INFSCI 2480: Adaptive Information Systems

Social Information Access-Chapter 12

Tag-Based Recommendation

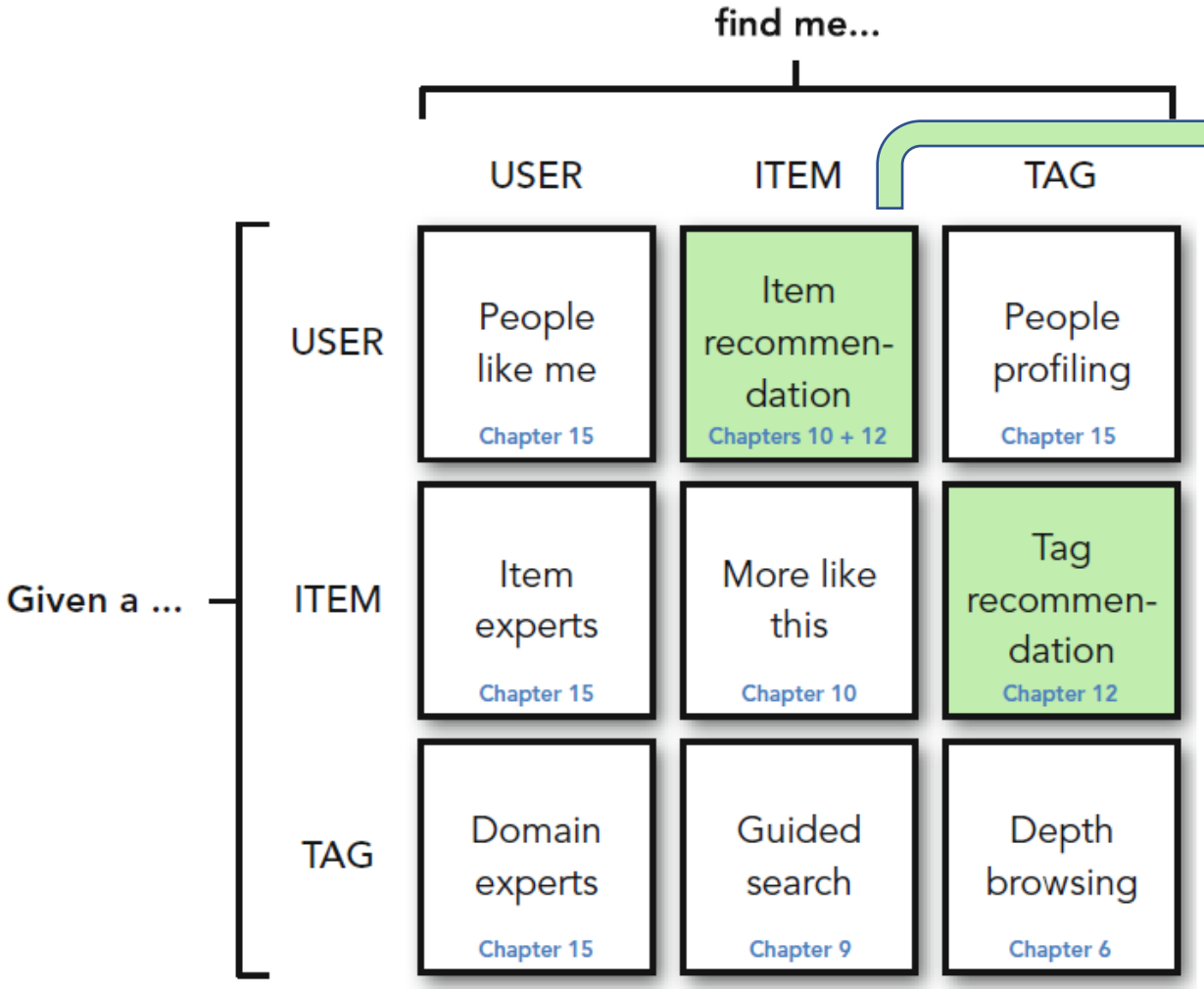
by Toine Bogers

Presenter: Kamil Akhuseyinoglu

Where we are

	Search	Navigation	Recommendation
Content-based			
Semantics / Metadata			
Social			

Chapter Overview



Tag-based item recommendation.
Improved recommendations by *tags*

Generating ***better*** recommendations by incorporating ***tags*** into the recommendation process

Recommending tags to items.
Not the main focus of this chapter

Tags

- Another source to generate recommendations
- Product of *social tagging*
  - Users describe and categorize items **for their own purposes**
  - User-centric, subjective
- May be a list of words, *keywords*
- Self-reference or task organization
- Result in bottom-up classification
- No hierarchy or grouping in nature of tags
  - But, tag taxonomies introduced

Examples…

Software Engineer

M*Modal *Pittsburgh, PA*

java spring

How do I calculate the cosine similarity of two vectors?

28

20

How do I find the cosine similarity between vectors?

I need to find the similarity to measure the relatedness between two lines of text.

For example, I have two sentences like:

> system for user interface
>
> user interface machine

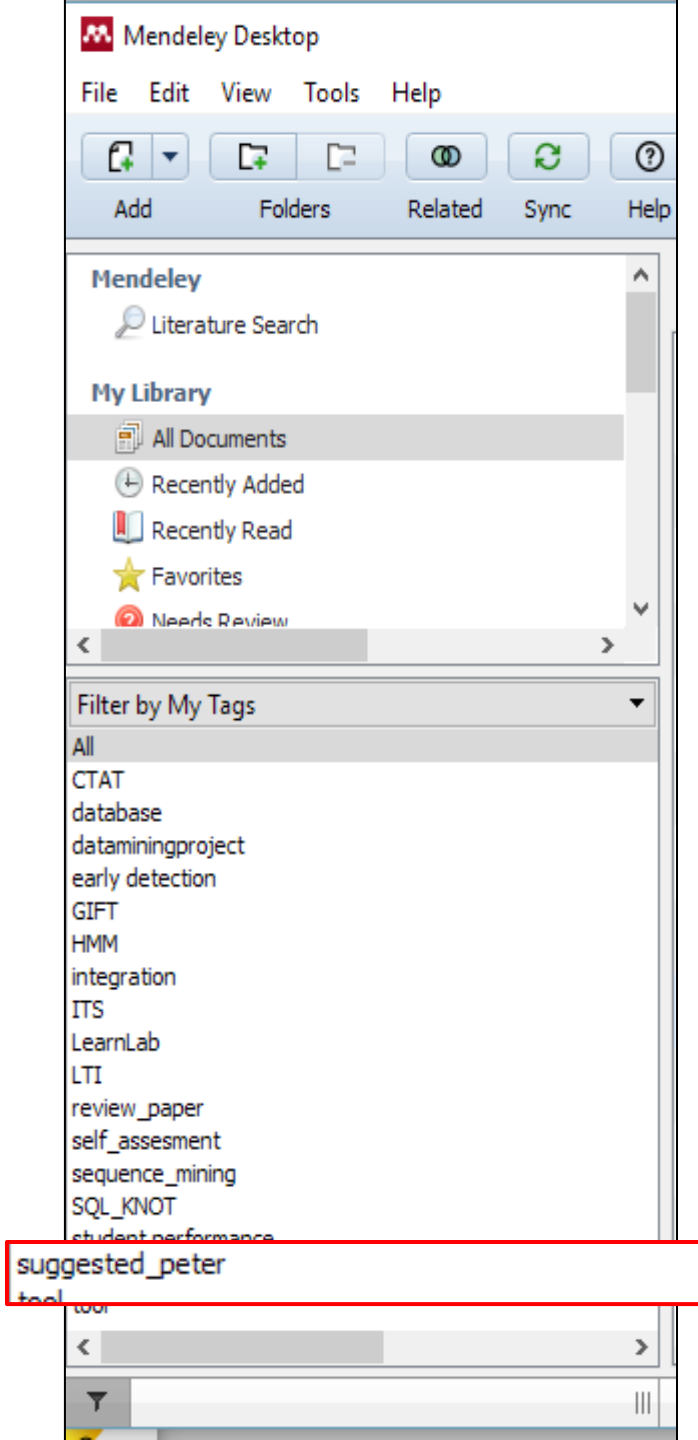
… and their respective vectors after tF-idf, followed by normalisation using LSI, for example `[1,0.5]` and `[0.5,1]`.

How do I measure the smiliarity between these vectors?

java vector trigonometry cosine tf-idf

Inbox 65

AIS Project Inbox × Important Emails × pitt ×

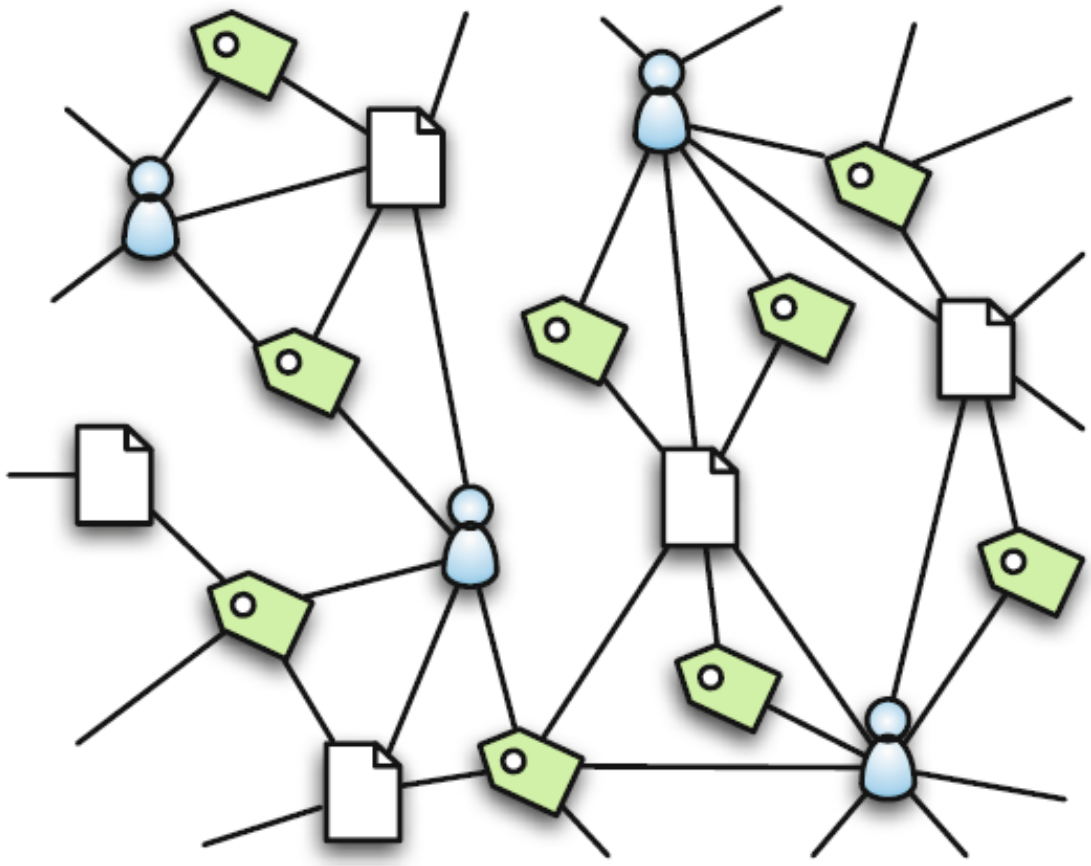


Preliminaries

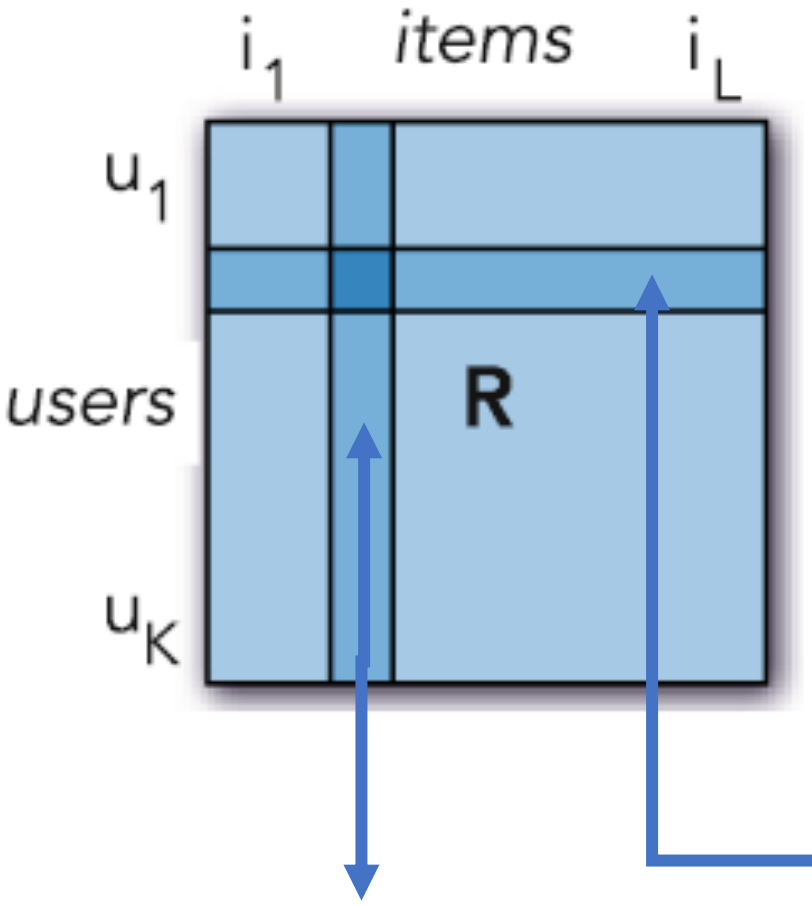
- Aggregation of all public tags by all users: **Folksonomy**
  - Serves as an extra annotation layer
  - Connects users and items
- ***Folks*** = People, ***[o]nomy*** = system of rules, knowledge
- Leads to tag-based classification (vs. taxonomies by content owners)

Folksonomy Graph

Undirected ternary relations represented by edges



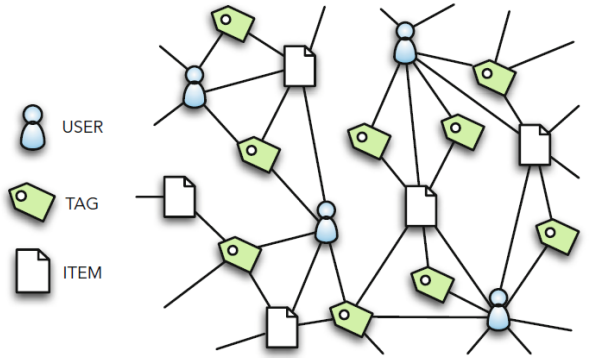
User-Item Matrix: R



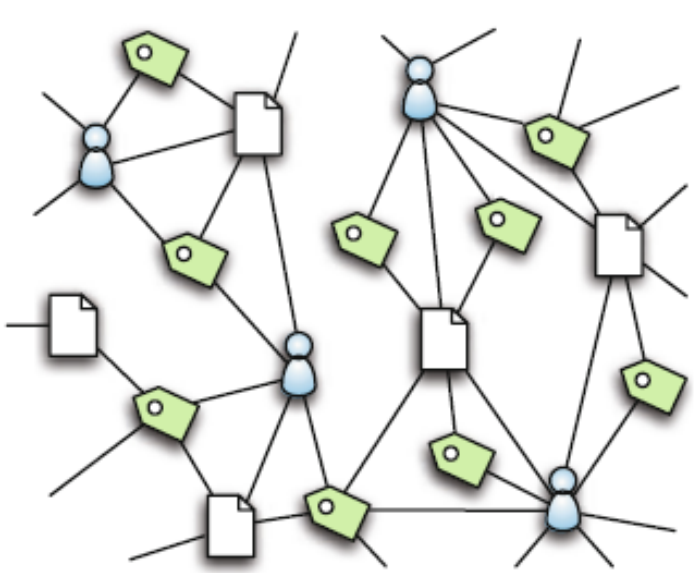
$\overrightarrow{u_k}$: User Profile Vector

$\overrightarrow{i_l}$: Item Profile Vector

- Conventional recommender systems use R
  - Collaborative filtering in particular
- Contains numerical user preference
  - Explicit: Ratings
  - Implicit: User behavior, play counts
  - Weight of user-item edges in graph
- What if we do not have ratings but tags?



User-Item Matrix by Utilizing Tags



items

3D Matrix or Tensor

D

users

tags

$\mathcal{D}(u_k, i_l, t_m) = 0 \quad or \quad 1$

$\sum$ Aggregation over tags

ratings

user u_k

tagged item i_l

with tag t_m

i_1 items i_L

u_1

users

u_K

R

{0 or 1}

binarized

tag counts

i_1 items i_L

u_1

users

u_K

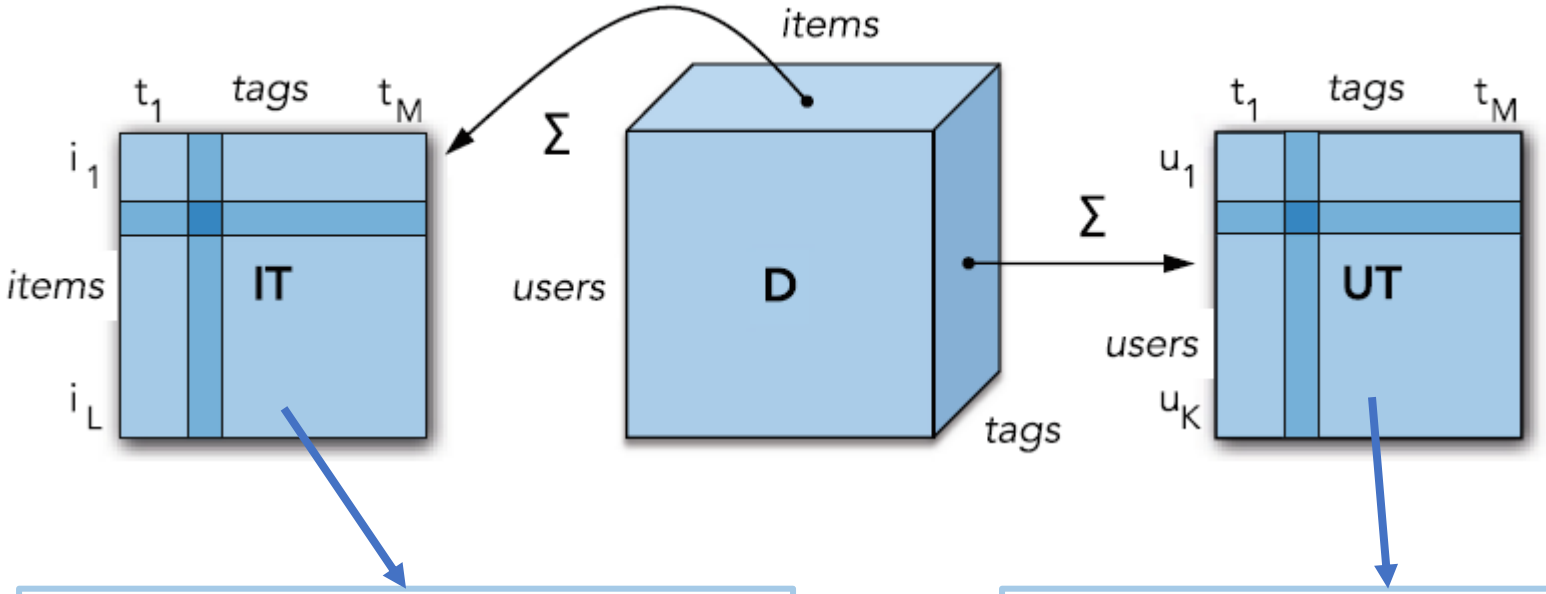
UI

Tag counts

[0, 1, 1, 1, 0, 1]

[0, 14, 3, 1, 0, 5]

Alternative Ways of Aggregating *D*



How many users assigned a certain tag to an item.

Row: $\overrightarrow{f_l}$: Item-Tag Profile Vector

How often each user used a certain tag to annotate their items.

Row: $\overrightarrow{d_k}$: User-Tag Profile Vector

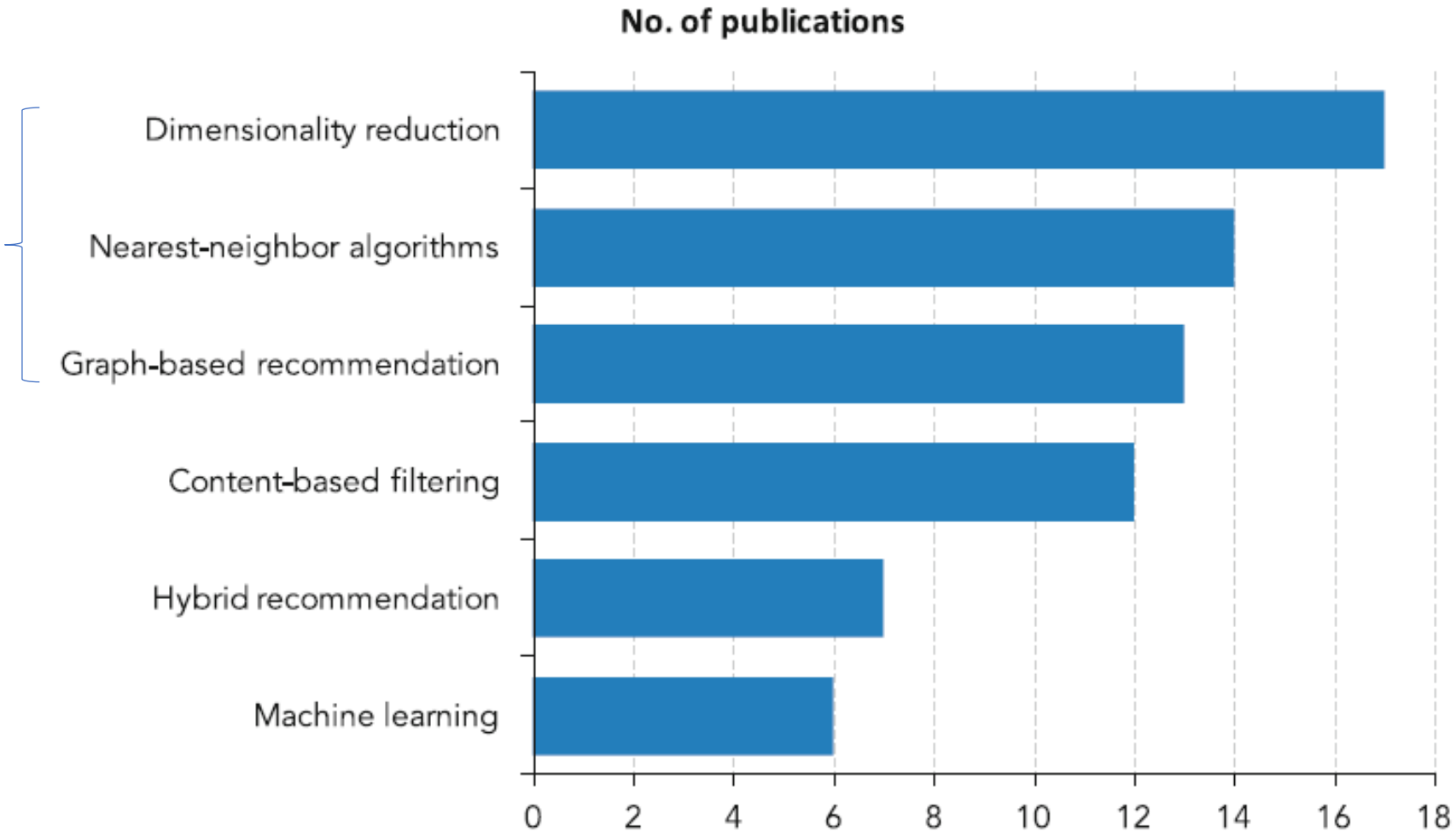
Tag-Based Item Recommendation

Task: Recommend interesting items to a user based on preferences

- Tags are integrated to regular item-recommendation algorithms
- Goal of these algorithms: Top-N item recommendation
  - Calculate a relevancy score for items (not in the profile of user)
  - Rank all items

Algorithm Frequencies in the Chapter

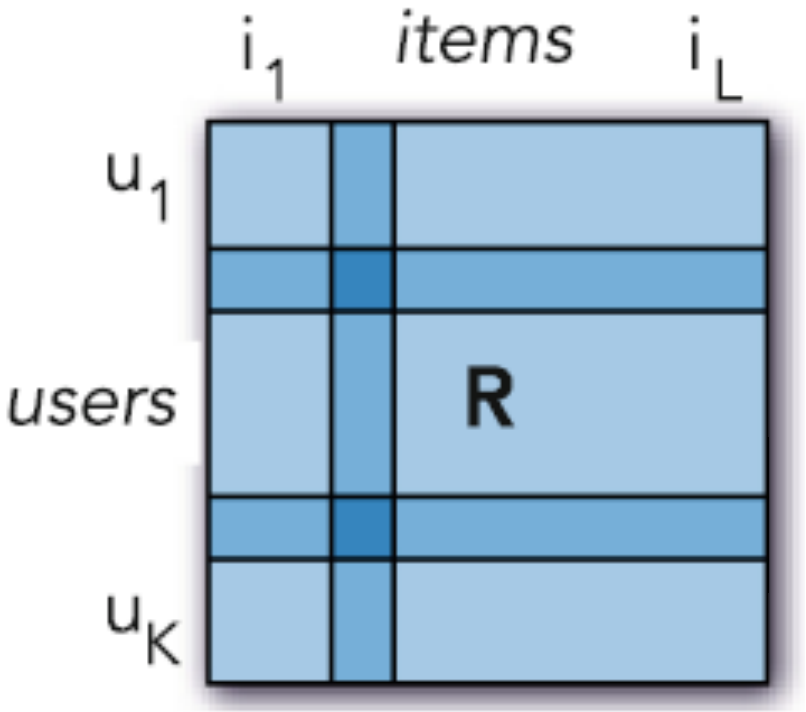
Collaborative Filtering



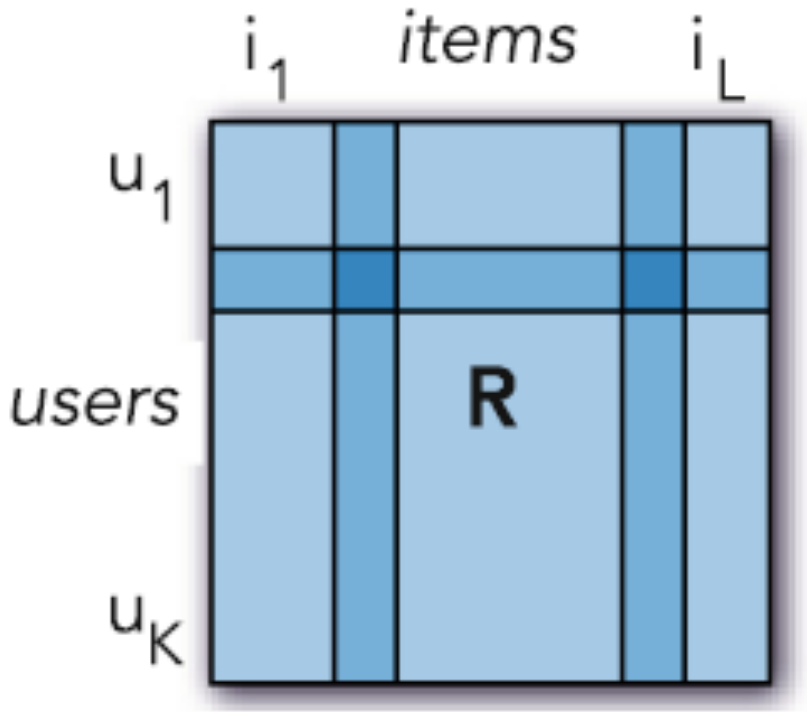
Collaborative Filtering – Nearest Neighbor Algorithms

- Simple, easy extensibility
- Known also as memory-based or lazy
  - Lazy: When user requests
- 2 Variants: User-based and Item-based
  - **User-based:** Find most similar users to active user in terms of consumption or rating behavior
  - **Item-based:** Find most similar items. Similar item means same set of users purchase or rated them highly
- More details in Chapter 10

User-based NN



Item-based NN



Calculate similarity and construct neighborhood, e.g., cosine similarity

Sort users/items by frequency or similarity

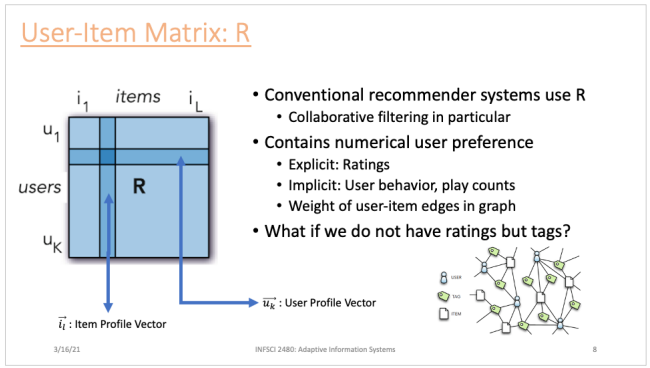
Generate recommendations

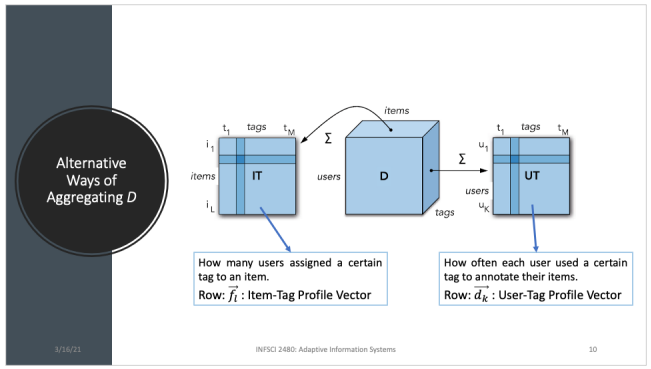
Tag-based NN Algorithms

- Use tags to aid calculation of user/item similarities
- Or replace **R** or **UI** matrices completely, use **UT** and **IT**
- Bogers reported average # of tags between 3.1 to 8.4
- Less sparse
  - Potential reduction in sparsity by factor of 3
- Researchers reported mixed results

Tag-aware Fusion Algorithm (Tso-Sutter et al.)

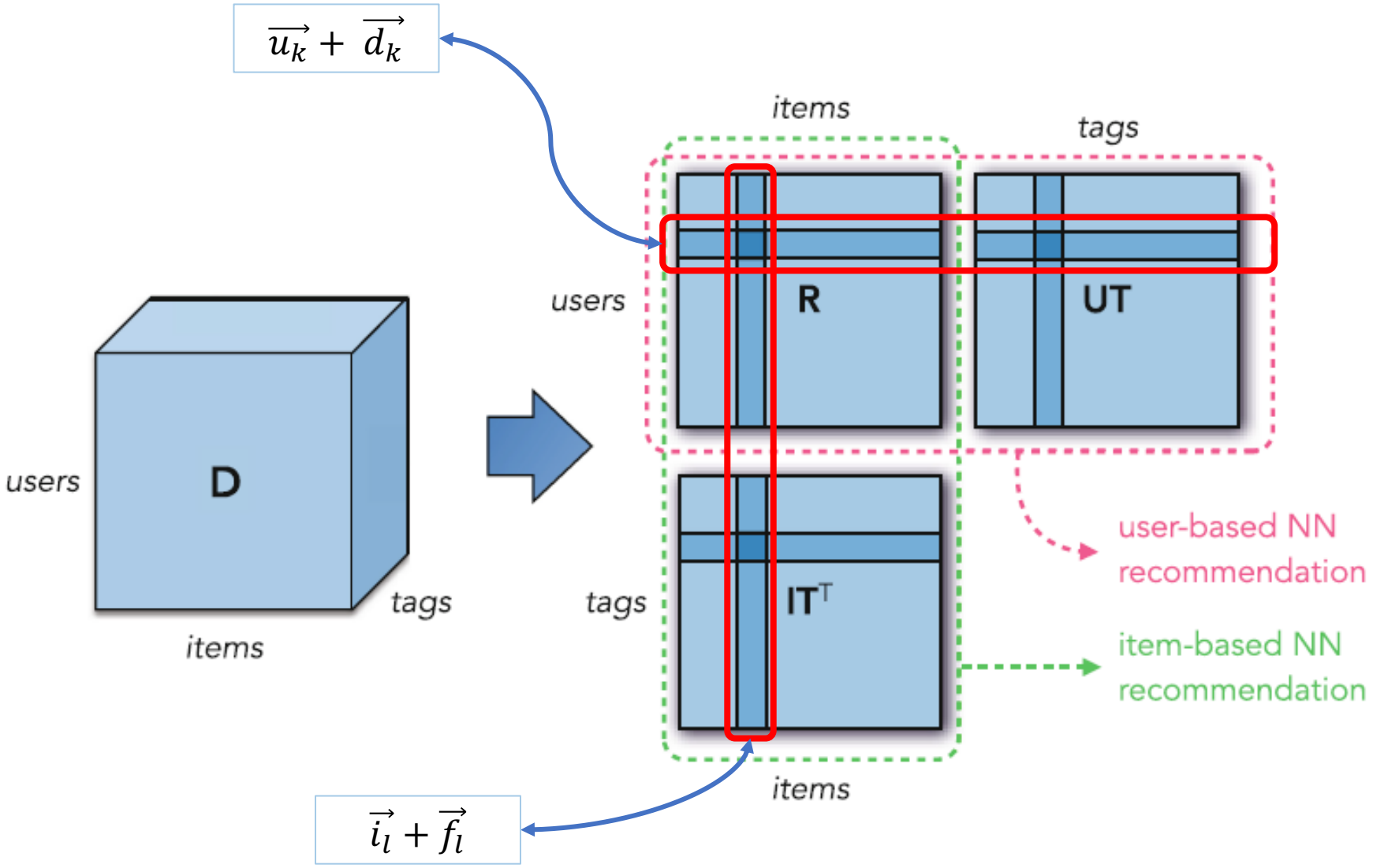
- Adaptation of standard NN that fuses 3 matrices: ***R, UT, IT***
- Combination of user and item-based CF
- Reduces sparsity
- Concatenates user's profile vector $\overrightarrow{u_k}$ with user-tag vector $\overrightarrow{d_k}$
- Concatenates item's profile vector $\overrightarrow{i_l}$ with item-tag vector $\overrightarrow{f_l}$





Extending R by tag-based matrices

Concatenate user's profile vector $\overrightarrow{u_k}$ with user-tag vector $\overrightarrow{d_k}$



Concatenate item's profile vector $\overrightarrow{i_l}$ with item-tag vector $\overrightarrow{f_l}$

User and Item Similarity Calculation in Fusion

Similarity Calculation

Extend user-item matrix **(R)** by including user tags **(UT)** as items and item tags **(IT)** as users.

Similarity Fusion

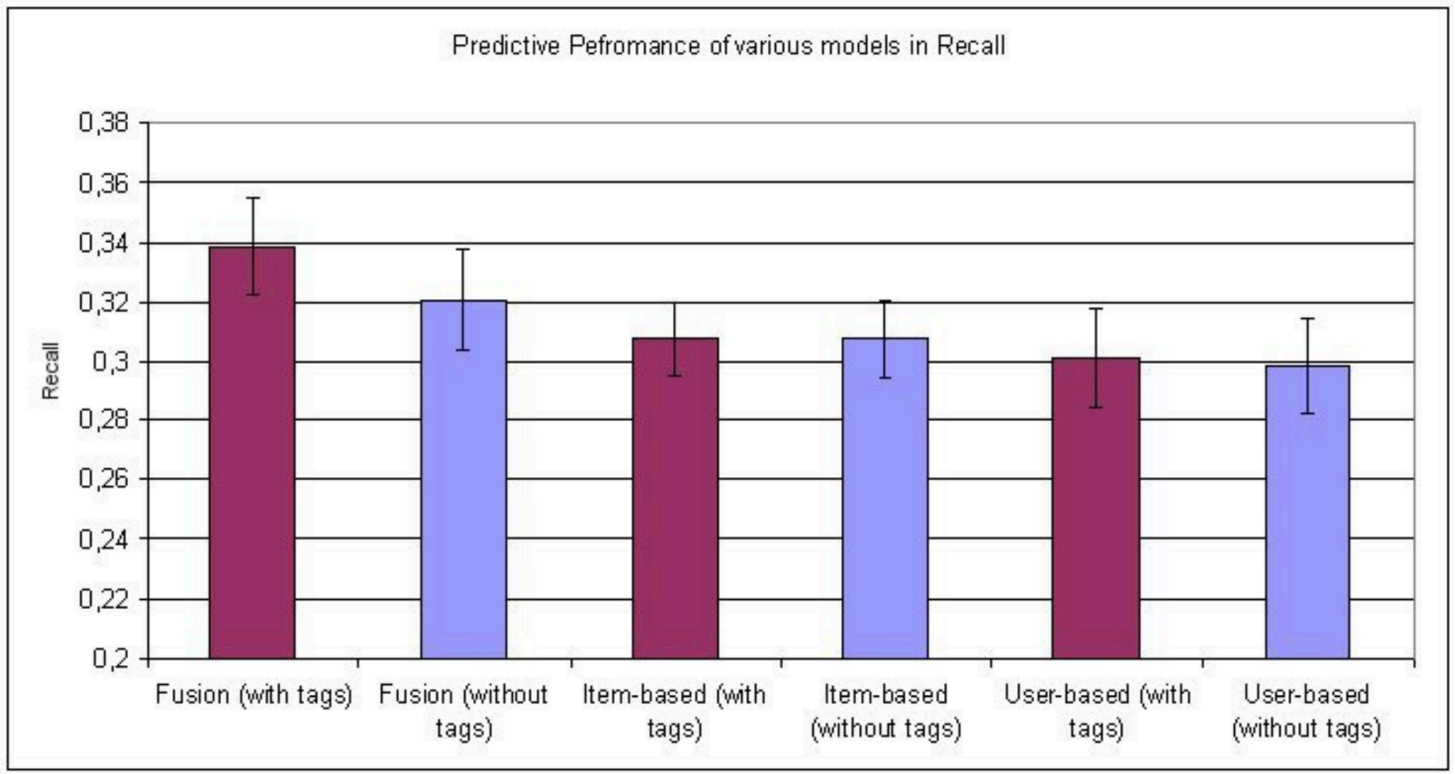
The fusion of the user- and item-based predictions was done by computing **linearly weighted sum** of the two conditional probabilities.

$$P(r_{u,i}|w(u,v))\lambda + P(r_{u,i}|w(i,z))(1-\lambda).$$

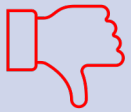
Adjust significance

Does Fusion Improve Recommendations?

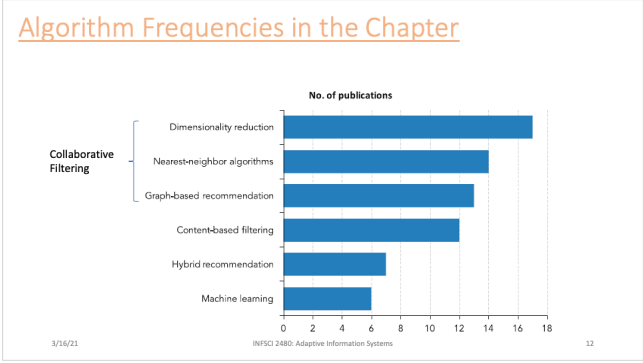
- Test with self-crawled dataset from Last.FM
- Against baseline NN
- No performance improvements in separate user-based and item-based variants
- Significant improvement of fused approach



Other Proposed Approaches and Results

Authors	Domain	CF Type	Approach Used	Evaluation vs. Baseline
Firan et al.	Music recommendation (Last.FM)	User-based NN	UT	
Nakamoto et al.	Social-tagging platform	User-based NN	UT	
Zhao et al.	Social-tagging (Dogear)	User-based NN	UT + semantic tag similarity	
Parra and Brusilovsky	Social-tagging (CiteULike)	User-based NN	UT + BM25 term weighting	
Kim et al.	Social-tagging (*BibSonomy)*	User-based NN	UT + tag similarity ambiguity + synonymity	
Bogers and Van den Bosch	Social-tagging (*BibSonomy* + CiteULike + Delicious)	User-based NN + Item-based NN	UT + IT	
Zeng and Li	Social-tagging (Delicious)	User-based NN + Item-based NN	UT + IT	

CF-Dimensionality Reduction



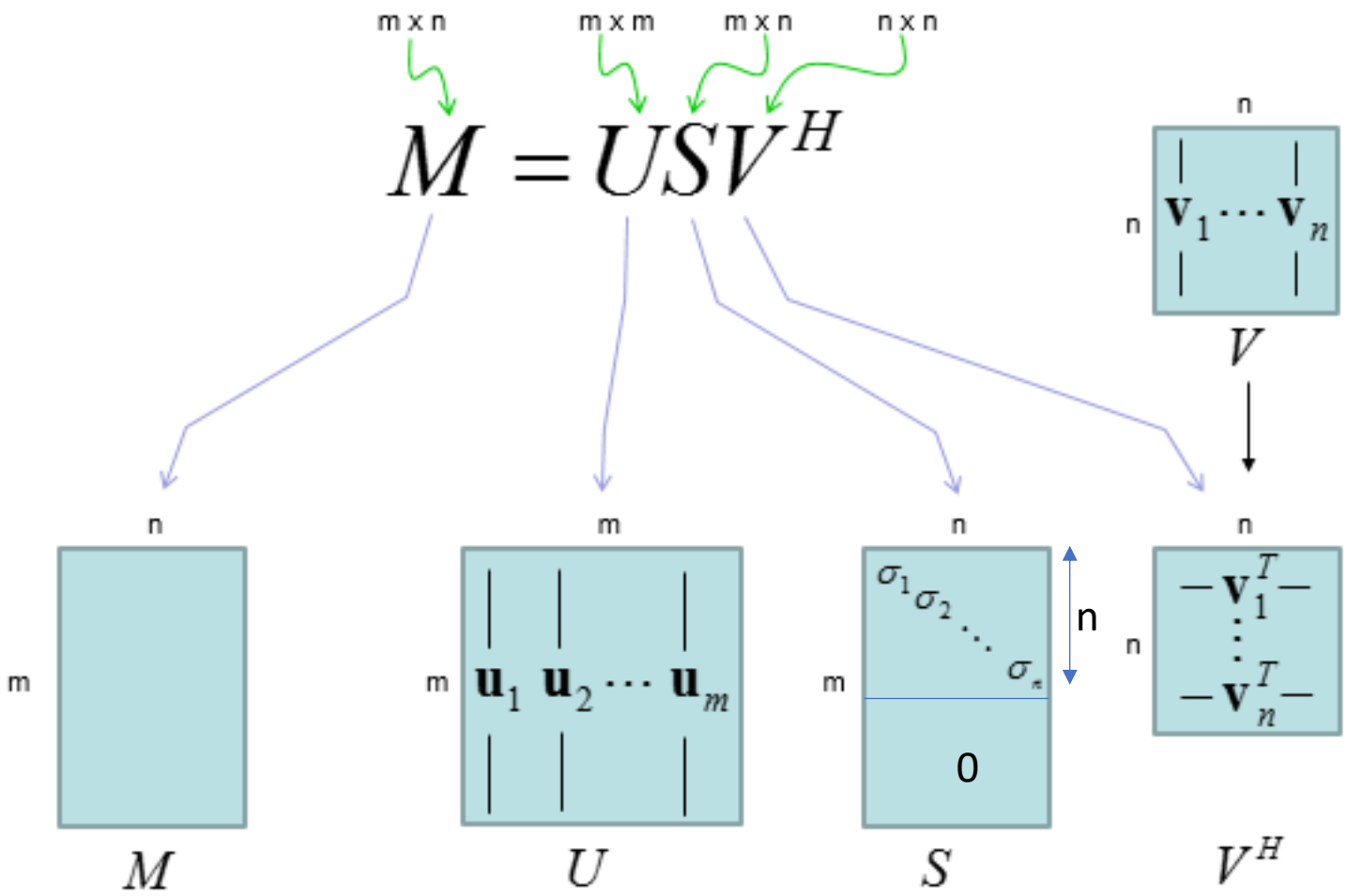
- **Goal:** To reduce the complexity of the R matrix
  - Transform both the users and items to lower-dimensional latent factor space
- First DR algorithm: *Latent Semantic Indexing*
  - Proposed by Sarwar et al. in 2002
  - For the Netflix Prize
- Details in Chapter 10 for CF algorithms with DR techniques

Tensor (D) Reduction (Symeonidis et al.)

- Tag-based CF Algorithm + DR Techniques
- Based on Singular Value Decomposition (SVD)
  - Data reduction
  - Data-driven generalization
  - Linear regression models
  - Principal Component Analysis (PCA)

Video lectures on SVD: https://www.youtube.com/watch?v=gXbThCXjZFM

SVD in 2-D



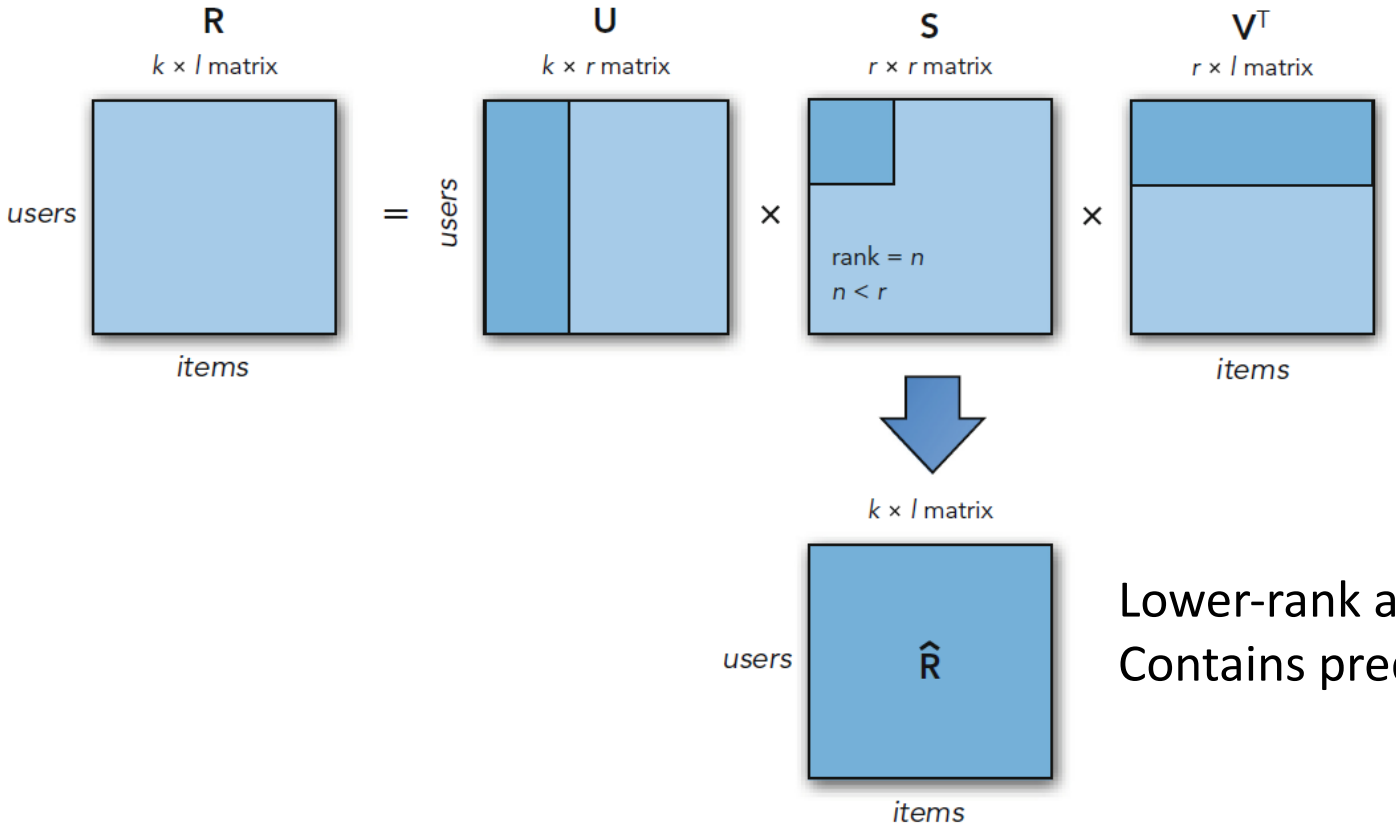
S non-negative, diagonal, ordered importance

$$M = \sigma_1 u_1 v_1^T + \ \ldots$$

Only n non-zero values in S
So, only first n columns of U are important

Tensor (D) Reduction (Symeonidis et al.)

- Tag-based CF Algorithm + DR Techniques
- Based on Singular Value Decomposition (SVD)



Two-dimensional

Lower-rank approximation of R.
Contains predicted ratings.

Eigenfaces

R

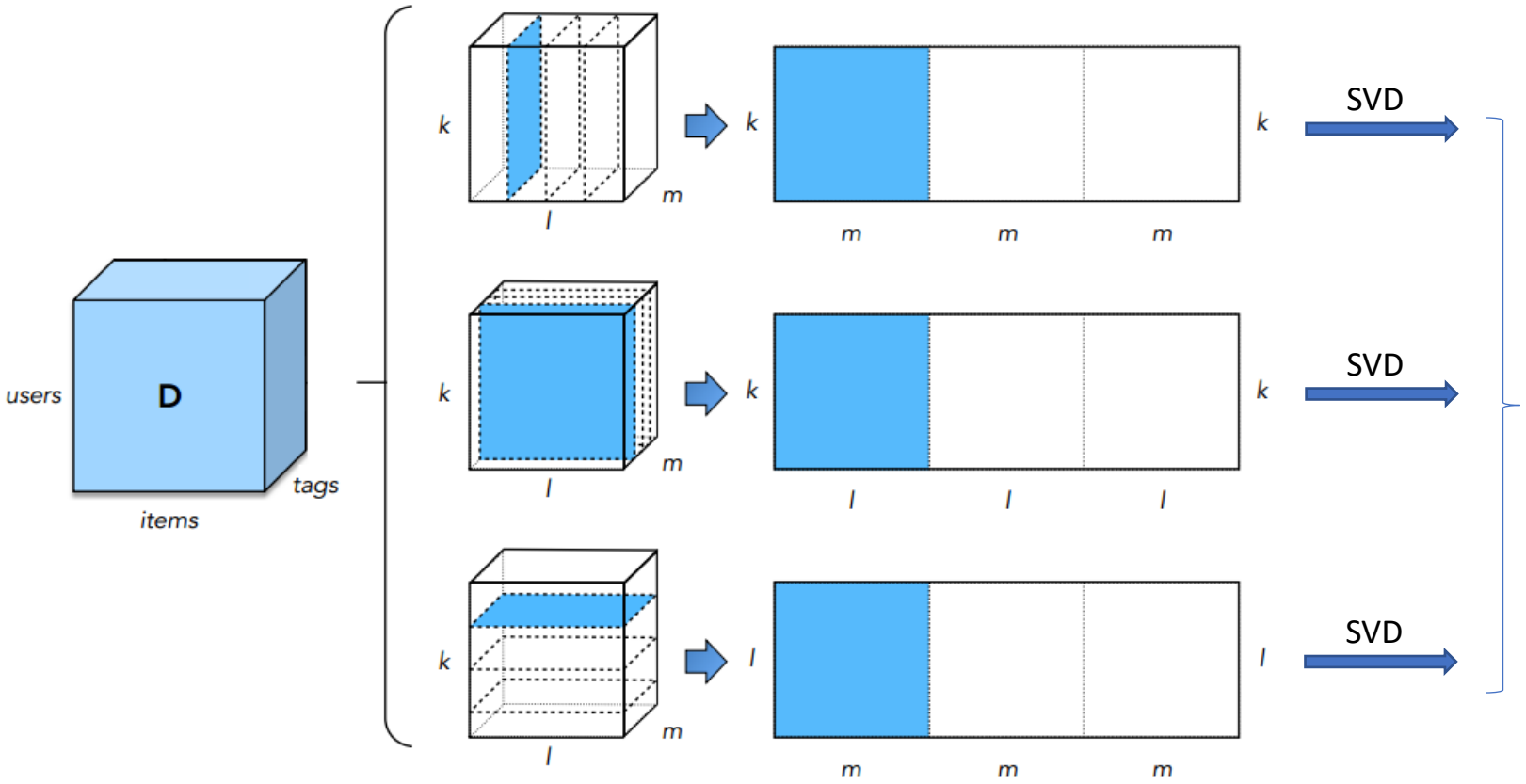
U

Lower rank estimation of R
N=250 (eigenfaces or components)

https://towardsdatascience.com/eigenfaces-recovering-humans-from-ghosts-17606c328184

Higher-Order Singular Value Decomposition (HOSVD)

Unfolding *D* in all three modes



- Reduce the dimensionality of the three resulting singular value matrices by 50%
- Reconstructed $\widehat{D}$ as reduced tensor
- Evaluated on Last.FM and BibSonomy for recommending items
  - Outperformed **FolkRank** algorithm

Other Tensor Reduction Approaches

HOSVD

- *Rafailidis and Daras*: Expand tags + k-means clustering + HOSVD

Tucker Decomposition

- *Peng et al.*: Reduce dimensionality

Probabilistic Latent Semantic Analysis (PLSA) approach

- *Wetzker et al.* and *Said et al.*
- Integrate tags by estimating the topic model from both user-item and item-tag occurrences
- Linearly combine the output of the two models

Tag-Enhanced Latent Dirichlet Analysis (LDA)

- *Wang et al.* applied to both UT and IT matrices
- Zhang et al. applied to entire tripartite graph instead 2D UT and IT matrices

Other Tensor Reduction Approaches

Matrix Factorization

- **Luo et al.** Integrates the latent factors of the item tags and ratings

Probabilistic Matrix Factorization

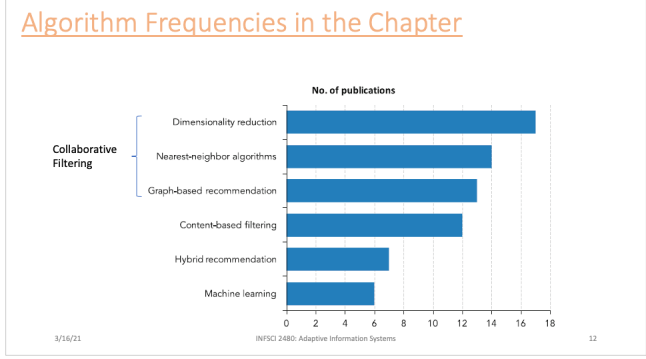
- **Xin et al.** Integrates the latent factors of the item tags and ratings
- **Zhen et al.** proposed *TagiCoFi:* Use tagging information to calculate a user-user similarity matrix, it can make the user-specific latent feature vectors as similar as possible if two users have the same tagging behavior
- **Yin et al.** proposed PMF + Bayesian approach, it allows them to model the relations between different data types

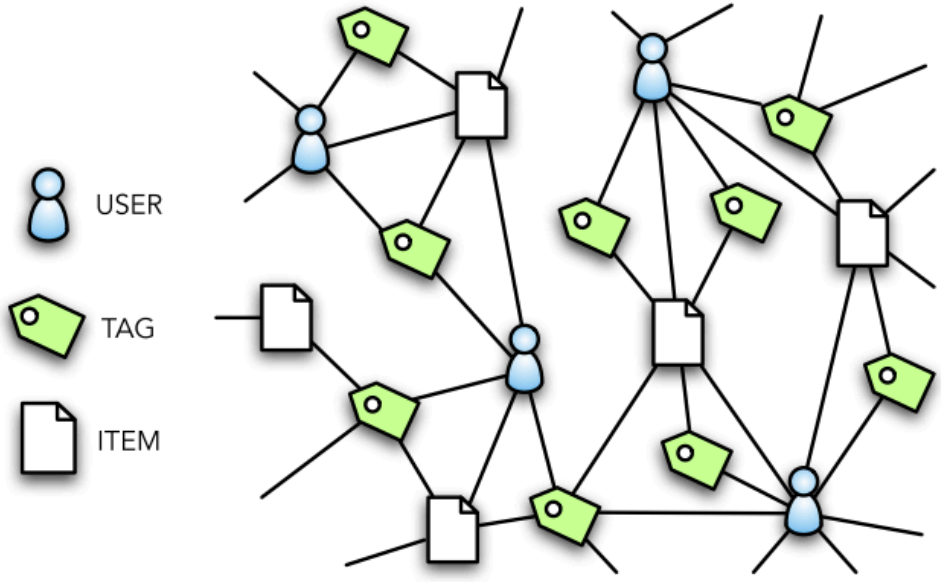
Cross-Domain Recommendation by Tags

- Tags used to link different domains
- E.g., recommending movies based on books ratings
  - Tags can link books to movies in same topic
- Enrich et al. proposed 3 MF algorithms on MovieLens and LibraryThing
  - Utilizing tags of active user; tags of all users; most discriminative tags
  - Enhance item factors
  - All outperform single-domain MF algorithm
- Fernandez-Tobias et al. extended Enrich et al. MF model
  - Improved both user and item factors by tags
  - Outperform Enrich et al. model

CF-Graph-Based Recommendation

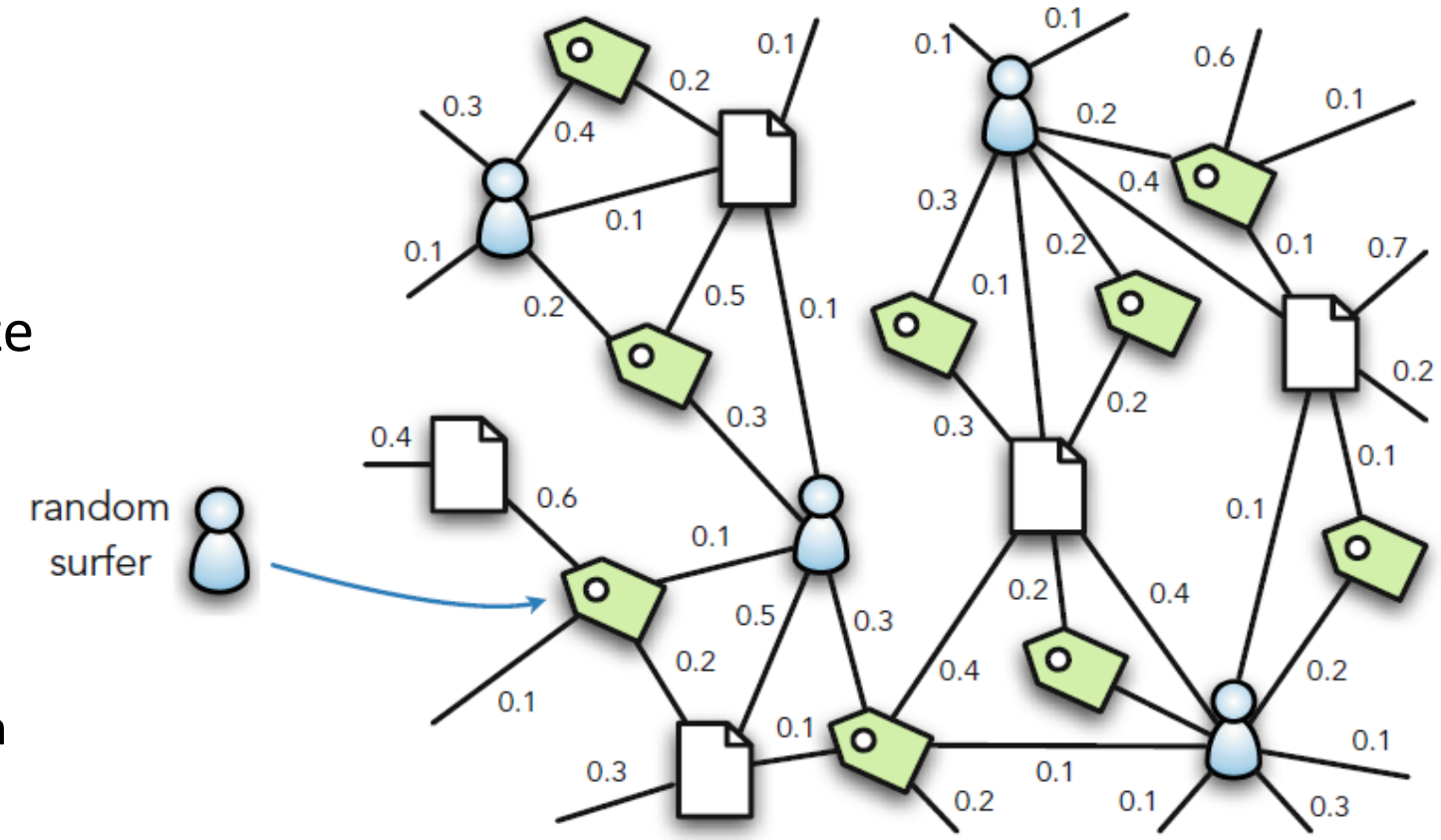
- Item preferences as bipartite network of user and item nodes
- Network structure is used to generate recommendations
- Tag-enhanced graph-based algorithms
  - Includes tags as additional nodes
  - Generate recommendations on tripartite network
- Some algorithms even add more nodes, e.g., actors



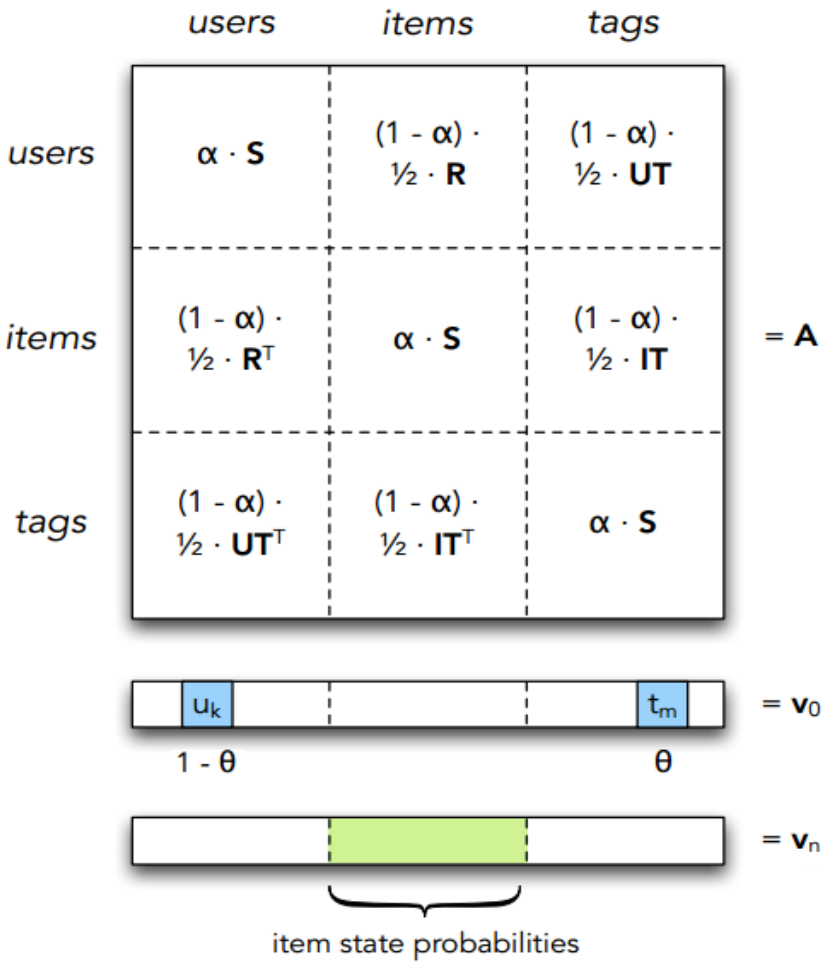


FolkRank Algorithm (Hotho et al.)

- Similar to PageRank
- Importance weights of nodes as the probability of random surfer to be found
- Include preference vectors to initialize the starting point(s) of the random walk
  - Some nodes might higher chance to be picked
- Personalized Recommendations
  - Calculating the rank difference between the ***preference vectors*** and the ***without such preference vectors***
- In BibSonomy data, outperform PageRank



Personalized Markov Random Walk Algorithm (Clements et al.)

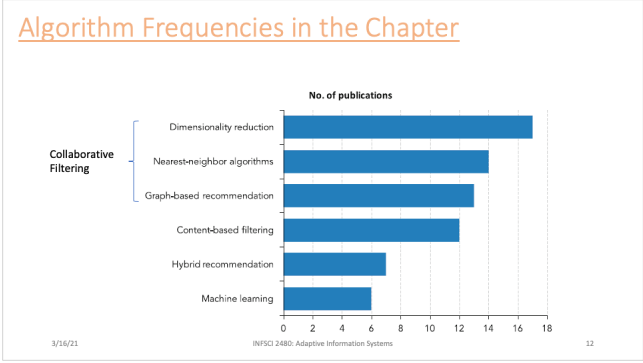


- Tripartite graph of user, items, and tags, created by all transactions and tagging actions, as a transition matrix A.
- The edges between these entities are determined by rating (R) or tag count (UT, IT)
- $\alpha \in [0,1]$, probability of self-reference
- Rows sum to 1 (normalized)
- v_0 initial state vector: index to target user and searched selected tag assigned using weight θ
  - to determine influence of personal profile vs. query tag
  - Θ = 1 means tag-based search
  - Θ = 0 means personalized item recommendation
- After n steps, item state probabilities taken (green part), ranked

$$\overrightarrow{v_{n+1}} = \overrightarrow{v_n} \cdot \mathbf{A}$$

- Outperformed traditional NN on LibraryThing and BibSonomy

Content-Based Filtering (CBF)



- Build a representation of the content in a system and then learning a profile of the user's interests
- Content representations are then matched against the user's profile to find the items that are most relevant
- Two CBF approaches

  Information Retrieval

  Document representations are matched to user representation on textual similarity

  Machine Learning

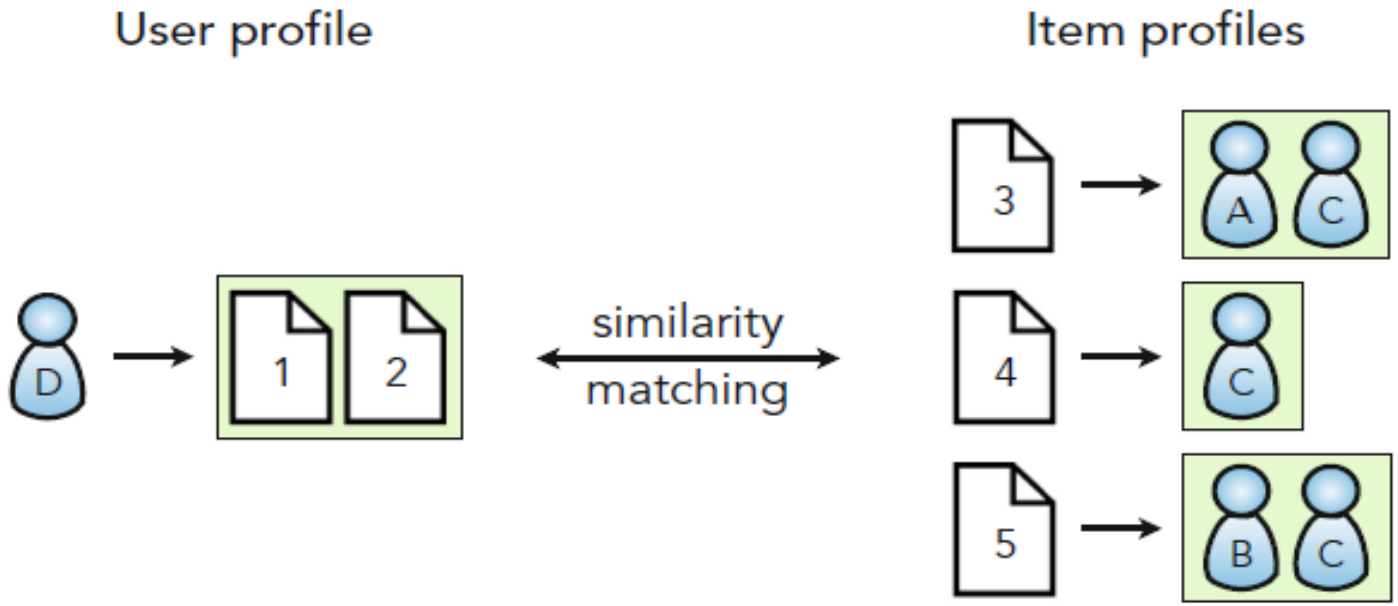
  Textual content is represented as feature vectors and prediction algorithm is trained

Tag-based CBF

- CBF is traditionally successful in **text-heavy** domains
- Tags are condensed textual descriptions of items
  - Automatically annotating multimedia is challenging
  - Present an opportunity to apply CBF to these domains
- Bogers and Van den Bosch proposed two CBF algorithms
  1. Profile-centric matching
  2. Post-centric matching

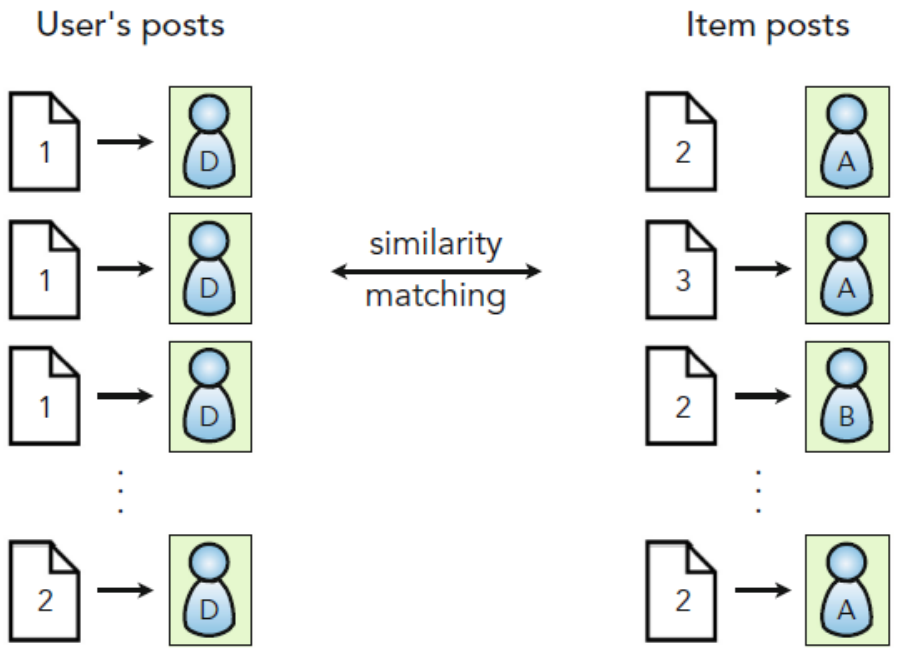
Profile-Centric Matching

- **User profiles**: aggregating all metadata and tags assigned to the active user's past items into a single textual representation
  - Includes tags assigned by other users
- **Item profiles**: aggregating all metadata and tags assigned to that item in its lifetime.
- **Matching:** User's profile is matched by items using IR language modeling.
  - Relevance ordered list of item recommendations remains



Post-centric Matching

- **User profiles**: Set of individual posts with general metadata and user-specific tags
- **Item profiles:** Aggregated posts of all users
- **Matching:** User's profile is matched to all other posts
  - List of matching posts with similarities calculated
  - Rank-corrected sum of normalized similarity scores used for recommendations



Evaluation

- BibSonomy, CiteULike and Delicious datasets
- Post-centric outperformed profile-centric matching
  - Due to dense item representations
  - Easier to match

Other Approaches in CBF

- **Cantador et al.**

  Find that the best performance is achieved using the **BM25 term weighting** scheme to calculate the tag weights and **using cosine similarity** to match user and item profiles.

- **Szomszor et al.**

  Propose a CBF approach that ranks the unseen items for an active user by the **overlap** between the **tags** assigned to those items and the **active user's tag cloud**.

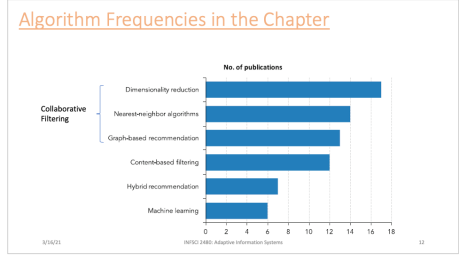
- **Wartena et al.**

  Propose **a topic-aware, tag-based CBF algorithm** that generates recommendations for each of the topics detected in a user's profile.

Vocabulary Problem in CBF Approaches

- People use different terms to describe the same objects.
- Semantic CBF approaches have been proposed.
  - Compute similarity between tags by constructing ***a tag-to-tag matrix*** derived from **UI matrix.**
  - **WordNet** to disambiguate tags.
  - **DBPedia**, a linked open data version of Wikipedia to match tags to other related items.

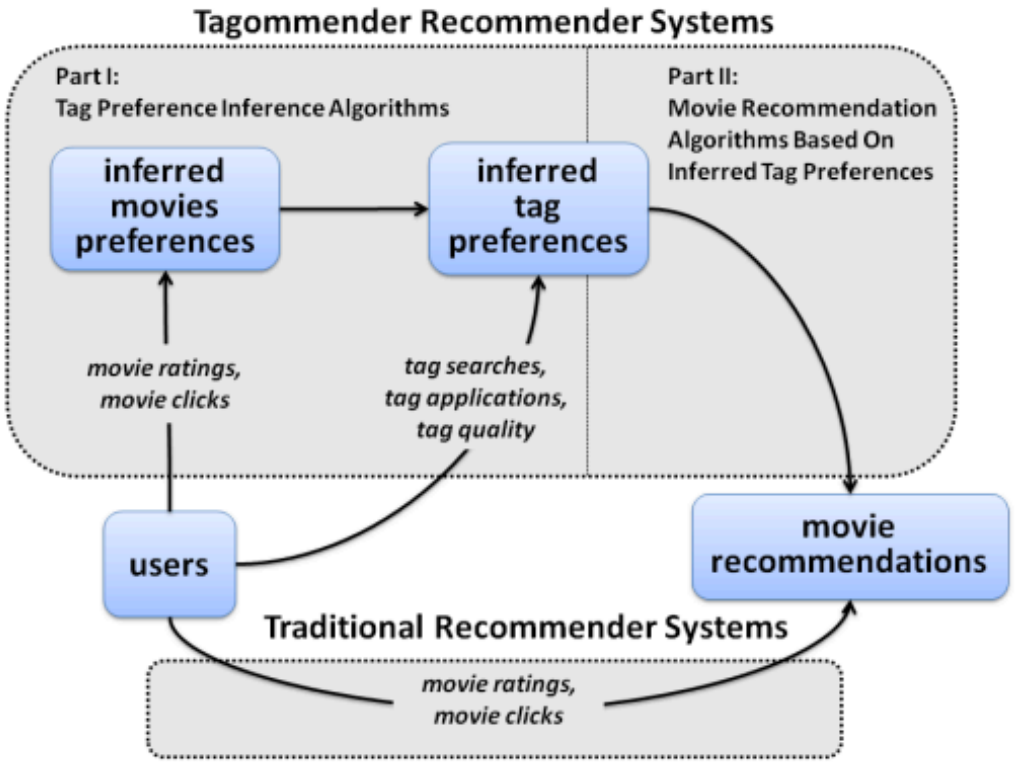
Tag-Based Machine Learning Algorithms



- **Kim et al.** used the **UT** matrix to generate tag profiles for each user.
  - Tags serve as input to ***Naive Bayes classifier***
  - Predicts items that user might like given user's profile tags and item-tag co-occurrence counts in **IT** matrix
  - Later included rating information to this approach
- **Vatturi et al.** proposed a tag-based model to recommend Web pages
  - Used ***Naive Bayes classifier***
  - Recently bookmarked items receive higher rating
- **Guan et al.** proposed a ML approach that learns
  - ***2D representation*** of the tripartite graph
  - Recommend items closer to compressed 2D space

Tag-Based Machine Learning Algorithms-2

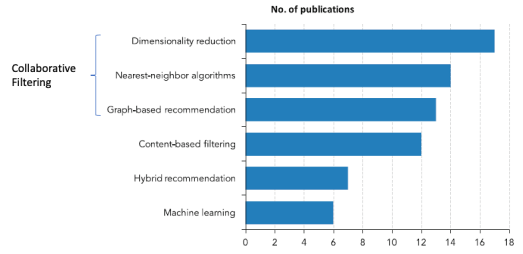
- **Sen et al.** proposed *tagommenders.*
- Stage 1: Infer ***user preferences for specific tags*** based on:
  - Searches for and clicks on specific tags
  - Ratings for movies tagged with specific tags
  - Bayesian generative model for predicting how users rate movies with specific tags
- Stage 2: Inferred tag preferences are used in five proposed recommendation algorithm.
  - 2 content-based algorithm to produce item rankings only, not ratings prediction.
    - Cosine similarity between user-tags and movie-tags
  - 3 algorithms attempt to predict the item's rating.



Hybrid Recommendation

- Hybrid recommenders combine aspects of different (types of) recommendation algorithms
  - Leverage strength of its components
  - Reduce weaknesses
- **Burke** provides a taxonomy of seven different methods for creating hybrid recommendation algorithms

Algorithm Frequencies in the Chapter



Hybridization method	Description
Mixed	Recommendations from several different recommenders are presented at the same time
Switching	The system switches between recommendation techniques depending on the current situation
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendations given by another
Feature augmentation	The output from one technique is used as an input feature to another technique
Meta-level	The model learned by one recommender is used as input to another
Weighted	The scores of several recommendation techniques are combined together to produce a single recommendation

Quick Examples

- **Bogers and Van den Bosch** compared 8 recommendation approaches with 6 different ***weighted*** combinations.
  - Used 3 different **results fusion** from field of IR
  - Experiments on BibSonomy, CiteULike and Delicious
  - Better to combine approaches that use **different data representations (tags+ metadata)** instead of only algorithmic variability
- **Gemmel et al.** propose a linearly weighted hybrid of four different NN algorithms:
  - User-based NN using R and user similarities by UT
  - Item-based NN using R and item similarities by IT
  - Single query tag => recommendation generation
  - Linearly weighted hybrid outperforms all individual algorithms consistently.

Conclusion

- A comprehensive overview of the most popular algorithms for item recommendation that incorporate tags
- Tag-augmented algorithms *tend* to provide better performance to state-of-the-art algorithms
  - Reporting negative results of tag-augmentation is rare.
- Unclear which algorithm is best-performing
  - Researchers had contradicting evaluation results of relative performances
- Three main issues:
  1. Lack of comparisons with state-of-the-art approaches
  2. Variation in data sets used for evaluation
  3. Lack of standardized evaluation setup for tag-based item recommendation.

Questions? Comments?