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
  - Default voting: assign default values to items that only one of the two users to be compared has rated (Breese et al. 1998)
- **Alternatives**
  - Use better algorithms (beyond nearest-neighbor approaches)
  - Example:
    - In nearest-neighbor approaches, the set of sufficiently similar neighbors might be too small to make good predictions
    - Assume "transitivity" of neighborhoods

Example algorithms for sparse datasets

- **Recursive CF** (Zhang and Pu 2007)
  - Assume there is a very close neighbor n of u who however has not rated the target item i yet.
  - Idea:
    - Apply CF-method recursively and predict a rating for item i for the neighbor
    - Use this predicted rating instead of the rating of a more distant direct neighbor

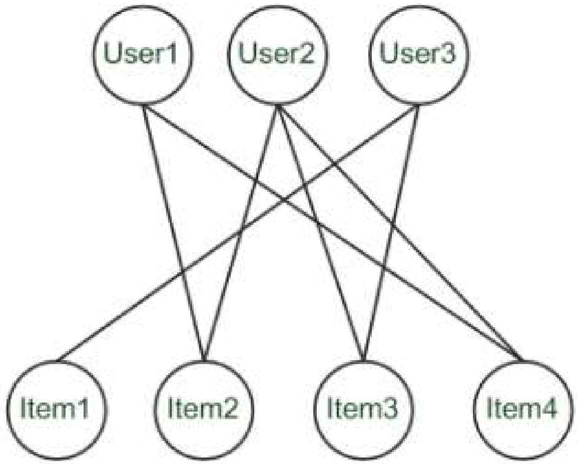
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	?
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0.85

Predict rating for User1

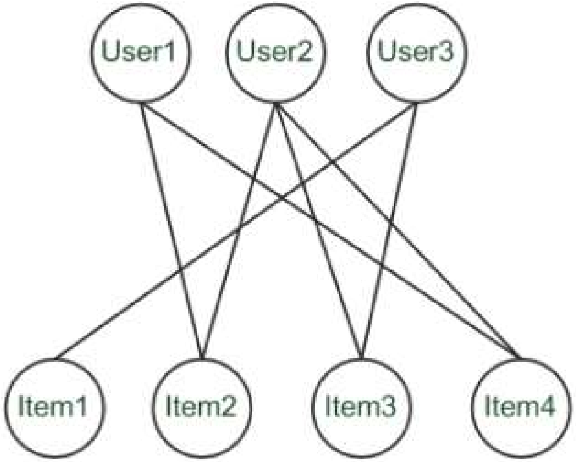
Graph-based methods (1)

- **"Spreading activation"** (Huang et al. 2004)
  - Exploit the supposed "transitivity" of customer tastes and thereby augment the matrix with additional information
  - Assume that we are looking for a recommendation for *User1*
  - When using a standard CF approach, *User2* will be considered a peer for *User1* because they both bought *Item2* and *Item4*
  - Thus *Item3* will be recommended to *User1* because the nearest neighbor, *User2*, also bought or liked it



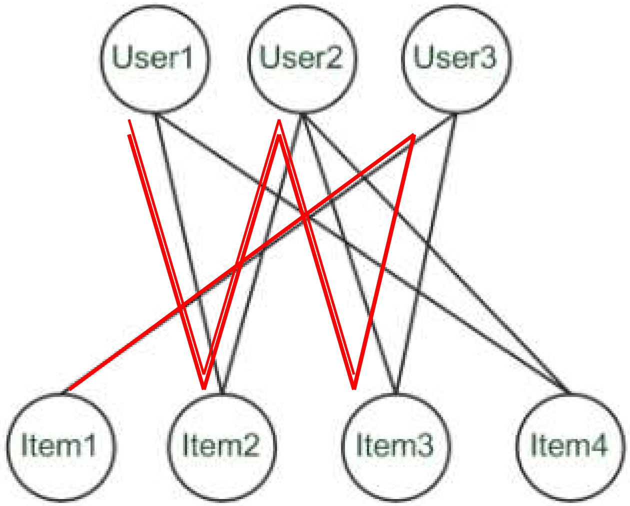
Graph-based methods (2)

- **"Spreading activation"** (Huang et al. 2004)
  - In a standard user-based or item-based CF approach, paths of length 3 will be considered – that is, *Item3* is relevant for *User1* because there exists a three-step path (*User1–Item2–User2–Item3*) between them
  - Because the number of such paths of length 3 is small in sparse rating databases, the idea is to also consider longer paths (indirect associations) to compute recommendations
  - Using path length 5, for instance



Graph-based methods (3)

- **"Spreading activation"** (Huang et al. 2004)
  - Idea: Use paths of lengths > 3
    to recommend items
  - Length 3: Recommend Item3 to User1
  - Length 5: Item1 also recommendable



More model-based approaches

- **Plethora of different techniques proposed in the last years, 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?

2000: *Application of Dimensionality Reduction in Recommender System*, B. Sarwar et al., WebKDD Workshop

- **Basic idea: Trade more complex offline model building for faster online prediction generation**
- **Singular Value Decomposition for dimensionality reduction of rating matrices**
  - Captures important factors/aspects and their weights in the data
  - factors can be genre, actors but also non-understandable ones
  - Assumption that k dimensions capture the signals and filter out noise (K = 20 to 100)
- **Constant time to make recommendations**
- **Approach also popular in IR (Latent Semantic Indexing), data compression,…**

Matrix factorization

- Informally, the SVD theorem (Golub and Kahan 1965) states that a given matrix M can be decomposed into a product of three matrices as follows

$$M = U \times \Sigma \times V^T$$

  - where U and V are called *left* and *right singular vectors* and the values of the diagonal of Σ are called the *singular values*

- We can approximate the full matrix by observing only the most important features – those with the largest singular values

- In the example, we calculate U, V, and Σ (with the help of some linear algebra software) but retain only the two most important features by taking only the first two columns of U and V^T

Example for SVD-based recommendation

- **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

V_k^T	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

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

  = 3 + 0.84 = 3.84

Discussion about dimensionality reduction (Sarwar et al. 2000a)

- **Prediction quality can decrease because…**
  - the original ratings are not taken into account
- **Prediction quality can increase as a consequence of…**
  - filtering out some "noise" in the data and
  - detecting nontrivial correlations in the data
- **Depends on the right choice of the amount of data reduction**
  - number of singular values in the SVD approach
  - Parameters can be determined and fine-tuned only based on experiments in a certain domain
  - Koren et al. 2009 talk about 20 to 100 factors that are derived from the rating patterns

Association rule mining

- **Commonly used for shopping behavior analysis**
  - aims at detection of rules such as
    "If a customer purchases beer then he also buys diapers in 70% of the cases"
- **Association rule mining algorithms**
  - can detect rules of the form X → Y (e.g., beer → diapers) from a set of sales transactions $D = \{t_1, t_2, \dots t_n\}$
  - measure of quality: support, confidence
    - used e.g. as a threshold to cut off unimportant rules
  - let $\sigma(X) = \frac{|\{x \mid x \subseteq ti,\ ti \in D\}|}{|D|}$
  - support = $\frac{\sigma(X \cup Y)}{|D|}$, confidence = $\frac{\sigma(X \cup Y)}{\sigma(X)}$

Recommendation based on Association Rule Mining

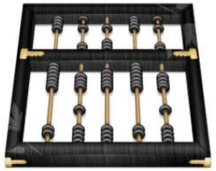
	Item1	Item2	Item3	Item4	Item5
Alice	1	0	0	0	?
User1	1	0	1	0	1
User2	1	0	1	0	1
User3	0	0	0	1	1
User4	0	1	1	0	0

- **Simplest approach**
  - transform 5-point ratings into binary ratings (1 = above user average)
- **Mine rules such as**
  - Item1 → Item5
    - support (2/4), confidence (2/2) (without Alice)
- **Make recommendations for Alice (basic method)**
  - Determine "relevant" rules based on Alice's transactions (the above rule will be relevant as Alice bought Item1)
  - Determine items not already bought by Alice
  - Sort the items based on the rules' confidence values

Probabilistic methods

- **Basic idea (simplistic version for illustration):**
  - given the user/item rating matrix
  - determine the probability that user Alice will like an item i
  - base the recommendation on such these probabilities
- **Calculation of rating probabilities based on Bayes Theorem**
  - How probable is rating value "1" for Item5 given Alice's previous ratings?
  - Corresponds to conditional probability P(Item5=1 | X), where
    - X = Alice's previous ratings = (Item1 =1, Item2=3, Item3= …)
  - Can be estimated based on Bayes' Theorem

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



  - Assumption: Ratings are independent (?)

Calculation of probabilities in simplistic approach

	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

X = (Item1 =1, Item2=3, Item3= …)

$$\boldsymbol{P(X|Item5 = 1)}$$
$$\boldsymbol{= P(Item1 = 1|Item5 = 1) \times P(Item2 = 3|Item5 = 1)}$$
$$\boldsymbol{\times P(Item3 = 3|Item5 = 1) \times P(Item4 = 2|Item5 = 1) = \frac{2}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}}$$
$$\boldsymbol{\approx 0.125}$$

$$\boldsymbol{P(X|Item5 = 2)}$$
$$\boldsymbol{= P(Item1 = 1|Item5 = 2) \times P(Item2 = 3|Item5 = 2)}$$
$$\boldsymbol{\times P(Item3 = 3|Item5 = 2) \times P(Item4 = 2|Item5 = 2) = \frac{0}{0} \times \cdots \times \cdots \times \cdots}$$
$$\boldsymbol{= 0}$$

- **More to consider**
  - Zeros (smoothing required)
  - like/dislike simplification possible

Practical probabilistic approaches

- **Use a cluster-based approach** (Breese et al. 1998)
  - assume users fall into a small number of subgroups (clusters)
  - Make predictions based on estimates
    - probability of Alice falling into cluster c
    - probability of Alice liking item i given a certain cluster and her previous ratings
    - $P(C = c, v_1, \ldots, v_n) = P(C = c) \prod_{i=1}^{n} P(v_i | C = c)$
  - Based on model-based clustering (mixture model)
    - Number of classes and model parameters have to be learned from data in advance (EM algorithm)
- **Others:**
  - Bayesian Networks, Probabilistic Latent Semantic Analysis, ….
- **Empirical analysis shows:**
  - Probabilistic methods lead to relatively good results (movie domain)
  - No consistent winner; small memory-footprint of network model

Slope One predictors (Lemire and Maclachlan 2005)

- **Idea of Slope One predictors is simple and is based on a *popularity differential* between items for users**
- **Example:**

	Item1	Item5
Alice	2	?
User1	1	2

-

- **p(Alice, Item5) = 2 + (2 - 1) = 3**
- **Basic scheme: Take the average of these differences of the co-ratings to make the prediction**
- **In general: Find a function of the form f(x) = x + b**
  - **That is why the name is "Slope One"**

RF-Rec predictors (Gedikli et al. 2011)

- **Idea: Take rating frequencies into account for computing a prediction**
- **Basic scheme:** $\hat{r}_{u,i} = \arg\max_{v \in R} f_{user}(u,v) * f_{item}(i,v)$
  - R: Set of all rating values, e.g., $R = \{1,2,3,4,5\}$ on a 5-point rating scale
  - $f_{user}(u,v)$ and $f_{item}(i,v)$ basically describe *how often* a rating v was assigned by user u and to item i resp.
- **Example:**

	Item1	Item2	Item3	Item4	Item5
Alice	1	1	?	5	4
User1	2		5	5	5
User2			1	1	
User3		5	2		2
User4	3		1	1	
User5	1	2	2		4

- **p(Alice, Item3) = 1**

MAE

- **Metrics measure error rate**
  - **Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

  - **Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation**

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

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 (novelty and surprising effect of recommendations)
    - Not yet fully understood
- **What about multi-dimensional ratings?**