Intro to Recommender Systems

Davide Frey
davide.frey@inria.fr

Images and slides by Romaric Gaudel (Univ Rennes), and Paolo Cremonesi (PoliMi)
Some Images and Theorems from "Recommender Handbook"

Course Objectives

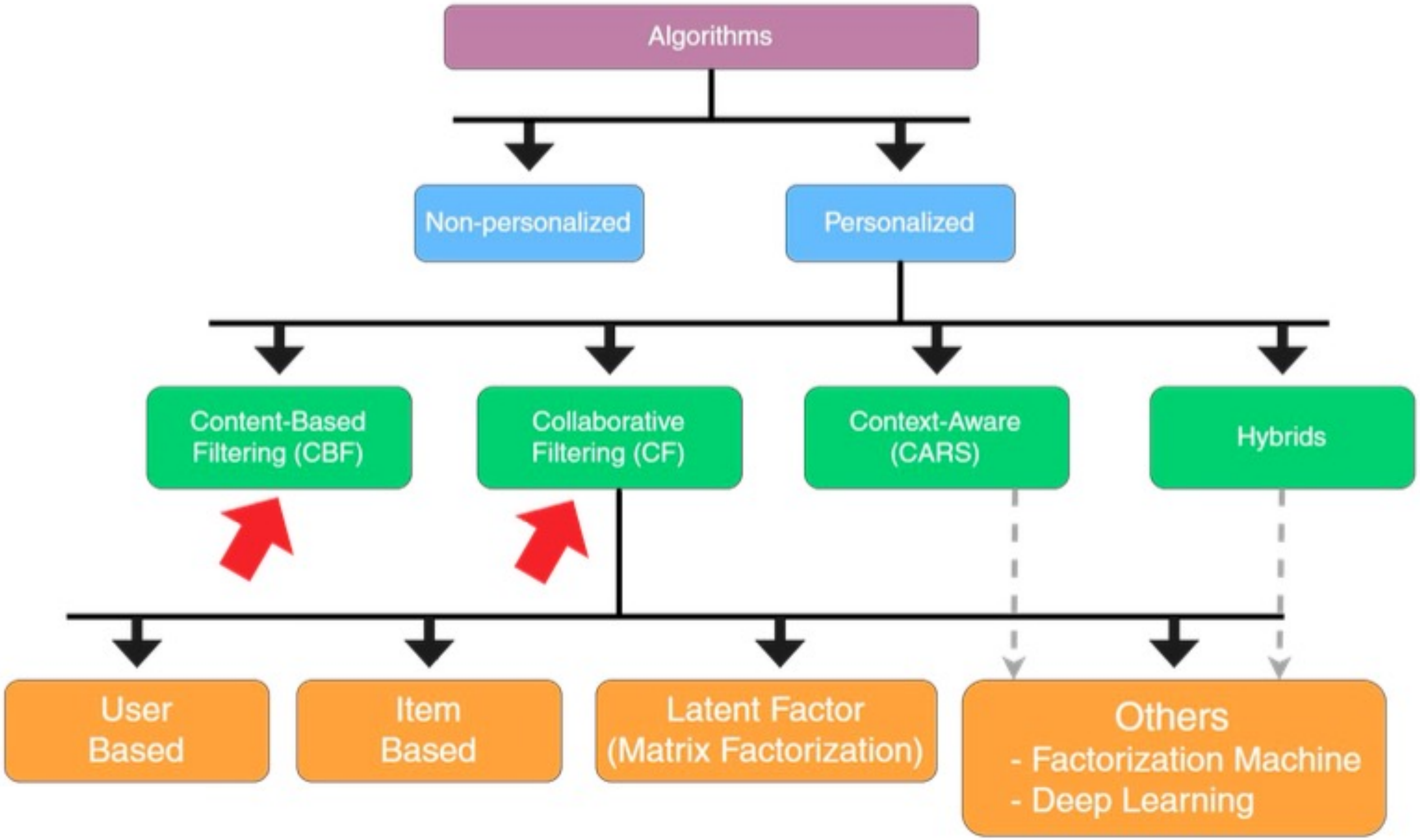
- **Gain an understanding of Recommendation Systems (RecSys)**
- **Master some useful tools**
- **Get the main idea how to build a RecSys**
- **See how wide this area can be**

Agenda

- What are recommendation Systems?
  - Examples

- Content-Based Recommenders
- Collaborative Filtering
- Recommendation Matching Real-Life Problems

Recommender System Taxonomy



Content-Based Recommenders

- Basic Approach to Recommendation
- Compare Items based on Attributes

User that liked an item is likely to like similar items

Item-Content Matrix (ICM)

- m x n
  - m number of items
  - N number of attributes

  - ICM(i,j) = 1
    if item i has attribute j

$$\begin{array}{cc} & \begin{array}{ccccc} & & \text{attribute} & & \end{array} \\ \begin{array}{c} \\ \\ \text{item} \\ \\ \\ \end{array} & \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \end{array}$$

Using the ICM

- Measure similarity between Items
- How?
  - Dot Product?

Dot product: $\vec{i} \cdot \vec{j} = |\mathbf{i}||j| \cos \vartheta = \sum_{i=1}^{n} \vec{i}_i \vec{j}_i$

  - Cosine Similarity (normalized dot product)

$$similarity = \cos \theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Similarity shrinking

- Only take into account most similar items
- Shrink term h

$$similarity = \frac{A \cdot B}{\|A\| \cdot \|B\| + h}$$

Estimate Ratings

- Estimate rating user would give to items
- Weighted average of previous ratings
- Use similarity as weight

$$rating(u,i) = \frac{\Sigma_j\, rating(u,j) * similarity(i,j)}{\Sigma_j\, similarity(i,j)}$$

Similarity Matrix

- n x n square matrix (n items in dataset)
- Precomputation of item similarity
- Dense matrix
- In practice only keep k-most similar items for each row (k-nearest neighbor)

$$\begin{array}{cc} & \textit{items} \\ \textit{items} & \begin{bmatrix} \mathbf{0.21} & \cdots & \mathbf{0.8} \\ \vdots & \ddots & \vdots \\ \mathbf{1} & \cdots & \mathbf{0.01} \end{bmatrix} \end{array}$$

Similarity Matrix - kNN

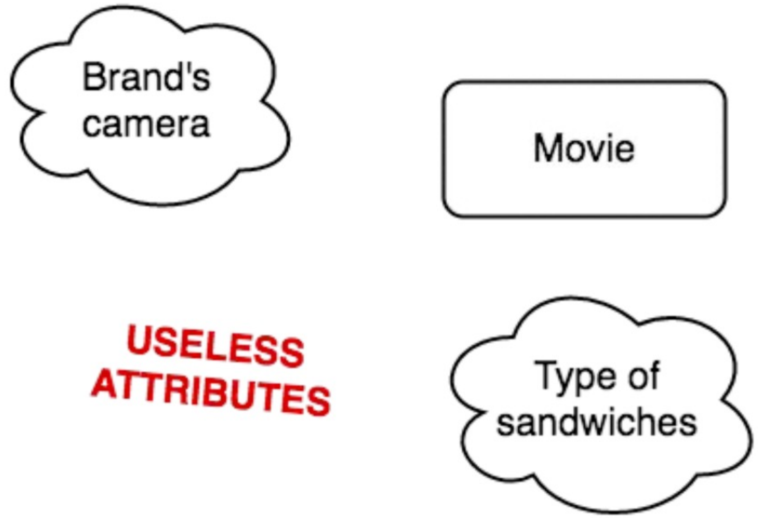
- Compare new item to its kNN

$$rating(u,i) = \frac{\Sigma_j \, rating(u,j) * similarity(i,j)}{\Sigma_j \, similarity(i,j)}$$

Average over the k most similar items j.

Weighting Attributes

- Attributes are not all equally important
- Add weight to most relevant attributes
- Small weight to useless attributes



Term-Frequency
Inverse Document Frequency

- TF-IDF
  - Automatically compute weights of natural-language attibutes

$$\text{TF-IDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log \frac{N}{n_k}}_{\text{IDF}}$$

- N documents in corpus
- n_k number of documents in the collection in which the term t_k occurs at least once.

TF-IDF - Cosine Normalization

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TF-IDF}(t_s, d_j)^2}}$$

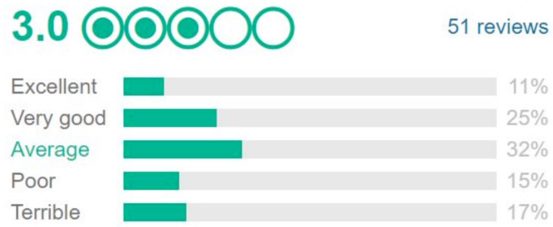
User-Based Content-Based Filtering

- Instead of computing similarity between items, compute it between users.

$$rating(u,i) = \frac{\Sigma_v\, rating(v,i) * similarity(u,v)}{\Sigma_v\, similarity(u,v)}$$

Ratings

- Key information for recommenders
  - Explicit
    - Likes
    - Star-scores
    - …



  - Implicit
    - Viewing time
    - Song streaming count
    - Purchases
    - Opened links
    - …

Example 1: Rule Mining

- Frequent Itemset Mining
- Data:
  - Background/train: logged user behavior
  - Target/query: an item (id)
- Model:
  - Rule: View A-> Buy B ; P(Buy B|View A)>.25
  - Rule Mining
  - Co-occurrence of items
    - Combinatorial Explosion
- Remarks:
  - Only requires logging user behavior
  - Need efficient algorithms

Example 2: Look-a-like Items

- Replacing an item
- Data:
  - Background/train: features of items (type, price, color, ...)
  - Target/query: an item (features)
- Model:
  - Similarity between items
    - Norm-2, cosine, dot-product, correlation
- Remarks:
  - Efficient: with approximate nearest-neighbor search
  - Requires features

Example 3: Sorted List of Web Pages

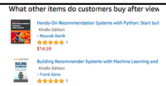
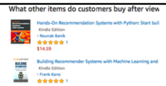
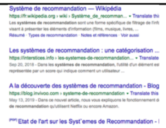
- Query: → sorted list of web-pages
- Data:
  - Background/Train: past recommendations and "feedback" (query, list of recommendations, click/no-click)
  - Target/query: current query (includes contextual information: hour, location. . .)
- Model
  - Binary classification: (query, web-page features, context) → click/no-click
  - N-binary classification problems (query, context) web page, features → click/no-click

Example 4: Personalized Recommendation

- Identify the best items for each user
- Data:
  - Background/train: past recommendations and "feedback" (idUser, idItem, rating)
  - Target/query: a user to serve (id)
- Learn/Train then Recommend

Characterizing Recommendation

- High-level Task: Choose K items among L >> K
- Diverse Approaches

	Objective	Available Information	Approach
	Basket completion	tickets / item id.	Data-Mining
	Replacement	items features / item features	content-based Recommender System
	Serve repeated requests	clicks history (at population level) / request/item features	content-based Recommender System
	Serve a user based only on its tastes	rates history per user / user id. and item id.	Collaborative Filtering

Recommendation vs Machine Learning

- Key Difference:
  - Ranking
    - Identify 10 best Items
    - Vs Identify 1000 relevant items
- ML for Recommendation
  - Several Models to Learn
  - Transfer Learning
  - Multitask Learning
  - Personalized Learning

Take-Home Message

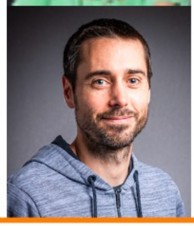
- Recommend
  - Choose k items among L >> k
  - Aim at top-k items or ranking on items
- Different Techniques depending on available data/application
- ML for Recommendation
  - Learning to Rank
  - Transfer Learning
  - Multi-Task Learning
  - Personalized Learning

Inria

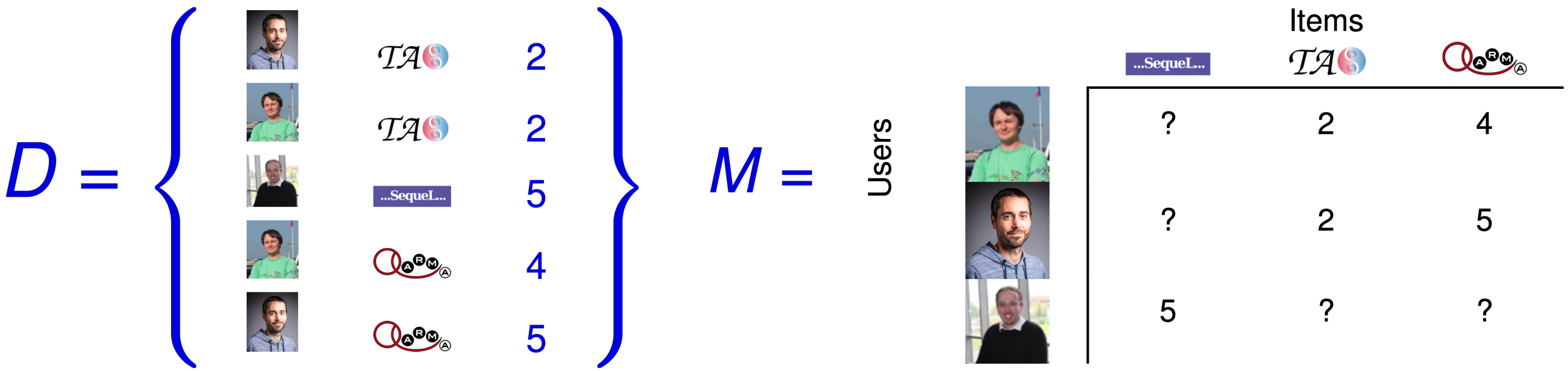
Collaborative Filtering

Items

Users

	...SequeL...	TAO	ARMA
	4	2	4
	6	2	5
	5	2	4

From Data to Models



Mathematical Objective: Matrix Completion

$$M = \underset{X}{\operatorname{argmin}}\ d(X,M)$$

Where X is low-rank

Algorithm 0: Most Popular

- Algorithm:
  - Recommend the item with the highest average rating
- Remarks:
  - Not really personalized
  - No matrix completion / Only basic matrix completion
  - Never recommend niche items

Which item will be recommended to the third user?

...Sequel...	TAO		...
?	2	4	2
?	2	5	?
5	?	?	2
?	4	2	3
?	3	?	1

Algorithm 1: k Nearest Neighbor

- Prediction for a pair (user,item) =
  - Weighted average of scores of the k most similar users

$$\hat{M}_{i,j} = \frac{\sum_{\ell=1}^{k} w_{i,(\ell)} M_{(\ell),j}}{\sum_{\ell=1}^{k} |w_{i,(\ell)}|}$$

  - Weight / similarity $w_{i,j}$: cosine of ratings (on items rated by both users)

$$CV(u,v) = \cos(\mathbf{x}_u, \mathbf{x}_v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}},$$

Algorithm 1: k Nearest Neighbor

Which score is associated to the forth item for the third user, with $k = 2$ and using cosine similarity?

...SequeL...	TA		. . .
?	2	5	2
1	2	5	?
?	2	5	?
1	3	4	4
?	4	2	3

Several Variants

- Different Similarities
  - Cosine

$$CV(u,v) = \cos(\mathbf{x}_u, \mathbf{x}_v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum\limits_{i \in \mathcal{I}_u} r_{ui}^2 \sum\limits_{j \in \mathcal{I}_v} r_{vj}^2}},$$

  - Pearson Correlation

$$\mathrm{PC}(u,v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum\limits_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

  - Weighted Average
    of deviations from mean

$$\hat{\boldsymbol{M}}_{i,j} = \hat{\mu}_i + \frac{\sum_{\ell=1}^{k} w_{i,(\ell)} (M_{(\ell),j} - \hat{\mu}_{(\ell)})}{\sum_{\ell=1}^{k} |w_{i,(\ell)}|}$$

Item-based Exercise

Which score is associated to the forth item for the third user, with $k = 2$ and using cosine similarity?

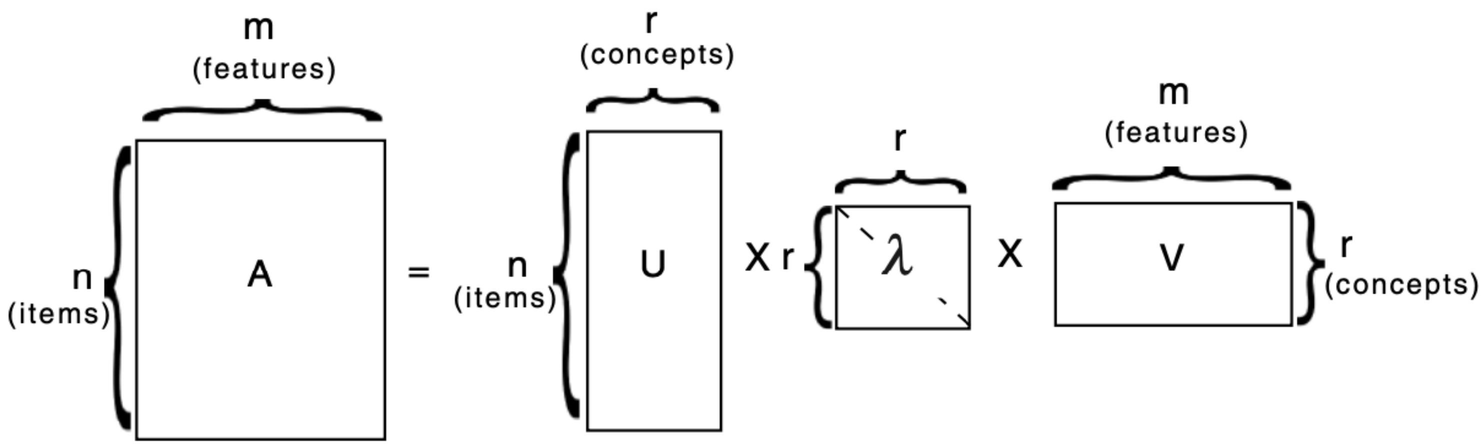
...SequeL...	TAO	ARMA	...
?	2	5	2
1	2	5	?
?	2	5	?
1	3	4	4
?	4	2	3

User-based vs Item based

- User based
  - See each user as a vector
  - Compute similarity between users
- Item based
  - See each item as a vector
  - Compute similarity between items

Algorithm 2: SVD

- Find a lower dimensional feature space
  - New features represent concepts
  - Strength of each concept in collection is computable
- Key theorem
  - Always possible to decompose a matrix



Algorithm 2: SVD

$$\widehat{M} = \underset{rang(X)=k}{\text{argmin}} \|X - M\|^2$$

- With unknown entries in M filled with 0

Theorem: Eckart Young

Let A be a matrix and $A = PSQ^T$ its singular value decomposition.
The solution of the optimization problem

$$\underset{rang(B)=k}{\text{argmin}} \|B - A\|^2$$

is the matrix $B = PS_kQ^T$, where S_k derives from S by keeping the k highest values (other are set to 0)

Algorithm 2: SVD

$$\widehat{M} = \underset{rang(X)=k}{\text{argmin}} \|X - M\|^2$$

- Algorithm
  - Compute the singular value decomposition of $M = PSQ^T$
  - Return $$\hat{M} = PS_kQ^T$$

  where S_k is derived from S by keeping the k highest values and setting others to 0