Introduction to recommender systems

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino

Politecnico
di Torino

Lecture goal

- Recommender systems fundamentals
- Content-based recommendation
- Collaborative filtering
- Hybrid methods

Objective

Recommender systems try to identify the need and preferences of users, filter the huge collection of data accordingly and present the best suited option before the users by using some well-defined mechanism

A literature review and classification of recommender systems research. Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. https://doi.org/10.1016/j.eswa.2012.02.038

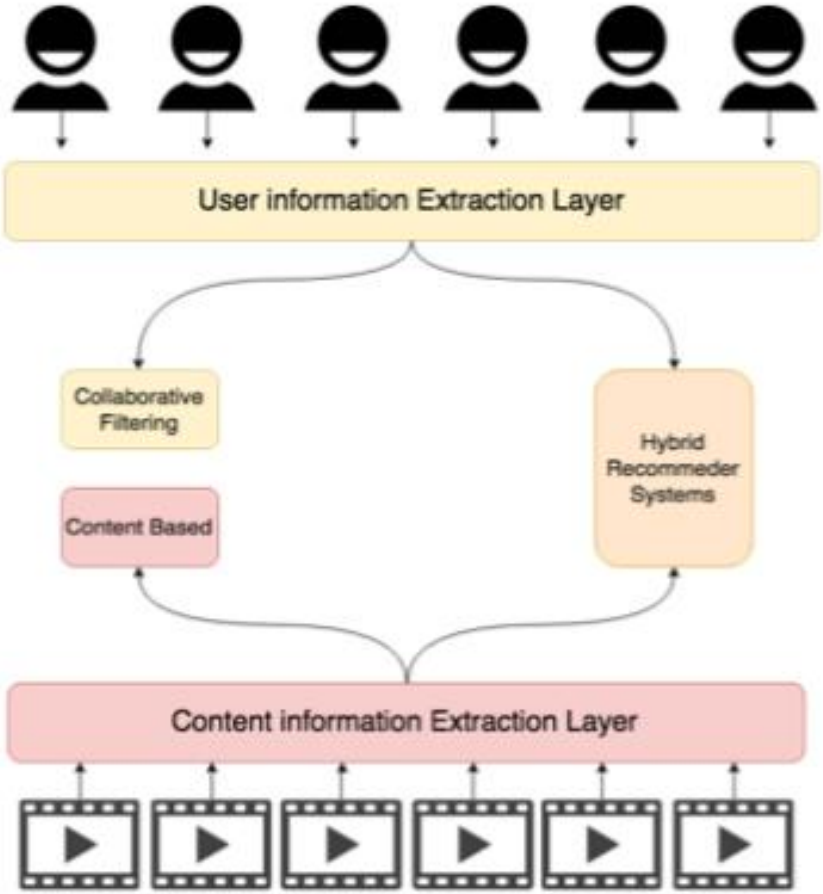
Objective

- Predict user responses or preferences
- Examples of application contexts
  - E-commerce
    - recommend the item to buy
  - On demand services
    - recommend the movie to watch
  - Content curation
    - recommend the news articles to read
  - Tourism and hospitality
    - recommend the places to visit and the hotels to book

System categorization

- Content-based systems
- Collaborative filtering
- Hybrid approaches

Framework



Content-based recommender systems

- Recommendations are based on the characteristics of the items recommended
- Recommend an item to a user similar to those that she/he previously rated highly
  - E.g., if a user frequently watches horror movies then it recommends a movie of that category

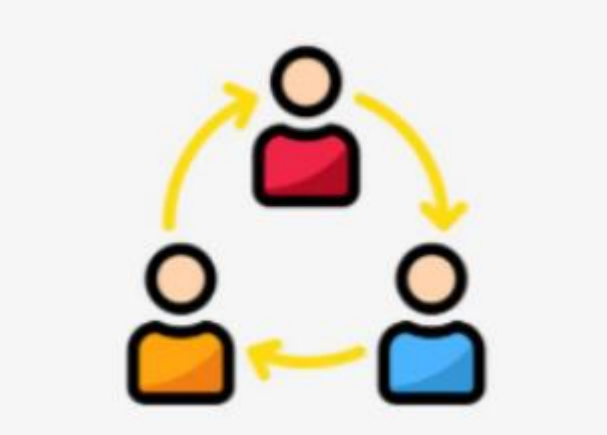
Content-based recommender systems: key idea

Content types

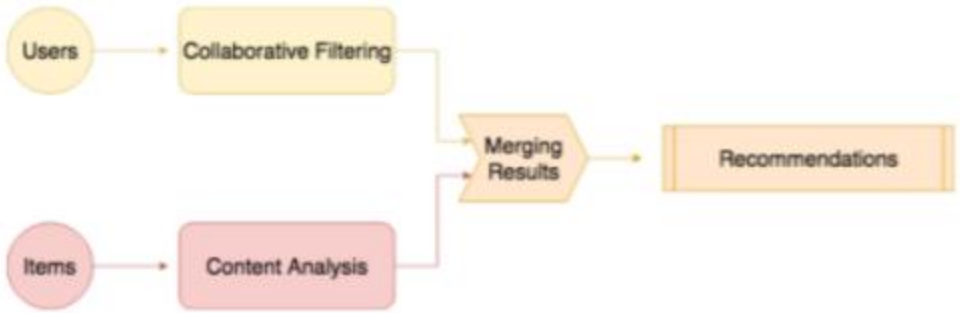
- Products
- Locations
- Images
- Tags
- Videos
- Teaching materials
- Etc.

Collaborative filtering

- Recommendations rely on user similarities
- If two users are similar then they are likely to prefer similar items
- Recommend to a user the items that are most popular for similar users
  - E.g., if you are a football player and similar users like Messi videos then it recommends Messi videos to you



Hybrid systems: key idea



Hybrid systems

- Combine content-based with collaborative filtering approaches
- Possible solutions:
  - **Ensemble methods**: Build separate methods and then combine the predictions
  - **Nested methods**: Integrate either content-based characteristics into a collaborative approach or collaborative characteristics into a content-based approach
  - **Mixed strategies**: Build a general model that integrates both content-based and collaborative characteristics

Utility matrix

- Rows: users
  - A, B, C, D
- Columns: items
  - 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
- Values: ratings
  - from 1 (very low) to 5 (very high)

Utility matrix

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

Example taken from 'Mining of Massive Datasets' by Jure Leskovec, Anand Rajaraman, and Jeff Ullman. http://www.mmds.org/

Utility matrix

	Item 1	Item 2	Item 3	Item 4
User 1	0	4	5	5
User 2	1	5	5	-
User 3	5	2	1	1
User 4	4	2	1	-

Scores: 0 - 5

Utility matrix

	Item 1	Item 2	Item 3	Item 4
User 1	0	4	5	**5**
User 2	1	5	5	**4**
User 3	5	2	1	**1**
User 4	4	2	1	**1**

Scores: 0 - 5

Feedback types

- Feedback can be classified as
  - Explicit: explicitly provided by the users
  - Implicit: deducted from users' behaviors

Utility matrix population

- Collecting utility matrix values is necessary
- How to get item rates?
  1. User annotation: ask users to rate items
     - Complete
     - Partial (some users are unwilling to provide ratings)
  2. Rate inference (implicit feedback)
     - deduce item ratings from user actions (e.g., time to leave, click rate, Auditel scores)
     - Infer item rankings using Machine Learning models (independently of the recommender system)

Problem statement

- U: set of users
- I: set of items
- R: ordered set of ratings
- F(·): $U \times I \rightarrow R$

Problem statement

- Given U, I, R, and the known known values for F(·)
- Predict the unknown values for F(·)
- For each user u ∈ U the recommender system returns the list of top-K items with highest rate
  - K is an input parameter

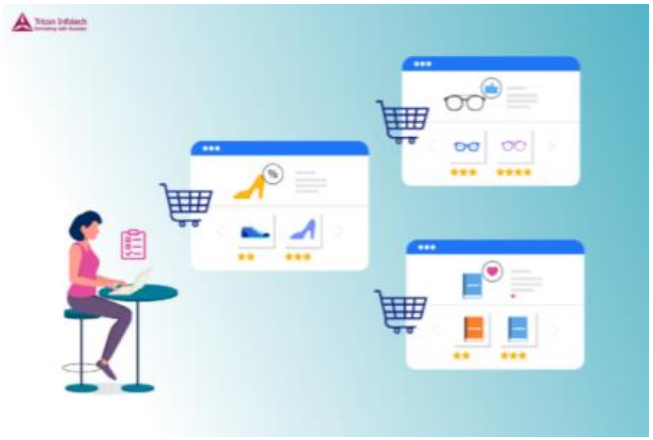


Image source: https://medium.com/@triconinfotech/how-recommender-systemsinfluence-customer-behavior-679d5d6288c1

Problem statement

- Predict cases when rating $r \in R$ is very high or very low is typically more interesting
  - E.g. predict the items that a user really likes is relevant to plan marketing strategies
  - E.g. predict the movie categories that a user dislikes avoids service drop outs

Main challenges

- Deal with sparse utility matrices
- Cold start
- First rater
- Popularity bias

Sparsity

- Many user and item combinations are missing
- Hard to build unified item and user profiles

First rater

- Recommend items with unknown ratings
- Items that have never been rated cannot be recommended
- Need for ad hoc solutions

Cold start problem

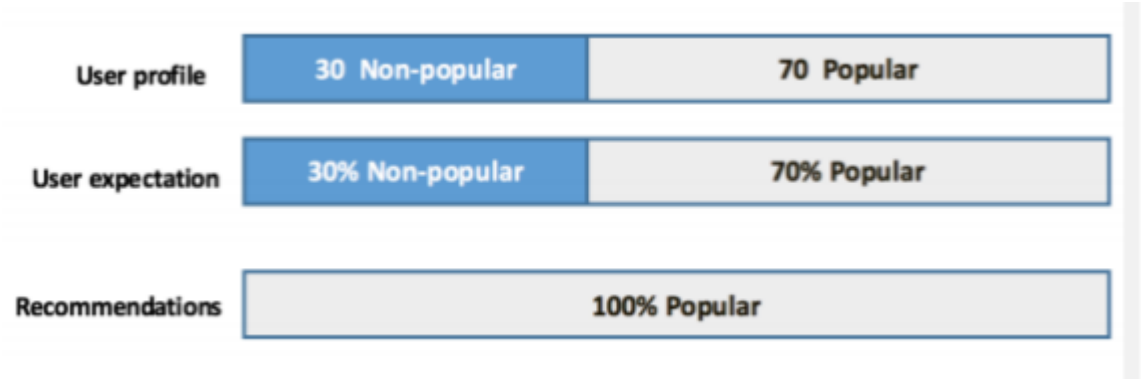
- Recommend items to new users
  - User ratings F(·) are not available
  - User preferences are unknown

Filter bubble

- Content-based filtering could be biased
  - It recommends only the items that are strictly related to the firstly evaluated items

Popularity bias

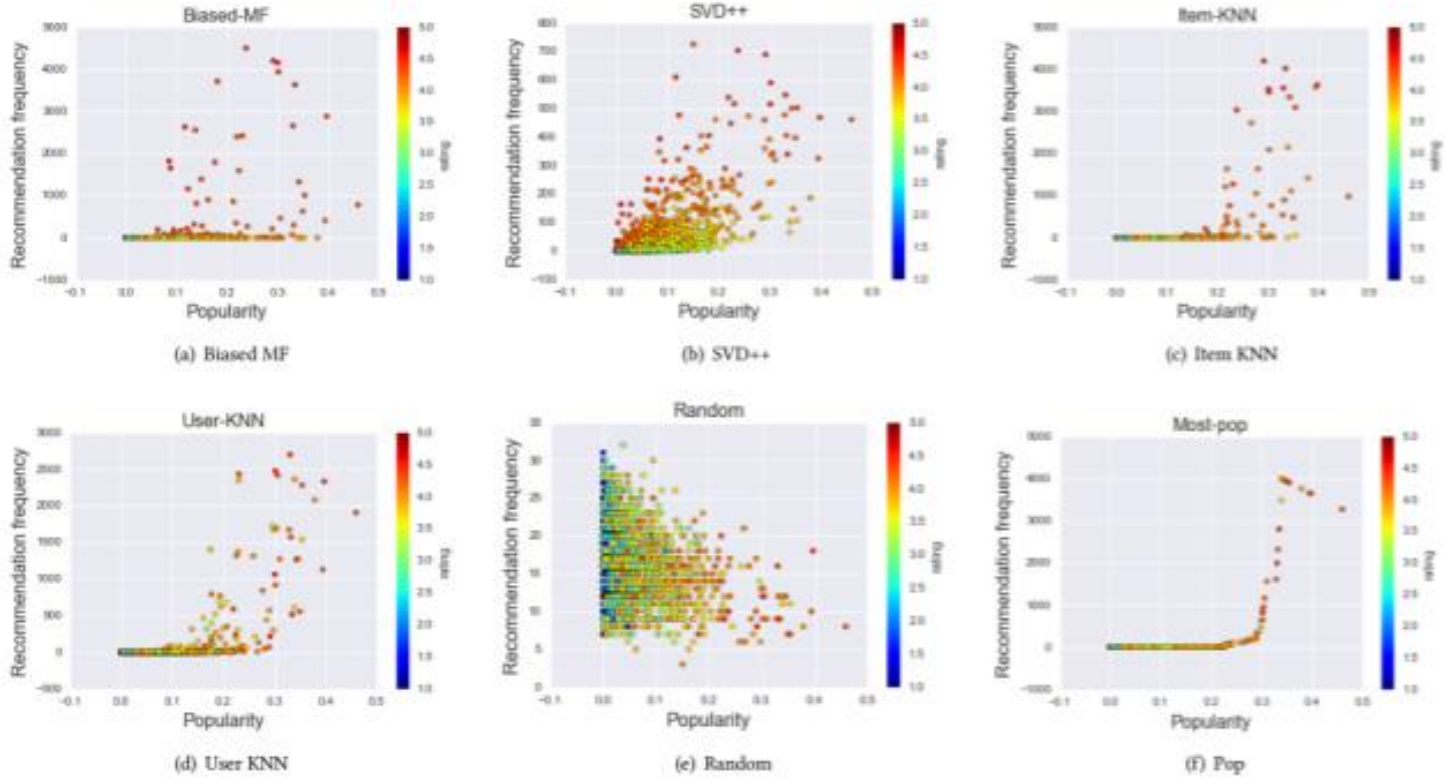
- Frequently rated items get over-exposure
- Rarely rated items get under-exposure



The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. Dominik Kowald, Markus Schedl, Elisabeth Lex. ECJR 2019

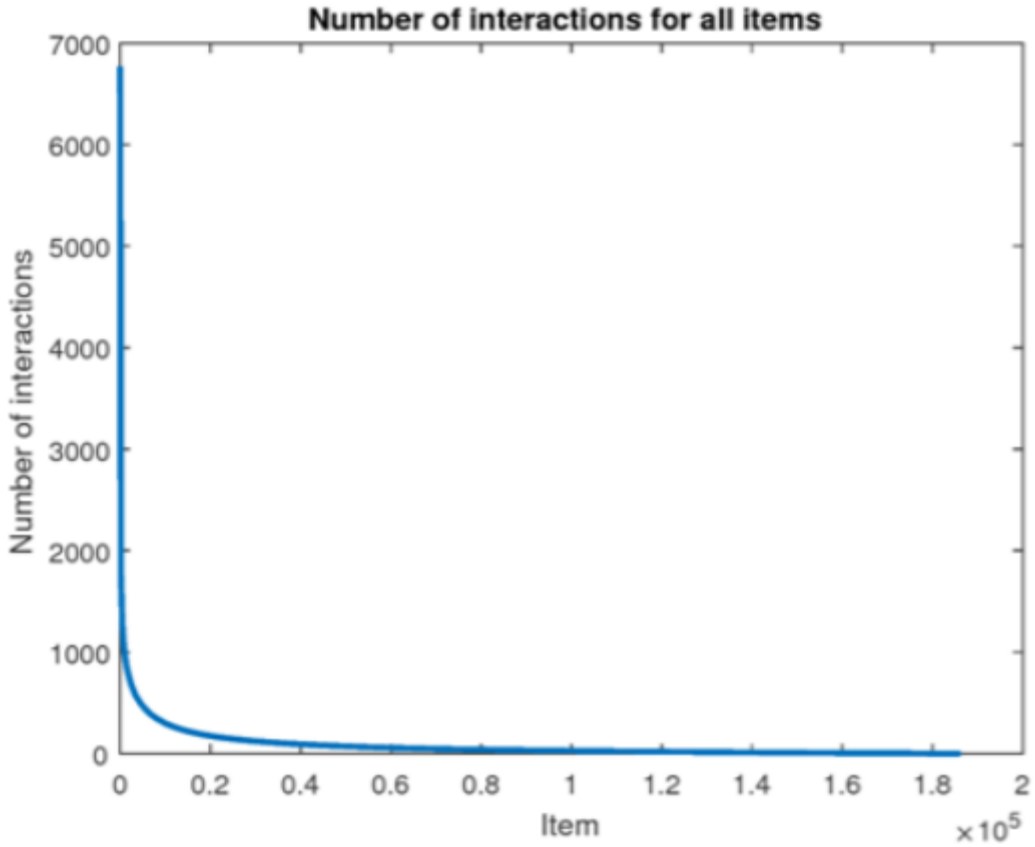
Popularity bias

- Frequently rated items get over-exposure
- Rarely rated items get under-exposure



The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. Dominik Kowald, Markus Schedl, Elisabeth Lex. ECJR 2019

The long-tail phenomenon



Source: https://en.wikipedia.org/wiki/Cold_start_(recommender_systems)

The long-tail phenomenon

- Item popularity is rather variable
- It is not possible to tailor the offer to each single customer
- This motivates the need for effective recommender systems

Description of the main system categories

- **Content-based systems**
- Collaborative filtering

Content-based recommender systems

- Item profile $\mathbf{V}_i$: vector representation of an item $i \in I$
- Approach
  - Define a set of item features
  - Compute the item profile of each candidate item V_{ci}
  - Given a user $u \in U$ pick the top-N highly rated items i^u_{1st}, i^u_{2nd}, ..., $i^u_{n\text{-}th}$
  - Compute the item profiles of the highly rated items
  - Compute pairwise item similarities
  - Recommend the most similar candidate items

$$\text{sim}(\mathbf{V}_{i^u_k}, \mathbf{V}_{j^u_k})$$

Content-based recommender systems

- Example: movie recommendation
- Possible item features
  - Genre
  - Production year
  - Main character name
  - Director name
  - Parental control on/of

Content-based recommender systems

- Example: document recommendation
- Item profile: tf-idf scores for each word in the documents
- $n_{i,j}$: number of occurrences of word i in document j
- d_j : number of words in document j
- |D|: number of documents in the collection

$$tf - idf_{i,j} = tf_{i,j} \cdot idf_i$$

$$tf_{i,j} = \frac{n_{i,j}}{d_j} \qquad idf_i = \log_{10} \frac{|D|}{|\{d : i \in d\}|}$$

Vector similarities

- Euclidean distance $$||\mathbf{A} - \mathbf{B}||^2 = (\mathbf{A} - \mathbf{B})^T \ (\mathbf{A} - \mathbf{B})$$
- Cosine similarity $$\frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \ ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}}$$

$$Eucl(\mathbf{A}, \mathbf{B}) = ||\mathbf{A} - \mathbf{B}||^2 = (\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{B}) = 2 \ (1 - cos(\mathbf{A}, \mathbf{B}))$$

Jaccard similarity and dissimilarity

- a: number of dimensions that equal 1 for both vectors V_i and V_j
- b: number of dimensions that equal 0 for vector V_i but equal 1 for vector V_j
- c: number of dimensions that equal 1 for vector V_i but equal 0 for vector V_j
- d: number of dimensions that equal 0 for both vectors Vi and V_j

$$JS_{i,j} = \frac{a}{a+b+c} \qquad JD(V_i, V_j) = \frac{b+c}{a+b+c} = 1\text{-}JS_{i,j}$$

Description of the main system categories

- Content-based systems
- **Collaborative filtering**

Description of the main system categories

Key steps

1. Utility matrix population: Collect the ratings for both target and annotating users
2. Neighbor formation: Compute the similarity between target and annotating users
3. Selection and ranking of candidate items: Pick the items that are highly rated by similar users
4. Rate prediction: For each candidate item forecast the rate that would be given by the target user
5. Item ranking: Rank candidate items and return the top-N recommended items

Description of the main system categories

- Content-based systems
- **Collaborative filtering**

Collaborative filtering

Main approaches

1. **Memory-based approaches**: Exploit the ratings in the utility matrix to compute similarities between users/items
2. **Model-based approaches**: Exploit the estimate or learn a model and then apply it to forecast the user rating

Collaborative filtering

Main approaches

1. **User-based neighbor models:** assign to the target item a rate similar to those assigned by the most similar users
2. **Item-based neighbor models**: assign to the target item a rate similar to those assigned by other users to the most similar items

User- vs. item-based collaborative filtering

Yang, N.; Chen, L.; Yuan, Y. An Improved Collaborative Filtering Recommendation Algorithm Based on Retroactive Inhibition Theory. Appl. Sci. 2021, 11, 843.https://doi.org/10.3390/app11020843

User-based collaborative filtering

- s_{ux}: similarity of users u and x
- r_x: vector of the user x ratings
- N(u, i): set of K user that are most similar to u who rated item i

$$\mathcal{F}(u,i) = \frac{1}{K} \sum_{x \in N} \mathcal{F}(x,j) = \frac{\sum_{x \in N(u,i)} s_{ux} \cdot \mathcal{F}(x,i)}{\sum_{x \in N(u,i)} s_{ux}}$$

Item-based collaborative filtering

- Find the items that are most similar to the target item
- Estimate the rating for the target item based on the ratings for similar items

Item-based collaborative filtering

- s_{ux}: similarity of users i and j
- *F*(u,i): rating of user u on item i
- N(u, i): set of items rated by user u and similar to item i

$$\mathcal{F}(x,i) = \frac{\sum_{j \in N(x,i)} s_{ij} \cdot \mathcal{F}(x,j)}{\sum_{j \in N(x,i)} s_{ij}}$$

Model-based collaborative filtering

- Main approaches
  - Dimensionality reduction
  - Probabilistic models
  - Machine Learning models

Dimensionality reduction

- Goal
  - estimate blank entries in the utility matrix
- Approach
  - Singular Value Decomposition (SVD)
- Idea
  - reduce the utility matrix to capture the most important aspects of the data, ignore the rest

Singular Value Decomposition

Decomposition of matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$ of rank $r \leq min(N, M)$

$$\mathbf{U} = \left[\underbrace{u_1, u_2, \ldots, u_r}_{\mathbf{U}_r}, \underbrace{u_{r+1}, \ldots, u_N}_{\mathbf{U}_r} \right] = [\mathbf{U}_r | \mathbf{U}_r]$$

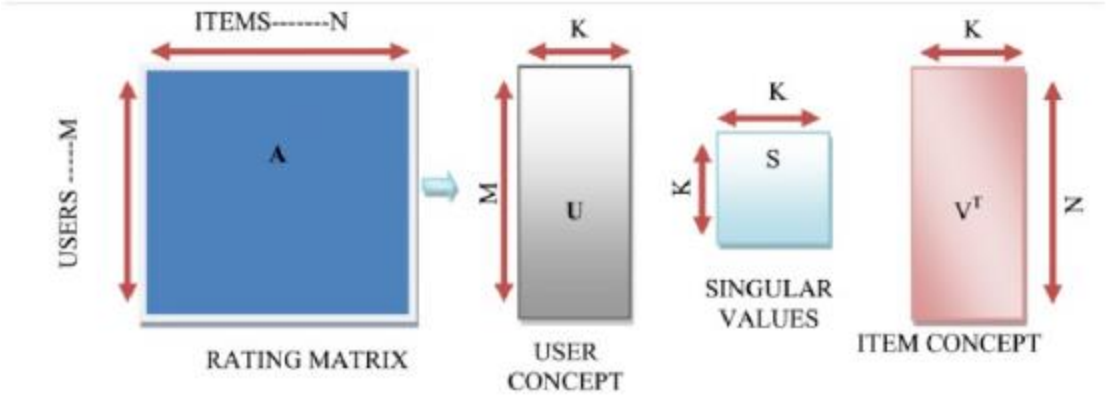
$$\mathbf{V} = \left[\underbrace{v_1, v_2, \ldots, v_r}_{\mathbf{V}_r}, \underbrace{v_{r+1}, \ldots, v_M}_{\mathbf{V}_r} \right] = [\mathbf{V}_r | \mathbf{V}_r]$$

$$A = [\mathbf{U}_r | \mathbf{U}_r] \begin{bmatrix} \boldsymbol{\Sigma}_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_r^T \end{bmatrix} = \mathbf{U}_r \boldsymbol{\Sigma}_r {\mathbf{V}_r}^T$$

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^{r} \sigma_i u_i u_i^T = u_1 v_1^T \sigma_1 + u_2 v_2^T \sigma_2 + u_3 v_3^T \sigma_3 + u_4 v_4^T \sigma_4 \\ &\quad + \ldots + u_r v_r^T \sigma_r, \quad rank\,(\mathbf{A}) = r \\ \mathbf{B} &= \sum_{i=1}^{k} \sigma_i u_i u_i^T = u_1 v_1^T \sigma_1 + u_2 v_2^T \sigma_2 + u_3 v_3^T \sigma_3 + u_4 v_4^T \sigma_4 \\ &\quad + \ldots + u_k v_k^T \sigma_k \quad rank\,(\mathbf{B}) = k \\ \mathbf{A} - \mathbf{B} &= \mathbf{U} \Sigma_{\mathbf{A}} \mathbf{V}^T - \mathbf{U} \Sigma_{\mathbf{B}} \mathbf{V}^T = \mathbf{U}^T \left[\Sigma_{\mathbf{A}} - \Sigma_{\mathbf{B}} \right] \mathbf{V}^T \rightarrow \|\mathbf{A} - \mathbf{B}\| = \sigma_{k+1} \end{aligned}$$

A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

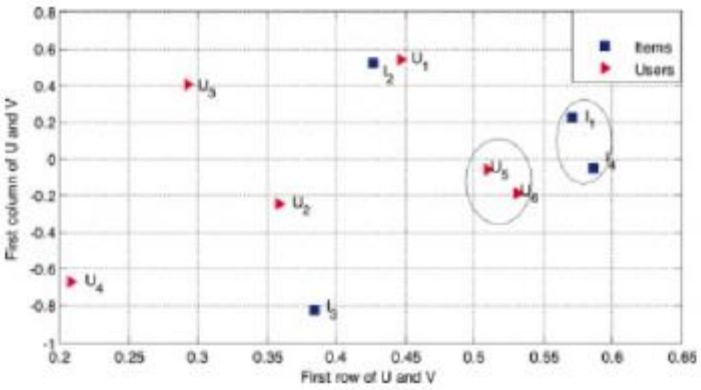
Singular Value Decomposition



A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

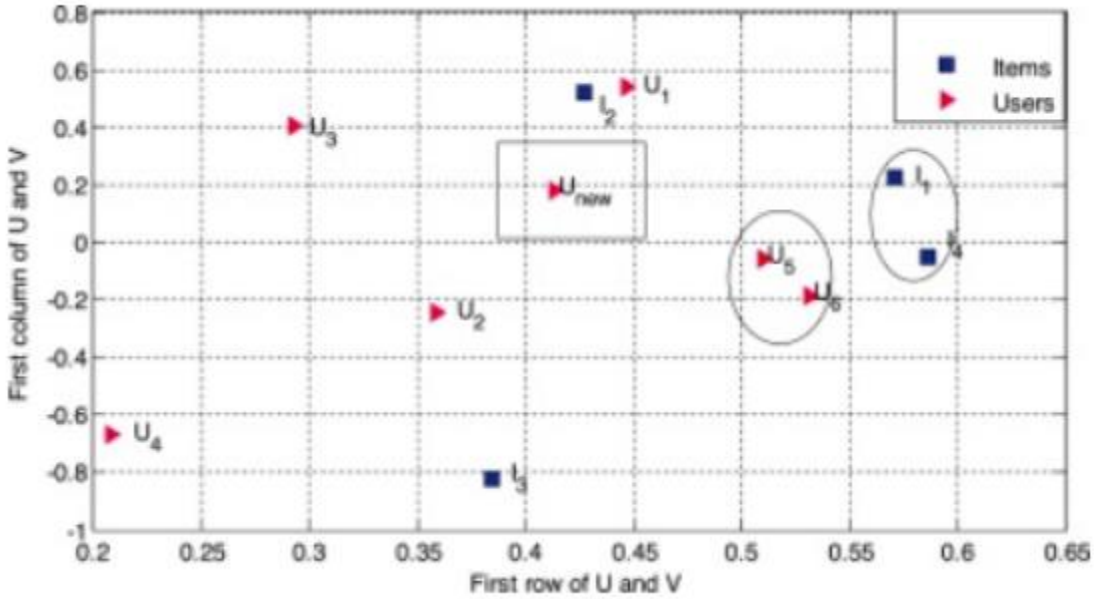
Singular Value Decomposition

	Item 1	Item 2	Item 3	Item 4
User 1	5	5	0	5
User 2	5	0	3	4
User 3	3	4	0	3
User 4	0	0	5	3
User 5	5	4	4	5
User 6	5	4	5	5



A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

Singular Value Decomposition



A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

SVD for recommendation

- Users 5 and 6 and items 1 and 4 are located very close to each other
- SVD for dimensionality reduction can be applied effectively to reveal the users that have similar taste and form neighbors for items and users
- SVD approximates the missing ratings based on the following matrix factorization

$$\hat{r} = \left(\mathbf{U}_k \mathbf{S}_k^{\frac{1}{2}}\right)_u \cdot \left(\mathbf{S}_k^{\frac{1}{2}} \mathbf{V}'_k\right)_i$$

A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

Probabilistic approaches: example

- Compute the probability that a user u ∈ U likes item i ∈ I
- Recommend the item with maximal likelihood

$$\arg_{i \in I} \max \mathrm{P}(\mathcal{F}(i)=HR|u)$$

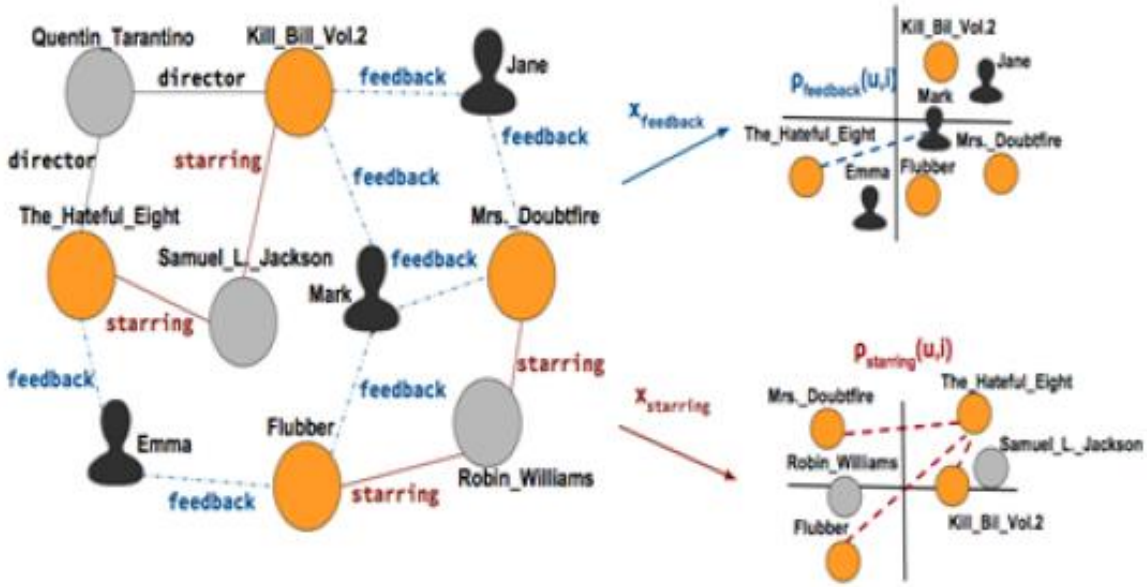
- Bayes theorem $\mathrm{P}(\mathcal{F}(i)=HR|u) = \frac{P(u|\mathcal{F}(i)=HR)P(\mathcal{F}(i)=HR)}{P(u)}$
  where *HR* is the highest rating
- P(u) is irrelevant
- P(F(i)=HR) is a count on the utility matrix
- under the Naive assumption that all user ratings are independent

$$\mathrm{P}(u|\mathcal{F}(i)=HR) \approx \prod_{r \in R} P(\mathcal{F}(i)=r|u)$$

A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. Expert Systems with Applications. 2018

Machine Learning-based approaches: example of graph-based hybrid method

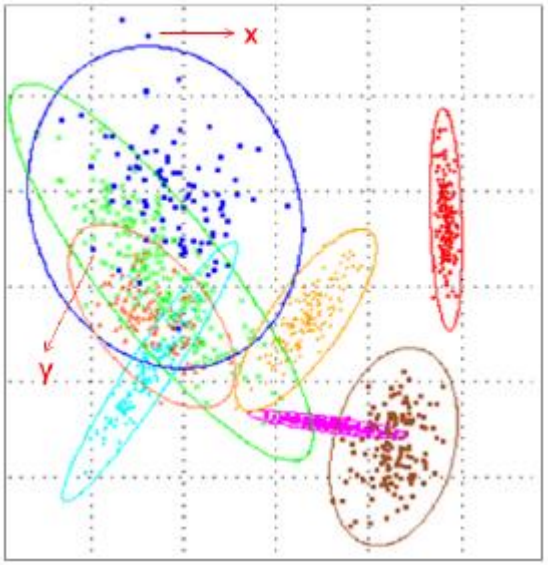
1. Represent user and content relationships in a graph
2. Enrich the model with semantic relationships (DBPedia https://wiki.dbpedia.org/)
3. Uses a random walk strategy to infer the item profiles



Enrico Palumbo, Giuseppe Rizzo, and Raphael Troncy. 2017.
Entity2rec: Learning User-Item Relatedness from Knowledge Graphs for Top-N Item Recommendation. ACM RecSys'17

Machine Learning-based approaches: example of clustering-based hybrid method

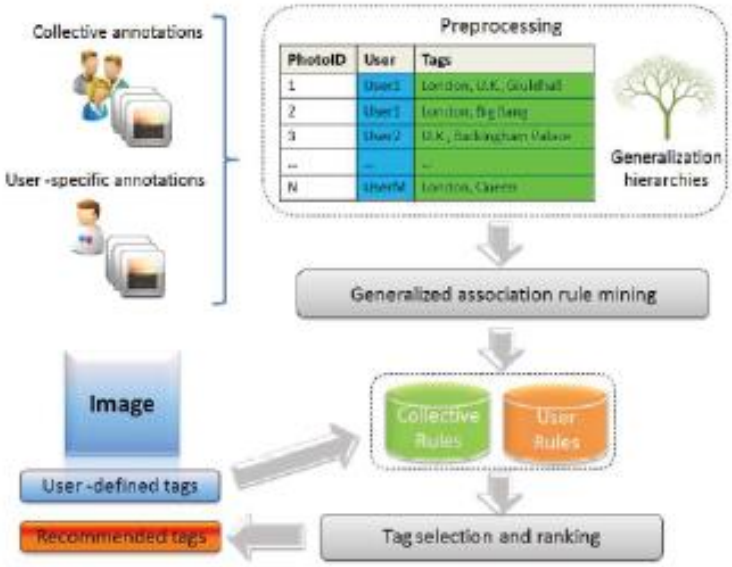
1. Apply knowledge expansion based on standard ontologies to generate item and user profiles
2. Cluster users and items separately using semantic similarity
3. Apply matrix factorization to predict the ratings



Mehrbakhsh Nilashi, Othman Ibrahim, Karamollah Bagherifard. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques, Expert Systems with Applications, 2018

Machine Learning-based approaches: example of association-based hybrid method

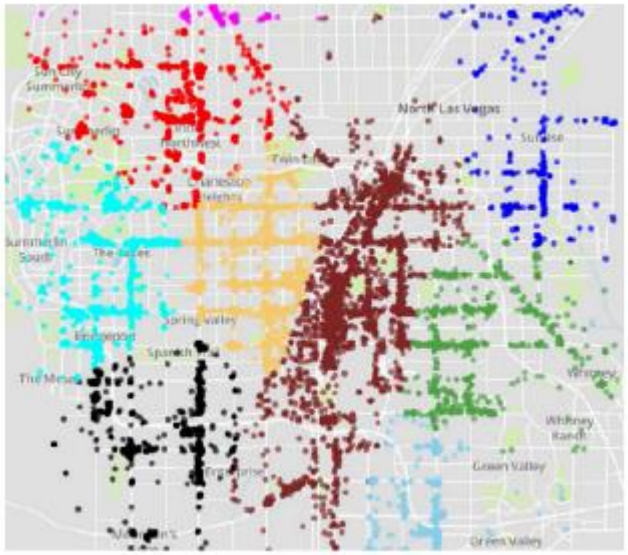
1. Recommend additional tags for partially annotated Flickr photos
2. Discover reliable associations among tags or tag categories from historical data
3. Recommend the most likely associated tags



Luca Cagliero, Alessandro Fiori, Luigi Grimaudo. 2014. Personalized tag recommendation based on generalized rules. ACM TIST 2014

Machine Learning-based approaches: example of Deep Learning-based hybrid method

1. Recommend relevant Point-of-Interest considering the reviews, categories, and geographical locations
2. Train a Recurrent Neural Network
3. Divide the users into different groups and then train an RNN tailored to each group



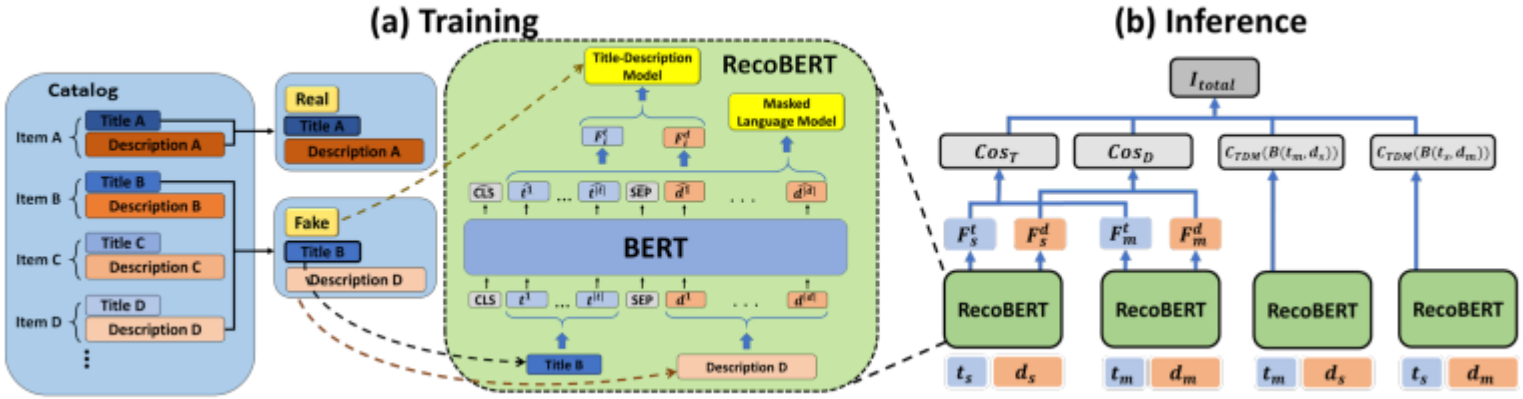
Guohui Li, Qi Chen, Bolong Zheng, Hongzhi Yin, Quoc Viet Hung Nguyen, and Xiaofang Zhou. 2020. Group-Based Recurrent Neural Networks for POI Recommendation. ACM/IMS TDS 2020

Text-based recommendation

- Recommendation using NLP techniques
  - Typically, hybrid approaches that combine usage data with either traditional or neural-based NLP methods
  - State-of-the-art approaches rely on contextualized embedding models
    - E.g., BERT

RecoBERT: A Catalog Language Model for Text-Based Recommendations

- BERT-based approach for learning catalog-specialized language models for text-based item recommendations
- It scores the similarities between pairs of items without the need for item similarity labels



Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, Noam Koenigstein: RecoBERT: A Catalog Language Model for Text-Based Recommendations. CoRR abs/2009.13292 (2020)

RecoBERT: A Catalog Language Model for Text-Based Recommendations

- Input
  - title-description pairs corresponding to positive (“real”) and negative (“fake”) samples, extracted from a given catalog
- Training
  - the title-description pairs are propagated through theBERT backbone and transformed into two feature vectors
  - The resulting vectors are then fed into the TDM, minimizing a cosine loss between them
- Inference
  - four scores are computed
  - Two scores propagate the seed and candidate items separately (“real” pairs)
  - The other two scores utilize the TDM head and propagate title-description pairs extracted from both seed and candidate items (“fake” pairs)

Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, Noam Koenigstein: RecoBERT: A Catalog Language Model for Text-Based Recommendations. CoRR abs/2009.13292 (2020)

Additional reading on RecoBERT

- Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, Noam Koenigstein. RecoBERT: A Catalog Language Model for Text-Based Recommendations. CoRR abs/2009.13292 (2020)
- Please read the paper: https://arxiv.org/pdf/2009.13292.pdf

Item-based Collaborative Filtering with BERT

- Item-based collaborative filtering
- Leverage the BERT architecture
  - To understand items, and score relevancy between different items
  - scores the similarities between pairs of items without the need for item similarity labels

Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, Noam Koenigstein: RecoBERT: A Catalog Language Model for Text-Based Recommendations. CoRR abs/2009.13292 (2020)

Additional reading on BERTbase

- Item-based Collaborative Filtering with BERT. Yuyangzi Fu. Tiang Wang.. CoRR abs/2009.13292 (2020)
- Please read the paper: https://aclanthology.org/2020.ecnlp-1.8.pdf

BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

- Prior work
  - employ sequential neural networks
    - encode users' historical interactions from left to right into hidden representations for making recommendations
    - Restrict the power of hidden representation in users' behavior sequences
- Solution
  - deep bidirectional self-attention to model user behavior sequences
  - Cloze objective to sequential recommendation, predicting the random masked items in the sequence by jointly conditioning on their left and right context

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In <i>Proceedings of the 28th ACM International Conference on Information and Knowledge Management< (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1441–1450. DOI:https://doi.org/10.1145/3357384.3357895

Additional reading on BERT4Rec

- Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1441–1450. DOI: https://doi.org/10.1145/3357384.3357895
- Please read the paper: https://arxiv.org/pdf/1904.06690.pdf

Results assessment

- Verify the quality of the predicted ratings
- Quantify errors, coverage, and diversity
- Compare alternative solutions (or the same solution with different configuration settings)

Benchmark datasets

- **MovieLens** (category: movies)
  (https://grouplens.org/datasets/movielens/)
- **Million Song Dataset** (category: music)
  (http://millionsongdataset.com/)
- **Amazon Review Data** (category: e-commerce)
  (https://nijianmo.github.io/amazon/index.html)
- **Book-Crossing** (category: books)
  (http://www2.informatik.uni-freiburg.de/ cziegler/BX/)

Benchmark dataset comparison

Dataset	Users	Items	Ratings	Density	Rating Scale
Movielens 1M	6040	3883	1,000,209	4.26%	[1-5]
Movielens 10M	69,878	10,681	10,000,054	1.33%	[0.5-5]
Movielens 20M	138,493	27,278	20,000,263	0.52%	[0.5-5]
Jester	124,113	150	5,865,235	31.50%	[-10, 10]
Book-Crossing	92,107	271,379	1,031,175	0.0041%	[1, 10], and implicit
Last.fm	1892	17632	92,834	0.28%	Play Counts
Wikipedia	5,583,724	4,936,761	417,996,366	0.0015%	Interactions
OpenStreetMap (Azerbaijan)	231	108,330	205,774	0.82%	Interactions
Git (Django)	790	1757	13,165	0.95%	Interactions

Source: https://www.kdnuggets.com/ (2016)

Results assessment

- $F_p(u, i)$: predicted rating for user u and item i
- $F_a(u, i)$: actual rating for user u and item i
- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{u \in U, i \in I} (\mathcal{F}_p(u,i) - \mathcal{F}_a(u,i))^2}$$

Results assessment

- Precision at top K (P@K)
  - percentage of correct items in top K recommended items for a given user
- Rank correlation
  - Spearman's correlation between the vector of predicted ratings and those of actual ratings

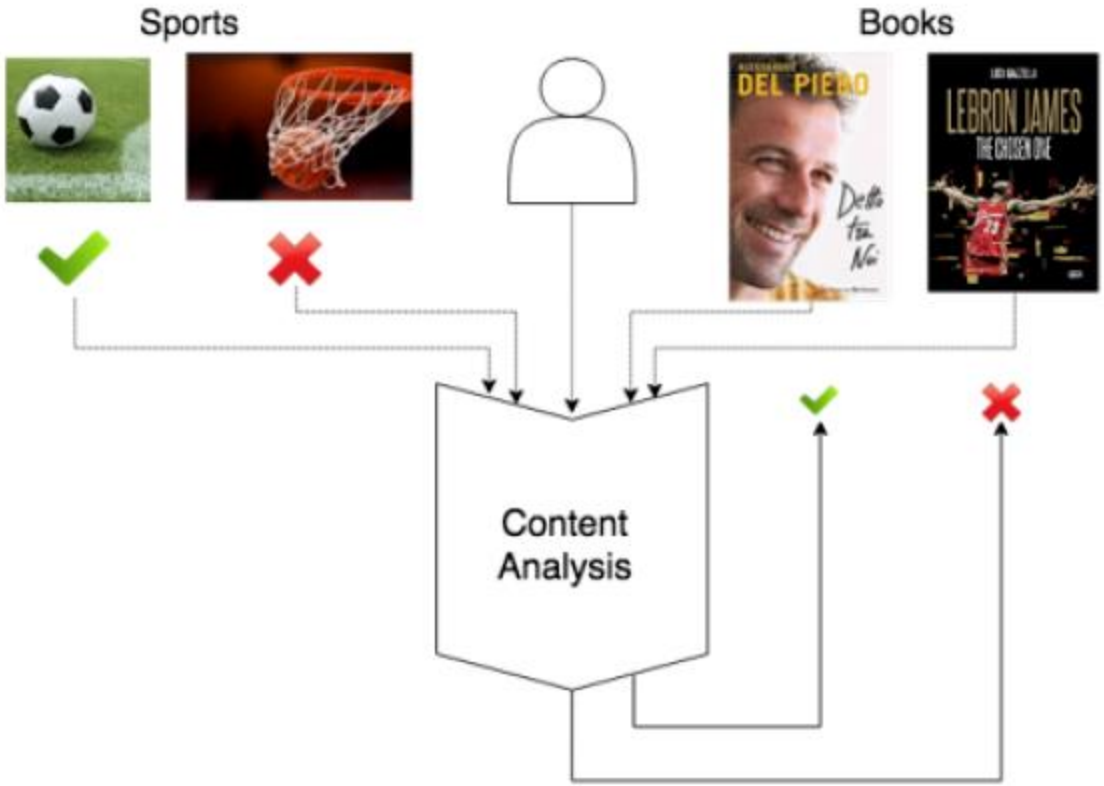
Results assessment: binary prediction

- Boolean rating: either relevant or not relevant
- Mean of Reciprocal Error $MRR = max_{i \in I^r} \frac{1}{rank(i)}$
  - Ir is the set of relevant items
  - rank(i) is the item rank in the recommended list
- Coverage
  - Number of items/users for which the recommender can make predictions
- Precision
  - Percentage of recommended relevant items
- Receiver operating characteristic (ROC)
  - Trade-off curve between false positives and false negatives

Results assessment: further aspects

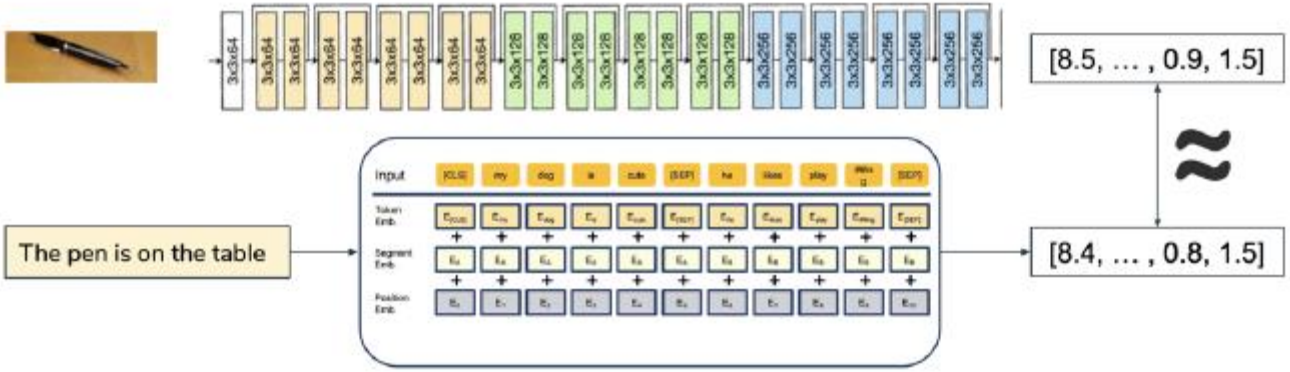
- Diversity
  - diversify recommendations with the purposes of
    - Increase user satisfaction
    - Mitigate the potential effect of overfitting
- Domain adaptation
  - Tailor the recommender systems to the actual context
- Prediction order
  - refine the output rank according to domain-specific constraints
- Biased annotations
  - Ratings either do not reflect the actual service needs or miss a global viewpoint

Challenges: contextualization



Challenges: transfer learning

Challenges: multimodal learning



Challenges: privacy and data protection

Challenges: real-time processing and scalability

- The greatest majority of the presented solutions are unsuitable for real time computation!
- Deep Learning solutions require dedicated hardware (GPUs)
- The inherent algorithm complexity limits the scalability towards Big Data (except for few cases)
- Take a look at https://spark.apache.org/mllib/

Acknowlegdements and copyright license

- Copyright licence
  - Attribution + Noncommercial + NoDerivatives

- Acknowledgements
  - I would like to thank Dr. Moreno La Quatra, who collaborated to the writing and revision of the teaching content
- Affiliation
  - The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
    - https://dbdmg.polito.it
    - https://smartdata.polito.it

Thank you!