# Text-Deconfounded Factorization for Selection-Biased Ratings

Jianfeng Ma

**Abstract**

We study rating prediction under *selection bias*: users are more likely to rate items they are exposed to, and exposure is correlated with both user preference and item content. We implement and evaluate a Text-Deconfounded Factorization Model (TDFM) that combines matrix factorization with a topic representation of review text and an exposure (selection) surrogate. On a cleaned subset with $N = 2021$ unique users, $M = 100$ items, and $|D| = 6085$ observed ratings (train/test split: 4868/1217), TDFM improves test RMSE from 0.6234 (MF) to 0.5369 and MAE from 0.4066 to 0.3333. A paired bootstrap over users ($B = 2000$) yields $\Delta$RMSE $[0.0611, 0.1138]$ and $\Delta$MAE $[0.0586, 0.0891]$ (MF $-$ TDFM). Under inverse-propensity weighting with $\varepsilon = 10^{-3}$ and $w_{\max} = 50$, TDFM remains better (wRMSE 0.6546 vs. 0.7342). These results support combining text topics and exposure adjustment for selection-biased rating prediction.

## 1 Problem and setup

Let $u \in \{1, \dots, N\}$ index users and $i \in \{1, \dots, M\}$ index items. We observe a sparse set of ratings $\{y_{ui}\}_{(u,i)\in D}$ with $y_{ui} \in [1, 5]$. The observation process is *not* uniform: users tend to rate items they are exposed to, and exposure depends on confounders such as user/item popularity and content-related factors. As a result, minimizing empirical error on $D$ can misrepresent performance on the counterfactual distribution where exposure is controlled [1, 2, 5].

**Exposure surrogate construction ($z_{ui}$).** We build a binary observation/exposure matrix $A^{\text{train}} \in \{0, 1\}^{N\times M}$ where $A^{\text{train}}_{ui} = 1$ if $(u, i) \in D_{\text{train}}$ (a rating is observed in the training split) and $A^{\text{train}}_{ui} = 0$ otherwise.

We then fit a Poisson factorization model

$$A^{\text{train}}_{ui} \sim \text{Poisson}(\lambda_{ui}), \qquad \lambda_{ui} = s_u^\top t_i, \qquad s_u, t_i \in \mathbb{R}_+^{K_{\text{PF}}}.$$

using nonnegative user/item factors $(s_u, t_i)$ [3]. This ensures the exposure surrogate is fit using training interactions only (no test leakage). After training, we define the exposure surrogate as the predicted rate

$$z_{ui} \;:=\; \hat{\lambda}_{ui} \;=\; s_u^\top t_i,$$

which yields a dense nonnegative matrix $Z = \{z_{ui}\}$ used in TDFM as a proxy for selection/exposure intensity. (Table 1 reports the min/max of this learned $Z$.)

**Notation** We use $N$ users and $M$ items, $K$ topics, latent dimension $d$, Poisson-factorization rank $K_{\text{PF}}$, exposure surrogate $z_{ui}$, and topic proportions $\theta_{ui} \in \Delta^{K-1}$. In our notebook implementation:

$$N = 2021, \quad M = 100, \quad K = 10, \quad K_{\text{PF}} = 20, \quad d = 8, \quad |D| = 6085, \quad \text{vocab size } V = 1000,$$

with train/test split sizes 4868/1217 (seed 0). Text is featurized as bag-of-words with `max_features`=1000 and `stop_words`=`english`.

# 2 Models

## 2.1 Baseline: matrix factorization (MF)

The MF baseline predicts

$$\hat{y}_{ui}^{\mathrm{MF}} = 1 + 4 \cdot \sigma\left(p_u^\top q_i + b_u + b_i\right),$$

where $p_u, q_i \in \mathbb{R}^d$ are embeddings, $b_u, b_i \in \mathbb{R}$ are biases, and $\sigma(\cdot)$ is the logistic function to map predictions into $[1, 5]$. For consistency with the TDFM notation in Section 2.2, you may view $(p_u, q_i)$ as the same type of user/item embeddings as $(\mathbf{u}_u, \mathbf{v}_i)$. We use different symbols only to distinguish the baseline from the hybrid model.

## 2.2 Text-Deconfounded Factorization Model (TDFM)

TDFM augments MF with (i) a topic representation of review text and (ii) an exposure surrogate. This follows the general idea of combining topic models with collaborative filtering for recommendation [4]. Let $x_{ui} \in \mathbb{R}^V$ be the bag-of-words vector for the review associated with $(u, i)$ (when available). We infer topic proportions via an encoder

$$h_{ui} = \mathrm{ReLU}(W_1 x_{ui}), \qquad \theta_{ui} = \mathrm{softmax}(W_2 h_{ui}) \in \Delta^{K-1},$$

and reconstruct text with a decoder producing log-probabilities $\log p(x_{ui} \mid \theta_{ui})$.

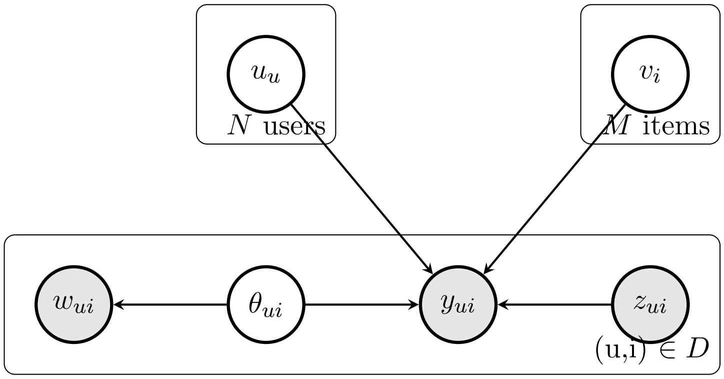

Figure 1: Graphical model for TDFM. The rating $y_{ui}$ is generated by user/item factors $(u_u, v_i)$, the exposure surrogate $z_{ui}$, and the latent topics $\theta_{ui}$ derived from review text $w_{ui}$.

For rating prediction we add two adjustment terms:

$$\hat{y}_{ui}^{\mathrm{TDFM}} = 1 + 4 \cdot \sigma\left(\mathbf{u}_u^\top \mathbf{v}_i + b_u + b_i + \beta_t(\theta_{ui}) + \beta_z(z_{ui})\right),$$

where $\mathbf{u}_u, \mathbf{v}_i \in \mathbb{R}^d$ are user/item embeddings, and $b_u, b_i$ are biases. The terms $\beta_t(\theta_{ui})$ and $\beta_z(z_{ui})$ represent the confounding functions for topics and exposure, respectively.

In the notebook implementation, these functions are parameterized as linear layers:

$$\beta_t(\theta_{ui}) = \alpha^\top \theta_{ui}, \quad \beta_z(z_{ui}) = \gamma z_{ui},$$

where $\alpha \in \mathbb{R}^K$ and $\gamma \in \mathbb{R}$ are learned weights.

**Note (poster vs. final implementation).** The poster described $\beta_z(\cdot)$ as a small non-linear MLP. In the final executed notebook used for reporting results, we use a linear exposure adjustment $\beta_z(z_{ui}) = \gamma z_{ui}$ for simplicity and stability; all reported metrics correspond to this final setting.

**Training objective.** Minimizing the squared error corresponds to maximizing the likelihood under a Gaussian assumption $y_{ui} \sim \mathcal{N}(\hat{y}_{ui}, \sigma^2)$. We optimize the joint objective:

$$\mathcal{L} = \underbrace{\frac{1}{|B|} \sum_{(u,i)\in B} (\hat{y}_{ui} - y_{ui})^2}_{\text{rating MSE}} + \lambda_{\text{recon}} \underbrace{\left( -\frac{1}{|B|} \sum_{(u,i)\in B} x_{ui}^{\top} \log \hat{p}(\cdot \mid \theta_{ui}) \right)}_{\text{text reconstruction}},$$

with $\lambda_{\text{recon}} = 0.002$.

## 3 Implementation details (from the executed notebook)

**Note (poster vs. final run).** Some hyperparameters shown in the poster reflect an earlier configuration (e.g., larger latent dimension and batch size). The values in Table 1 and all numerical results in this report are taken from the final executed notebook run. [1]

**Reproducibility note.** All results in Sections 4–6 use the final executed notebook configuration summarized in Table 1.

| Setting | Value |
|---|---|
| Random seed | 0 |
| Train/test split sizes | 4868/1217 (after dropping zeros: 6085 rows) |
| Users/items/vocab | $N = 2021$, $M = 100$, $V = 1000$ |
| Topics / latent dim | $K = 10$, $d = 8$ |
| Optimizer | Adam |
| Learning rate / weight decay | 0.01 / $10^{-4}$ |
| Batch size / epochs | 64 / 20 |
| Dropout (encoder / MF) | 0.3 / 0.2 |
| Exposure surrogate $Z = \{z_{ui}\}$ (PF rate $\hat{\lambda}_{ui}$) | shape $(2021, 100)$; min/max $8.10 \times 10^{-11}$ / 2.1944 |

Table 1: Key hyperparameters and data sizes filled from the executed notebook.

## 4 Results: rating prediction

### 4.1 Main comparison

All numbers below are on the *aligned* test set (same split, no text leakage).
The absolute improvement of TDFM over MF is $\Delta$RMSE $= 0.0864$ and $\Delta$MAE $= 0.0733$ (MF − TDFM, using the rounded values above). MF + LDA residual correction narrows the MAE gap in this split (0.3348 vs. 0.3333), but TDFM retains a clearer advantage on RMSE and remains stronger under IPW reweighting (Section 5), consistent with added benefit from the exposure adjustment.

[1] Code and data processing pipelines are available at: `https://github.com/mjf88530/Text-Deconfounded-Factorization`

[2] LDA topics fitted on train text; a Ridge model is trained to predict MF residuals from topic proportions, then added back to MF predictions.

| Model | RMSE ↓ | MAE ↓ |
|---|---|---|
| MF baseline | 0.6234 | 0.4066 |
| MF + LDA residual correction[2] | 0.6123 | 0.3348 |
| TDFM (full) | **0.5369** | **0.3333** |
| Improvement (MF − TDFM) | 0.0864 | 0.0733 |

Table 2: Main comparison on the aligned test set (same split, no text leakage). Lower is better.

### 4.2 Ablation: trained no-topic (fair)

We retrain an ablation that removes the topic/text component (no $\theta_{ui}$ term) while keeping the same split and training procedure:

$$\text{NoTopic(trained)} : \text{RMSE } 0.6465, \ \text{MAE } 0.4610.$$

Relative to this ablation, the topic/text component contributes about

$$\Delta\text{RMSE} = 0.1097, \qquad \Delta\text{MAE} = 0.1277$$

(NoTopic(trained) − Full, using rounded values).

### 4.3 Bootstrap significance (paired by user)

We run a paired bootstrap across users ($B = 2000$ resamples). For MF vs. full TDFM:

$$\Delta\text{RMSE} \in [0.0611, 0.1138], \qquad \Delta\text{MAE} \in [0.0586, 0.0891],$$

and 68.6% (RMSE) / 69.6% (MAE) of users show improvement.

For MF vs. NoTopic(trained), the deltas are negative (NoTopic worse):

$$\mathbb{E}[\Delta\text{RMSE}] = -0.0361, \ 95\%\,\text{CI} \ [-0.0556, -0.0172], \qquad \mathbb{E}[\Delta\text{MAE}] = -0.0594, \ 95\%\,\text{CI} \ [-0.0723, -0.0462].$$

## 5 Propensity weighting (IPW) for selection bias

To probe sensitivity to exposure bias, we fit a logistic propensity model on train pairs vs. sampled negatives using simple exposure features. We use inverse-propensity weighting as a standard counterfactual/selection-bias diagnostic in recommender settings [2].

**Note.** The PF-based exposure surrogate $z_{ui}$ is part of the *TDFM scoring model*, whereas the propensity $\hat{p}_{ui}$ is a separate logistic estimator used only to define IPW evaluation weights.

$$\phi(u, i) = \big(\log(1 + \deg(u)), \ \log(1 + \deg(i))\big).$$

We compute user/item degrees $\deg(u), \deg(i)$ using **training interactions only** (i.e., from $D_{\text{train}}$) to avoid any test-set leakage.

We then compute test weights:

$$w_{ui} = \min\left(\frac{1}{\max(\hat{p}_{ui}, \, \varepsilon)}, \, w_{\max}\right), \quad \varepsilon = 10^{-3}, \ w_{\max} = 50.$$

Weight statistics on the test set are:

$$\min w = 1.0196, \ \text{median } 2.0643, \ \text{mean } 2.9680, \ \max 34.5164.$$

Using these weights, we report IPW-weighted errors:

- MF: wRMSE 0.7342, wMAE 0.4821.
- TDFM (full): wRMSE 0.6546, wMAE 0.4014.

TDFM remains better than MF under this reweighting, with wRMSE improvement 0.0796 and wMAE improvement 0.0808 (MF − TDFM, using the rounded values above).

## 6 Implicit ranking robustness sweep (negative sampling)

**Definitions.** **TDFM_obs** ranks items using the full observed-rating score from TDFM (the same score used for rating RMSE/MAE).

As a supplementary evaluation, we compute Recall@K and NDCG@K under negative sampling with $N_{\text{neg}} \in \{20, 50, 100\}$ and $K \in \{5, 10, 20\}$. The evaluation uses 875 users with at least one positive in the test set and scores all $M = 100$ items.

For $N_{\text{neg}} = 20$ and $K = 10$:

MF: Recall@10 0.6225, NDCG@10 0.4009; **TDFM_obs**: Recall@10 0.6679, NDCG@10 0.3957.

For $N_{\text{neg}} = 50$ and $K = 10$:

MF: Recall@10 0.4058, NDCG@10 0.2764; **TDFM_obs**: Recall@10 0.4127, NDCG@10 0.2520.

Across sweeps, **TDFM_obs** is generally close to MF on NDCG while improving Recall in some regimes. (These ranking results are sensitive to the implicit-feedback construction and negative-sampling protocol, so we treat them as diagnostic rather than the primary success criterion.)

## 7 Conclusion

The implemented TDFM improves rating prediction compared to MF on this selection-biased dataset, and the retrained no-topic ablation indicates that incorporating text topics is important for predictive accuracy (see Appendix B for a qualitative inspection of learned topics). IPW-weighted metrics further suggest that the improvement persists when reweighting for estimated exposure propensities. Future work to better match the intended causal story includes: (i) using a richer exposure model (e.g., adding content/popularity covariates or a small MLP for the exposure adjustment), (ii) validating robustness across alternative splits and larger item universes, and (iii) reporting additional diagnostics such as calibration plots and sensitivity to the IPW clipping threshold.

# Appendix:

## A Training Algorithm (TDFM)

**Algorithm 1** TDFM Training (linear surrogates; final setting)

```
Input: observed ratings $Y$, exposure indicator $A$, review text $W$ (bag-of-words)
Pre-compute: exposure surrogate $z_{ui} = \phi(u,i)$ (e.g., PF/log-degrees)
Pre-train: topic encoder on text to obtain topic proportions $\theta_{ui} = \text{Encoder}(w_{ui})$
Initialize: user/item factors $U, V$, topic weights $\alpha$, exposure weight $\gamma$
for epoch = 1 to $E$ do
    for batch $(u,i) \in \mathcal{O}$ do
        Compute score $s_{ui} = u_u^\top v_i + \alpha^\top \theta_{ui} + \gamma z_{ui}$
        Predict $\hat{y}_{ui} = 1 + 4\sigma(s_{ui})$
        $\mathcal{L} \leftarrow (y_{ui} - \hat{y}_{ui})^2 + \lambda\Big(\|u_u\|^2 + \|v_i\|^2 + \|\alpha\|^2 + \gamma^2\Big)$
        Backpropagate and update $U, V, \alpha, \gamma$ (Adam)
    end for
end for
```

## B Topic inspection (qualitative)

**Note (seeds).** This topic inspection uses the best checkpoint from a 5-seed sweep (best seed = 2), while all predictive metrics reported in Sections 4–6 (e.g., Table 2) use the fixed seed-0 run for a consistent split and reproducibility.

**Top words by topic.** We list representative top words for each learned topic (computed from the decoder/topic-word distribution in the trained model). These topics provide a qualitative check that the text component captures coherent semantic structure.

| Topic | Prevalence | Top words (by decoder weights) |
|---|---|---|
| 9 | 0.8607 | glaze, excellent, favorite, cake, fresh, turkey, hand, loved, great, potatoes |
| 7 | 0.0634 | coffee, good, pie, key, simple, lime, zucchini, soups, follow, cupcakes |
| 0 | 0.0148 | completely, thighs, celery, care, did, breasts, breast, don, really, soy |
| 8 | 0.0134 | completely, celery, thighs, don, soy, powder, follow, breasts, breast, cheesecake |
| 2 | 0.0121 | follow, cups, cheesecake, served, try, powder, don, soy, recipes, completely |
| 5 | 0.0093 | did, time, make, recipe, definitely, like, half, little, fat, end |
| 1 | 0.0077 | try, cups, dish, cheesecake, fudge, follow, leave, served, recipe, mushroom |
| 3 | 0.0073 | cups, try, dish, fudge, follow, cheesecake, leave, recipe, mushroom, served |
| 4 | 0.0056 | dish, cheesecake, try, fudge, better, leave, served, mushroom, follow, cups |
| 6 | 0.0056 | try, cheesecake, better, dish, fudge, leave, served, follow, mushroom, felt |

Table 3: Topic inspection from the best checkpoint (seed 2). Prevalence is the average topic proportion on sampled BOW rows.

**Interpretation.** Topic 9 dominates the corpus (prevalence 0.8607) and appears to capture general recipe/food-review language (e.g., *cake*, *fresh*, *great*), while smaller topics pick out more specific concepts (e.g., Topic 7: coffee/pie/lime; Topic 0/8: soy/celery/thighs).

# C References

[1] Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei. Causal Inference for Recommender Systems. In *Proceedings of the Fourteenth ACM Conference on Recommender Systems (RecSys '20)*, 2020.

[2] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[3] Prem Gopalan, Jake M. Hofman, and David M. Blei. Scalable Recommendation with Hierarchical Poisson Factorization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

[4] Chong Wang and David M. Blei. Collaborative Topic Modeling for Recommending Scientific Articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[5] Harald Steck. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys)*, 2010.