IT492: Recommendation Systems

Lecture - 06

Collaborative Filtering: Model-based Methods

Arpit Rana

11th 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

Positive definiteness

- Positive definite matrix helps us solve optimization problems, decompose the matrix into a more simplified matrix, etc.
- To determine if the matrix is positive definite or not, we check the following conditions.

  1) check if the matrix is symmetric

  2) check if all eigenvalues are positive

  3) check if all the sub-determinants are also positive

  4) check if the quadratic form is positive

Required Concepts of Linear Algebra

Quadratic Form

- Let's define and check what's a quadratic form is.

Quadratic Form: $x^T A x$

where, $x \in R^{m \times 1}$

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

$x^T A x$ is a scalar value.

Required Concepts of Linear Algebra

Quadratic Form:

$$x^T A x = [x_1, x_2, \cdots, x_m] \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$= [x_1, x_2, \cdots, x_m] \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m \end{bmatrix}$$

$$\begin{aligned} &= x_1(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m) \\ &\quad + x_2(a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m) \\ &\qquad \vdots \\ &\quad + x_m(a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m) \end{aligned}$$

$$= \sum_{i \le j}^{m} a_{ij} x_i x_j$$

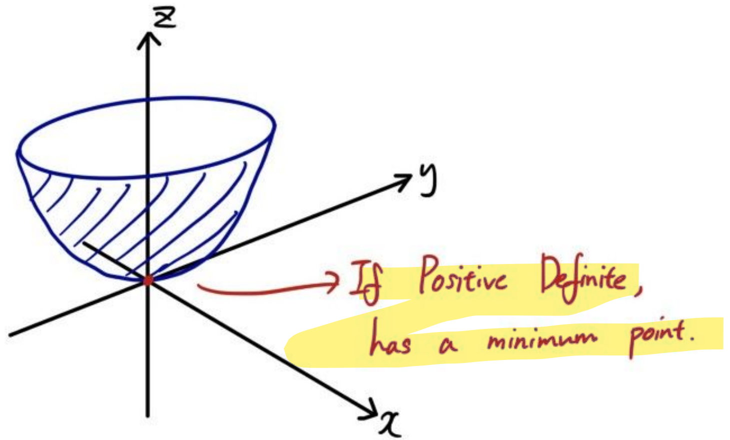
Required Concepts of Linear Algebra

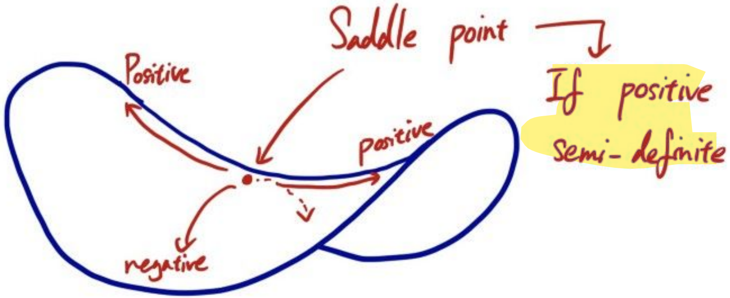
Based on the signs of the ***quadratic form***, we can classify the definiteness into three categories:

- ***Positive definite if (Quadratic form) > 0***
- ***Positive semi-definite if (Quadratic form) ≥ 0***
- ***Negative definite if (Quadratic form) < 0***

Required Concepts of Linear Algebra

Geometric Interpretation of Positive Definiteness





Required Concepts of Linear Algebra

Positive Definiteness

- If a matrix "A" is not symmetric, we can still make a use of positive definiteness.
- Matrix A^TA is symmetric and square and the quadratic form of such a matrix is -

$$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 > 0$$

- ***So, we can simply multiply the matrix that's not symmetric by its transpose and the product will become symmetric, square, and positive definite!***

Required Concepts of Linear Algebra

Singular Value Decomposition

- Unlike eigendecomposition where the matrix has to be a square matrix, ***SVD* allows *to decompose a rectangular matrix*.**

$$A = U\Sigma V^T$$

$$\underset{m \times n}{\overset{\mathrm{A}}{\begin{pmatrix} x_{11} & x_{12} & & x_{1n} \\ & & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times m}{\overset{\mathrm{U}}{\begin{pmatrix} u_{11} & & u_{m1} \\ & \ddots & \\ u_{1m} & & u_{mm} \end{pmatrix}}} \underset{m \times n}{\overset{\boldsymbol{\Sigma}}{\begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_r & \\ & & & \ddots \\ 0 & & & 0 \end{pmatrix}}} \underset{n \times n}{\overset{\mathrm{V}^{\mathrm{T}}}{\begin{pmatrix} v_{11} & & v_{1n} \\ & \ddots & \\ v_{n1} & & v_{nn} \end{pmatrix}}}$$

Required Concepts of Linear Algebra

Singular Value Decomposition

- Unlike eigendecomposition where the matrix has to be a square matrix, ***SVD*** allows ***to decompose a rectangular matrix***.

A unitary matrix is a square matrix of complex numbers. A unitary matrix is a matrix, whose inverse is equal to its conjugate transpose.

$$A = U\Sigma V^T$$

where,

$A \in R^{m \times n}$ (Real or Complex matrix)

$U \in R^{m \times m}$ (Real or Complex ***Unitary*** matrix)

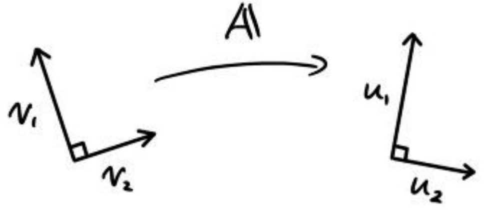
$\Sigma \in R^{m \times n}$ (Real or Complex ***Diagonal*** matrix)

$V \in R^{n \times n}$ (Real or Complex ***Unitary*** matrix)

Required Concepts of Linear Algebra

Singular Value Decomposition

Intuitive explanation of SVD



$$\begin{cases} u_1 = A v_1 \\ u_2 = A v_2 \end{cases}$$

To make it orthonormal normalize u with singular values

$$\begin{cases} \sigma_1 u_1 = A v_1 \\ \sigma_2 u_2 = A v_2 \end{cases}$$

$$A \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \Rightarrow \boxed{AV = U\Sigma} \Rightarrow A = U\Sigma V^{-1} = U\Sigma V^T$$

Required Concepts of Linear Algebra

Singular Value Decomposition

- Unlike eigendecomposition where the matrix has to be a square matrix, ***SVD*** allows ***to decompose a rectangular matrix***.

$$A = U\Sigma V^T$$

- So, "U" and "V" are *orthogonal* matrices when real. i.e.

$$UU^T = U^TU = I$$
$$VV^T = V^TV = I$$

Required Concepts of Linear Algebra

Singular Value Decomposition

$$A = U \Sigma V^T$$

$$A^T = V \Sigma U^T$$

(i) $A^T A = V \Sigma U^T U \Sigma V^T$

$$= V |\Sigma|^2 V^T \quad (\because U^T U = I)$$

$A^T A$ is Positive Definite!

$\therefore V$ is orthonormal eigenvectors of $A^T A$

(Recall $\underset{\text{symmetric}}{A} = Q \Lambda Q^T$)

(ii) $A A^T = U \Sigma V^T V \Sigma U^T$

$$= U |\Sigma|^2 U^T$$

$\hookrightarrow$ Get U

Required Concepts of Linear Algebra

Singular Value Decomposition

- "U" and "V" are orthogonal matrices with orthonormal eigenvectors chosen from AA^T and A^TA respectively.
- Σ is a diagonal matrix with r elements equal to the root of the positive eigenvalues of AA^T or A^TA (both matrices have the same positive eigenvalues anyway).

$$
\begin{pmatrix}
\sqrt{\lambda_1} & & & & & \\
& \sqrt{\lambda_2} & & & & \\
& & \ddots & & & \\
& & & \sqrt{\lambda_r} & & \\
& & & & \ddots & \\
& & & & & 0
\end{pmatrix}
\equiv
\overset{\Sigma}{\begin{pmatrix}
\sigma_1 & & & & & \\
& \sigma_2 & & & & \\
& & \ddots & & & \\
& & & \sigma_r & & \\
& & & & \ddots & \\
& & & & & 0
\end{pmatrix}}
$$

σ_2: *singular value*

Factorization of User Item Rating Matrix

- The problems of cold-start and limited coverage can also be alleviated by factoring the user-item rating matrix.

Factorization of User Item Rating Matrix

Let ***R*** be a user-item rating matrix of rank ***n*** and the order of ***|U| x |I|***.

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

$$
\begin{aligned}
E(P, Q) &= ||R - PQ^\top||_F^2 \\
&= \sum_{u,i} \left(r_{ui} - \mathbf{p}_u \mathbf{q}_i^\top\right)^2 .
\end{aligned}
$$

- Finding the factor matrices P and *Q* is equivalent to computing the ***Singular Value decomposition of R***.

$$R = P\Sigma Q^T$$

Factorization of User Item Rating Matrix

- A user *u*'s rating of item *i*, which is denoted by r_{ui}, leading to the estimate -

$$\hat{r}_{ui} = q_i^T p_u.$$

- Applying SVD in the collaborative filtering domain requires factoring the user-item rating matrix.
- Conventional SVD is undefined when knowledge about the matrix is incomplete.
- Applying SVD on user-item rating matrix raises difficulties due to its sparsity.
- Moreover, carelessly addressing only the relatively few known entries is highly prone to overfitting.

Factorization of User Item Rating Matrix

- Recent works suggested modeling directly the observed ratings only, while avoiding overfitting through a regularized model.
- To learn the factor vectors (p_u and q_i), the system minimizes the regularized squared error on the set of known ratings:

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

Here, κ is the set of the (u, i) pairs for which r_{ui} is known (the training set).

Next lecture -

Collaborative Methods
(Model-based contd…)