

Tutorial: Matrix Factorization for Recommendation Systems

Cheat Sheet and Practice Problems

ES335 - Machine Learning
IIT Gandhinagar

July 23, 2025

1 Summary from Slides

1.1 Problem Setup

Collaborative Filtering Problem:

- Users rate items (movies, products, songs)
- Rating matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ where $\mathbf{A}_{ij}$ is user i 's rating for item j
- Most entries are missing (sparse matrix)
- Goal: Predict missing ratings

Real-world Scale:

- Netflix: 200M+ users, 15K+ movies
- Amazon: 300M+ users, millions of products
- Typical user rates ~1% of available items

1.2 Matrix Factorization Approach

Key Insight: User preferences and item characteristics can be captured by latent features.

Mathematical Formulation:

$$\mathbf{A} \approx \mathbf{W}\mathbf{H}$$

Where:

- $\mathbf{A} \in \mathbb{R}^{n \times m}$: Rating matrix
- $\mathbf{W} \in \mathbb{R}^{n \times r}$: User feature matrix (user i preferences)
- $\mathbf{H} \in \mathbb{R}^{r \times m}$: Item feature matrix (item j characteristics)
- r : Number of latent features (typically $r \ll \min(n, m)$)

Prediction: $\hat{\mathbf{A}}_{ij} = \mathbf{w}_i^T \mathbf{h}_j = \sum_{k=1}^r \mathbf{W}_{ik} \mathbf{H}_{kj}$

1.3 Learning Algorithms

Alternating Least Squares (ALS):

1. Fix $\mathbf{H}$, solve for $\mathbf{W}$: $\min_{\mathbf{W}} \|\mathbf{A}_{\Omega} - \mathbf{W}\mathbf{H}_{\Omega}\|_F^2$
2. Fix $\mathbf{W}$, solve for $\mathbf{H}$: $\min_{\mathbf{H}} \|\mathbf{A}_{\Omega} - \mathbf{W}\mathbf{H}_{\Omega}\|_F^2$
3. Repeat until convergence

Stochastic Gradient Descent (SGD): For each observed rating $(i, j) \in \Omega$:

1. Predict: $\hat{a}_{ij} = \mathbf{w}_i^T \mathbf{h}_j$
2. Compute error: $e_{ij} = a_{ij} - \hat{a}_{ij}$
3. Update:

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \alpha \cdot e_{ij} \cdot \mathbf{h}_j \quad (1)$$

$$\mathbf{h}_j \leftarrow \mathbf{h}_j + \alpha \cdot e_{ij} \cdot \mathbf{w}_i \quad (2)$$

Loss Function: $L = \sum_{(i,j) \in \Omega} (a_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2$

2 Practice Problems

Basic Matrix Factorization Setup

Consider a movie rating system with 4 users and 5 movies. The rating matrix is:

$$\mathbf{A} = \begin{bmatrix} 5 & 4 & ? & 3 & ? \\ ? & 5 & 1 & 4 & ? \\ 4 & ? & 1 & 5 & 2 \\ 3 & 4 & ? & ? & 1 \end{bmatrix}$$

- a) How many ratings are observed? What's the sparsity percentage? b) If we use $r = 2$ latent features, what are the dimensions of $\mathbf{W}$ and $\mathbf{H}$? c) How many parameters do we need to learn in total?

Prediction Calculation

Given user feature vector $\mathbf{w}_1 = [0.8, 0.2]$ and movie feature vector $\mathbf{h}_1 = [0.9, 0.3]^T$:

- a) Calculate the predicted rating $\hat{a}_{11}$ b) If the actual rating is $a_{11} = 4$, what is the prediction error? c) Interpret what the feature values might represent (e.g., genres, preferences)

SGD Update Step

For the previous problem, assume learning rate $\alpha = 0.01$:

- a) Calculate the SGD updates for $\mathbf{w}_1$ and $\mathbf{h}_1$ b) What are the new feature vectors after one SGD step? c) How does the prediction change after the update?

ALS Formulation

For the ALS algorithm, when fixing $\mathbf{H}$ and solving for user i 's features:

- a) Write the optimization problem for $\mathbf{w}_i$ b) What is the closed-form solution? c) Why is this approach called "alternating" least squares?

Latent Feature Interpretation

Consider a movie recommendation system where we discover these latent features: - Feature 1: Action movies have high values, Romance movies have low values - Feature 2: Bollywood movies have high values, Hollywood movies have low values

a) What would a user vector $[0.9, 0.1]$ suggest about their preferences? b) How would you classify a movie with features $[0.2, 0.8]$? c) Predict who would rate this movie higher: User A $[0.8, 0.3]$ or User B $[0.1, 0.9]$?

Regularization in Matrix Factorization

The regularized loss function is:

$$L = \sum_{(i,j) \in \Omega} (a_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda (\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2)$$

a) Why is regularization necessary in matrix factorization? b) How does the parameter λ affect the solution? c) Write the SGD updates with regularization

Cold Start Problem

a) What is the "cold start" problem in recommendation systems? b) How can we handle new users who haven't rated any items? c) How can we handle new items that haven't been rated by any users? d) Suggest at least two approaches for each scenario

Evaluation Metrics

For recommendation systems, we need appropriate evaluation metrics:

a) Why is standard RMSE not always the best metric? b) What is the difference between rating prediction and ranking metrics? c) Explain precision@k and recall@k in the recommendation context d) When would you use each type of metric?

Scalability Considerations

a) What is the computational complexity of one ALS iteration? b) What is the computational complexity of one SGD epoch? c) Why is SGD preferred for very large datasets? d) How can we parallelize ALS? What about SGD?

Implicit Feedback

Many recommendation systems have implicit feedback (clicks, views, purchases) rather than explicit ratings:

a) How does the problem formulation change for implicit feedback? b) What does the confidence matrix represent? c) How do the ALS updates change for implicit feedback? d) Give examples of implicit vs explicit feedback in real systems

Advanced Extensions

a) How would you incorporate user and item biases into matrix factorization? b) What is non-negative matrix factorization (NMF) and when is it useful? c) How can temporal dynamics be incorporated (ratings change over time)? d) Describe how to handle multiple types of feedback (ratings, clicks, purchases)

Real-world Implementation

Design a movie recommendation system for a streaming service:

- a) What data would you collect from users?
- b) How would you handle the cold start problem for new users?
- c) How would you evaluate your system before deploying?
- d) What are the key engineering challenges at scale?
- e) How would you ensure diversity in recommendations?