Week 6: Collaborative Based RSs – Part II

CS3448: Recommender Systems /

CSX4207/ITX4207: Decision Support and Recommender Systems

Asst. Prof. Dr. Rachsuda Setthawong

Objectives

- To introduce model-based collaborative based filtering algorithms

Outlines

- Model-based collaborative based filtering algorithms

Recall the Main Idea of Collaborative Approach

- To exploit information about the past behavior/opinions of an existing user community.
- To predict which items the current user will most probably like.

Asst. Prof. Dr. Rachsuda Setthawong

How to Generate Recommendation Using Collaborative Based Filtering Approach

Target
User's Preferences

Recommendation Algorithm(s)

Prediction
(numerical value)

Rank

Recommendations
(A list of top-k suggested items)

Community's Preferences

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

Memory-based VS Model-based Approaches

Memory-based approach

- Modeless (no model created)
- Directly applying rating matrix to find neighbors and to recommend items.
- Time consuming and not scalable
- Example, user-based NN

Model-based approach

- Offline process the raw data.
- At run time, require only the “learned” model to make prediction.
- Update/retrained the model periodically.
- Example, matrix factorization methods, association rule mining, etc.

Model-based Algorithms

- Association rule mining
- Probabilistic recommendation approaches
- Slope one predictor
- Matrix factorization methods

Asst. Prof. Dr. Rachsuda Setthawong

Association Rule Mining

- **Basic idea:** identify **rule-like relationship patterns** (X -> Y) in large-scale sales transactions.

Example,

- ***If*** a customer **buys** baby food ***then*** he/she **also buys** diapers in 70 percent of the cases.
  {baby food} -> {diapers}

- In RSs, "**If** user *X* **liked** both *item1* and *item4*, **then** *X* will **most probably also like** *item3*."

  {Item1, Item4} -> {Item3}

Transaction ID	**Baby Food**	Bread	Beer	**Diaper**	Coke
1	**Y**		Y	**Y**	
2		Y			Y
3	**Y**		Y	**Y**	
4	**Y**	Y	Y	**Y**	
5		Y			
6					Y
7	**Y**		Y	**Y**	Y
8	**Y**	Y	Y	**Y**	
9	**Y**	Y	Y	**Y**	
10	**Y**	Y	Y	**Y**	

Asst. Prof. Dr. Rachsuda Setthawong

Generating Association Rules

- Step 1: Frequent Itemset Generation
    - Generate all **itemsets whose support ≥ min_support_threshold**
- Step 2: Rule Generation
    - Generate **high confidence rules** from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Asst. Prof. Dr. Rachsuda Setthawong

Step 1: Frequent Itemset Generation

User-Item rating matrix

	Item1	Item2	Item3	Item4
User1	5	3	3	
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

Terminologies:

$Itemset_{length\_k}$ = a set of *k*-items

e.g., $\text{Itemset}_{\text{length\_2}}$ = { {Item1, Item2}, {Item1, Item3}, {Item1, Item4 }, {Item2, Item3}, {Item2, Item4}, {Item3, Item4} }

min_support_threshold: a user-defined value in the range [0 – 1].

$FrequentItemset_{length\_k}$ = a set of *k*-items whose **support** is greater than or equal to *min_support_threshold*

$$support(x) = \frac{count(x)}{n},$$

where *count(x)*: no. of rows that *x* present in a dataset

n: no. of rows in a dataset

x: an itemset in a dataset

Asst. Prof. Dr. Rachsuda Setthawong

Step 1: Frequent Itemset Generation – *Cont.*

User-Item rating matrix

	Item1	Item2	Item3	Item4
User1	5	3	3	
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

Suppose that *min_support_threshold* = 4/6 = 0.67
$Itemset_{length\_2}$ = { {Item1, Item2}, {Item1, Item3}, {Item1, Item4}, {Item2, Item3}, {Item2, Item4}, {Item3, Item4} }
$FrequentItemset_{length\_2}$ = { {Item1, Item2}, {Item1, Item4}, {Item2, Item4} }
support({Item1, Item2}) = 5/6 = 0.83
support *({Item1, Item3})* = 3/6 = 0.5
support({Item1, Item4}) = 4/6 = 0.67
...
support({Item2, Item4}) = 4/6 = 0.67

Terminologies:

$Itemset_{length\_k}$ = a set of *k*-items

min_support_threshold: a user-defined value in the range [0 – 1].

$FrequentItemset_{length\_k}$ = a set of *k*-items whose support is greater than or equal to *min_support_threshold*

$$support(x) = \frac{count(x)}{n},$$

where *count(x)*: no. of rows that *x* presents in a dataset

n: no. of rows in a dataset

x: an itemset in a dataset

Asst. Prof. Dr. Rachsuda Setthawong

Step 2: Rule Generation

- Generate frequent rules *(high confidence rules)* from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

User-Item rating matrix

	Item1	Item2	Item3	Item4
User1	5	3	3	
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

Terminologies:
a rule, $x \rightarrow y$ = a **co-occurrence** of two itemsets, x and y

e.g., the *candidate rules* created from the frequent itemset;

Given
$\text{Frequent\_Itemset}_{\text{length\_2}}$ = {{Item1, Item2}, {Item1, Item4}, {Item2, Item4}}

∴ Candidate Rules = { Item1 -> Item2,
Item2 -> Item1,
Item1 -> Item4,
Item4 -> Item1,
Item2 -> Item4,
Item4 -> Item2 }

Asst. Prof. Dr. Rachsuda Setthawong

Step 2: Rule Generation – *Cont.*

- Generate frequent rules *(high confidence rules)* from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

User-Item rating matrix

	Item1	Item2	Item3	Item4
User1	5	3	3	
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

Terminologies:

a rule, $x \rightarrow y$ = a co-occurrence of two itemsets, x and y

$$confident\ of\ the\ rule, conf(x \rightarrow y) = \frac{support(x \cap y)}{support(x)} = \frac{count(x \cap y)}{count(x)}$$

min_confident_threshold: a user-defined value in the range [0 – 1].

Given

Frequent_Itemset$_{length2}$ = {{Item1, Item2}, {Item1, Item4}},

and suppose that Min_confident_threshold = 0.8

∴ Frequent Rules = {

Item1 -> Item2 (conf = 5/5 =1),

Item2 -> Item1 (conf = 5/5 = 1),

Item1 -> Item4 (conf = 4/5 = 0.8),

Item4 -> Item1 (conf = 4/5 = 0.8),

Item2 -> Item4 (conf = 4/5 = 0.8),

Item4 -> Item2 (conf = 4/5 = 0.8)

}

Asst. Prof. Dr. Rachsuda Setthawong

Step 2: Rule Generation – *Cont.*

- Another example,

Alter the dataset

	Item1	Item2	Item3	Item4
User1	5	3	3	4
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

Given
Frequent_Itemset$_{length_3}$ = { {Item1, Item2, Item4} },

and suppose that min_confident_threshold = 0.9

Candidate Rules = {
Item1 -> Item2, Item4 (conf = 5/5 =1.0),
Item2 -> Item1, Item4 (conf = 5/5 = 1.0),
Item4 -> Item1, Item2 (conf = 5/6 = **0.8**), *<< not frequent!!*
Item1, Item2 -> Item4 (conf = 5/5 = 1.0),
Item1, Item4 -> Item2 (conf = 5/5 = 1.0),
Item2, Item4 -> Item1 (conf = 5/5 = 1.0)
}

Frequent Rules = {
Item1 -> Item2, Item4 (conf = 5/5 =1.0),
Item2 -> Item1, Item4 (conf = 5/5 = 1.0),
Item1, Item2 -> Item4 (conf = 5/5 = 1.0),
Item1, Item4 -> Item2 (conf = 5/5 = 1.0),
Item2, Item4 -> Item1 (conf = 5/5 = 1.0)
}

Asst. Prof. Dr. Rachsuda Setthawong

Applying Association Rules in RSs – 1/4

- Calculation of the set of interesting association rules with high confidence and support is performed offline.
  1. Determine the set of $X \rightarrow Y$ association rules that are relevant for the target user.
  2. Compute the union of items appearing in the consequent Y of these association rules not previously experienced by the target user.
  3. Sort the products according to the confidence of the rule.
  4. Return the first N elements of this ordered list as a recommendation.

Asst. Prof. Dr. Rachsuda Setthawong

Applying Association Rules in RSs – 2/4

- Calculation of the set of interesting association rules with high confidence and support is performed offline.

1. Determine **the set of $X \rightarrow Y$ association rules** that are relevant for the target user (Item1).

   Frequent Rules = {r1: **Item1** -> Item2, r2: Item2 -> **Item1**, r3: **Item1** -> Item4, r4: Item4 -> **Item1**, r5: Item2 -> Item4, r6: Item4 -> Item2}

	Item1	Item2	Item3	Item4
User7	4			

2. Compute the union of items appearing in the consequent *Y* of these association rules not previously experienced by the target user.
3. Sort the products according to the confidence of the rule.
4. Return the first *N* elements of this ordered list as a recommendation.

Asst. Prof. Dr. Rachsuda Setthawong

Applying Association Rules in RSs – 3/4

- Calculation of the set of interesting association rules with high confidence and support is performed offline.

1. Determine the set of $X \rightarrow Y$ association rules that are relevant for the target user.

   Frequent Rules = {r1: Item1 -> **Item2**, r2: Item2 -> **Item1**, r3: Item1 -> **Item4**, r4: Item4 -> **Item1**, r5: Item2 -> Item4, r6: Item4 -> Item2}

2. Compute the **union (∪)** of items appearing in the consequent *Y* of these association rules NOT previously experienced by the target user.

	Item1	Item2	Item3	Item4
User7	4			

{Item1 -> **Item2**} ∪ {Item1 -> **Item4**}={Item2, Item4}

	Item1	Item2	Item3	Item4
User1	5	3	3	
User2	4	5	4	5
User3	5	5		4
User4	4	3	3	5
User5	5	4		4
User6				4

3. Sort the products according to the confidence of the rule.
4. Return the first *N* elements of this ordered list as a recommendation.

Asst. Prof. Dr. Rachsuda Setthawong

Applying Association Rules in RSs – 4/4

- Calculation of the set of interesting association rules with high confidence and support is performed offline.

1. Determine the set of $X \rightarrow Y$ association rules that are relevant for the target user.
2. Compute the union of items appearing in the consequent Y of these association rules not previously experienced by the target user.
3. **Sort** the products according to the confidence of the rule.

   Rank 1: Item2 (because Conf(r1: Item1 -> **Item2**)= **1**)

   Rank 2: Item4 (because Conf(r3: Item1 -> **Item4**)= **0.8**)
4. Return the first N elements of this ordered list as a recommendation.
   1. Recommended items = {**Item2**, **Item4**}

Probabilistic Recommendation Approaches

- **Basic idea:** Assigning an object to one of several predefined categories (classification problem)
- Example, Naïve Bayes classifiers

Asst. Prof. Dr. Rachsuda Setthawong

Naïve Bayes Classifiers

1. With conditionally independent attributes, calculate conditional probabilities using Bayes Theorem for each possible rating value (Y) given the target user's other ratings.

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

2. Select the rating value with the highest probability as a prediction.

Asst. Prof. Dr. Rachsuda Setthawong

Example, given Alice's rating = {1, 3, 3, 2}

Predict Alice's rating for item 5

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	1	3	3	2	?

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	2	4	2	2	4
User 2	**1**	**3**	**3**	**5**	**1**
User 3	4	5	2	3	3
User 4	**1**	**1**	**5**	**2**	**1**

- P(Y)
  - P(Item5=1) = 2/4
  - P(Item5=2) = 0
  - P(Item5=3) = ¼
  - P(Item5=4) = ¼
  - P(Item5=5) = 0
- P(X|Item5=1) =

  P(Item1=**1**|Item5=1) *

  P(Item2=**3**|Item5=1) *

  P(Item3=**3**|Item5=1) *

  P(Item4=**2**|Item5=1)

  = 2/2 * 1/2 * 1/2 * 1/2

  = 0.125
- P(X|Item5=2) =

  P(Item1=**1**|Item5=2) *

  P(Item2=**3**|Item5=2) *

  P(Item3=**3**|Item5=2) *

  P(Item4=**2**|Item5=2)

  = 0/0 * … * … * …

  = 0

  …

• • •

- P(X|Item5=5) =

  …

Asst. Prof. Dr. Rachsuda Setthawong

Example – Cont.

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

Constant

predicted rating of item 5 for Alice = 1

- **P(Item5=1|X) = P(X|Item5=1) * P(Item5=1) = 0.125 * 2/4 = 0.0625**
- P(Item5=2|X) = P(X|Item5=2) * P(Item5=2) = 0 * 0 = 0
- P(Item5=3|X) = P(X|Item5=3) * P(Item5=3) = 0 * 1/4 = 0
- P(Item5=4|X) = P(X|Item5=4) * P(Item5=4) = 0 * 1/4 = 0
- P(Item5=5|X) = P(X|Item5=5) * P(Item5=5) = 0 * 0 = 0
- *Note: The data X exclude Alice's ratings*

Asst. Prof. Dr. Rachsuda Setthawong

Slope One Predictor – 1/3

- Basic Idea: calculate "popularity differential" between items for users.

	Item1	Item5
Alice	2	?

	Item1	Item5
User1	1	2

popularity differential = 2-1 = 1

- Example 1,
  - Predict rating of item 5 for Alice *using Slope One Prediction*:

    = Alice's rating + popular differential

    = 2 + (2-1) = 3

- In reality,
  - There are several items rated by the *target user* (e.g., Alice)
  - There are several users rated the *target item* (e.g., item 5)
- Therefore,
  - The complete calculation of Slope One Prediction is shown in the next slide.

Asst. Prof. Dr. Rachsuda Setthawong

Slope One Predictor - 2/3

- The average deviation *dev* between the target item *j*, and other items *i* of all **users co-rated both items**,

$$dev_{j,i} = \sum_{(u_j,u_i)\in S_{j,i}(R)} \frac{u_j - u_i}{|S_{j,i}(R)|}$$

Note: *j* is the target item that we want to predict rating for the target user *u*.

u_j: rating that a user *u* gives on item *j*

$S_{j,i}(R)$: no. of users rated both items *i*, *j*

- Predicted rating on the target item *j* for the **target user *u***,

$$pred(u,j) = \frac{\sum_{i\in Relevant(u,j)}(\boxed{dev_{j,i}} + u_i)}{|Relevant(u,j)|}$$

Note: u and u_i in the 2nd and 3rd formula refer to different users.

- Weighted predicted rating on the target item *j* for the **target user *u*** (weighting using no. of users co-rated both items),

$$pred(u,j) = \frac{\sum_{i\in S(u)-\{j\}}(\boxed{dev_{j,i}} + u_i) \times |S_{j,i}(R)|}{\sum_{i\in S(u)-\{j\}}|S_{j,i}(R)|}$$

Asst. Prof. Dr. Rachsuda Setthawong

Slope One Predictor – 3/3

	Item1	Item2	Item3
Alice	2	5	?

	Item1	Item2	Item3
User1	3	2	5
User2	4		3

- $dev_{Item3,\ Item1} = \Sigma(rating_{Item3} - rating_{Item1})/Num\_users\_rated\_both\_items$

  $= ((5-3) + (3-4)) / 2 = 0.5$
- $dev_{Item3,\ Item2} = \Sigma(rating_{Item3} - rating_{Item2})/Num\_users\_rated\_both\_items$

  $= (5-2) / 1 = 3$
- pred(Alice, Item3)

  $= [\ \Sigma(dev_{3,\ i} + rating_{Alice,Item\_i}) * Num\_users\_rated\_both\_items\ ] / \Sigma(Num\_users\_rated\_both\_items)$

  $= [\ \underbrace{((0.5+2)*2)}_{item1} + \underbrace{((3+5)*1)}_{item2}\ ]/ (2+1) = 4.33$

$$pred(u,j) = \frac{\sum_{i \in S(u)-\{j\}} (dev_{j,i} + u_i) \times |S_{j,i}(R)|}{\sum_{i \in S(u)-\{j\}} |S_{j,i}(R)|}$$

Matrix Factorization Methods

- **Basic idea:** derive a set of latent (hidden) factors from the rating patterns and characterize both user and items by such vectors of factors.
- A recommendation for an item *i* is made when the target user and the item *i* are similar wrt these factors.
- Example,
  - Singular Value Decomposition (SVD)
  - Alternating Least Square (ALS)

Additional Resource: ALS http://bugra.github.io/work/notes/2014-04-19/alternating-least-squares-method-for-collaborative-filtering/
Additional Resources: SVD https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Singular Value Decomposition (SVD)

- **SVD Theorem:**
- A given matrix A can be decomposed into a product of three matrices using linear algebra

$$A = U\Sigma V^T$$

- where,
  - U = left singular vector
  - V = right singular vector
  - Σ = the singular values
- **Key idea:** retain only the most important features (with largest singular values) by taking only the first few columns of U and V^T.

Additional Resources: SVD Tutorial
https://datajobs.com/data-science-repo/SVD-Tutorial-[Kirk-Baker].pdf
http://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm

Asst. Prof. Dr. Rachsuda Setthawong

SVD Example - 1/2

	User1	User2	User3	User4
Item1	3	4	3	1
Item2	1	3	2	6
Item3	2	4	1	5
Item4	3	3	5	2

U2 (Item)	
-0.4312	0.4931
-0.5327	-0.5305
-0.5237	-0.4052
-0.5058	0.5578

V2 (User)	
-0.3593	0.3676
-0.5675	0.0879
-0.4428	0.5686
-0.5938	-0.7305

Σ2	
12.2215	0
0	4.9282

*: The first two columns of U, V and Σ are used here.

Asst. Prof. Dr. Rachsuda Setthawong

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

SVD Example - 2/2

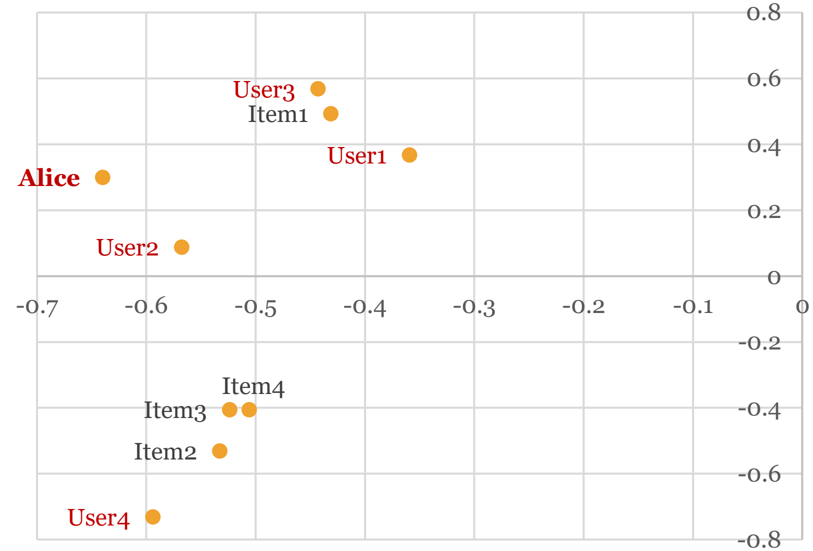
- Given Alice's rating vector [5, 3, 4, 4],
  - Calculate Alice's vector in 2D space:

    $\text{Alice}_{2D} = \text{Alice} \times U2 \times \Sigma_2^{-1}$

    $= [-0.64, 0.30]$

  - To suggest items, find the most similar users using the using the 2D matrix (V2), *e.g., cosine sim,* and suggest unseen, high-rating items

A plot of all users (items)' vector in 2D space:



	U2 (Item)	
Item1	-0.4312	0.4931
Item2	-0.5327	-0.5305
Item3	-0.5237	-0.4052
Item4	-0.5058	0.5578

	V2 (User)	
User1	-0.3593	0.3676
User2	-0.5675	0.0879
User3	-0.4428	0.5686
User4	-0.5938	-0.7305

Matrix Inverse: http://mathworld.wolfram.com/MatrixInverse.html

Demonstration of Full SVD Calculation – 1/11

- Given
  - A rectangular matrix A
- Generate
  - An orthogonal matrix U
  - A diagonal matrix S
  - Transpose of an orthogonal matrix V^T

S.T.

- $A_{mn} = U_{mm} S_{mn} V^T_{nn}$

  where

  $U^T U = I$; the columns of U are orthonormal eignenvector of AA^T,

  $V^T V = I$; the columns of V are orthonormal eignenvector of $A^T A$, and

  S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 2/11

Example:

$$A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

Step 1) Find U by start with

1.1. **Find AA^T**

$$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 3/11

1.2. **Find Eigenvalues** of AA^T

- By Solving equation $A\vec{v} = \lambda\vec{v}$

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

rewrite this as the set of equations

$11x_1 + x_2 = \lambda x_1 \rightarrow (11 - \lambda)x_1 + x_2 = 0$ eq. 1

$x_1 + 11x_2 = \lambda x_2 \rightarrow x_1 + (11 - \lambda)x_2 = 0$ eq. 2

Solve for λ by setting the determinant of the coefficient matrix to zero,

$$\begin{vmatrix} (11 - \lambda) & 1 \\ 1 & (11 - \lambda) \end{vmatrix} = 0$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Demonstration of Full SVD Calculation – 4/11

Solve for λ by setting the determinant of the coefficient matrix to zero,

$$\begin{vmatrix} (11-\lambda) & 1 \\ 1 & (11-\lambda) \end{vmatrix} = 0$$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Equivalent to

$$|A| = ad - bc$$

$$(11-\lambda)(11-\lambda) - 1 \cdot 1 = 0$$

$$(\lambda - 10)(\lambda - 12) = 0$$

$$\lambda = 10,$$

$$\lambda = 12$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 5/11

1.3. **Find corresponding eigenvectors** of the eigenvalues in 1.2. by substituting values in Eq. 1

For $\lambda = 10$,

$$(11 - 10)x_1 + x_2 = 0 \quad \rightarrow \quad x_1 = -x_2$$

From several valid values, let's select $x_1 = 1$ *and* $x_2 = -1$

For $\lambda = 12$,

$$(11 - 12)x_1 + x_2 = 0 \quad \rightarrow \quad x_1 = x_2$$

From several valid values, let's select $x_1 = 1$ *and* $x_2 = 1$

Construct a matrix from the eigenvectors sorted by eigenvalues in descending order: $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 6/11

- 1.4. Apply Gram-Schmidt orthonormalization process to the column vectors to **convert Eigenvectors (matrix) into an orthogonal matrix**.

$$\begin{array}{c} \overrightarrow{v_1} \\ \overrightarrow{v_2} \end{array} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\overrightarrow{u_1} = \frac{\overrightarrow{v_1}}{|\overrightarrow{v_1}|} = \frac{[1,1]}{\sqrt{1^2+1^2}} = \frac{[1,1]}{\sqrt{2}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

Compute $\overrightarrow{w_2} = \overrightarrow{v_2} - \overrightarrow{u_1} \cdot \overrightarrow{v_2} * \overrightarrow{u_1} = [1,-1] - \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right] \cdot [1,-1] * \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right] = [1, -1]$

and normalize $\overrightarrow{u_2} = \frac{\overrightarrow{w_2}}{|\overrightarrow{w_2}|} = \left[\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right]$

to give $U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 7/11

Step 2) Find V by starting with

2.1. Find A^TA

$$A^TA = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 8/11

2.2. Find Eigenvalues of A^TA

- By Solving equation $A\vec{v} = \lambda\vec{v}$ (Same as in 1.2.)
  - $\lambda = 0$,
  - $\lambda = 10$,
  - $\lambda = 12$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 9/11

2.3. **Find corresponding eigenvectors** of the eigenvalues in 2.2.

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & 2 \\ 1 & 0 & 5 \end{bmatrix}$$

2.4. Apply Gram-Schmidt orthonormalization process to the column vectors to **convert Eigenvectors (matrix) into an orthogonal matrix**.

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-5}{\sqrt{30}} \end{bmatrix},$$

and calculate

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Demonstration of Full SVD Calculation – 10/11

Step 3) Find S by taking the square roots of the non-zero eigenvalues and populate the diagonal with them, putting the largest in s_{11}, the next largest in s_{22} and so on.

- $\lambda = 10$,
- $\lambda = 12$

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Demonstration of Full SVD Calculation – 11/11

- $A_{mn} = U_{mm} S_{mn} V_{nn}^{T}$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

Ref: https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Root Mean Square Error (RMSE)

- **Root mean square error** takes the difference for each observed and predicted value.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

where

- x_i : actual rating
- $\hat{x}_i$: predicted rating
- N : number of predicted items in a test set.

Asst. Prof. Dr. Rachsuda Setthawong

Example

UserID	Item ID	Actual Rating	Predicted Rating
1	1	5	4.5
1	2	3	2.5
2	3	4	4
2	4	4	3.5
3	1	5	4.5
3	3	4	4

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

$$= \sqrt{\frac{(5-4.5)^2+(3-2.5)^2+(4-4)^2+(4-3.5)^2+(5-4.5)^2+(4-4)^2}{6}}$$

=0.41

Further reading

- SVD:
  - https://alyssaq.github.io/2015/singular-value-decomposition-visualisation/
  - https://alyssaq.github.io/2015/20150426-simple-movie-recommender-using-svd/
- ALS:
  - https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1
- SLIM:
  - http://glaros.dtc.umn.edu/gkhome/node/774