Lec 05-2. Item to Item Collaborative Filtering

Contents

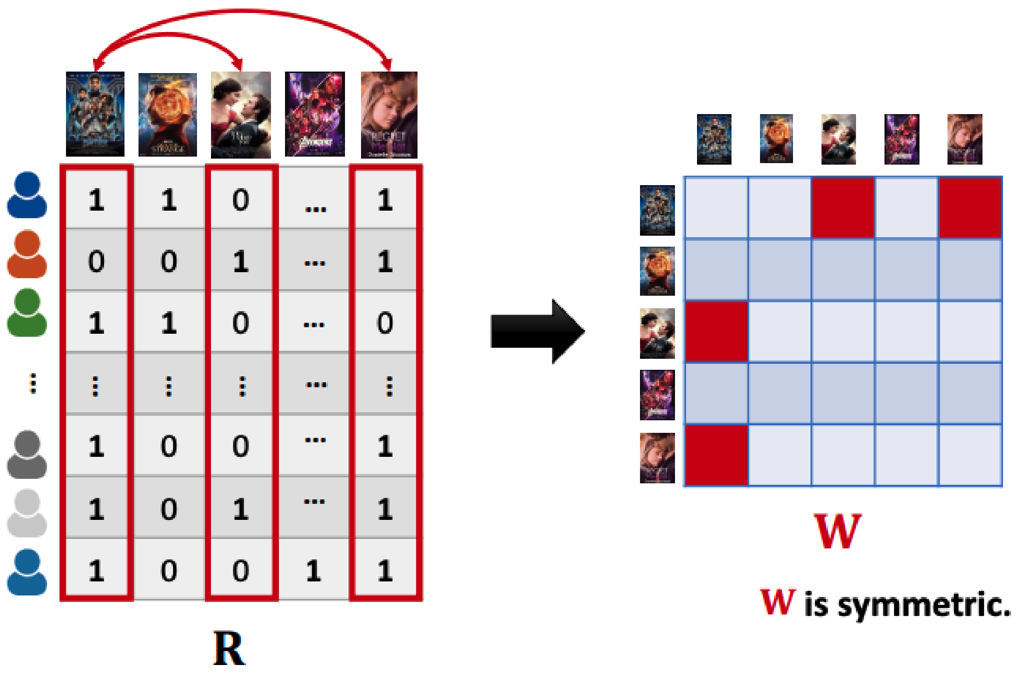
1. Sparse Linear Method (SLIM)
2. Factored Item Similarity Models (FISM)
3. Embarrassingly Shallow Autoencoders($EASE^R$)

---

1. Sparse Linear Method (SLIM)

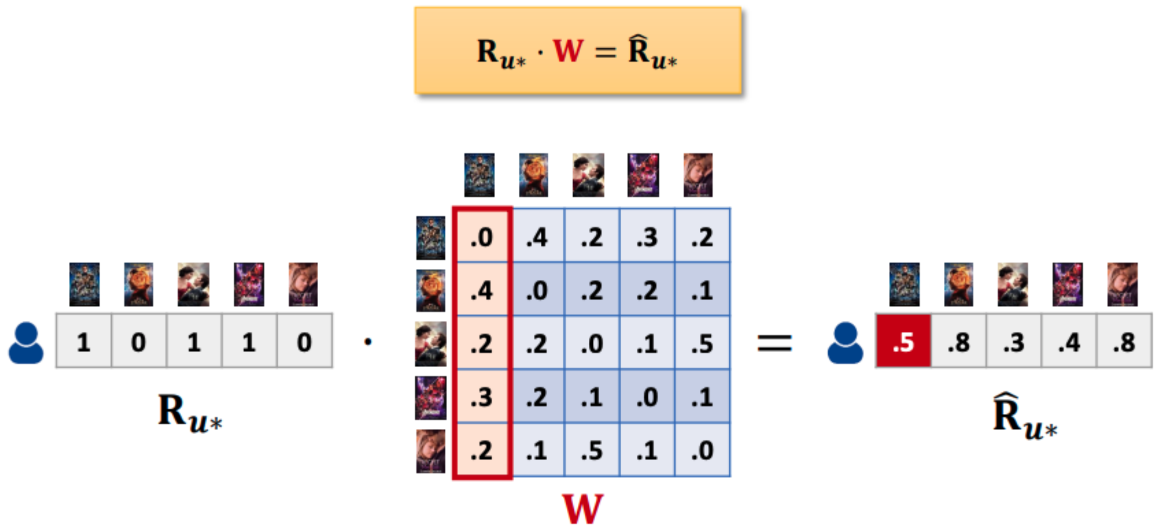
Basic Idea of Item-based CF

- To identify **a set of similar items** for a target item, the item-item similarities are calcuated from the rating matrix R.



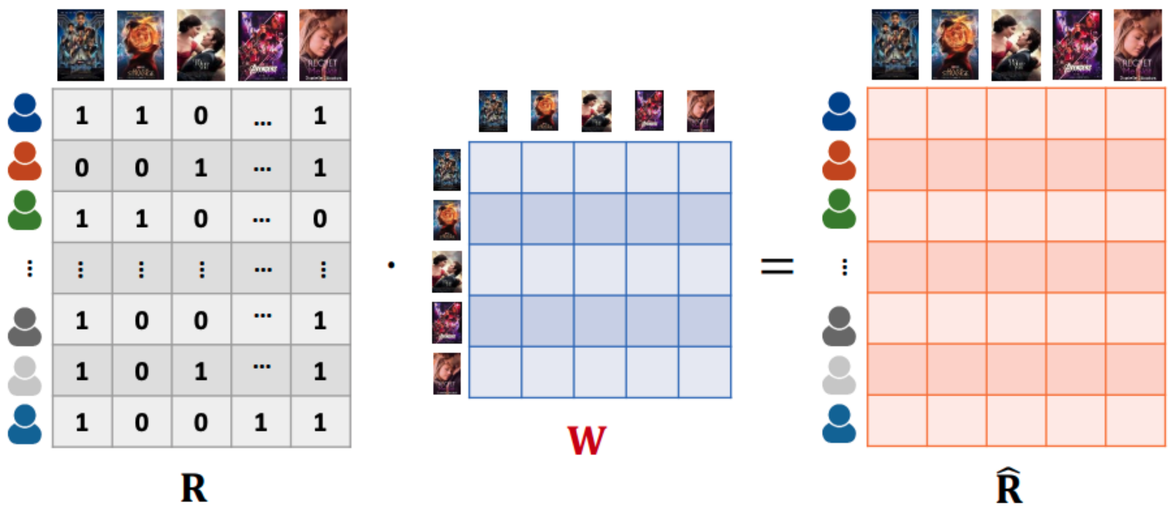
- Recommend similar items to what the user has clicked.

- **Fast speed** : it provides sparse item neighborhoods.
- **Low accuracy** : it is not a learning-based model.



Item-based CF : A Matrix View

- The item similarity matrix is used in practice.
  - Keep only top-k similarities, e.g., Jaccard and cosine similarities.
  - **It is simple, but learning is limited.**



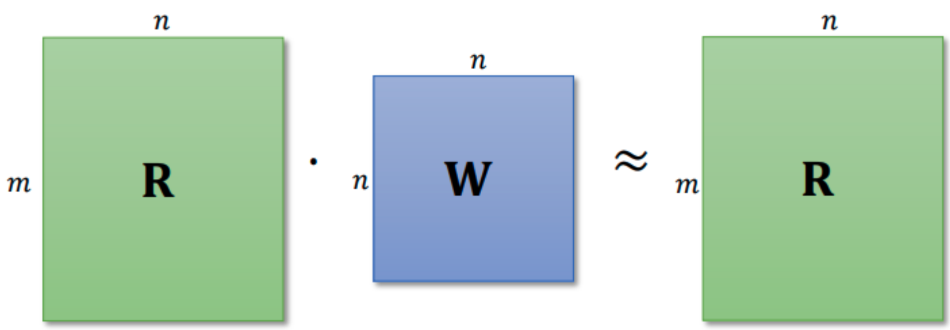
Sparse Linear Method (SLIM)

- Motivation

- Generating **fast** recommendation lists
- **High-quality** recommendations

- Key ideas
  - Retain the nature of itemKNN : **keep W being sparse.**
  - Optimize the recommendation performance : **learn W from R.**
- How to model this idea?

➢ **Learning the item-item similarity matrix W by approximating the user-item rating matrix R**



➢ **However, without the constraint for diagonal elements of W, the solution can be trivial, i.e., identity matrix.**

제약 조건 : 대각선 0 , 불필요한 value 0

Learning SLIM

➢ **The optimization problem is formulated by:**

L2-norm regularization

L1-norm regularization: making $\mathbf{W}$ sparse

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{R}-\mathbf{R}\mathbf{W}\|_F^2 + \boxed{\frac{\beta}{2}\|\mathbf{W}\|_F^2} + \boxed{\lambda\|\mathbf{W}\|_1}$$

$\|A\|_2$: Frobenius norm

$$\|A\|_2 = \sqrt{\sum_{u=1}^{m}\sum_{i=1}^{n}|a_{ij}|^2}$$

subject to $\mathbf{W} \geq \mathbf{0}$ **and** $\mathbf{diag}(W) = \mathbf{0}$.

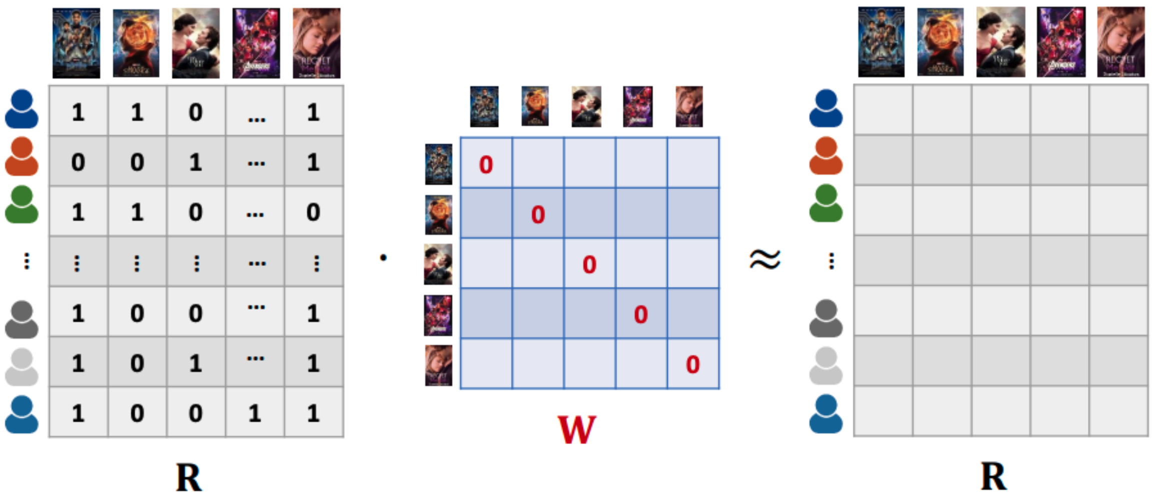
All similarity values are non-negative.

Diagonal elements in $\mathbf{W}$ should be zero.

➢ **How to compute $\mathbf{W}$?**

- The columns of $\mathbf{W}$ are independent: **easy to parallelize**.

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{R}-\mathbf{R}\mathbf{W}\|_F^2 + \frac{\beta}{2}\|\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_1$$



Advantages of SLIM

- The recommendation score for a new item can be calculated as an aggregation of other items.
- A psarse aggregation coefficient matrix W is learned for SLIM to make the aggregation very fast.

- It is learned by solving the 1-norm and 2-norm regularized optimization problems for sparse representation.
- It can be computed in parallel.

---

2. Factored Item Similarity Models (FISM)

Recap : Sparse Linear Models (SLIM)

➢ **Prediction**

- $\mathbf{R}_{u*}$: the item rating vector of user u on all items, $\mathbb{R}^{1\times n}$
- $\mathbf{W}$: $n \times n$ sparse item matrix

$$\hat{\mathbf{R}}_{u*} = \mathbf{R}_{u*}\mathbf{W}$$

➢ **Objective function**

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \frac{\beta}{2}\|\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_1$$

$$\mathbf{R}\ \mathbf{W} \approx \mathbf{R}$$

subject to $W \geq 0$ **and** $\mathrm{diag}(W) = 0.$

Regularized SVD (RSVD)

- **Prediction for user u and movie i**

$$\hat{r}_{ui} = b_u + b_i + \mathbf{U}_u \mathbf{V}_i^{\mathrm{T}}$$

- b_u, b_i: user bias and item bias
- $\mathbf{U}_u$: latent user vector for user u, i.e., $\mathbb{R}^{1\times k}$ vector
- $\mathbf{V}_i$: latent item vector for user i , i.e., $\mathbb{R}^{1\times k}$ vector

- **Objective function: training with SGD**

$$\min_{\mathbf{U},\mathbf{V},\mathbf{b}_u,\mathbf{b}_i} \frac{1}{2}\left\|\mathbf{R} - \left(\mathbf{U}\mathbf{V}^{\mathrm{T}} + \mathbf{b}_u + \mathbf{b}_i\right)\right\|_F^2 + \frac{\lambda}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{b}_u\|_F^2 + \|\mathbf{b}_i\|_F^2\right)$$

Neighborhood-based SVD (NSVD)

- **RSVD has $O(mk + nk)$ parameters.**
  - m is the number of users and n is the number of items.
  - k is the number of features.

- **To decrease the number of parameters, model $\mathbf{U}_u$ as a function of binary vectors indicating which items the user rated.**

- **How?**

Neighborhood-based SVD (NSVD)

- Making a prediction for user u and movie i

The user vector is approximated by

$$\hat{r}_{ui} = b_u + b_i + \boxed{\frac{1}{\sqrt{|\mathcal{R}_u^+| + 1}} \sum_{j \in \mathcal{R}_u^+} \boldsymbol{P}_j} \boldsymbol{Q}_i^T$$

$\boldsymbol{P}_j \in \mathbb{R}^{n \times k}$, $\boldsymbol{Q}_i \in \mathbb{R}^{n \times k}$, and $k \ll n$

$\mathcal{R}_u^+$: a set of items rated by user u

- Minimizing the following objective function

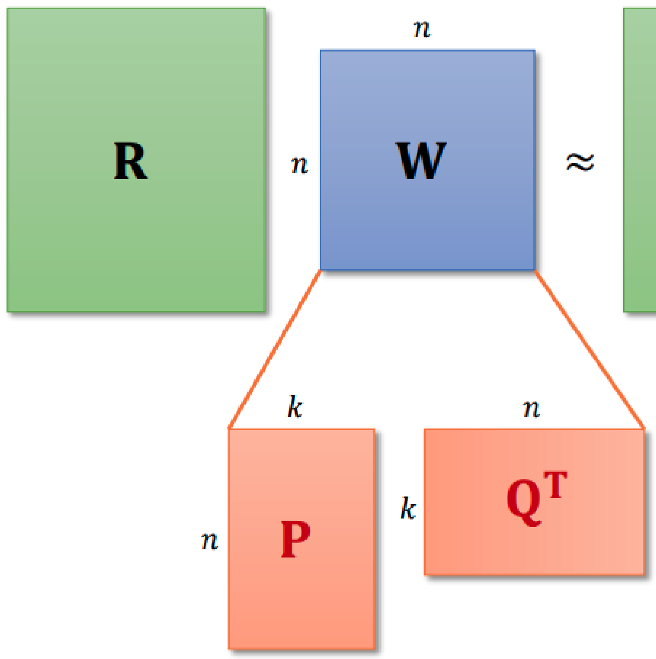
$$\min_{\mathbf{P},\mathbf{Q},\mathbf{b}_u,\mathbf{b}_i} \frac{1}{2} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \|r_{ui} - \hat{r}_{ui}\|_F^2 + \frac{\lambda}{2} (\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2 + \|\mathbf{b}_u\|_F^2 + \|\mathbf{b}_i\|_F^2)$$

Motivation of FISM

- SLIM relies on learning similarities between items.
  - It fails to capture **the dependencies between items that have not been co-rated** by at least one user.
- Matrix factorization alleviates this problem by **projecting the data into a low-dimensional space**.
  - It is better to learn the relationships between the users and items, including items which are not co-rated.
- NSVD **does not exclude the diagonal entries** while estimating the ratings during learning and prediction phases.
  - It may lead to rather **trivial estimates**, i.e., **recommending itself**.

Factored Item Similarity Models (FISM)

- Factorize the item similarity matrix into the inner product of two matrices.



The matrix $\mathbf{W}$ is decomposed into two latent matrices $\mathbf{P}$ and $\mathbf{Q}$, i.e., $\mathbb{R}^{n \times k}$.

Factored Item Similarity Models (FISM)

- **Making a prediction for user u and movie i**

$$\hat{r}_{ui} = b_u + b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{P}_j \mathbf{Q}_i^T$$

$\mathcal{R}_u^+$: a set of items rated by user u

n_u^+: the number of items rated by user u

α: a user specified parameter in [0, 1]

- **Exclude the predicted item i from $\mathcal{R}_u^+$.**

Factored Item Similarity Models (FISM)

➢ **Minimizing the following optimization problem**

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{b}_u,\mathbf{b}_i} \frac{1}{2}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\|r_{ui}-\hat{r}_{ui}\| + \frac{\beta}{2}(\|\mathbf{P}\|_F^2+\|\mathbf{Q}\|_F^2) + \frac{\lambda}{2}(\|\mathbf{b}_u\|_F^2) + \frac{\gamma}{2}(\|\mathbf{b}_i\|_F^2)$$

- $\mathbf{b}_u$, $\mathbf{b}_i$: user bias and item bias
- β, λ, γ: regularization weights

➢ **The optimization problem is solved using SGD.**

➢ **It is called $\text{FISM}_{\text{RMSE}}$.**

Extension for Ranking Problem

➢ **Utilize pair-wise preferences for relative ranking.**

➢ **Loss function: AUC**

Actual ranking *Predicted ranking*

$$\min \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{R}_u^+, j\in\mathcal{R}_u^-}\left(\boxed{(r_{ui}-r_{uj})} - \boxed{(\hat{r}_{ui}-\hat{r}_{uj})}\right)^2$$

$\mathcal{R}_u^+$: a set of items rated by user u

$\mathcal{R}_u^-$: a set of items unrated by user u

➢ **It is similar to Bayesian personalized ranking (BPR).**

Extension for Ranking Problem

- **Making prediction for user u and movie i**

$$\hat{r}_{ui} = b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{P}_j \mathbf{Q}_i^{\mathrm{T}}$$

- **Minimizing the following objective function**

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{b}_i} \frac{1}{2} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{R}_u^+, j \in \mathcal{R}_u^-} \left\| (r_{ui} - r_{uj}) - (\hat{r}_{ui} - \hat{r}_{uj}) \right\|_F^2 + \frac{\beta}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \frac{\gamma}{2}(\|\mathbf{b}_i\|_F^2)$$

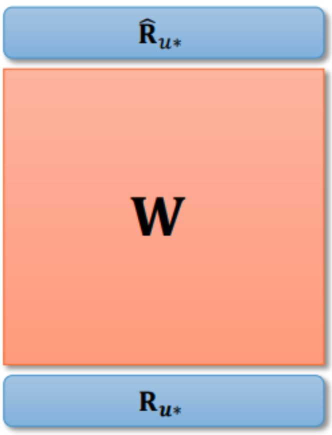
- **It is called $\mathrm{FISM}_{\mathrm{AUC}}$.**

---

3. Embarrassingly Shallow Autoencoders($EASE^R$)

Definition of $EASE^R$

- **The item-item matrix is viewed as the shallow autoencoder.**



Model prediction

$$\mathbf{R}_{u*} \times \mathbf{W} = \hat{\mathbf{R}}_{u*}$$

It is equal to SLIM.

- **The self-similarity is used to reproduce the input as its output.**

SLIM vs. EASE^R

- **The optimization problem of SLIM is formulated by:**

L2-norm regularization

L1-norm regularization: making $\mathbf{W}$ sparse

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \boxed{\frac{\beta}{2}\|\mathbf{W}\|_F^2}$$

subject to and $\mathrm{diag}(\mathbf{W}) = \mathbf{0}$.

Diagonal elements in $\mathbf{W}$ are zero.

- **For efficient computation, EASE^R removes L1 regularization and non-negative condition.**

SLIM vs. EASE^R

- **The optimization problem of EASE^R is formulated by**

$$\min_{\mathbf{W}} \|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_F^2$$

subject to $\boxed{\mathrm{diag}(\mathbf{W}) = \mathbf{0}.}$

Diagonal elements in $\mathbf{W}$ should be zero.

$$\min_{\mathbf{W}} \|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_F^2 + \boxed{2 \cdot \gamma^T \cdot \mathrm{diag}(\mathbf{W})}$$

$\gamma = (\gamma_1, \dots, \gamma_n)^T$ is the vector of Lagrangian multipliers, $\mathbb{R}^{n \times n}$.

Constrained Optimization Problem

➢ **It is solved by minimizing the Lagrangian multiplier.**

$$\hat{\mathbf{W}} = \left(\mathbf{R}^{\mathrm{T}}\mathbf{R} + \lambda\mathbf{I}\right)^{-1} \cdot \left(\mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathrm{diagMat}(\gamma)\right)$$

$\mathrm{diagMat}(\gamma)$: the diagonal matrix, i.e., $\mathbb{R}^{n\times n}$

$\mathbf{I}$: the identity matrix, i.e., $\mathbb{R}^{n\times n}$

➢ **The closed-form solution is as follows.**

$$\hat{\mathbf{W}} = \mathbf{I} - \hat{\mathbf{P}} \cdot \mathrm{diagMat}\left(\vec{\mathbf{1}} \oslash \mathrm{diag}(\hat{\mathbf{P}})\right)$$

$$\hat{\boldsymbol{P}} = \left(\mathbf{R}^{\mathrm{T}}\mathbf{R} + \lambda\mathbf{I}\right)^{-1}$$

$\mathrm{diag}(\hat{\mathbf{P}})$: the diagonal vector , i.e., $\mathbb{R}^{n\times 1}$

29

Solution of EASE[R]

➢ **The learned weights are given by:**

$$\hat{\mathbf{W}}_{ij} = \begin{cases} 0 & \text{if } i = j \\ -\dfrac{\hat{\mathbf{P}}_{ij}}{\hat{\mathbf{P}}_{jj}} & \text{otherwise} \end{cases}$$

- This solution obeys the zero-diagonal condition.

➢ **The off-diagonal elements are determined by $\hat{\mathrm{P}}$.**

- The j-th column is divided by its diagonal element $\hat{\mathbf{P}}_{jj}$.

➢ **Note that $\hat{\mathrm{W}}$ is an asymmetric matrix, while $\hat{\mathrm{P}}$ is symmetric.**