电子科技大学
University of Electronic Science and Technology of China

Rec Sys Using Co-clustering Algorithm

Qinyuan Liu

13/Jan/2016

Data Mining Lab, Big Data Research Center, UESTC
Email：junmshao@uestc.edu.cn
http://staff.uestc.edu.cn/shaojunming

- **Outline:**
  - 1.Introduction
  - 2.Co-clustering Algorithm
  - 3.A rec process based on co-clustering
  - 4.Conclusion

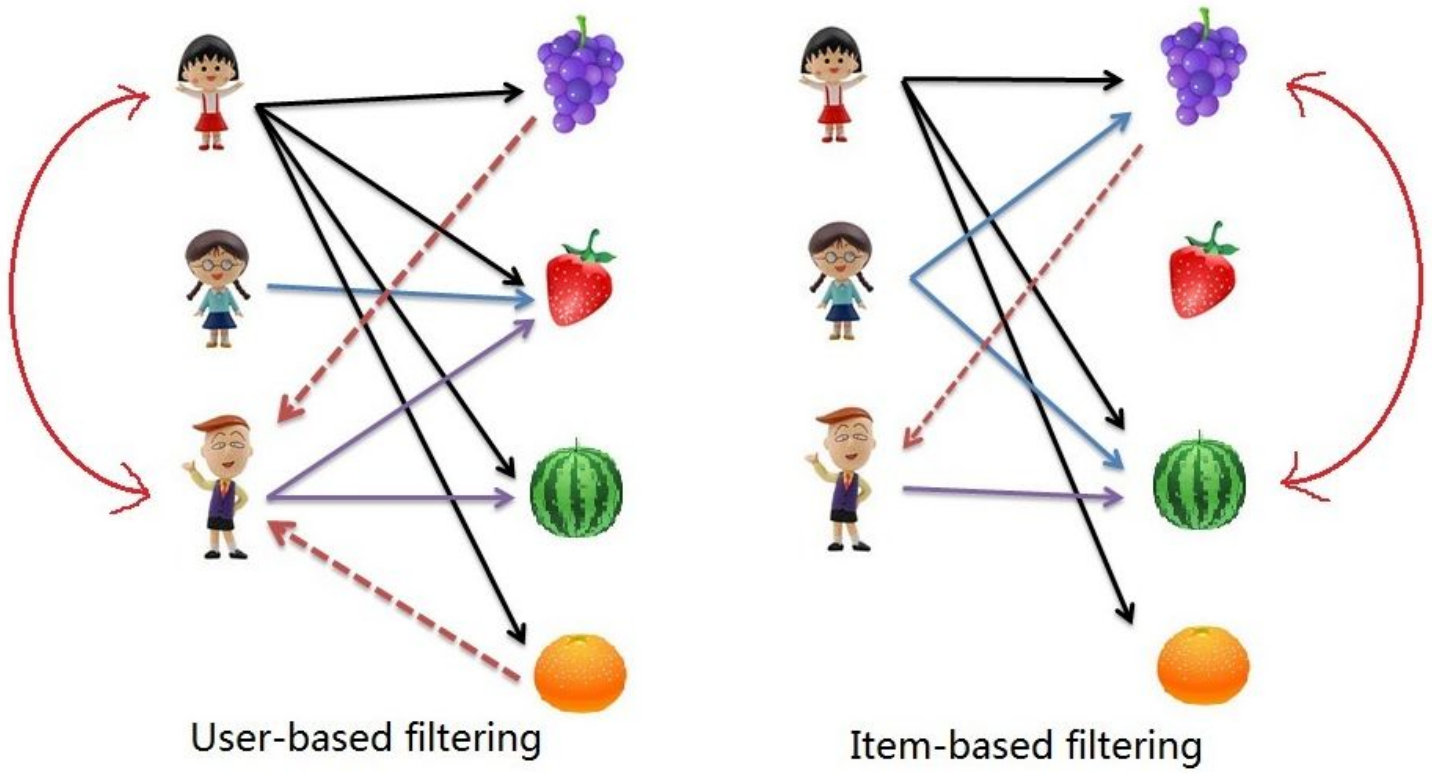
1.Introduction

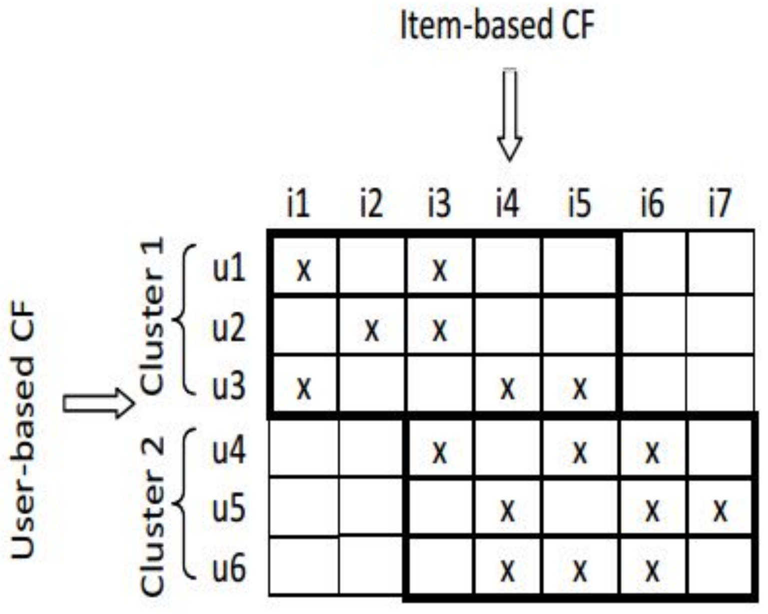
- Why？
  - - Why we need a Rec sys?
- How?
  - - Collaborative filtering
- What？
  - - What’s the shortage of CF based on clustering?

1.1 Why we need a recommender system?

数据挖掘实验室
Data Mining Lab

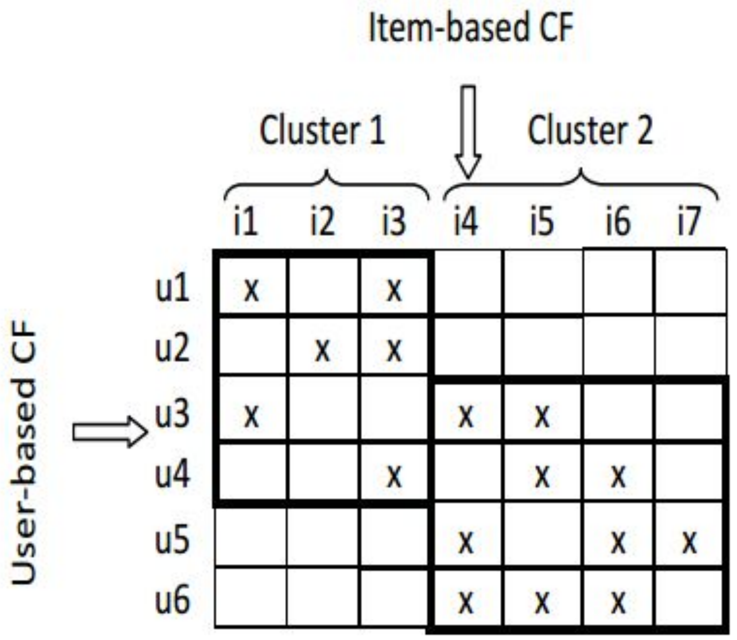
1.2 How to recommend an item to a user?





	i1	i2	i3	i4	i5	i6	i7
u1	x		x				
u2		x	x				
u3	x			x	x		
u4			x		x	x	
u5				x		x	x
u6				x	x	x	

(a) User clustering



	i1	i2	i3	i4	i5	i6	i7
u1	x		x				
u2		x	x				
u3	x			x	x		
u4			x		x	x	
u5				x		x	x
u6				x	x	x	

(b) Item clustering

Please , please give me a recommendation quickly !
I have a large number of user-item data , what should i do ?
... So am I...

- Collaborative Filtering Algorithm Based on the Item Clustering
- Collaborative Filtering Algorithm Based on the User Clustering

1.3 What’s the shortage of CF based on clustering?

➢ Assumption:

- **Case 1:**You and your friend are both xueba, you and he have the some taste in choosing classes， but he likes political lesson and you don’t. Here ,based on user similarity , what will happen if we recommend the political class to you?
- **Case 2:**Suppose that a man comes from the Southeast Asia and he likes pineapple, coco, and other tropical fruits except durian. Here, based on fruits similarity, what will happen if we recommend the durian to him?

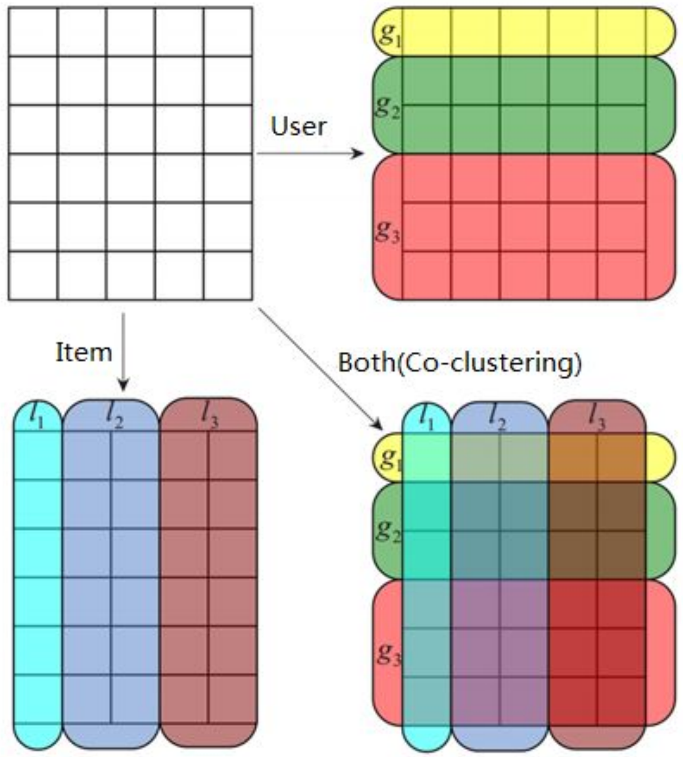
- **Keyword:** we just clustering the data using one-dimension information and drop the information in the other dimension. In order to solve this problem we…

2.Co-clustering

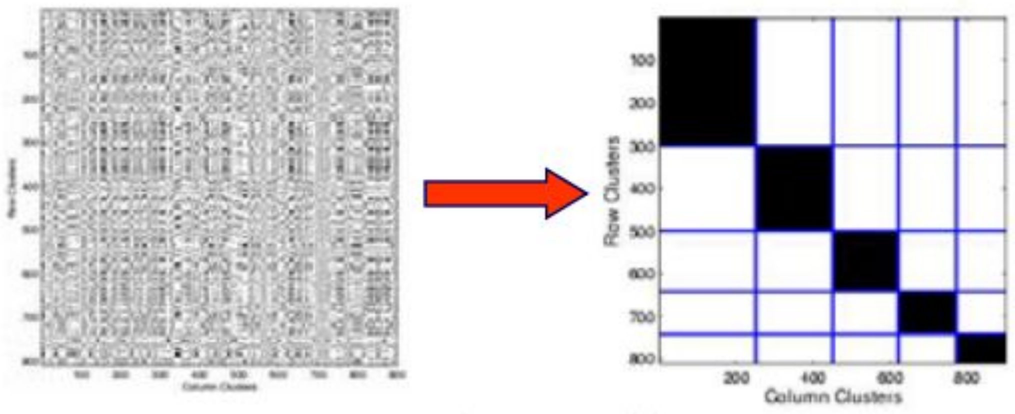
- - 2.1 Introduction
- - 2.2 Information theory concept
- - 2.3 Co-clustering algorithm
- - 2.4 Properties
- - 2.5 Application

2.1 Introduction

- Co-clustering：



- **Co-clustering Definition:** Finding a pair of maps from rows to row-clusters and from columns to column-clusters



- Co-clustering Methods:

➢ **Hierarchical co-clustering:** Biological and medical sciences

➢ **Spectral co-clustering:** solved as an instance of graph partitioning (k-cut) and can be relegated to an eigenvector computation problem

- **Information-theoretic co-clustering:** Co-clustering by finding a pair of maps from rows to row clusters and from columns to column-clusters, with minimum mutual information loss
- **Optimization-based co-clustering:** include information-theoretic-based objective functions ,or other residue functions

2.2 Information Theory Concept

- Entropy of a random variable X with probability distribution p:

$$H(p) = -\sum_x p(x) \log p(x)$$

Measure of the average uncertainty

- The Kullback-Leibler(KL) Divergence:

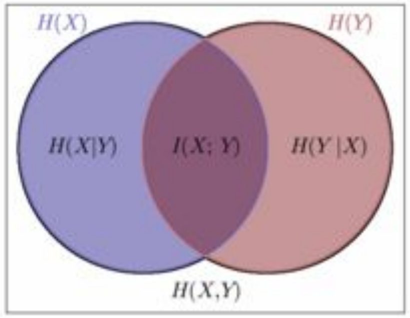
$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Measure of how different two probability distributions are.

➢ Mutual Information between random variables X and Y :

- $\mathrm{I(X, Y)} = H(X) - H(X|Y)$
- $\quad = H(X) + H(Y) - H(X,Y)$
- $\quad = \sum_x p(x) log \frac{1}{p(x)} + \sum_y p(y) log \frac{1}{p(y)} - \sum_{x,y} p(x,y) log \frac{1}{p(x,y)}$
- $\quad = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$

The amount of information X contains about Y



- **Co-Clustering's goal:**

➢ Finding maps C_X and C_Y

- $C_X = \{x_1, x_2, \dots, xm\} \rightarrow \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k\}$
- $C_Y = \{y_1, y_2, \dots, y_n\} \rightarrow \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_l\}$

➢ loss in mutual information is minimized

$$I(X;Y) - I(\hat{X};\hat{Y})$$

Example: $\begin{pmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{pmatrix} \rightarrow \begin{pmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{pmatrix}$ Loss: 0.0957

- Finding optimal Co-Clustering

- **Lemma:** The loss in mutual information can be expressed as the distance of p(X, Y) to an approximation q(X, Y)

$$I(X;Y) - I(\hat{X};\hat{Y}) = D(p(X,Y) \,||\, q(X,Y))$$

q is approximation of p:

$$q(x,y) = p(\hat{x},\hat{y})p(x|\hat{x})p(y|\hat{y}), \text{ where } x \in \hat{x}, y \in \hat{y}.$$

$$D(p(X,Y)||q(X,Y)) = I(X;Y) - I(\hat{X};\hat{Y})$$

$$= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x\in\hat{x}}\sum_{y\in\hat{y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} - \sum_{\hat{x}}\sum_{\hat{y}}(\sum_{x\in\hat{x}}\sum_{y\in\hat{y}} p(x,y))\log\frac{p(\hat{x},\hat{y})}{p(\hat{x})p(\hat{y})}$$

$$= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x\in\hat{x}}\sum_{y\in\hat{y}} p(x,y)\log\frac{p(x,y)}{p(\hat{x},\hat{y})\frac{p(x)}{p(\hat{x})}\frac{p(y)}{p(\hat{y})}}$$

$$= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x\in\hat{x}}\sum_{y\in\hat{y}} p(x,y)\log\frac{p(x,y)}{q(x,y)}$$

Example: Calculating q(x; y): approximation of p(x; y)

$$p(x,y) = \begin{array}{cc} & n \\ m & \begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix} \end{array}$$

$$\underset{p(x\,|\,\hat{x})}{\begin{array}{cc} & k \\ m & \begin{bmatrix} .5 & 0 & 0 \\ .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \\ 0 & 0 & .5 \end{bmatrix} \end{array}} \underset{p(\hat{x},\hat{y})}{\begin{array}{cc} & l \\ k & \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix} \end{array}} \underset{p(y\,|\,\hat{y})}{\begin{array}{cc} & n \\ l & \begin{bmatrix} .36 & .36 & .28 & 0 & 0 & 0 \\ 0 & 0 & 0 & .28 & .36 & .36 \end{bmatrix} \end{array}} = \underset{q(x,y)}{\left[\begin{array}{ccc|ccc} .054 & .054 & .042 & 0 & 0 & 0 \\ .054 & .054 & .042 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & .042 & .054 & .054 \\ 0 & 0 & 0 & .042 & .054 & .054 \\ \hline .036 & .036 & 028 & .028 & .036 & .036 \\ .036 & .036 & .028 & .028 & .036 & .036 \end{array}\right]}$$

- Objective function for loss in mutual information
- The loss in mutual information can be expressed as

➢ a weighted sum of relative entropies between row distribution and row cluster distribution

$$D(p(X,Y) \parallel q(X,Y)) = \sum_{\hat{x}} \sum_{x:\in\hat{x}} p(x) D(p(Y|x) \parallel q(Y|\hat{x}))$$

➢ a weighted sum of relative entropies between column distribution and column cluster distribution

$$D(p(X,Y) \parallel q(X,Y)) = \sum_{\hat{y}} \sum_{y\in\hat{y}} p(y) D(p(X|y) \parallel q(X|\hat{y}))$$

$$D(p(X,Y)||q(X,Y)) = D(p(X,Y,\hat{X},\hat{Y})||q(X,Y,\hat{X},\hat{Y})).$$

$$\begin{aligned}
&D(p(X,Y,\hat{X},\hat{Y})||q(X,Y,\hat{X},\hat{Y})) \\
&= \sum_{\hat{x},\hat{y}} \sum_{x:C_X(x)=\hat{x},y:C_Y(y)=\hat{y}} p(x,y,\hat{x},\hat{y}) \log \frac{p(x,y,\hat{x},\hat{y})}{q(x,y,\hat{x},\hat{y})} \\
&\overset{(a)}{=} \sum_{\hat{x},\hat{y}} \sum_{x:C_X(x)=\hat{x},y:C_Y(y)=\hat{y}} p(x)p(y|x) \log \frac{p(x)p(y|x)}{p(x)q(y|\hat{x})} \\
&= \sum_{\hat{x}} \sum_{x:C_X(x)=\hat{x}} p(x) \sum_{y} p(y|x) \log \frac{p(y|x)}{q(y|\hat{x})},
\end{aligned}$$

$$q(x) = \sum_y q(x,y) = \sum_{\hat{y}} \sum_{y\in\hat{y}} p(\hat{x},\hat{y})p(x|\hat{x})p(y|\hat{y}) = p(x,\hat{x}) = p(x)$$

2.3 Co-Clustering Algorithm

- Input:

  $p(X; Y)$ - the joint probability distribution

  k - the desired number of row clusters

  l - the desired number of column clusters

- Output:

  The partition function C_X and C_Y

- Step 1:

  Initialization:

  Set t = 0. Start with some initial partition functions $C_X^{(0)}$ and $C_Y^{(0)}$.Compute:

$$q^{(0)}(\hat{X},\hat{Y}),\ q^{(0)}(X|\hat{X}),\ q^{(0)}(Y|\hat{Y}),$$

  and the distribution $q^{(0)}(Y|\hat{x})$,$1 \leq \hat{x} \leq k$

- Step 2:Compute row clusters: For each row x, find its new cluster index as

$$\sum_X^{(t+1)}(x) = argmin\hat{x}\ \mathrm{D(p(Y|x)||}q^{(t)}(Y|\hat{x}))$$

数据挖掘实验室
Data Mining Lab

- Step 3. Computer distributions
$$q^{(t+1)}(\hat{X},\hat{Y}),\ q^{(t+1)}(X|\hat{X}),\ q^{(t+1)}(Y|\hat{Y}),$$
and the distribution $q^{(\mathrm{t}+1)}(X|\hat{y}), 1 \le \hat{y} \le l$
- Step 4. Compute column clusters: For each column y, find its new cluster index as
$$\sum\nolimits_{\mathrm{Y}}^{(t+2)}(\mathrm{y}) = argmin\hat{y}\ \mathrm{D}(\mathrm{p}(\mathrm{X}|\mathrm{y})||q^{(t+1)}(\mathrm{X}|\hat{\mathrm{y}}))$$
- Step 5:Compute distributions
$$q^{(t+2)}(\hat{X},\hat{Y}),\ q^{(t+2)}(X|\hat{X}),\ q^{(t+2)}(Y|\hat{Y}),$$
and the distribution $q^{(\mathrm{t}+2)}(\mathrm{Y}|\hat{\mathrm{x}}), 1 \le \hat{\mathrm{x}} \le \mathrm{k}$

- Step 6: Stop and return

$$C_X = C_X^{(t+2)} \text{ and } C_Y = C_Y^{(t+2)}$$

if the change in objective function value, that is, $D(p(X, Y)||q^{(t)}(X, Y)) - D(p(X, Y)||q^{(t+2)}(X, Y))$, is “small” (say 10^{-3});

Else set $t = t + 2$ and go to step 2.

2.4 Properties

- Co-clustering monotonically decreases loss in mutual information
- Co-clustering converges to a local minimum
- Can be generalized to multi-dimensional
- Implicit dimensionality reduction at each step helps overcome sparsity & high-dimensionality
- Computationally efficient: O(nt(k + l))

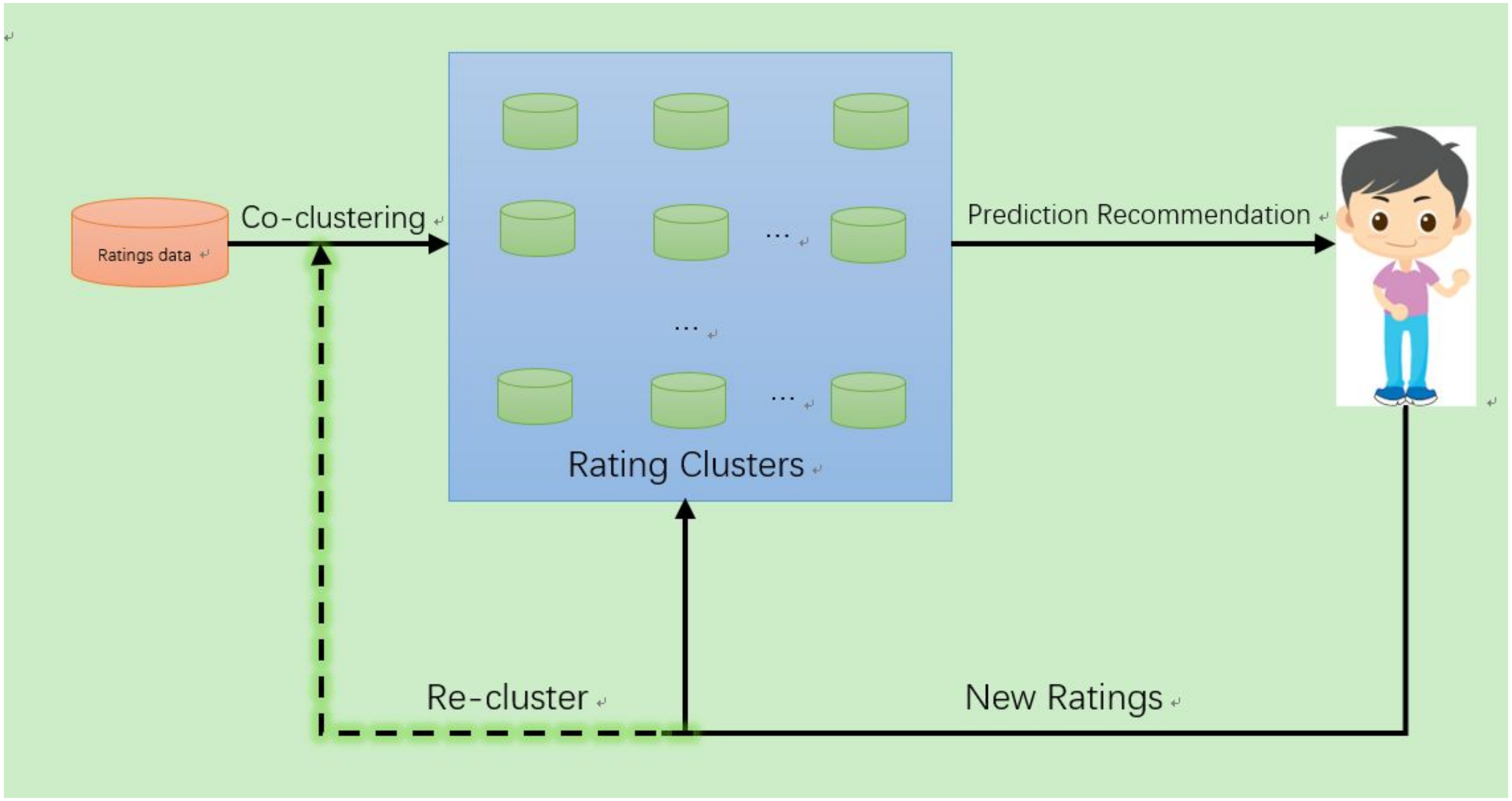
2.5 Application

- In biological and medical sciences: gene expression
- NLP: text analysis
- Image processing: data compression
- Rec sys： rating prediction 、 Top-N、 Location Recommendation
- …

3. A rec process based on co-clustering

- 3.1 Framework of this system
- 3.2 Prediction
- 3.3 Incremental
- 3.4 Parallel

• 3.1 Framework of this system



- 3.2 Prediction
  - 3.2.1 COCLUST
  - 3.2.2 MF

3.2.1 COCLUST

- The key idea is to simultaneously obtain user and item neighborhoods via co-clustering and generate predictions based on the average ratings of the co-clusters (user-item neighborhoods) while taking into account the individual biases of the users and items.

- **Prediction function:**

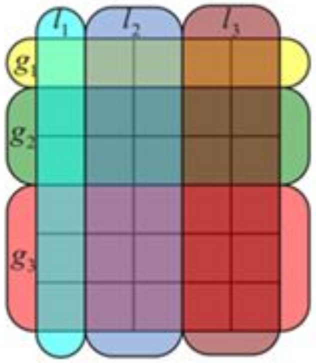
$$\widehat{A_{ij}} = A_{gh}^{COC} + \left(A_i^R - A_g^{RC}\right) + \left(A_j^C - A_h^{CC}\right)$$

$$A_{gh}^{COC} = \frac{\sum_{i'|p(i')=g}\sum_{j'|r(j')=h} A_{i'j'}}{\sum_{i'|p(i')=g}\sum_{j'|r(j')=h} W_{i'j'}}$$

$$A_g^{RC} = \frac{\sum_{i'|p(i')=g}\sum_{j'=1}^{n} A_{i'j'}}{\sum_{i'|p(i')=g}\sum_{j'=1}^{n} W_{i'j'}}$$

$$A_h^{CC} = \frac{\sum_{i'=1}^{m}\sum_{j'|r(j')=h} A_{i'j'}}{\sum_{i'=1}^{m}\sum_{j'|r(j')=h} W_{i'j'}}$$

$$A_i^R = \frac{\sum_{j'=1}^{n} A_{ij'}}{\sum_{j'=1}^{n} W_{ij'}} \qquad A_j^C = \frac{\sum_{i'=1}^{m} A_{i'j}}{\sum_{i'=1}^{m} W_{i'j}}$$



- Algorithm： Prediction
- Input： $A^R, A^{RC}, A^C, A^{CC}, A^{COC}, A^{glob}$, co-clustering($\rho$, γ) user set U, item set P, user u_i, item p_j
- Output: Rating r
- Method:

➢ Case $u_i \in U$ and $p_j \in P$ **{old user-old item}**

$$g \leftarrow \rho(i), h \leftarrow \gamma(j)$$

$$r \leftarrow A^{COC}_{gh} + \left(A^R_i - A^{RC}_g\right) + \left(A^C_j - A^{CC}_h\right)$$

➢ Case $u_i \in U$ and $p_j \notin P$ **{old user-new item}**

$$g \leftarrow \rho\,(i)$$

$$r \leftarrow A^R_i$$

- Case $u_i \notin U$ and $p_j \in P$ **{new user-old item}**

  $h \leftarrow \gamma(j)$

  $r \leftarrow A_j^C$

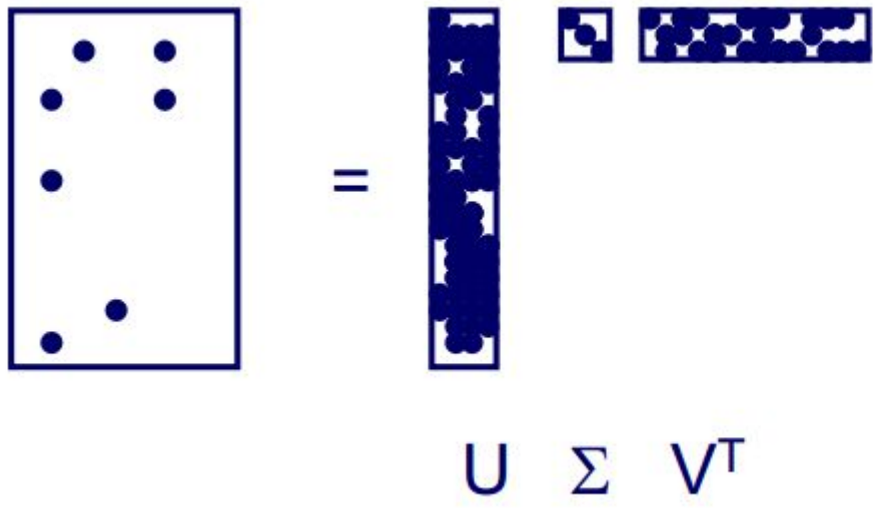
- Case $u_i \notin U$ and $p_j \notin P$ **{new user-new item}**

  $r \leftarrow A^{glob}$

3.2.2 MF

- Matrix factorization (MF) is one of the most often applied techniques for CF problems. The idea behind MF techniques is very simple. Suppose we want to approximate the matrix R as the product of two matrices: $R \approx PQ$, where P is an $N \times K$ and Q is a $K \times M$ matrix. This factorization gives a low dimensional numerical representation of both users and items.
  - SVD
  - NMF

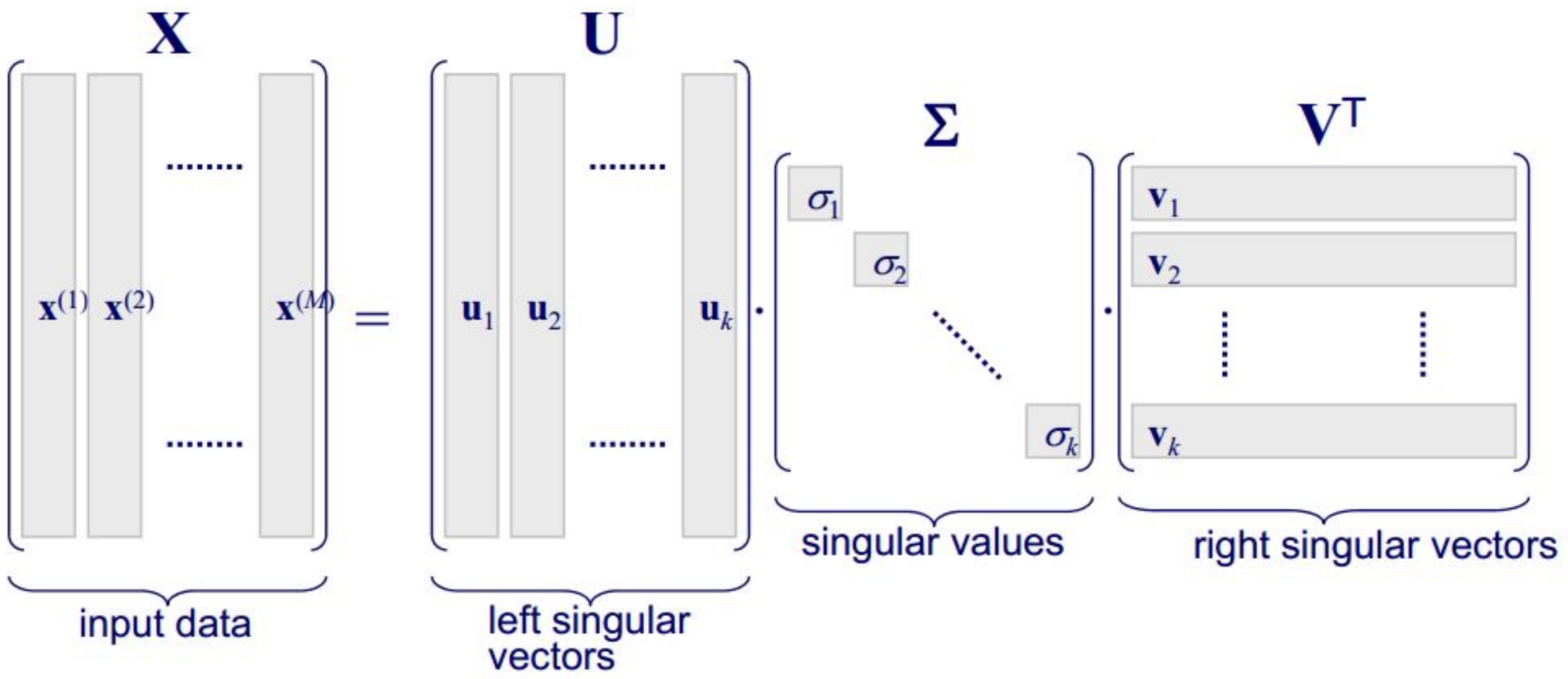
- SVD:



SVD
Destroys sparsity

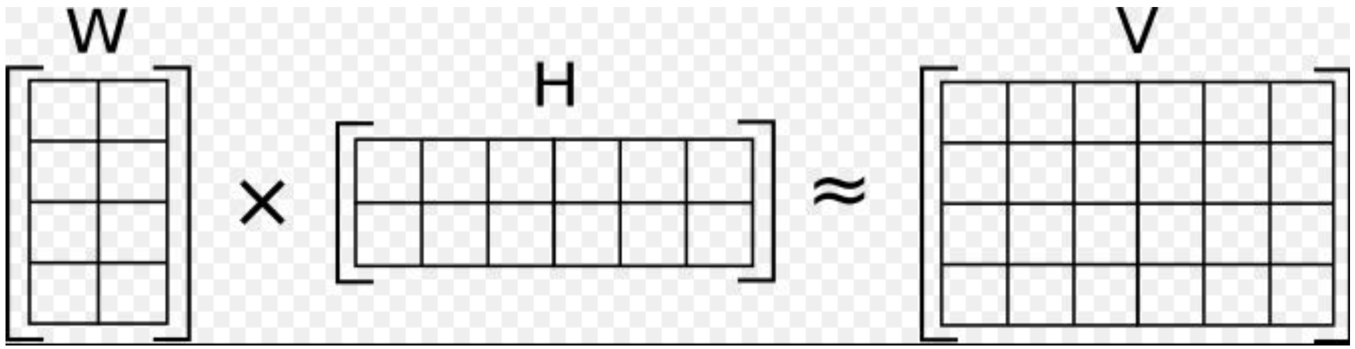
- SVD:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$$



- The left-singular vectors of X are eigenvectors of XX^T.
- The right-singular vectors of M are eigenvectors of X^TX.
- The non-zero singular values of X (found on the diagonal entries of Σ) are the square roots of the non-zero eigenvalues of both X^TX and XX^T.

- Non-negative Matrix Factorization (NMF) is one type of the methods that focus on the analysis of non-negative data matrices which are often originated from text, images and biology. We can regard W as basis matrix and H is coefficient matrix.



- Given an n×m matrix X composed of non-negative elements, the task is to factorize X into a non-negative matrix W of size n × r and another non-negative matrix H of size r × m such that X ≈ W H. It is usually written down as an optimization:

$$min_{W\geq 0,H\geq 0}||X - WH||^2 = min_{W\geq 0,H\geq 0}\sum_{ij}(X - WH)_{ij}^2$$

- In general, the derived algorithm of NMF is as follows:

➢ Randomize W and H with positive numbers in [0, 1]. Select the cost function to be minimized.

➢ Fixing W, update H, then update W for the updated H and so on until the process converges.

The advantages of the SVD are:

- (1) an optimality property;
- (2) initialization does not affect SVD algorithms.

The advantages of NMF are:

- (1) sparsity and non negativity;
- (2) reduction in storage;
- (3) interpretability.

- 3.3 Incremental
- The key idea here is that since the prediction of the ratings depends only on the summary statistics (matrix averages), updating these statistics using the new ratings will incorporate most of the information in the new ratings. When the new rating corresponds to an existing user and item, it is straightforward to update the corresponding averages.

- **Algorithm: Incremental Training**

➢ Input: Matrix averages A^R, A^{RC}, A^C, A^{CC}, A^{COC}, A^{glob} ,Non-zero counts W^R, W^{RC}, W^C, W^{CC}, W^{COC}, W^{glob}, co-clustering(ρ, γ) user set U, item set P, user u_i, item p_j

➢ Output: Updated matrix averages and non-zero counts

➢ Method:

Step 1.Identify user-item clusters

Case $u_i \in U$ and $p_j \in P$ **{old user-old item}**

$g \leftarrow \rho(i), h \leftarrow \gamma(j)$

Case $u_i \in U$ and $p_j \notin P$ **{old user-new item}**

$g \leftarrow \rho(i), h \leftarrow 0, \gamma(j) \leftarrow 0, P \leftarrow P \cup \{p_j\}$

Case $u_i \notin U$ and $p_j \in P$ **{new user-old item}**

$g \leftarrow 0, h \leftarrow \gamma(j), \rho(i) \leftarrow 0, U \leftarrow U \cup \{u_i\}$

Case $u_i \notin U$ and $p_j \notin P$ **{new user-new item}**

$g \leftarrow 0, h \leftarrow , \rho(i) \leftarrow 0, \gamma(j) \leftarrow 0, U \leftarrow U \cup \{u_i\}, P \leftarrow P \cup \{p_j\}$

Step 2.Update the matrix averages

Step 3.Increment non-zero counts

- 3.4 Parallel
  - Parallel Co-clustering
  - Parallel Prediction
  - Parallel Incremental Training

Algorithm: Parallel Co-clustering

- Input: Ratings Matrix A, Non-zero matrix W, #row clusters l, #col. Clusters k
- Output: Locally optimal co-clustering(ρ , γ) and averages A^{COC}, A^{RC}, A^{CC}, A^{R}, and A^{C}
- Method:

Processor p gets sub-matrices $A^{rp}(m_p \times n)$ & $A^{cp}(m \times n_p)$

数据挖掘实验室
Data Mining Lab

– 1.Randomly initialize (ρ^{p}, γ^{p})

Repeat

- 2a.Compute partial contributions to matrix averages and non-zero counts based on A^{rp} and A^{cp}
- 2b.Accumulate the partial contributions to obtain overall matrix averages.
- 2c.Update row assignments ρ^{p} for all rows in A^{rp}
- 2d.Update column assignments γ^{p} for all columns in A^{cp}

Until convergence

– 3.Concatenate the cluster maps(ρ^{p}, γ^{p}) to obtain (ρ , γ)

4. Conclusion

- Solve problem : scalability and deployment for dynamic scenarios
- Advantage: provide high quality predictions at a much lower computational cost compared to traditional correlation and SVD-based approaches.
- Notice: The weighted matrix 、 Soft Cluster method

Thanks

Qinyuan Liu
liuqinyuan1992@163.com