

# Two-level monotonic multistage recommender systems\*

Ben Dai

*School of Statistics, University of Minnesota,  
Minneapolis, MN 55455,  
Department of Statistics, The Chinese University of Hong Kong,  
Shatin, Hong Kong SAR  
e-mail: [bendai@cuhk.edu.hk](mailto:bendai@cuhk.edu.hk)*

Xiaotong Shen

*School of Statistics, University of Minnesota,  
Minneapolis, MN 55455  
e-mail: [xshen@umn.edu](mailto:xshen@umn.edu)*

Wei Pan

*Division of Biostatistics, University of Minnesota,  
Minneapolis, MN 55455  
e-mail: [panxx014@umn.edu](mailto:panxx014@umn.edu)*

**Abstract:** A recommender system learns to predict the user-specific preference or intention over many items simultaneously for all users, making personalized recommendations based on a relatively small number of observations. One central issue is how to leverage three-way interactions, referred to as user-item-stage dependencies on a monotonic chain of events, to enhance the prediction accuracy. A monotonic chain of events occurs, for instance, in an article sharing dataset, where a “follow” action implies a “like” action, which in turn implies a “view” action. In this article, we develop a multistage recommender system utilizing a two-level monotonic property characterizing a monotonic chain of events for personalized prediction. Particularly, we derive a large-margin classifier based on a nonnegative additive latent factor model in the presence of a high percentage of missing observations, particularly between stages, reducing the number of model parameters for personalized prediction while guaranteeing prediction consistency. On this ground, we derive a regularized cost function to learn user-specific behaviors at different stages, linking decision functions to numerical and categorical covariates to model user-item-stage interactions. Computationally, we derive an algorithm based on blockwise coordinate descent. Theoretically, we show that the two-level monotonic property enhances the accuracy of learning as compared to a standard method treating each stage individually and an ordinal method utilizing only one-level monotonicity. Finally, the proposed method compares favorably with existing methods in simulations and an article sharing dataset.

**MSC2020 subject classifications:** Primary 62H30; secondary 68T05.

**Keywords and phrases:** Monotonicity, nonconvex minimization, implicit feedback, nonnegative latent factor model.

Received May 2021.

---

\*Research supported in part by NSF grants DMS-1712564, DMS-1721216, DMS-1952539, and NIH grants 1R01GM126002, R01HL105397, R01AG069895.

## Contents

1	Introduction . . . . .	5546
1.1	Literature review and our contributions . . . . .	5547
2	Multistage classification on a monotonic chain . . . . .	5548
2.1	Two-level monotonicity . . . . .	5548
2.2	Two-level monotonic multistage classification . . . . .	5550
2.3	Connection with existing frameworks . . . . .	5552
3	Large-scale computation . . . . .	5553
4	Theory . . . . .	5556
5	Numerical examples . . . . .	5557
5.1	Simulations . . . . .	5558
5.2	Real application: <i>Deskdrop</i> article sharing . . . . .	5559
6	Discussion . . . . .	5560
A	Technical proofs . . . . .	5561
	Supplementary Material . . . . .	5567
	References . . . . .	5567

## 1. Introduction

A multistage recommender system on a monotonic chain of events predicts a user’s preference of a large collection of items based on only a few user-item feedback at multiple stages, where a user’s positive feedback of subsequent stages can only be observed given his/her positive feedback at present stages. As a result, a user may exhibit an increasing level of intention and preference given the feedback, ranging from the most implicit to the most explicit. It has been widely used for personalized prediction in e-commerce and social networks as well as individual drug responses over multiple phases in personalized medicine [23]. In such a situation, the objective of a multistage recommender system is to predict a user’s subsequent behavior from his/her previous feedback with a high percentage of missing observations. For instance, as displayed in Figure 1, the *Deskdrop* article sharing dataset<sup>1</sup> consists of a sequence of actions ranging from *view*, *like*, to *follow*, where only less than 0.2% of values are observations with a total of 73,000 users and more than 3,000 articles. A user *likes* an article only if this user has *viewed* it, and the user may *follow* an article only when he/she has *liked* it. Thus, three pairwise predictions are made based on three pairs of present and subsequent stages are performed, namely, the prediction of if the user will like an article given that he has *viewed*, that if he/she will *follow* given that he has *viewed*, and that if he/she will *follow* given that he has *liked*.

One key characteristic of multistage prediction on a monotonic chain of events is a two-level monotonic property, as precisely defined in (2.3). On the one hand, monotonicity occurs over subsequent stages given a present stage. For instance, based on the fact that a user has *viewed* the article, this user won’t *follow* it if he/she doesn’t *like* it. On the other hand, monotonicity exhibits over present

<sup>1</sup><https://www.kaggle.com/gspmoreira/articles-sharing-reading-from-cit-deskdrop>

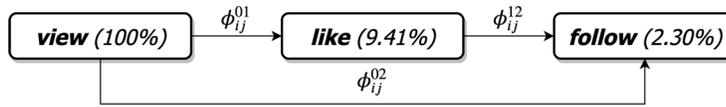


FIG 1. Demonstration of multistage prediction for Deskdrop dataset, a user may like an article only if this user has viewed it, and the user may follow an article only if this user has liked it. Three stagewise predictions based on different pairs of a given stage and a subsequent stage are considered: given view to predict like, given view to predict follow, and given like to predict follow.

stages given a subsequent stage in a reversed order. That said, if the user has decided to *follow* only he/she *viewed* it, then the user will *follow* an article when he/she has *liked*.

### 1.1. Literature review and our contributions

Despite recent progress in recommender systems, multistage personalized prediction on a monotonic chain of events remains largely unexplored. A standard method is to predict a user's behaviors at a subsequent stage given a present stage individually, ignoring the monotonic property, as described in (2.11). In this sense, essentially all existing methods of recommender systems for a single-stage are applicable, including latent factor models [16, 9], matrix factorization [18, 29], tensor factorization [2, 24, 22], and deep neural networks [14]. The worth of note is that ordinal regression/classification may be adopted by treating subsequent stages as an ordered class given a present stage, as in (2.12). For example, in [28, 25], latent factor models are developed based on an ordered logit model. Yet, a two-level monotonic property introduced by a monotonic chain of events is not met, which considers only forward monotonicity of subsequent stages given the present one but not the backward monotonicity over present stages given the subsequent one. As to be seen, the two-level monotonicity yields not only a high accuracy of prediction but also prediction consistency (2.3) among different stage pairs. In practice, prediction consistency is indeed desirable for decision-making.

This article develops a multistage recommender system that models user-item-stage interactions and integrates the two-level monotonic property into personalized prediction on a monotonic chain of events. Three key contributions in this article can be summarized:

- The proposed method can produce a recommendation for any subsequent stage given observations at any present stage, which is highly demanded in real applications; see Figure 1. Yet, most conventional recommender systems focus on a fixed present stage.
- A novel multistage loss function is proposed to treat multistage prediction for any pair of present and subsequent stages. This loss function admits user-item interactions observed at different stages and evaluates the prediction accuracy on all subsequent stages.

- The two-level monotonic property is fully accounted for by our nonnegative additive latent factor model based on the Bayes rule in Lemma 2.1. As a result, it substantially reduces model parameters and most importantly ensures prediction consistency across different stages. By comparison, none of aforementioned methods can guarantee prediction consistency, c.f., Tables 2 and 3.
- An algorithm is developed to implement the proposed method based on blockwise coordinate descent. Moreover, a learning theory is established to quantify the generalization error to demonstrate the benefits of modeling user-item-stage interactions based on the two-level monotonic property.

## 2. Multistage classification on a monotonic chain

Consider a recommender system in which triplets  $(\Delta_{ij}, \mathbf{X}_{ij}, \mathbf{Y}_{ij})$ , with  $\mathbf{Y}_{ij} = (Y_{ij}^1, \dots, Y_{ij}^T)$  are observed for user  $i$  on item  $j$  at stage  $t$ ;  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ,  $1 \leq t \leq T$ . Here  $n$  and  $m$  are the number of users and items, respectively.  $\Delta_{ij} = 1/0$  indicates if  $(\mathbf{X}_{ij}, \mathbf{Y}_{ij})$  is observed or missing and  $\Omega_t = \{(i, j) : \Delta_{ij} = 1, y_{ij}^t = 1\}$  is an index set of observed positive feedback at stage  $t$ , where  $Y_{ij}^t = \pm 1$  with  $1/-1$  indicating positive/negative feedback at stage  $t$ , and  $^\top$  denotes transpose. Moreover,  $\mathbf{x}_{ij} = (\mathbf{u}_i^\top, \mathbf{v}_j^\top, \mathbf{s}_i^\top, \mathbf{o}_j^\top)^\top$  consists of both numerical and categorical features, where  $\mathbf{u}_i \in [0, 1]^{p_1}$  and  $\mathbf{v}_j \in [0, 1]^{p_2}$  are user-specific and item-specific numerical predictors, respectively, and normalization is performed for each feature to scale to  $[0, 1]$  otherwise, and moreover  $\mathbf{s}_i = (s_{i1}, \dots, s_{id_1})^\top$  with  $s_{il} \in \{1, \dots, n_l\}$  and  $\mathbf{o}_j = (o_{j1}, \dots, o_{jd_2})$  with  $o_{jl} \in \{1, \dots, m_l\}$  are  $d_1$ -dimensional and  $d_2$ -dimensional user-specific and item-specific categorical predictor vectors. For instance, in the *Deskdrop* dataset<sup>2</sup>,  $\mathbf{v}_j$  is a numerical embedding for the content,  $\mathbf{s}_i$  consists of “person Id”, “user agent”, “user region”, “user country”,  $\mathbf{o}_j$  consists of “content Id”, “author Id”, “language”, and  $Y_{ij}^1$  and  $Y_{ij}^2$  indicate if article  $j$  is *liked* and *followed* by user  $i$ , respectively. One important characteristic of this kind of data is that subsequent events will not occur if one present event does not occur. For instance, if a user does not *view* an item, then subsequent events of *like* and *follow* will not occur.

On this ground, we define a monotonic behavior chain as follows:

$$Y_{ij}^t = -1 \text{ if } Y_{ij}^{t-1} = -1; \quad 1 \leq i \leq m, 1 \leq j \leq n, t = 1, \dots, T, \quad (2.1)$$

which says that event  $Y_{ij}^{t-1} = -1$  implies that any subsequent event  $Y_{ij}^t = -1$  must occur, which encodes a certain causal relation as defined by the local Markov property [10] in a directed acyclic graphical model.

### 2.1. Two-level monotonicity

Multistage classification learns to predict the outcome of  $Y_{ij}^t$  at a subsequent stage  $t$  given observations at a present stage  $Y_{ij}^{t'}$  and  $\mathbf{x}_{ij}$ ;  $0 \leq t' < t \leq T$ . With

<sup>2</sup><https://www.kaggle.com/gspmoreira/articles-sharing-reading-from-cit-deskdrop>

the convention, we set  $Y_{ij}^0 = 1$ . To predict, we introduce a decision function  $\phi_{ij}^{t't} = \phi_{ij}^{t't}(\mathbf{x}_{ij}, y_{ij}^{t'})$  for classification, which depends on observations  $\mathbf{x}_{ij}$  and  $y_{ij}^{t'}$ . For user  $i$  on item  $j$  across  $T$  stages, decision functions can be expressed as  $\phi_{ij} = (\phi_{ij}^{t't})_{0 \leq t' < t \leq T}$ . For all users over items across  $T$  stages, decision functions are  $\phi = (\phi_{ij})_{1 \leq i \leq n; 1 \leq j \leq m}$ .

To evaluate the overall performance of  $\phi$ , we define the multistage misclassification error

$$e(\phi) = (nm)^{-1} \sum_{i=1}^n \sum_{j=1}^m e_{ij}(\phi_{ij}),$$

$$e_{ij}(\phi_{ij}) = \sum_{0 \leq t' < t \leq T} w_{t't} \mathbb{E}_{ij}(\Delta_{ij} \mathbb{I}(Y_{ij}^t \phi_{ij}^{t't}(\mathbf{X}_{ij}, Y_{ij}^{t'}) \leq 0)), \quad (2.2)$$

where  $e_{ij}(\phi_{ij})$  is the pairwise misclassification error for user  $i$  on item  $j$  across  $T$  stages, and  $\mathbb{P}_{ij}(\cdot) = \mathbb{P}_{ij}(\cdot | \mathbf{X}_{ij} = \mathbf{x}_{ij})$  and  $\mathbb{E}_{ij}(\cdot) = \mathbb{E}(\cdot | \mathbf{X}_{ij} = \mathbf{x}_{ij})$  denote the conditional probability and expectation given  $\mathbf{X}_{ij} = \mathbf{x}_{ij}$ . In (2.2),  $w_{t't} \geq 0$  is a pre-specified weight for predicting the outcome at subsequent stage  $t$  based on present stage  $t'$ ;  $0 \leq t' < t \leq T$ , reflecting the relative importance with respect to prediction at different stages in the overall evaluation. In particular, (2.2) reduces to the next-stage prediction and last-stage prediction when  $w_{t't} = I(t - t' = 1)$  and  $w_{t't} = I(t = T)$ , respectively. Moreover, if missing occurs completely at random, or missing pattern  $\Delta_{ij}$  is independent of  $\mathbf{Y}_{ij}$ , then (2.2) is proportional to the standard misclassification error.

Lemma 2.1 gives the multistage Bayes decision function  $\bar{\phi}_{ij} = \operatorname{argmin}_{\phi_{ij}} e_{ij}(\phi_{ij})$  for user  $i$  on item  $j$  in (2.2) subject to a constraint that predicted  $Y_{ij}^t$  by the Bayes rule satisfies the monotonic behavior chain property (2.1).

**Lemma 2.1** (Multistage Bayes-rule). *The optimal multistage pairwise decision function  $\bar{\phi}_{ij}$  minimizing (2.2) can be written as: for  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ,  $0 \leq t' < t \leq T$ ,*

$$\bar{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, y_{ij}^{t'}) = \begin{cases} -1, & \text{if } y_{ij}^{t'} = -1, \\ \operatorname{sgn}(\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})) = \operatorname{sgn}(\mathbb{P}_{ij}(Y_{ij}^t = 1 | Y_{ij}^{t'} = 1, \Delta_{ij} = 1) - 1/2), & \text{if } y_{ij}^{t'} = 1; \end{cases}$$

The multistage Bayes rule for predicting  $Y_{ij}^t$  by  $\mathbf{x}_{ij}$  and  $Y_{ij}^{t'}$  is  $\operatorname{sgn}(\bar{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, Y_{ij}^{t'}))$ . Here  $\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})$  satisfies the two-level monotonic property (2.3) iff  $Y_{ij}^t$  follows (2.1), that is, for  $0 \leq t' < t \leq T$ ,

$$\begin{aligned} \text{Forward:} & \quad \operatorname{sgn}(\bar{f}_{ij}^{t'(t+1)}(\mathbf{x}_{ij})) = -1, \text{ if } \operatorname{sgn}(\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})) = -1, \\ \text{Backward:} & \quad \operatorname{sgn}(\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})) = -1, \text{ if } \operatorname{sgn}(\bar{f}_{ij}^{(t'+1)t}(\mathbf{x}_{ij})) = -1. \end{aligned} \quad (2.3)$$

Moreover, there exists  $\bar{h}_{ij}^r(\mathbf{x}_{ij}) \geq 0$  such that  $\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})$  can be written in an additive form:

$$\bar{f}_{ij}^{t't}(\mathbf{x}_{ij}) = \bar{h}_{ij}^0(\mathbf{x}_{ij}) - \sum_{r=t'+1}^t \bar{h}_{ij}^r(\mathbf{x}_{ij}), \text{ with } \bar{h}_{ij}^r \geq 0; 0 \leq r \leq T. \quad (2.4)$$

The two-level monotonic property (2.3) says that  $\text{sgn}(\bar{f}_{ij}^{t't}(\mathbf{x}_{ij}))$  is decreasing in  $t$  for any fixed  $t'$  (forward monotonicity) whereas is increasing in  $t'$  for any fixed  $t$  (backward monotonicity). Note that (2.4) guarantees (2.3).

In view of Lemma 2.1, we introduce our multistage pairwise decision functions to mimic the additive multistage Bayes rule: for some  $h_{ij}^r(\mathbf{x}_{ij}) \geq 0; 0 \leq r \leq T$ .

$$\phi_{ij}^{t't}(\mathbf{x}_{ij}, y_{ij}^{t'}) = \begin{cases} -1, & \text{if } y_{ij}^{t'} = -1, \\ f_{ij}^{t't}(\mathbf{x}_{ij}) = h_{ij}^0(\mathbf{x}_{ij}) - \sum_{r=t'+1}^t h_{ij}^r(\mathbf{x}_{ij}), & \text{if } y_{ij}^{t'} = 1. \end{cases} \quad (2.5)$$

## 2.2. Two-level monotonic multistage classification

Based on the representation of decision function in (2.5), we rewrite  $e_{ij}(\phi_{ij})$  in (2.2) as follows:

$$e_{ij}(\mathbf{f}_{ij}) = \sum_{0 \leq t' < t \leq T} w_{t't} \mathbb{E}_{ij}(\Delta_{ij} \mathbb{I}(Y_{ij}^t f_{ij}^{t't}(\mathbf{X}_{ij}) \leq 0) \mathbb{I}(Y_{ij}^{t'} = 1)). \quad (2.6)$$

Note that the indicator function  $\mathbb{I}(\cdot)$  in (2.6) is difficult to treat in optimization. Therefore, we replace it by a surrogate loss  $V(u)$  for large-margin classification, in which  $V$  is a function of the corresponding functional margin  $Y_{ij}^t f_{ij}^{t't}(\mathbf{x}_{ij})$ . This includes, but is not limited to, the hinge loss  $V(u) = (1 - u)_+$  [6], the import vector machine  $V(z) = \log(\exp(-z)/(1 - \exp(-z)))$  [34], and the  $\psi$ -loss  $V(u) = \min(1, (1 - u)_+)$  [21]. On this ground, we propose a multistage large-margin loss function:

$$L(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) = \sum_{0 \leq t' < t \leq T} w_{t't} \Delta_{ij} V(Y_{ij}^t f_{ij}^{t't}(\mathbf{x}_{ij})) \mathbb{I}(Y_{ij}^{t'} = 1), \quad (2.7)$$

where  $\mathbf{Z}_{ij} = (\Delta_{ij}, \mathbf{Y}_{ij})$  and  $\mathbf{f}_{ij} = (f_{ij}^{t't})_{0 \leq t' < t \leq T}$ .

Lemma 2.2 says that a minimizer of the cost  $l(\mathbf{f}) = (nm)^{-1} \sum_{ij} \mathbb{E}_{ij} L(\mathbf{f}_{ij}, \mathbf{Z}_{ij})$  with respect to  $\mathbf{f}$  satisfies the Bayes rule in Lemma 2.1.

**Lemma 2.2** (Multistage Fisher consistency). *The minimizer of  $l(\mathbf{f})$  is Fisher-consistent in that it satisfies the Bayes rule in Lemma 1 if the surrogate loss  $V(\cdot)$  is Fisher consistency in binary classification.*

Next we parametrize our decision functions based on an additive latent factor model with to incorporate the collaborative information across users, items and stages. In particular, we define  $h_{ij}^r(\mathbf{x}_{ij}) = (a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j))^\top \mathbf{q}_r$  in (2.4), and set  $\mathbf{q}_0 = \mathbf{1}$  to avoid the over-parametrization. Moreover,  $a(\mathbf{u}_i, \mathbf{s}_i)$  and  $b(\mathbf{v}_j, \mathbf{o}_j)$

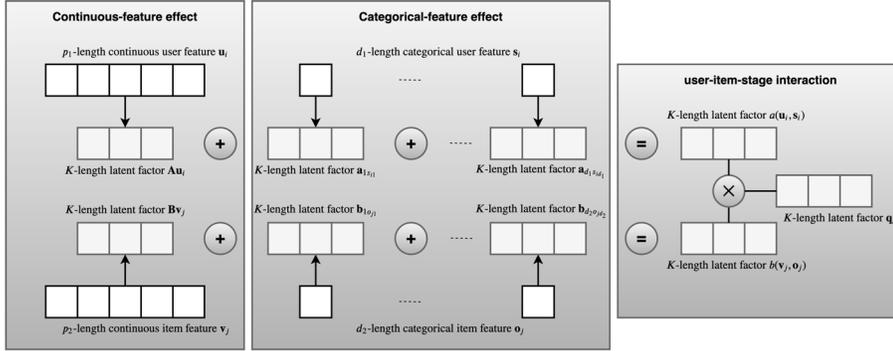


FIG 2. Architecture of proposed nonnegative additive latent factor models for continuous and categorical user-item specific features as well as the user-item-stage effect.

are proposed to link our decision function linearly to numerical predictors  $\mathbf{u}_i$  and  $\mathbf{v}_j$ , as well as additive latent factors structured by categorical predictors  $\mathbf{s}_i$  and  $\mathbf{o}_j$ . Then, the proposed prediction function can be written as

$$f_{ij}^{t't}(\mathbf{x}_{ij}) = \mathbf{a}^\top(\mathbf{u}_i, \mathbf{s}_i) \mathbf{b}(\mathbf{v}_j, \mathbf{o}_j) - \sum_{r=t'+1}^t (\mathbf{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \mathbf{b}(\mathbf{v}_j, \mathbf{o}_j))^\top \mathbf{q}_r, \quad (2.8)$$

$$\mathbf{a}(\mathbf{u}_i, \mathbf{s}_i) = \mathbf{A}\mathbf{u}_i + \sum_{l=1}^{d_1} \mathbf{a}_{l s_{il}}, \quad \mathbf{b}(\mathbf{v}_j, \mathbf{o}_j) = \mathbf{B}\mathbf{v}_j + \sum_{l=1}^{d_2} \mathbf{b}_{l o_{jl}},$$

subject to  $\mathbf{A} \geq \mathbf{0}$ ,  $\mathbf{B} \geq \mathbf{0}$ ,  $\mathbf{q}_r \geq \mathbf{0}$ ;  $r = 1, \dots, T$ ,  $\mathbf{a}_{lh} \geq \mathbf{0}$ ;  $l = 1, \dots, d_1$ ;  $h = 1, \dots, n_l$ ;  $\mathbf{b}_{lh} \geq \mathbf{0}$ ;  $l = 1, \dots, d_2$ ;  $h = 1, \dots, m_l$ , where  $\circ$  is the Hadamard product,  $\mathbf{A} \in \mathbb{R}^{K \times p_1}$  and  $\mathbf{B} \in \mathbb{R}^{K \times p_2}$  are two matrices, which transform the user-specific and item-specific numerical features to  $K$ -dimensional latent vectors, and  $\mathbf{a}_{l s_{il}}$  and  $\mathbf{b}_{l o_{jl}}$  are two  $K$ -dimensional latent factor for  $s_{il}$  and  $o_{jl}$ .

Representation (2.8) is highly interpretable, it leverages all feature-interaction between users and items based on an additive model. For example, in Deskdrop article sharing dataset, the interaction effect of “userId”-“articleId”, “userId”-“authorId”, “userAgent”-“articleId”, “userAgent”-“authorId”, are all captured in (2.8). Furthermore, nonnegative constraints for  $\mathbf{A} \geq \mathbf{0}$ ,  $\mathbf{B} \geq \mathbf{0}$ ,  $\mathbf{a}_{lh} \geq \mathbf{0}$  and  $\mathbf{b}_{lh} \geq \mathbf{0}$  are enforced to ensure the two-level monotonic property (2.3).

Given observations  $(\mathbf{x}_{ij}, y_{ij}^t)_{(i,j) \in \Omega_0, 1 \leq t \leq T}$  and positive index set  $\Omega_{t'} = \{(i, j) : \Delta_{ij} = 1, y_{ij}^{t'} = 1\}$  at stage  $t'$ , we propose a regularized multistage large-margin loss

$$\min_{\mathbf{f} \in \mathcal{F}} \tilde{l}_{nm}(\mathbf{f}); \quad \tilde{l}_{nm}(\mathbf{f}) = (nm)^{-1} \sum_{0 \leq t' < t \leq T} \sum_{(i,j) \in \Omega_{t'}} w_{t't} V(y_{ij}^t f_{ij}^{t't}(\mathbf{x}_{ij})) + J_\lambda(\mathbf{f}), \quad (2.9)$$

subject to  $\mathbf{A} \geq \mathbf{0}$ ,  $\mathbf{B} \geq \mathbf{0}$ ,  $\mathbf{a}_{lh} \geq \mathbf{0}$ ;  $1 \leq l \leq d_1$ ;  $1 \leq h \leq n_l$ ,  $\mathbf{b}_{lh} \geq \mathbf{0}$ ;  $1 \leq l \leq d_2$ ,  $1 \leq h \leq m_l$ , where  $J_\lambda(\mathbf{f}) = \lambda_1 (\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) + \lambda_2 (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2) + \lambda_3 \|\mathbf{q}\|_2^2$ ,  $\mathbf{a} =$

$(\mathbf{a}_1^\top, \dots, \mathbf{a}_{d_1}^\top)^\top$ ,  $\mathbf{b} = (\mathbf{b}_1^\top, \dots, \mathbf{b}_{d_2}^\top)^\top$ ,  $\mathbf{q} = (\mathbf{q}_1^\top, \dots, \mathbf{q}_T^\top)^\top$ ,  $\mathbf{a}_l = (\mathbf{a}_{l1}^\top, \dots, \mathbf{a}_{ln_l}^\top)^\top$ ,  $\mathbf{b}_l = (\mathbf{b}_{l1}^\top, \dots, \mathbf{b}_{lm_l}^\top)^\top$ ,  $\lambda_1, \lambda_2, \lambda_3 \geq 0$  are tuning parameters controlling the trade-off between learning and regularization, and  $\mathcal{F}$  is a space of candidate decision functions in (2.8), which can be written as

$$\mathcal{F} = \left\{ \mathbf{f} = (\mathbf{f}_{ij}^{t't}) : \mathbf{f}_{ij}^{t't}(\mathbf{x}_{ij}) = \mathbf{a}^\top(\mathbf{u}_i, \mathbf{s}_i) \mathbf{b}(\mathbf{v}_j, \mathbf{o}_j) - \sum_{r=t'+1}^t (a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j))^\top \mathbf{q}_r, \right. \\ \left. \begin{aligned} \mathbf{a}_{lh} &\in \mathbb{R}_+^K, \quad l = 1, \dots, d_1, h = 1, \dots, n_l; \\ \mathbf{b}_{lh} &\in \mathbb{R}_+^K, \quad l = 1, \dots, d_2, h = 1, \dots, m_l, \\ \mathbf{q}_t &\in \mathbb{R}_+^K, \quad t = 1, \dots, T; \quad \mathbf{A} \in \mathbb{R}_+^{K \times p_1}, \mathbf{B} \in \mathbb{R}_+^{K \times p_2} \end{aligned} \right\}, \quad (2.10)$$

where  $a(\cdot, \cdot)$  and  $b(\cdot, \cdot)$  are defined in (2.8).

Minimization (2.9) in  $(\mathbf{A}, \mathbf{B}, \mathbf{q}, \mathbf{b}, \mathbf{q})$  yields an estimated  $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{q}})$  thus  $\hat{\mathbf{f}}_{ij}^{t't}$  by (2.8) and  $\hat{\phi}_{ij}^{t't}$  by (2.5). Finally, the outcome of  $Y_{ij}^t$  is predicted by  $\text{sgn}(\hat{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, Y_{ij}^t))$ ;  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ,  $0 \leq t' < t \leq T$ .

### 2.3. Connection with existing frameworks

This section compares the proposed method with a standard method treating each stage individually and an ordinal method treating the stage as an ordinal class.

**Standard.** A standard method treats  $(t', t)$ -pairwise classification separately and then combine the prediction results for  $0 \leq t' < t \leq T$ . In fact, it estimates  $\mathbf{f}^{t't} = (f_{ij}^{t't})_{1 \leq i \leq n; 1 \leq j \leq m}$  by solving:

$$\min_{\mathbf{f}^{t't} \in \mathcal{F}^{t't}} (nm)^{-1} \sum_{(i,j) \in \Omega_{t'}} V(y_{ij}^t, f_{ij}^{t't}(\mathbf{x}_{ij})) + J_\lambda(\mathbf{f}^{t't}); \quad 0 \leq t' < t \leq T, \quad (2.11)$$

where  $\mathcal{F}^{t't}$  is a parameter space of candidate pairwise decision functions. A standard method includes latent factor models [16], gradient boosting [5], and deep neural networks [14], and  $J_\lambda(\mathbf{f}^{t't})$  regularizes  $\mathbf{f}^{t't}$  with a nonnegative tuning parameter  $\lambda \geq 0$ .

**Ordinal.** An ordinal method treats each stage  $t'$  separately, which estimates  $\mathbf{f}^{t'} = (\mathbf{f}^{t'})_{t' < t \leq T}$  by solving:

$$\min_{\mathbf{f}^{t'} \in \mathcal{F}^{t'}} (nm)^{-1} \sum_{(i,j) \in \Omega_{t'}} \sum_{t=t'+1}^T w_{t't} V(y_{ij}^t, f_{ij}^{t't}(\mathbf{x}_{ij})) + J_\lambda(\mathbf{f}^{t'}), \quad \text{for } 0 \leq t' < T, \quad (2.12)$$

subject to  $f_{ij}^{t'(t'+1)} \geq f_{ij}^{t'(t'+2)} \geq \dots \geq f_{ij}^{t'T}$ , where  $\mathcal{F}^{t'}$  is a parameter space of candidate stagewise decision functions and  $J_\lambda(\mathbf{f}^{t'})$  regularizes  $\mathbf{f}^{t'}$ . For example, [7] formulates a parallel decision function  $f_{ij}^{t't}(\mathbf{x}_{ij}) = \beta_{t'}^\top \mathbf{x}_{ij} + \beta_{t't}$  and imposes

the constraint  $\beta_{t'(t'+1)} \geq \beta_{t'(t'+2)} \cdots \geq \beta_{t'T}$ . Moreover, since  $\mathbf{x}_{ij}$  is pre-scaled in  $[0, 1]$ , we can set  $\mathbf{f}_{ij}^{t't} = (\beta_{t'0} - \sum_{t=t'+1}^T \beta_{t't})^\top \mathbf{x}_{ij}$ , where the forward monotonicity is ensured by positive constraints  $\beta_{t't} \geq \mathbf{0}$ ,  $t = t' + 1, \dots, T$ . Note that when  $V(y_{ij}^t, f_{ij}^{t't})$  is replaced as a negative log-likelihood function, then (2.12) is a formulation for ordinal regression [19, 1].

Ordinal classification incorporates the forward monotonicity but ignores backward monotonicity. Whereas such monotonicity helps to reduce the size of the parameter space, by comparison, the proposed method can further reduce parameters utilizing the backward monotonicity. In contrast, a standard method does not leverage any level of monotonicity. Most critically, both standard and ordinal methods fail to yield prediction consistency.

### 3. Large-scale computation

This section develops a computational scheme to solve nonconvex minimization (2.9). For illustration, consider the hinge loss  $V(u) = (1 - u)_+$ . The scheme minimizes a nonconvex cost function (2.9) by solving a sequence of relaxed convex subproblems via a block successive minimization (BSM). The scheme uses blockwise descent to alternate the following convex subproblems. For each subproblem, we decompose (2.9) into an equivalent form of many small optimizations for parallelization and for alleviating the memory requirement. Let  $\mathbf{q}^{t't} = \mathbf{1}_K - \sum_{r=t'+1}^t \mathbf{q}_r$ , we solve (2.9) as follows.

**User-effect block  $\mathbf{A}$ .** This convex optimization solves for  $\mathbf{A}$ :

$$\begin{aligned} \min_{\text{vec}(\mathbf{A}) \geq \mathbf{0}} (nm)^{-1} \sum_{0 \leq t' < t \leq T} \sum_{(i,j) \in \Omega_{t'}} w_{t't} \xi_{ij}^{t't} + \lambda_1 \sum_{h=1}^{n_l} \|\text{vec}(\mathbf{A})\|_2^2, \\ \xi_{ij}^{t't} \geq 1 - y_{ij}^t \left( \text{vec}(\mathbf{A})^\top (\mathbf{u}_i \otimes (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}^{t't})) + \left( \sum_{l=1}^{d_1} \mathbf{a}_{ls_{il}} \right)^\top (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}^{t't}) \right), \\ \xi_{ij}^{t't} \geq 0; (i, j) \in \Omega_{t'}, 0 \leq t' < t \leq T, \end{aligned} \quad (3.1)$$

where  $\otimes$  is the Kronecker product and  $\text{vec}(\mathbf{A})$  is the column vectorization of matrix  $\mathbf{A}$ .

**User-effect block  $\mathbf{a}$ .** This convex optimization solves for  $(\mathbf{a}_1, \dots, \mathbf{a}_{d_1})$ :

$$\begin{aligned} \min_{\mathbf{a}_{lh} \geq \mathbf{0}} (nm)^{-1} \sum_{h=1}^{n_l} \sum_{0 \leq t' < t \leq T} \sum_{\{i: s_{il}=h\}} \sum_{j \in \{(i,j) \in \Omega_{t'}\}} w_{t't} \xi_{ij}^{t't} + \lambda_2 \sum_{h=1}^{n_l} \|\mathbf{a}_{lh}\|_2^2, \\ \xi_{ij}^{t't} \geq 1 - y_{ij}^t \left( \mathbf{a}_{ls_{il}}^\top (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}^{t't}) + (\mathbf{A}\mathbf{u}_i + \sum_{l' \neq l} \mathbf{a}_{l's_{il'}})^\top (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}^{t't}) \right), \\ \xi_{ij}^{t't} \geq 0; (i, j) \in \Omega_{t'}; 0 \leq t' < t \leq T. \end{aligned} \quad (3.2)$$

Note that (3.2) can be separately solved for each  $\mathbf{a}_{lh}$  in a parallel fashion,

$$\min_{\mathbf{a}_{lh} \geq \mathbf{0}} (nm)^{-1} \sum_{0 \leq t' < t \leq T} \sum_{\{i: s_{il}=h\}} \sum_{j \in \{(i,j) \in \Omega_{t'}\}} w_{t't} \xi_{ij}^{t't} + \lambda_2 \|\mathbf{a}_{lh}\|_2^2,$$

$$\begin{aligned}\xi_{ij}^{t't} &\geq 1 - y_{ij}^t \left( \mathbf{a}_{lh}^\top (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}_{t't}) + (\mathbf{A}\mathbf{u}_i + \sum_{l' \neq l} \mathbf{a}_{l's_{il'}})^\top (b(\mathbf{v}_j, \mathbf{o}_j) \circ \mathbf{q}_{t't}) \right), \\ \xi_{ij}^{t't} &\geq 0; (i, j) \in \Omega_{t'}; 0 \leq t' < t \leq T.\end{aligned}\quad (3.3)$$

**Item-effect block  $\mathbf{B}$ .** This convex optimization solves for  $\mathbf{B}$ :

$$\begin{aligned}\min_{\text{vec}(\mathbf{B}) \geq 0} (nm)^{-1} &\sum_{0 \leq t' < t \leq T} \sum_{(i,j) \in \Omega_{t'}} w_{t't} \xi_{ij}^{t't} + \lambda_1 \sum_{h=1}^{n_l} \|\text{vec}(\mathbf{B})\|_2^2, \\ \xi_{ij}^{t't} &\geq 1 - y_{ij}^t \left( \text{vec}(\mathbf{B})^\top (\mathbf{v}_j \otimes (a(\mathbf{u}_i, \mathbf{s}_j) \circ \mathbf{q}_{t't})) + \left( \sum_{l=1}^{d_1} \mathbf{b}_{l's_{il}} \right)^\top (a(\mathbf{u}_i, \mathbf{s}_i) \circ \mathbf{q}_{t't}) \right), \\ \xi_{ij}^{t't} &\geq 0; (i, j) \in \Omega_{t'}, 0 \leq t' < t \leq T.\end{aligned}\quad (3.4)$$

**Item-effect block  $\mathbf{b}$ .** For  $l = 1, \dots, d_2$ ,  $(\mathbf{b}_{lh})_{h=1}^{m_l}$  is solved in a parallel fashion,

$$\begin{aligned}\min_{\mathbf{b}_{lh} \geq 0} (nm)^{-1} &\sum_{0 \leq t' < t \leq T} \sum_{\{j: \mathbf{o}_{jl}=h\}} \sum_{i \in \{i: (i,j) \in \Omega_{t'}\}} w_{t't} \xi_{ij}^{t't} + \lambda_2 \|\mathbf{b}_{lh}\|_2^2, \\ \xi_{ij}^{t't} &\geq 1 - y_{ij}^t \left( \mathbf{b}_{lh}^\top (a(\mathbf{u}_i, \mathbf{s}_i) \circ \mathbf{q}_{t't}) + (\mathbf{B}\mathbf{v}_j + \sum_{l' \neq l} \mathbf{b}_{l'o_{jl'}})^\top (a(\mathbf{u}_i, \mathbf{s}_i) \circ \mathbf{q}_{t't}) \right), \\ \xi_{ij}^{t't} &\geq 0; j \in \{j: \mathbf{o}_{jl}=h\}, (i, j) \in \Omega_{t'}; 0 \leq t' < t \leq T.\end{aligned}\quad (3.5)$$

**Stage-effect block  $\mathbf{q}$ .** This convex optimization solves for  $\mathbf{q}_r$ , for  $r = 1, \dots, T$ , given the present values for all other variables:

$$\begin{aligned}\min_{\mathbf{q}_r \geq 0} (nm)^{-1} &\sum_{0 \leq t' < r \leq t \leq T} \sum_{(i,j) \in \Omega_{t'}} w_{t't} \xi_{ij}^{t't} + \lambda_3 \|\mathbf{q}_r\|_2^2, \\ \xi_{ij}^{t't} &\geq 1 - y_{ij}^t \left( -\mathbf{q}_r^\top (a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j)) \right. \\ &\quad \left. + \left( \mathbf{1} - \sum_{r'=t'+1; r' \neq r}^t \mathbf{q}_{r'} \right)^\top (a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j)) \right), \\ \xi_{ij}^{t't} &\geq 0; (i, j) \in \Omega_{t'}; 0 \leq t' < t \leq T.\end{aligned}\quad (3.6)$$

As a technical note, (3.1)-(3.6) are standard SVMs with fixed intercept and positive constrains for model parameters, which can be solved via one efficient implementation in our python package `varsvm`<sup>3</sup> based on the coordinate decent algorithm [32].

The aforementioned scheme is summarized in Algorithm 1.

**Algorithm 1 (Parallelized version: hinge loss)**

**Step 1** (Initialization). Initialize the values of  $(\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}, \mathbf{q})$  and specify the tolerance error.

<sup>3</sup><https://pypi.org/project/varsvm/>

**Step 2** (Update  $\mathbf{A}$  and  $\mathbf{a}$ ). Update  $\mathbf{A}$  by solving (3.1) given present values of the other variables. For  $l = 1, \dots, d_1$ , update  $\mathbf{a}_{lh}$ ;  $h = 1, \dots, n_l$  in a parallel fashion by solving (3.3) given present values of the other variables.

**Step 3** (Update  $\mathbf{B}$  and  $\mathbf{b}$ ). Update  $\mathbf{B}$  by solving (3.4) given present values of the other variables. For  $l = 1, \dots, d_2$ , update  $\mathbf{b}_{lh}$ ;  $h = 1, \dots, m_l$  in a parallel fashion by solving (3.5) given present values of the other variables.

**Step 4** (Update  $\mathbf{q}$ ). For  $r = 1, \dots, T$ , update  $\mathbf{q}_r$  by solving (3.6) given present values of the other variables.

**Step 5.** Iterate **Steps 2-4** until the decrement of the cost function in (2.9) is less than the tolerance error.

Algorithm 1 returns an estimate  $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{q}})$  at termination, which in turn yields an estimated decision functions  $\hat{f}_{ij}^{t't}(\mathbf{x}_{ij}) = \hat{\mathbf{a}}^\top(\mathbf{u}_i, \mathbf{s}_i)\hat{\mathbf{b}}(\mathbf{v}_j, \mathbf{o}_j) - \sum_{r=t'+1}^t (\hat{\mathbf{a}}(\mathbf{u}_i, \mathbf{s}_i) \circ \hat{\mathbf{b}}(\mathbf{v}_j, \mathbf{o}_j))^\top \hat{\mathbf{q}}_r$ , and  $\hat{\phi}_{ij}^{t't} = -1$  if  $Y_{ij}^{t'} = -1$ ,  $\hat{f}_{ij}^{t't}$  otherwise. Finally, the prediction at the  $t$ -th stage given the  $t'$ -th stage is  $\text{sgn}(\hat{\phi}_{ij}^{t't}(y_{ij}^{t'}, \mathbf{x}_{ij}))$  for new user-item covariates  $\mathbf{x}_{ij}$ .

To implement Algorithm 1, we develop a Python library to solve an SVM-type of optimization, including weighted SVMs, drifted SVMs, and non-negative SVMs, based on coordinate descent of the dual problem [32]. This implementation has been available in `varsvm` library in Python on the GitHub repository `varsvm`<sup>4</sup>, and the source code is released with the MIT license using GitHub and the release is available via PyPI. The development undergoes an integration of Ubuntu Linux and Mac OS X, in Python 3.

**Lemma 3.1** (Convergence of Algorithm 1). *A solution  $(\hat{\mathbf{A}}, \hat{\mathbf{a}}, \hat{\mathbf{B}}, \hat{\mathbf{b}}, \hat{\mathbf{q}})$  of Algorithm 1 is a stationary point of  $l_{nm}$  in (2.9) in that*

$$\begin{aligned} \tilde{\mathbf{A}} &= \underset{\mathbf{A}}{\text{argmin}} \tilde{l}_{nm}(\mathbf{A}, \tilde{\mathbf{a}}, \tilde{\mathbf{B}}, \tilde{\mathbf{b}}, \tilde{\mathbf{q}}), & \tilde{\mathbf{B}} &= \underset{\mathbf{B}}{\text{argmin}} \tilde{l}_{nm}(\tilde{\mathbf{A}}, \tilde{\mathbf{a}}, \mathbf{B}, \tilde{\mathbf{b}}, \tilde{\mathbf{q}}), \\ \tilde{\mathbf{a}}_l &= \underset{\mathbf{a}_l}{\text{argmin}} \tilde{l}_{nm}(\tilde{\mathbf{A}}, \tilde{\mathbf{a}}_1, \dots, \mathbf{a}_l, \dots, \tilde{\mathbf{a}}_{d_1}, \tilde{\mathbf{B}}, \tilde{\mathbf{b}}, \tilde{\mathbf{q}}); & 1 \leq l \leq d_1; \\ \tilde{\mathbf{b}}_l &= \underset{\mathbf{b}_l}{\text{argmin}} \tilde{l}_{nm}(\tilde{\mathbf{A}}, \tilde{\mathbf{a}}, \tilde{\mathbf{B}}, \tilde{\mathbf{b}}_1, \dots, \mathbf{b}_l, \dots, \tilde{\mathbf{b}}_{d_2}, \tilde{\mathbf{q}}); & 1 \leq l \leq d_2; \\ \tilde{\mathbf{q}}_t &= \underset{\mathbf{q}_t}{\text{argmin}} \tilde{l}_{nm}(\tilde{\mathbf{A}}, \tilde{\mathbf{a}}, \tilde{\mathbf{B}}, \tilde{\mathbf{b}}, \tilde{\mathbf{q}}_1, \dots, \mathbf{q}_t, \dots, \tilde{\mathbf{q}}_T); & 1 \leq t \leq T. \end{aligned}$$

The successive update in Algorithm 1 suffices to ensure its convergence because the cost function in (2.9) is strictly blockwise convex, that is, (3.1), (3.3), (3.4), (3.5), and (3.6) are strictly convex in terms of its parameters within each block, yielding a unique minimizer at each step. This aspect differs from the maximum block improvement [4] that guarantees the convergence of blockwise coordinate descent for a general objective function. One benefit of successive updating is that it significantly reduces computational complexity. Note that the solution of Algorithm 1 can be a global minimizer when certain additional assumptions are made [13].

<sup>4</sup><https://github.com/statmlben/variant-svm>

### 4. Theory

This section investigates the generalization aspect of the proposed multistage recommender  $\hat{\mathbf{f}}$  in terms of the accuracy of classification, as measured by the classification regret, defined as  $e(\hat{\mathbf{f}}) - e(\mathbf{f})$ , where  $e(\cdot)$  is the generalization error defined in (2.2).

Assume that  $(\Delta_{ij}, Y_{ij})$  given  $\mathbf{X}_{ij}; 1 \leq i \leq n; 1 \leq j \leq m$ , are conditionally independent, although  $(\Delta_{ij}, \mathbf{X}_{ij}, Y_{ij})$ 's may not be independent as multiple items can be purchased for a same user or multiple users may purchase a same item. Let  $\bar{\mathbf{f}} \in \mathcal{F}$  be a Bayes decision function, with  $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{a}}, \bar{\mathbf{b}}, \bar{\mathbf{q}})$ , and the number of latent factors be  $\bar{K}$ . Let the truncated  $V$ -loss be  $V_B(u) = \min(V(u), B)$ , where  $B \geq V(Y_{ij}^t \bar{\mathbf{f}}^{t,t}(\mathbf{x}_{ij}))$  for any  $i = 1, \dots, n; j = 1, \dots, m; 0 \leq t' < t \leq T$ , and the corresponding loss defined by  $V_B(u)$  be

$$L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) = \sum_{0 \leq t' < t \leq T} w_{t't} \Delta_{ij} V_B(Y_{ij}^t J_{ij}^{t',t}(\mathbf{X}_{ij})) \mathbb{I}(Y_{ij}^{t'} = 1),$$

and the truncated cost function is  $l_B(\mathbf{f}) = (nm)^{-1} \sum_{i,j} \mathbb{E}_{ij} L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})$ . The following technical assumptions are assumed:

**Assumption A (Conversion).** There exist constants  $0 \leq \alpha \leq \infty$  and  $c_1 > 0$  such that for all  $0 < \epsilon \leq B$  and  $\mathbf{f} \in \mathcal{F}$ ,

$$\sup_{\{\mathbf{f} \in \mathcal{F}; l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq \epsilon\}} e(\mathbf{f}) - e(\bar{\mathbf{f}}) \leq c_1 \epsilon^\alpha.$$

**Assumption B (Variance property).** There exist constants  $0 \leq \mu \leq 2$  and  $c_2 > 0$  such that for any  $\epsilon \geq 0, n, m \geq 1$ , and  $\mathbf{f} \in \mathcal{F}$ ,

$$\sup_{\{\mathbf{f} \in \mathcal{F}; l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq \epsilon\}} (nm)^{-1} \sum_{i=1}^n \sum_{j=1}^m \mathbb{E}_{ij} \left( (L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) - L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}))^2 \right) \leq c_2 \epsilon^\mu.$$

Assumptions A and B are two local smooth conditions regarding a connection between two different metrics and the relation between the mean and variance of the regret function in a neighborhood of  $\bar{\mathbf{f}}$ . Similar assumptions have been used in binary classification can be founded in [33], for example,  $\alpha = \mu = 1$  when  $V(\cdot)$  is the  $\psi$ -loss [21] or the hinge loss under low-noise assumption.

Let  $K$  and  $\bar{K}$  be the numbers of latent factor of  $\hat{\mathbf{f}}$  and  $\bar{\mathbf{f}}$ , respectively.

**Theorem 4.1.** Let  $\hat{\mathbf{f}}$  be a global minimizer of (2.9) with  $K \geq \bar{K}$ . If Assumptions A and B hold, then there exist constants  $c_3 > 0$  and  $c_4 > 0$  such that

$$\mathbb{P}(e(\hat{\mathbf{f}}) - e(\bar{\mathbf{f}}) \geq \epsilon_{nm}^{2\alpha}) \leq c_3 \exp(-c_4 nm (\lambda \bar{J})^{2-\bar{\mu}}),$$

where  $\bar{J} = \max(J(\bar{\mathbf{f}}), 1)$ ,  $J_\lambda(\bar{\mathbf{f}}) \sim \epsilon_{nm}^2/2$ , and

$$\epsilon_{nm}^2 = O\left\{ \left( \frac{(\Lambda + T)K}{nm} \log \left( \frac{nm \bar{w} B \bar{J} (p_1 + p_2 + d_1 + d_2) (6T^3 + T^2 K)}{(\Lambda + T)K} \right) \right)^{\frac{1}{2-\bar{\mu}}} \right\}, \tag{4.1}$$

where  $\bar{\mu} = \min(1, \mu)$ ,  $\bar{w} = \sum_{0 \leq t' < t \leq T} w_{t't}^2$ ,  $(\Lambda + T)K$  is the total number of model parameters, and  $\Lambda = p_1 + p_2 + \sum_{l=1}^{d_1} n_l + \sum_{l=1}^{d_2} m_l$ .

When  $nm \geq (p_1 + p_2 + d_1 + d_2)(\Lambda + T)K(6T^3 + T^2K)$ , (4.1) suggests that the convergence rate for the proposed method is  $O\left(\left(\frac{(\Lambda+T)K}{nm} \log\left(\frac{nm}{(\Lambda+T)K}\right)\right)^{\frac{\alpha}{2-\mu}}\right)$ , which becomes  $O\left(\frac{(\Lambda+T)K}{nm} \log\left(\frac{nm}{(\Lambda+T)K}\right)\right)$  when  $\alpha = \mu = 1$ , and matches with the minimax rate of the standard classification in [27].

As indicated in (4.1), the convergence rate is improved by the two-level monotonic chain property in (2.3) and the additive property in (2.4). As illustrated in Table 1, based on the factorization model in (2.8) while ignoring the two-level monotonicity (2.3), a standard method and an ordinal method involve  $\Lambda KT(T+1)/2$  and  $\Lambda KT$  parameters, which are more than  $(\Lambda+T)K$  and significantly impedes the performance when  $T$  becomes large. Furthermore, due to the monotonic property in (2.1), the effective sample size for  $Y_{ij}^{t'} = 1$  decreases exponentially as  $t'$  increases. Then the standard and ordinal methods suffer from over-fitting at late stages with a small sample size. In contrast, the proposed method leverages the two-level monotonicity to reduce the dimension of the underlying problem.

TABLE 1

*Monotonicity and model parameters of the proposed, standard, and ordinal frameworks, denoting (2.9), (2.11) and (2.12). Here forward and backward monotonicity is defined in (2.3), consistency denotes if the methods can provide a consistent prediction result, and #Parameters denotes the number of model parameters based on (2.4).*

Method	Forward monotonicity	Backward monotonicity	Consistency	#Parameters
Proposed	✓	✓	✓	$(\Lambda + T)K$
Standard	✗	✗	✗	$\Lambda KT(T + 1)/2$
Ordinal	✓	✗	✗	$\Lambda KT$

## 5. Numerical examples

This section examines the proposed method in (2.9) with the hinge loss  $V(u) = (1-u)_+$  and compares it with standard methods, including latent factor models (SVD<sup>++</sup>; [16]), gradient boosting (GradBoost) [11], support vector machine (SVM) [6], deep neural network (DeepNN) [20], ordinal method (2.12), namely ordinal support vector machine (OSVM), in a simulation and an article sharing benchmark.

For a standard method, SVD<sup>++</sup>, GradBoost, SVM, and DeepNN treat each  $(t', t)$  pair separately for all possible pairs, where we use one-hot encoder [30] to convert a categorical covariate to 0-1 numerical predictors for training. Moreover, for SVD<sup>++</sup>, the number of latent factors is set as 20, and latent factors are estimated using the first columns of  $\mathbf{s}_i$  and  $\mathbf{o}_j$  with the rating  $(Y_{ij}^t + 1)/2 \in \{0, 1\}$  while the label is predicted as 1 if the predicted rating exceeds 0.5 and  $-1$  otherwise; for GradBoost, the minimum number of samples to split an internal node is set to be 2 and the minimum number of samples required to be at a leaf node is 1, and the number of boosting stages is set to be 10. For DeepNN, a ReLU network is used with the number of node in each layer being fixed at 32. For the ordinal method, we set the OSVM classifier in (2.12) as  $\mathbf{f}_{ij}^{t't} = (\boldsymbol{\beta}_{t'0} - \sum_{t_0=t'+1}^t \boldsymbol{\beta}_{t't_0})^\top \mathbf{x}_{ij}$

since  $\mathbf{x}_{ij}$  is pre-scaled in  $[0, 1]$ , where the forward monotonicity is ensured by positivity constraints  $\beta_{t't} \geq \mathbf{0}$ ,  $t = t' + 1, \dots, T$ .

For implementation, we use our Python library `varsvm` for the proposed method and OSVM, the Python library `sklearn`<sup>5</sup> for GradBoost, SVM, and DeepNN, and the Python library `Surprise`<sup>6</sup> for SVD<sup>++</sup>. For all methods, the prediction of  $Y_{ij}^t$  is automatically set as  $-1$  when  $Y_{ij}^{t'} = -1$ .

### 5.1. Simulations

In simulations, data  $\{\mathbf{s}_i, \mathbf{o}_j, y_{ij}^t\}_{i=1, \dots, n; j=1, \dots, m; t=0, \dots, T}$  are generated to mimic a multistage recommender system on a monotonic chain, with  $d_1 = 3$ ,  $d_2 = 2$ ,  $n = n_1 = 50$ ,  $n_2 = 30$ ,  $n_3 = 50$ ,  $m = m_1 = 100$ , and  $m_2 = 40$ . Then we set  $K = 20$ , and for each  $(l, h, t)$ ,  $\mathbf{a}_{lh} \sim \chi^2(1)$ ,  $\mathbf{b}_{lh} \sim \chi^2(1)$ , and  $\mathbf{q}_t \sim \chi^2(1)$  are independently sampled from the chi-squared distribution on 1 degree of freedom, and  $(s_{il})_{l=1}^3$  and  $(o_{jl})_{l=1}^2$  are uniformly sampled from  $\{1, \dots, n_l\}$  and  $\{1, \dots, m_l\}$ , respectively. Moreover, we set  $y_{ij}^0 = 1$  at stage 0, set  $p_{ij}^1 = \mathbf{a}^\top(\mathbf{s}_i)\mathbf{b}(\mathbf{o}_j) - (\mathbf{a}(\mathbf{s}_i) \circ \mathbf{b}(\mathbf{o}_j))^\top \mathbf{q}_1$ ,  $y_{ij}^1 = \text{sgn}(p_{ij}^1 + \sigma_1 N(0, .1))$  at stage 1, and set  $p_{ij}^2 = \mathbf{a}^\top(\mathbf{s}_i)\mathbf{b}(\mathbf{t}_j) - (\mathbf{a}(\mathbf{s}_i) \circ \mathbf{b}(\mathbf{t}_j))^\top \mathbf{q}_2$ ,  $y_{ij}^2 = \text{sgn}(p_{ij}^2 + \sigma_2 N(0, .1))$  for  $y_{ij}^1 = 1$ , and  $y_{ij}^2 = -1$  for  $y_{ij}^1 = -1$  at stage 2 to satisfy the monotonic chain property (2.1), where  $\sigma_t$  is the standard deviation for  $(p_{ij}^t)_{(i,j) \in \Omega_0}$ ;  $t = 1, 2$ . To generate dataset, we choose indices  $(i, j)$  of  $\Omega_0$  randomly with  $|\Omega_0| = 50000$ , and the resulting missing ratio is 99.98%, and  $\Omega_1$  and  $\Omega_2$  are generated as  $\Omega_1 = \{(i, j) \in \Omega_0 : y_{ij}^1 = 1\}$ ,  $\Omega_2 = \{(i, j) \in \Omega_0 : y_{ij}^2 = 1\}$ . This leads to  $\{\mathbf{s}_i, \mathbf{o}_j, \mathbf{y}_{ij}\}_{(i,j) \in \Omega_0}$ . Finally, the data is split into training, validation, and testing sets with a partition ratio of 10%, 10%, and 80%, respectively.

To avoid overfitting, for gradient boosting, the percentage of features are tuned via a grid search based on the overall misclassification error on the validation set, where the grid set is chosen as  $\{.001, .005, .2, .4, .6, .8, 1.\}$ . For the neural network, its depth is tuned over a set  $\{1, 2, 3, 5, 7, 9, 11\}$ . For SVM, OSVM, and SVD<sup>++</sup>, the penalization parameters are tuned based on validation set through  $\{.001, .005, .01, .05, .1, .5, 1, 5, 10, 50, 100, 500\}$ . For the proposed method,  $K = 20$  and three tuning parameters  $(\lambda_1, \lambda_2, \lambda_3)$  are tuned by minimizing the misclassification error on a validation set via a grid search over  $\lambda_1 \in \{.001, .005, .01\}$ ,  $\lambda_2 \in \{.01, .05, .1\}$ , and  $\lambda_3 \in \{.0001, .0005, .001\}$ . For evaluation, we set all stage weights  $w_{t't} = 1$ .

As suggested by Table 2, the proposed multistage recommender performs the best for one-step, two-step, and three-step forecasting, followed by DeepNN, SVM, and GradBoost. In terms of the overall performance, this same phenomenon is also observed and the amount of improvements of the proposed method are 43%, 22%, 20%, and 13% over Gradient Boosting, OSVM, SVM, and DeepNN. Concerning inconsistency prediction, SVM and OSVM perform the worst, followed by DeepNN and Gradient Boosting. Note that there is no sin-

<sup>5</sup><https://scikit-learn.org>

<sup>6</sup><http://surpriselib.com/>

gle instance in which prediction by the proposed method is inconsistent, which is ensured by the two-level monotonicity property of the classifier.

TABLE 2

Test errors of six competitors and their estimated standard deviations in parentheses on simulated examples over 50 replications as well as a proportion of inconsistent instances violating monotonicity (2.3), denoted by %Inconsist. Here proposed, SVD<sup>++</sup>, GradBoost, SVM, DeepNN and OSVM denote the proposed method in (2.9) with the hinge loss, the latent factor model [16], gradient boosting [11], support vector machine [6], deep learner [20], and conditional ordinal support vector machine in (2.12). The best performer in each case is bold-faced.

(observe $t'$ , predict $t$ )	Proposed	SVD <sup>++</sup>	GradBoost	SVM	DeepNN	OSVM
(Stage 0, Stage 1)	<b>.043(.000)</b>	.095(.001)	.062(.000)	.044(.000)	.044(.000)	.059(.000)
(Stage 0, Stage 2)	<b>.120(.001)</b>	.387(.002)	.217(.000)	.153(.000)	.146(.000)	.165(.000)
(Stage 0, Stage 3)	<b>.148(.000)</b>	.417(.003)	.272(.001)	.187(.000)	.185(.000)	.199(.000)
(Stage 1, Stage 2)	<b>.097(.001)</b>	.294(.002)	.158(.000)	.103(.000)	.100(.000)	.116(.000)
(Stage 1, Stage 3)	<b>.133(.000)</b>	.414(.002)	.246(.001)	.163(.000)	.160(.000)	.175(.000)
(Stage 2, Stage 3)	<b>.095(.001)</b>	.341(.003)	.157(.001)	.101(.000)	.099(.000)	.102(.000)
Overall	<b>.106(.000)</b>	.325(.001)	.185(.000)	.125(.000)	.122(.000)	.136(.000)
%Inconsist	<b>.000%(.000)</b>	.680%(.001)	2.22%(.000)	10.9%(.000)	6.01%(.000)	10.9%(.001)

## 5.2. Real application: Deskdrop article sharing

A *Deskdrop* article sharing dataset contains a sample of 12 months logs and about 73,000 logged user interactions on more than 3,000 public articles. In particular, article-specific features are “article Id”, “author Id”, “plain text”, as well as “language”, and user-specific features include: “user Id”, “user agent”, and “user region”, and users’ actions are logged, including *view*, *like*, and *follow*. The goal is to predict user-item interaction over three stage pairs.

For the proposed method,  $\mathbf{v}_j$  is the numerical embedding based on Doc2Vec [17] of the “plain text” of item  $j$ ,  $\mathbf{o}_j$  consists of “article Id”, “author Id”, and “language” of item  $j$ , and  $\mathbf{s}_i$  is composed of “user Id”, “user agent”, and “user region” of user  $i$ . For other competing methods, we use all features based on a one-hot encoder to convert categorical covariates to zero-one dummy predictors. Moreover, we predict ( $t' = 0, t = 1$ ) based on the proposed method with  $T = 1$ , and predict ( $t' = 0, t = 2$ ) and ( $t' = 1, t = 2$ ) based on the proposed method with  $T = 2$ .

For evaluation, we set all stage weights  $w_{t't} = 1$  and adjust the class weights at each stage inversely proportional to the number of observations in each class since feedback at each stage is highly imbalanced for this data. Hence, all methods are fitted based on balanced class weights or over- or under-sampling, except SVD<sup>++</sup> which based on a regression approach. Furthermore, since the data is highly imbalanced, we also report F1-score for reference in Supplementary (Table 4). Yet, for consistency, we focus on balanced zero-one loss for comparison since all methods are fitted and tuned based on a balanced-weighted classification loss.

The misclassification errors of all competing methods for each ( $t', t$ ) pair are computed based on a nested 15-fold cross-validation with an outer 5-fold loop and an inner 3-fold loop [3]. For the proposed method,  $K = 7$  and three tuning

parameters  $(\lambda_1, \lambda_2, \lambda_3)$  are tuned by minimizing the misclassification error on a validation set via a grid search over  $\lambda_1 \in \{.3, .5, .7, 1\}$ ,  $\lambda_2 \in \{2., 2.5, 3.\}$ , and  $\lambda_3 \in \{.001, .005\}$ . For the neural network, SVM, OSVM, and SVD<sup>++</sup>, the tuning parameters are selected with the same manner as simulation.

As suggested by Table 3, the proposed multistage recommender outperforms SVD<sup>++</sup>, GradBoost, SVM, DeepNN, and OSVM, in terms of the overall performance with the amount of improvement ranging from 75.3% to 8.20%, in terms of the class-balanced zero-one loss. Similarly, for one-step and two-step forward prediction, it is either the best or nearly the best. Concerning inconsistency prediction, all other methods have inconsistent cases ranging from 13.7% to 0.98%. Interestingly, the conditional ordinal method (OSVM) performs similarly to the conditional individual method (SVM), indicating that it does not fully account for the monotonicity property.

TABLE 3

*Class-balanced zero-one losses of six competitors and their estimated standard deviations in parentheses on the Deskdrop benchmark, in addition to %Inconsist denoting a proportion of inconsistent instances violating the monotonicity (2.3). Here proposed, SVD<sup>++</sup>, GradBoost, SVM, DeepNN and OSVM denote the proposed method in (2.9) with the hinge loss, the latent factor model [16], gradient boosting [11], support vector machine [6], deep learner [20], and ordinal support vector machine (2.12). The best performance in each case is bold-faced.*

(observe $t'$ , predict $t$ )	Proposed	SVD <sup>++</sup>	GradBoost	SVM	DeepNN	OSVM
(Stage 0, Stage 1)	<b>0.136(.001)</b>	0.499(.003)	0.166(.000)	0.139(.000)	0.154(.001)	0.140(.000)
(Stage 0, Stage 2)	<b>0.191(.001)</b>	0.499(.007)	0.218(.001)	0.218(.001)	0.223(.006)	0.216(.001)
(Stage 1, Stage 2)	<b>0.043(.000)</b>	0.497(.007)	0.049(.000)	0.045(.000)	0.139(.011)	0.045(.000)
Overall	<b>0.123(.001)</b>	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)

## 6. Discussion

The article introduces a two-monotonic recommender system for monotonic sequential chain, which constructs a positive constrained latent factor model to consider pairwise prediction among different stages. Numerically, we design a parallel algorithm based on blockwise coordinate descent to decompose the large-scale optimization as many linear SVMs. Theoretically, we establish a learning theory to demonstrate that the proposed method is advantageous to the standard separate approaches and one-way ordinal approaches in terms of the prediction accuracy. Although the advantage of the proposed method has been demonstrated empirically and theoretically, further investigation may be necessary. In particular, how deep learning models in recommender systems could utilize the monotonic property to further improve the performance.

## Appendix A: Technical proofs

**Proof of Lemma 2.1.** It suffices to show that  $\bar{\phi}_{ij}^{t't}$  is a global minimizer of  $l_{ij}(\mathbf{f}_{ij})$ . By (2.1),

$$\begin{aligned} & \mathbb{E}_{ij}(\mathbb{I}(Y_{ij}^t \phi_{ij}^{t't}(\mathbf{x}_{ij}, Y_{ij}^{t'}) \leq 0) | \Delta_{ij} = 1) \\ &= \mathbb{P}_{ij}(Y_{ij}^{t'} = 1 | \Delta_{ij} = 1) \mathbb{E}_{ij}(\mathbb{I}(Y_{ij}^t \phi_{ij}^{t't}(\mathbf{x}_{ij}, 1) \leq 0) | \Delta_{ij} = 1, Y_{ij}^{t'} = 1) \\ & \quad + \mathbb{P}_{ij}(Y_{ij}^{t'} = -1 | \Delta_{ij} = 1) \mathbb{E}_{ij}(\mathbb{I}(-\phi_{ij}^{t't}(\mathbf{x}_{ij}, -1) \leq 0) | \Delta_{ij} = 1, Y_{ij}^{t'} = -1), \end{aligned} \tag{A.1}$$

which yields that  $\bar{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, y_{ij}^t)$  is a global minimizer of (A.1), since  $\bar{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, 1) = \text{sgn}(\mathbb{P}_{ij}(Y_{ij}^t = 1 | Y_{ij}^{t'} = 1, \Delta_{ij} = 1) - 1/2)$  minimizes  $\mathbb{E}_{ij}(\mathbb{I}(Y_{ij}^t \phi_{ij}^{t't}(\mathbf{x}_{ij}, 1) \leq 0) | Y_{ij}^{t'} = 1, \Delta_{ij} = 1)$  and  $\bar{\phi}_{ij}^{t't}(\mathbf{x}_{ij}, -1) = -1$  minimizes  $\mathbb{E}_{ij}(\mathbb{I}(-\phi_{ij}^{t't}(\mathbf{x}_{ij}, -1) \leq 0) | \Delta_{ij} = 1, Y_{ij}^{t'} = -1)$ .

Next, we verify the two-level monotonic property (2.3) of  $\bar{f}_{ij}^{t't}$ . Note that

$$\begin{aligned} \mathbb{P}_{ij}(Y_{ij}^t = 1 | \Delta_{ij} = 1) &= \mathbb{P}_{ij}(Y_{ij}^t = 1, Y_{ij}^{t-1} = 1 | \Delta_{ij} = 1) \\ & \quad + \mathbb{P}_{ij}(Y_{ij}^t = 1, Y_{ij}^{t-1} = -1 | \Delta_{ij} = 1) \\ &= \mathbb{P}_{ij}(Y_{ij}^t = 1 | Y_{ij}^{t-1} = 1, \Delta_{ij} = 1) \mathbb{P}_{ij}(Y_{ij}^{t-1} = 1 | \Delta_{ij} = 1) \\ & \leq \mathbb{P}_{ij}(Y_{ij}^{t-1} = 1 | \Delta_{ij} = 1), \end{aligned} \tag{A.2}$$

which implies that  $\mathbb{P}_{ij}(Y_{ij}^t = 1 | \Delta_{ij} = 1)$  is nondecreasing over stage  $t$ . Moreover,

$$\begin{aligned} \mathbb{P}_{ij}(Y_{ij}^t = 1 | Y_{ij}^{t'} = 1, \Delta_{ij} = 1) &= \frac{\mathbb{P}_{ij}(Y_{ij}^t = 1, Y_{ij}^{t'} = 1 | \Delta_{ij} = 1)}{\mathbb{P}_{ij}(Y_{ij}^{t'} = 1 | \Delta_{ij} = 1)} \\ &= \frac{\mathbb{P}_{ij}(Y_{ij}^t = 1 | \Delta_{ij} = 1)}{\mathbb{P}_{ij}(Y_{ij}^{t'} = 1 | \Delta_{ij} = 1)}, \end{aligned}$$

provided that  $\mathbb{P}_{ij}(Y_{ij}^{t'} = 1 | \Delta_{ij} = 1) \neq 0$ , which is non-decreasing when  $t$  increases while  $t'$  is fixed because  $\mathbb{P}_{ij}(Y_{ij}^t = 1 | \Delta_{ij} = 1)$  is non-decreasing in  $t$ , and is non-increasing when  $t'$  increases while  $t$  is fixed because  $\frac{1}{\mathbb{P}_{ij}(Y_{ij}^{t'} = 1 | \Delta_{ij} = 1)}$  is non-increasing in  $t'$ . This implies the two-level monotonic property.

Next, we construct an alternative form of the Bayes decision function  $\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})$ , which is additive in stage  $t$ :

$$\bar{f}_{ij}^{t't}(\mathbf{x}_{ij}) = \bar{h}_{ij}^0(\mathbf{x}_{ij}) - \sum_{r=t'+1}^t \bar{h}_{ij}^r(\mathbf{x}_{ij}),$$

where  $\bar{h}_{ij}^0(\mathbf{x}_{ij}) = c_{ij}(\mathbf{x}_{ij}) \log_{\alpha_{ij}(\mathbf{x}_{ij})}(2) > 0$ , and  $\bar{h}_{ij}^r(\mathbf{x}_{ij}) > 0$  defined as

$$\bar{h}_{ij}^r(\mathbf{x}_{ij}) = c_{ij}(\mathbf{x}_{ij}) \log_{\alpha_{ij}(\mathbf{x}_{ij})}(\mathbb{P}_{ij}(Y_{ij}^{t-1} = 1 | \Delta_{ij} = 1) / \mathbb{P}_{ij}(Y_{ij}^t = 1 | \Delta_{ij} = 1)),$$

and  $c_{ij}(\mathbf{x}_{ij}) > 0$  and  $\alpha_{ij}(\mathbf{x}_{ij}) > 0$  are two arbitrary positive functions. Clearly,  $\bar{f}_{ij}^{t't}$  automatically satisfies the two-level monotonic property. Now, we need to prove that  $\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})$  has the same sign of  $\tilde{f}_{ij}^{t't}(\mathbf{x}_{ij})$ , that is,  $\text{sgn}(\bar{f}_{ij}^{t't}(\mathbf{x}_{ij})) = \text{sgn}(c_{ij}(\mathbf{x}_{ij}) \log_{\alpha_{ij}(\mathbf{x}_{ij})}(2\mathbb{P}_{ij}(Y_{ij}^t = 1|Y_{ij}^{t'} = 1, \Delta_{ij} = 1))) = 1$ , when  $\mathbb{P}_{ij}(Y_{ij}^t = 1|Y_{ij}^{t'} = 1, \Delta_{ij} = 1) > 1/2$  and equals to  $-1$  otherwise. The desired result then follows.  $\square$

**Proof of Lemma 2.2.** The result follows from the fact that  $\bar{\mathbf{f}}_{ij}^{t't}$  is a global minimizer of  $\mathbb{E}_{ij}(L(\mathbf{f}_{ij}, \mathbf{Z}_{ij}))$ , when  $V(\cdot)$  is Fisher consistency for binary classification.

**Proof of Lemma 3.1.** Note that (3.1), (3.3), (3.4), (3.5), and (3.6) are convex minimization problems. Then convergence of Algorithm 1 follows from [26].  $\square$

**Proof of Theorem 4.1.** Our treatment of bounding  $\mathbb{P}(e(\hat{\phi}) - e(\bar{\phi}) \geq \epsilon_{nm}^2)$  relies on a chain argument of empirical process over a suitable partition of  $\mathcal{F}$  induced by  $e(\hat{\phi}) - e(\bar{\phi})$ , as in [8, 31]. For  $u \geq 1$  and  $v \geq 0$ , define  $A_{uv} = \{\mathbf{f} \in \mathcal{F} : 2^{u-1}\epsilon_{nm}^2 \leq l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq 2^u\epsilon_{nm}^2, 2^{v-1}\bar{J} \leq J(\mathbf{f}) \leq 2^v\bar{J}\}$  and  $A_{u0} = \{\mathbf{f} \in \mathcal{F} : 2^{u-1}\epsilon_{nm}^2 \leq l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq 2^u\epsilon_{nm}^2, J(\mathbf{f}) \leq \bar{J}\}$ . By Assumption A,

$$\begin{aligned} \mathbb{P}(e(\hat{\mathbf{f}}) - e(\bar{\mathbf{f}}) \geq c_2\epsilon_{nm}^{2\alpha}) &\leq \mathbb{P}(l_B(\hat{\mathbf{f}}) - l_B(\bar{\mathbf{f}}) \geq \epsilon_{nm}^2) \\ &\leq \mathbb{P}\left(\sup_{\{\mathbf{f} \in \mathcal{F}; l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \geq \epsilon_{nm}^2\}} (nm)^{-1} \sum_{i,j} (\tilde{L}(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - \tilde{L}(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \geq 0\right) \\ &\leq \mathbb{P}\left(\sup_{\cup_{u=1}^{\infty} \cup_{v=0}^{\infty} A_{uv}} (nm)^{-1} \sum_{i,j} (\tilde{L}_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - \tilde{L}_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \geq 0\right) \\ &\leq \sum_{u=1}^{\infty} \sum_{v=1}^{\infty} \mathbb{P}\left(\sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \geq \delta_{uv}\right) \\ &\quad + \sum_{u=1}^{\infty} \mathbb{P}\left(\sup_{\{\mathbf{f} \in A_{u0}\}} (nm)^{-1} \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \geq \delta_{u0}\right) = \sum_{u=1}^{\infty} \sum_{v=1}^{\infty} I_{uv} + \sum_{u=1}^{\infty} I_{u0}, \end{aligned} \tag{A.3}$$

where  $g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) = L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) - (L_B(\bar{\mathbf{f}}_{ij}) - L_B(\mathbf{f}_{ij}))$ ,  $\delta_{uv} = 2^{u-1}\epsilon_{nm}^2 + \lambda(2^{v-1} - 1)\bar{J}$ ,  $\delta_{u0} = 2^{u-2}\epsilon_{nm}^2$ ,

$$I_{uv} = \mathbb{P}\left(\sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \geq \delta_{uv}\right),$$

and  $\lambda J(\mathbf{f}) = \lambda_1 \|\boldsymbol{\beta}\|_2^2 + \sum_{k=1}^K \lambda_k (\|\mathbf{a}^k\|_2^2 + \|\mathbf{b}^k\|_2^2)$ .

To apply Talagrand's inequality [12] to  $I_{uv}$ , let

$$\sigma_{uv}^2 = \sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \sum_{i,j} \text{Var}(g(\mathbf{f}_{ij}, \mathbf{Z}_{ij})), \quad u \geq 1, v \geq 0.$$

Then

$$I_{uv} \leq \mathbb{P}\left(\sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \geq \delta_{uv}\right)$$

$$\begin{aligned}
&\leq \mathbb{P}\left(\sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \left| \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right| \right. \\
&\quad \left. - \mathbb{E} \sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \left| \sum_{i,j} g(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right| \leq \delta_{uv} - 2\mathbb{E}(\mathcal{R}_{uv})\right) \\
&\leq a_1 \exp\left(-\frac{nm\delta_{uv}^2}{2a_1(2\sigma_{uv}^2 + 16B\mathbb{E}(\mathcal{R}_{uv}) + B\delta_{uv})}\right) \\
&\leq a_1 \exp\left(-\frac{nm\delta_{uv}^2}{2a_1(4^\mu c_2 \delta_{uv}^\mu + 5B\delta_{uv})}\right) \\
&\leq a_1 \exp\left(-\frac{nm\delta_{uv}^{2-\min(1,\mu)}}{2a_1 a_2}\right), \tag{A.4}
\end{aligned}$$

where  $a_2 = 32c_2 B^{\mu-\min(1,\mu)} + 9B^{2-\min(1,\mu)}$  and the second inequality follows from the fact that

$$\begin{aligned}
&\mathbb{E}\left(\frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right. \right. \\
&\quad \left. \left. - \mathbb{E}(L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \right| \right) \\
&\leq \mathbb{E}\left(\frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \mathbb{E}\left(\sum_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right. \right. \right. \\
&\quad \left. \left. - (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}'_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}'_{ij})) \mid \mathbf{Z}\right) \right| \right) \\
&\leq \mathbb{E}\left(\frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right. \right. \\
&\quad \left. \left. - (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}'_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}'_{ij})) \right| \right) \\
&\leq \mathbb{E}\left(\frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} \tau_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) \right. \right. \\
&\quad \left. \left. - (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}'_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}'_{ij})) \right| \right) \\
&\leq 2\mathbb{E}\left(\frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} \tau_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \right| \right) \leq 2\mathbb{E}(\mathcal{R}_{uv}),
\end{aligned}$$

where  $\mathbf{Z}'_{ij}$  is an independent copy of  $\mathbf{Z}_{ij}$ ,

$$\mathcal{R}_{uv} = \mathbb{E}_\tau \left( \frac{1}{nm} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} \tau_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \right| \right)$$

is the Rademacher complexity of  $A_{uv}$ , and  $\tau_{ij}$  are independent random variables drawn from the Rademacher distribution. The third inequality follows from the Talagrand's inequality and the last inequality follows from the fact that,

$$\sigma_{uv}^2 = \sup_{\{\mathbf{f} \in A_{uv}\}} (nm)^{-1} \sum_{i,j} \text{Var}_{ij} (L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}) - L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}))$$

$$\leq c_2(2^u \epsilon_{nm}^2)^\mu \leq 16c_2 \delta_{uv}^\mu. \tag{A.5}$$

A combination of (A.3) and (A.4) yield

$$\begin{aligned} \mathbb{P}(e(\hat{\mathbf{f}}) - e(\bar{\mathbf{f}}) \leq c_2 \epsilon_{n,m}^{2\alpha}) &\leq \sum_{u=1}^\infty \sum_{v=1}^\infty I_{uv} + \sum_{u=1}^\infty I_{u0} \\ &\leq \sum_{i=1}^\infty \sum_{v=1}^\infty a_1 \exp\left(-\frac{nm \delta_{uv}^{2-\min(1,\mu)}}{2a_1 a_2}\right) + \sum_{u=1}^\infty a_1 \exp\left(-\frac{nm \delta_{u0}^{2-\min(1,\mu)}}{2a_1 a_2}\right) \\ &\leq 4a_1 \exp\left(-\frac{nm(\lambda \bar{J})^{2-\min(1,\mu)}}{4a_1 a_2}\right). \end{aligned}$$

The desired result follows immediately. □

**Lemma A.1.** Let  $A_{uv} = \{\mathbf{f} \in \mathcal{F} : 2^{u-1} \epsilon_{nm}^2 \leq l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq 2^u \epsilon_{nm}^2, 2^{v-1} \bar{J} \leq J(\mathbf{f}) \leq 2^v \bar{J}\}$ ,  $A_{u0} = \{\mathbf{f} \in \mathcal{F} : 2^{u-1} \epsilon_{nm}^2 \leq l_B(\mathbf{f}) - l_B(\bar{\mathbf{f}}) \leq 2^u \epsilon_{nm}^2, J(\mathbf{f}) \leq \bar{J}\}$ , and  $\delta_{uv} = 2^{u-1} \epsilon_{nm}^2 + \lambda(2^{v-1} - 1)\bar{J}$ , and  $\delta_{u0} = 2^{u-2} \epsilon_{nm}^2$ , where

$$\epsilon_{nm}^2 \sim \left(\frac{(\Lambda + T)K}{nm} \log\left(\frac{nm \bar{w} B \bar{J} \kappa(6T^3 + T^2 K)}{(\Lambda + T)K}\right)\right)^{\frac{1}{2-\min(1,\mu)}}.$$

Then

$$4\mathbb{E}(\mathcal{R}_{uv}) \leq \delta_{uv}, \tag{A.6}$$

where  $\mathcal{R}_{uv} = \mathbb{E}_\tau\left((nm)^{-1} \sup_{\{\mathbf{f} \in A_{uv}\}} \left| \sum_{i,j} \tau_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) \right| \right)$  is the Rademacher complexity of  $A_{uv}$ .

**Proof of Lemma A.1.** To bound  $\mathbb{E}(\mathcal{R}_{uv})$ , we compute the metric entropy for

$$\mathcal{L}_{uv} = \left\{ (nm)^{-1} \sum_{i,j} (L_B(\bar{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij})) : \mathbf{f} \in A_{uv} \right\}.$$

For any functions  $\tilde{\mathbf{f}}$  and  $\mathbf{f}$ ,

$$\begin{aligned} &(nm)^{-1} \sum_{i,j} (L_B(\tilde{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}))^2 \\ &= \left( \sum_{t' < t} w_{t't} (V_B(Y_{ij}^t \tilde{f}_{ij}^{t't}(\mathbf{x}_{ij})) - V_B(Y_{ij}^t f_{ij}^{t't}(\mathbf{x}_{ij}))) \right)^2 \\ &\leq \bar{w} \left( \sum_{t' < t} (\tilde{f}_{ij}^{t't}(\mathbf{x}_{ij}) - f_{ij}^{t't}(\mathbf{x}_{ij}))^2 \right), \end{aligned} \tag{A.7}$$

where  $\bar{w} = \sum_{0 \leq t' < t \leq T} w_{t't}^2$ , and the first inequality follows from the Cauchy-Schwarz inequality. Next, we turn to bound  $\tilde{f}_{ij}^{t't}(\mathbf{x}_{ij}) - f_{ij}^{t't}(\mathbf{x}_{ij})$ ,

$$\begin{aligned} &(\tilde{f}_{ij}^{t't}(\mathbf{x}_{ij}) - f_{ij}^{t't}(\mathbf{x}_{ij}))^2 \\ &= \left( (\tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j))^\top \tilde{q}(t', t) - (a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j))^\top q(t', t) \right)^2 \end{aligned}$$

$$\begin{aligned} &\leq 2 \left( (\tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j))^\top \left( \sum_{r=t'+1}^t (\mathbf{q}_r - \tilde{\mathbf{q}}_r) \right) \right)^2 \\ &\quad + 2 \left( (\tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j) - a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j))^\top q(t', t) \right)^2 = I_1 + I_2, \end{aligned}$$

where  $q(t', t) = \mathbf{1} - \sum_{r=t'+1}^t \mathbf{q}_r$ . Then we bound  $I_1$  and  $I_2$  separately. For  $I_1$ ,

$$\begin{aligned} I_1 &\leq 2 \|\tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j)\|_2^2 \left\| \sum_{r=t'+1}^t \mathbf{q}_r - \tilde{\mathbf{q}}_r \right\|_2^2 \\ &\leq 2 \|\tilde{a}(\mathbf{u}_i, \mathbf{s}_i)\|_2^2 \|\tilde{b}(\mathbf{v}_j, \mathbf{o}_j)\|_2^2 \left\| \sum_{r=t'+1}^t \mathbf{q}_r - \tilde{\mathbf{q}}_r \right\|_2^2 \\ &\leq 8(t-t')\rho_1\rho_2(\|\mathbf{A}\|_F^2 + \|\mathbf{a}\|_2^2)(\|\mathbf{B}\|_F^2 + \|\mathbf{b}\|_2^2) \left( \sum_{r=t'+1}^t \|\mathbf{q}_r - \tilde{\mathbf{q}}_r\|_2^2 \right) \\ &\leq 2^{2v+3}\bar{J}^2(t-t')\rho_1\rho_2 \left( \sum_{r=t'+1}^t \|\mathbf{q}_r - \tilde{\mathbf{q}}_r\|_2^2 \right), \end{aligned} \tag{A.8}$$

where  $\rho_i = p_i + d_i$ ; ( $i = 1, 2$ ), the last inequality follows from the fact that

$$\|a(\mathbf{u}_i, \mathbf{s}_i)\|_2^2 \leq 2(\|\mathbf{A}\|_F^2 \|\mathbf{u}_i\|_2^2 + d_1 \sum_{l=1}^{d_1} \max_{i=1, \dots, n_i} \|\mathbf{a}_{li}\|_2^2) \leq 2(p_1 \|\mathbf{A}\|_F^2 + d_1 \|\mathbf{a}\|_2^2),$$

and

$$\|b(\mathbf{v}_j, \mathbf{o}_j)\|_2^2 \leq 2(\|\mathbf{B}\|_F^2 \|\mathbf{v}_j\|_2^2 + d_2 \sum_{l=1}^{d_2} \max_{j=1, \dots, m_l} \|\mathbf{b}_{lj}\|_2^2) \leq 2(p_2 \|\mathbf{B}\|_F^2 + d_2 \|\mathbf{b}\|_2^2).$$

For  $I_2$ ,

$$\begin{aligned} I_2 &\leq 2 \left( (\tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j) - a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j))^\top q(t', t) \right)^2 \\ &\leq 2 \left\| \mathbf{1} - \sum_{r=t'+1}^t \mathbf{q}_r \right\|_2^2 \left\| \tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j) - a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j) \right\|_2^2 \\ &\leq 4(K + (t-t') \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2) \left\| \tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ \tilde{b}(\mathbf{v}_j, \mathbf{o}_j) - \tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j) \right\|_2^2 \\ &\quad + 4(K + (t-t') \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2) \left\| \tilde{a}(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j) - a(\mathbf{u}_i, \mathbf{s}_i) \circ b(\mathbf{v}_j, \mathbf{o}_j) \right\|_2^2 \\ &\leq 4(K + (t-t') \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2) \|\tilde{a}(\mathbf{u}_i, \mathbf{s}_i)\|_2^2 \left\| (\tilde{\mathbf{B}} - \mathbf{B})\mathbf{v}_j - \sum_{l=1}^{d_2} (\tilde{\mathbf{b}}_{lo_l} - \mathbf{b}_{lo_l}) \right\|_2^2 \end{aligned}$$

$$\begin{aligned}
& + 4(K + (t - t')) \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2 \|b(\mathbf{v}_j, \mathbf{o}_j)\|_2^2 \|(\tilde{\mathbf{A}} - \mathbf{A})\mathbf{u}_i - \sum_{l=1}^{d_1} (\tilde{\mathbf{a}}_{lsil} - \mathbf{a}_{lsil})\|_2^2 \\
& \leq 2^{v+4} \kappa \bar{J} (K + (t - t')) \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2 \\
& \quad \left( p_1 \|\tilde{\mathbf{A}} - \mathbf{A}\|_F^2 + p_2 \|\tilde{\mathbf{B}} - \mathbf{B}\|_F^2 + d_1 \|\tilde{\mathbf{a}} - \mathbf{a}\|_2^2 + d_2 \|\tilde{\mathbf{b}} - \mathbf{b}\|_2^2 \right), \quad (\text{A.9})
\end{aligned}$$

where  $\kappa = p_1 + p_2 + d_1 + d_2$ . Combining (A.7), (A.8) and (A.9),

$$\begin{aligned}
& (nm)^{-1} \sum_{i,j} (L_B(\tilde{\mathbf{f}}_{ij}, \mathbf{Z}_{ij}) - L_B(\mathbf{f}_{ij}, \mathbf{Z}_{ij}))^2 \\
& \leq \bar{w} \left( 2^{2v+6} \bar{J}^2 (p_1 + d_1)(p_2 + d_2) \sum_{t' < t} (t - t') \sum_{r=t'+1}^t \|\mathbf{q}_r - \tilde{\mathbf{q}}_r\|_2^2 \right. \\
& \quad \left. + \kappa^2 2^{v+4} \bar{J} (\|\tilde{\mathbf{A}} - \mathbf{A}\|_F^2 + \|\tilde{\mathbf{B}} - \mathbf{B}\|_F^2 + \|\tilde{\mathbf{a}} - \mathbf{a}\|_2^2 + \|\tilde{\mathbf{b}} - \mathbf{b}\|_2^2) \right. \\
& \quad \left. \left( \sum_{t' < t} (K + (t - t')) \sum_{r=t'+1}^t \|\mathbf{q}_r\|_2^2 \right) \right) \\
& \leq \bar{w} \left( 2^{2v+2} \bar{J}^2 T^3 (p_1 + d_1)(p_2 + d_2) \|\mathbf{q} - \tilde{\mathbf{q}}\|_2^2 \right. \\
& \quad \left. + \kappa^2 2^v \bar{J} ((T + 1)TK + T^3 \|\mathbf{q}\|_2^2) \right. \\
& \quad \left. (\|\tilde{\mathbf{A}} - \mathbf{A}\|_F^2 + \|\tilde{\mathbf{B}} - \mathbf{B}\|_F^2 + \|\tilde{\mathbf{a}} - \mathbf{a}\|_2^2 + \|\tilde{\mathbf{b}} - \mathbf{b}\|_2^2) \right).
\end{aligned}$$

Therefore, the metric entropy for  $\mathcal{L}_{uv}$  for  $\nu \leq 1$  is upper bounded:

$$\log \mathcal{N}(\nu, \mathcal{L}_{uv}) \leq \log \mathcal{N}\left(\frac{\nu}{2^v \bar{J} B a_3}, \mathcal{B}_+((\Lambda + T)K)\right) \leq \gamma \log\left(\frac{2^{v+1} \bar{J} B a_3}{\nu}\right), \quad (\text{A.10})$$

where  $a_3 = \kappa \sqrt{\bar{w}(6T^3 + (T + 1)TK)}$ ,  $\mathcal{B}_+((\Lambda + T)K)$  is the unit  $l_2$ -ball in  $\mathbb{R}_+^{(\Lambda+T)K}$ . Therefore, the Rademacher complexity  $\mathbb{E}(\mathcal{R}_{uv})$  can be computed based on Theorem 3.12 of [15]. Specifically, there exists a constant  $a_4$ , such that

$$\mathbb{E}(\mathcal{R}_{uv}) \leq a_4 \max\left(\frac{\delta_{uv}^{\mu/2} \sqrt{(\Lambda + T)K}}{\sqrt{nm}} \sqrt{\log\left(\frac{2^{v+1} \bar{J} B a_3}{\delta_{uv}^{\mu/2}}\right)}, \frac{B(\Lambda + T)K}{nm} \log\left(\frac{2^{v+1} \bar{J} B a_3}{\delta_{uv}^{\mu/2}}\right)\right).$$

Then there exists a constant  $a_5 > 0$ , such that

$$\epsilon_{nm}^2 = a_5 \left( \frac{(\Lambda + T)K}{nm} \log\left(\frac{nm \bar{w} B \bar{J} \kappa (6T^3 + T^2 K)}{(\Lambda + T)K}\right) \right)^{\frac{1}{2 - \min(1, \mu)}},$$

and

$$\begin{aligned}
& \frac{\delta_{uv}}{4\mathbb{E}(\mathcal{R}_{uv})} \\
& \geq \frac{1}{4a_4} \min \left( \frac{\sqrt{nm}\delta_{uv}^{1-\mu/2}}{\sqrt{(\Lambda+T)K}} \left( \log \left( \frac{2^{v+1}\bar{J}Ba_3}{\delta_{uv}^{\mu/2}} \right) \right)^{-1/2}, \right. \\
& \qquad \qquad \qquad \left. \frac{nm\delta_{uv}}{B(\Lambda+T)K} \left( \log \left( \frac{2^{v+1}\bar{J}Ba_3}{\delta_{uv}^{\mu/2}} \right) \right)^{-1} \right) \\
& \geq \frac{1}{4a_4} \min \left( \frac{\sqrt{nm}\delta_{u0}^{1-\mu/2}}{\sqrt{(\Lambda+T)K}} \left( \log \left( \frac{2\bar{J}Ba_3}{\delta_{u0}^{\mu/2}} \right) \right)^{-1/2}, \frac{nm\delta_{u0}}{B(\Lambda+T)K} \left( \log \left( \frac{2\bar{J}Ba_3}{\delta_{u0}^{\mu/2}} \right) \right)^{-1} \right) \\
& \geq \frac{1}{4a_4} \min \left( \frac{\sqrt{nm}\epsilon_{nm}^{2-\mu}}{\sqrt{(\Lambda+T)K}} \left( \log \left( \frac{2\bar{J}Ba_3}{\epsilon_{nm}^{\mu}} \right) \right)^{-1/2}, \frac{nm\epsilon_{nm}^2}{B(\Lambda+T)K} \left( \log \left( \frac{2\bar{J}Ba_3}{\epsilon_{nm}^{\mu}} \right) \right)^{-1} \right) \\
& \geq 1.
\end{aligned}$$

where the first two inequalities follow from non-decreasing of  $(u, v)$ . This completes the proof.  $\square$

## Supplementary Material

### F-1 score for Deskdrop article sharing

F1 scores of six competitors on the *Deskdrop* dataset are reported in Table 4.

TABLE 4

*F1 scores of six competitors and their estimated standard deviations in parentheses on the Deskdrop benchmark. Here proposed, SVD<sup>++</sup>, GradBoost, SVM, DeepNN and OSVM denote the proposed method in (2.9) with the hinge loss, the latent factor model [16], gradient boosting [11], support vector machine [6], deep learner [20], and ordinal support vector machine (2.12). The best performance in each case is bold-faced.*

(observe $t'$ , predict $t$ )	Proposed	SVD <sup>++</sup>	GradBoost	SVM	DeepNN	OSVM
(Stage 0, Stage 1)	0.561(.006)	0.003(.001)	<b>0.569(.005)</b>	0.551(.005)	0.523(.004)	0.552(.004)
(Stage 0, Stage 2)	<b>0.152(.006)</b>	0.000(.000)	0.145(.005)	0.151(.007)	0.121(.001)	0.151(.006)
(Stage 1, Stage 2)	0.329(.005)	0.010(.003)	0.329(.005)	0.328(.006)	<b>0.371(.008)</b>	0.328(.006)
Overall	<b>0.348(.005)</b>	0.004(.001)	<b>0.348(.005)</b>	0.344(.005)	0.338(.004)	0.344(.005)

## References

- [1] BHASKAR, S. A. Probabilistic low-rank matrix completion from quantized measurements. *The Journal of Machine Learning Research* 17, 1 (2016), 2131–2164. [MR3504620](#)
- [2] BI, X., QU, A., AND SHEN, X. Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics* 46, 6B (2018), 3308 – 3333. [MR3852653](#)
- [3] CAWLEY, G. C., AND TALBOT, N. L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research* 11, Jul (2010), 2079–2107. [MR2678023](#)

- [4] CHEN, B., HE, S., LI, Z., AND ZHANG, S. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization* 22, 1 (2012), 87–107. [MR2902686](#)
- [5] CHENG, C., XIA, F., ZHANG, T., KING, I., AND LYU, M. R. Gradient boosting factorization machines. In *Proceedings of the 8th ACM Conference on Recommender systems* (2014), pp. 265–272.
- [6] CORTES, C., AND VAPNIK, V. Support vector networks. *Machine Learning* 20, 3 (1995), 273–297.
- [7] CRAMMER, K., AND SINGER, Y. Pranking with ranking. In *Advances in Neural Information Processing Systems* (2002), pp. 641–647.
- [8] DAI, B., SHEN, X., WANG, J., AND QU, A. Scalable collaborative ranking for personalized prediction. *Journal of the American Statistical Association* (Jan. 2019). [MR4309267](#)
- [9] DAI, B., WANG, J., SHEN, X., AND QU, A. Smooth neighborhood recommender systems. *Journal of Machine Learning Research* 20, 16 (2019), 1–24. [MR3911423](#)
- [10] EDWARDS, D. *Introduction to graphical modelling*. Springer Science & Business Media, 2012. [MR1880319](#)
- [11] FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232. [MR1873328](#)
- [12] GINÉ, E., KOLTCHINSKII, V., ET AL. Concentration inequalities and asymptotic results for ratio type empirical processes. *Annals of Probability* 34, 3 (2006), 1143–1216. [MR2243881](#)
- [13] HAEFFELE, B. D., AND VIDAL, R. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540* (2015).
- [14] HE, X., LIAO, L., ZHANG, H., NIE, L., HU, X., AND CHUA, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (2017), pp. 173–182.
- [15] KOLTCHINSKII, V. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d’Eté de Probabilités de Saint-Flour XXXVIII-2008*, vol. 2033. Springer Science & Business Media, 2011. [MR2829871](#)
- [16] KOREN, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2008), pp. 426–434.
- [17] LE, Q., AND MIKOLOV, T. Distributed representations of sentences and documents. In *International Conference on Machine Learning* (2014), pp. 1188–1196.
- [18] MAZUMDER, R., HASTIE, T., AND TIBSHIRANI, R. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research* 11 (2010), 2287–2322. [MR2719857](#)
- [19] MCCULLAGH, P., AND NELDER, J. A. *Generalized linear models*. Routledge, 2019. [MR3223057](#)
- [20] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

- [21] SHEN, X., TSENG, G. C., ZHANG, X., AND WONG, W. H. On  $\psi$ -learning. *Journal of the American Statistical Association* 98, 463 (2003), 724–734. [MR2011686](#)
- [22] SUN, W. W., LU, J., LIU, H., AND CHENG, G. Provable sparse tensor decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79, 3 (2017), 899–916. [MR3641413](#)
- [23] SUPHAVILAI, C., BERTRAND, D., AND NAGARAJAN, N. Predicting cancer drug response using a recommender system. *Bioinformatics* 34, 22 (2018), 3907–3914.
- [24] TANG, X., BI, X., AND QU, A. Individualized multilayer tensor learning with an application in imaging analysis. *Journal of the American Statistical Association* 115, 530 (2020), 836–851. [MR4107683](#)
- [25] TRAN, T., PHUNG, D. Q., AND VENKATESH, S. A sequential decision approach to ordinal preferences in recommender systems. In *Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012).
- [26] TSENG, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109, 3 (2001), 475–494. [MR1835069](#)
- [27] TSYBAKOV, A. B., ET AL. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics* 32, 1 (2004), 135–166. [MR2051002](#)
- [28] WAN, M., AND MCAULEY, J. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), ACM, pp. 86–94.
- [29] WANG, J., WONG, R. K., MAO, X., AND CHAN, K. C. G. Matrix completion with model-free weighting. In *Proceedings of the 38th International Conference on Machine Learning* (2021). [MR4030156](#)
- [30] WEINBERGER, K., DASGUPTA, A., ATTENBERG, J., LANGFORD, J., AND SMOLA, A. Feature hashing for large scale multitask learning. *arXiv preprint arXiv:0902.2206* (2009).
- [31] WONG, W. H., SHEN, AND XIAOTONG. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *Annals of Statistics* 23, 2 (1995), 339–362. [MR1332570](#)
- [32] WRIGHT, S. J. Coordinate descent algorithms. *Mathematical Programming* 151, 1 (2015), 3–34. [MR3347548](#)
- [33] ZHANG, C., AND LIU, Y. Multicategory angle-based large-margin classification. *Biometrika* 101, 3 (2014), 625–640. [MR3254905](#)
- [34] ZHU, J., AND HASTIE, T. Kernel logistic regression and the import vector machine. In *Advances in Neural Information Processing Systems* (2002), pp. 1081–1088.