Advanced Data Mining

Piotr Lipiński

Recommender Systems

- **general recommendations** **(static)**
  - **manual (editorial) recommendations**
  - **recommendations based on simple statistics**
    - TOP10, most popular, recent uploads

- **general recommendations** **(interactive)**
  - **recommendations based on advanced statistics**
    - complementary/opposed/similar products
    - association rules

Recommender Systems

- **personalized recommendations**
  - **recommendations based on advanced statistics**
    - complementary/opposed/similar products
    - association rules
  - **content-based recommendations**
  - **model-based recommendations**
  - **collaborative filtering**
- **other approaches**
  - **cold start problems** often force **hybrid approaches**
  - **Factorization Machines, etc.**

General recommendations

- **general recommendations** **(static)**
  - **manual (editorial) recommendations**
    - weaknesses: low scalability, low accuracy, low coverage
    - advantages: do not require much data
  - **recommendations based on simple statistics**
    - weaknesses: low accuracy, low diversity, low coverage
    - advantages: do not require much data
    - improvements: statistics in categories

- **general recommendations** **(interactive)**
  - weaknesses: low accuracy, better diversity, better coverage
  - advantages: do not require much data

Personalized recommendations

- **personalized recommendations**
  - **content-based recommendations**
    - requires product feeds and a definition of the product profile
    - requires some user data and a definition of the user profile
    - requires metrics between user profiles and product profiles
  - **model-based recommendations**
    - requires a classifier for each user / group of users
    - large computational requirements
  - **collaborative filtering**

Content-based recommendations

- **content-based recommendations**
  - requires product feeds and a definition of the product profile
  - requires some user data and a definition of the user profile
  - requires metrics between user profiles and product profiles
- **example:**
  - product = movie, product features = genres:
    - LOTR = [action=0, adventure=1, comedy=0, fantasy = 1, historical=0, romance=0]
  - user features = interest in genres (see next slides for details)
  - metrics = cosine measure between the user profile and the product profile

	Ac	Ad	Co	Fa	Hi	Ro
I_1	0	1	0	1	0	0
I_2	0	0	0	0	1	1
I_3	0	0	1	0	0	1

	Ac	Ad	Co	Fa	Hi	Ro
U_1	0.3	0.2	0.0	0.1	0.1	0.0
U_2	0.0	0.4	0.3	0.0	0.0	0.1
U_3	0.0	0.0	0.5	0.5	0.0	0.1
U_4	0.4	0.4	0.0	0.0	0.5	0.2
U_5	0.0	0.5	0.3	0.0	0.4	0.8

Content-based recommendations

- **content-based recommendations**
  - REQUIREMENT1: the product profile is usually defined on the basis of the product features
    - sometimes the number of features is large and feature selection techniques are necessary
  - REQUIREMENT2: the user profile is usually defined on the basis of the user interests
    - how to detect the interests of the user?
  - REQUIREMENT3: the metrics is usually the cosine measure

- **explicit receiving user interests is often not reliable**

- **implicit detecting user interests is not easy**
  - requires some data on user activities
  - requires some studies on how user activities corresponds to user interests
  - e.g. the TF-IDF approach may be applied to evaluate the interests in the particular features

TF-IDF

- **Term-Frequency (TF) matrix:**

$$TF[i, j] = M[i, j] / sum(\{M[k, j] : k = 1, 2, \ldots, d\})$$

- **Inverse-Document-Frequency (IDF) vector:**

$$IDF[i] = \log (N / |\{j : M[i, j] > 0\}|)$$

- **TF-IDF matrix:**

$$TF\text{-}IDF[i, j] = TF[i, j]\ IDF[i]$$

where M[i, j] is the number of occurences of the term i in the document j, d is the number of terms, and N is the number of documents.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Collaborative Filtering

- Input data: matrix R of ratings

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	3				5	
U_2		4	3			1
U_3				5		
U_4	4	4			5	2
U_5		5	3		4	

- Output data: utility U(u, i) of the item i for the user u

Collaborative Filtering

- **Collaborative Filtering**

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	3				5	
U_2		4	3			1
U_3				5		
U_4	4	4			5	2
U_5		5	3		4	

- For each user u, find the most similar k users u_1, u_2, …, u_k and let

$$U(u, i) = (U(u_1, i) + U(u_2, i) + \ldots + U(u_k, i)) / k$$

- Extensions: normalization with means, with standard deviations, with baselines, etc.

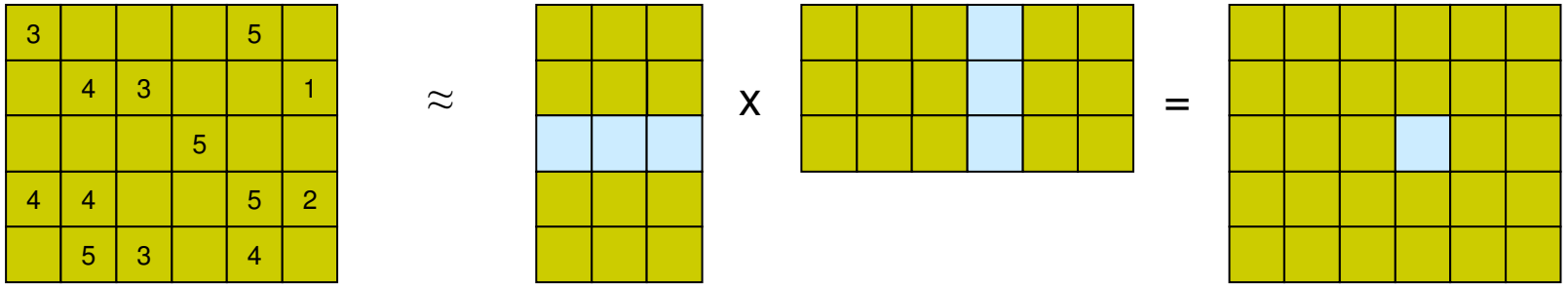
Collaborative Filtering

- **Matrix Factorization**

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	3				5	
U_2		4	3			1
U_3				5		
U_4	4	4			5	2
U_5		5	3		4	

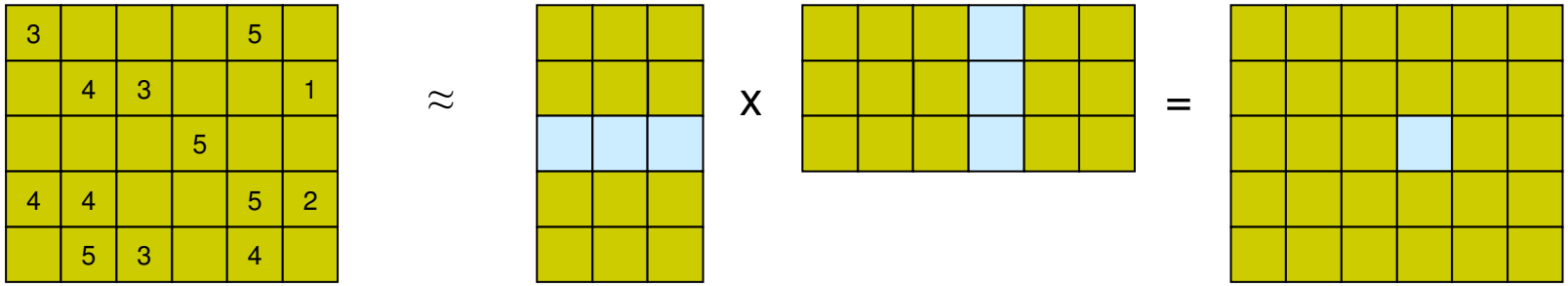
- Find two matrices A and B, such that $R \approx A \times B$, then

$$U(u, i) = a_u \times b_i$$



Collaborative Filtering

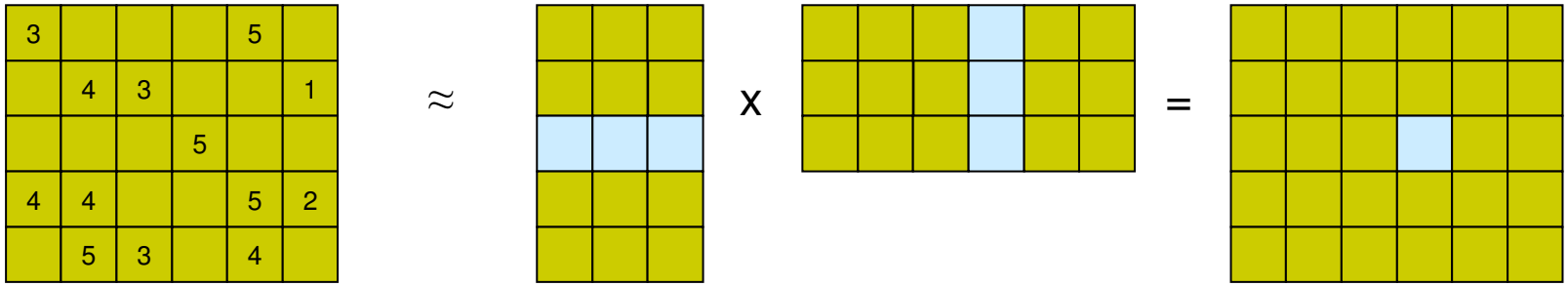
- **Matrix Factorization**



  - Algorithms: SVD, SVD++, NMF, PMF, etc.

Collaborative Filtering

- **Matrix Factorization**



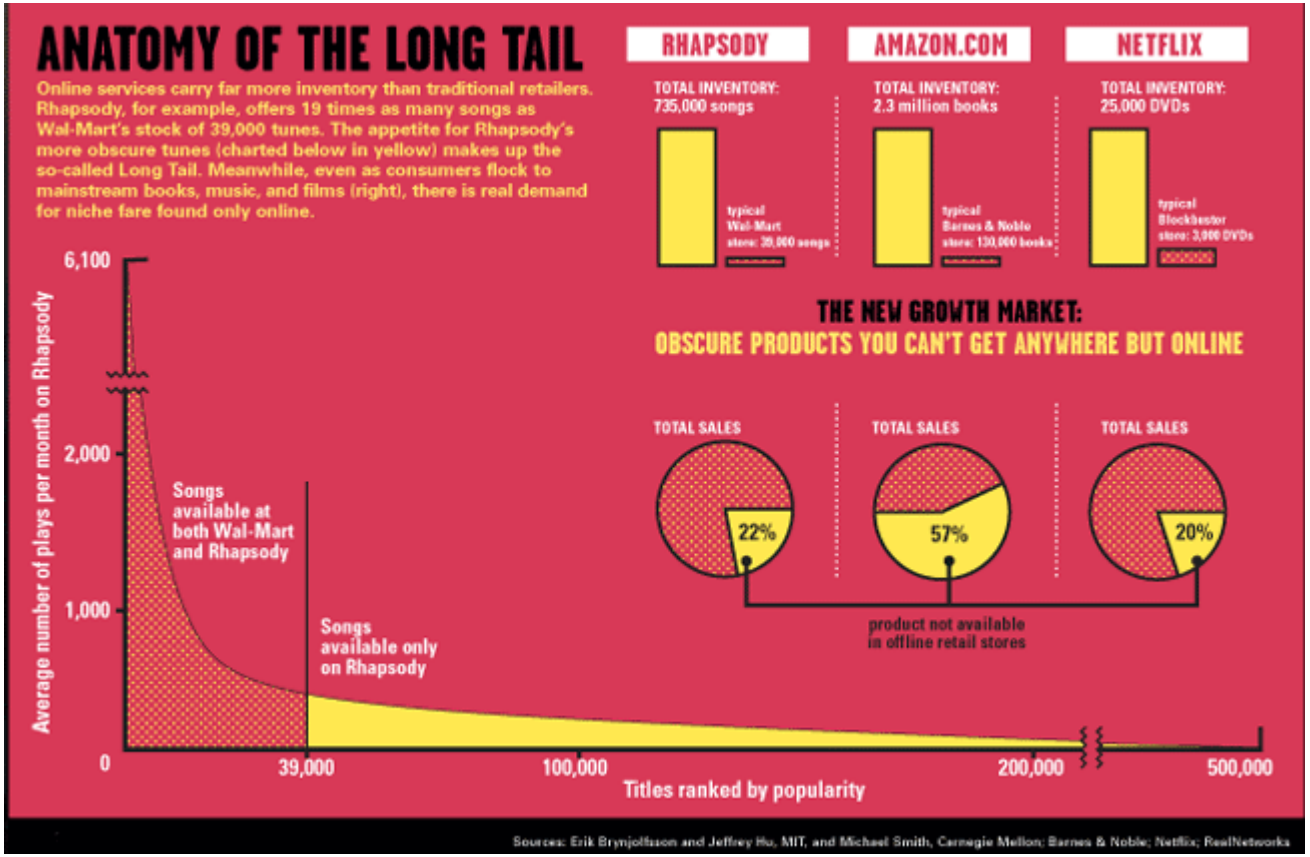
- Funk Matrix Factorization (SVD): $U(u, i) = a_u \times b_i$

$$|| R - R' || + \alpha ||A|| + \beta ||B||$$

- SVD++: $U(u, i) = mean + bias_u + bias_i + a_u \times b_i$

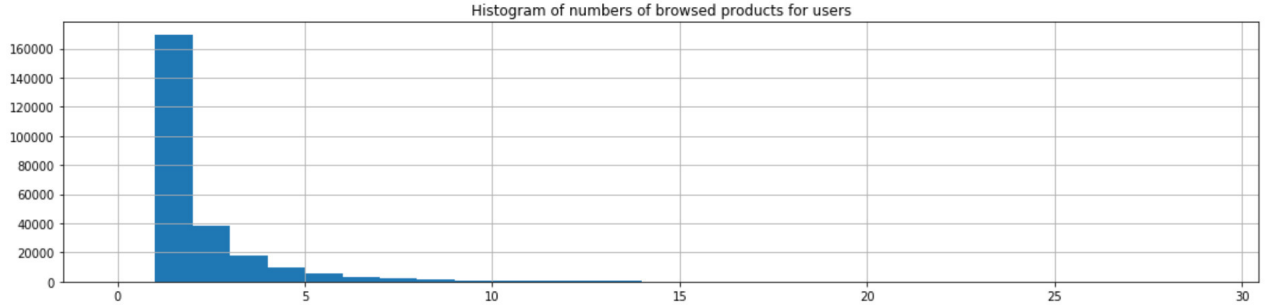
Long-tailed / Heavy-tailed Data

- **Long-tailed data**
  - some years ago: **Pareto 80/20 principle**
    - e.g. *80% of income is generated by 20% of customers*
  - currently: **long-tailed data** (Chris Anderson, 2004)



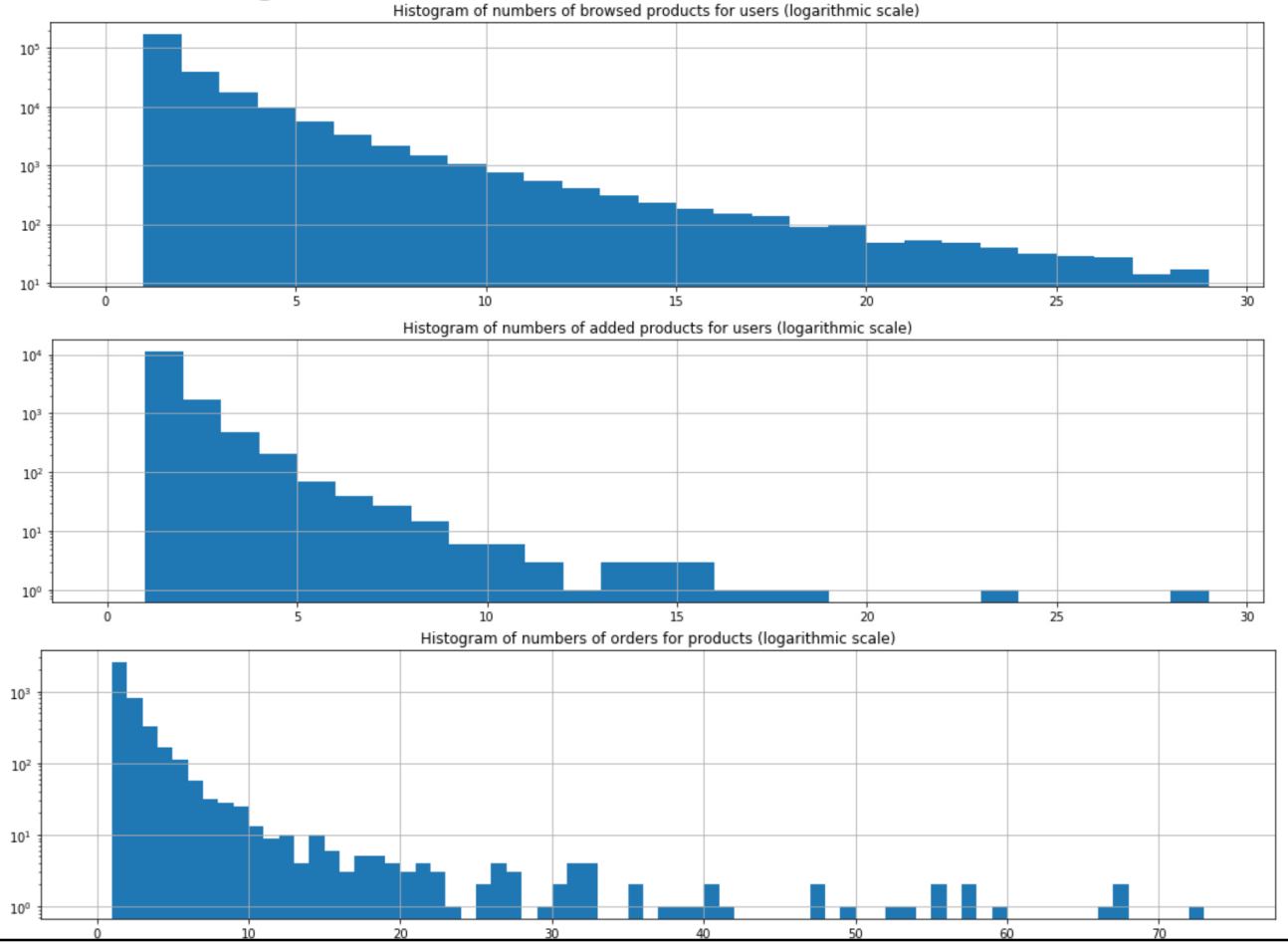
Long-tailed / Heavy-tailed Data

- **Long-tailed data**
  - some years ago: **Pareto 80/20 principle**
    - e.g. *80% of income is generated by 20% of customers*
  - currently: **long-tailed data** (Chris Anderson, 2004)



Long-tailed / Heavy-tailed Data

- **Long-tailed data**
  - some years ago: **Pareto 80/20 principle**
    - e.g. *80% of income is generated by 20% of customers*
  - currently: **long-tailed data** (Chris Anderson, 2004)

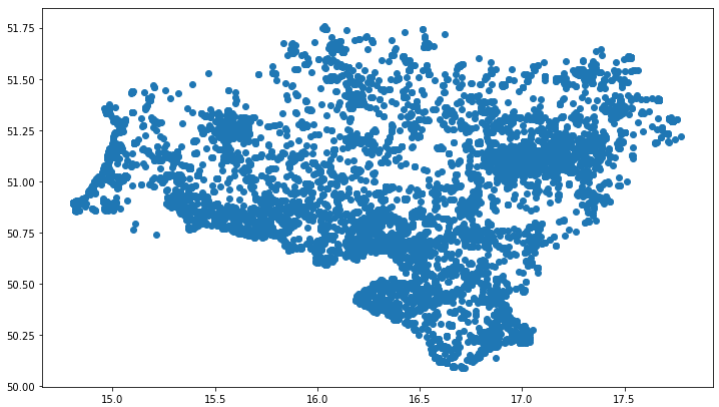


Personalized recommendation

- Input data: additional data

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

<Product>
<Product>
<Product>
<Product>
<Product>
<Product_ID>8100</Product_ID>
<SKU>WKS-6016</SKU>
<Name>Uptown Girl Blouse</Name>
<Product_URL>https://www.domain.com/product/wks-6016</Product_URL>
<Price>58</Price>
<Retail_Price>89.95</Retail_Price>
<Thumbnail_URL>https://www.domain.com/images/wks-6016_600x600.png</Thumbnail_URL>
<Description>Sociosqu facilisis duis ...</Description>
<Category>Clothing>Tops>Blouses</Category>
<Category_ID>285</Category_ID>
<Brand>Entity Apparel</Brand>
<Child_SKU>WKS-6016-CORD-MD|WKS-6016-KEGR-MD</Child_SKU>
<Child_Price></Child_Price>
<Color>Coral Red|Kelly Green</Color>
<Color_Family>Red|Green</Color_Family>
<Size>Medium</Size>
<Shoe_Size></Shoe_Size>
<Pants_Size></Pants_Size>
<Season>Summer|Spring</Season>
<Badges>Exclusive</Badges>
duct>



Goals

- **Off-line evaluation measures:**
  - rating prediction accuracy
    - MSE, MAE, MAPE, etc.
  - user preference
    - users may irrationally prefer some approaches to others
  - usage prediction accuracy
    - precision/recall@N, TPR, FPR, ROC, AUC, etc.
  - ranking measures
  - coverage measures
    - item space coverage
    - user space coverage
  - diversity measures
  - novelty / serendipity / adaptivity / etc.
- **On-line evaluation measures:**
  - A/B tests and above approaches

Recommender Systems

- Additional issues:
  - Explicit/Implicit Feedback
  - Cold Start – new users, new products
  - Personalization
  - Context
  - Volatility over time
  - Noise (e.g. special periods – Christmas, holidays, …)