“Para o prazer e para ser feliz, é que é preciso a gente saber tudo, formar alma, na consciência; para penar, não se carece.”

(Guimarães Rosa *in* **Grande Sertão: Veredas**,1956)

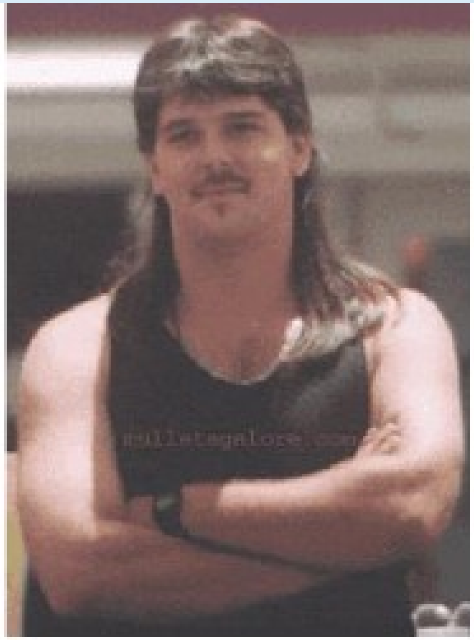
Introd. Inteligência Artificial

Roteiro da aula:

- Sistemas de recomendação;
- Modelo de recomendação por conteúdo;
- Exemplos;
- Extensões;

Com slides adaptados de J. Leskovec (Stanford)

Recomendações como resultado de predições



- **Customer X**
  - Buys Metallica CD
  - Buys Megadeth CD

- **Customer Y**
  - Clicks on Metallica album
  - Recommender system suggests Megadeth from data collected about customer **X**

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

- **Non-personalized recommendations:**
  - Editorial and hand curated
    - List of favorites
    - List of "essential" items
  - Simple aggregates
    - Top 10, Most Popular, Recent Uploads

- **Personalized recommendations:**
  - Tailored to individual users
  - Examples: Amazon, Netflix, Youtube,...

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

- ***X*** = set of **Customers**
- ***S*** = set of **Items**

- **Utility function** ***u***: ***X*** × ***S*** → ***R***
  - ***R*** = set of ratings
  - ***R*** is a totally ordered set
  - e.g., **1-5** stars, real number in **[0,1]**

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Utility Matrix

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Example 9.1 : In Fig. 9.1 we see an example utility matrix, representing users' ratings of movies on a 1–5 scale, with 5 the highest rating. Blanks represent the situation where the user has not rated the movie. The movie names are HP1, HP2, and HP3 for *Harry Potter* I, II, and III, TW for *Twilight*, and SW1, SW2, and SW3 for *Star Wars* episodes 1, 2, and 3. The users are represented by capital letters A through D.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Figure 9.1: A utility matrix representing ratings of movies on a 1–5 scale

slides/imagens MMD *Leskovec, Rajaraman & Ullman)

Recomendações como resultado de predições

Key Problems

- (1) Gathering "known" ratings for matrix
  - How to collect the data in the utility matrix
- (2) Extrapolating unknown ratings from the known ones
  - Mainly interested in high unknown ratings
    - We are not interested in knowing what you don't like but what you like
- (3) Evaluating extrapolation methods
  - How to measure success/performance of recommendation methods

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

(1) Gathering Ratings

- **Explicit**
  - Ask people to rate items
  - Doesn't work well in practice – people don't like being bothered
  - Crowdsourcing: Pay people to label items
- **Implicit**
  - Learn ratings from user actions
    - E.g., purchase implies high rating
  - What about low ratings?

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

(2) Extrapolating Utilities

- **Key problem:** Utility matrix ***U*** is **sparse**
  - Most people have not rated most items
  - **Cold start:**
    - New items have no ratings
    - New users have no history
- **Three approaches to recommender systems:**
  - **1)** Content-based
  - **2)** Collaborative filtering
  - **3)** Latent factor based

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Content-based Recommendations

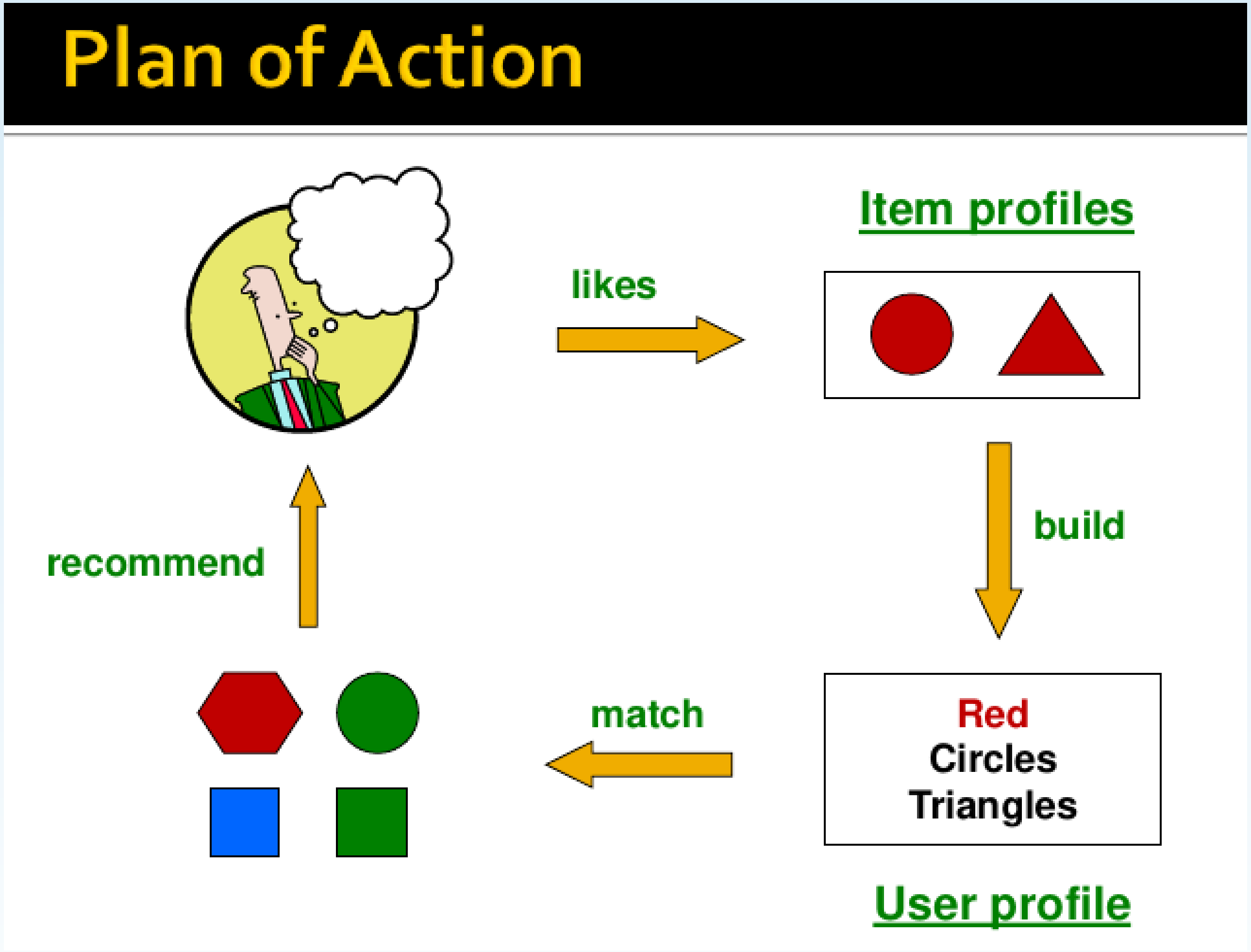
- **Main idea:**
  - Items have profiles:
    - Video -> [genre, director, actors, plot, release year]
    - News -> [set of keywords]
  - Recommend items to customer ***x*** similar to previous items rated highly by ***x***

Example:

- **Movie recommendations**
  - Recommend movies with same actor(s), director, genre, …
- **Websites, blogs, news**
  - Recommend other sites with “similar” content

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições



slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Item Profiles

- For each item, create an **item profile**
- **Profile is a set (vector) of features**
  - **Movies:** author, title, actor, director,…
  - **Text:** Set of "important" words in document
- **How to pick important features?**
  - Usual heuristic from text mining is **TF-IDF** (Term frequency * Inverse Doc Frequency)
    - **Term … Feature**
    - **Document … Item**

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Sidenote: TF-IDF

f_{ij} = frequency of term (feature) i in doc (item) j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

Note: we normalize TF to discount for "longer" documents

n_i = number of docs that mention term i

N = total number of docs

$$IDF_i = \log \frac{N}{n_i}$$

TF-IDF score: $w_{ij} = TF_{ij} \times IDF_i$

Doc profile = set of words with highest **TF-IDF** scores, together with their scores

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

User Profiles and Prediction

- **User profile possibilities:**
  - Weighted average of rated item profiles
  - **Variation:** weight by difference from average rating for item
- **Prediction heuristic: Cosine similarity of user and item profiles**
  - Given user profile ***x*** and item profile ***i***, estimate $u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Pros: Content-based Approach

- **+: No need for data on other users**
  - No item cold-start problem, no sparsity problem
- **+: Able to recommend to users with unique tastes**
- **+: Able to recommend new & unpopular items**
  - No first-rater problem
- **+: Able to provide explanations**
  - Can provide explanations of recommended items by listing content-features that caused an item to be recommended

slides/imagens J. Leskovec (Stanford)

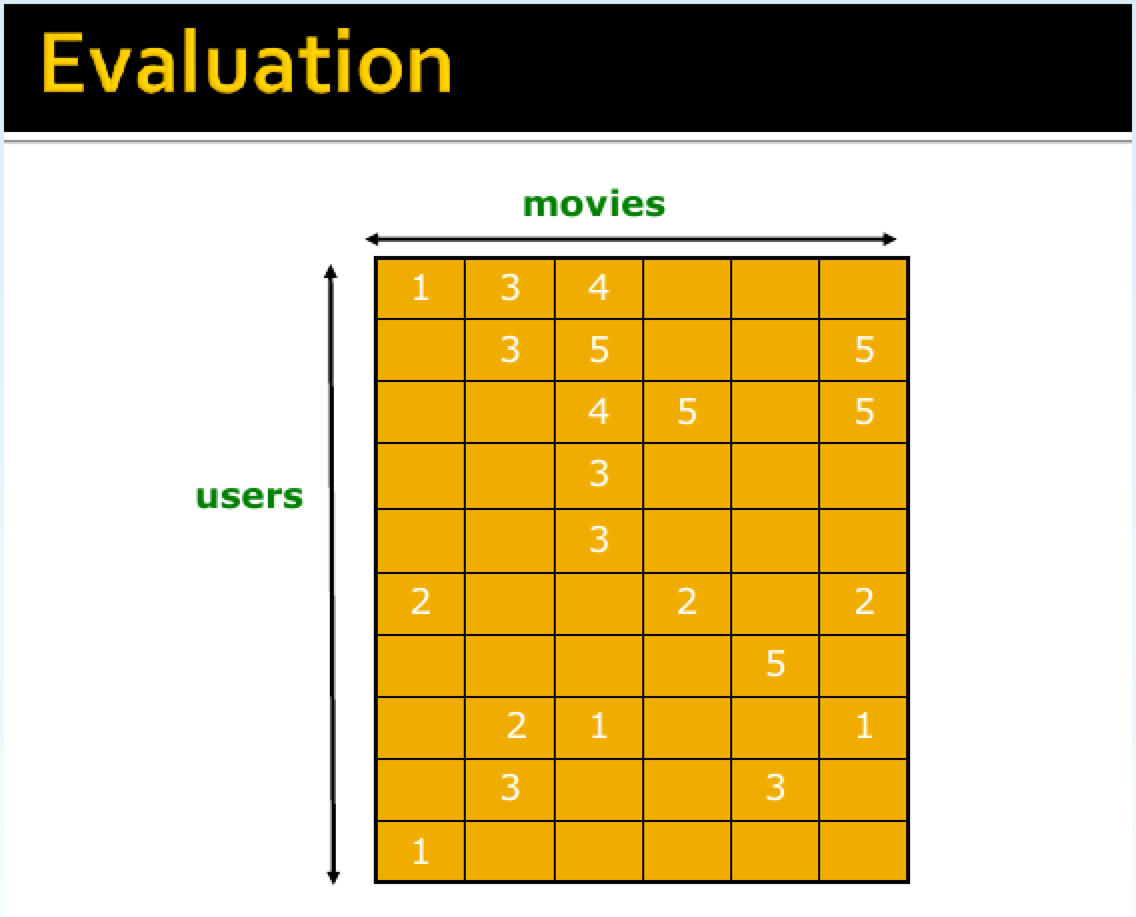
Recomendações como resultado de predições

Cons: Content-based Approach

- **–: Finding the appropriate features is hard**
  - E.g., images, movies, music
- **–: Recommendations for new users**
  - **How to build a user profile?**
- **–: Overspecialization**
  - Never recommends items outside user's content profile
  - People might have multiple interests
  - **Unable to exploit quality judgments of other users**

slides/imagens J. Leskovec (Stanford)

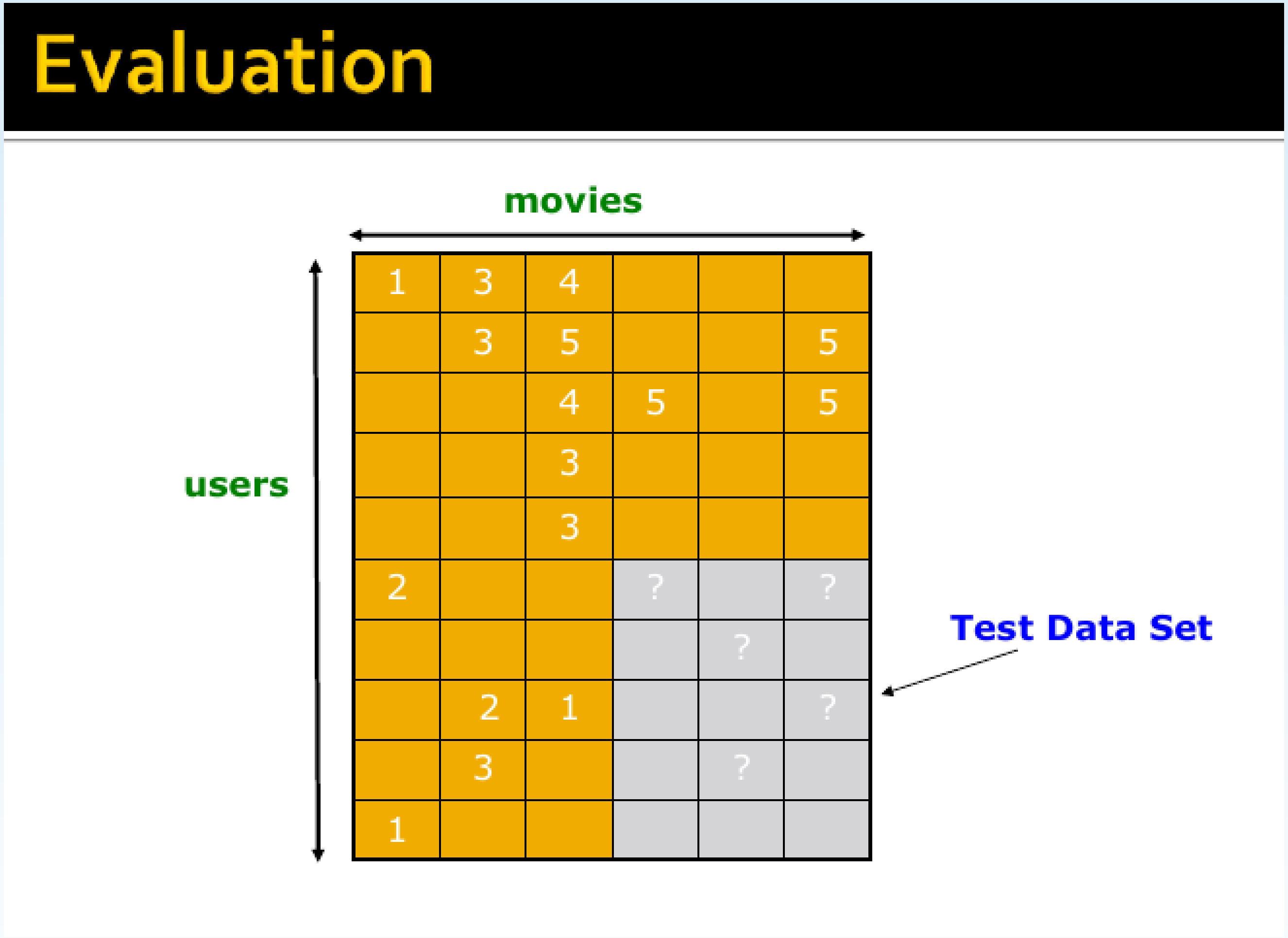
Recomendações como resultado de predições



1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			2		2
				5	
	2	1			1
	3			3	
1					

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições



slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Evaluating Predictions

- Compare predictions with known ratings
  - **Root-mean-square error** (RMSE)
    - $\sqrt{\frac{1}{N}\sum_{xi}(r_{xi} - r^*_{xi})^2}$ where $\boldsymbol{r}_{xi}$ is predicted, $\boldsymbol{r}^*_{xi}$ is the true rating of $\boldsymbol{x}$ on $\boldsymbol{i}$
      - ***N is the number of points we are making comparisons on***
  - **Precision at top 10**:
    - % of relevant items in top 10
- Another approach: 0/1 model
  - **Coverage:**
    - Number of items/users for which the system can make predictions
  - **Precision:**
    - Accuracy of predictions
  - **Receiver operating characteristic** (ROC)
    - Tradeoff curve between false positives and false negatives

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Problems with Error Measures

- Narrow focus on accuracy sometimes misses the point
  - Prediction Diversity
  - Prediction Context
  - Order of predictions
- In practice, we care only to predict high ratings:
  - RMSE might penalize a method that does well for high ratings and badly for others

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Collaborative Filtering: Complexity

- Expensive step is finding ***k*** most similar customers: **O(|X|)**
- **Too expensive to do at runtime**
  - Could pre-compute
- Naïve pre-computation takes time **O(k ·|X|)**
    - X … set of customers
- **We already know how to do this!**
  - Near-neighbor search in high dimensions (**LSH**)
  - Clustering
  - Dimensionality reduction

slides/imagens J. Leskovec (Stanford)

Recomendações como resultado de predições

Tip: Add Data

- **Leverage all the data**
  - Don't try to reduce data size in an effort to make fancy algorithms work
  - Simple methods on large data do best
- **Add more data**
  - e.g., add IMDB data on genres
- **More data beats better algorithms**

http://anand.typepad.com/datawocky/2008/03/more-data-usual.html

slides/imagens J. Leskovec (Stanford)

Exemplo de sistema para avaliar

https://www.kaggle.com/code/rushiekarteekchalla/amazon-product-recommendation-system

Referências Bibliográficas

- Leskovec, J. Rajaraman, A. & Ullman, J. **Mining of Massive Datasets**, Cambridge University Press, 3rd ed., 2012.