

---

# COLLABORATIVE FILTERING ALGORITHM IMPLEMENTATION

Gradient Descent with Probabilistic Assumptions  
V.S.

Alternating Least Squares

GROUP 7

Lu, Ran (Presenter); Shang, Haoyu; Wang, Rui  
Zhang, Qin; Zhang, Tianya

---

---

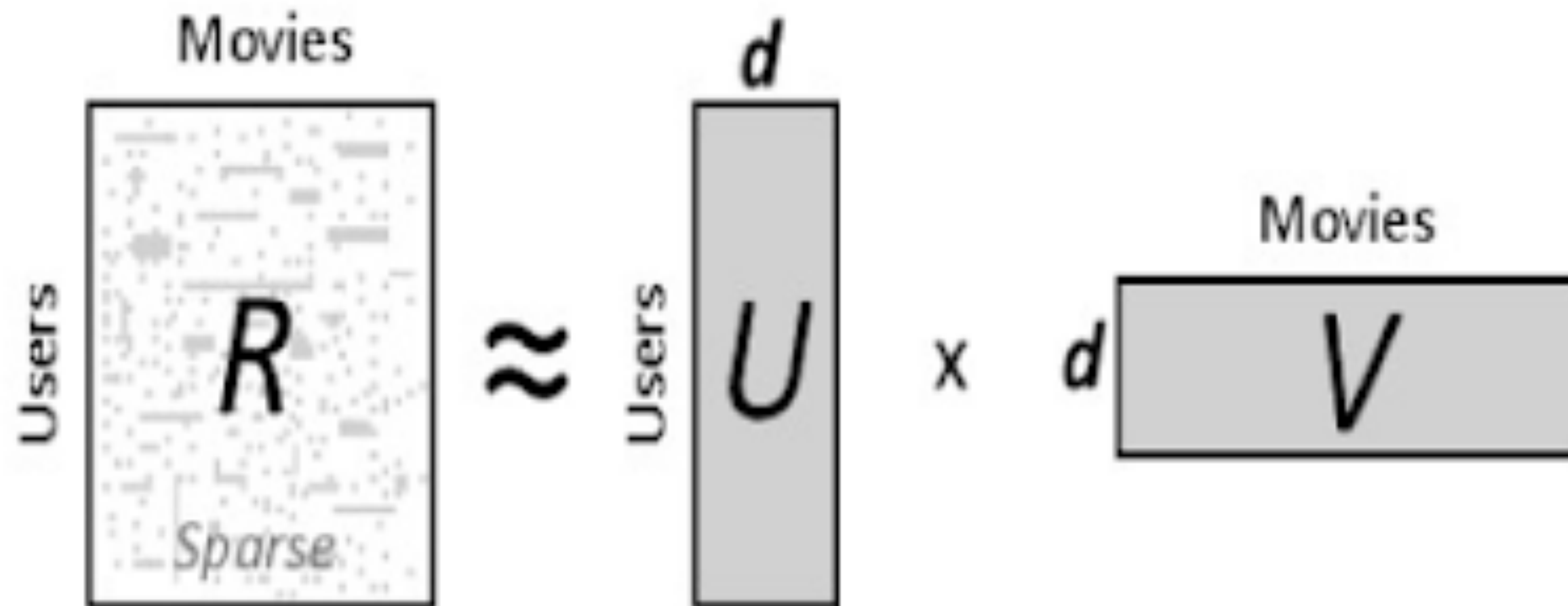
# COLLABORATIVE FILTERING

- **FILTERING:** Making automatic predictions
- **COLLABORATING:** Collecting preference information from many users

---

# MATRIX FACTORIZATION TECHNIQUES

$$R_{ij} = Q_j^T P_i$$



---

# OBJECTIVE

## ➤ **Gradient Descent with Probabilistic Assumption**

- **Probabilistic Matrix Factorization**

## ➤ **Alternative Least Square**

## ➤ **Post-Processing:**

- **SVD with KNN**
-

---

# PROBABILISTIC MATRIX FACTORIZATION

We have  $M$  movies and  $N$  users, and denote  $Q$  and  $P$  as latent movie and user feature matrices. Adopt a probabilistic linear model with Gaussian observation noise, we define the conditional distribution over the observed ratings:

$$p(R|P, Q, \sigma^2) = \prod_{i=1}^N \prod_{j=1}^M [\mathcal{N}(R_{ij}|Q_i^T P_j, \sigma^2)]^{I_{ij}}$$

with

$$p(P|\sigma_P) = \prod_{j=1}^M \mathcal{N}(P_j|0, \sigma_P^2 \mathbf{I})$$

$$p(Q|\sigma_Q) = \prod_{i=1}^N \mathcal{N}(Q_i|0, \sigma_Q^2 \mathbf{I})$$

$I_{ij}$  is the indicator function that is equal to 1 if user  $i$  rated movie  $j$  and 0 otherwise

---

---

# PROBABILISTIC MATRIX FACTORIZATION

Maximizing the log-posterior over movie and user features with hyperparameters is equivalent to minimizing the sum of squared errors objective function with quadratic regularization terms:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - Q_i^T P_j)^2 + \frac{\sigma^2}{2\sigma_Q^2} \sum_{i=1}^N \|Q_i\|_F^2 + \frac{\sigma^2}{2\sigma_P^2} \sum_{j=1}^M \|P_j\|_F^2$$

---

# GRADIENT DESCENT WITH PROBABILISTIC ASSUMPTION

We are going to update each iteration with the following:

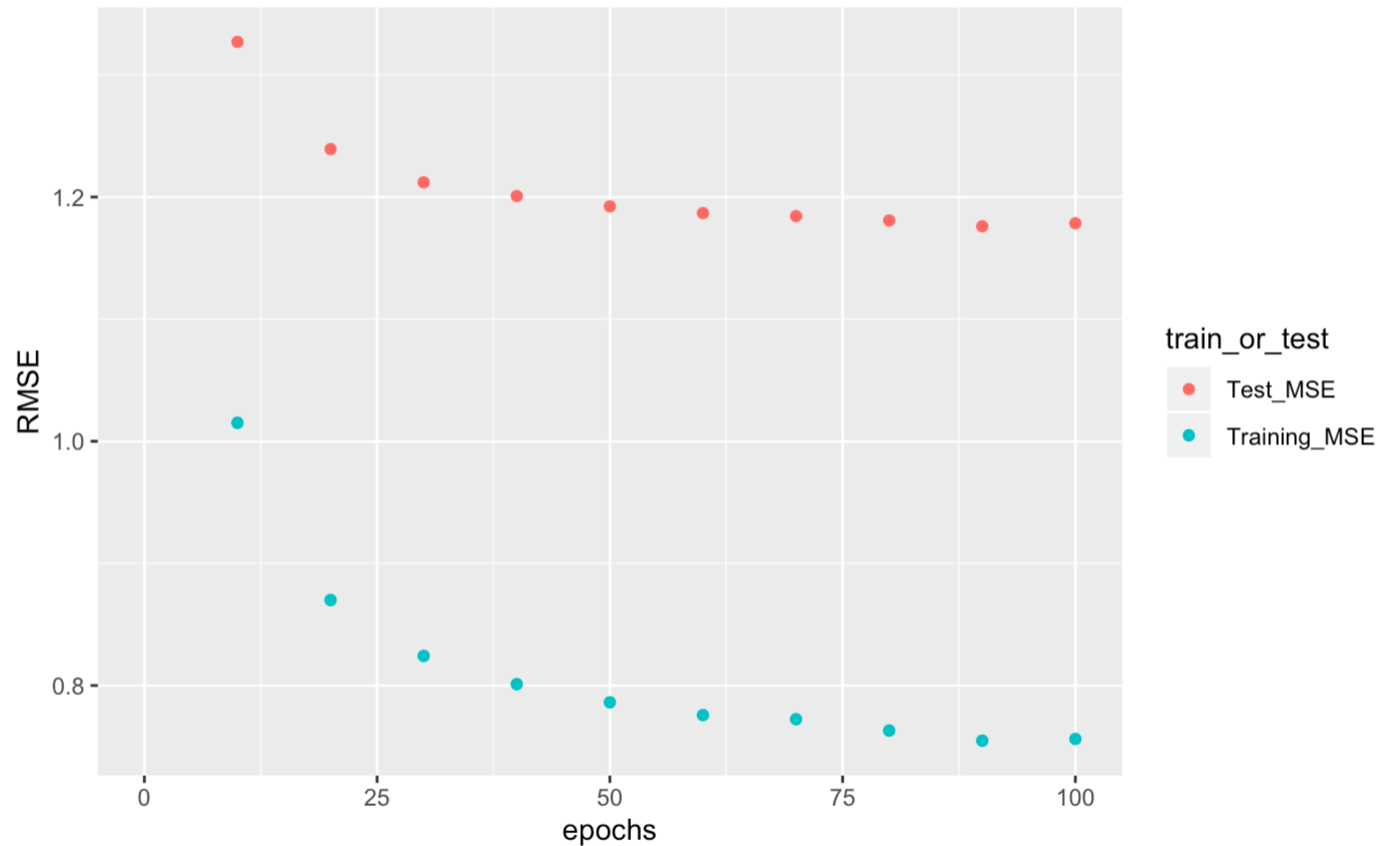
$$P_j^{(t+1)} = P_j^{(t)} + \text{lrate} \sum_{i=1}^N I_{ij}(R_{ij} - Q_i^{T(t)} P_j^{(t)}) Q_i^{T(t)} - \frac{\sigma^2}{\sigma_P^2} P_j^{(t)}$$

$$Q_i^{(t+1)} = Q_i^{(t)} + \text{lrate} \sum_{j=1}^M I_{ij}(R_{ij} - Q_i^{T(t)} P_j^{(t)}) P_j^{(t)} - \frac{\sigma^2}{\sigma_Q^2} Q_i^{(t)}$$

# PROBABILISTIC MATRIX FACTORIZATION

## ➤ Tune parameters through cross-validation

- **$f = 5$**
- **$\sigma_p = 0.5$**
- **$\sigma_q = 0.5$**
- **$\text{lrate} = 0.01$**





---

# ALTERNATIVE LEAST SQUARE

ALS with Tikhonov regulation (penalizes large parameters)

$$\min_{Q,P} f(Q,P) = \sum_{i,j \in K} (R_{ij} - Q_i^T P_j)^2 + \lambda \left( \sum_i n_{Q_i} \|Q_i\|^2 + \sum_j n_{P_j} \|P_j\|^2 \right)$$

During each iteration, fix Q to solve P, and fix P to solve Q by:

$$P_i = (Q_{I_i} Q_{I_i}^T + \lambda n_{P_i} E)^{-1} Q_{I_i} R^T(i, I_i)$$
$$Q_j = (P_{I_j} P_{I_j}^T + \lambda n_{Q_j} E)^{-1} P_{I_j} R(I_j, j)$$

$n_{P_i}$  and  $n_{Q_j}$  are the numbers of ratings of user i and movie j respectively

E is the  $n_f \times n_f$  identity matrix

$Q_{I_i}$  is the sub-matrix of M where columns  $j \in I_i$  are selected

$R(i, I_i)$  is the row vector where columns  $j \in I_i$  of the i-th row of R is taken

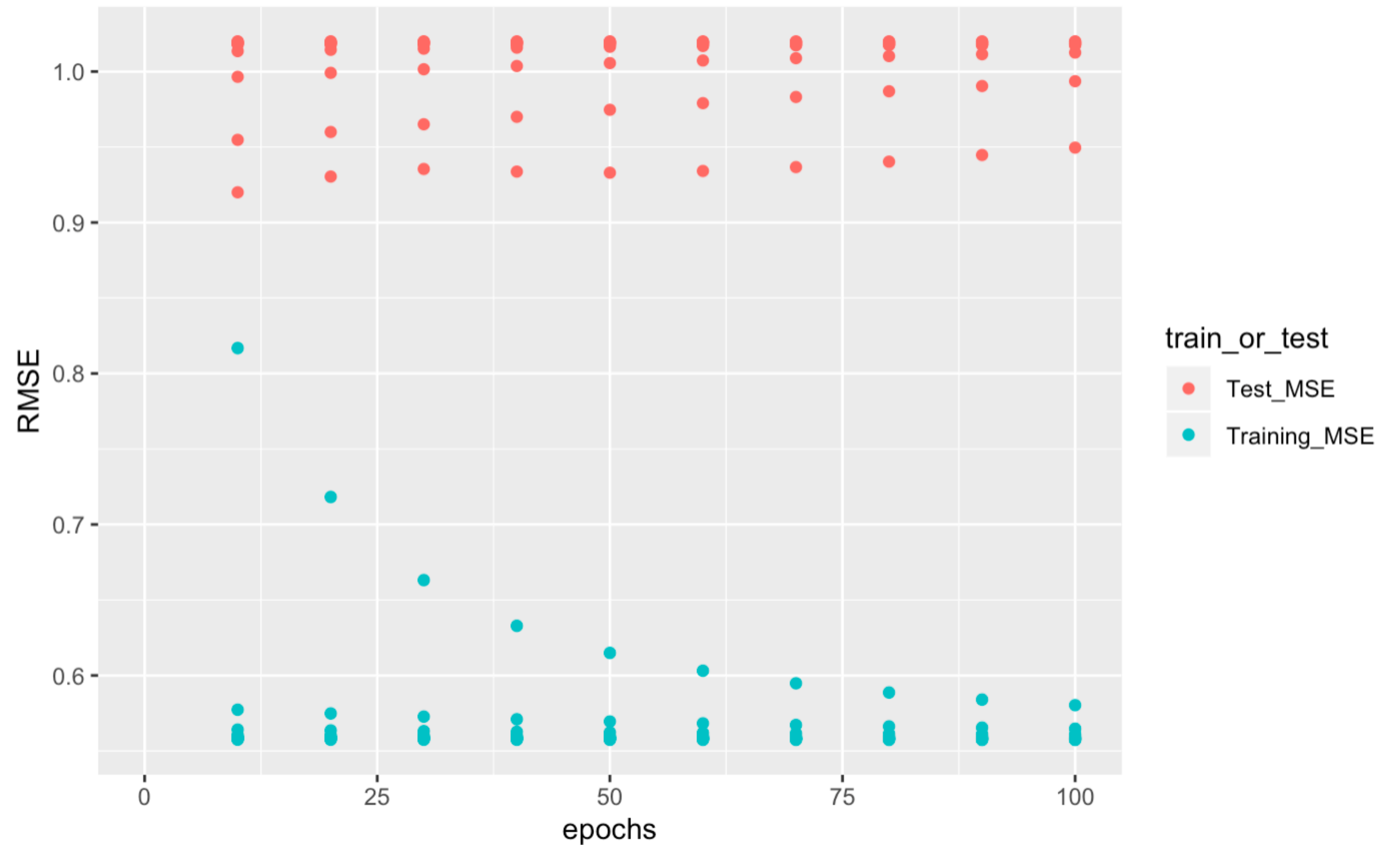
$R(I_j, j)$  is the column vector where rows  $i \in I_i$  of the j-th column of R is taken

---

# ALTERNATIVE LEAST SQUARE

## ➤ Tune parameters through cross-validation

- **$f = 5$**
- **$\lambda = .1$**



---

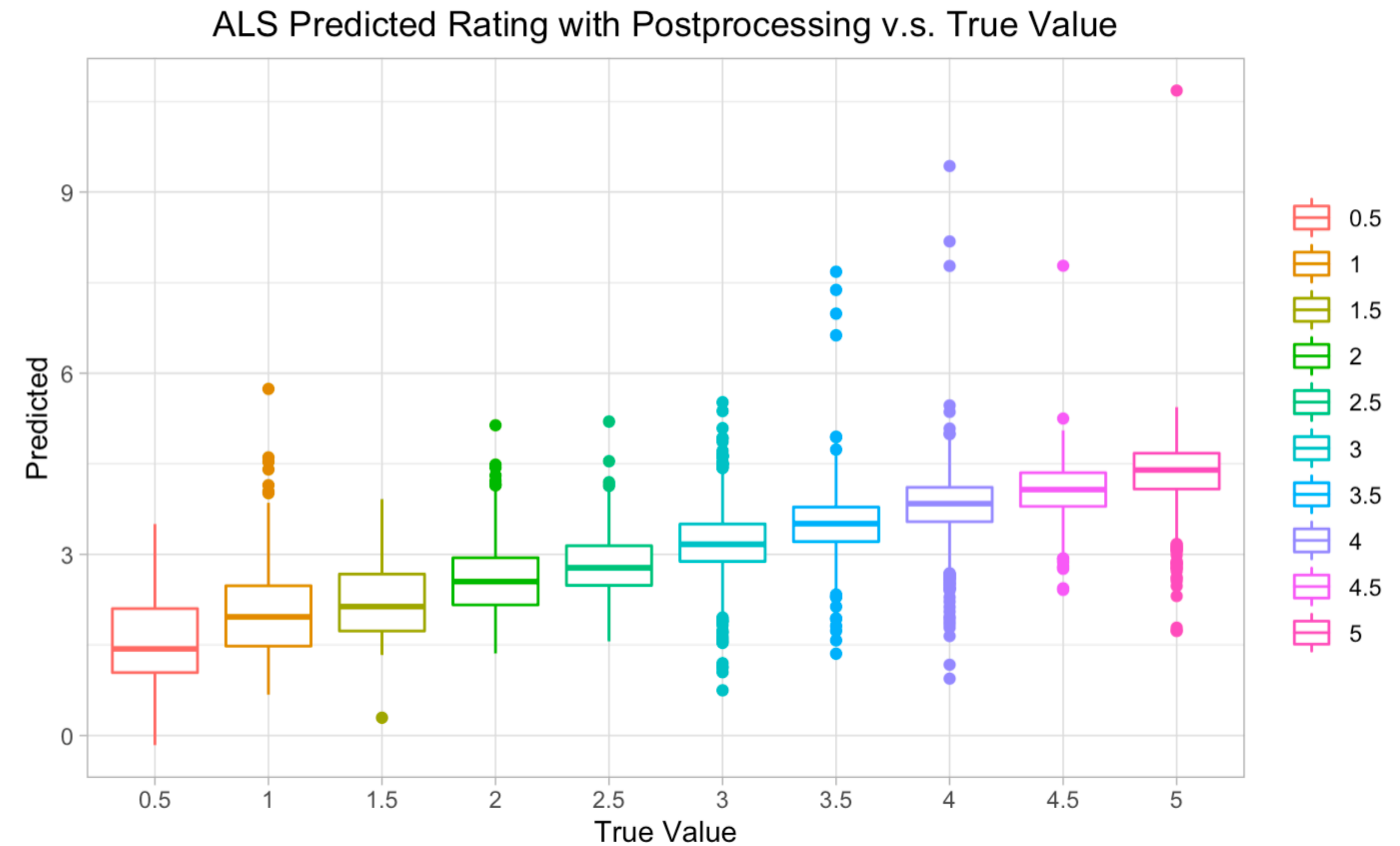
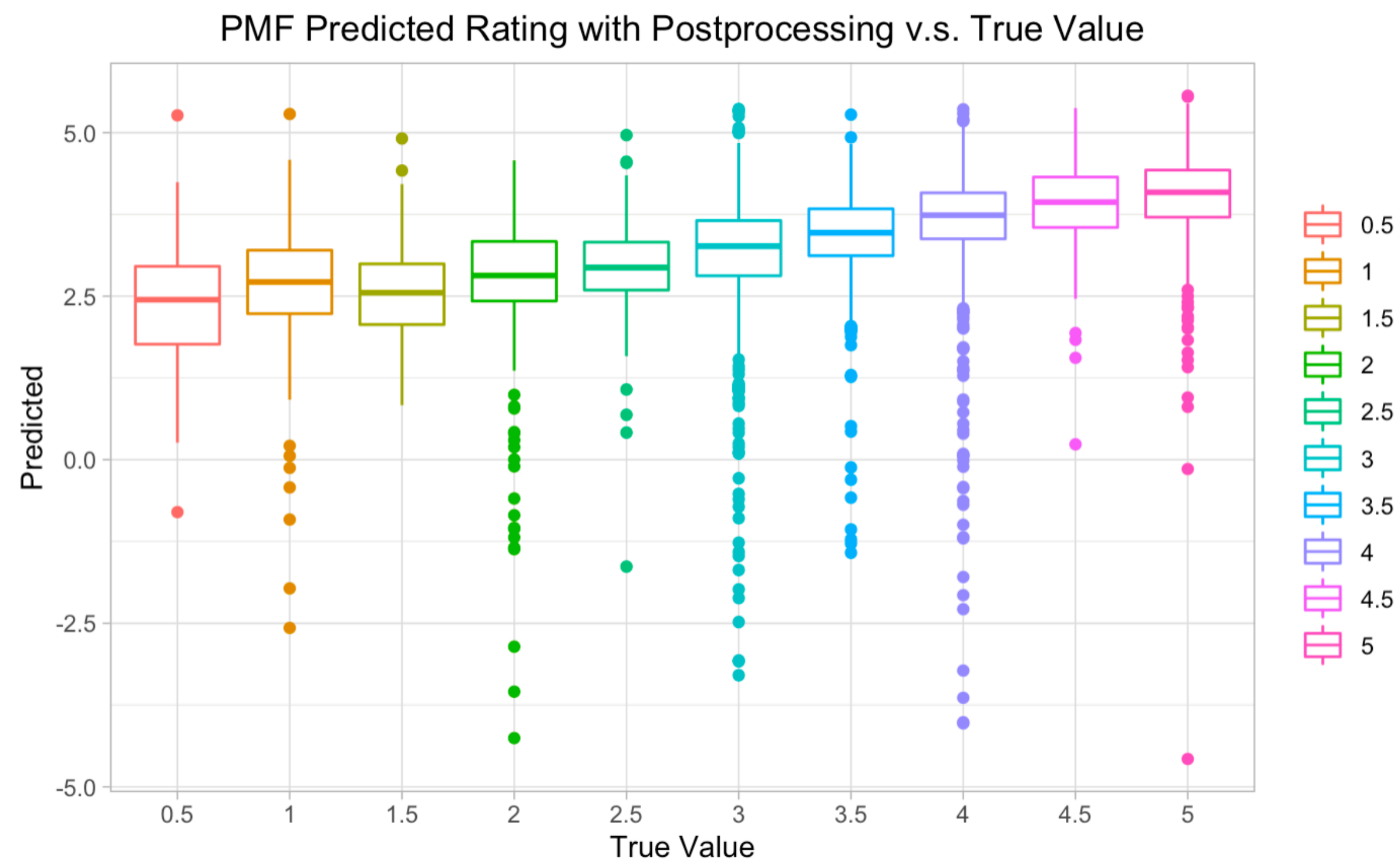
# POSTPROCESSING

## ➤ SVD with KNN

Define similarity between movies  $j$  and  $j_2$  as cosine similarity between vectors  $v_j$  and  $v_{j_2}$  as

$$s(v_j, v_{j_2}) = \frac{v_j^T v_{j_2}}{\|v_j\| \|v_{j_2}\|}$$

# POSTPROCESSING RESULTS



---

# RESULTS

	Gradient Descent with Probabilistic Assumptions	Alternative Least Squares
RMSE before postprocessing	Train 0.756 Test 1.179	Train 0.557 Test 1.017
RMSE after postprocessing	Test 0.985	Test 0.664

---

---

**THANK YOU**

---