Recommender System

Spring 2024

Hongchang Gao

Evaluation-1

- 1. MAE and RMSE
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | p_i - r_i |$$

  - Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (p_i - r_i)^2}$$

Evaluation-2

Actually good
Item 237
Item 899

Recommended (predicted as good)
Item 345
Item 237
Item 187

- 2. Precision and Recall
  - Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved
    - E.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

  - Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items
    - E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

Evaluation-3

- Rank position is important!
  - Relevant items are more useful when they appear earlier in the recommendation list
  - Particularly important in recommender systems as lower ranked items may be overlooked by users

Actually good
Item 237
Item 899

Scenario-1: Recommended (predicted as good)
Item 237
Item 345
Item 187

Scenario-2: Recommended (predicted as good)
Item 345
Item 237
Item 187

Scenario-3: Recommended (predicted as good)
Item 345
Item 187
Item 237

Evaluation-3

- Precision@K
  - Precision for the top-k recommended items

Top-k recommendation (sorted result)

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

$P@1 = 1/1 = 1$

$P@2 = 1/2 = 0.5$

$P@3 = 1/3 = 0.33$

$P@4 = 2/4 = 0.5$

$P@5 = 3/5 = 0.6$

$P@n = 3/n$

Fig: https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

Evaluation-3

- Recall@K
  - Recall for the top-k recommended items
    - Could return trivial result!

Top-k recommendation (sorted result)

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

Recall@1=

Recall@2=

Recall@3=

Recall@4=

Recall@5=

Recall@6=

Recall@7=

Evaluation-3

- Average Precision

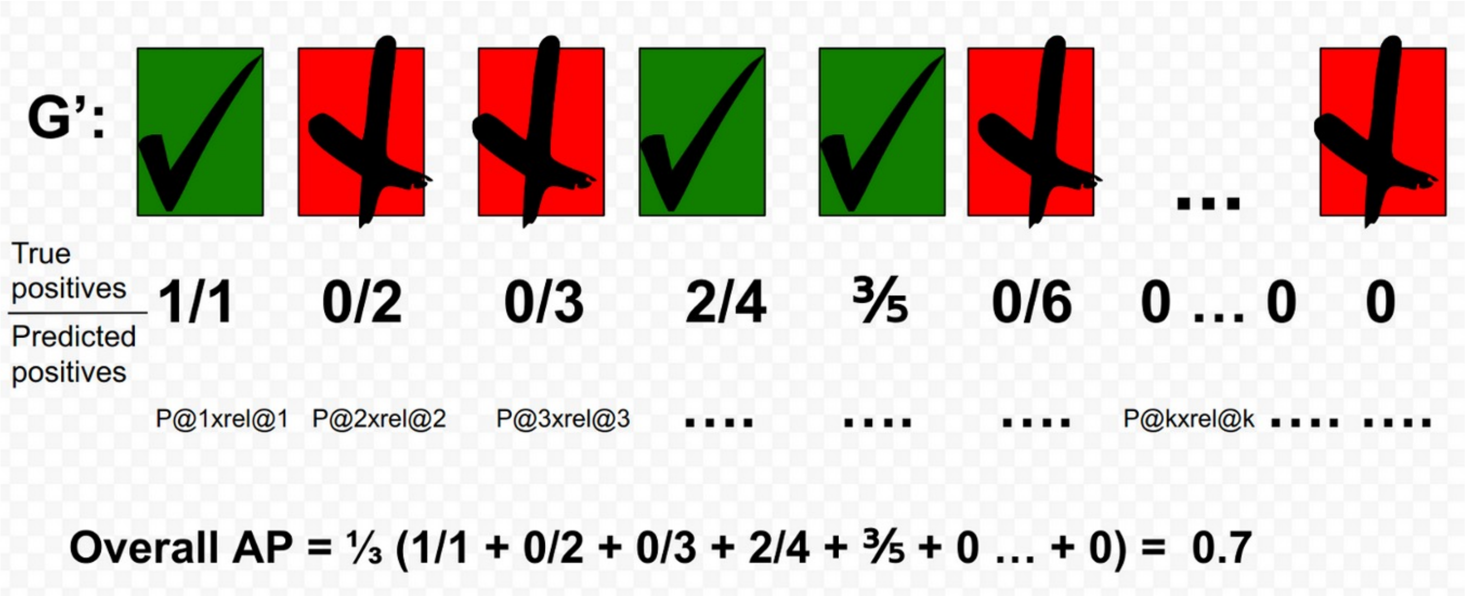
$$\mathrm{AP@}n = \frac{1}{GTP} \sum_{k=1}^{n} \mathrm{precision@}k \times \mathrm{relevance@}k$$

- n the total number of items you are interested in
- GTP the total number of ground truth positives
- $\mathrm{relevance@}k$: an indicator function which equals 1 if the item at rank k is relevant and equals to 0 otherwise.

Evaluation-3

- Average Precision

$$AP@n = \frac{1}{GTP} \sum_{k=1}^{n} \text{precision}@k \times \text{relevance}@k$$



$P@1 = 1/1 = 1$

$P@2 = 1/2 = 0.5$

$P@3 = 1/3 = 0.33$

$P@4 = 2/4 = 0.5$

$P@5 = 3/5 = 0.6$

$P@n = 3/n$

Evaluation-4

- Normalized Discounted Cumulative Gain (NDCG)
  - Measure the ranking quality

- Cumulative Gain (CG)
  - The sum of the relevance of recommendations
  - Does not consider the rank!

$$CG_{pos} = \sum_{i=1}^{pos} rel_i$$

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$CG_{pos} = 0 + 1 + 1 + 1 + 0 = 3$$

Evaluation-4

- Discounted Cumulative Gain (DCG)
  - Logarithmic reduction factor
  - highly relevant item appearing lower in the recommendation list should be penalized

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

Rank	Hit?
1	
2	
3	X
4	X
5	x

Evaluation-4

- Idealized discounted cumulative gain (IDCG)
  - Assume that items are ordered by decreasing relevance
  - the maximum possible DCG

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

Rank	Hit?	Ideal
1		x
2	X	x
3	X	x
4	X	
5		

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

Evaluation-4

- Normalized discounted cumulative gain (NDCG)
  - Normalized to the interval [0..1]

$$NDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

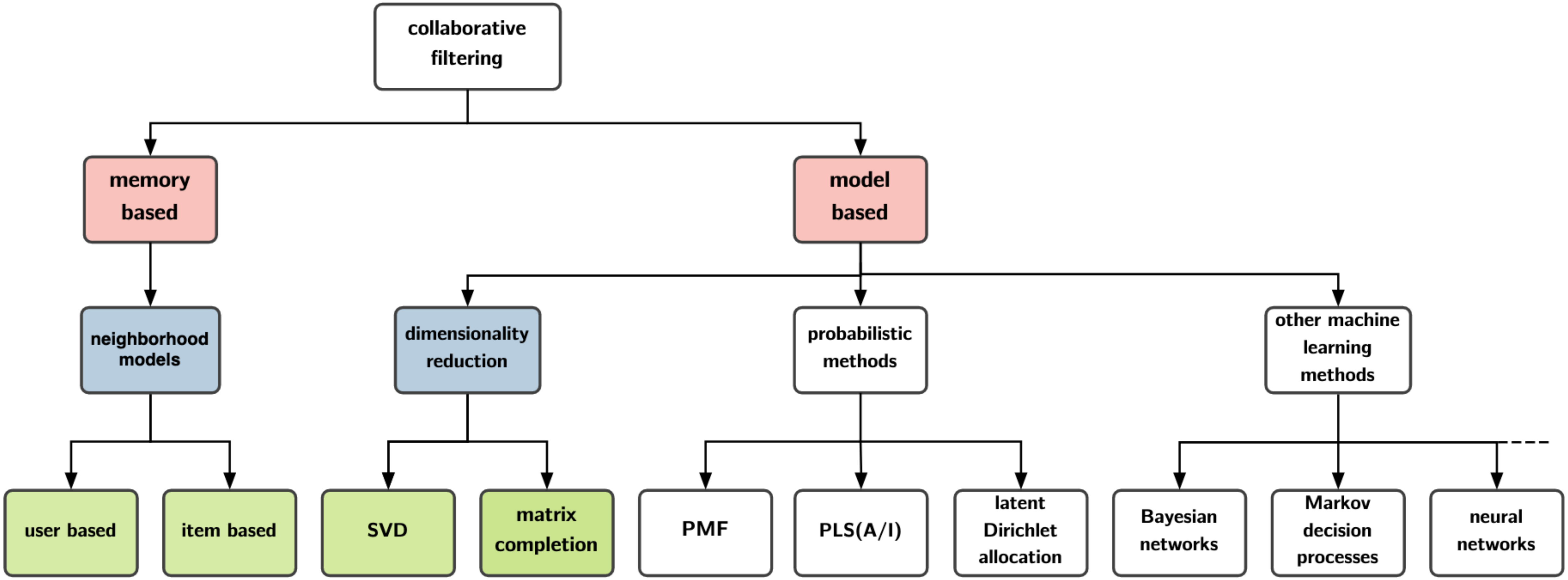
Rank	Hit?	Ideal
1		x
2	X	x
3	X	x
4	X	
5		

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

$$NDCG_5 = \frac{DCG_5}{IDCG_5} = \frac{2.13}{2.63} \approx 0.81$$

Collaborative Filtering



Singular Value Decomposition

- Provide a way to understand the hidden structure in the data
  - Each row of U can be viewed as the representation of a user
  - Each column of V can be viewed as the representation of an item

$$
\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\top}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}
$$

- **X**: *m x n* matrix (e.g., m users, n videos)
- **U**: *m x r* matrix (m users, r factors)
- **S**: *r x r* diagonal matrix (strength of each ‘factor’) (r: rank of the matrix)
- **V**: r x n matrix (n videos, r factor)

Singular Value Decomposition

- Provide a low-rank approximation for the rating matrix

Truncated SVD $\boldsymbol{A}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^T$ of $\boldsymbol{A}$ thus satisfies

$$\|\boldsymbol{A} - \boldsymbol{A}_k\|_F = \min_{\mathrm{rank}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_F$$

	Avatar	The Matrix	Up
Marco	?	4	2
Luca	3	2	?
Anna	5	?	3

$$\boldsymbol{A}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^T$$

	Avatar	The Matrix	Up
Marco	? 5	4	2
Luca	3	2	2 ?
Anna	5	? 2	3

$$\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

Singular Value Decomposition

- Problems:
  - Complete input matrix:
    - All elements should be available
  - Large portion of missing values
  - Heuristics to pre-fill missing values
    - Item's average rating
    - Missing values as zeros

	Avatar	The Matrix	Up
Marco	?	4	2
Luca	3	2	?
Anna	5	?	3

Matrix Completion

- Matrix completion:
  - No need to pre-fill missing values
  - Good performance
    - The best single-model approach to collaborative filtering

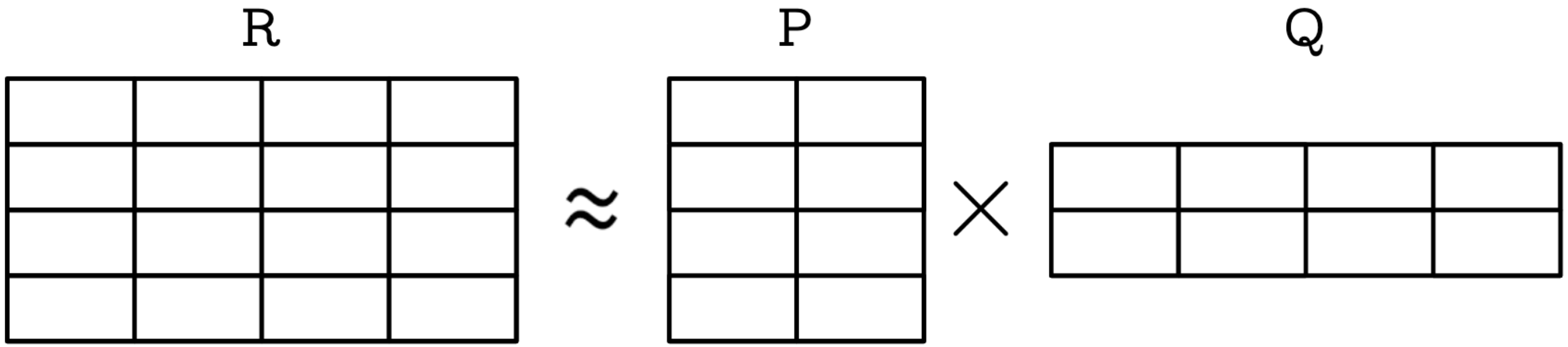
	Avatar	The Matrix	Up
Marco	?	4	2
Luca	3	2	?
Anna	5	?	3

$r_{ij} \approx p_i q_j$

	Avatar	The Matrix	Up
Marco	? 5	4	2
Luca	3	2	2 ?
Anna	5	? 2	3

Matrix Completion

- Matrix completion
  - Learn a latent representation (vector) for each user and each item
  - Missing values are estimated by the dot product



$$r_{ij} \approx p_i q_j$$

Matrix Completion-Example

- Assume the dimensionality of the latent representation is 1

	Avatar	The Matrix	Up
Anni		4	2
Bob	3	2	
Charlie	5		3

Learn latent representation

Latent representation for items

	Avatar (2.24)	The Matrix (1.92)	Up (1.18)
Anni (1.98)		4	2
Bob (1.21)	3	2	
Charlie (2.30)	5		3

Latent representation for users

Matrix Completion

- Loss function

$$\min_{\boldsymbol{Q},\boldsymbol{P}} \sum_{(i,j)\in\Omega} (v_{ij} - [\boldsymbol{Q}^T\boldsymbol{P}]_{ij})^2$$

	Avatar (*2.24*)	**The Matrix** (*1.92*)	**Up** (*1.18*)
Anni (*1.98*)		**4** (*3.8*)	**2** (*2.3*)
Bob (*1.21*)	**3** (*2.7*)	**2** (*2.3*)	
Charlie (*2.30*)	**5** (*5.2*)		**3** (*2.7*)

Matrix Completion

- Inference

	Avatar (2.24)	**The Matrix** (1.92)	**Up** (1.18)
Anni (1.98)		**4** (3.8)	**2** (2.3)
Bob (1.21)	**3** (2.7)	**2** (2.3)	
Charlie (2.30)	**5** (5.2)		**3** (2.7)

	Avatar (2.24)	**The Matrix** (1.92)	**Up** (1.18)
Anni (1.98)	? (4.4)	**4** (3.8)	**2** (2.3)
Bob (1.21)	**3** (2.7)	**2** (2.3)	? (1.4)
Charlie (2.30)	**5** (5.2)	? (4.4)	**3** (2.7)

Matrix Completion

- Introduce bias and regularization

$$\min_{\boldsymbol{Q},\boldsymbol{P},\boldsymbol{u},\boldsymbol{m}} \sum_{(i,j)\in\Omega} \left(v_{ij} - \mu - u_i - m_j - [\boldsymbol{Q}^T\boldsymbol{P}]_{ij}\right)^2$$

$$+ \lambda\left(\|\boldsymbol{Q}\| + \|\boldsymbol{P}\| + \|\boldsymbol{u}\| + \|\boldsymbol{m}\|\right)$$

  - mu: the average rating over all items
  - m_j: the bias for the j-th item
  - u_i: the bias for the i-th user

Matrix Completion

- Optimization (simple case)

$$L_{ij}(P,Q) = (r_{ij} - p_i q_j)^2$$

SGD to minimize the squared loss iteratively computes:

$$p_i \leftarrow p_i - \eta \frac{\partial L_{ij}(P,Q)}{\partial p_i} = p_i + \eta(\varepsilon_{ij} \cdot q_j)$$
$$q_j \leftarrow q_j - \eta \frac{\partial L_{ij}(P,Q)}{\partial q_j} = q_j + \eta(\varepsilon_{ij} \cdot p_i)$$

where $\varepsilon_{ij} = r_{ij} - p_i q_j$