Lec 03. Model-based Collaborative Filtering

Contents

1. Model-based methods basics
2. Association rule mining
3. Probabilistic models
4. Slope one predictions

---

1. Model-based Methods Basics

Why are Model-based Methods?

- CF can be interpreted as the conventional classification or regression problem.
  - We have an m x n matrix, in which n-1 columns are **feature variables** and the last column is **a label variable.**
- Model-based methods are created from a rating matrix using **supervised** or **unsupervised** methods.
  - The training phase is clearly separated from the prediction phase.
- Examples
  - Rule-based methods, Bayes classifiers, regression models
  - Latent factor models

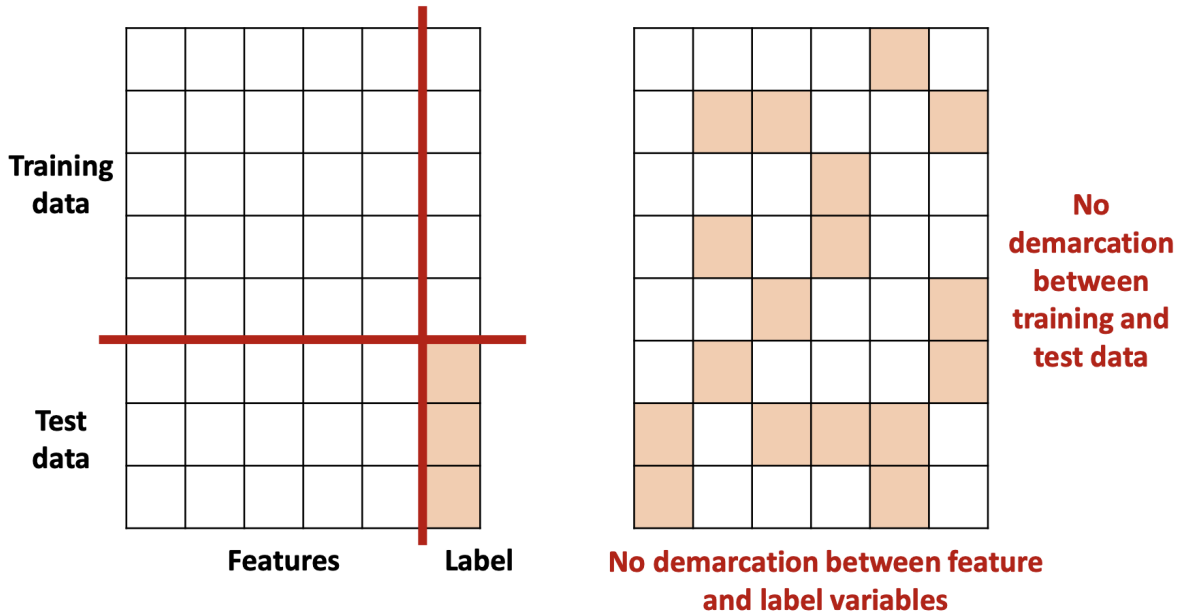
Recap : User-Item Rating Matrix

- We are given a user-item rating matrix $R \in R^{m*n}$.
  - R : a user-item rating matrix (m x n matrix)

- Predict the ratings of missing items by users.

Classification vs. Matrix Completion

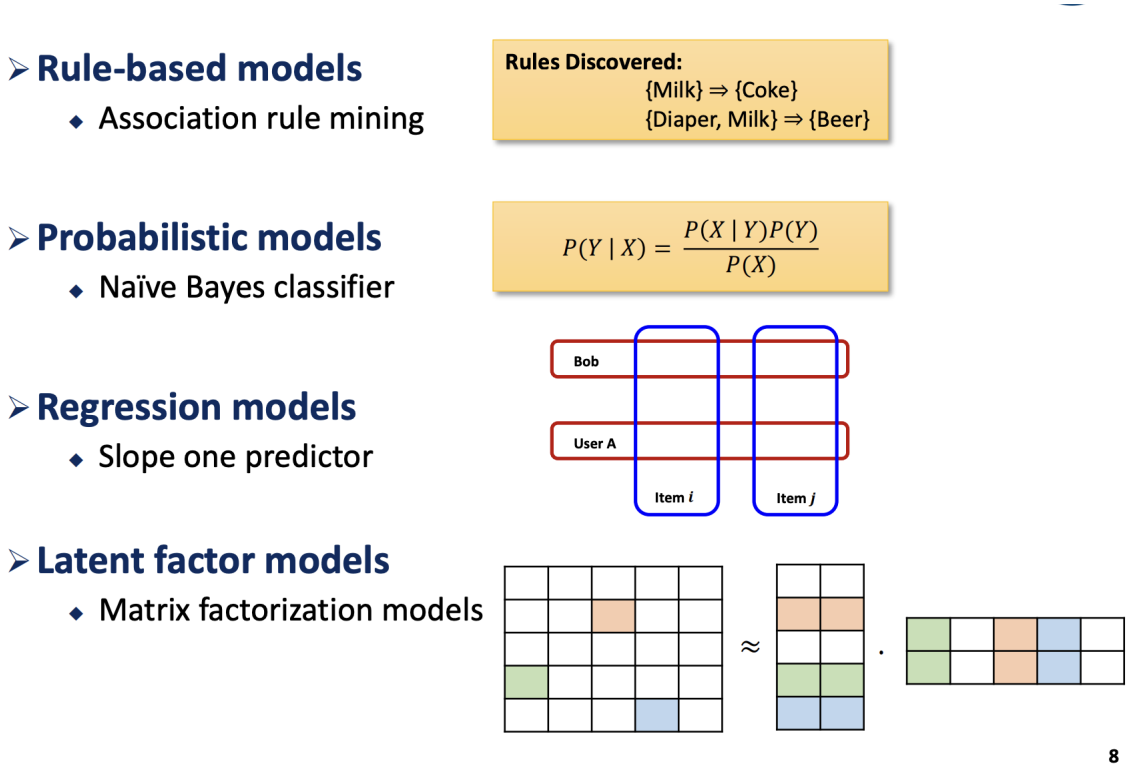
- Some entries in the rating matrix may be missing.



Properties of Model-based Methods

- Neighborhood-base methods require **quadratic time complexity** for # of users or # of items.
- Advantages
  - Training and efficiency : it is usually faster in the pre-processing phase of building the trained model.
  - Space-efficiency : the size of the learned model is much smaller than the original rating matrix.
- How to **generalize existing classification models** to the **matrix completion problem?**
  - It is crucial to handle missing entries in the rating matrix.

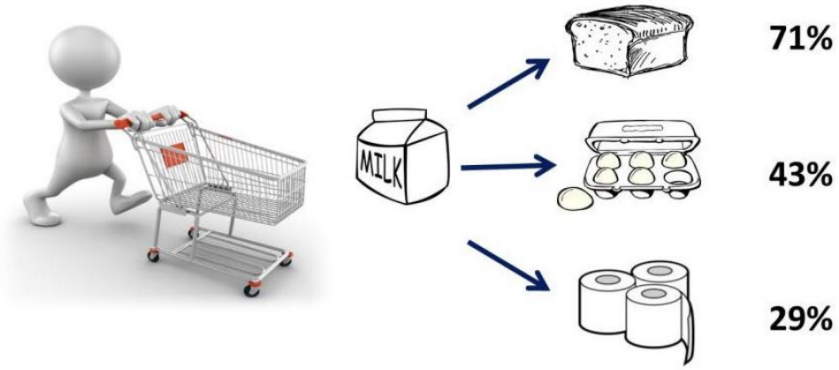
Example : Model-based Methods



2. Rule-based Models

What is Association Rule Mining?

- A common technique used to identify **rule-like relationships** in large transcation data
  - It is also called **frequent pattern ananysis.**
  - What items are **frequently purchased together?**
- Example
  - Diaper ⇒ Beer [0.5, 0.75] **(support, confidence)**



- Finding inherent rules from data
  - What products are often purchased together?
    - Beer and diapers?
  - What are the subsequent purchases after buying an item?
- Applications
  - Basket data analysis, cross-marketing
  - Catalog design, sale campaign analysis
  - Web log (click stream) analysis
  - DNA sequence anaysis

Baskedt Model

- We are given a large set of baskets.
  - Items are products sold in a supermarket.

- Each basket is a set of items, e.g., the products that a customer buys for one transaction.

- **Goal : we want to discover association rules, i.e., if-then rule!**

Input:

Tid	Items
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs

Output:

Rules Discovered:
{Milk} ⇒ {Coke}
{Diaper, Milk} ⇒ {Beer}

Example: Basket Model

- **Items = products**
- **Baskets = sets of products someone bought**
- **Amazon's people who bought X also bought Y.**
- **Applications**
  - Tells me how typical **customers navigate stores** and **lets them position tempting items**.
  - Run sales on **diapers** and raise the price of **beer**.

Terminology

- Itemset : a set of items, i.e., **k-itemset** $X = \{X_1, \ldots, X_k\}$

- Support
  - (Absolute)support or support count of X
    - The frequency of occurring an itemset X
  - (Relative) support is the fraction of transactions with X
    - The probability that a transaction contains X
- Minimum support : an itemset X is frequent if X's support is no less than a aminimum support threshold.
  - It is pre-defined before building a model.

➢ **Goal: Finding all rules $X \Rightarrow Y$ with minimum support and confidence**

➢ **Support: the probability of transactions for $X \Rightarrow Y$**

$$support(X \Rightarrow Y) = P(X,Y) = \frac{\#\ of\ transactions\ containing\ X\ and\ Y}{\#\ of\ total\ transactions}$$

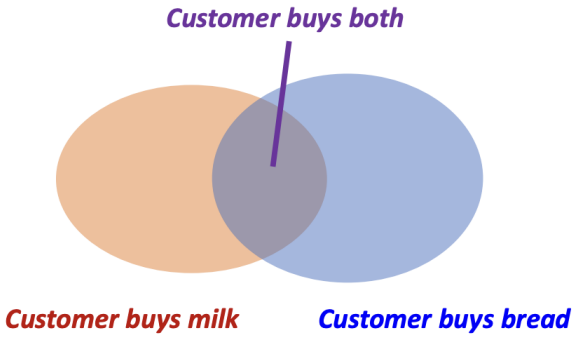
➢ **Confidence: the conditional probability for $X \Rightarrow Y$**

$$confidence(X \Rightarrow Y) = P(Y \mid X) = \frac{\#\ of\ transactions\ containing\ X\ and\ Y}{\#\ of\ transactions\ containing\ X}$$

Example : Frequent Patterns

- **Milk ⇒ Bread (support: 0.6, confidence: 1.0)**
- **Bread ⇒ Milk (support: 0.6, confidence: 0.75)**

Tid	Items bought
10	Milk, Nuts, Bread
20	Milk, Coffee, Bread
30	Milk, Bread, Eggs
40	Nuts, Eggs, Coffee
50	Nuts, Coffee, Bread



Leveraging Association Rules for CF

- Step1 : discover all the association rules at a pre-specified level of minimum support and minimum confidence.
- Step 2 : the set of rules is the model, which is used to perform recommendations for a specific user.
- Given a target user, find all relevant association rules.
  - If the item in the antecedent X of the rule is a subset of items preferred by the user, the association rule is fired.
  - All the fired rules are sorted by decreasing order of confidence.
  - The first k items are used for a recommendation.

➢ **Example**

- Minimum support: 0.5, minimum confidence: 0.6

Tid	Items bought
10	Beer, Egg, Diaper
20	Beer, Coffee
30	Beer, Diaper, Milk
40	Milk, Diaper
50	Beer, Egg

Output:

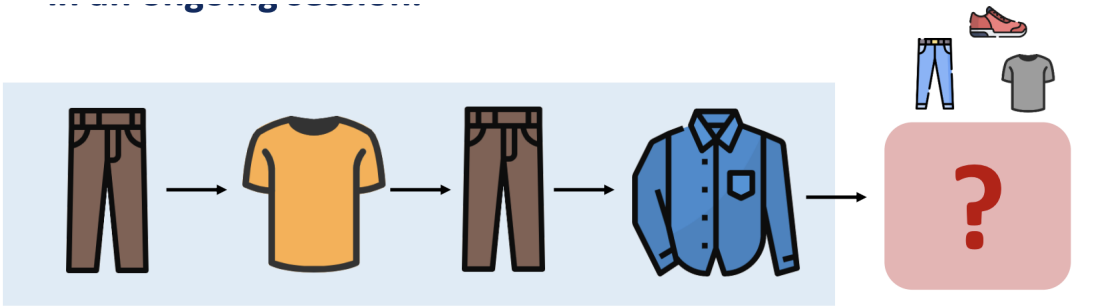
Rules Discovered:
{Diaper} ⇒ {Beer}

➢ **If a user buys "diaper", then "beer" is recommended.**

Session-based Recommendation

association rule을 기반으로 하는 추천 모델

- Do not require **the existence of user-profiles** or **their entire historical preferences.**
- Provide recommendations solely based on a **user's interactions** in an ongoing session.



Simple Association Rules (AR)

- A simplified version of the association rule mining technique with a maximum rule size of two.

- Count the **frequency of two co-occurring items i and j.**

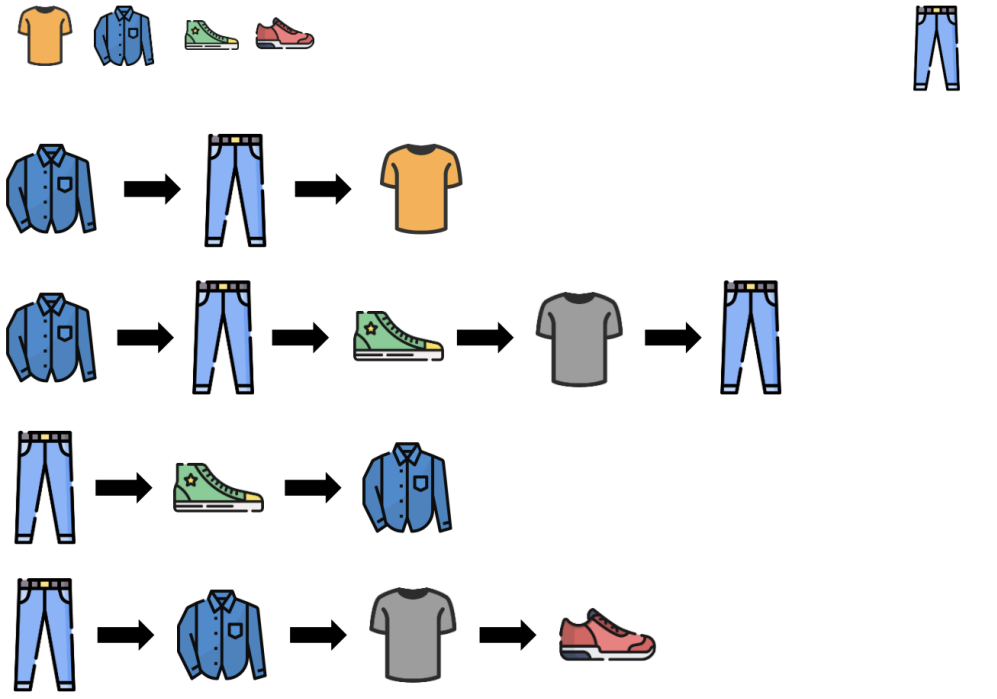
$$score_{AR}(i,j) = \frac{1}{\sum_{s\in\mathcal{S}}\sum_{x=1}^{|s|}\mathbf{1}_{EQ}(i,s_x)\cdot(|s|-1)}\sum_{s\in\mathcal{S}}\sum_{x=1}^{|s|}\sum_{y=1}^{|s|}\mathbf{1}_{EQ}(i,s_x)\cdot\mathbf{1}_{EQ}(j,s_y)$$

Normalization

Counting scheme: how often two items i and j co-occur.

$\mathcal{S}$: a set of all sessions
s: a session, $s = (s_1, \dots, s_{|s|})$
$\mathbf{1}_{EQ}(i,j)$: it is 1 if i and j are same and 0 otherwise.

- **The sequential order does not matter.**
- **Which items are recommended after viewing jeans?**



Markov Chains (MC)

- Consider the **trainsition probability** between two subsequent items in a session.
- Count how oftern **users viewed item j immediately after viewing item i.**

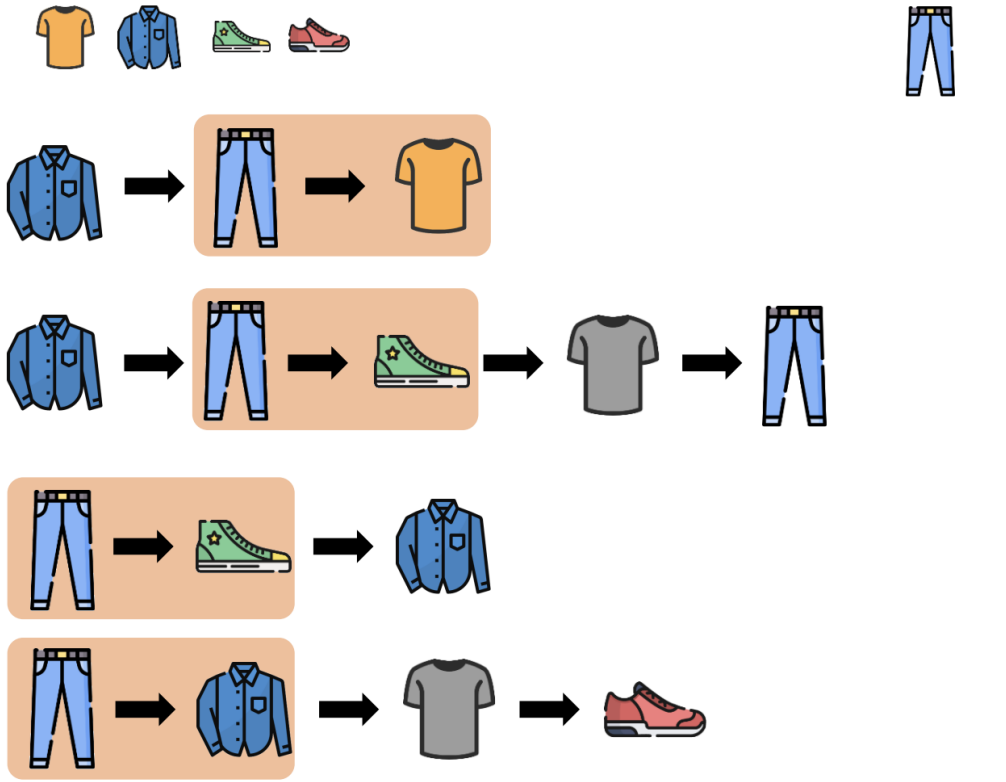
$$score_{MC}(i,j) = \frac{1}{\sum_{s\in\mathcal{S}}\sum_{x=1}^{|s|-1} \mathbf{1}_{EQ}(i,s_x)} \sum_{s\in\mathcal{S}} \sum_{x=1}^{|s|-1} \mathbf{1}_{EQ}(i,s_x)\cdot\mathbf{1}_{EQ}(j,s_{x+1})$$

Normalization

Counting scheme: how often two consecutive items i and j co-occur.

$\mathcal{S}$: a set of all sessions
$\mathbf{1}_{EQ}(i,j)$: it is 1 if i and j are same and 0 otherwise.

➢ **Which item is recommended after viewing jeans?**



Sequential Rules (SR)

- A variation of MC or AR
- Consider a rule **when an item j after an item i in a session even when other items appear between i and j.**

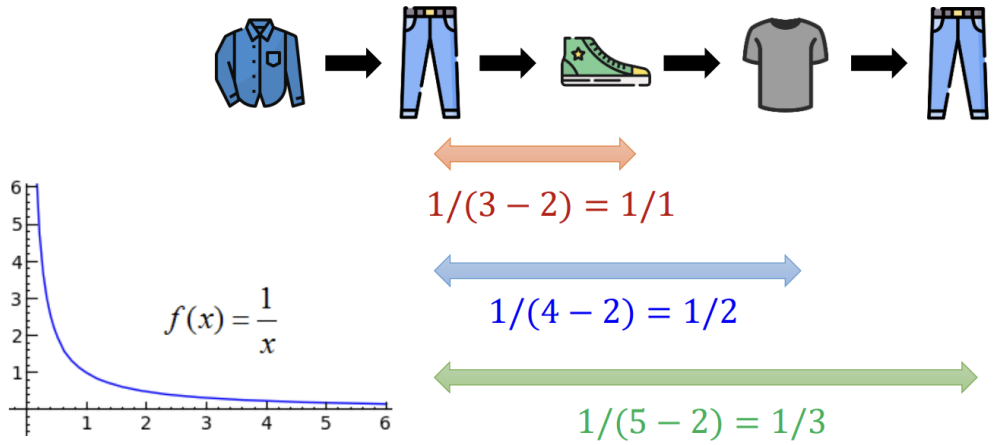
$$score_{SR}(i,j) = \frac{1}{\sum_{s\in\mathcal{S}}\sum_{x=2}^{|s|}\mathbf{1}_{EQ}(i,s_x)\cdot x}\sum_{s\in\mathcal{S}}\sum_{x=2}^{|s|}\sum_{y=1}^{x-1}\mathbf{1}_{EQ}(i,s_y)\cdot\mathbf{1}_{EQ}(j,s_x)\cdot w_{SR}(x-y)$$

Normalization **Counting scheme** **Weighting scheme**

The weighting scheme considers the distance between two sequential items.

$w_{SR}(x) = 1/x$, where x corresponds to the number of steps between the two items

➢ **Weights for items are decayed depending on positions.**



3. Probabilistic Models

Introduction : Probabilistic Models

- We predict a user rating using probability theory.
  - Bayes classification model can be used.

- For the classification problem, it is the task of **assigning an item to one of several pre-defined categories.**
- Given a target user, we address the problem of **calculating the most probable rating value.**

$$P(r_{ui} = ? \mid User's\ ratings)$$

Bayesian Classification

➢ Statistical classifier

- Performs probabilistic prediction, i.e., predicts **class membership probabilities**.
- It is based on **Bayes' Theorem**.

Likelihood: the function of Y given fixed X

Prior: prior knowledge of the output Y before observing any data

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Posterior: the probability of output Y given X

Evidence

➢ The naïve Bayes classifier is commonly used.

Recap: Bayes' Theorem

- Let $\mathcal{D}$ be a training set of samples and their associated classes.
- Each sample is represented by an d-dimensional vector $\mathbf{x} = (x_1, x_2, \dots, x_d)$
  - Suppose that there are m classes $c_1, c_2, \dots, c_m$.
- Find the maximum posterior $P(c_i \mid \mathbf{x})$.

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i)P(c_i)$$

  - Since $P(\mathbf{x})$ is constant for all classes, only the numerator is considered.

Challenge of Bayesian Classification

- Predicts x belongs to c_i if the probability $P(c_i|x)$ is the highest among all the $P(c_i|x)$ for all the k classes.
  - **Practical difficulty** : Require initial knowledge of many probabilities and significantly high computational costs.
- Simplified assumption
  - **Attributes are conditionally independent.**

$$P(c_i \mid \mathbf{x}) \propto P(\mathbf{x} \mid c_i)P(c_i) = \prod_{j=1}^{d} P(x_j \mid c_i)\, P(c_i)$$

  - This greatly reduces the computation cost.

Naïve Bayes Classifiers

- Let's estimate as follows.

$$f(\mathbf{x}) = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} P(c_i \mid x_1, x_2, \cdots, x_d)$$

- It is equivalent to:

$$f(\mathbf{x}) = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} \frac{P(x_1, x_2, \cdots, x_d \mid c_i) P(c_i)}{P(x_1, x_2, \cdots, x_d)} = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} P(x_1, x_2, \cdots, x_d \mid c_i) P(c_i)$$

Adding the **conditional independence assumption**

$$= \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} \prod_{j=1}^{d} P(x_j \mid c_i)\, P(c_i)$$

Probabilistic Models for CF

- How to predict the probability of rating value 5 for Item 5 with Bob's other ratings?

$$P(Item5 = 5 \mid Bob's\ rating)$$
$$= P(Item5 = 5 \mid Item1 = 4, Item2 = 3, Item3 = 4, Item4 = 5)$$




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

Probabilistic Models for CF

- **Under the assumption that the items are conditionally independent, the naïve Bayes classifier can be used.**

$$P(Item5 = 5 \mid Bob's\ rating)$$
$$= P(Item5 = 5 \mid Item1 = 4, Item2 = 3, Item3 = 4, Item4 = 5)$$
$$\propto P(Item1 = 4, Item2 = 3, Item3 = 4, Item4 = 5 \mid Item5 = 5)P(Item5 = 5)$$

- **How to compute the likelihood?**

$$P(I1 = 4 \mid I5 = 5)P(I2 = 3 \mid I5 = 5)P(I3 = 4 \mid I5 = 5)P(I4 = 5 \mid I5 = 5)$$
$$= 1/3 \times 2/3 \times 1/3 \times 3/3$$

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

33

Probabilistic Models for CF

- **We also compute the probability for other ratings and determine the rating with the highest probability.**

$$P(Item5 = 4 \mid Bob's\ rating)$$
$$= P(Item5 = 4 \mid Item1 = 4, Item2 = 3, Item3 = 4, Item4 = 5)$$
$$\propto P(Item1 = 4, Item2 = 3, Item3 = 4, Item4 = 5 \mid Item5 = 4)P(Item5 = 4)$$

- **How to compute the likelihood?**

$$P(I1 = 4 \mid I5 = 4)P(I2 = 3 \mid I5 = 4)P(I3 = 4 \mid I5 = 4)P(I4 = 5 \mid I5 = 4)$$
$$= 0/1 \times 0/1 \times 1/1 \times 1/1$$

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

34

Discussion

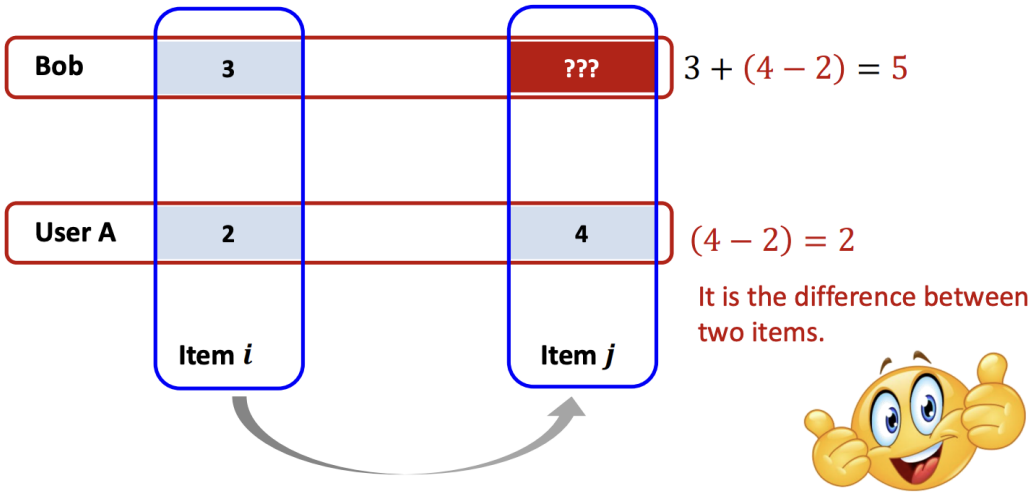
- It can be easily extended to **binary rating data.**
- It does not work well with **small or sparse rating matrices.**
- The probabilistic model is used with the clustering method.
  - Group similar users or items into clusters.
  - Predict the probability of a user u filling into a certain cluster.
- In some domains, the probabilistic model slightly outperforms the other algorithms.
  - In the popular movie domain, it is worse than user-based CF.

---

4. Slope One Predictors

What are Slope One Predictors?

- It is the **simplest form** of item-based CF.
  - It is a form of regression with a single parameter, i.e., $f(x) = x + b$.
    - Pairwise comparison of how items are liked by different users
  - It is computed by the average difference between two items' ratings.



Example: Slope One Predictors

- For a pair of (Item1, Item3), there are two co-ratings.
  - The average difference between the two items is 0.5.
- So, Bob's prediction for Item3 is $2 + 0.5 = 2.5$.

	Item1	Item2	Item3
Bob	2	5	???
User1	3	2	5
User2	4		3

The average distance between the two items is $(2 + (-1))/2 = 0.5$.

$(5 - 3) = +2$

$(3 - 4) = -1$

Example: Slope One Predictors

- For a pair of (Item2, Item3), there is one co-rating.
  - The difference between the two items is 3.
- So, Bob's prediction for Item3 is $5 + 3 = 8$.

	Item1	Item2	Item3
Bob	2	5	???
User1	3	2	5
User2	4		3

The average distance between the two items is $+3$.

$(5 - 2) = +3$

39

Example: Slope One Predictors

- **Bob's prediction for Item3 is finally computed by using the weighted average for the number of co-rated items.**

$$Pred(Bob, Item3) = \frac{2 \times 2.5 + 1 \times 8}{2 + 1} = 4.33$$

Bob's prediction is $2 + 0.5 = 2.5$.

	Item1	Item2	Item3
Bob	2	5	???
User1	3	2	5
User2	4		3

Bob's prediction is $5 + 3 = 8$.

	Item1	Item2	Item3
Bob	2	5	???
User1	3	2	5
User2	4		3

Slope One Predictors

- **The average deviation of two items is calculated.**

$$dev_{i,j} = \sum_{(r_{ui}, r_{uj}) \in \mathcal{S}_{i,j}(\mathbf{R})} \frac{r_{uj} - r_{ui}}{|\mathcal{S}_{i,j}(\mathbf{R})|}$$

  - Let $\mathcal{S}_{i,j}(\mathbf{R})$ denote the set of evaluations that contain both ratings for two items i and j.

- **The prediction for user u and item j using $dev_{i,j}$ is**

$$\hat{r}_{uj} = dev_{i,j} + r_{ui}$$

Slope One Predictors

- **A simple combination is computed by the average of the predictions for all co-rated items.**

$$\hat{r}_{uj} = \frac{\sum_{i \in Relevant(u,j)} (dev_{i,j} + r_{ui})}{|Relevant(u,j)|}$$

$Relevant(u,j)$: a set of items that have **at least one co-rating** with item j by user u

- **It is extended by weighting the predictions based on the number of co-ratings.**

$$\hat{r}_{uj} = \frac{\sum_{i \in \mathcal{S}(u)-\{j\}} (dev_{i,j} + r_{ui}) \cdot |\mathcal{S}_{i,j}(\mathbf{R})|}{\sum_{i \in \mathcal{S}(u)-\{j\}} |\mathcal{S}_{i,j}(\mathbf{R})|}$$

$\mathcal{S}(u)$: a set of items that the user u have rated

Discussion

- It is **easy to implement** as an average engineer.
- It supports o**n-the-fly data update.**
- It is **efficient at query time.**
- It provides v**alid recommendations** for cold-start users.
- It shows a **reasonable recommendation quality** despite its simplicity.