iSplit LBI: Individualized Partial Ranking with Ties via Split LBI

Qianqian Xu, Xinwei Sun, Zhiyong Yang, Xiaochun Cao, Qingming Huang and Yuan Yao
ICT, CAS; MSRA; IIE, CAS; UCAS;BDKM, CAS; Dep. of Math, HKUST;PCL.

SCAN ME

Introduction

- The majority work of crowd-sourced preference learning either focuses on instance-wise preference learning or assumes that the candidates are comparable.
- Introducing pair-wise comparisons into the preferential learning helps to avoid calibration bias.
- Facing pair-wise data , the tie results are ubiquitous, since the annotators might abstain from a decision whenever he/she thinks that none of the objects in a given pair can win the other easily.
- Seeing the issues mentioned above, we propose a unified framework, called `iSplitLBI`, for personalized partial ranking, tie state recognition and abnormal user detection.

Methodology

- **The comparison graph** $G = (V, E)$. Let $V = \{1, 2, \ldots, n\}$: the vertex set of n items, $E = \{(u, i, j) : i, j \in V$ are compared by user $u, u \in U\}$. The annotation results $y_{ij}^u = 1$ means u prefers i to j and $y_{ij}^u = -1$ otherwise.
- **Probabilistic Model**
  - We assume that there is a true list of personalized score : $s^u = [s_1^u, \cdots, s_{n_{u_i}}^u], \forall u$
  - We assume that y_{ij}^u is produced by comparing the score difference $s_i^u - s_j^u$ with the threshold λ^u, yielding to the following decision rule.

$$y_{ij}^u = \begin{cases} 1, & s_i^u - s_j^u + \epsilon_{ij}^u > \lambda^u; \\ -1, & s_i^u - s_j^u + \epsilon_{ij}^u \leq -\lambda^u; \\ 0, & \text{else}. \end{cases} \tag{1}$$

  - ϵ_{ij}^u is a random noise whic has a c.d.f $\Phi(t)$.
  - The probability to observe y_{ij}^u becomes:

$$P\{y_{ij}^u\} = \left[P_{1,ij}^u\right]^{1\{y_{ij}^u=1\}} \left[P_{0,ij}^u\right]^{1\{y_{ij}^u=0\}} \left[P_{-1,ij}^u\right]^{1\{y_{ij}^u=-1\}}.$$

- **Parameter Decomposition**:
  - We assume the majority of participants share a common preference interest and behave rationally, while deviations from that exist but are sparse.

$$s^u = c_s + p_s^u, \ \lambda^u = c_\lambda + p_\lambda^u, \ y_{ij}^u = (c_{s_i} + p_{s_i}^u) - (c_{s_j} + p_{s_j}^u) + \epsilon_{ij}^u. \tag{2}$$

  - We use group lasso regularization to force the personalized parameters to be sparse in a user-specific manner.
  - Variable splitting: Since practically personalized parameters might not obey the sparsity rule ,we transfer the sparsity constraints to the auxiliary parameters Γ and penalize the distance between Γ and P.
- **Overall Objective Function**

$$\begin{aligned} \min_{\Theta} \ & \overbrace{\sum_u \mathcal{L}(\mathcal{O}^u, \mathcal{Y}^u | s^u, \lambda^u)}^{\text{log-likelihood}} + \overbrace{\mathcal{S}_\nu(\Gamma, P)}^{\text{Variable Splitting}} + \overbrace{\mathcal{J}(\Gamma_s, \Gamma_\lambda)}^{\text{Group Lasso Penalty}} \\ s.t. \ & s^u = c_s + \Gamma_s^u, \ \lambda^u = c_\lambda + \Gamma_\lambda^u, \\ & \lambda^u \geq \delta, \ c_\lambda \geq \delta. \end{aligned} \tag{3}$$

- In the constraints we use $\lambda^u \geq \delta$, $c_\lambda \geq \delta$, where $\delta > 0$, as closed and convex approximations of the positivity constraints $\lambda^u > 0, c_\lambda > 0$. The benefit to employ the relaxations are two-fold: 1) The closed domain constraints induce closed-form solution; 2) The threshold δ improves the quality of the solution to avoid ill-conditioned cases being too close to zero.

Optimization

Instead of directly solving the above-mentioned problem, we adopt the Split Linearized Bregman Iterations which we call individualized Split LBI (`iSplitLBI`), which gives rise to a regularization path where both the model parameters and hyper-parameters are simultaneously evolved. The $k+1$-th iteration of such path is given as:

$$\begin{pmatrix} \mathbf{c}_s^{u,k+1} \\ c_\lambda^{u,+1} \end{pmatrix} = \mathbf{Prox}_{J_c}\left(\begin{pmatrix} \mathbf{c}_s^{u,k} \\ c_\lambda^{u,k} \end{pmatrix} - \kappa\alpha_k \nabla_c \mathcal{L}(\Theta^k)\right), \forall u \in \mathcal{U} \tag{4a}$$

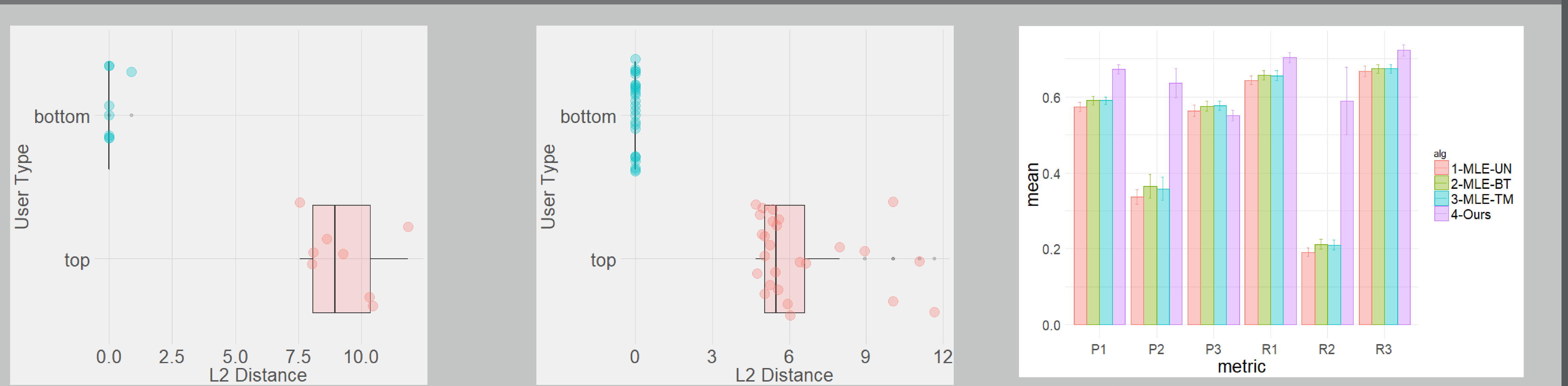
$$\begin{pmatrix} P_s^{k+1} \\ P_\lambda^{k+1} \end{pmatrix} = \mathbf{Prox}_{J_P}\left(\begin{pmatrix} P_s^k \\ P_\lambda^k \end{pmatrix} - \kappa\alpha_k \nabla_P \mathcal{L}(\Theta^k)\right), \tag{4b}$$

$$\begin{pmatrix} Z_s^{k+1} \\ Z_\lambda^{k+1} \end{pmatrix} = \begin{pmatrix} Z_s^k \\ Z_\lambda^k \end{pmatrix} - \alpha_k \nabla_\Gamma \mathcal{L}(\Theta^k), \tag{4c}$$

$$\begin{pmatrix} \Gamma_s^{k+1} \\ \Gamma_\lambda^{k+1} \end{pmatrix} = \kappa \cdot \mathbf{Prox}_J\left(\begin{pmatrix} Z_s^k \\ Z_\lambda^k \end{pmatrix}\right), \tag{4d}$$

where the proximal h is defined by $\mathbf{Prox}_h(z) = \arg\min_x \|z - x\|^2/2 + h(x)$.
The initial choice $c_\lambda^{u,0} = 1 \in \mathbb{R}^1$, $\mathbf{c}_s^{u,0} = 0 \in \mathbb{R}^p$, $P_s^0 = Z_s^0 = 0 \in \mathbb{R}^{U \times p}$, $P_\lambda^0 = Z_\lambda^0 = 0 \in \mathbb{R}^U$, parameters $\kappa > 0, \ \alpha > 0, \ \nu > 0$. The $J_c(c_\lambda)$ and $J_P(P_\lambda)$ are denoted as the indicator function for the set $c_\lambda \geq \delta$ and $\lambda^u \geq \delta$ respectively (an indicator function of a set is 0 when the input variable is in the set, otherwise it is $+\infty$).

Experiments



(a) Age dataset (b) College dataset (c) Fine-grained comparison

Figure: (a)-(b) The L_2 distance between individualized ranking scores and common ranking scores of selected users on age and college datasets. (c) Fine-grained comparison on college dataset. P1, P2, P3 represent the precision for class -1, 0, 1, respectively; while R1, R2, R3 represent the corresponding recalls, respectively.

(a) Age: Micro-F1 / (b) Age: Macro-F1

types	algorithms	min	median	max	std	min	median	max	std
α-cut	LRLasso	.428	.443	.458	.008	.327	.358	.381	.016
	LRRidge	.422	.443	.457	.008	.330	.364	.387	.015
	SVMlasso	.422	.442	.463	.010	.331	.355	.371	.013
	SVMRidge	.424	.443	.457	.008	.330	.351	.379	.014
	LSLasso	.423	.441	.455	.008	.335	.361	.383	.014
	LSRidge	.426	.442	.464	.009	.335	.365	.398	.014
	SVRlasso	.418	.431	.449	.008	.333	.364	.397	.014
	SVRRidge	.422	.432	.450	.007	.337	.367	.381	.012
MLE	Uni.	.692	.705	.738	.012	.589	.606	.641	.012
	BT	.731	.741	.755	.008	.599	.628	.647	.012
	TM	.728	.739	.756	.008	.603	.623	.647	.012
Ours	Ours	.765	**.779**	.791	.007	.680	**.694**	.712	.010

(c) Col:Micro-F1 / (d) Col: Macro-F1

types	algorithms	min	median	max	std	min	median	max	std
α-cut	LRLasso	.318	.350	.408	.026	.328	.349	.367	.011
	LRRidge	.325	.352	.408	.023	.323	.348	.364	.010
	SVMlasso	.325	.343	.404	.029	.333	.345	.371	.009
	SVMRidge	.327	.354	.402	.025	.327	.345	.365	.010
	LSLasso	.305	.320	.377	.016	.331	.342	.362	.010
	LSRidge	.331	.345	.403	.023	.334	.345	.365	.009
	SVRlasso	.306	.328	.378	.018	.327	.346	.363	.010
	SVRRidge	.323	.348	.402	.023	.323	.350	.361	.012
MLE	Uni.	.521	.536	.550	.009	.482	.496	.514	.009
	BT	.539	.552	.565	.008	.496	.513	.526	.010
	TM	.539	.551	.565	.008	.496	.511	.526	.010
Ours	Ours	.637	**.649**	.663	.008	.608	**.645**	.674	.016

Table: Comparison Results on Age and College Dataset

https://qianqianxu010.github.io
xuqianqian@ict.ac.cn