Matrix Factorization In Recommender Systems

Yong Zheng, PhDc
Center for Web Intelligence, DePaul University, USA

March 4, 2015

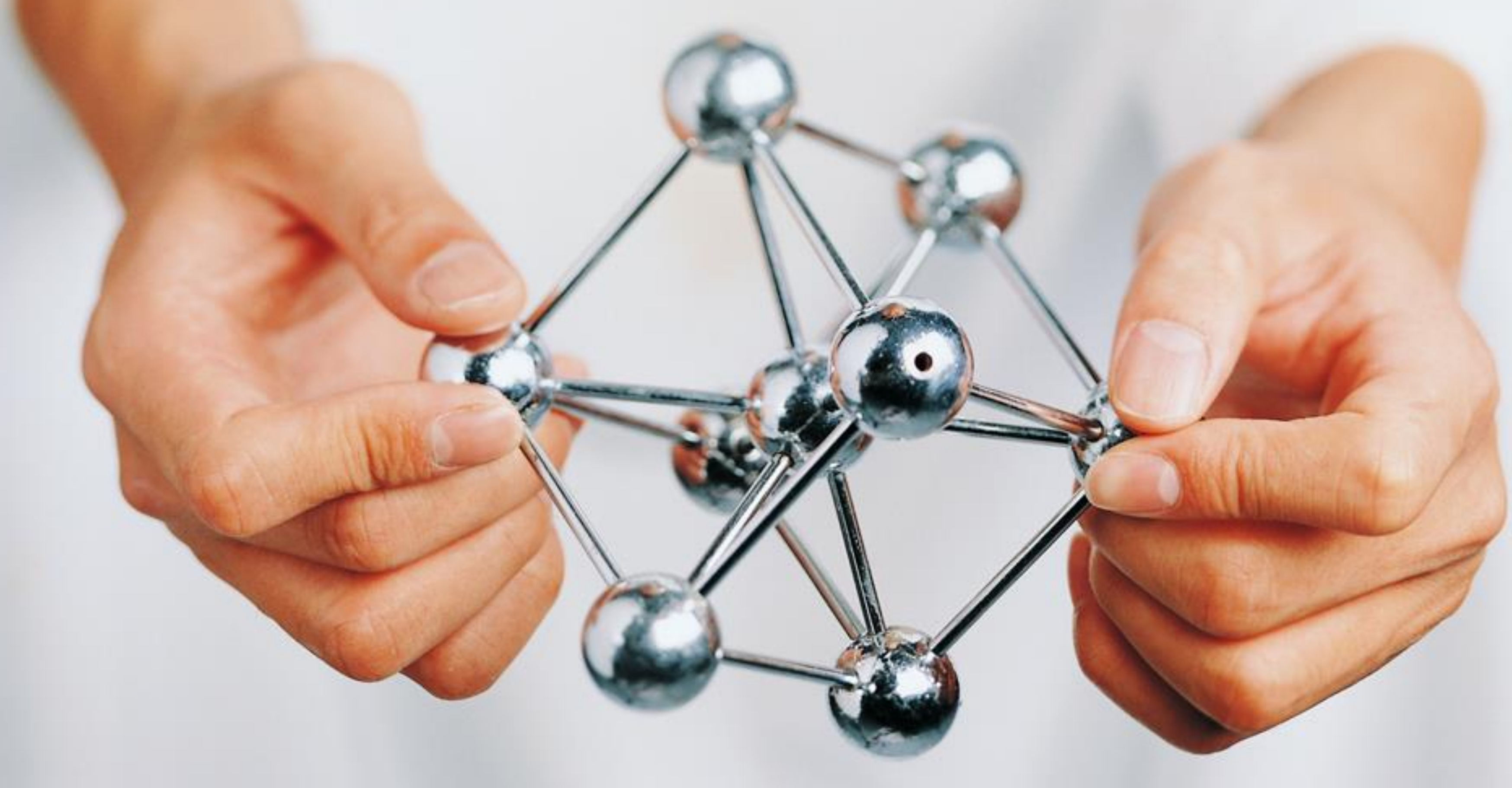


Table of Contents

- **Background: Recommender Systems (RS)**
- **Evolution of Matrix Factorization (MF) in RS**

  PCA ➔ SVD ➔ Basic MF ➔ Extended MF
- **Dimension Reduction: PCA & SVD**

  Other Applications: image processing, etc
- **An Example: MF in Recommender Systems**

  Basic MF and Extended MF

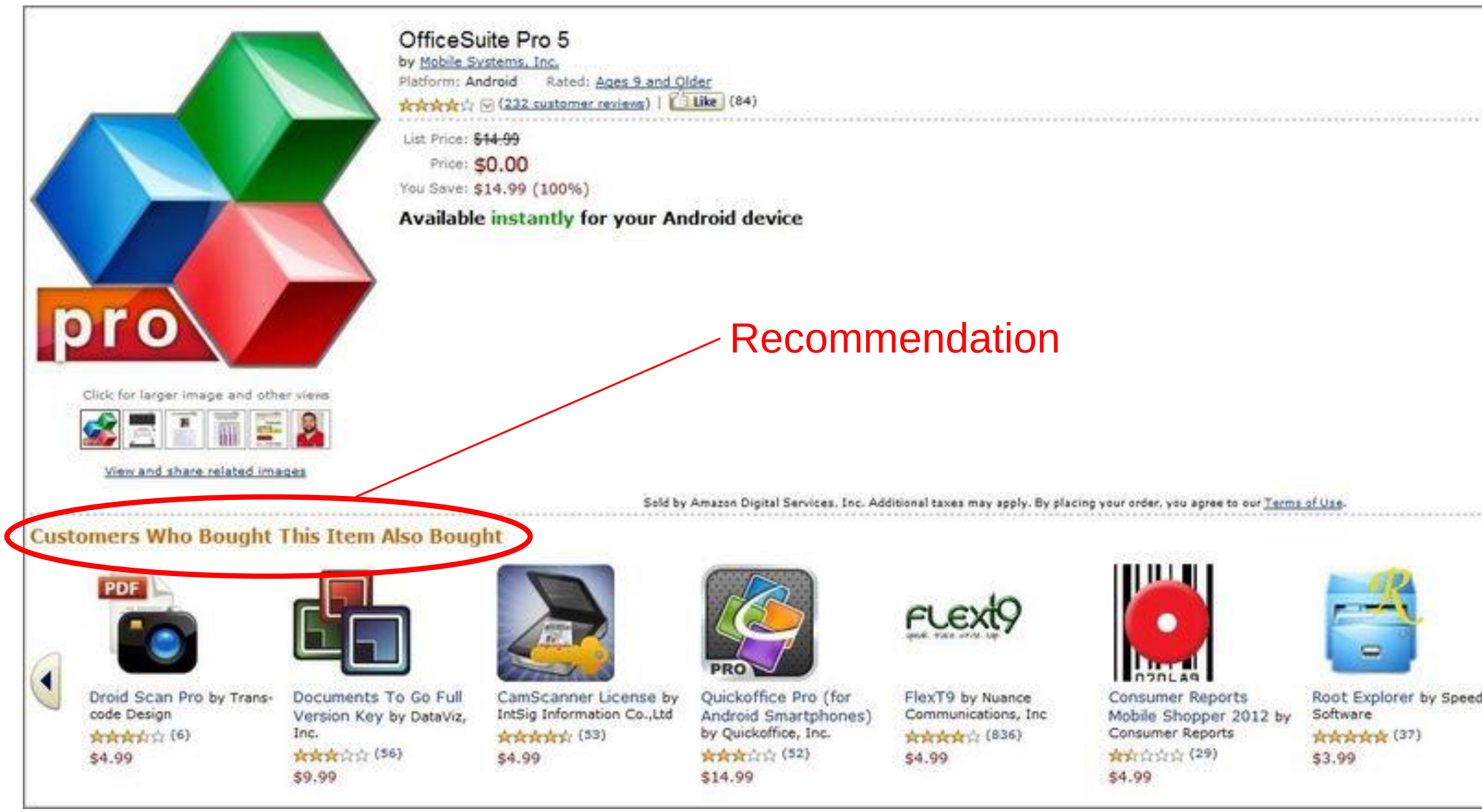
Table of Contents

1. Recommender System (RS)

- Definition: RS is a system able to provide or suggest items to the end users.

1. Recommender System (RS)

- Function: Alleviate information overload problems



1. Recommender System (RS)

- Function: Alleviate information overload problems

Social RS (Twitter)

Tagging RS (Flickr)

1. Recommender System (RS)

- Task-1: Rating Predictions

User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	?	2	4
U3	4	2	?

- Task-2: Top-N Recommendation

  Provide a short list of recommendations;
  For example, top-5 twitter users, top-10 news;

1. Recommender System (RS)

- Evolution of Recommendation algorithms

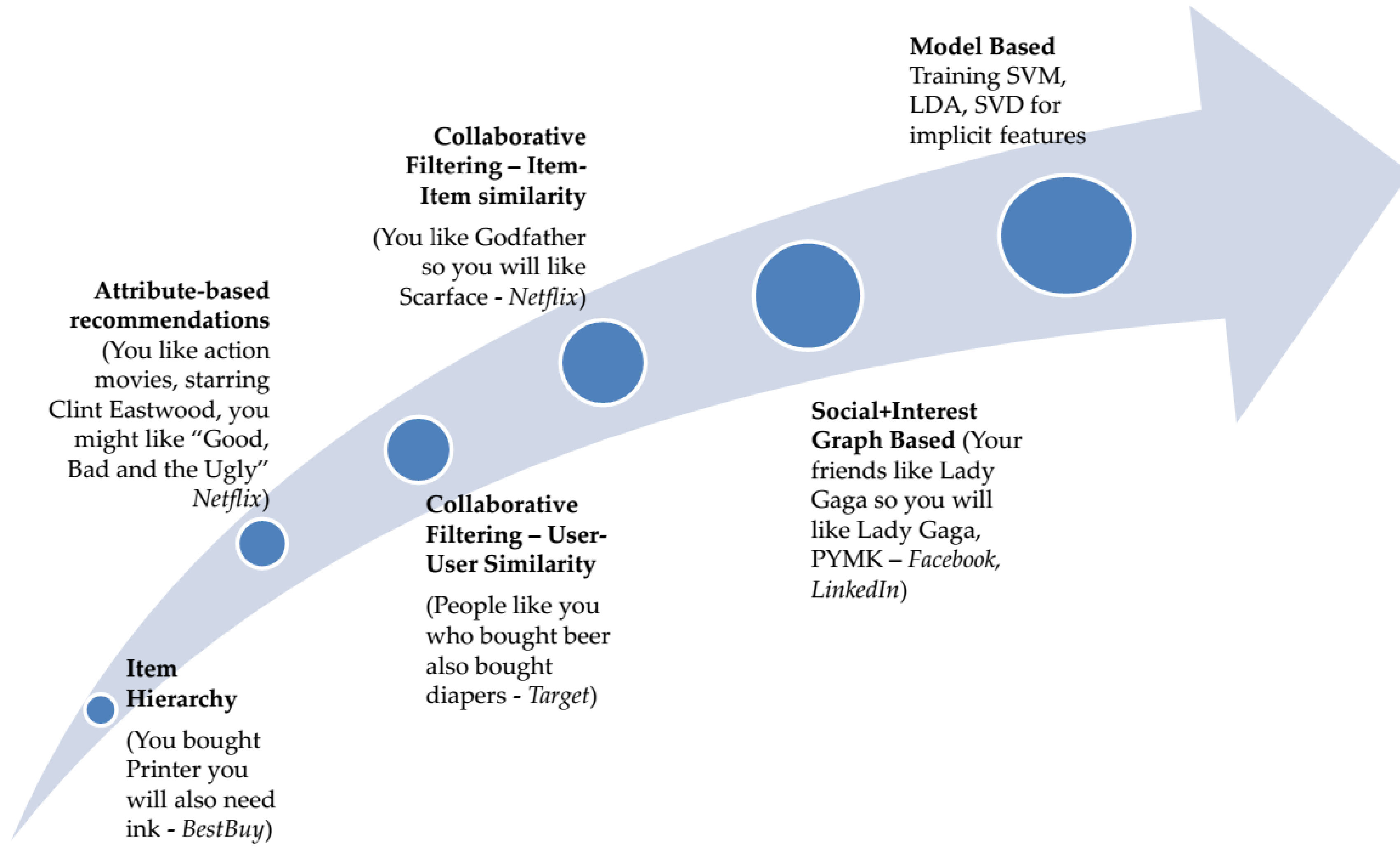


Table of Contents

2. Matrix Factorization In RS

- Why we need MF in RS?

The very beginning: dimension reduction

Amazon.com: thousands of users and items;
Netflix.com: thousands of users and movies;

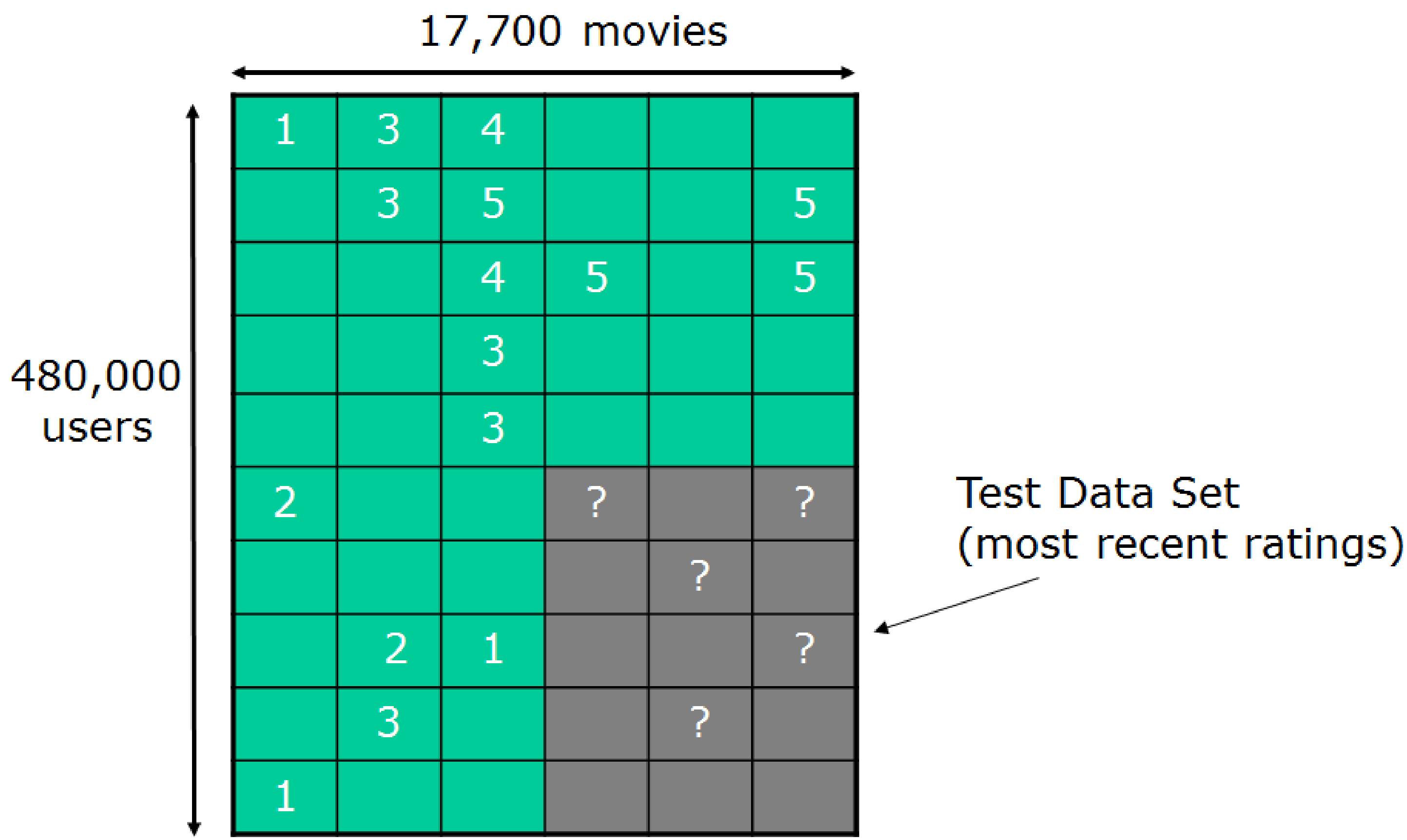
How large the rating matrix it is ??????????
Computational costs ➔ high !!!

2. Matrix Factorization In RS

- Netflix Prize ($1 Million Contest), 2006-2009

2. Matrix Factorization In RS

- Netflix Prize ($1 Million Contest), 2006-2009



2. Matrix Factorization In RS

- Netflix Prize ($1 Million Contest), 2006-2009

How about using BigData mining, such as MapReduce?

2003 - Google launches project Nutch
2004 - Google releases papers with MapReduce
2005 - Nutch began to use GFS and MapReduce
2006 - Yahoo! created Hadoop based on GFS & MapReduce
2007 - Yahoo started using Hadoop on a 1000 node cluster
2008 - Apache took over Hadoop
2009 - Hadoop was successfully process large-scale data
2011 - Hadoop releases version 1.0

2. Matrix Factorization In RS

- Evolution of Matrix Factorization (MF) in RS

Dimension Reduction Techniques: PCA & SVD

↓

SVD-Based Matrix Factorization (SVD)

Early Stage

↓

Modern Stage

Basic Matrix Factorization

↓

Extended Matrix Factorization

Table of Contents

3. Dimension Reduction

- Why we need dimension reduction?

The curse of dimensionality: various problems that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as 2D or 3D spaces.

- Applications

Information Retrieval: Web documents, where the dimensionality is the vocabulary of words

Recommender System: Large scale of rating matrix

Social Networks: Facebook graph, where the dimensionality is the number of users

Biology: Gene Expression

Image Processing: Facial recognition

3. Dimension Reduction

- There are many techniques for dimension reduction

Linear Discriminant Analysis (LDA) tries to identify attributes that account for the most variance between classes. In particular, LDA, in contrast to PCA, is a supervised method, using known class labels.

Principal Component Analysis (PCA) applied to this data identifies the combination of linearly uncorrelated attributes (principal components, or directions in the feature space) that account for the most variance in the data. Here we plot the different samples on the 2 first principal components.

Singular Value Decomposition (SVD) is a factorization of a real or complex matrix. Actually SVD was derived from PCA.

3. Dimension Reduction

- Brief History

Principal Component Analysis (PCA)

- Draw a plane closest to data points (Pearson, 1901)
- Retain most variance of the data (Hotelling, 1933)

Singular Value Decomposition (SVD)

- Low-rank approximation (Eckart-Young, 1936)
- Practical applications or Efficient Computations (Golub-Kahan, 1965)

Matrix Factorization In Recommender Systems

- “Factorization Meets the Neighborhood: a Multifaceted. Collaborative Filtering Model”, by Yehuda Koren, ACM KDD 2008 (formally reach an agreement by this paper)

3. Dimension Reduction

Principal Component Analysis (PCA)

Assume we have a data with multiple features

1). Try to find principle components – each component is a combination of the linearly uncorrelated attributes/features;

2). PCA allows to obtain an ordered list of those components that account for the largest amount of the variance from the data;

3). The amount of variance captured by the first component is larger than the amount of variance on the second component, and so on.

4). Then, we can reduce the dimensionality by ignoring the components with smaller contributions to the variance.

3. Dimension Reduction

Principal Component Analysis (PCA)

How to obtain those principal components?

The basic principle or assumption in PCA is:
The eigenvector of a covariance matrix equal to a principal component, because the eigenvector with the largest eigenvalue is the direction along which the data set has the maximum variance.

Each eigenvector is associated with a eigenvalue;
Eigenvalue ➔ tells how much the variance is;
Eigenvector ➔ tells the direction of the variation;

The next step: how to get the covariance matrix and how to calculate the eigenvectors/eigenvalues?

3. Dimension Reduction

Principal Component Analysis (PCA)

Step by steps ➔Assume a data with 3 dimensions

Step1: Center the data by subtracting the mean of each column

Original Data

X	Y	Z
2.5	9.4	6.5
0.2	5.6	4.2
6	3.2	0.3
4.2	3.9	6.1
2.3	5	5.2
11	7	0.56
2.6	0.3	0.9
3.4	0.02	1.81
3.3	6.5	2.13
8	2	4.2

Mean(X) = 4.35
Mean (Y) = 4.292
Mean (Z) = 3.19

For example,
X11 = 2.5
Update:
X11 = 2.5-4.35 = -1.85

Transformed Data

X	Y	Z
-1.85	5.108	3.31
-4.15	1.308	1.01
1.65	-1.092	-2.89
-0.15	-0.392	2.91
-2.05	0.708	2.01
6.65	2.708	-2.63
-1.75	-3.992	-2.29
-0.95	-4.272	-1.38
-1.05	2.208	-1.06
3.65	-2.292	1.01

3. Dimension Reduction

Principal Component Analysis (PCA)

Step2: Compute covariance matrix based on the transformed data

X	Y	Z
-1.85	5.108	3.31
-4.15	1.308	1.01
1.65	-1.092	-2.89
-0.15	-0.392	2.91
-2.05	0.708	2.01
6.65	2.708	-2.63
-1.75	-3.992	-2.29
-0.95	-4.272	-1.38
-1.05	2.208	-1.06
3.65	-2.292	1.01

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

```
>> C=cov(M)

C =

   10.0228    0.0329   -3.0013
    0.0329    9.0331    2.7696
   -3.0013    2.7696    5.4495
```

Matlab function: cov (Matrix)

3. Dimension Reduction

Principal Component Analysis (PCA)

Step3: Calculate eigenvectors & eigenvalues from covariance matrix

```
>> C=cov(M)

C =

   10.0228    0.0329   -3.0013
    0.0329    9.0331    2.7696
   -3.0013    2.7696    5.4495
```

```
>> [V,D]=eig(C)

V =

    0.3603   -0.5838    0.7275
   -0.3863   -0.8033   -0.4533
    0.8491   -0.1177   -0.5150


D =

    2.9155         0         0
         0    9.4629         0
         0         0   12.1270
```

Matlab function:
[V,D] = eig (Covariance Matrix)
V = eigenvectors (each column)
D = eigenvalues (in the diagonal line)

For example, 2.9155 correspond to vector <0.3603, -0.3863, 0.8491>
Each eigenvector is considered as a principle component.

3. Dimension Reduction

Principal Component Analysis (PCA)

Step4: Order the eigenvalues from largest to smallest, where the eigenvectors will also be re-ordered; and then we can select the top-K one; for example, we set K=2

```
>> [V,D]=eig(C)

V =

    0.3603   -0.5838    0.7275
   -0.3863   -0.8033   -0.4533
    0.8491   -0.1177   -0.5150


D =

    2.9155         0         0
         0    9.4629         0
         0         0   12.1270
```

K=2, it indicates that we will reduce the dimensions to be 2. The last second columns are extracted to have an EigenMatrix

```
VM =

   -0.5838    0.7275
   -0.8033   -0.4533
   -0.1177   -0.5150
```

3. Dimension Reduction

Principal Component Analysis (PCA)

Step5: Project the original data to those eigenvectors to formulate the new data matrix

Original data, D, 10x3

X	Y	Z
2.5	9.4	6.5
0.2	5.6	4.2
6	3.2	0.3
4.2	3.9	6.1
2.3	5	5.2
11	7	0.56
2.6	0.3	0.9
3.4	0.02	1.81
3.3	6.5	2.13
8	2	4.2

Transformed data, TD, 10x3

X	Y	Z
-1.85	5.108	3.31
-4.15	1.308	1.01
1.65	-1.092	-2.89
-0.15	-0.392	2.91
-2.05	0.708	2.01
6.65	2.708	-2.63
-1.75	-3.992	-2.29
-0.95	-4.272	-1.38
-1.05	2.208	-1.06
3.65	-2.292	1.01

EigenMatrix, EM, 3X2

VM =

-0.5838	0.7275
-0.8033	-0.4533
-0.1177	-0.5150

FinalData (10xk) = TD (10x3) x EM (3xk), here k = 2

3. Dimension Reduction

Principal Component Analysis (PCA)

Step5: Project the original data to those eigenvectors to formulate the new data matrix

Original data, D, 10x3

X	Y	Z
2.5	9.4	6.5
0.2	5.6	4.2
6	3.2	0.3
4.2	3.9	6.1
2.3	5	5.2
11	7	0.56
2.6	0.3	0.9
3.4	0.02	1.81
3.3	6.5	2.13
8	2	4.2

After PCA

Final Data, FD, 10x2

```
FD =

   -3.4128   -5.3660
    1.2532   -4.1322
    0.2541    3.1837
    0.0600   -1.4301
    0.3915   -2.8475
   -5.7481    4.9648
    4.4980    1.7158
    4.1487    1.9561
   -1.0359   -1.2189
   -0.4086    3.1742
```

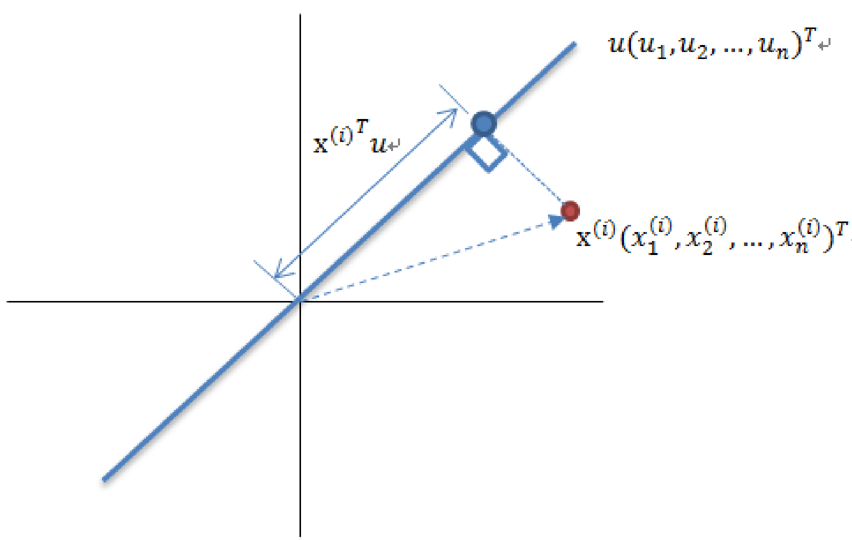
FinalData (10xk) = TD (10x3) x EM (3xk), here k = 2

3. Dimension Reduction

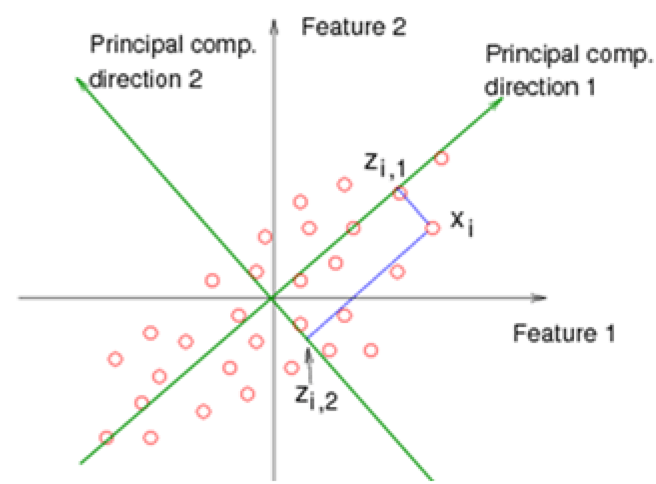
Principal Component Analysis (PCA)

Step5: Project the original data to those eigenvectors to formulate the new data matrix

The idea of Projection



Visualization of PCA



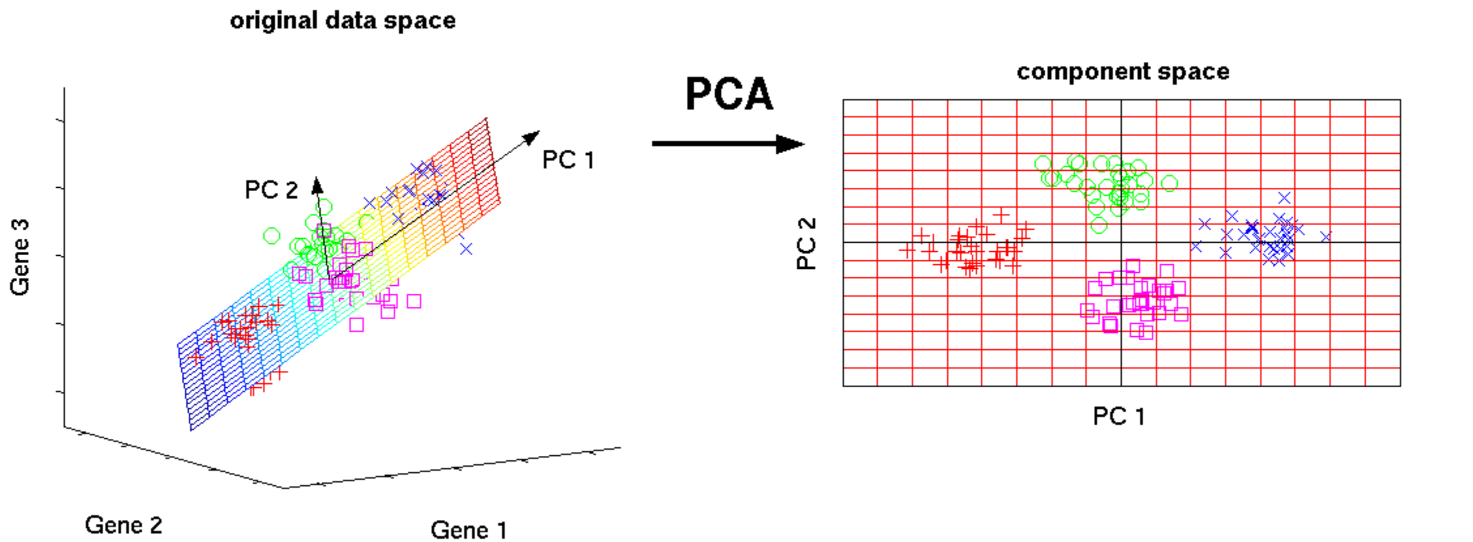
PCA finds a linear projection of high dimensional data into a lower dimensional subspace.

3. Dimension Reduction

Principal Component Analysis (PCA)

PCA reduces the dimensionality (the number of features) of a data set by maintaining as much variance as possible.

Another example: Gene Expression



The original expression by 3 genres is projected to two new dimensions, Such two-dimensional visualization of the samples allow us to draw qualitative conclusions about the separability of experimental conditions (marked by different colors).

3. Dimension Reduction

Singular Value Decomposition (SVD)

- SVD is another approach used for dimension reduction;
- Specifically, it is developed for Matrix Decomposition;
- SVD actually is another way to do PCA;
- The application of SVD to the domain of recommender systems was boosted by the Netflix Prize Contest

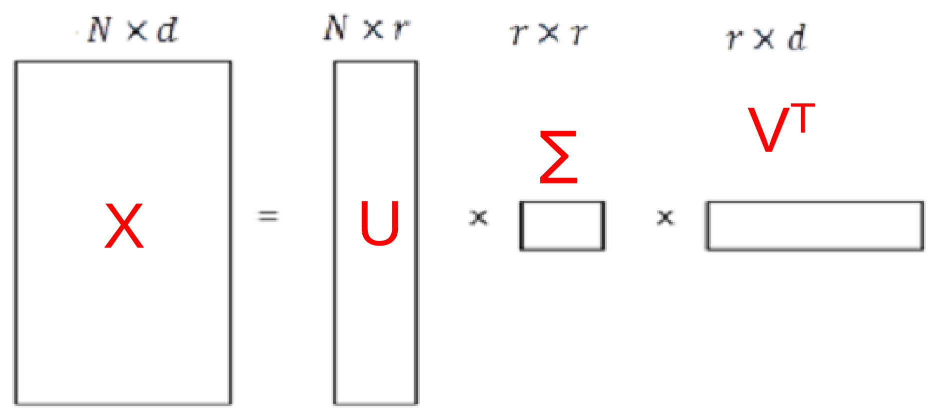
Winner: BELLKOR’S PRAGMATIC CHAOS

3. Dimension Reduction

Singular Value Decomposition (SVD)

Any N × d matrix X can be decomposed in this way:

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



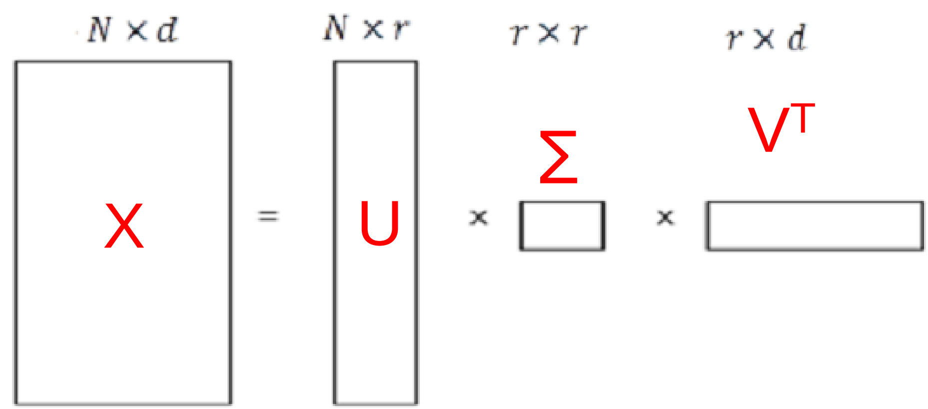
- r is the rank of matrix X, = the size of the largest collection of linearly independent columns or rows in matrix X ;
- U is a column-orthonormal N × r matrix;
- V is a column-orthonormal d × r matrix;
- Σ is a diagonal r × r matrix, where the singular values are sorted in descending order.

3. Dimension Reduction

Singular Value Decomposition (SVD)

Any N × d matrix X can be decomposed in this way:

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



Relations between PCA and SVD:

$C = \frac{1}{N-1} X^T X$ be the $d \times d$ covariance matrix.

The eigenvectors of C are the same as the right singular vectors of X. In other words, V contains eigenvectors, and the ordered eigenvalues are present in Σ.

3. Dimension Reduction

Singular Value Decomposition (SVD)

$$\text{Data } D = \begin{matrix} 10 & 20 & 10 \\ 2 & 5 & 2 \\ 8 & 17 & 7 \\ 9 & 20 & 10 \\ 12 & 22 & 11 \end{matrix}$$

$$D = U \Sigma V^t$$

$$\text{where } U = \begin{matrix} 0.50 & 0.14 & -0.19 \\ 0.12 & -0.35 & 0.07 \\ 0.41 & -0.54 & 0.66 \\ 0.49 & -0.35 & -0.67 \\ 0.56 & 0.66 & 0.27 \end{matrix}$$

$$\text{where } \Sigma = \begin{matrix} 48.6 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.2 \end{matrix}$$

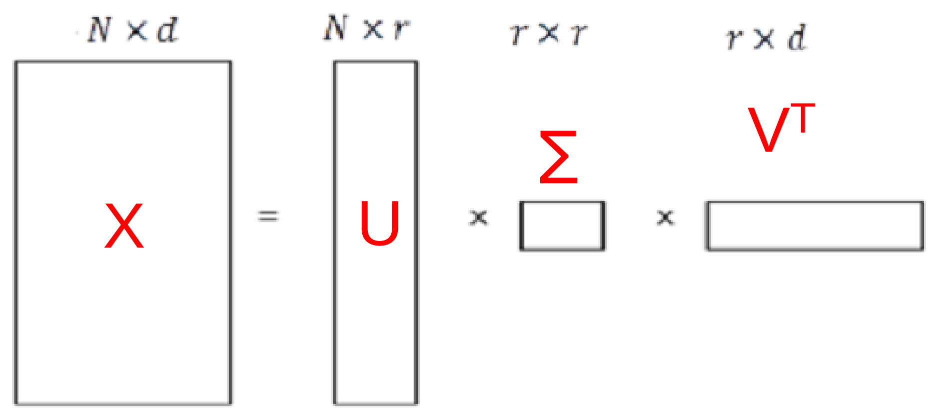
$$\text{and } V^t = \begin{matrix} 0.41 & 0.82 & 0.40 \\ 0.73 & -0.56 & 0.41 \\ 0.55 & 0.12 & -0.82 \end{matrix}$$

3. Dimension Reduction

Singular Value Decomposition (SVD)

Any N × d matrix X can be decomposed in this way:

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



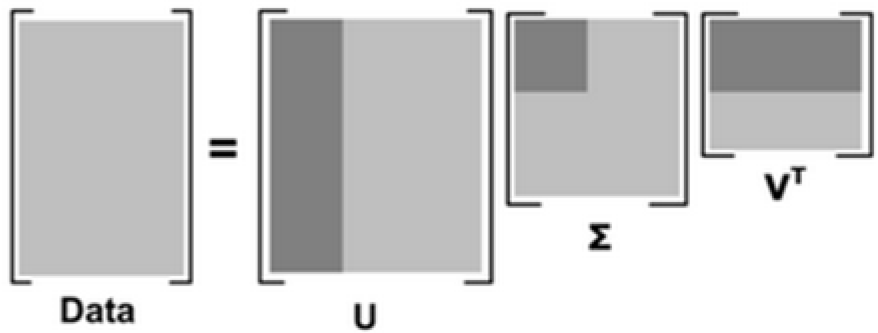
So, how to realize the dimension reduction in SVD?

Simply, based on the computation above, SVD tries to find a matrix to approximate the original matrix X by reducing the value of r – only the selected eigenvectors corresponding to the top-K largest eigenvalues will be obtained; other dimension will be discarded.

3. Dimension Reduction

Singular Value Decomposition (SVD)

So, how to realize the dimension reduction in SVD?



SVD is an optimization algorithm. Assume original data matrix is X, and SVD process is to find an approximation

$$\underset{m \times n}{D} \approx \underset{m \times k}{U} \; \underset{k \times k}{\Sigma} \; \underset{k \times n}{V^t}$$

minimize the Frobenius norm over all rank-k matrices $||X - D||_F$

3. Dimension Reduction

References

- Pearson, Karl. "Principal components analysis." The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 6.2 (1901): 559.
- De Lathauwer, L., et al. "Singular Value Decomposition." Proc. EUSIPCO-94, Edinburgh, Scotland, UK. Vol. 1. 1994.
- Wall, Michael E., Andreas Rechtsteiner, and Luis M. Rocha. "Singular value decomposition and principal component analysis." A practical approach to microarray data analysis. Springer US, 2003. 91-109.
- Sarwar, Badrul, et al. Application of dimensionality reduction in recommender system-a case study. No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science, 2000.

3. Dimension Reduction

Relevant Courses at CDM, DePaul University

CSC 424 Advanced Data Analysis
CSC 529 Advanced Data Mining
CSC 478 Programming Data Mining Applications
ECT 584 Web Data Mining for Business Intelligence

Table of Contents

- **Background: Recommender Systems (RS)**
- **Evolution of Matrix Factorization (MF) in RS**

  PCA ➔ SVD ➔ Basic MF ➔ Extended MF
- **Dimension Reduction: PCA & SVD**

  Other Applications: image processing, etc
- **An Example: MF in Recommender Systems**

  Basic MF and Extended MF

4. MF in Recommender Systems

- Evolution of Matrix Factorization (MF) in RS

Dimension Reduction Techniques: PCA & SVD

↓

SVD-Based Matrix Factorization (SVD)

Early Stage

Modern Stage

↓

Basic Matrix Factorization

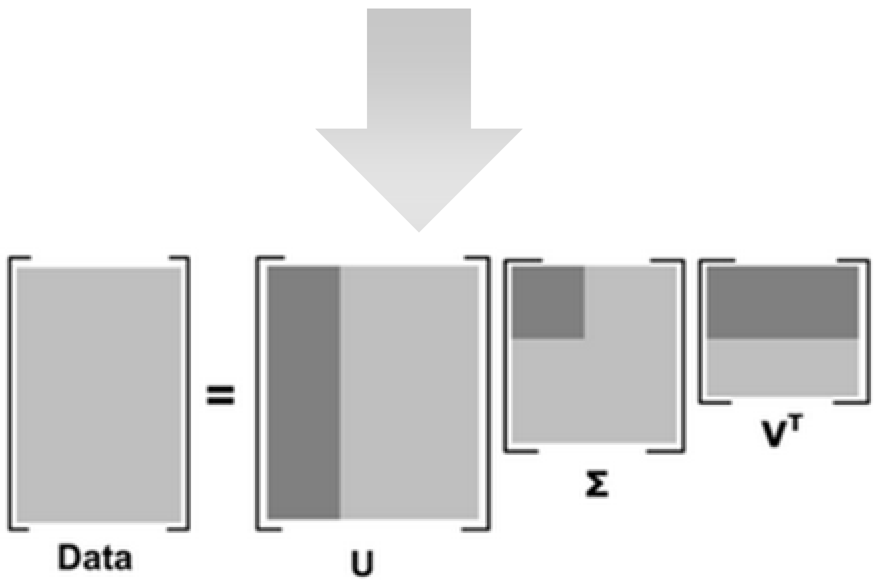
↓

Extended Matrix Factorization

4. MF in Recommender Systems

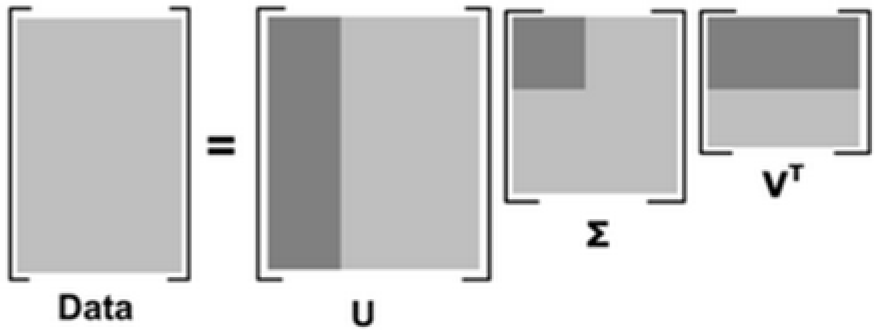
- From SVD to Matrix Factorization

User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	?	2	4
U3	4	2	?



4. MF in Recommender Systems

- From SVD to Matrix Factorization



Rating Prediction function in SVD for Recommendation

$$C_{P_{pred}} = \overline{C} + U_K . \sqrt{S_k}'(c) \cdot \sqrt{S_k} . V_k'(P) .$$

C is a user, P is the item (e.g. movie)

We create two new matrices: $U_K . \sqrt{S_k}'$ and $\sqrt{S_k} . V_k'$

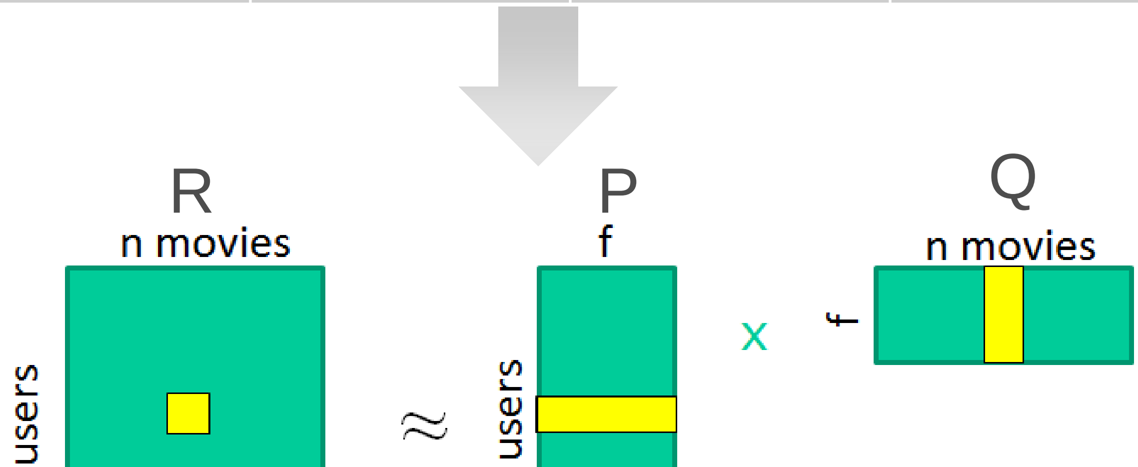
They are considered as user and item matrices

We extract the corresponding row (by c) and column (by p) from those matrices for computation purpose.

4. MF in Recommender Systems

- From SVD to Matrix Factorization

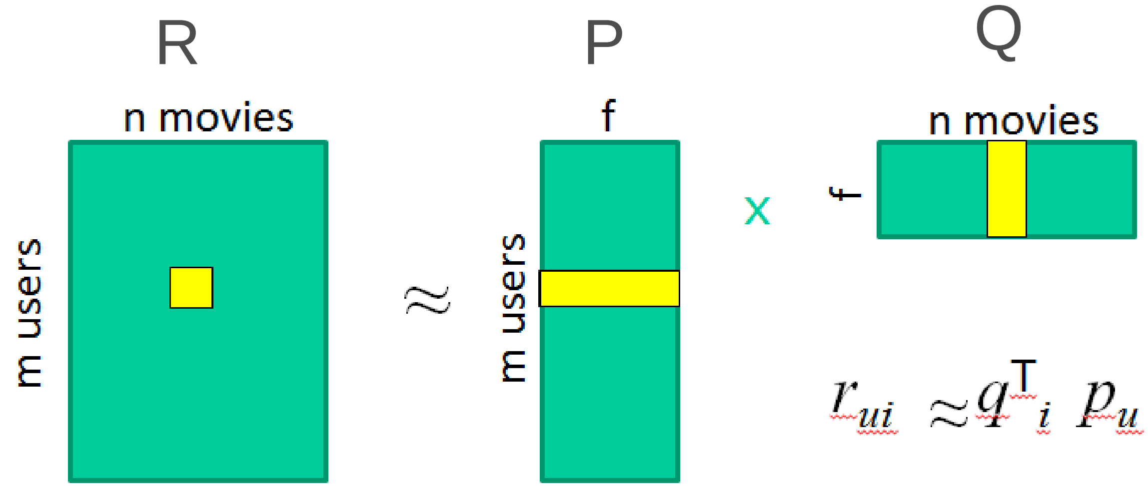
User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	?	2	4
U3	4	2	?



$$r_{ui} \approx q^T{}_i \; p_u$$

4. MF in Recommender Systems

- Basic Matrix Factorization



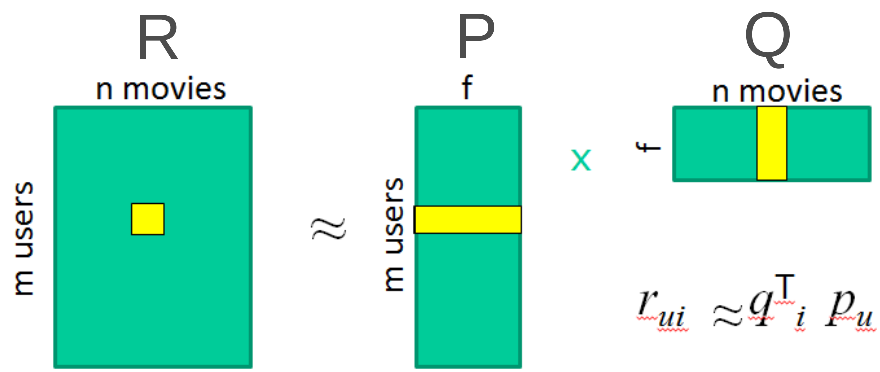
$$r_{ui} \approx q^{T}_{i}\, p_u$$

R = Rating Matrix, m users, n movies;
P = User Matrix, m users, f latent factors/features;
Q = Item Matrix, n movies, f latent factors/features;

A rating r_{ui} can be estimated by dot product of user vector p_u and item vector q_i.

4. MF in Recommender Systems

- Basic Matrix Factorization



$$r_{ui} \approx q^{T}_{i}\ p_u$$

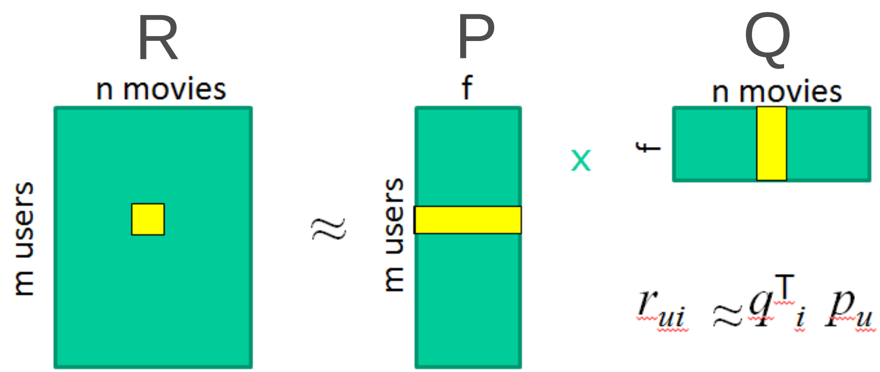
Interpretation:

p_u indicates how much user likes f latent factors;
q_i means how much one item obtains f latent factors;
The dot product indicates how much user likes item;

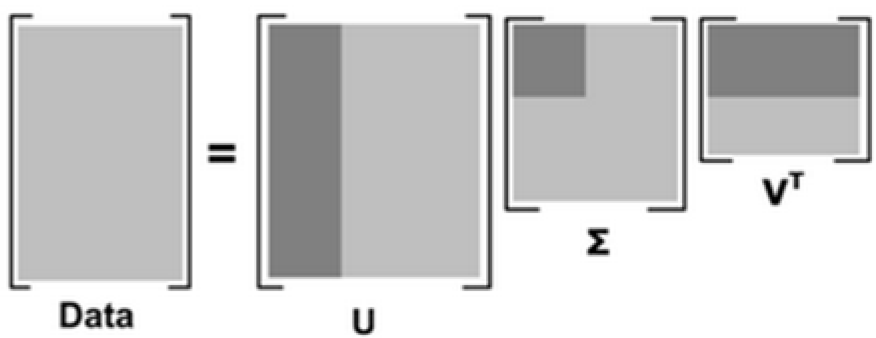
For example, the latent factor could be “Movie Genre”, such as action, romantic, comics, adventure, etc

4. MF in Recommender Systems

- Basic Matrix Factorization



$$r_{ui} \approx q^{T}_{i}\ p_u$$



Relation between SVD &MF:

P = user matrix

Q = item matrix

$U_K \cdot \sqrt{S_k}'$ = user matrix

$\sqrt{S_k} . V_k'$ = item matrix

4. MF in Recommender Systems

- Basic Matrix Factorization

Optimization: to learn the values in P and Q

$$\min_{q^*.p^*} \sum_{(u,i)\in k} (r_{ui} - x_{ui})^2$$

To learn the factor vectors (p_u and q_i)

Minimizing Cost Function (Least Squares Problem)

r_{ui} : actual rating for user u on item I
x_{ui} : predicted rating for user u on item I

Xui is the value from the dot product of two vectors

4. MF in Recommender Systems

- Basic Matrix Factorization

Optimization: to learn the values in P and Q

$$r_{ui} \approx q^{t}_{i}\ p_{u}, \quad \min_{q,p} \sum_{(u,i)\,\varepsilon\,R} (\, r_{ui} - q^{t}_{i}\ p_{u})$$

goodness of fit

regularization

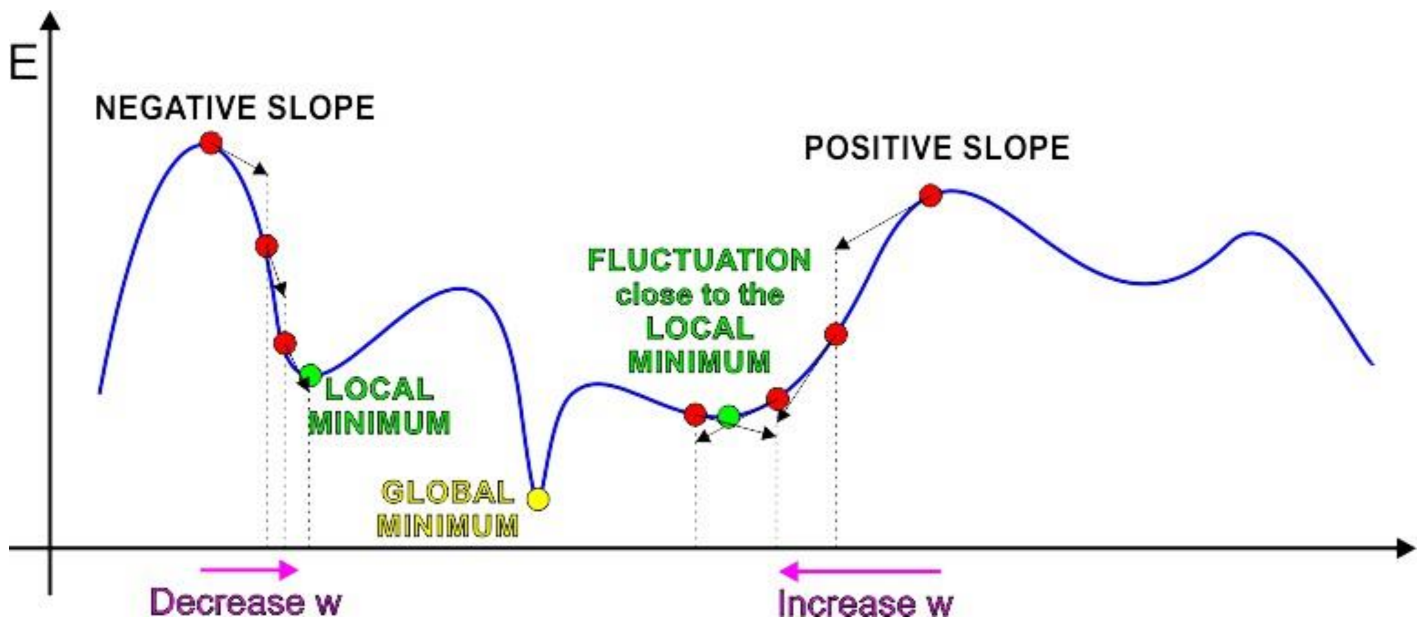
$$\min_{q,p} \sum_{(u,i)\,\varepsilon\,R} (\, r_{ui} - q^{t}_{i}\ p_{u})^{2} + \lambda\ (\,|q_{i}|^{2} + |p_{u}$$

Goodness of fit: to reduce the prediction errors;
Regularization term: to alleviate the overfitting;
The function above is called cost function;

4. MF in Recommender Systems

- Basic Matrix Factorization

Optimization using stochastic gradient descent (SGD)



4. MF in Recommender Systems

- Basic Matrix Factorization

Optimization using stochastic gradient descent (SGD)
Samples for updating the user and item matrices:

$$e_{ui} \overset{def}{=} r_{ui} - q_i^T p_u .$$

Then it modifies the parameters by a magnitude proportional to γ in the opposite direction of the gradient, yielding:

- $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$
- $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$

4. MF in Recommender Systems

- Basic Matrix Factorization – A Real Example

User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	?	2	4
U3	4	2	?

User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	0	2	4
U3	4	2	0

4. MF in Recommender Systems

- Basic Matrix Factorization – A Real Example

User	HarryPotter	Batman	Spiderman
U1	5	3	4
U2	0	2	4
U3	4	2	0

Set k=5, only 5 latent factors

	F1	F2	F3	F4	F5
U1	0.2763678	-0.37747	-1.26192	-1.54754	0.4738
U2	0.3999	-0.52747	-0.28946	-1.51597	0.73743
U3	0.2252336	-0.29125	-1.06624	-1.22463	0.37353

	HarryPotter	Batman	Spiderman
F1	0.301758	0.14297	0.43409
F2	-0.405407	-0.20245	-0.57514
F3	-1.399694	-0.9232	-0.46807
F4	-1.677619	-0.94013	-1.72021
F5	0.5118556	0.23861	0.79576

Predicted Rating (U3, Spiderman) =
Dot product of the two Yellow vectors = 3.16822

4. MF in Recommender Systems

- Extended Matrix Factorization

According to the purpose of the extension, it can be categorized into following contexts:

1). Adding Biases to MF

User bias, e.g. user is a strict rater = always low ratings
Item bias, e.g. a popular movie = always high ratings

$$r_{ui} \approx \mu + b_u + b_i + q_i^t p_u$$

μ	b_u	b_i
overall mean rating	Rating bias for user u	Rating bias for movie i

4. MF in Recommender Systems

- Extended Matrix Factorization

2). Adding Other influential factors

a). Temporal effects
For example, user’s rating given many years before may have less influences on predictions;
Algorithm: Time SVD++

b). Content profiles
For example, users or items share same or similar content (e.g. gender, user age group, movie genre, etc) may contribute to rating predictions;

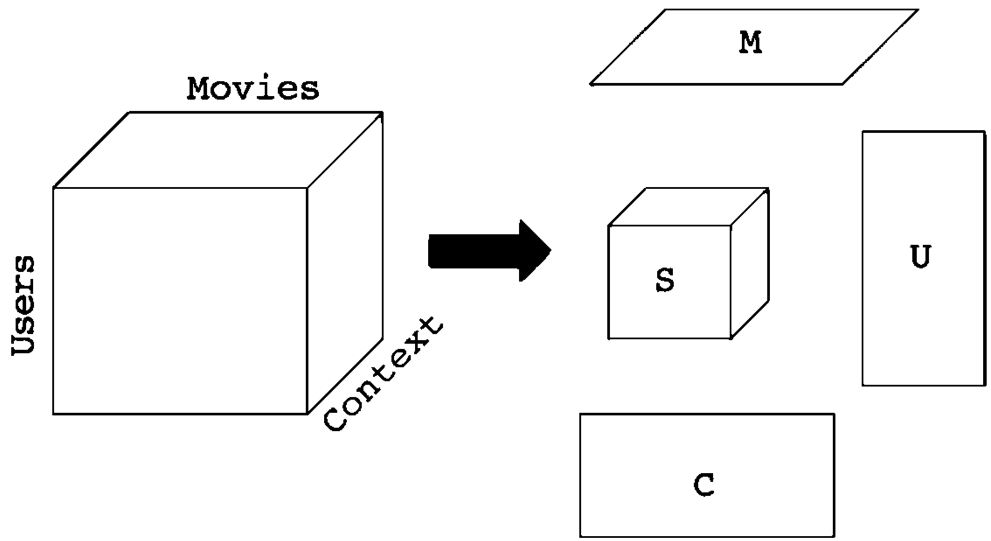
c). Contexts
Users’ preferences may change from contexts to contexts

d). Social ties from Facebook, twitter, etc

4. MF in Recommender Systems

- Extended Matrix Factorization

3). Tensor Factorization (TF)

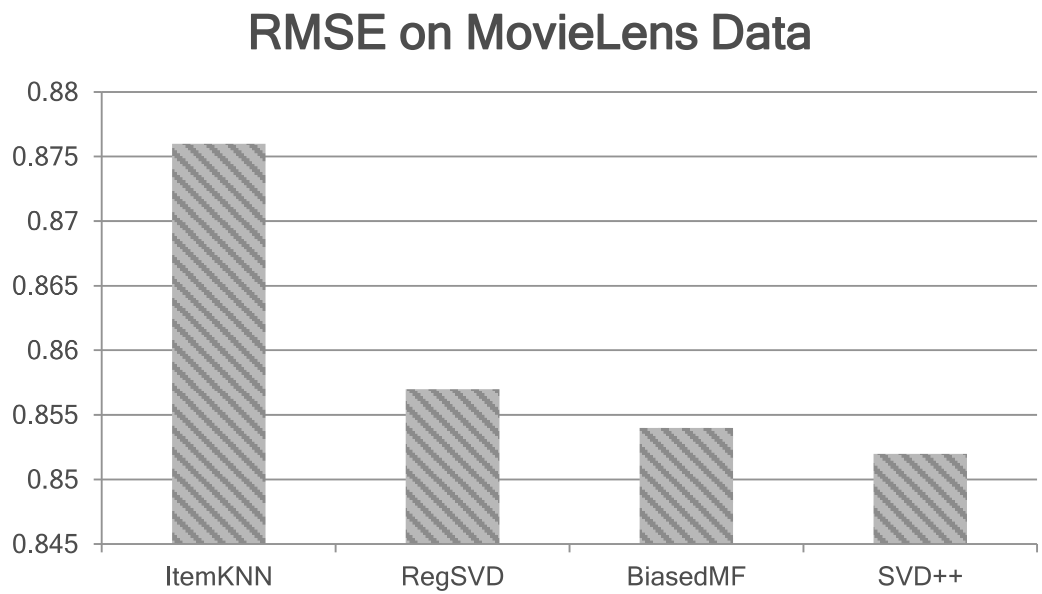


There could be more than 2 dimensions in the rating space – multidimensional rating space;

TF can be considered as multidimensional MF.

4. MF in Recommender Systems

- Evaluate Algorithms on MovieLens Data Set



ItemKNN = Item-Based Collaborative Filtering
RegSVD = SVD with regularization
BiasedMF = MF approach by adding biases
SVD++ = A complicated extension over MF

4. MF in Recommender Systems

- References

- Paterek, Arkadiusz. "Improving regularized singular value decomposition for collaborative filtering." Proceedings of KDD cup and workshop. Vol. 2007. 2007.
- Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.
- Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 8 (2009): 30-37.
- Karatzoglou, Alexandros, et al. "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering." Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010.
- Baltrunas, Linas, Bernd Ludwig, and Francesco Ricci. "Matrix factorization techniques for context aware recommendation." Proceedings of the fifth ACM conference on Recommender systems. ACM, 2011.
- Introduction to Matrix Factorization for Recommendation Mining, By Apache Mahount, https://mahout.apache.org/users/recommender/matrix-factorization.html

4. MF in Recommender Systems

- Relevant Courses at CDM, DePaul University

CSC 478 Programming Data Mining Applications
CSC 575 Intelligent Information Retrieval
ECT 584 Web Data Mining for Business Intelligence

Matrix Factorization In Recommender Systems

THANKS!