Bilinear Bandits with Partially Observable Features

Wooseong Cho   Ji Hyeong Park   Min-hwan Oh

Seoul National University

Problem Setting

Static side information: partially observed user and item features

$$\boldsymbol{x}_i^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{u}_i \end{bmatrix} \\ \text{latent} \end{array} \qquad \boldsymbol{y}_j^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{v}_j \end{bmatrix} \\ \text{latent} \end{array}$$

Problem Setting

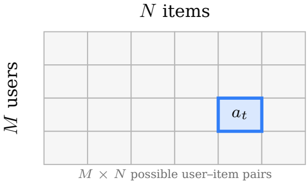
Static side information: partially observed user and item features

$$\boldsymbol{x}_i^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{u}_i \end{bmatrix} \\ \text{latent} \end{array} \qquad \boldsymbol{y}_j^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{v}_j \end{bmatrix} \\ \text{latent} \end{array}$$

For each round $t = 1, \dots, T$:

1. Choose one user–item pair: $a_t = (i_t, j_t) \in [M] \times [N]$

2. Observe reward for the chosen pair: $r_{a_t,t}$



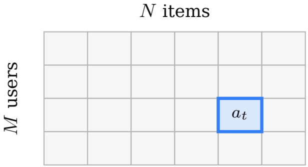
Problem Setting

Static side information: partially observed user and item features

$$\boldsymbol{x}_i^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{u}_i \end{bmatrix} \\ \text{latent} \end{array} \qquad \boldsymbol{y}_j^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{v}_j \end{bmatrix} \\ \text{latent} \end{array}$$

For each round $t = 1, \ldots, T$:

1. Choose one user–item pair: $a_t = (i_t, j_t) \in [M] \times [N]$
2. **Observe reward for the chosen pair:** $r_{a_t,t}$



$M \times N$ possible user–item pairs

$$r_{a_t,t} = {\boldsymbol{x}_{i_t}^*}^\top \boldsymbol{\Theta}_* \boldsymbol{y}_{j_t}^* + \varepsilon_{a_t,t}$$

bilinear in full user and item features

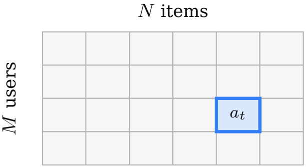
Problem Setting

Static side information: partially observed user and item features

$$\boldsymbol{x}_i^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{u}_i \end{bmatrix} \\ \text{latent} \end{array} \qquad \boldsymbol{y}_j^* = \begin{array}{c} \text{observed} \\ \begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{v}_j \end{bmatrix} \\ \text{latent} \end{array}$$

For each round $t = 1, \ldots, T$:

1. Choose one user–item pair: $a_t = (i_t, j_t) \in [M] \times [N]$
2. Observe reward for the chosen pair: $r_{a_t,t}$



$M \times N$ possible user–item pairs

$$\boxed{r_{a_t,t} = \boldsymbol{x}_{i_t}^{*\top} \boldsymbol{\Theta}_* \boldsymbol{y}_{j_t}^* + \varepsilon_{a_t,t}}$$

bilinear in full user and item features

Objective: minimize cumulative regret relative to the best user–item pair

Prior Work and Gap

Bilinear and matrix bandits

- Exploit bilinear or matrix-valued reward structure (Jun et al., 2019; Jang et al., 2021; Lu et al., 2021; Kang et al., 2022)
- But the relevant feature information is assumed to be fully observed

Prior Work and Gap

Bilinear and matrix bandits

- Exploit bilinear or matrix-valued reward structure (Jun et al., 2019; Jang et al., 2021; Lu et al., 2021; Kang et al., 2022)
- But the relevant feature information is assumed to be fully observed

Partially observable bandits

- RoLF handles latent feature components using feature augmentation and doubly robust estimation (Kim et al., 2025)
- But it uses a linear reward model and does not preserve the two-sided bilinear interaction

Prior Work and Gap

Bilinear and matrix bandits

- Exploit bilinear or matrix-valued reward structure (Jun et al., 2019; Jang et al., 2021; Lu et al., 2021; Kang et al., 2022)
- But the relevant feature information is assumed to be fully observed

Partially observable bandits

- RoLF handles latent feature components using feature augmentation and doubly robust estimation (Kim et al., 2025)
- But it uses a linear reward model and does not preserve the two-sided bilinear interaction

Can we handle latent features while preserving the two-sided bilinear structure?

Key Idea 1: Two-Sided Orthogonal Feature Augmentation

Construction

- Collect observed features $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ and $\boldsymbol{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$.
- Augment each observed row space by an orthogonal complement

$$\widetilde{\mathbf{X}} := \begin{bmatrix} \boldsymbol{X} \\ \mathbf{B}_{\mathbf{X}^\perp}^\top \end{bmatrix} \qquad \widetilde{\mathbf{Y}} := \begin{bmatrix} \boldsymbol{Y} \\ \mathbf{B}_{\mathbf{Y}^\perp}^\top \end{bmatrix}$$

Key Idea 1: Two-Sided Orthogonal Feature Augmentation

Construction

- Collect observed features $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ and $\boldsymbol{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$.
- Augment each observed row space by an orthogonal complement

$$\widetilde{\mathbf{X}} := \begin{bmatrix} \boldsymbol{X} \\ \mathbf{B}_{\mathbf{X}^\perp}^\top \end{bmatrix} \qquad \widetilde{\mathbf{Y}} := \begin{bmatrix} \boldsymbol{Y} \\ \mathbf{B}_{\mathbf{Y}^\perp}^\top \end{bmatrix}$$

Equivalent representation

- Preserve all pairwise mean rewards after augmentation:

$$\boldsymbol{X}_*^\top \boldsymbol{\Theta}_* \boldsymbol{Y}_* = \widetilde{\mathbf{X}}^\top \boldsymbol{\Phi}_* \widetilde{\mathbf{Y}}, \qquad \boldsymbol{\Phi}_* = \begin{bmatrix} \boldsymbol{\Phi}_*^{(o,o)} & \boldsymbol{\Phi}_*^{(o,h)} \\ \boldsymbol{\Phi}_*^{(h,o)} & \boldsymbol{\Phi}_*^{(h,h)} \end{bmatrix}$$

- o: observed row-space coordinates
- h: orthogonal-complement coordinates, not original latent features

Algorithm: BiRoLF

For each round $t = 1, \ldots, T$ do:

1. Choose a candidate pair using $\hat{\mathbf{\Phi}}_{t-1}$
2. Couple the chosen action with a pseudo-action by resampling
3. Play a_t and observe $r_{a_t,t}$. Form DR pseudo-rewards for all pairs:

$$\widetilde{r}_{(i,j),t} = \widetilde{\mathbf{x}}_i^\top \check{\mathbf{\Phi}}_t \widetilde{\mathbf{y}}_j + \frac{\mathbb{1}\{(i,j) = a_t\}}{p} \left(r_{a_t,t} - \widetilde{\mathbf{x}}_{i_t}^\top \check{\mathbf{\Phi}}_t \widetilde{\mathbf{y}}_{j_t} \right)$$

4. Update the matrix-valued Lasso estimator

$$\hat{\mathbf{\Phi}}_t \leftarrow \operatorname*{argmin}_{\mathbf{\Phi}} \sum_{\tau \le t} \mathbb{1}\{\mathcal{M}_\tau\} \left\| \widetilde{\mathbf{R}}_\tau - \widetilde{\mathbf{X}}^\top \mathbf{\Phi} \widetilde{\mathbf{Y}} \right\|_{\mathrm{F}}^2 + \lambda_t \|\mathbf{\Phi}\|_{1,1}$$

- We exploit the block structure of $\mathbf{\Phi}$ to obtain an exact solver

Key Idea 2: Exact Blockwise Solver

The matrix-valued Lasso update separates into four smaller problems

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}^{(o,o)} & \mathbf{\Phi}^{(o,h)} \\ \mathbf{\Phi}^{(h,o)} & \mathbf{\Phi}^{(h,h)} \end{bmatrix}$$

Block	Variable	Solver	Main cost
(o,o)	$\mathbf{\Phi}^{(o,o)} \in \mathbb{R}^{d_x \times d_y}$	matrix Lasso	$\mathcal{O}(d_x^2 d_y + d_x d_y^2)$
(o,h)	$N - d_y$ vectors in $\mathbb{R}^{d_x}$	vector Lassos	$\mathcal{O}((N-d_y)d_x^2)$
(h,o)	$M - d_x$ vectors in $\mathbb{R}^{d_y}$	vector Lassos	$\mathcal{O}((M-d_x)d_y^2)$
(h,h)	matrix in $\mathbb{R}^{(M-d_x)\times(N-d_y)}$	soft-thresholding	$\mathcal{O}((M-d_x)(N-d_y))$

- The four subproblems solve the original update exactly
- Total cost is the sum of the four block costs, rather than the dense $\mathcal{O}(M^2N^2)$ Kronecker-linearized update

Regret Guarantee

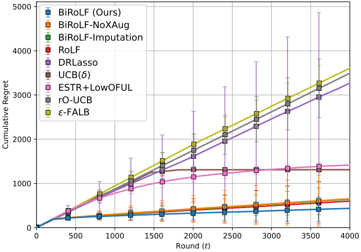
Theorem (High-probability regret bound of BiRoLF)

Under standard assumptions, the cumulative regret over horizon T is upper-bounded by

$$\textbf{\textit{Regret}}(T) \le \widetilde{\mathcal{O}}\left(\sqrt{(d_x + d_{h_x})(d_y + d_{h_y})T}\right)$$

- d_x, d_y: observed user/item feature dimensions
- d_{h_x}, d_{h_y}: active dimensions outside observed row spaces
- The bound scales with the effective bilinear dimension $(d_x + d_{h_x})(d_y + d_{h_y})$ without observing the latent coordinates

Numerical Experiments



Method	Time / trial	vs. BiRoLF
RoLF (Kim et al., 2025)	6.532 ± 0.979s	$9.6\times$ slower
BiRoLF w/o blockwise	2.688 ± 0.245s	$4.0\times$ slower
BiRoLF (Ours)	0.679 ± 0.025s	$1.0\times$

- BiRoLF maintains strong regret performance compared with both partially observable and bilinear baselines
- Exact blockwise Lasso updates substantially reduce optimization time
- Ablations support feature augmentation, DR correction, and blockwise optimization

References

Jang, K., Jun, K.-S., Yun, S.-Y., and Kang, W. (2021). Improved regret bounds of bilinear bandits using action space analysis. In International Conference on Machine Learning, pages 4744–4754. PMLR.

Jun, K.-S., Willett, R., Wright, S., and Nowak, R. (2019). Bilinear bandits with low-rank structure. In International Conference on Machine Learning, pages 3163–3172. PMLR.

Kang, Y., Hsieh, C.-J., and Lee, T. C. M. (2022). Efficient frameworks for generalized low-rank matrix bandit problems. Advances in Neural Information Processing Systems, 35:19971–19983.

Kim, W., Park, S., Iyengar, G., Zeevi, A., and Oh, M.-H. (2025). Linear bandits with partially observable features. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J., editors, Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pages 30688–30719. PMLR.

Lu, Y., Meisami, A., and Tewari, A. (2021). Low-rank generalized linear bandit problems. In International conference on artificial intelligence and statistics, pages 460–468. PMLR.