INT 303 BIG DATA ANALYTICS

Lecture14: Summary & Review

Pengfei FAN
pengfei.fan@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University
西交利物浦大學

GRADES

Sequence	Method	Learning outcomes assessed	Duration	Timing	% of final mark	Resit
#1	Project 1: Data Scraping	All	See notice	S2	15%	NO
#2	Project 2: Big Data Competition	All	See notice	S2	15%	NO
#3	Written Exam	All	See notice	S2	70%	NO

WHAT WE HAVE LEARNED

Module 1:

<u>The basics.</u>

- What is big data analytics?
- What is data and big data?
- Big data Gramma

WHAT WE HAVE LEARNED

Module 2:

Big data collection and visualization.

- Data collection and Data scraping
- Data Virtualization

WHAT WE HAVE LEARNED

Module 3:

Systems and software . It introduces the popular platforms available for big data processing.

- Large-scale computing
- Distributed file system
- MapReduce: Distributed computing programming model
- Spark: Extends MapReduce

WHAT WE HAVE LEARNED

Module 4:

<u>The data processing methods and algorithms.</u>

- Representing Data and Engineering Features
- Dimensionality Reduction
- Big Data Analysis Models
- Bagging
- Boosting

WHAT WE HAVE LEARNED

Module 5:

<u>The big data application.</u>

- Recommendation System

Questions:

1. Big Data Conception

Data cleaning

Feature engineering

Data preprocessing

Infrastructure for large-scale computing

…

Questions:

2. Dimensionality Reduction

PCA & SVD

sklearn.decomposition.PCA

class `sklearn.decomposition.`**`PCA`**(*n_components=None*, ***, *copy=True*, *whiten=False*, *svd_solver='auto'*, *tol=0.0*, *iterated_power='auto'*, *n_oversamples=10*, *power_iteration_normalizer='auto'*, *random_state=None*) ¶

[source]

Principal component analysis (PCA).

Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD.

It uses the LAPACK implementation of the full SVD or a randomized truncated SVD by the method of Halko et al. 2009, depending on the shape of the input data and the number of components to extract.

It can also use the scipy.sparse.linalg ARPACK implementation of the truncated SVD.

Notice that this class does not support sparse input. See **`TruncatedSVD`** for an alternative with sparse data.

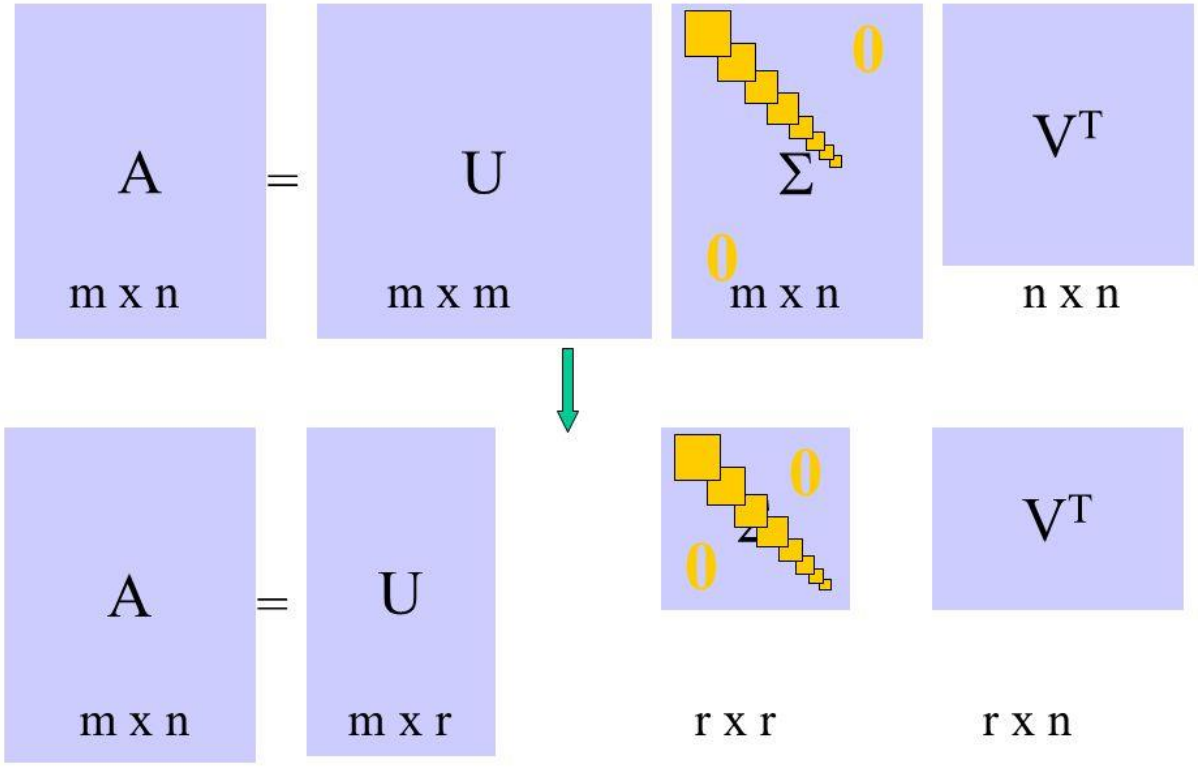
Read more in the User Guide.

Questions:

2. Dimensionality Reduction
PCA & SVD

Singular Value Decomposition
Low Rank Approximations
https://intoli.com/blog/pca-and-svd/
https://web.stanford.edu/class/cs168/l/l9.pdf

The Singular Value Decomposition



r = the rank of A
= number of linearly independent columns/rows

Simple Matrix Algebra

consider

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}.$$

Then

$$\mathbf{AB} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}.$$

Consider the 4×5 matrix

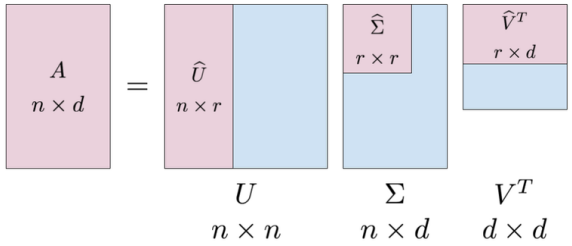
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

A singular value decomposition of this matrix is given by $\mathbf{U\Sigma V^*}$

$$\mathbf{U} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{\Sigma} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{5} & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0} & 0 \end{bmatrix}$$

$$\mathbf{V^*} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ -\sqrt{0.2} & 0 & 0 & 0 & -\sqrt{0.8} \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix}$$



Questions:

(1) Singular Values
(2) Low-rank Approximation
(3) Reconstruction Error*

*The reconstruction error between two matrices is defined as the Frobenius norm of their difference.

$$\|A_F\| = \sqrt{\Sigma(A_{ij}^2)}$$

Questions:

3. Data Grammar

Data preprocessing
DataFrame
Code analysis
Python operations
…

Questions:

4. Decision Tree

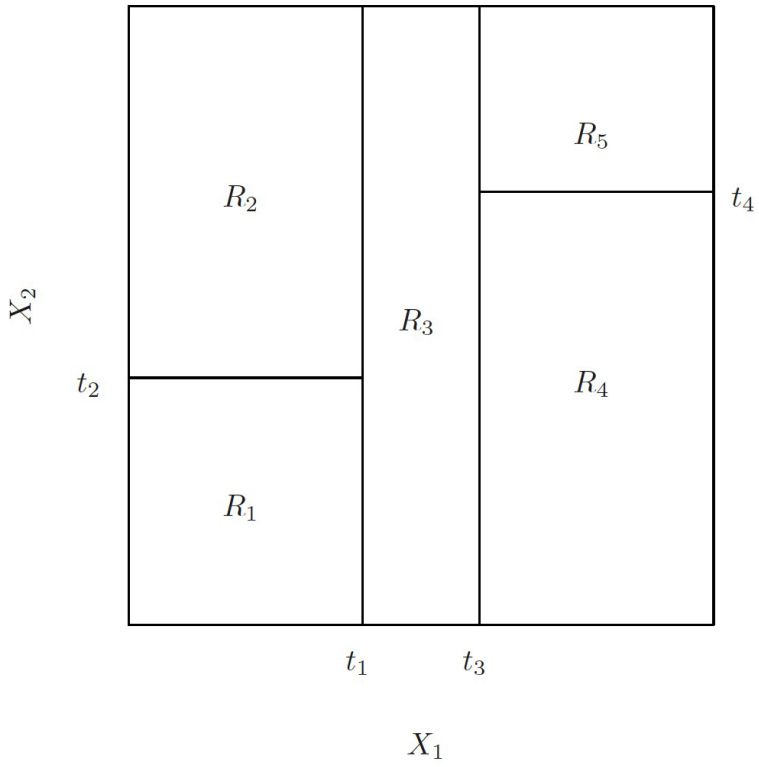
Decision Trees
Bagging
Random Forests
…

DECISION TREES

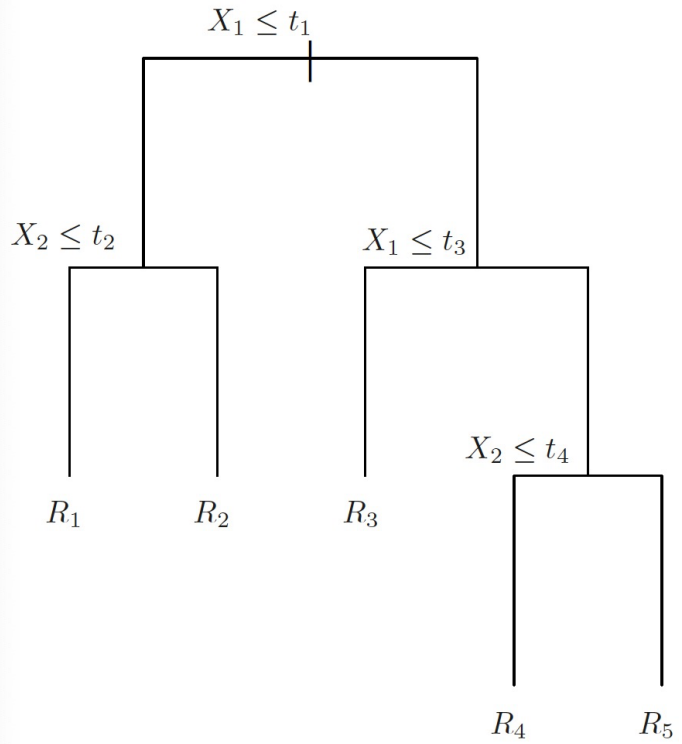
- Motivation
- Decision Trees
- Classification Trees
- Regression Trees
- Text Reading: Section 8.1, in the textbook.

QUESTION:

Sketch the tree corresponding to the partition of the predictor space



Create partitions of the predictor space using the tree



Questions:

5. MapReduce

Distributed computing programming model

…

Questions:

6. Boosting

Gradient Boosting
Adaboost
…

SUMMARY OF TREE ENSEMBLE METHODS

In **bagging**, the trees are grown independently on random samples of the observations. Consequently, the trees tend to be quite similar to each other. Thus, bagging can get caught in local optima and can fail to thoroughly explore the model space.

In **random forests**, the trees are once again grown independently on random samples of the observations. However, each split on each tree is performed using a random subset of the features, thereby decorrelating the trees, and leading to a more thorough exploration of model space relative to bagging.

In **boosting**, we only use the original data, and do not draw any random samples. The trees are grown successively, using a "slow" learning approach: each new tree is fit to the signal that is left over from the earlier trees, and shrunken down before it is used.

Boosting

High-level idea: combine a lot of classifiers

- Sequentially construct / identify these classifiers, $h_t(\cdot)$, one at a time
- Use *weak* classifiers to arrive at a complex decision boundary (*strong* classifier), where β_t is the contribution of each weak classifier

$$h[\boldsymbol{x}] = \text{sign}\left[\sum_{t=1}^{T} \beta_t h_t(\boldsymbol{x})\right]$$

Our plan

- Describe AdaBoost algorithm
- Derive the algorithm

P16-P22 in the slides:
https://www.cs.cmu.edu/~atalwalk/teaching/winter17/cs260/lectures/lec13.pdf

QUESTION:

Recall that AdaBoost learns a classifier H using a weighted sum of weak learners h_t as follows

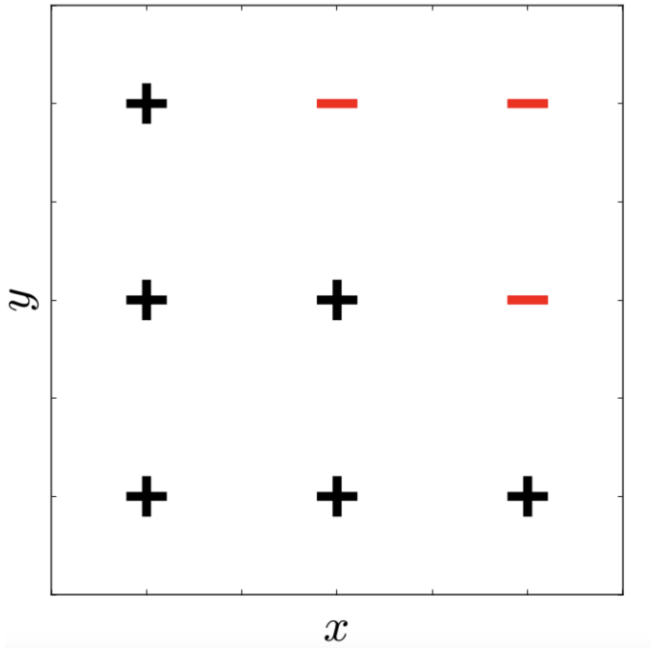
$$H(x) = sgn\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

In this question we will use decision trees as our weak learners, which classify a point as {1, −1} based on a sequence of threshold splits on its features (here *x, y*).

QUESTION:

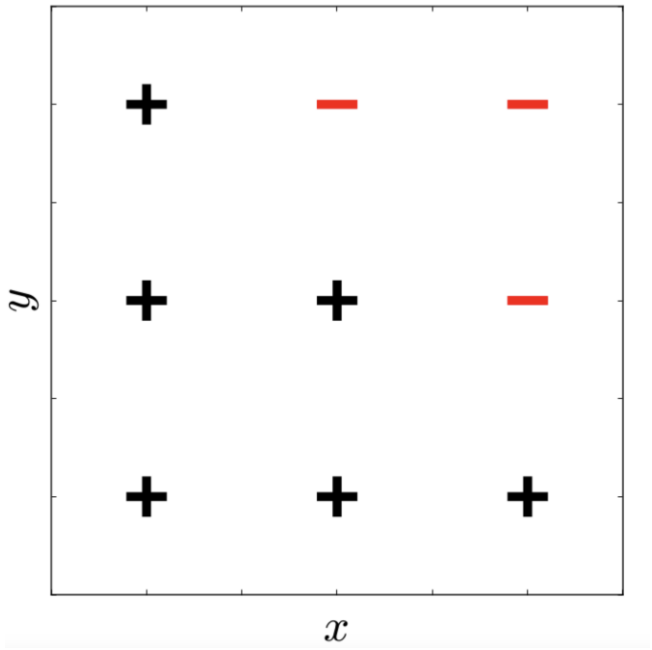
(a) Assume that our weak learners are decision trees of depth 1 (i.e. decision stumps), which minimize the weighted training error.

(i) Using the dataset below, draw the decision boundary learned by h_1.



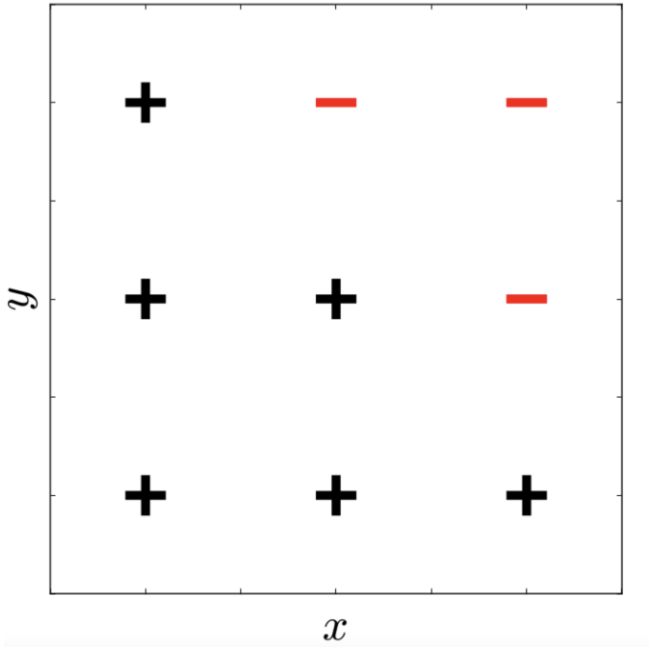
QUESTION:

(ii) On the dataset below, circle the point(s) with the highest weights on the second iteration, and draw the decision boundary learned by h_2. Explain the reasons.



QUESTION:

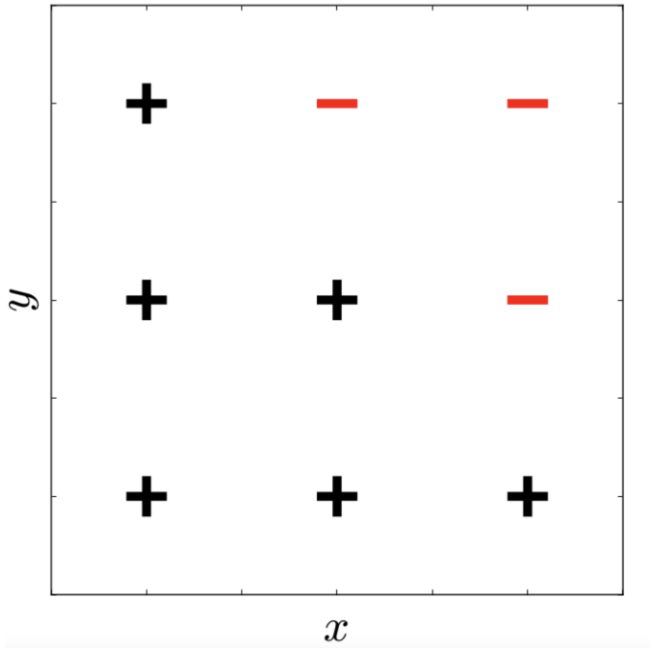
(iii) On the dataset below, draw the decision boundary of $H = sgn(\alpha_1 h_1 + \alpha_2 h_2)$. Explain the reasons.
(Hint, you do not need to explicitly compute the α's).



QUESTION:

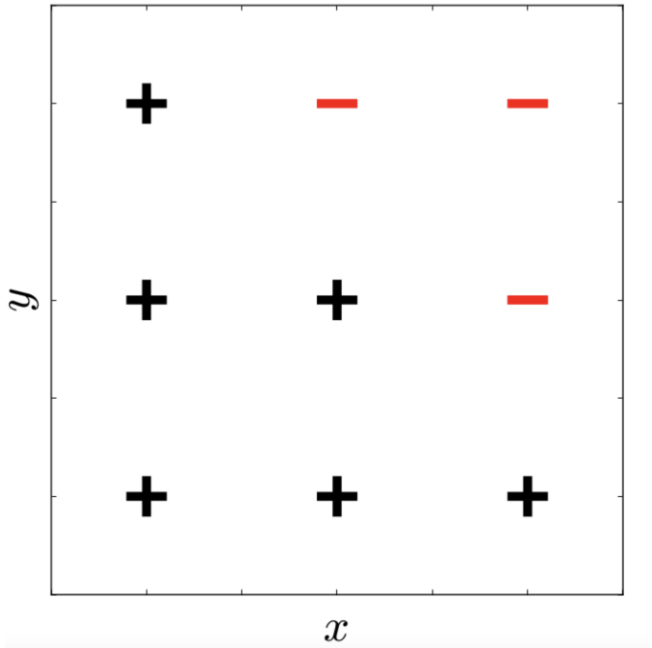
(b) Now assume that our weak learners are decision trees of maximium depth 2, which minimize the weighted training error.

(i) Using the dataset below, draw the decision boundary learned by h_1.



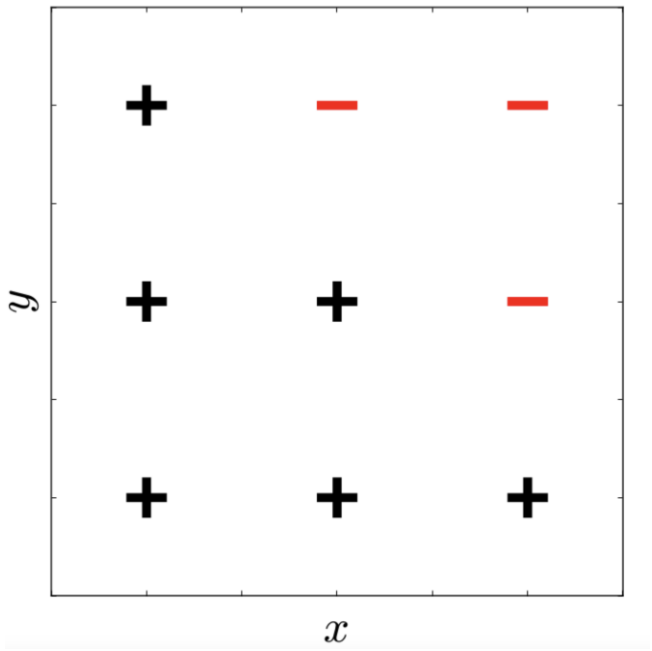
QUESTION:

(ii) On the dataset below, circle the point(s) with the highest weights on the second iteration, and draw the decision boundary learned by h_2. Explain the reasons.



QUESTION:

(iii) On the dataset below, draw the decision boundary of $H = sgn(\alpha_1 h_1 + \alpha_2 h_2)$. Explain the reasons. (Hint, you do not need to explicitly compute the α's).



Questions:

7. Recommendation Systems

Content-based Systems

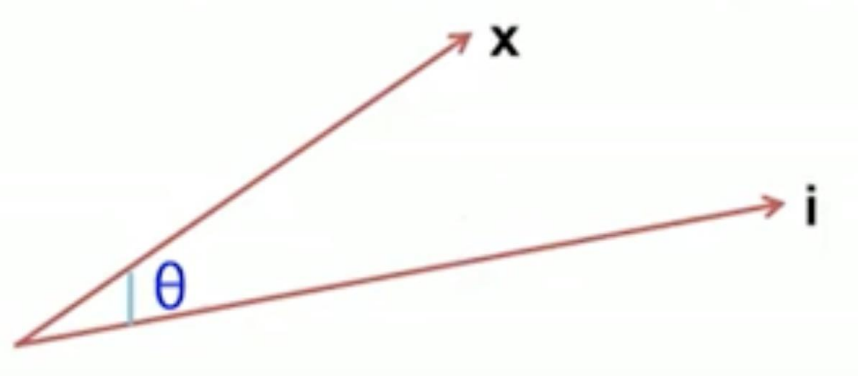
Collaborative Filtering

- User-user CF
- Item-item CF

…

COSINE SIMILARITY

- Estimate $u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$



For convenience, we use $\cos(\theta)$ as our similarity measure and call it the "cosine similarity" in this context.

USER-USER CF
ITEM-ITEM CF

Predict by taking
(1) cosine similarity
(2) weighted average

users

movies

	1	2	3	4	**5**	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- estimate rating of movie **1** by user **5**

ITEM-ITEM CF (|N|=2)

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		**2.6**	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

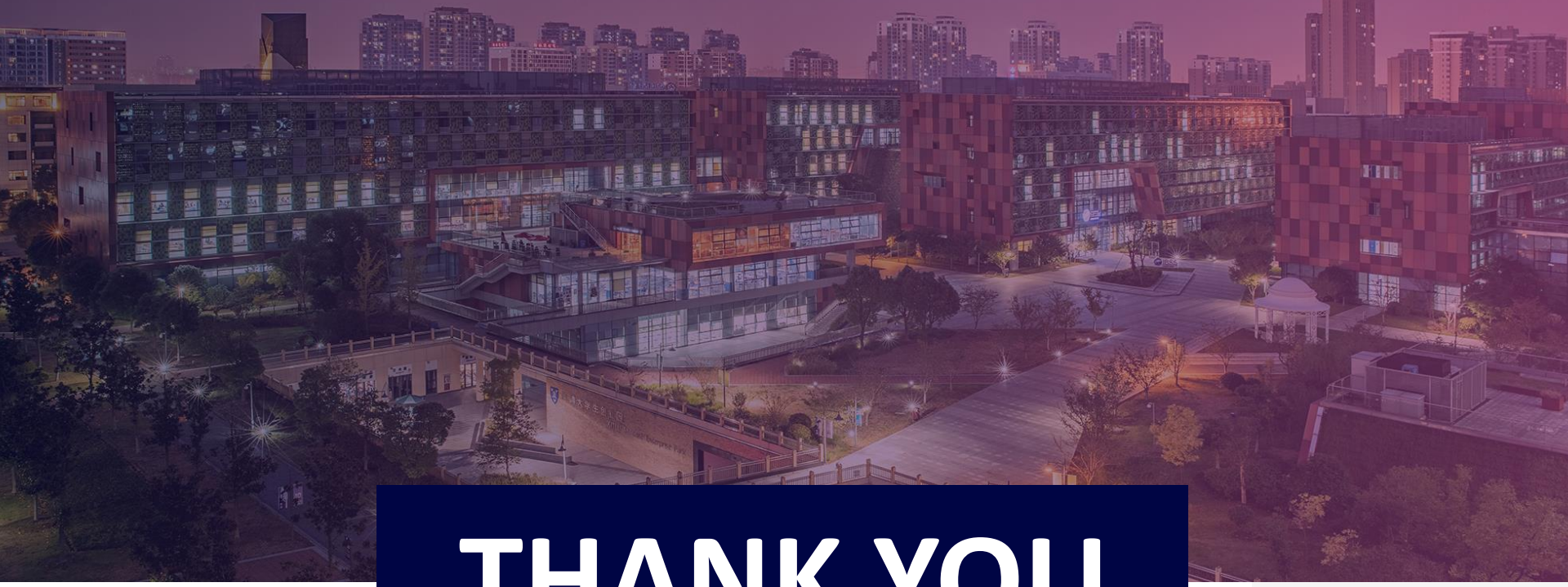
Predict by taking weighted average:

$r_{1.5}$ = **(0.41*2 + 0.59*3) / (0.41+0.59) = 2.6**

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

REVIEW TIPS

1. Review the slides.
2. The investigated knowledge points are all given on the LM.
3. The exam questions are similar to the questions in the slides.



THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學