Explanation of the Recommendation Algorithm Incorporating User Priorities and Company Scores

Contents

1 Introduction **1**

2 Problem Description **2**

3 Matrix Factorization **2**

4 Objective Function and Regularization **2**
- 4.1 Reconstruction Error . . . 2
- 4.2 Regularization . . . 3

5 Gradient Descent Updates **3**
- 5.1 Update Rules . . . 3

6 Handling Known Columns in U and V **3**
- 6.1 Mask Matrices . . . 4
- 6.2 Modified Update Rules . . . 4

7 Incorporating User Priorities and Company Scores **4**
- 7.1 User Priorities . . . 4
- 7.2 Company Scores . . . 4
- 7.3 Impact on Alignment Scores . . . 4

8 Computing Final Alignment Scores **5**
- 8.1 Contribution of Known Features . . . 5
- 8.2 Interpretation . . . 5

9 Conclusion **5**

1 Introduction

This document provides a comprehensive explanation of a recommendation algorithm based on matrix factorization using gradient descent. It details how user priorities and company scores are integrated into the model to compute final alignment scores between users and items (e.g., movies, products). The explanation includes all relevant equations and a section-by-section breakdown of the algorithm's components.

2 Problem Description

We aim to predict missing values in a sparse rating matrix M, where:

- **Users** are represented by rows.
- **Items** (e.g., movies) are represented by columns.
- **Ratings** $M_{u,i}$ are the observed preferences of user u for item i.

The goal is to build a personalized recommendation system that can predict unknown ratings and recommend items that align closely with user preferences.

3 Matrix Factorization

Matrix factorization assumes that the rating matrix M can be approximated by the product of two lower-dimensional matrices:

- $U \in \mathbb{R}^{n_u \times r}$: User feature matrix.
- $V \in \mathbb{R}^{n_i \times r}$: Item feature matrix.
- r: The number of latent factors (rank).

The approximation is given by:

$$M_{u,i} \approx U_u^\top V_i = \sum_{k=1}^{r} U_{u,k} V_{i,k}$$

Where:

- U_u is the u-th row of U (user u's latent feature vector).
- V_i is the i-th row of V (item i's latent feature vector).

4 Objective Function and Regularization

4.1 Reconstruction Error

The objective is to minimize the squared reconstruction error over all observed ratings:

$$E(U, V) = \sum_{(u,i) \in M} \left(M_{u,i} - U_u^\top V_i \right)^2$$

4.2 Regularization

To prevent overfitting and encourage generalization, a regularization term is added:

$$E(U,V) = \sum_{(u,i)\in M} \left(M_{u,i} - U_u^\top V_i\right)^2 + \lambda \left(\sum_{u=1}^{n_u} \|U_u\|^2 + \sum_{i=1}^{n_i} \|V_i\|^2 \right)$$

Where:

- λ: Regularization parameter.
- $\|U_u\|^2 = \sum_{k=1}^{r} U_{u,k}^2$
- $\|V_i\|^2 = \sum_{k=1}^{r} V_{i,k}^2$

5 Gradient Descent Updates

To minimize $E(U,V)$, we use gradient descent to update U and V.

5.1 Update Rules

For each user u and factor k:

$$U_{u,k} \leftarrow U_{u,k} + \mu \left(2 \sum_{i\in\mathcal{I}_u} e_{u,i} V_{i,k} - 2\lambda U_{u,k} \right)$$

For each item i and factor k:

$$V_{i,k} \leftarrow V_{i,k} + \mu \left(2 \sum_{u\in\mathcal{U}_i} e_{u,i} U_{u,k} - 2\lambda V_{i,k} \right)$$

Where:

- μ: Learning rate.
- $e_{u,i} = M_{u,i} - U_u^\top V_i$: Prediction error for user u and item i.
- $\mathcal{I}_u$: Set of items rated by user u.
- $\mathcal{U}_i$: Set of users who have rated item i.

6 Handling Known Columns in U and V

In some cases, certain columns (features) of U and V are known a priori and should remain fixed during optimization. For instance, these known features might represent:

- **User Priorities**: Predefined preferences or characteristics of users.
- **Company Scores**: Pre-assigned scores or attributes of items determined by the company.

6.1 Mask Matrices

Define binary mask matrices K_U and K_V:

- $K_U \in \{0,1\}^{n_u \times r}$: Indicates known (1) and unknown (0) features in U.
- $K_V \in \{0,1\}^{n_i \times r}$: Indicates known (1) and unknown (0) features in V.

6.2 Modified Update Rules

Update only the unknown (masked) elements.

For $U_{u,k}$:

$$\text{If } K_{U_{u,k}} = 0, \text{ then } U_{u,k} \leftarrow U_{u,k} + \mu \left(2 \sum_{i \in \mathcal{I}_u} e_{u,i} V_{i,k} - 2\lambda U_{u,k} \right)$$

For $V_{i,k}$:

$$\text{If } K_{V_{i,k}} = 0, \text{ then } V_{i,k} \leftarrow V_{i,k} + \mu \left(2 \sum_{u \in \mathcal{U}_i} e_{u,i} U_{u,k} - 2\lambda V_{i,k} \right)$$

Known elements where $K = 1$ remain unchanged during updates.

7 Incorporating User Priorities and Company Scores

7.1 User Priorities

- Represented as known columns in U.
- Capture predefined preferences, such as favored genres, product categories, or other attributes.
- These priorities are encoded into specific factors (columns) of U and are fixed during optimization.

7.2 Company Scores

- Represented as known columns in V.
- Reflect company-assigned scores or attributes for items, such as quality ratings, promotional priorities, or strategic importance.
- These scores are encoded into specific factors (columns) of V and are also fixed during optimization.

7.3 Impact on Alignment Scores

By incorporating user priorities and company scores into U and V, the model ensures that:

- Recommendations align with both user preferences and company objectives.
- Fixed features influence the predicted ratings through the dot product $U_u^\top V_i$.

8 Computing Final Alignment Scores

The final predicted rating (alignment score) for user u and item i is computed as:

$$\hat{M}_{u,i} = U_u^\top V_i = \sum_{k=1}^{r} U_{u,k} V_{i,k}$$

8.1 Contribution of Known Features

- **User Priorities** (U) and **Company Scores** (V) contribute directly to $\hat{M}_{u,i}$ via their respective known columns.
- The dot product naturally combines these fixed features with the learned features to produce a final score.

8.2 Interpretation

- A higher $\hat{M}_{u,i}$ indicates a stronger alignment between user u's preferences and item i's attributes, considering both personal and company-influenced factors.
- The model balances user satisfaction with company goals.

9 Conclusion

By using known user priorities and company scores into the matrix factorization framework, we enhance the recommendation system to account for both individual preferences and organizational objectives. The algorithm updates only the unknown features during gradient descent, ensuring that the known features remain intact and influence the final alignment scores. This approach results in personalized recommendations that are aligned with both user interests and company strategies.