# IT492: Recommendation Systems

Lecture - 05

# Collaborative Filtering: Model-based Methods

Arpit Rana

7th Feb 2022

# Flaws in Memory-based Collaborative Methods

***Memory-based methods*** rely on rating correlation and have the following flaws.

- These methods assume that users can be neighbors only if they have rated common items.
  - This assumption is very limiting, as users having rated a few or no common items may still have similar preferences.

# Flaws in Memory-based Collaborative Methods

***Memory-based methods*** rely on rating correlation and have the following flaws.

- Since only items rated by neighbors can be recommended, the (catalog) coverage of such methods can also be limited.

# Flaws in Memory-based Collaborative Methods

***Memory-based methods*** rely on rating correlation and have the following flaws.

- These methods suffer from (or are sensitive to) the lack of available ratings (a.k.a. ***sparsity***).
  - Users or items newly added to the system may have no ratings at all, a problem known as ***cold-start***.

# Learning-based Collaborative Methods

***Learning-based methods*** obtain the similarity or affinity between users and items

- by defining a parametric model that describes the relation between users, items or both, and then
- computes the model parameters through an optimization process.

# Learning-based Collaborative Methods

***Learning-based methods*** have a few advantages over memory-based methods.

- These methods can capture high-level patterns and trends in the data, are generally more robust to outliers,
- They are known to generalize better than approaches solely based on local relations.
- These methods require less memory because the relations between users and items are encoded in a limited set of parameters.
- Since the parameters are usually learned offline, the online recommendation process is generally faster.

# Learning-based Methods

***Learning-based methods*** that use neighborhood or similarity information can be divided in two categories:

- Factorization methods (e.g. MF), and
- Adaptive neighborhood learning methods (e.g. SLIM).

# Factorization Methods

***Factorization methods***

- These methods project users and items into a reduced latent space that captures their most salient features.
- A relation between two users can be found, even though these users have rated different items, thus, are generally less sensitive to sparse data.
- There are essentially two ways in which factorization can be used:
  - Factorization of a ***sparse similarity matrix***, and
  - Factorization of a **user-item rating matrix**.

# Required Concepts of Linear Algebra

## Eigenvalues and Eigenvectors

- *Eigenvectors* are the vectors that ***does not change its orientation*** when multiplied by the transition matrix, but it ***just scales by a factor*** of corresponding *eigenvalues*.

$$Av = \lambda v$$

**Where**

$$A \in R^{m \times m}$$

$$v \in R^{m \times 1}$$

$$\lambda \in R^{m \times m}$$

# Required Concepts of Linear Algebra

## Eigenvalues and Eigenvectors

- *Eigenvectors* are the vectors that ***does not change its orientation*** when multiplied by the transition matrix, but it ***just scales by a factor*** of corresponding *eigenvalues*.

$$Av = \lambda v$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

## Diagonalization

- If square matrix "*A*" has same number of linearly independent eigenvectors as the rank, it can be turned into a diagonal matrix.
- Suppose, we have "*m*" linearly independent eigenvectors of *A*, we denote their matrix as "*S*" -

$$AS = S\Lambda$$

$$S^{-1}AS = \Lambda$$

# Required Concepts of Linear Algebra

**Decomposition**

- If square matrix "*A*" has same number of linearly independent eigenvectors as the rank, it can be turned into a diagonal matrix.
- Suppose, we have "*m*" linearly independent eigenvectors of *A*, we denote their matrix as "*S*" -

$$AS = S\Lambda$$

$$A = S\Lambda S^{-1}$$

# Required Concepts of Linear Algebra

## Symmetric Matrix

- If a square matrix "*A*" remains unchanged when we take its transpose, it is called *symmetric matrix*.

$$A^T = A$$

- Symmetric matrix has two important properties:
  - The eigenvalues are ***real*** (i.e. doesn't have imaginary part)
  - The eigenvectors are ***orthogonal*** (i.e. perpendicular to each other)

# Required Concepts of Linear Algebra

## Eigendecomposition of a Symmetric matrix

- Since, symmetric matrix has orthogonal eigenvectors -

$$
\begin{aligned}
A &= Q\Lambda Q^{-1} \\
&= Q\Lambda Q^{T}
\end{aligned}
$$

# Factorization of Sparse Similarity Matrix

- Neighborhood similarity measures (e.g. correlation similarity) are usually very sparse.
- A simple solution to densify a sparse similarity matrix is to compute a ***low-rank*** approximation of this matrix with a factorization method.

# Factorization of Sparse Similarity Matrix

Let $W$ be a symmetric matrix of rank $n$ representing either user or item similarities.

- We wish to approximate W with a matrix $\hat{W} = QQ^T$ of lower rank $k < n$ by minimizing the following objective function.

$$
\begin{aligned}
E(Q) &= \|W - QQ^\top\|_F^2 \\
&= \sum_{i,j} \left( w_{ij} - \mathbf{q}_i \mathbf{q}_j^\top \right)^2,
\end{aligned}
$$

- Finding the factor matrix $Q$ is equivalent to computing the ***eigenvalue decomposition of W***.

$$W = V \Lambda V^T$$

# Factorization of Sparse Similarity Matrix

Let *W* be a symmetric matrix of rank *n* representing either user or item similarities.

- Let $V_k$ be a matrix formed by the *k* principal (normalized) eigenvectors of W, which correspond to the axes of the *k*-dimensional latent subspace.
- The coordinates $q_i \in R^k$ of an item *i* in this subspace is given by the $i^{th}$ row of matrix $Q = V_k \Lambda_k^{1/2}$
- The item similarities computed in this latent subspace is given by -

$$\begin{aligned}\hat{W} &= QQ^T \\ &= V_k \Lambda_k V_k^T\end{aligned}$$

# Factorization of Sparse Similarity Matrix

Let $W$ be a symmetric matrix of rank $n$ representing either user or item similarities.

- A user $u$, represented by the $u^{th}$ row $r_u$ of the rating matrix $R$, is projected in the plane defined by $V_k$

$$\mathbf{r}'_u = \mathbf{r}_u V_k.$$

- In an offline step, the users of the system are clustered in this subspace using recursive hierarchical clustering.
- Then, the rating of user $u$ for an item $i$ is evaluated as the mean rating for $i$ made by users in the same cluster as $u$.

**Next lecture -**

Collaborative Methods
(Model-based contd…)