Who Likes What?

SplitLBI in Exploring Preferential Diversity of Ratings

Qianqian Xu, Jiechao Xiong, Zhiyong Yang, Xiaochun Cao, Qingming Huang and Yuan Yao

Institute of Computing Technology, CAS
Tencent AI Lab
Institute of Information Engineering, CAS
University of Chinese Academy of Sciences
Key Lab. of BDKM, CAS
Peng Cheng Lab.
Department of Mathematics, HKUST

Outline

Intro

Finding your favorites out of Overloaded Choices

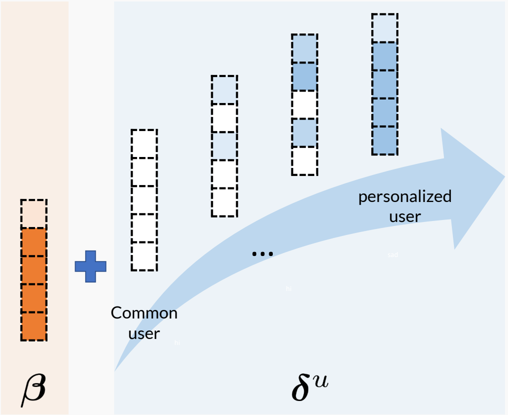
- In an era of data deluge, people often confront with the “information overload”.
- The ubiquitous data flow of user behavior provides us a plethora of preference revelation of participants
- Learning proper utility/ranking score functions facilitates decision-making efficiency.

Related Work

- **Learning to Rank Methods**: learns common ranking functions, not a natural fit for individual bias.
- **Personalized Learning Methods**: could capture user-specific understandings but ignore the hierarchy of personality where different users owns different strength of individualized bias.

What we do in a nutshell

- A hierarchical model which learns both the common or social preference functions
- A method to estimate the regularization path of the user-specific parameters at different sparsity levels.
- An acceleration method which allows an almost linear speedup with synchronized parallelization for large-scale data.



Methods

Problem Description

Data

- **Comparisions**: Comparison graph $G = (V; E)$, where $V = \{1, 2, \ldots, n\}$ be the vertex set consists of items and $E = \{(u, i, j) : i, j \in V, u \in U\}$ be the sets consists of annotation triplets.

- **Annotations**: A user u provides his/her preference between choice i and j with $y_{ij}^u = 1 \Leftrightarrow i \succeq^u j$, and $y_{ij}^u = -1$ otherwise.

Goal

To learn a preference model to predict the personalized annotations y_{ij}^u, taking into account both the **common consensus** and **users' diversity**.

Model

$$y_{ij}^u = (\boldsymbol{X}_i - \boldsymbol{X}_j)^\top(\boldsymbol{\beta} + \boldsymbol{\delta}^u) + \varepsilon_{ij}^u,\ \varepsilon_{ij}^u \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2). \tag{1}$$

variables

- $\boldsymbol{\beta}$ captures the popular opinion.
- $\boldsymbol{\delta}^u$ captures the personalized deviation from the popular point-of-view conveyed by $\boldsymbol{\beta}$.
- Under a linear model, $\boldsymbol{X}_i^\top(\boldsymbol{\beta} + \boldsymbol{\delta}^{\boldsymbol{u}})$ captures the personalized score from u.
- $(\boldsymbol{X}_i - \boldsymbol{X}_j)^\top(\boldsymbol{\beta} + \boldsymbol{\delta}^{\boldsymbol{u}})$ naturally suggests the preference from u.
- ε_{ij}^u captures the randomness in sampling which is of zero mean and variance σ^2.

Matrix Form

One reformulate Eq.(1) as the following form:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\omega} + \boldsymbol{\varepsilon}, \tag{2}$$

variables

- $\boldsymbol{\omega} = [\boldsymbol{\beta}, \boldsymbol{\delta}] \in \mathbb{R}^{d(1+|U|)}$ concatenates all the parameters from $\boldsymbol{\beta}$ and $\boldsymbol{\delta}^u$.
- $\boldsymbol{\varepsilon}$ concatenates all $\varepsilon^u_{i,j}$ in a similar way as $\boldsymbol{\omega}$.
- $\boldsymbol{X}$ is the corresponding matrix for the linear operator $\mathcal{X} : \mathbb{R}^{d(1+|U|)} \rightarrow \mathbb{R}^E$ such that $\mathcal{X}(w(u,i,j)) = (\boldsymbol{X}_i^\top \boldsymbol{\beta} + \boldsymbol{X}_i^\top \boldsymbol{\delta}^u) - (\boldsymbol{X}_j^\top \boldsymbol{\beta} + \boldsymbol{X}_j^\top \boldsymbol{\delta}^u)$

Remark

This model can be straightforwardly extended to multi-level models with more than two levels, by considering hierarchies of user types for example, which are particularly appropriate for research designs where user information are organized at more than two levels. Another extension is to the family of generalized linear models.

Remark

After obtaining these two sets of parameters, we can not only predict the preference for seen items rated by different users but also solve the cold-start problem in the following sense. When a new item (which have not yet received any judgments from the community) comes, given the low-level feature $\boldsymbol{x}_{new}$ *we can use a linear function* $\boldsymbol{x}_{new}^\top(\boldsymbol{\beta} + \boldsymbol{\delta}^u)$ *to predict his preference score. Besides, when a new active user comes, we can use the common score* $f(\boldsymbol{x}) = \boldsymbol{x}^\top\boldsymbol{\beta}$ *to predict the user's preference.*

Objective Function

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\gamma}) = \frac{1}{2m}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\omega}\|_2^2 + \frac{1}{2\nu}\|\boldsymbol{\omega} - \boldsymbol{\gamma}\|_2^2. \tag{3}$$

- Here we adopt the squared loss for prediction, which yields the loss function
$$\ell(\boldsymbol{y}, \boldsymbol{\omega}) = \frac{1}{2m}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\omega}||_2^2.$$
- Since not all the users hold significant deviation against the popular opinion, it is natural to find sparse solutions for $\boldsymbol{\delta}$.
- We introduce a auxiliary variable $\boldsymbol{\gamma}$ and employ a variable splitting scheme by encouraging the proximity regularization $\frac{1}{2\nu}\|\boldsymbol{\omega} - \boldsymbol{\gamma}\|_2^2$, which allows weak signals to come in to the model.
- Together with the SplitLBI algorithm, we can obtain a regularization path for the sparse $\boldsymbol{\delta}^u$s and thus find out solutions with different strength of personality for different users.

Update Rules

According to spliltLBI, we adopt the following rule to update $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$:

$$\boldsymbol{z}^{k+1} = \boldsymbol{z}^{k} - \alpha \nabla_{\boldsymbol{\gamma}} L(\boldsymbol{\omega}^{k}, \boldsymbol{\gamma}^{k}), \tag{4a}$$

$$\boldsymbol{\gamma}^{k+1} = \kappa \cdot \mathrm{prox}_{||\cdot||_1}(\boldsymbol{z}^{k+1}), \tag{4b}$$

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^{k} - \kappa\alpha \nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}^{k}, \boldsymbol{\gamma}^{k}), \tag{4c}$$

where we used the proximal mapping of ℓ_1:

$$\begin{aligned} \mathrm{prox}_{||\cdot||_1}(z) = & \underset{\boldsymbol{v} \in \mathbb{R}^{d(1+|U|)}}{\operatorname{argmin}} \left(\frac{1}{2} \|\boldsymbol{v} - \boldsymbol{z}\|^2 + ||\boldsymbol{v}||_1 \right), \\ =: & \; \boldsymbol{Shrinkage}(\boldsymbol{z}). \end{aligned} \tag{5}$$

Algorithm 1 `Sparse regularization path`

Input: Data $(\boldsymbol{X}, \boldsymbol{y})$, damping factor κ, step size α.
Initialization: $\gamma^0 = 0, z^0 = 0, t^0 = 0$
$\boldsymbol{H} \leftarrow (\nu \boldsymbol{X}^T \boldsymbol{X} + m\boldsymbol{I})^{-1} \boldsymbol{X}^\top$;
for $k = 0, \ldots, K$ **do**

$$\boldsymbol{z}^{k+1} \leftarrow \boldsymbol{z}^k + \alpha \boldsymbol{H}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\gamma}^k);$$
$$\boldsymbol{\gamma}^{k+1} \leftarrow \kappa \cdot \mathbf{Shrinkage}(\boldsymbol{z}^{k+1});$$
$$t^{k+1} \leftarrow (k+1)\alpha;$$

end for
Output: Solution path $\{t^k, \gamma^k\}_{k=0,1,\ldots,K}$.

Dynamics

From a dynamical point-of-view, if we regard the stepsize as $\alpha = \Delta t$ a discrete time difference, the update rules can be considered as a discretization of the dynamics known as the *inverse-scale spaces* with an infinitesimal stepsize:

$$\frac{d\boldsymbol{z}^t}{dt} = -\nabla_{\gamma}\mathcal{L}(\boldsymbol{\omega}^t, \boldsymbol{\gamma}^t), \tag{7}$$

$$\boldsymbol{z}^t - \frac{\boldsymbol{\gamma}^t}{\kappa} \in \partial||\boldsymbol{\gamma}^t||_1, \tag{8}$$

$$\frac{d\boldsymbol{\omega}^t}{dt} = -\kappa \cdot \nabla_{\boldsymbol{\omega}}\mathcal{L}(\boldsymbol{\omega}^t, \boldsymbol{\gamma}^t). \tag{9}$$

According the properties of the *inverse-scale spaces*, the time $\boldsymbol{\delta}^u$ jump out from zero decreases in proportion to the strength of the personality of u. In this way we find a good hierarchy of user personality in the model with the sparsity of $\boldsymbol{\delta}^u$.

- The regularization path of SplitLBI provides automatic tuning for the degree of sparsity.
- However, without a stopping-time controlling mechanism, we have $t \to \infty$, where the dynamics might encounter over-fitting models when noises exist.

Cross Validation

- Given the training data, fix κ and α, then split the data $\mathcal{S}$ into $\mathcal{S}_1, \cdots, \mathcal{S}_K$, where $\mathcal{S}_i \cup \mathcal{S}_j = \phi, i \neq j, \bigcup_{i=1}^{K} \mathcal{S}_i = \mathcal{S}$.
- **for** $k = 1, \ldots, K$ **do**
  1. Run SplitLBI on the training data $\mathcal{S} \backslash \mathcal{S}_k$ to get the solution path.
  2. For pre-decided parameter list of t, use a linear interpolation to get $\boldsymbol{\gamma}^t$.
  3. Use the estimator $\boldsymbol{\gamma}^t$ to predict on $\mathcal{S}_k$, and then compute prediction error.

  end for
- Return the optimal t_{cv} with minimal average prediction error.

- Let the support set of $\boldsymbol{\gamma}$ be defined as $\mathrm{Supp}(\boldsymbol{\gamma}) = \{(i,j) : \boldsymbol{\gamma}_{ij} \neq 0\}$.
- We can naturally decompose $\boldsymbol{\omega}$ into $\boldsymbol{\omega}_{\mathrm{supp}(\gamma)} + \boldsymbol{\omega}_{\mathrm{supp}(\gamma)^{\perp}}$. Here $\boldsymbol{\omega}_{\mathrm{A}}$ denotes the entry-wise projection onto the set A:

$$\boldsymbol{\omega}_{\mathrm{A}\,i,j} = \begin{cases} \boldsymbol{\omega}_{i,j} & (i,j) \in \mathrm{A} \\ 0 & otherwise \end{cases}.$$

- The strong signals play the role of user selectors, which could spot the users holding strong deviation against the popular opinions.
- The weak signals seek the complement information from the residual $\boldsymbol{\gamma} - \boldsymbol{\omega}_{\mathrm{supp}(\gamma)}$, which is necessary for reducing the prediction error.

Experiments

Datasets

- **Movie-Lens** dataset is comprised of 3952 movies rated by 6040 users. Each movie is rated on a scale from 1 to 5, with 5 indicating the best movie and 1 indicating the worst movie.
- **Dining** dataset is adopted in this experiment to predict personalized preference for restaurants, comprised of 130 Mexican restaurants rated by 138 users. Each restaurant is rated on a scale from 0 to 2, with 2 indicating the best restaurant and 0 indicating the worst one.

Performance Comparisons for MoviewLens

Table 1: Coarse-grained vs. fine-grained (i.e., Ours) model in movie dataset.

	min	mean	max	std
RankSVM	0.4039	0.4304	0.4532	0.0131
RankBoost	0.4318	0.4554	0.4776	0.0131
RankNet	0.4056	0.4403	0.4625	0.0157
gbdt	0.3653	0.3850	0.4060	0.0115
dart	0.3704	0.3856	0.4023	0.0102
HodgeRank	0.4065	0.4303	0.4590	0.0126
URLR	0.4064	0.4300	0.4553	0.0124
Lasso	0.4089	0.4301	0.4557	0.0118
Ours	0.1204	**0.1473**	0.1814	0.0163

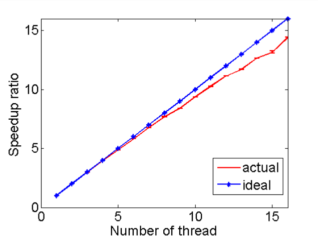
Performance Comparisons for Dinning

Table 2: Coarse-grained vs. Fine-grained (i.e., Ours) model on dining dataset.

	min	mean	max	std
RankSVM	0.4345	0.4734	0.4963	0.0142
RankBoost	0.4493	0.4734	0.5011	0.0148
RankNet	0.4529	0.4791	0.5127	0.0154
gdbt	0.4256	0.4587	0.4807	0.0141
dart	0.4335	0.4577	0.4827	0.0140
HodgeRank	0.4547	0.4769	0.4951	0.0113
URLR	0.4403	0.4684	0.4839	0.0117
Lasso	0.4374	0.4684	0.4895	0.0130
Ours	0.3450	**0.3708**	0.4069	0.0149

Speed-up for MovieLens

M	T(M)(s)	M	T(M)(s)
1	3155.63	9	380.09
2	1575.51	10	337.28
3	1054.71	11	308.54
4	794.88	12	282.12
5	648.38	13	268.97
6	543.54	14	249.66
7	466.64	15	241.45
8	410.50	16	218.84



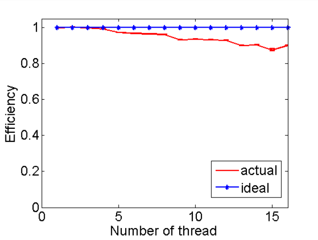
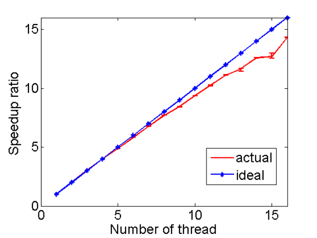


Figure 1: Experimental results of SynPar-SplitLBI with thread number changing from 1 to 16 in movie dataset.

Speed-up for Dining

M	T(M)(s)	M	T(M)(s)
1	759.40	9	90.28
2	377.85	10	81.20
3	252.26	11	74.50
4	190.16	12	68.32
5	156.39	13	65.67
6	130.52	14	60.32
7	112.57	15	59.66
8	98.67	16	53.11



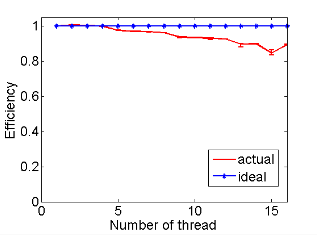


Figure 2: Experimental results of SynPar-SplitLBI with thread number changing from 1 to 16 in dining dataset.

Conclusion

Conclusions

- A preference learning model that takes into account of both the common consensus preference and users' preferential diversity.
- A dynamic path from the common preference to personalized diversity, with different levels of sparsity on personalization.
- A synchronized parallel version of our method is proposed to meet the needs of large-scale data analysis.
- The effectiveness of our model has been validated on the movie preference prediction and dining restaurant preference datasets.

Questions?