Introduction to Computer Science & Engineering

Lecture 9: Information Systems

Jeonghun Park

KNU KYUNGPOOK NATIONAL UNIVERSITY

Managing Information

- Information system
  - Software that helps the user organize and analyze data
  - Software tools that allow the user to organize, manage, and analyze data in various ways

jeonghun.park@knu.ac.kr

Spreadsheets

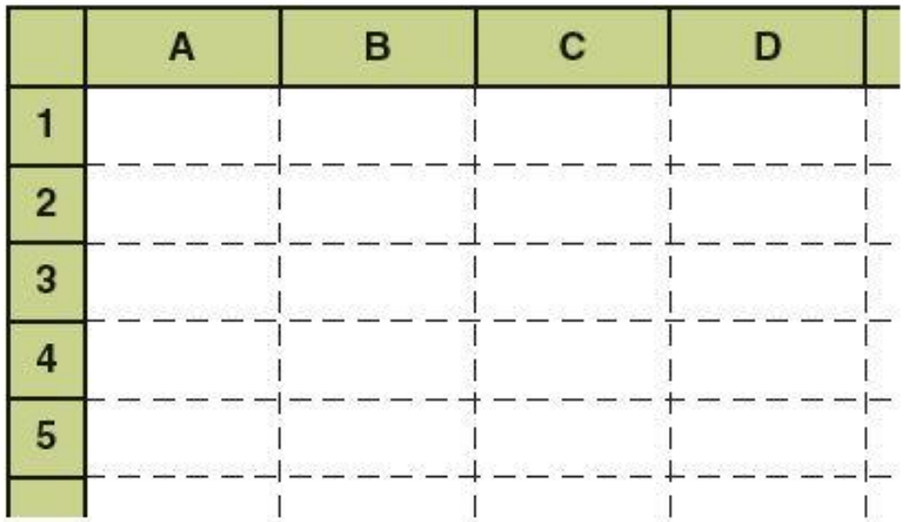


FIGURE 12.1 A spreadsheet, made up of a grid of labeled cells

- Spreadsheet
  - A software application that allows the user to organize and analyze data using a grid of labeled **cells**
  - A cell can contain data or a formula that is used to calculate a value
  - Data stored in a cell can be text, numbers, or “special” data such as dates
  - Spreadsheet cells are referenced by their row and column designation

Spreadsheets

- Suppose we have collected data on the number of students that came to get help from a set of tutors over a period of several weeks

엑셀로 flexible 한
modify가 가능.

	A	B	C	D	E	F	G	H
1								
2				Tutor				
3			Hal	Amy	Frank	Total	Avg	
4		1	12	10	13	35	11.67	
5		2	14	16	16	46	15.33	
6	Week	3	10	18	13	41	13.67	
7		4	8	21	18	47	15.67	
8		5	15	18	12	45	15.00	
9		Total	59	83	72	214	71.33	
10		Avg	11.80	16.60	14.40	42.80	14.27	
11								
12								

FIGURE 12.2 A spreadsheet containing data and computations

Spreadsheet Formulas

- The power of spreadsheets comes from the formulas that we can create and store in cells
  - When a formula is stored in a cell, the *result* of the formula is displayed in the cell
  - If we've set up the spreadsheet correctly, we could
    - add or remove tutors.
    - add additional weeks of data.
    - change any of the data we have already stored and the corresponding calculations would automatically be updated.

Spreadsheet Formulas

formulas

	A	B	C	D	E	F	G	H
1								
2				Tutor				
3			Hal	Amy	Frank	Total	Avg	
4		1	12	10	13	35	11.67	
5		2	14	16	16	46	15.33	
6	Week	3	10	18	13	41	13.67	
7		4	8	21	18	47	15.67	
8		5	15	18	12	45	15.00	
9		Total	59	83	72	214	71.33	
10		Avg	11.80	16.60	14.40	42.80	14.27	
11								
12								

=SUM(C4..E4)

=F7/COUNT(C7..E7)

=F9/COUNT(C4..E8)

=SUM(C4..C8)

=E9/COUNT(E4..E8)

FIGURE 12.3 The formulas behind some of the cells

Spreadsheet Formulas

Function	Computes
SUM(val1, val2, ...) SUM(range)	Sum of the specified set of values
COUNT(val1, val2, ...) COUNT(range)	Count of the number of cells that contain values
MAX(val1, val2, ...) MAX(range)	Largest value from the specified set of values
SIN(angle)	The sine of the specified angle
PI()	The value of PI
STDEV(val1, val2, ...) STDEV(range)	The standard deviation from the specified sample values
TODAY()	Today's date
LEFT(text, num_chars)	The leftmost characters from the specified text
IF(test, true_val, false_val)	If the test is true, it returns the true_val; otherwise, it returns the false_val
ISBLANK (value)	Returns true if the specified value refers to an empty cell

FIGURE 12.4 Some common spreadsheet functions

You might use Excel for this. Now use MATLAB for more sophisticated analysis!

Spreadsheet Analysis

통계적으로 접근하는것이

스프레드시트의 장점이다.

- Track sales
- Analyze sport statistics
- Maintain student grades
- Keep a car maintenance log
- Record and summarize travel expenses
- Track project activities and schedules
- Plan stock purchases

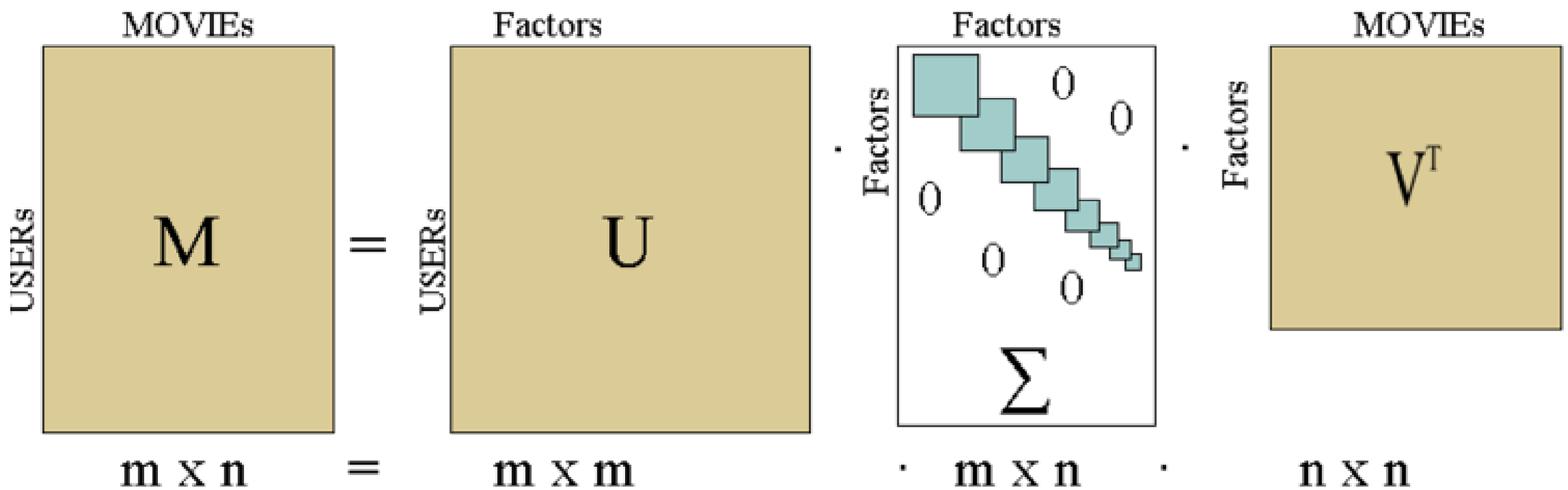
Netflix Problem!

- The reason why we have to learn math & computer engineering to do something nice
- The **Netflix Prize** was an open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films, i.e. without the users or the films being identified except by numbers assigned for the contest.
  - $1,000,000!!

즉

Matrix Factorization Approach

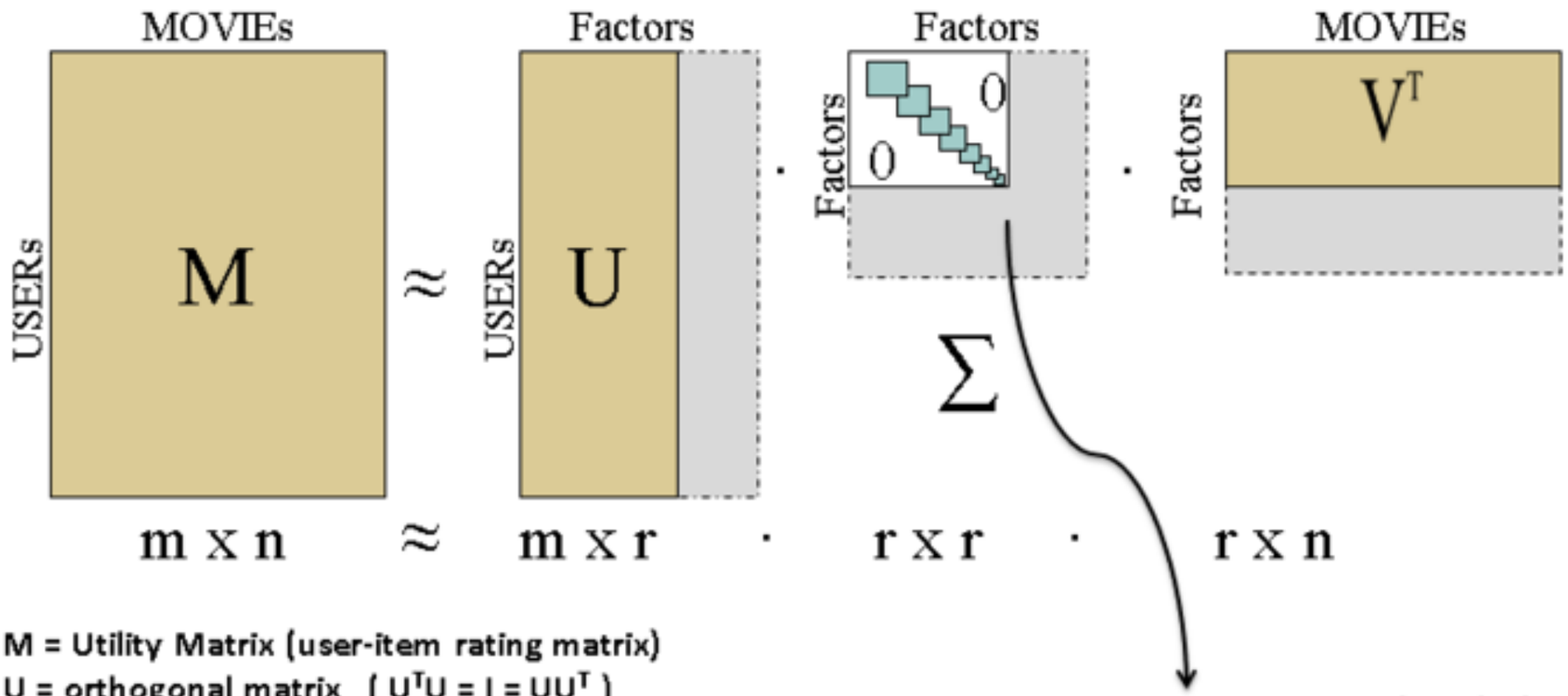
FULL SVD decomposition



M = Utility Matrix (user-item rating matrix)
U = orthogonal matrix ($U^TU = I = UU^T$)
∑ = diagonal matrix (diagonal elements show weight of Factors)
V = orthogonal matrix ($V^TV = I = VV^T$)

Matrix Factorization Approach

Truncated SVD decomposition



M = Utility Matrix (user-item rating matrix)
U = orthogonal matrix ($U^TU = I = UU^T$)
∑ = diagonal matrix (diagonal elements show weight of Factors)
V = orthogonal matrix ($V^TV = I = VV^T$)

Dimensionality Reduction !

1. Make calculations cheaper (More Scalability)
2. Get rid of noise

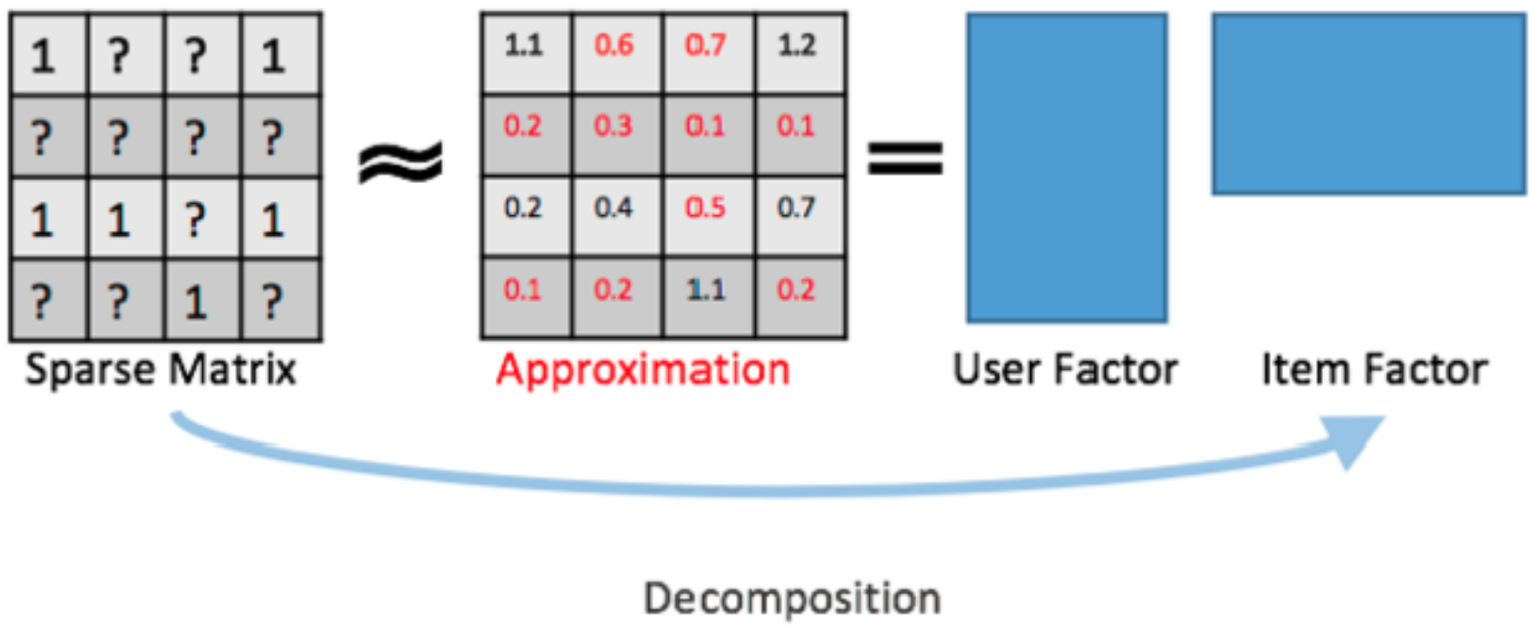
r is selected by users.

r < k = rank(M)

Do it in R : svd (base)

Matrix Factorization Approach

Anything else for Sparsity problem?



long user/product characteristics can be simplified to few latent factors.

Remember SVD decomposition.
- imagine ∑ is folded into U and V.
- Then we have only 2 matrixes.

Lesson

- Study hard!