Two-level monotonic multistage recommender systems

(Joint work with Xiaotong Shen and Wei Pan)

by Ben Dai (The Chinese University of Hong Kong)
on EcoSta2022

» Single-stage (binary) Recommender Systems

	Item 1	Item 2	⋯	Item m
User 1	✓	?	⋯	✗
User 2	?	✓	⋯	?
⋯				
User n	?	✗	⋯	✓

A single-stage (binary) Recommender System.

Input *a* user-item pair (i,j)

Goal predict its corresponding response $Y_{ij} = \pm 1$

Example predict if *a* user will **book** *a* item

» Multi-stage Recommender Systems

Deskdrop dataset

* **Deskdrop** is an internal communications platform, which allows companies employees to share relevant articles with their peers, and collaborate around them

Size It contains about **73k logged users interactions** on more than **3k public articles** shared in the platform

User info [Id, agent, country]

Article info [Id, country, type, url]

Feedback follow > like > view

» Multi-stage Recommender Systems

Key characteristics to **Deskdrop** dataset

RS A typical RS dataset

Side info more features (continuous or categorical) for users/items

Feedback Monotonic property: follow > like > view

$$Y^t = -1 \text{ if } Y^{t-1} = -1, \text{ for } t = 1, \cdots, T$$

* The feedback at stage-t' may already be observed.

Goal predict its responses Y_{ij}^t based on $Y_{ij}^{t'}$ for $1 \leq t' < t \leq T$.

* obs a user **viewed** the item $\rightarrow$ if **like**?
* obs a user **viewed** the item $\rightarrow$ if **follow**?
* obs a user **liked** the item $\rightarrow$ if **follow**?

» Multi-stage Recommender Systems

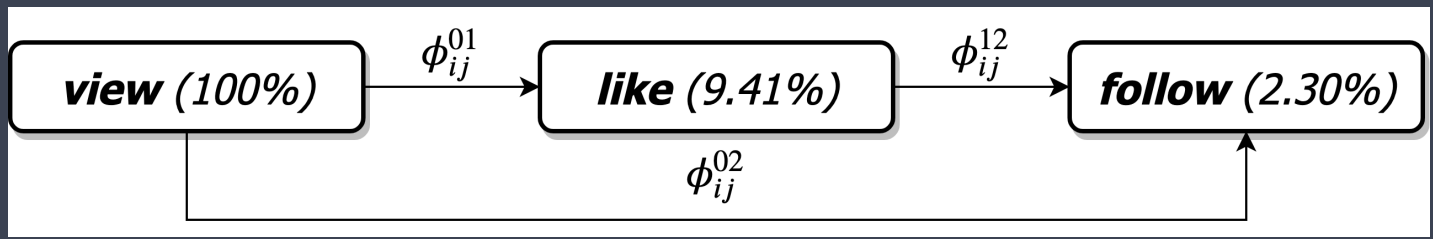
Input side info for a pair (i,j) + obs feedback $Y_{ij}^{t'}$ at stage-t'

Goal predict its response Y_{ij}^{t} at stage-t with $t > t'$

* A pairwise decision function based on the (t', t)-pair

$$\phi = (\phi^{t't}(\boldsymbol{x}, y^{t'}))_{0 \le t' < t \le T}$$

Example In **Deskdrop** dataset



» Multi-stage RS: Evaluation

* Pairwise mis-classification error:

$$e(\phi) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{t'<t} w_{t't} \boldsymbol{E}\left(I\left(Y_{ij}^{t} \phi^{t't}(\boldsymbol{X}_{ij}, Y_{ij}^{t'}) \leq 0\right)\right) \qquad (1)$$

* $w_{t't}$ is a stagewise weight to quantify the **importance** of prediction for the t-th stage based on the t'-th stage
  * **Next-stage-prediction**: $w_{t't} = I(t - t' = 1)$
  * **Last-stage-prediction**: $w_{t't} = I(t = T)$
* In general, $w_{t't} \geq 0$ can be a custom weight to indicate the relative importance among different stage-pairs

» Multi-stage RS: Bayes decision rule

Lemma 1 indicates two facts for the **optimal stagewise classifier** $\bar{\phi}$ with respect to the loss (1).

A. If $y_{ij}^{t'} = -1$ then the best prediction is -1.

$$\bar{\phi}^{t't}(\boldsymbol{x}_{ij}, y_{ij}^{t'}) = \begin{cases} -1, & \text{if } y_{ij}^{t'} = -1, \\ \operatorname{sgn}\left(\bar{f}^{t't}(\boldsymbol{x}_{ij})\right), & \text{if } y_{ij}^{t'} = 1, \end{cases}$$

» Multi-stage RS: Bayes decision rule

Lemma 1 indicates two facts for the **optimal stagewise classifier** $\bar{\phi}$ with respect to the loss (1).

A. If $y_{ij}^{t'} = -1$ then the best prediction is -1.

$$\bar{\phi}^{t't}(\boldsymbol{x}_{ij}, y_{ij}^{t'}) = \begin{cases} -1, & \text{if } y_{ij}^{t'} = -1, \\ \operatorname{sgn}\left(\bar{f}^{t't}(\boldsymbol{x}_{ij})\right), & \text{if } y_{ij}^{t'} = 1, \end{cases}$$

B. The **optimal conditional classifier** $\bar{f}^{t't}(\boldsymbol{x}_{ij})$ satisfies

$$\operatorname{sgn}\left(\bar{f}_{ij}^{t't}(\boldsymbol{x}_{ij})\right) = \operatorname{sgn}\left(\boldsymbol{P}_{ij}\left(Y_{ij}^{t} = 1 | Y_{ij}^{t'} = 1, \boldsymbol{X} = \boldsymbol{x}_{ij}\right) - 1/2\right),$$

iff. **two-level monotonic property**

$$\text{Forward: } \operatorname{sgn}(\bar{f}_{ij}^{t'(t+1)}(\boldsymbol{x}_{ij})) = -1, \quad \text{if } \operatorname{sgn}(\bar{f}_{ij}^{t't}(\boldsymbol{x}_{ij})) = -1,$$

$$\text{Backward: } \operatorname{sgn}(\bar{f}_{ij}^{t't}(\boldsymbol{x}_{ij})) = -1, \quad \text{if } \operatorname{sgn}(\bar{f}_{ij}^{(t'+1)t}(\boldsymbol{x}_{ij})) = -1. \tag{2}$$

» Multi-stage RS: Bayes decision rule

C. **Additive** form. There exists $\bar{h}_{ij}^{r}(\boldsymbol{x}_{ij}) \geq 0$, (it suffices to model h_{ij}^{r})

$$\bar{f}_{ij}^{t't}(\boldsymbol{x}_{ij}) = \bar{h}_{ij}^{0}(\boldsymbol{x}_{ij}) - \sum_{r=t'+1}^{t} \bar{h}_{ij}^{r}(\boldsymbol{x}_{ij}), \text{ with } \bar{h}_{ij}^{r} \geq 0;\ 0 \leq r \leq T. \tag{3}$$

This yields when $t' \to t$, $\bar{f}_{ij}^{t't}(\boldsymbol{x}_{ij})$ increases.

» Classifier: formulation

* FunkSVD got **third place** in Netflix's competition (first and second places are resemble models)
* Formulate the **interaction** btw the i-th user and j-th item.

$$h_{ij}^{r} = \boldsymbol{a}_i^{\mathsf{T}} \boldsymbol{b}_j$$

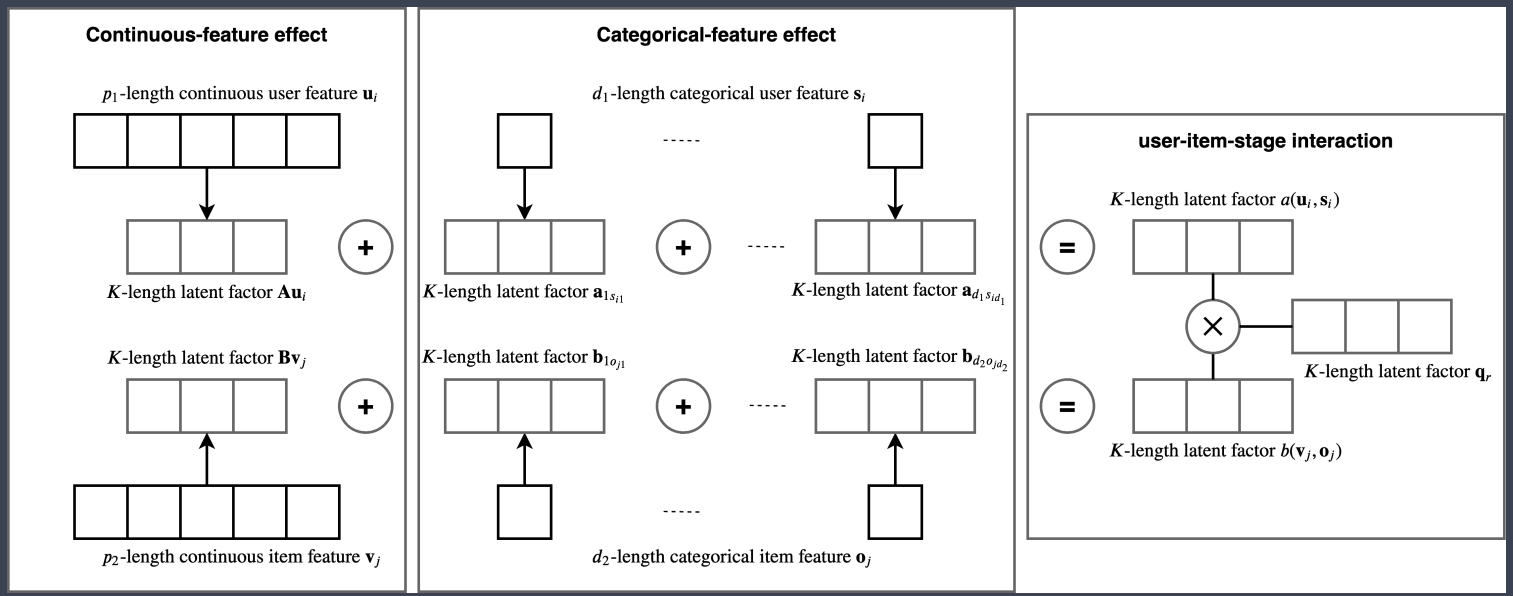
* How to incorporate the **side information**? additional continuous and discrete features for users and items
* $\underbrace{sideInfo}_{\boldsymbol{x}_{ij}}$: $(\underbrace{user\_cont}_{\boldsymbol{u}_i \in [0,1]^{p_1}} + \underbrace{user\_cate}_{\boldsymbol{v}_j \in [n_1] \times \cdots [n_{d_1}]})$ $+ (\underbrace{item\_cont}_{\boldsymbol{s}_i \in [0,1]^{p_2}} + \underbrace{item\_cate}_{\boldsymbol{o}_j \in [m_1] \times \cdots [m_{d_2}]})$
  * **User info**: [Id, agent, country]
  * **Article info**: [Id, country, type, url]

» Classifier: formulation

* Our formulation:

$$h_{ij}^{r}(\boldsymbol{x}_{ij}) = \left(a(\boldsymbol{u}_i, \boldsymbol{s}_i) \circ b(\boldsymbol{v}_j, \boldsymbol{o}_j)\right)^{\top} \boldsymbol{q}_r,$$

$$a(\boldsymbol{u}_i, \boldsymbol{s}_i) = \boldsymbol{A}\boldsymbol{u}_i + \sum_{l=1}^{d_1} \boldsymbol{a}_{ls_{il}}, \; b(\boldsymbol{v}_j, \boldsymbol{o}_j) = \boldsymbol{B}\boldsymbol{v}_j + \sum_{l=1}^{d_2} \boldsymbol{b}_{lo_{jl}}, \qquad (4)$$



* Additional constraints: $h_{ij}^{r}(\boldsymbol{x}_{ij}) \geq 0$

» Optimization: emprical loss

* Given observations $(\boldsymbol{x}_{ij}, y_{ij}^t)_{(i,j)\in\Omega_0, 1\le t\le T}$ and positive index set $\Omega_{t'} = \{(i,j) : y_{ij}^{t'} = 1\}$ at stage t', the empirical loss is given as

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{a},\boldsymbol{b},\boldsymbol{q}} \sum_{t'<t} \sum_{(i,j)\in\Omega_t} w_{t't} V\Big(y_{ij}^t \big(h_{ij}^0(\boldsymbol{x}_{ij}) - \sum_{r=t'+1}^{t} h_{ij}^r(\boldsymbol{x}_{ij})\big)\Big) + \lambda \mathrm{Reg},$$
$$\boldsymbol{A} \ge \boldsymbol{0},\ \boldsymbol{B} \ge \boldsymbol{0},$$
$$\boldsymbol{a}_{lh} \ge \boldsymbol{0}_K, h = 1, \cdots, n_l; l = 1, \cdots, d_1,$$
$$\boldsymbol{b}_{lh} \ge \boldsymbol{0}_K, h = 1, \cdots, m_l; l = 1, \cdots, d_2, \qquad (5)$$

» Optimization: BCD

- Jointly non-convex problem
- Blockwise convex with respect to $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{a}_l$, $\boldsymbol{b}_l$, $\boldsymbol{q}_r$
- Motivate to **blockwise coordinate descent** (BCD)
  - Fixing the others, update one parameter
  - Each subproblem is a non-negative + drifted + weighted SVM.
  - sovled by our Python library `varsvm`

» Optimization: BCD

User-effect block A. This convex optimization solves for A by fixing (B, a, b, q):

$$\min_{\mathrm{vec}(A)\geq 0} \sum_{t'<t} \sum_{(i,j)\in\Omega_{t'}} \underbrace{w_{t't}}_{\text{weights}} V\Big(\underbrace{y_{ij}^{t}}_{\text{label}} \Big(\underbrace{\mathrm{vec}(A)^{\top}}_{\text{coeff.}} \underbrace{\big(u_i \otimes \big(b(v_j, o_j) \circ q_{t't}\big)\big)}_{\text{input feats.}} + \underbrace{\Big(\sum_{l=1}^{d_1} a_{ls_{il}}\Big)^{\top} \big(b(v_j, o_j) \circ q_{t't}\big)}_{\text{fixed intercept (drift)}} \Big)\Big) + \lambda_1 \sum_{h=1}^{n_l} \|\mathrm{vec}(A)\|_2^2,$$

where $q_{t't} = 1_K - \sum_{r=t'+1}^{t} q_r$, $\otimes$ is the Kronecker product and $\mathrm{vec}(A)$ is the column vectorization of matrix A.

» Python library varsvm

- Python *scikit-learn* estimators module
- weighted SVMs, drifted SVMs, non-negative SVMs
- Much faster than `sklearn.svm.SVC` in Python for weighted SVMs

Doc `https://variant-svm.readthedocs.io`

» Experiments: real application

- **Deskdrop** article sharing
- Features:
  - User **cate feats**: [**user_Id**, **user_agent**, **user_region**]
  - Item **cont feats**: Doc2Vec embedding of "plain text" of an item
  - Item **cate feats**: [**article_Id**, **author_Id**, **language**]
- Evaluation:
  - Pairwise Mis-classification Error
  - Equal stage weight: $w_{t't} = 1$
- Comeptitors: DeepNN, GradBoost, SVM, OSVM, SVD++

» Experiments: real application

(observe t', predict t)	Proposed	SVD^{++}	GradBoost	SVM	DeepNN	OSVM
(Stage 0, Stage 1)	**0.136(.001)**	0.499(.003)	0.166(.000)	0.139(.000)	0.154(.001)	0.140(.000)
(Stage 0, Stage 2)	**0.191(.001)**	0.499(.007)	0.218(.001)	0.218(.001)	0.223(.006)	0.216(.001)
(Stage 1, Stage 2)	**0.043(.000)**	0.497(.007)	0.049(.000)	0.045(.000)	0.139(.011)	0.045(.000)
Overall	**0.123(.001)**	0.498(.006)	0.144(.000)	0.134(.000)	0.172(.005)	0.134(.000)
%Inconsist	0.00%(.000)	7.79%(.002)	1.40%(.000)	1.90%(.000)	13.7%(.005)	0.98%(.002)

Class-balanced zero-one losses of six competitors on the *Deskdrop* dataset, in addition to %Inconsist denoting a proportion of inconsistent instances violating the monotonicity (2).

- Overall performance with the amount of improvement over 8.20%
- With respect to inconsistency prediction, all other methods have inconsistent cases ranging from 0.98% to 13.7%.

» Summary

- The proposed method can produce a recommendation for any subsequent stage given observations at any present stage, which is highly demanded in real applications. Yet, most recommender systems focus on a fixed present stage.
- The two-level monotonic property is fully accounted for by our nonnegative additive latent factor model. As a result, it ensures prediction consistency across different stages.
- Asympototics and more numerical examples are included in our paper published in EJS.

Thank you!