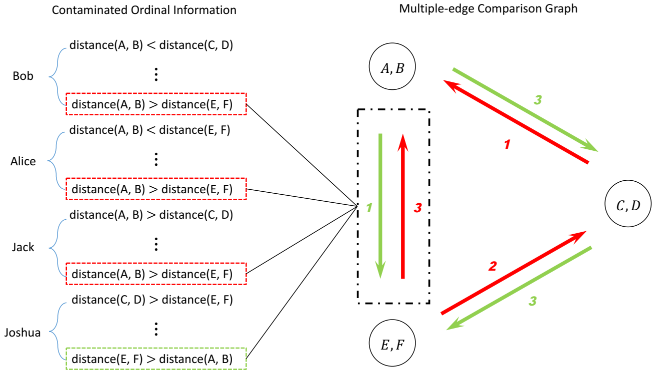


Suppose the relative comparisons, *e.g.* (i, j, l, k), are consistent with a low-dimensional embedding, our goal is to associate each item with a point $\boldsymbol{x} \in \mathbb{R}^p$ such that

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 < \|\boldsymbol{x}_l - \boldsymbol{x}_k\|_2.$$

Wait a minute

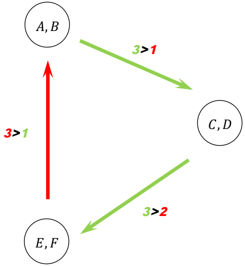
Actually, we only collect the contaminated relative comparisons from different annotators



Multi-stage Method

- The multiple-edges are aggregated, *e.g.* majority voting.

Condorcet's paradox !

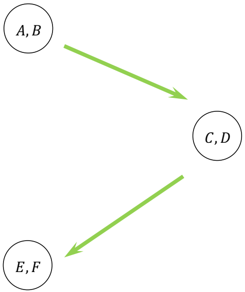


Cons:

- pruning the right comparison direction
- leading to the well-known 'Condorcet's paradox'

Multi-stage Method

- Maximum acyclic subgraph approximation.



Cons:

- NP-complete problem
- the available data would be rare.

Preliminaries

- object set $\mathcal{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n\}$
- similarity function $\zeta: \ \mathcal{O}^2 \to \mathbb{R}^+$
- relative comparison set

  $$\mathcal{C}_{\mathcal{U}} = \left\{ (i,j,l,k)_u \;\middle|\; \begin{array}{c} i,\ j,\ l,\ j \in [n],\ u \in \mathcal{U} \\ i \neq j,\ l \neq k,\ (i,j) \neq (l,k) \end{array} \right\}$$

  and the corresponded comparison graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{v_{ij} | i,j \in [n]\}$ and $\mathcal{E} = \{\boldsymbol{e}_c^{\mathcal{U}}, \boldsymbol{e}_{\bar{c}}^{\mathcal{U}}\}$.
- multiple edge $\boldsymbol{e}_c^{\mathcal{U}} = \{e_c^u, c_u \in \mathcal{C}_{\mathcal{U}}\}$, and the indicator y_c^u on e_c^u is $y_c^u = 1$ if e_c^u existed. Furthermore, the weight of $\boldsymbol{e}_c^{\mathcal{U}}$ is

  $$w_c = \sum_{u \in \mathcal{U}} [y_c^u = 1],\ c_u \in \mathcal{C}_{\mathcal{U}}$$

The Proposed Unified Framework

- Detecting the outliers in the edge set $\mathcal{E}$. A set of unknown variables $\boldsymbol{\gamma} = \{\gamma_c\} \in \mathbb{R}^{|\mathcal{E}|}$ indicate whether the edge $\boldsymbol{e}_c^{\mathcal{U}}$ is an outlier or not. The outlier detection task in $\mathcal{C}_{\mathcal{U}}$ thus becomes the problem of estimating $\boldsymbol{\gamma}$ with $\mathcal{G}$.

- Obtaining an embedding $\boldsymbol{X} \in \mathbb{R}^{p \times n}$. Given $\boldsymbol{e}_c^{\mathcal{U}} \in \mathcal{E}$ which represents the correct relative similarity measurement, we hope the embedding $\boldsymbol{X}$ satisfies

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 < \|\boldsymbol{x}_l - \boldsymbol{x}_k\|_2^2, \; c_u = (i, j, l, k)_u \in \mathcal{C}_{\mathcal{U}}.$$

A Linear Model

- Given an edge $\boldsymbol{e}_c^{\mathcal{U}} \in \mathcal{E}$, its corresponding direction indicator y_c is modeled as

$$\begin{aligned} y_c &= \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 - \|\boldsymbol{x}_l - \boldsymbol{x}_k\|_2^2 + \gamma_c + \varepsilon_c \\ &= d_{ij}^2 - d_{lk}^2 + \gamma_c + \varepsilon_c. \end{aligned} \tag{1}$$

- It is known that there is a map between the distance matrix $\boldsymbol{D}$ and the Gram matrix $\boldsymbol{G} = \{g_{ij}\} = \boldsymbol{X}^\top \boldsymbol{X}$ as

$$d_{ij} = g_{ii} - 2g_{ij} + g_{jj}$$

and (1) can be written as

$$\boldsymbol{y} = \boldsymbol{Z} \odot \boldsymbol{G} + \boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \tag{2}$$

where

$$\boldsymbol{Z} \odot \boldsymbol{G} = \boldsymbol{Z}\boldsymbol{g} = \boldsymbol{Z} \cdot vec(\boldsymbol{G}).$$

Robust Ordinal Embedding

$$\begin{aligned} \underset{\boldsymbol{G},\ \gamma}{\textit{minimize}} \quad & \mathcal{L}_{\boldsymbol{w}}(\boldsymbol{G}, \gamma) + \lambda \|\gamma\|_{1,\boldsymbol{w}} \\ \textit{subject to} \quad & \mathbf{rank}(\boldsymbol{G}) = p, \ \ \boldsymbol{G} \succeq 0 \end{aligned} \tag{3}$$

where

$$\begin{aligned} \mathcal{L}_{\boldsymbol{w}}(\boldsymbol{G}, \gamma) &= \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{Z} \odot \boldsymbol{G} - \gamma\|_{2,\boldsymbol{w}}^2 \\ &= \frac{1}{2} \sum_{\boldsymbol{e}_c^{\mathcal{U}} \in \mathcal{E}} w_c^2 (y_c - \gamma_c - d_{ij} + d_{lk})^2 \\ &= \frac{1}{2} \|\boldsymbol{W}\boldsymbol{y} - (\boldsymbol{W}\boldsymbol{Z}) \odot \boldsymbol{G} - \boldsymbol{W}\gamma\|_2^2 \end{aligned} \tag{4}$$

and

$$\|\gamma\|_{1,\boldsymbol{w}} = \sum_{\boldsymbol{e}_c^{\mathcal{U}} \in \mathcal{E}} w_c |\gamma_c| = \|\boldsymbol{W}\gamma\|_1 \tag{5}$$

Optimization

$$\begin{aligned} \mathcal{L}_{\boldsymbol{w}}(\boldsymbol{G},\gamma) &= \frac{1}{2}\left\|\boldsymbol{y}_{\boldsymbol{w}} - \boldsymbol{W}\boldsymbol{Z}\cdot\boldsymbol{g} - \boldsymbol{W}\gamma\right\|_2^2 \\ &= \frac{1}{2}\left\|\boldsymbol{y}_{\boldsymbol{w}} - \begin{bmatrix}\boldsymbol{W}\boldsymbol{Z} & \\ & \boldsymbol{W}\end{bmatrix}\begin{pmatrix}\boldsymbol{g}\\ \gamma\end{pmatrix}\right\|_2^2 \\ &= \frac{1}{2}\left\|\boldsymbol{y}_{\boldsymbol{w}} - \boldsymbol{A}\cdot\beta\right\|_2^2 := f(\beta) \end{aligned} \tag{6}$$

$$\lambda\|\boldsymbol{\gamma}\|_{1,\boldsymbol{w}} = \lambda\left\|\begin{bmatrix}\mathbf{0} & \\ & \boldsymbol{W}\end{bmatrix}\begin{pmatrix}\boldsymbol{g}\\ \gamma\end{pmatrix}\right\|_1 = \lambda\|\boldsymbol{B}\cdot\boldsymbol{\beta}\|_1 := g(\boldsymbol{\beta}). \tag{7}$$

With (6), (7) and ignoring the constraints on $\boldsymbol{G}$, (3) is equivalent to a *LASSO* formulation

$$\arg\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) := f(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) \tag{8}$$

Let $\mathcal{F}$ be the feasible set of (8) and suppose $\boldsymbol{G}^*$ is a optimal solution of (3), it holds that $\boldsymbol{G}^* \in \mathcal{F}$. As a consequence, we come to a semi-definite programming with rank equality constraint

$$\begin{aligned} \textit{find} \quad & \boldsymbol{G},\ \gamma \\ \textit{subject to} \quad & \boldsymbol{G},\ \gamma \in \mathcal{F},\ \boldsymbol{G} \succeq 0,\ \mathbf{rank}(\boldsymbol{G}) = p. \end{aligned} \tag{9}$$

- Solving a SDP with rank equality constraints like (9) is notoriously difficult.
- Here we adopt the ran-reduction for semi-definite programming.

Solving the SDP

First, we solve the following optimization problem

$$\begin{aligned} \textit{find} \quad & \boldsymbol{G},\ \gamma \\ \textit{subject to} \quad & \boldsymbol{G},\ \gamma \in \mathcal{F},\ \boldsymbol{G} \succeq 0. \end{aligned} \tag{10}$$

For any $L > 0$, consider the following quadratic approximation of $F(\boldsymbol{\beta}) := f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$ at a given point $\boldsymbol{\beta}_0$:

$$Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}_0) = f(\boldsymbol{\beta}_0) + \langle \boldsymbol{\beta} - \boldsymbol{\beta}_0, \nabla f(\boldsymbol{\beta}_0) \rangle + \frac{L}{2} \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|^2 + g(\boldsymbol{\beta}) \tag{11}$$

which admits a unique minimizer

$$P_L(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}} \left\{ g(\boldsymbol{\beta}) + \frac{L}{2} \left\| \boldsymbol{\beta} - \left(\boldsymbol{\beta}_0 - \frac{1}{L} \nabla f(\boldsymbol{\beta}_0) \right) \right\|^2 \right\}.$$

$$\boldsymbol{G}_{t+1} = \Pi_{\mathcal{S}_+^n} \left(\boldsymbol{G}_t - \frac{1}{L}\ \nabla_{\boldsymbol{G}}\ f(\boldsymbol{\beta}_t) \right) \tag{12a}$$

$$\gamma_{t+1} = \mathcal{T}_{\boldsymbol{\mu}} \left(\gamma_t - \frac{1}{L}\ \nabla_{\gamma}\ f(\boldsymbol{\beta}_t) \right) \tag{12b}$$

Rank Reduction

Here we reformulate (10) as a standard SDP

$$\begin{aligned} \textit{minimize} \quad & \|\boldsymbol{\gamma}\|_{1,\boldsymbol{w}} \\ \textit{subject to} \quad & \langle \boldsymbol{G}, \boldsymbol{A}_c \rangle + \gamma_c = y_c,\ \boldsymbol{e}_{\boldsymbol{c}}^{\mathcal{U}} \in \mathcal{E}, \\ & \boldsymbol{G} \succeq 0, \end{aligned} \tag{13}$$