Chapter 15

Big Data Mining - Recommendation

Acknowledgements

- Singular value decomposition and principal component analysis. http://public.lanl.gov/mewall/kluwer2002.html
- Recommender system. https://en.wikipedia.org/wiki/Recommender_system
- Big Data Behind Recommender Systems. https://indatalabs.com/blog/data-science/big-data-behind-recommender-systems
- Special thanks to my junior grad fellow:

  Hui Ding (丁慧). School of Computer Science and Engineering, NEU.

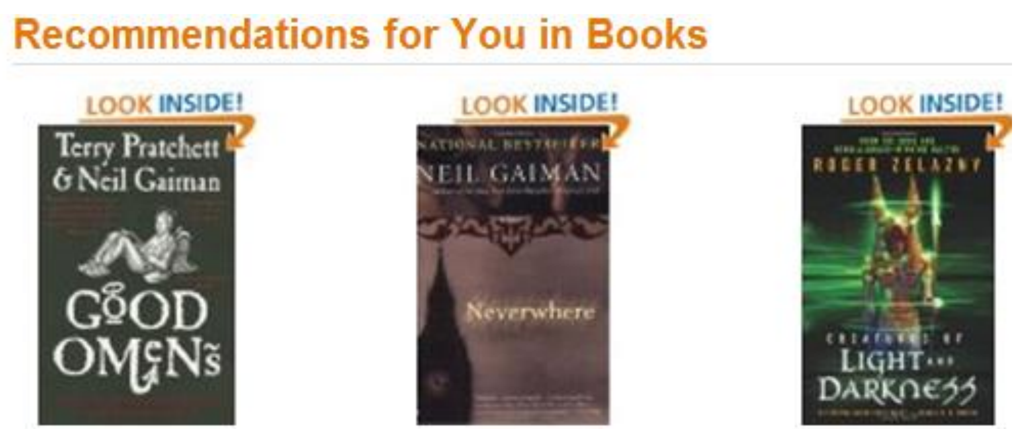
Chapter Outline

- What is a Recommender System ?
- Warmups
  - Collaborative Filtering
  - Content-based Recommendations
  - Hybrid Recommender Systems
- Advanced topics
  - Other Approaches
  - Evaluation of Recommender Systems

What Is A Recommender System ?

What is a Recommender System ?

- Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user.
- The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.



Types of Used Data

- User behavior data
  - on-site activity
    - clicks, searches, page and item views
  - off-site activities
    - tracking clicks in emails, in mobile applications and in their push notifications
- Particular item details
  - title
  - category
  - price
  - description
  - *etc.*
- Contextual information
  - device used
  - current location
  - referral URL

Why Using Recommender Systems?

- Value for the **customer**
  - Narrow down the set of choices
  - Discover new interesting things
- Value for the **provider**
  - Additional and probably unique personalized service for the customer
  - Increase the number of sold items
  - Sell more diverse items
  - Increase the user satisfaction
  - Increase trust and customer loyalty
  - Obtain more knowledge about customers

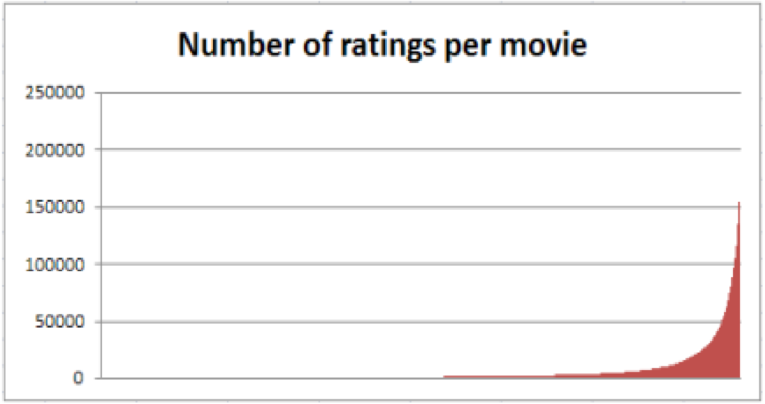
Challenges

- **Scalability**
  Millions of objects
  100s of millions of users

- **Cold start**
  New users
  New Items

- **Imbalanced dataset**
  User activity / item reviews
  are power law distributed

  (20%:80%)



Problem Definition

- C:= {users}
  S:= {recommendable items}
  u:= utility function, which measures the usefulness of items to user c,

$$u : C \times S \rightarrow R$$

where

$$R := \{\text{recommended items}\}$$

- For each user c, we want to choose the items s that maximize u

$$c \in C \quad s'_c = argmax_u u(c, s)$$

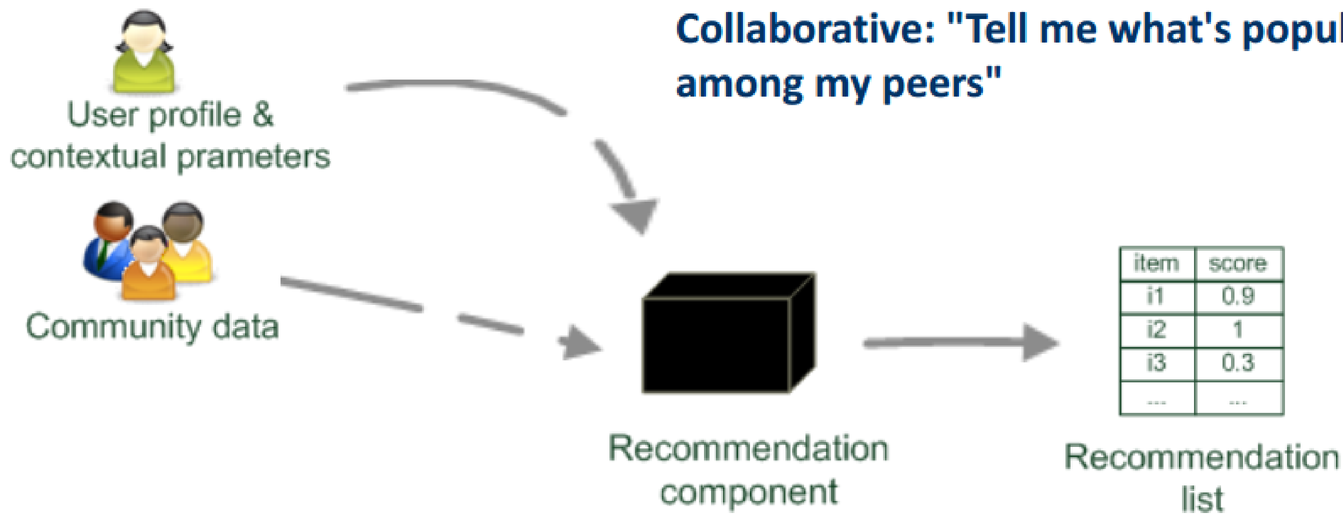
Warm-up Methods

Warm-ups

Collaborative Filtering
Content-based Recommendations
Hybrid Recommender Systems

Collaborative Filtering

Collaborative: "Tell me what's popular among my peers"



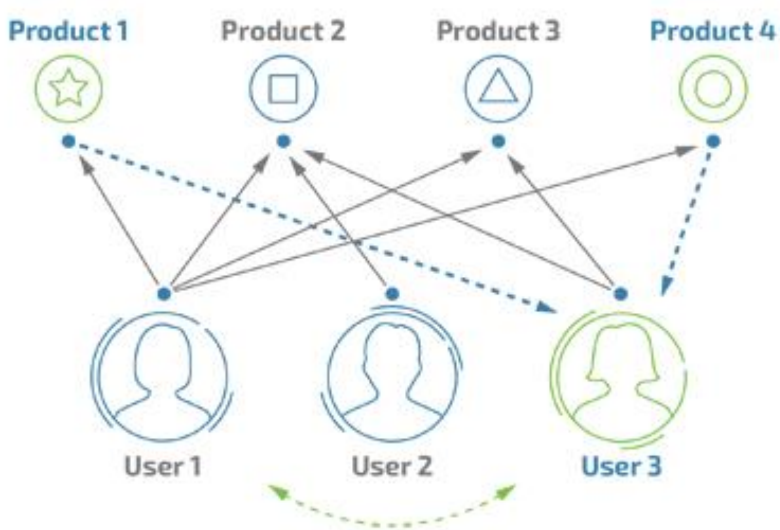
Collaborative Filtering

- The algorithm finds the user's preferences by mining the user's historical behavior data, groups the users based on different preferences, and recommends products with similar tastes.

- Collaborative filtering recommendation algorithm is divided into two categories:
  - **user-based collaborative filtering** algorithm
  - **item-based collaborative filtering** algorithm

User-based Collaborative Filtering

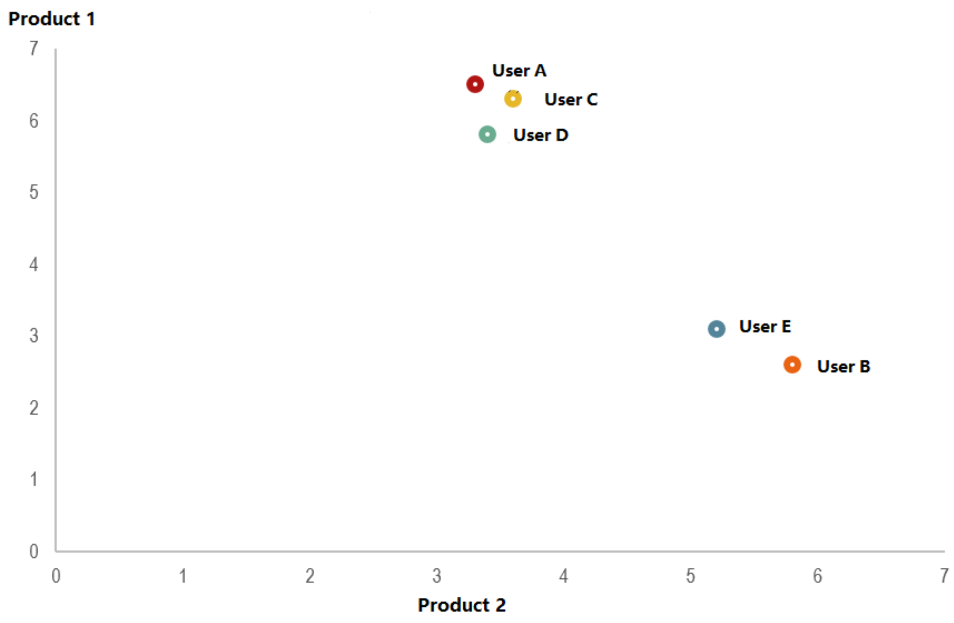
- The relationship between users is calculated based on the attitude and preference of different users to the same product or content. And then recommend products to users with the same preferences.



User-based Collaborative Filtering

- **Find users with similar preferences**

Users	Produc1	Produc2
User A	3.3	6.5
User B	5.8	2.6
User C	3.6	6.3
User D	3.4	5.8
User E	5.2	3.1



User-based Collaborative Filtering

- **Similarity measure**
  —Pearson correlation coefficient
  - Similarity Pearson *r* correlation *sim*(*u,v*) between users *u* & *v*

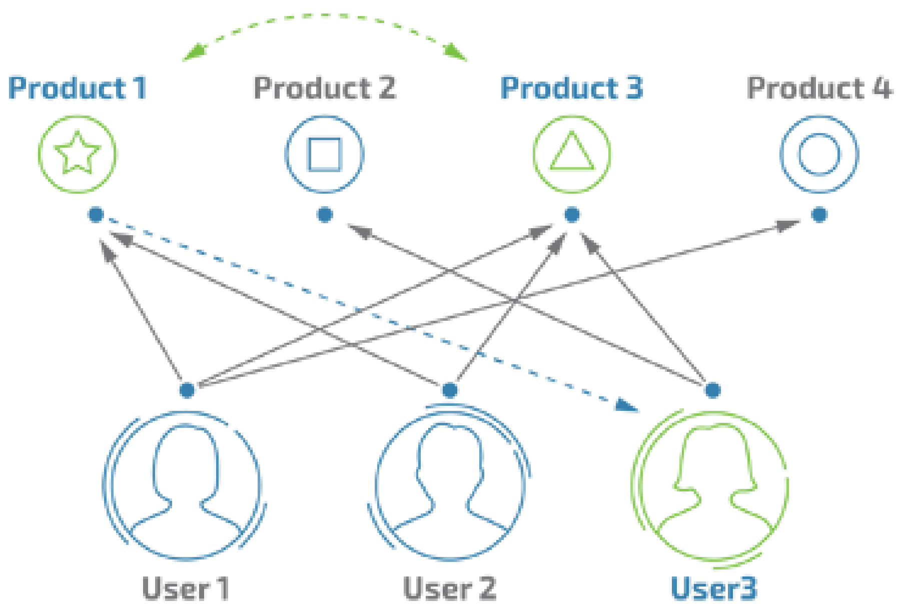
$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (y_{u,i} - \hat{y}_u)(y_{v,i} - \hat{y}_v)}{\sqrt{\sum_{i \in I_{uv}} (y_{u,i} - \hat{y}_u)^2 \sum_{i \in I_{uv}} (y_{v,i} - \hat{y}_v)^2}}$$

- **Predicted rating** $y^*(u,i)$

$$y^*(u,i) = \hat{y}_u + \frac{\sum_{j \in I_{y_{*j} \neq 0}} sim(v_j, u)(y_{v_j,i} - \hat{y}_{v_j})}{\sum_{j \in I_{y_{*j} \neq 0}} |sim(v_j, u)|}$$

Item-based Collaborative Filtering

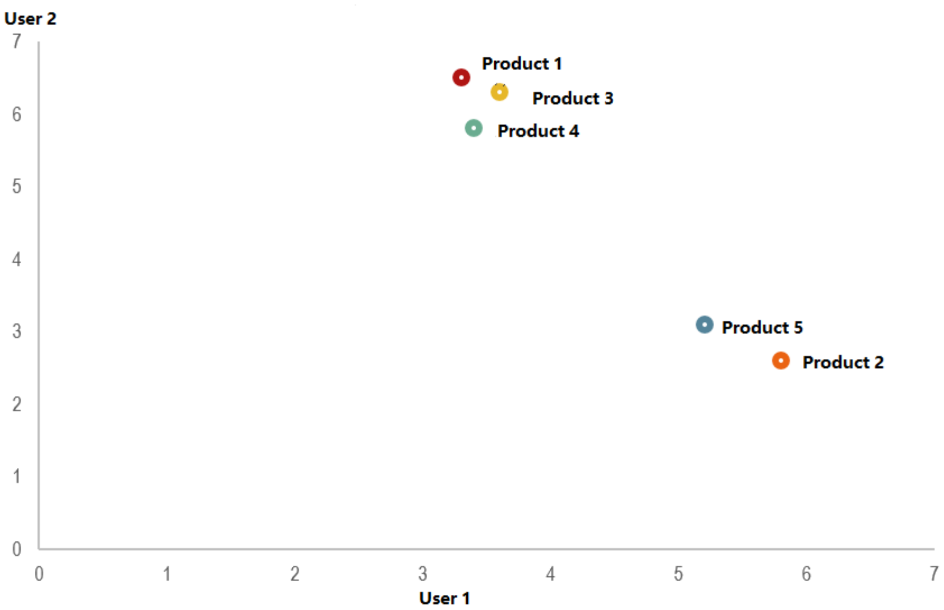
- The relationship between items is obtained by calculating the scores of different items by different users. And then recommend items to users with the similar items .



Item-based Collaborative Filtering

- **Find similar items**

Products	User1	User2
Product A	3.3	6.5
Product B	5.8	2.6
Product C	3.6	6.3
Product D	3.4	5.8
Product E	5.2	3.1



Item-based Collaborative Filtering

- **Similarity measure**
  —Pearson correlation coefficient
  - Similarity *sim*(*i*,*j*) between items *i* and *j* (Pearson correlation)

$$sim(i,j) = \frac{\sum_{u \in I_{ij}} (y_{u,i} - \hat{y}_i)(y_{u,j} - \hat{y}_j)}{\sqrt{\sum_{u \in I_{ij}} (y_{u,i} - \hat{y}_i)^2 \sum_{u \in I_{ij}} (y_{u,j} - \hat{y}_j)^2}}$$

- **Predicted rating** $y^*(u,i)$

$$y^*(u,i) = \hat{y}_i + \frac{\sum_{v \in I_{y_{u*} \neq 0}} sim(i, j_v)(y_{u,j_v} - \hat{y}_{j_v})}{\sum_{v \in I_{y_{u*} \neq 0}} |sim(i, j_u)|}$$

More On Ratings

- Pure CF-based systems only rely on the rating matrix
- **Explicit ratings**
  - Most commonly used (1 to 5, 1 to 7, *etc*.)
  - Research topics
    - "Optimal" granularity of scale
      - indication that 10-point scale is better accepted in movie domain
    - Multidimensional ratings (multiple ratings per movie)
  - Challenge
    - Users not always willing to rate many items; sparse rating matrices
    - How to stimulate users to rate more items?
- **Implicit ratings**
  - Clicks, page views, time spent on some page, demo downloads …
  - Can be used in addition to explicit ones; question of correctness of interpretation

Data Sparsity Problems

- **Cold start problem**
  - How to recommend new items?
  - What to recommend to new users?
- **Straightforward approaches**
  – Ask/force users to rate a set of items
  – Use another method (e.g., content-based, demographic or simply non-personalized) in the initial phase
- **Alternatives**
  – Use better algorithms (beyond nearest-neighbor approach)
  – Example:
    - In nearest-neighbor approaches, the set of sufficiently similar neighbors might be too small to make good predictions
    - Assume "transitivity" of neighborhoods

Memory & Model-based Approaches

- **User-based CF is said to be "memory-based"**
  – the rating matrix is directly used to find neighbors / make predictions
  – does not scale for most real-world scenarios
  – large e-commerce sites have tens of millions of customers and millions of items

- **Model-based approaches**
  – based on an offline pre-processing or "model-learning" phase
  – at run-time, only the learned model is used to make predictions
  – models are updated / re-trained periodically
  – large variety of techniques used
  – model-building and updating can be computationally expensive

More Model-based Approaches

- **Plethora of different techniques . e.g.**
  - Matrix factorization techniques, statistics

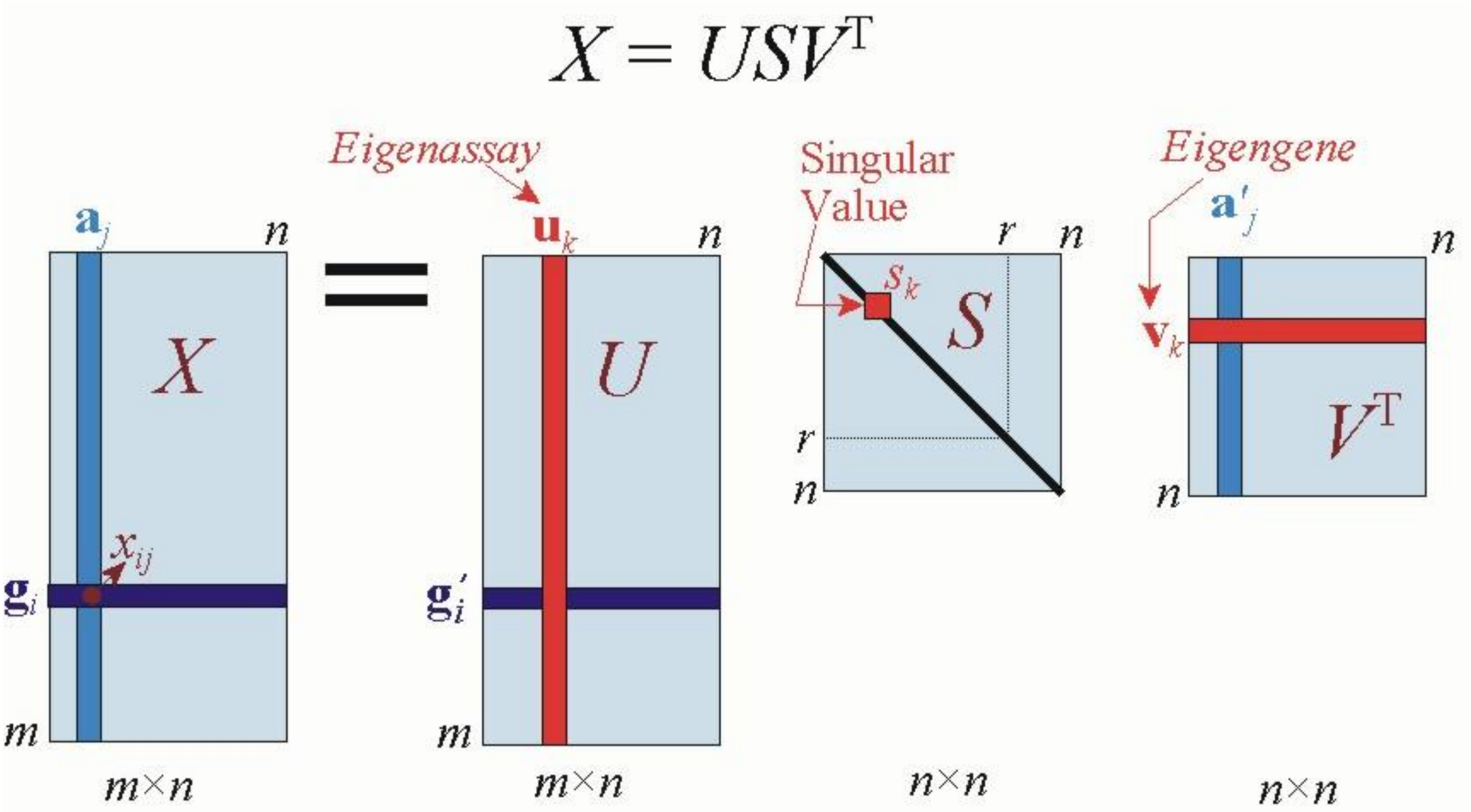
    singular value decomposition, principal component analysis
  - Association rule mining

    compare: shopping basket analysis
  - Probabilistic models

    clustering models, Bayesian networks, probabilistic Latent Semantic Analysis
  - Various other machine learning approaches
- **Costs of pre-processing**
  - Usually not discussed
  - Incremental updates possible?

Matrix Factorization

- Singular value decomposition



Matrix Factorization

- **SVD:**

$$M_k = U_k \times \Sigma_k \times V_k^T$$

U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

V_k^T					
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

- **Prediction:**

$$\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$$

= 3 + 0.84 = 3.84

Probabilistic Methods

- **Calculation of rating probabilities based on Bayes Theorem**

	Item1	Item2	Item3	Item4	Item5
Alice	1	3	3	2	?
User1	2	4	2	2	4
User2	1	3	3	5	1
User3	4	5	2	3	3
User4	1	1	5	2	1

– How probable is rating value "1" for Item5 given Alice's previous ratings?

– Corresponds to conditional probability P(Item1=1 | X),

where X = Alice s 's previous ratings = (Item1 =1, Item2 3, =3, Item3= …)

– Can be estimated based on Bayes' Theorem

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)} \qquad P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

– Assumption: Ratings are independent (?)

Probabilistic Methods

- **Calculation of rating probabilities based on Bayes Theorem**

$$
\begin{aligned}
P(X|\text{Item5}=1) &= P(\text{Item1}=1|\text{Item5}=1) \times P(\text{Item2}=3|\text{Item5}=1) \times \\
&\quad\; P(\text{Item3}=3|\text{Item5}=1) \times P(\text{Item4}=2|\text{Item5}=1) \\
&= 2/4 \times 1/4 \times 1/4 \times 1/4 \\
&\approx 0.0078125 \\
P(X|\text{Item5}=2) &= P(\text{Item1}=1|\text{Item5}=2) \times P(\text{Item2}=3|\text{Item5}=2) \times \\
&\quad\; P(\text{Item3}=3|\text{Item5}=2) \times P(\text{Item4}=2|\text{Item5}=2) \\
&= 0/4 \times \ldots \times \ldots \times \ldots. \\
&= 0
\end{aligned}
$$

- **Does not work in practice …**
  - Zeros (smoothing required), computationally expensive, …
  - like/dislike simplification possible
- **Practical probabilistic approaches**
  - Bayesian Networks, Probabilistic Latent Semantic Analysis, ….

Collaborative Filtering Issues

- **Pros:**
  – well-understood, works well in some domains, no knowledge engineering required
- **Cons:**
  – requires user community, sparsity problems, no integration of other knowledge sources, no explanation of results
- **What is the best CF method?**
  – In which situation and which domain?

    Inconsistent findings; always the same domains and data sets; Differences between methods are often very small (1/100)
- **How to evaluate the prediction quality?**
  – MAE / RMSE: What does an MAE of 0.7 actually mean?
  – Serendipity: Not yet fully understood
- **What about multi-dimensional ratings?**

Warm-ups

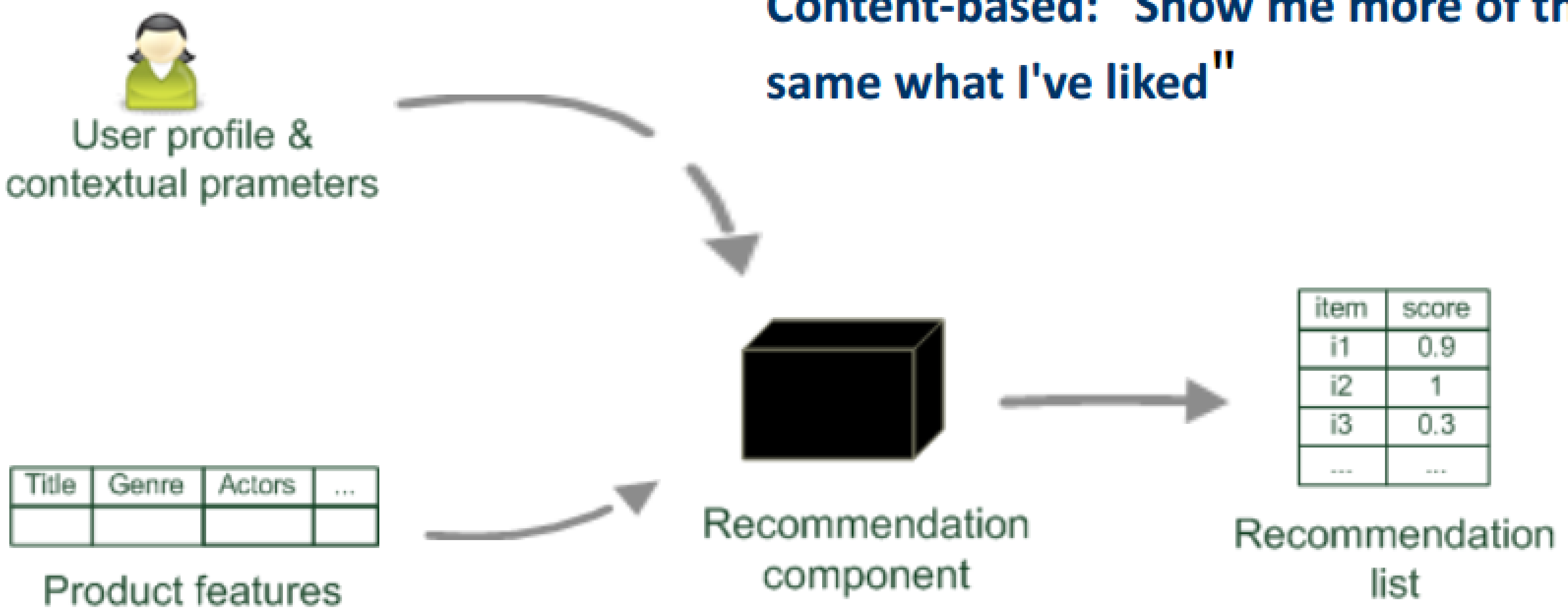
Collaborative Filtering

Content-based Recommendations

Hybrid Recommender Systems

Content-based Recommendation

Content-based: "Show me more of the same what I've liked"



Content-based Recommendation

- **While CF-methods do not require any information about the items**
  - it might be reasonable to exploit such information
  - recommend fantasy novels to people who liked fantasy novels in the past
- **What do we need**
  - some information about the available items such as the *genre* ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- **The task:**
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences

What Is "Content "?

- **Explicit attributes or characteristics**

  e.g. for a movie:
  Genre: Action / adventure
  Feature: Bruce Willis
  Year: 1995

- **Textual content**

  e.g. for a book:
  title,
  description,
  table of content

Content-based Recommendation

- Suitable for text-based products (web pages, books)
- Items are “described” by their features
  (e.g. keywords)
- Users are described by the keywords in the items they bought
- Recommendations based on the match between the content (item keywords) and user keywords
- The user model can also be a classifier (Neural Networks ,SVM , Naïve Bayes...)

Content-based Methods

- Content(s):= item profile,
  i.e. a set of attributes/keywords characterizing item s.
- weight w_{ij} measures the 'Importance” (or“informativeness”) of word k_j in document d_j
- term frequency/inverse document frequency (TF-IDF) is a popular weighting technique in IR

An (unrealistic) Example

COUNT	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Management	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				1			1		1		1	1							1										
Accelerating customer relationships: using CRM and relationship technologies		1	1						1	1											2			1			1		
Mastering Data Mining: the art and science of Customer Relationship Management			1		1					1	1					1		1	1	1	1		1		1				
Data Mining your website											1								1									1	1
Introduction to Marketing														1			1									1			
Consumer behavior						1		1																					
Marketing Research: a Handbook	1												1				1					1							
Customer Knowledge Management										1					1	1													

An (unrealistic) Example

TFIDF Normed Vectors	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Managemen	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				0.502			0.502		0.344		0.251	0.502							0.251										
Accelerating customer relationships: using CRM and relationship technologies		0.432	0.296						0.296	0.216											0.468			0.432			0.432		
Mastering Data Mining: the art and science of Customer Relationship Management			0.256		0.374					0.187	0.187					0.256		0.374	0.187	0.374	0.256		0.374		0.374				
Data Mining your website											0.316								0.316									0.632	0.632
Introduction to Marketing														0.636			0.436									0.636			
Consumer behavior						0.707		0.707																					
Marketing Research: a Handbook	0.537												0.537				0.368					0.537							
Customer Knowledge Management										0.381					0.736	0.522													

An (unrealistic) Example

How to compute recommendations of books based only on their title?

- A customer buys the book: *Building data mining applications for CRM*

- 7 Books are possible candidates for a recommendation:

  Accelerating Customer Relationships: Using CRM and Relationship Technologies
  Mastering Data Mining: The Art and Science of Customer Relationship Management
  Data Mining Your Website
  Introduction to marketing
  Consumer behaviour
  Marketing research, a handbook
  Customer knowledge management

An (unrealistic) Example

- Computes distances between this book & all others

- Recommends the closest books:

  #1: Data Mining Your Website
  #2: Accelerating Customer Relationships: Using CRM and Relationship Technologies
  #3: Mastering Data Mining: The Art and Science of Customer Relationship Management

Advantages of CB Approach

- No need for data on other users
- No cold-start or sparsity problems
- Can recommend to users with unique tastes
- Can recommend new and unpopular items
- Can provide explanations of recommended items by listing content-features that caused an item to be recommended

Disadvantages of CB Approach

- Only for content that can be encoded as meaningful features
- Some types of items (e.g. movies, music)are not amenable to easy feature extraction methods
- Even for texts, IR techniques cannot consider multimedia information, aesthetic qualities, download time: a positive rating could be not related to the presence of certain keywords
- Users’ tastes must be represented as a learnable function of these content features
- Hard to exploit quality judgements of other users
- Difficult to implement serendipity

Warm-ups

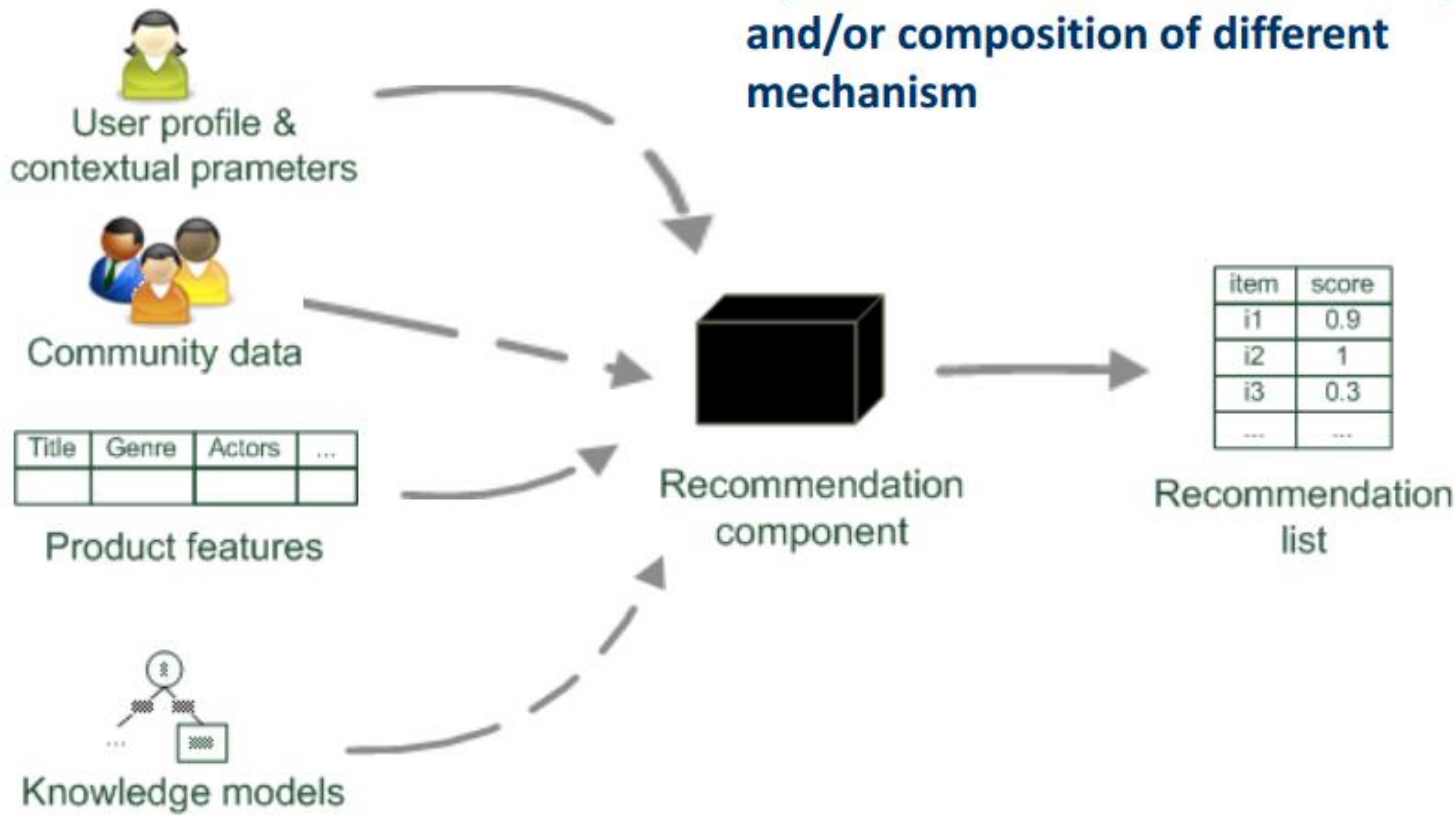
Collaborative Filtering

Content-based Recommendations

Hybrid Recommender Systems

Hybrid Recommendation

Hybrid: combinations of various inputs and/or composition of different mechanism

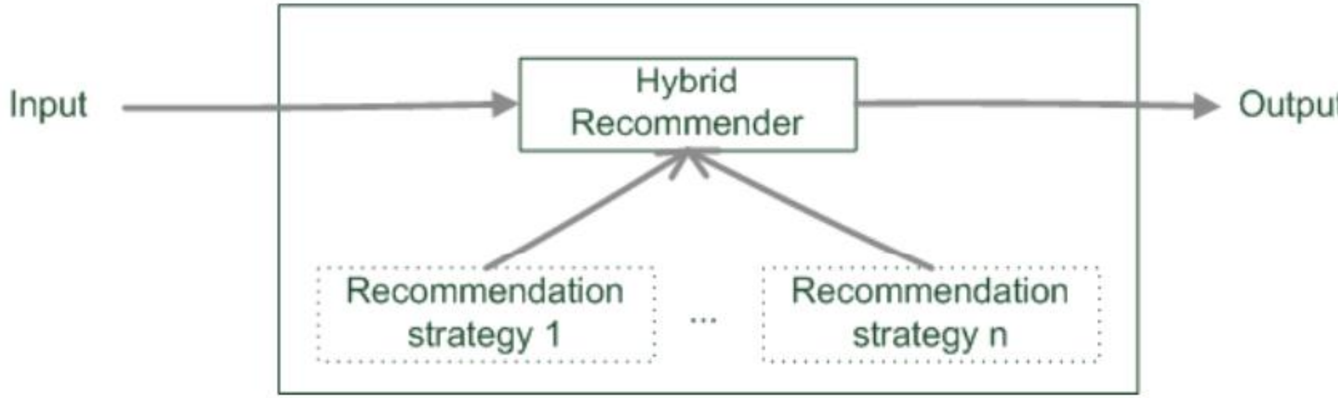


Hybrid Recommender Systems

- **All three base techniques are naturally incorporated by a good sales assistance (at different stages of the sales act) but have their shortcomings**
- **Idea of crossing two (or more) species/implementations**
  – *hybrida* [lat.]: denotes an object made by combining two different elements
  – Avoid some of the shortcomings
  – Reach desirable properties not (or only inconsistently) present in parent individuals
- **Different hybridization designs**
  – Parallel use of several systems
  – Monolithic exploiting different features

Monolithic Hybridization Design

- **Only a single recommendation component**



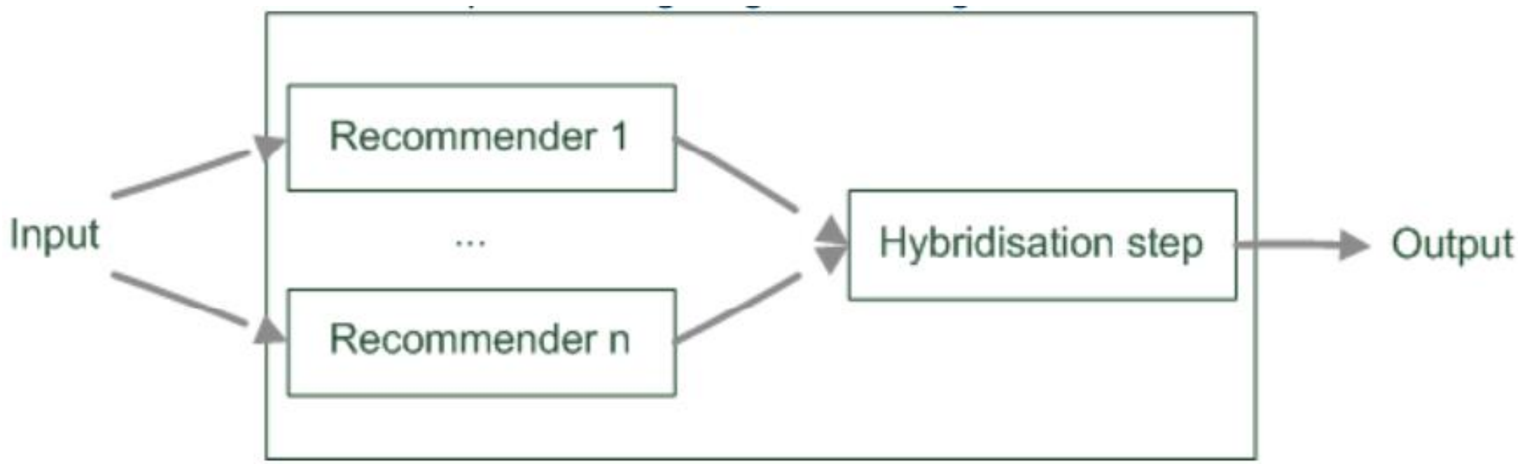
- **Hybridization is 'virtual' in the sense that**
  – Features/knowledge sources of different paradigms are combined

Feature Combination

- **Combination of several knowledge sources**
  – E.g.: Ratings and user demographics or explicit requirements and needs used for similarity computation

- **'Hybrid' content features:**
  – Social features: Movies liked by user
  – Content features: Comedies liked by user, dramas liked by user
  – Hybrid features: user likes many movies that are comedies, …

Parallelized Hybridization Design



- **Output of several existing implementations combined**
- **Least invasive design**
- **Some weighting or voting scheme**
  - Weights can be learned dynamically
  - Extreme case of dynamic weighting is switching

Weighted

- Compute weighted sum:

$$rec_{weighted}(u,i) = \sum_{k=1}^{n} \beta_k \times rec_k(u,i)$$

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender weighted (0.5:0.5)		
Item1	0,65	1
Item2	0,45	2
Item3	0,35	3
Item4	0,05	4
Item5	0,00	

Limitations

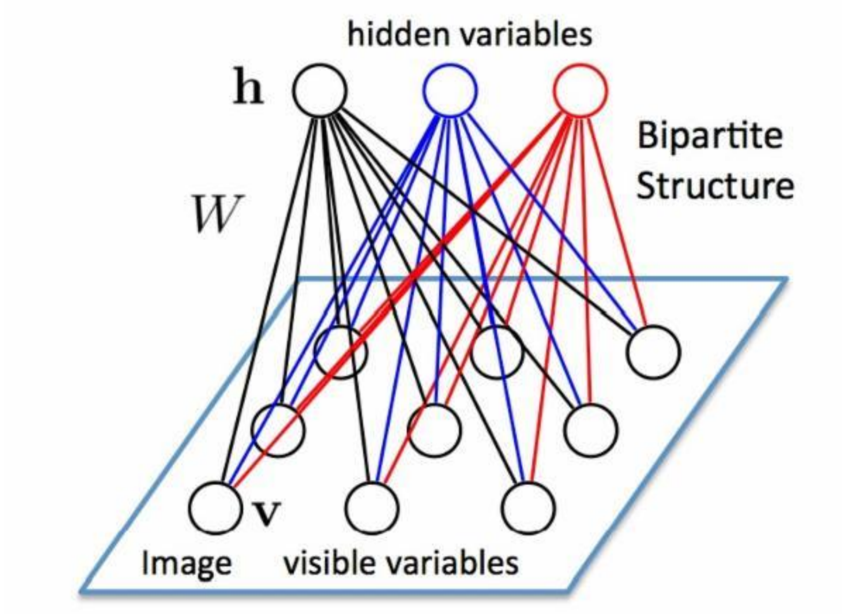
- **Only few works that compare strategies from the meta-perspective**
  – Most datasets do not allow to compare different recommendation paradigms
    - i.e. ratings, requirements, item features, domain knowledge, critiques rarely available in a single dataset
  – Thus few conclusions that are supported by empirical findings
    - Monolithic: some preprocessing effort traded-in for more knowledge included
    - Parallel: requires careful matching of scores from different predictors

- **Netflix competition – "stacking" recommender systems**
  – Weighted design based on >100 predictors – recommendation functions

Other Approaches
Evaluation

Restricted Boltzmann Machines

- Each unit is in a state which can be active or not active
- Each input of a unit is associated to a weight
- The transfer function **Σ** calculates for each unit a score based on the weighted sum of the inputs
- This score is passed to the activation function φ which calculated the probability that the unit state is active

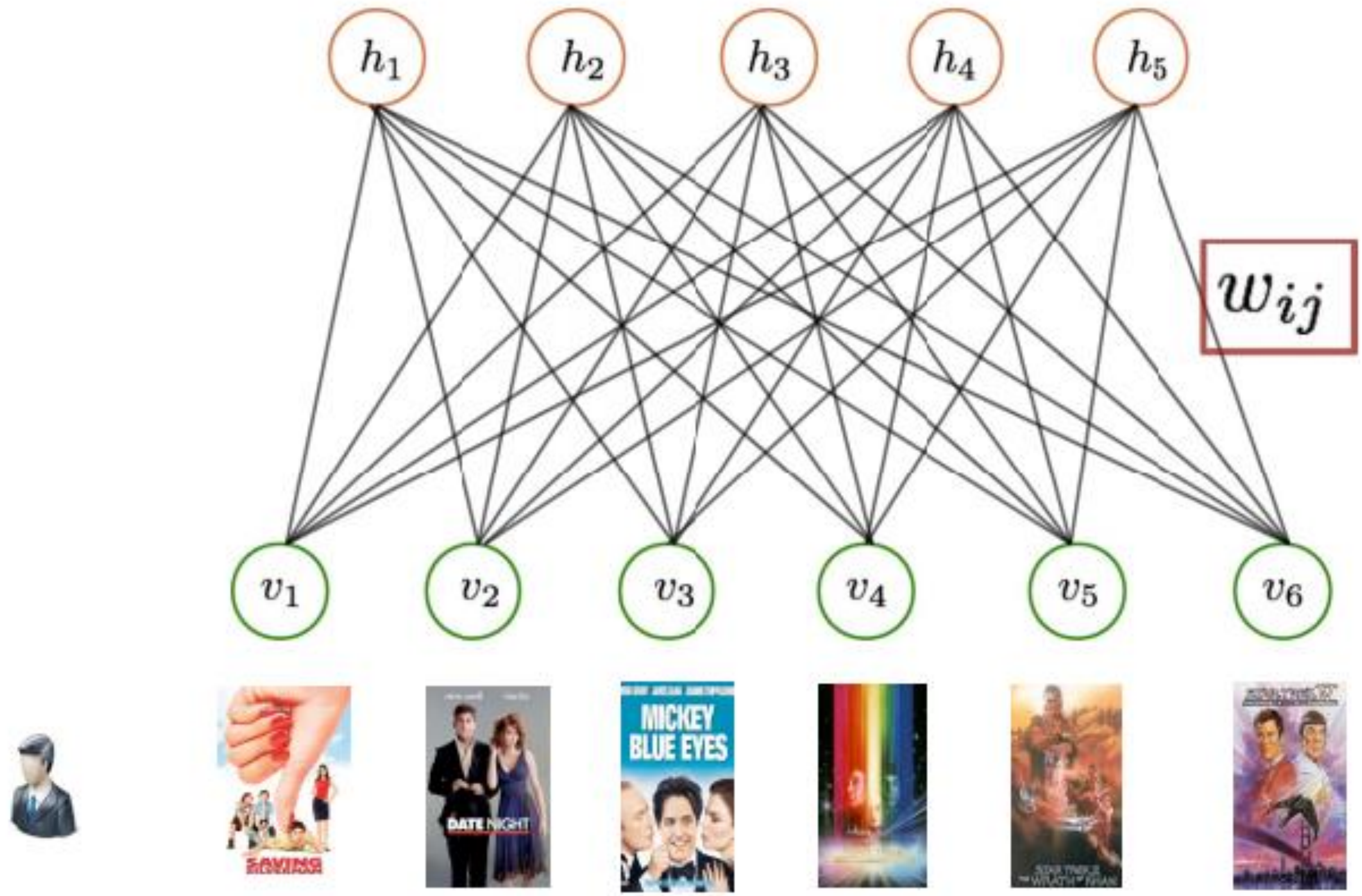


Restricted Boltzmann Machines

- Each unit in the visible layer v_i corresponds to one item
- The number of the hidden units h_j is a parameter
- Each v_i is connected to each h_j through a weight w_{ij}

- In the training phase, for each user:
  - if the user purchased the item the corresponding v_i is activated.
  - The activation states of all v_i are the input of each h_j
  - Based on this input the activation state of each h_j is calculated
  - The activation state of all h_j become now the input of each v_i
  - The activation state of each v_i is recalculated
  - For each v_i the difference between the present activation state and the previous is used to update the weights w_{ij} and thresholds ϑ_j

Restricted Boltzmann Machines



Restricted Boltzmann Machines

- In the prediction phase, using a trained RBM, when recommending to a user:
  - For the items of the user the corresponding v_i is activated.
  - The activation states of all v are the input of each h_j
  - Based on this input the activation state of each h_j is calculated
  - The activation state of all h_j become now the input of each v_i
  - The activation state of each v_i is recalculated
  - The activation probabilities are used to recommend items

Advanced topics

Other Approaches

Evaluation

Evaluation of Recommender Systems

- **What are the measures in practice?**

  Total sales numbers
  Promotion of certain items
  ...
  Click-through-rates
  Interactivity on platform
  ...
  Customer return rates
  Customer satisfaction and loyalty

When does a RS do its job well?

- "Recommend widely unknown items that users might actually like!"
- 20% of items accumulate 74% of all positive ratings

Purpose And Success Criteria

- **Different perspectives/aspects**
  – Depends on domain and purpose
  – No wholistic evaluation scenario exists

- **Retrieval perspective**
  – Reduce search costs
  – Provide correct proposals
  – Users know in advance what they want

- **Recommendation perspective**
  – Serendipity - identify items from the Long Tail
  – Users did not know about existence

Purpose And Success Criteria

- **Prediction perspective**
  - Predict to what degree users like an item
  - Most popular evaluation scenario in research

- **Interaction perspective**
  - Give users a good feeling
  - Educate users about the product domain
  - Convince/persuade users - explain

- **Finally, conversion perspective**
  - Commercial situations
  - Increase ‚hit‘, clickthru‘, lookers to bookers‘ rates
  - Optimize sales margins and profit

End of Chapter 15