Evaluating Recommender Systems

MEI

CONSTANTINO MARTINS, CATARINA FIGUEIREDO, DULCE MOTA AND FÁTIMA RODRIGUES

Disclaimer

- **Some of this material/slides are adapted from several:**
  - Presentations found on the internet;
  - Papers
  - Books;
  - Web sites
  - …

Evaluating Recommender Systems

- ❑ Is a RS efficient with respect to a specific criteria like:
  - ✓ Accuracy, user satisfaction, response time, **serendipity**, online conversion,, ….
- ❑ Do customers like/buy recommended items?
- ❑ Do customers buy items they otherwise would have not?
- ❑ Are they satisfied with a recommendation after purchase?

Evaluating Recommender Systems

- ❑Evaluating the quality of recommendation algorithms essentially involves assessing the degree of acceptance of the recommendations, by quantifying the number of times users accept or reject recommended items
- ❑This evaluation can be conducted using different **types of metrics**, which can be either **precision or coverage**
- ❑The metrics for measuring the precision of recommendation systems (RS) can be divided into **statistical metrics and decision support precision metrics**

Decision support precision metrics

- ❑ Measure the system's ability to select the preferred products for the user
- ❑ A recommendation of an item can have four possible outcomes:
  1. True Positive (TP)
  2. True Negative (TN)
  3. False Positive (FP)
  4. False Negative (FN)

	Recommended	Not Recommended
Preferred	TP	FN
Not Preferred	FP	TN

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Accuracy - calculates the ratio between the correct predictions, TP and TN, and all predictions made

Precision

- **Precision:** a measure of exactness, determines the fraction of **relevant items retrieved out of all items retrieved**

Precision = $\frac{TP}{TP+FP}$

Precision - calculates the probability of a recommended item being relevant (Good recommendations/all recommendations)
Related with the **quality** of relevant items retrieved

- E.g. the proportion of recommended movies that are actually good

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

A precision of 0.7 (70%) or higher can be considered good, although higher values are preferable

Recall

- ❑**Recall:** a measure of completeness, determines the fraction of **relevant items retrieved out of all relevant items**

Recall = $\frac{TP}{TP+FN}$

Recall calculates the probability of a relevant item being recommended (Good recommendations/”all good recommendations”)
Related with the **quantity** of relevant items retrieved

- ❑E.g. the proportion of all good movies recommended

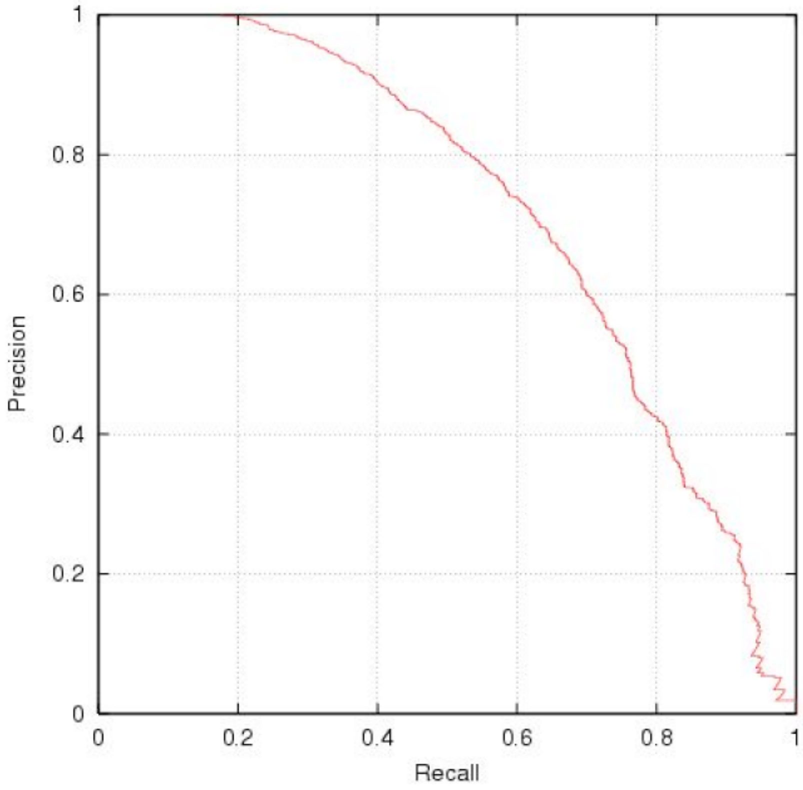
$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

A good recall value for recommendation systems is generally above 0.6, with values over 0.8 being excellent
However, it is always crucial to consider the balance between recall and precision to ensure that the system provides relevant and useful recommendations to users

Precision vs. Recall

- E.g. typically when a recommender system is tuned to increase precision, recall decreases as a result (or vice versa)

Precision and **recall** are inversely related in many cases. Increasing one may decrease the other
Precision focuses on the quality of the recommendations
Recall focuses on the quantity of relevant items recommended



F-measure

- The **F-measure** or **F_1 Metric (F1-score)** attempts to combine Precision and Recall into a single value for comparison purposes
- May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- The F_1 Metric gives equal weight to precision and recall

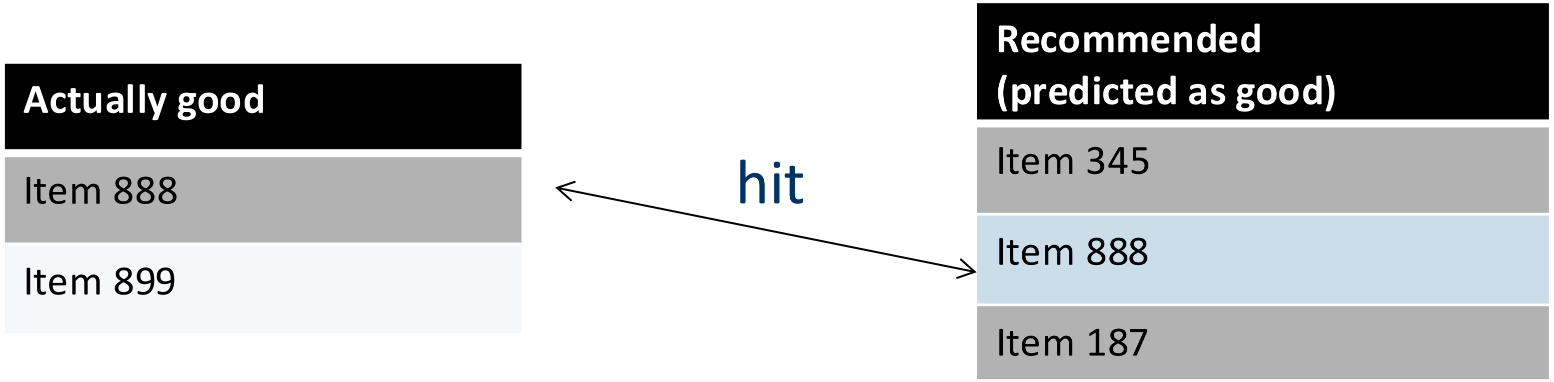
Harmonic Mean=a+b2ab

The harmonic mean is used because it gives more weight to the smaller value between the two numbers. This is important for evaluating classification (or recommendation) models because **precision** and **recall** are often inversely related. The harmonic mean helps balance both metrics, as when one value is very low, it significantly reduces the overall score.

The result of the F-measure ranges from 0 to 1, where 1 indicates the best possible performance (perfect precision and recall) and 0 indicates the worst performance

Metrics: Rank position matters

For a user:



Actually good
Item 888
Item 899

Recommended (predicted as good)
Item 345
Item 888
Item 187

Rank metrics extend recall and precision to take the positions of correct items in a ranked list into account

- 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

Rank Score

❑ **Rank Score** extends the recall metric to take the positions of correct items in a ranked list into account

- ✓ Particularly important in recommender systems as lower ranked items may be **overlooked/undervalued by users**

❑ Rank Score is defined as the ratio of the Rank Score of the correct items to best theoretical Rank Score achievable for the user, i.e.

$$rankscore = \frac{rankscore_p}{rankscore_{\max}}$$

$$rankscore_p = \sum_{i \in h} 2^{-\frac{rank(i)-1}{\alpha}}$$

$$rankscore_{\max} = \sum_{i=1}^{|T|} 2^{-\frac{i-1}{\alpha}}$$

Where:

- *h* is the set of correctly recommended items, i.e. hits
- *rank* returns the position (rank) of an item
- *T* is the set of all items of interest
- *α* is the *ranking half life*, i.e. an exponential reduction factor
- *α é o ranking* half-life, que controla a rapidez

com que a relevância diminui à medida que a posição do item desce.

Example

- **Assumptions**:
  - |T| = 3
  - Ranking half life (alpha) = 2

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$rankscore_p = \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} + \frac{1}{2^{\frac{4-1}{2}}} = 1.56$$

$$rankscore_{\max} = \frac{1}{2^{\frac{1-1}{2}}} + \frac{1}{2^{\frac{2-1}{2}}} + \frac{1}{2^{\frac{3-1}{2}}} = 2.21$$

$$rankscore = \frac{rankscore_p}{rankscore_{\max}} \approx 0.71$$

Example

- Reducing the ranking half life (alpha) = 1

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$rankscore_p = \frac{1}{2^{\frac{2-1}{1}}} + \frac{1}{2^{\frac{3-1}{1}}} + \frac{1}{2^{\frac{4-1}{1}}} = 0.875$$

$$rankscore_{\max} = \frac{1}{2^{\frac{1-1}{1}}} + \frac{1}{2^{\frac{2-1}{1}}} + \frac{1}{2^{\frac{3-1}{1}}} = 1.75$$

$$rankscore = \frac{rankscore_p}{rankscore_{\max}} = 0.5$$

Statistical accuracy metri

They are based on comparing the predictions made by the system with the actual user ratings, in order to determine their accuracy

- □ Metrics measure error rate
- □ **Mean Absolute Error** (***MAE***) computes the deviation between predicted ratings and actual ratings
- ✓ Measures the average difference between predicted ratings and actual ratings

$$MAE = \frac{\sum_{i=1}^{N} |p_{u,i} - r_{u,i}|}{N}$$

P$_{u,i}$ represents the predicted rating value of user u for item i, and R$_{u,i}$ represents the actual rating value provided by user u for item i. N stands for the total number of ratings

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

$$RMSE = \frac{\sqrt{\sum_{i=1}^{N} (p_{u,i} - r_{u,i})^2}}{N}$$

When the RMSE or MAE value is lower, it indicates higher quality in the predicted ratings by the algorithm

Data sparsity

- □ Natural datasets include historical interaction records of real users
  - ✓ Explicit user ratings
  - ✓ Datasets extracted from web server logs (implicit user feedback)
- □ Sparsity of a dataset is derived from ratio of empty and total entries in the user-item matrix:
  - ○ Sparsity = $1 - (|R|/(|I| \cdot |U|))$
  - ○ R = ratings
  - ○ I = items
  - ○ U = users

The **denominator** |I|.|U| calculates the total number of **possible ratings** that could be made if every user rated every item
The **numerator** |R| represents the actual number of ratings made by users.
Sparsity is the **proportion of missing ratings** in the matrix, or how much of the matrix is **empty**
A sparsity value of **1** indicates that the matrix is **completely sparse**

Data sparsity

Example:

- Number of ratings (|R|) = 1200 (the number of interactions between users and items)
- Number of items (|I|) = 80 (the number of items in the system).
- Number of users (|U|) = 40 (the number of users).

Calculate the sparsity…

Alter the values of |R|, |I|, and |U| and comment on how these changes affect the sparsity value

How to evaluate SR?

- ❑Scenario:
  - ✓A movie streaming service aims to recommend movies to users based on their preferences and viewing history
- ❑Evaluation Metrics: RMSE and MAE to assess the accuracy of predicted movie ratings
- ❑Precision and Recall can be used to evaluate the relevance of recommendations to movies users like
- ❑Others Metrics:
  - ✓ Normalized Discounted Cumulative Gain (NDCG)
  - ✓Discounted Cumulative Gain (DCG)
  - ✓Mean Average Precision (MAP)
  - ✓Liftindex
  - ✓…….

Evaluation SR

- Surveys / Questionnaires
- Longitudinal research
  - Observations over long period of time
  - E.g. customer life-time value, returning customers
- Case studies
  - Focus on answering research questions about how and why
  - E.g. answer questions like: How recommendation technology contributed to Amazon.com‘s becomes the world‘s largest book retailer?
- Focus group
  - Interviews
  - …

Explanations

MEI

CONSTANTINO MARTINS, CATARINA FIGUEIREDO, DULCE MOTA AND JOAQUIM SANTOS

Explanations in recommender systems

- ❑ Motivation
  - ✓ "The music of U2 is a must-buy for you because . . . ."
  - ❑ Why should recommender systems deal with explanations at all?
  - ❑ The rationale lies in the interaction between the two parties involved—those providing recommendations and those receiving them:
    - ✓ A selling agent may be interested in promoting particular products
    - ✓ A buying agent is concerned about making the right buying decision
    - ✓ Explanations play a pivotal role here, providing insights into why certain products are recommended
    - ✓ This transparency helps users assess relevance and suitability, ultimately fostering trust and confidence in the recommendation process

Types of explanations

❑ Functional
   - ✓ "The car type Jumbo-Family-Van of brand Rising-Sun would be well suited to your family because you have four children and the car has seven seats"

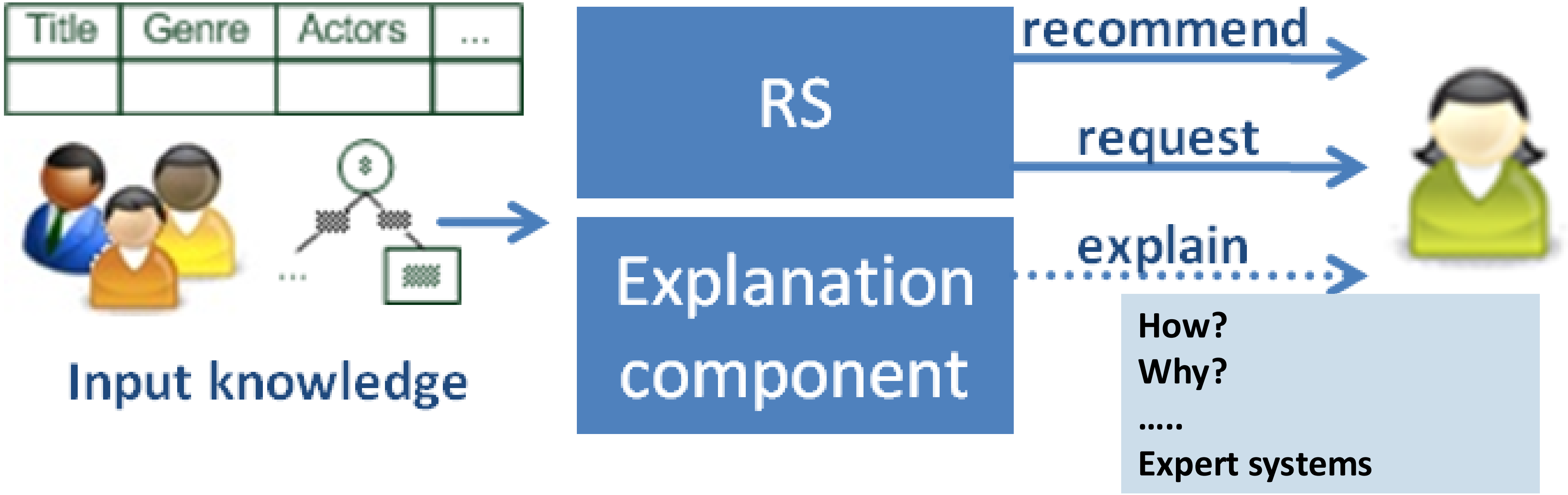
❑ Causal
   - ✓ "The light bulb shines because you turned it on"

❑ Intentional
   - ✓ "I washed the dishes because my brother did it last time"
   - ✓ "You have to do your homework because your dad said so"

❑ Scientific explanations
   - ✓ Express relations between the concepts formulated in various scientific fields and are typically based on refutable theories

What is an explanation in recommender systems?

❑Additional information to explain the system’s output following some objectives



Title	Genre	Actors	...

How?
Why?
.....
Expert systems

The goals for providing explanations

- ❑ Transparency
  - ✓ Provide information so the user can comprehend the reasoning used to generate a specific recommendation
  - ✓ Provide information as to why one item was preferred over another
- ❑ Validity
  - ✓ Allow a user to check the validity of a recommendation
  - ✓ Not necessarily related to transparency
    - ✓ E.g., a neural network (NN) decides that product matches to requirements. Transparent disclosure of NN's computations, will not help, but a comparison of required and offered product features allows customer to judge the recommendation's quality

The goals for providing explanations

- ❑ Trustworthiness
  - ✓ Reduce the uncertainty about the quality of a recommendation
- ❑ Persuasiveness
  - ✓ Persuasive explanations for recommendations aim to change the user's buying behavior
  - ✓ E.g., a recommender system may intentionally highlight a product's positive aspects while downplaying or omitting the negative aspects
- ❑ Effectiveness
  - ✓ The support a user receives for making high-quality decisions
  - ✓ Help the customer discover his or her preferences
  - ✓ Help users make better decisions

The goals for providing explanations

- ❑ Efficiency
  - ✓ Reduce the decision-making effort
  - ✓ Reduce the time needed for decision making
  - ✓ Another measure might also be the perceived cognitive effort
- ❑ Satisfaction
  - ✓ Improve the overall satisfaction stemming from the use of a recommender system
- ❑ Relevance
  - ✓ Additional information may be required in conversational recommenders
  - ✓ Explanations can be provided to justify why additional information is needed from the user

The goals for providing explanations

- ❑ Comprehensibility
  - ✓ Recommenders can never be sure about the knowledge of their users
  - ✓ Support the user by relating the user's known concepts to the concepts employed by the recommender
- ❑ Education
  - ✓ Educate users to help them better understand the product domain
  - ✓ Deep knowledge about the domain helps customers rethink their preferences and evaluate the pros and cons of different solutions
  - ✓ Eventually, as customers become more informed, they are able to make wiser purchasing decisions
  - ✓ .....

References

Zhongqi Lu, (2016) Collaborative Evolution for User Profiling in Recommender Systems, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Telmo André Moreira Cardoso, (2022) Sistema de recomendação para uma loja online de livros, Dissertação de mestrado,

Diego Sánchez Moreno, (202), Improving collaborative filtering music recommender systems: A focus on user characterization from behavioral and contextual factors, Doctoral Thesis.

Pradeep Kumar Singh Kamla Nehru, (2021) Institute ofRecommender Systems: An Overview, Research Trends, and Future Directions, Article in International Journal of Business and Systems Research · January 2021

Joel P. Lucas, Constantino Martins, A hybrid recommendation approach for a tourism system

Vikas Kumar, Recommender System, pptx

Recommender Systems Based Rajaraman and Ullman: Mining Massive Data Sets & Francesco Ricci et al. Recommender Systems Handbook

Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich, Recommender Systems, Cambridge University Press

Radek Pelanek Recommender Systems, pptx

Drew Culbert, Recommender Systems, pptx

Mark Levene, Recommender Systems & Collaborative Filtering, pptx

Jure Leskovec, Anand Rajaraman, Jeff Ullman, Recommender Systems: Content-based Systems & Collaborative Filtering