Department of Algorithms and Programming Technologies

MIPT
MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

Recommender Systems

BigDATAteam

Alexey Dral

Evgeny Frolov

APT dept., FIVT MIPT

Moscow, November-December 2017

Outline (Lectures)

1. Introduction into RecSys (RS). RS classification. Non-Personalised RS. Content-Based RS.

2. **Collaborative Filtering RS. UU and II CF. Explicit & Implicit Feedback. Matrix Factorization for RS**

3. Advanced RS (context-aware, tensor decomposition, …)

A word of caution

Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata

István Pilászy *
Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
pila@mit.bme.hu

Domonkos Tikk *,†
Dept. of Telecom. and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
tikk@tmit.bme.hu

ABSTRACT

The Netflix Prize (NP) competition gave much attention to collaborative filtering (CF) approaches. Matrix factorization (MF) based CF approaches assign low dimensional feature vectors to users and items. We link CF and content-based filtering (CBF) by finding a linear transformation that transforms user or item descriptions so that they are as close as possible to the feature vectors generated by MF for CF.

We propose methods for explicit feedback that are able to handle 140 000 features when feature vectors are very sparse. With movie metadata collected for the NP movies we show that the prediction performance of the methods is comparable to that of CF [illegible] used to predict user preferences on new [illegible]

We also [illegible] of movie metadata compared to movie rati[illegible] predictive power. We compare

1. INTRODUCTION

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recommendation is based on the former and current activity of the users, and metadata about users and items, if available.

There are two basic strategies that can be applied when generating recommendations. Collaborative filtering (CF) methods are based only on the activity of users, while content-based filtering (CBF) methods use only metadata. In this paper we propose hybrid methods, which try to benefit from both information sources.

The two most important families of CF methods are matrix factorization (MF) and neighbor-based approaches. Usually, the goal of MF is to find a low dimensional representation for both users and movies, i.e. each user and movie is associated with a feature vector. Movie metadata (which

NETFL[illegible]

Amatriain – July 2014 – Recommender Systems

credit to: Xavier Amatrian

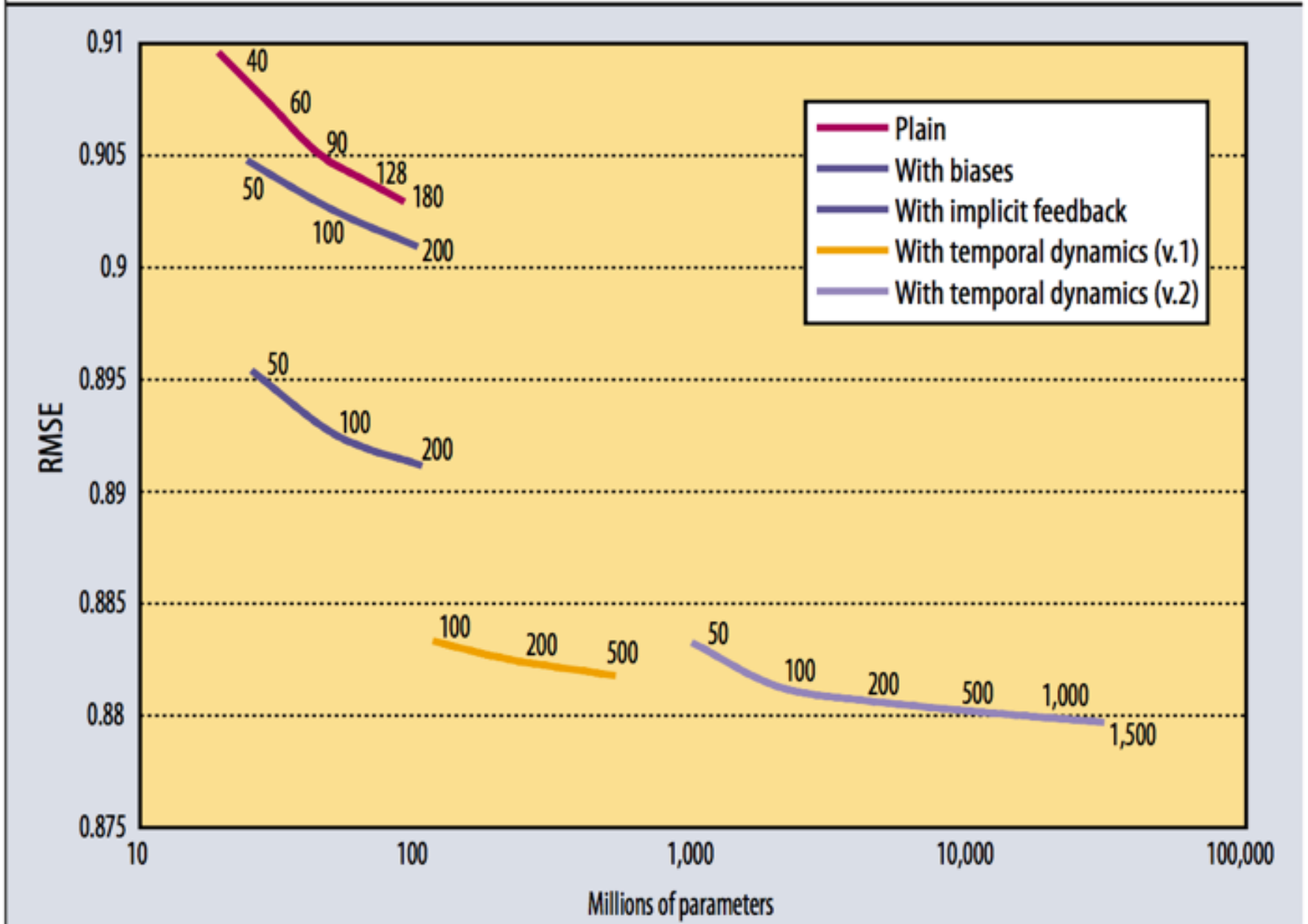


Figure 4. Matrix factorization models' accuracy. The plots show the root-mean-square error of each of four individual factor models (lower is better). Accuracy improves when the factor model's dimensionality (denoted by numbers on the charts) increases. In addition, the more refined factor models, whose descriptions involve more distinct sets of parameters, are more accurate. For comparison, the Netflix system achieves RMSE = 0.9514 on the same dataset, while the grand prize's required accuracy is RMSE = 0.8563.

Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems, IEEE Computer, pp. 42-49, August 2009.

From CF to MF RS: Goals

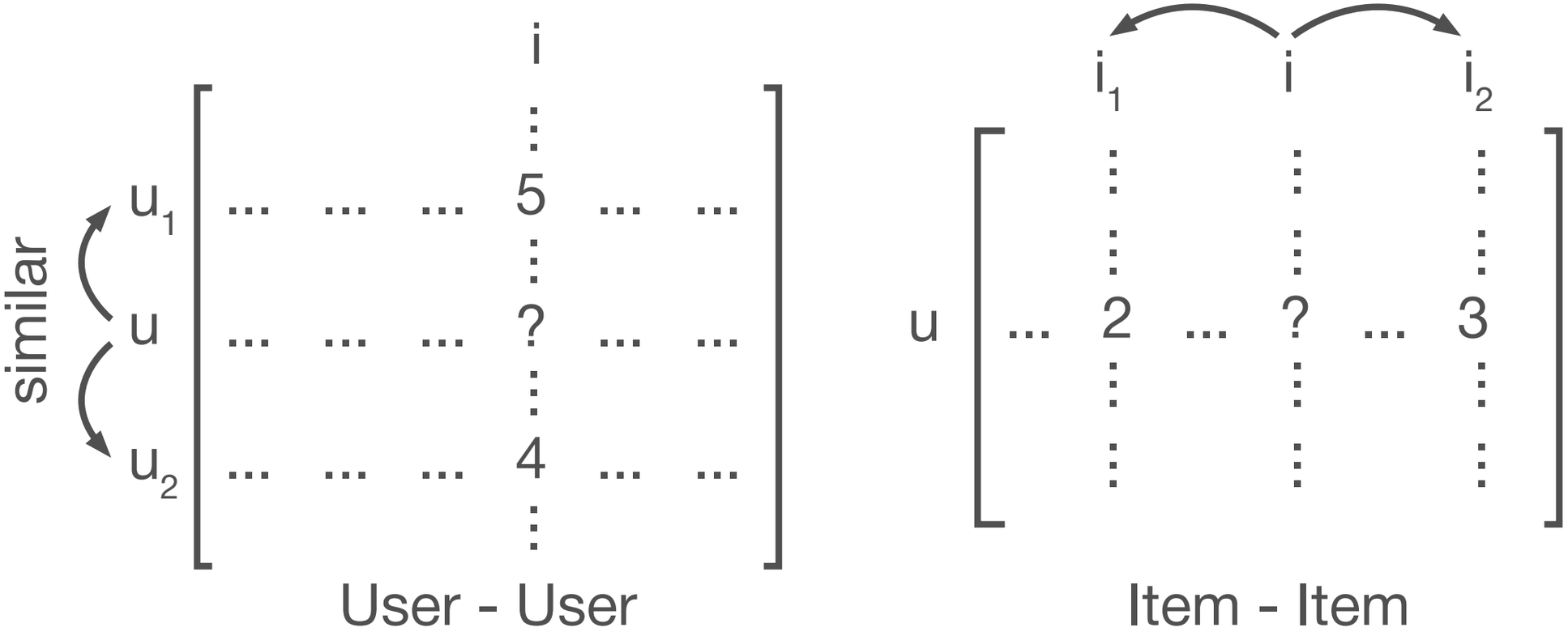
- **build** User-User (**UU**) and Item-Item (**II**) Collaborative Filtering Recommender System algorithms;
- **explain** the difference between **explicit** and **implicit** feedback;
- **optimize** learning with the help of **ALS** / **iALS**;
- **use** matrix factorization (**MF**) for recommendations;
- **explain** why SVD-like and **not** SVD.

From CF to MF RS: Outline

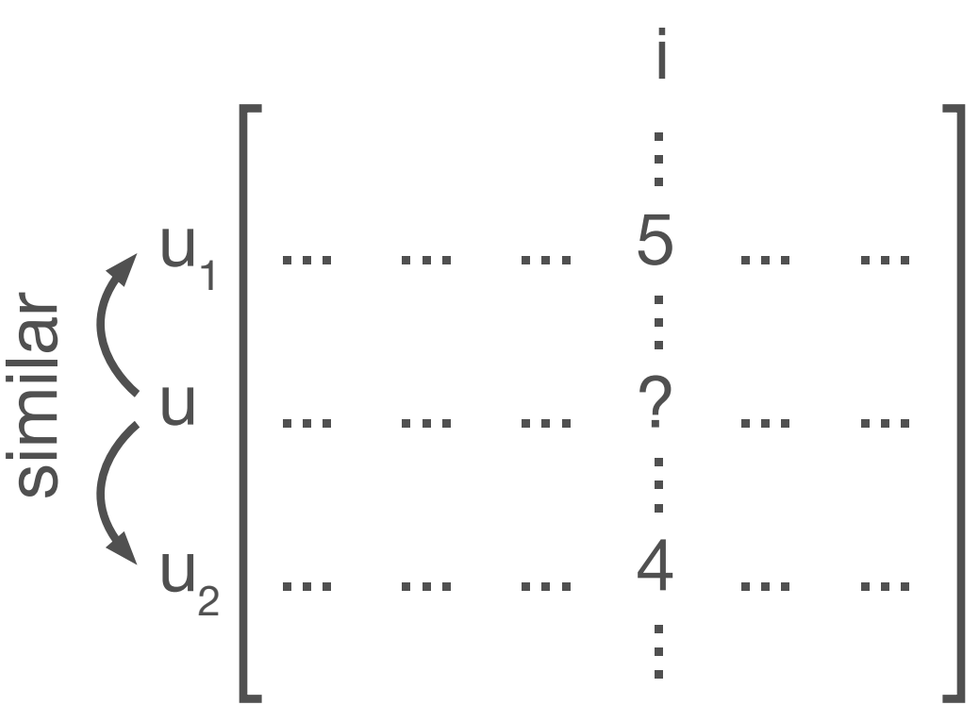
- User-User (UU) and Item-Item (II) CF
- Matrix Factorization (MF)
- Explicit and Implicit Feedback; ALS / iALS

User-User (UU) and Item-Item (II) CF

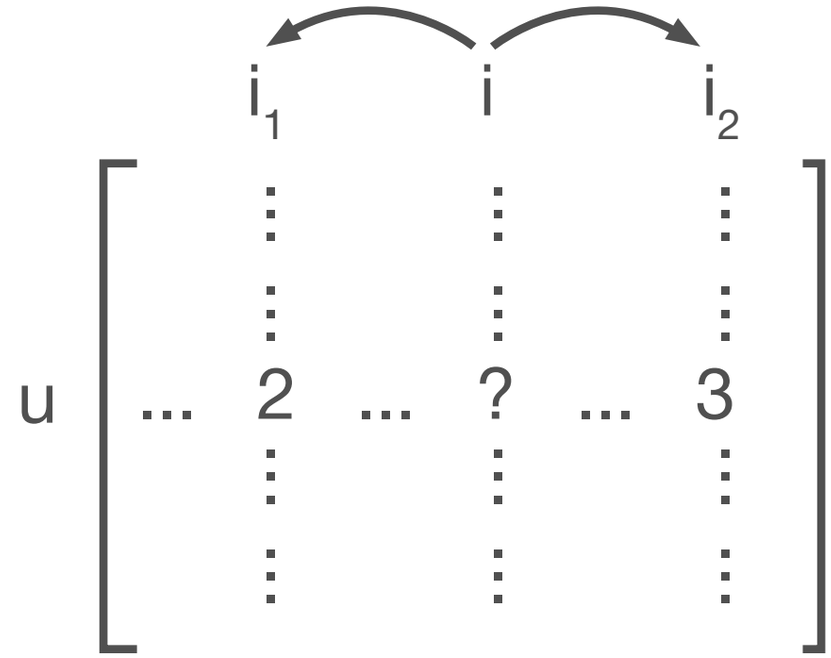
Neighbourhood Models



Neighbourhood Models



User - User



Item - Item

Main components:

1. Normalization
2. Similarity measure
3. Neighbourhood selection

Main components:

1. Normalization

2. Similarity measure

3. Neighbourhood selection




	...	2	...	2	...	5	...	5
	...	1	...	2	...	4	...	5

μ_1: 3.5

μ_2: 3

$r_i - \mu_i$

	...	-1.5	...	-1.5	...	1.5	...	1.5
	...	-2	...	-1	...	1	...	2





··· 2 ··· 2 ··· 5 ··· 5 μ_1: 3.5; $\sigma_1 \approx 1.7$


··· 1 ··· 2 ··· 4 ··· 5 μ_2: 3; $\sigma_2 \approx 1.8$

$$\text{Z-score} = \frac{r_i - \mu_i}{\sigma_i}$$


··· -0.88 ··· -0.88 ··· 0.88 ··· 0.88


··· -1.18 ··· -0.59 ··· 0.59 ··· 1.18





··· 2 ··· 2 ··· 5 ··· 5 μ_1: 3.5; $\sigma_1 \approx 1.7$


··· 1 ··· 2 ··· 4 ··· 5 μ_2: 3; $\sigma_2 \approx 1.8$

$p_i \times \sigma_i + \mu$ ⇕ $Z\text{-score} = \frac{r_i - \mu_i}{\sigma_i}$


··· -0.88 ··· -0.88 ··· 0.88 ··· 0.88


··· -1.18 ··· -0.59 ··· 0.59 ··· 1.18

Main components:

1. Normalization
2. **Similarity measure**
3. Neighbourhood selection

1. Metrics intended for real-valued vector spaces
2. Metrics intended for integer-valued vector spaces
3. Metrics intended for boolean-valued vector spaces

... User-defined distance

$$cos(u,v) = \frac{\sum_{i \in I_{uv}} r_{u_i} r_{v_i}}{\sqrt{\sum_{i \in I_u} r_{u_i}^2 \sum_{i \in I_v} r_{v_i}^2}}$$

I_u - items rated by u

I_{uv}- items rated by u&v

$$cos - centered(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u_i} - \overline{r}_u)(r_{v_i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_u} r_{u_i}^2 \sum_{i \in I_v} r_{v_i}^2}}$$

↳aka adjusted cosine

$$PearsonCorrelation(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u_i} - \overline{r}_u)(r_{v_i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u_i} - \overline{r}_u)^2 \sum_{i \in I_{uv}} (r_{v_i} - \overline{r}_v)^2}}$$

$$PearsonCorrelation(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u_i} - \overline{r}_u)(r_{v_i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u_i} - \overline{r}_u)^2 \sum_{i \in I_{uv}} (r_{v_i} - \overline{r}_v)^2}}$$

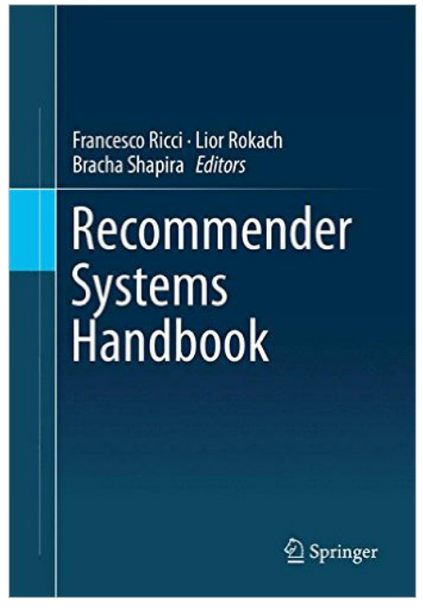
Spearman's Rank Correlation

$$SRC(u,v) = \frac{\sum_{i \in I_{uv}} (k_{u_i} - \overline{k}_u)(k_{v_i} - \overline{k}_v)}{\sqrt{\sum_{i \in I_{uv}} (k_{u_i} - \overline{k}_u)^2 \sum_{i \in I_{uv}} (k_{v_i} - \overline{k}_v)^2}}$$

More variations:

1. Use "support"
2. Use "Bayesian damps"
3. Use "shrinkage"

see: Recommender Systems Handbook, by Editors: Ricci, Francesco, Rokach, Lior, Shapira, Bracha (Eds.)

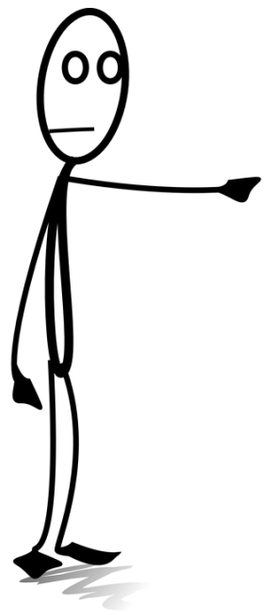


Main components:

1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

Strategies:

1. **All**

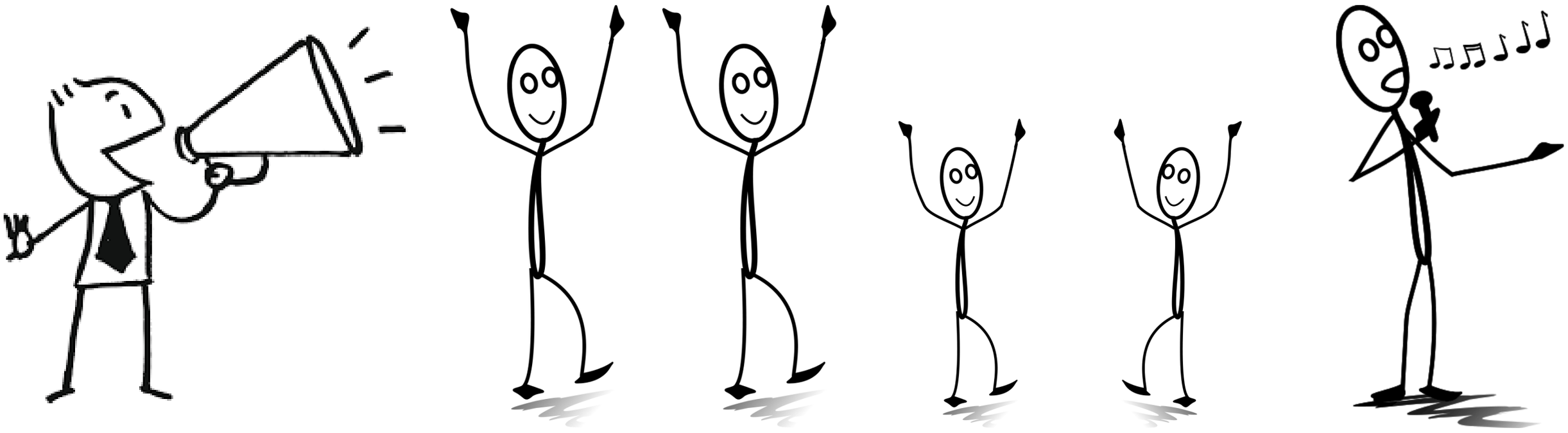


Main components:

1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

Strategies:

1. All
2. **Top-N**



top-6 (example)

Main components:

1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

Strategies:

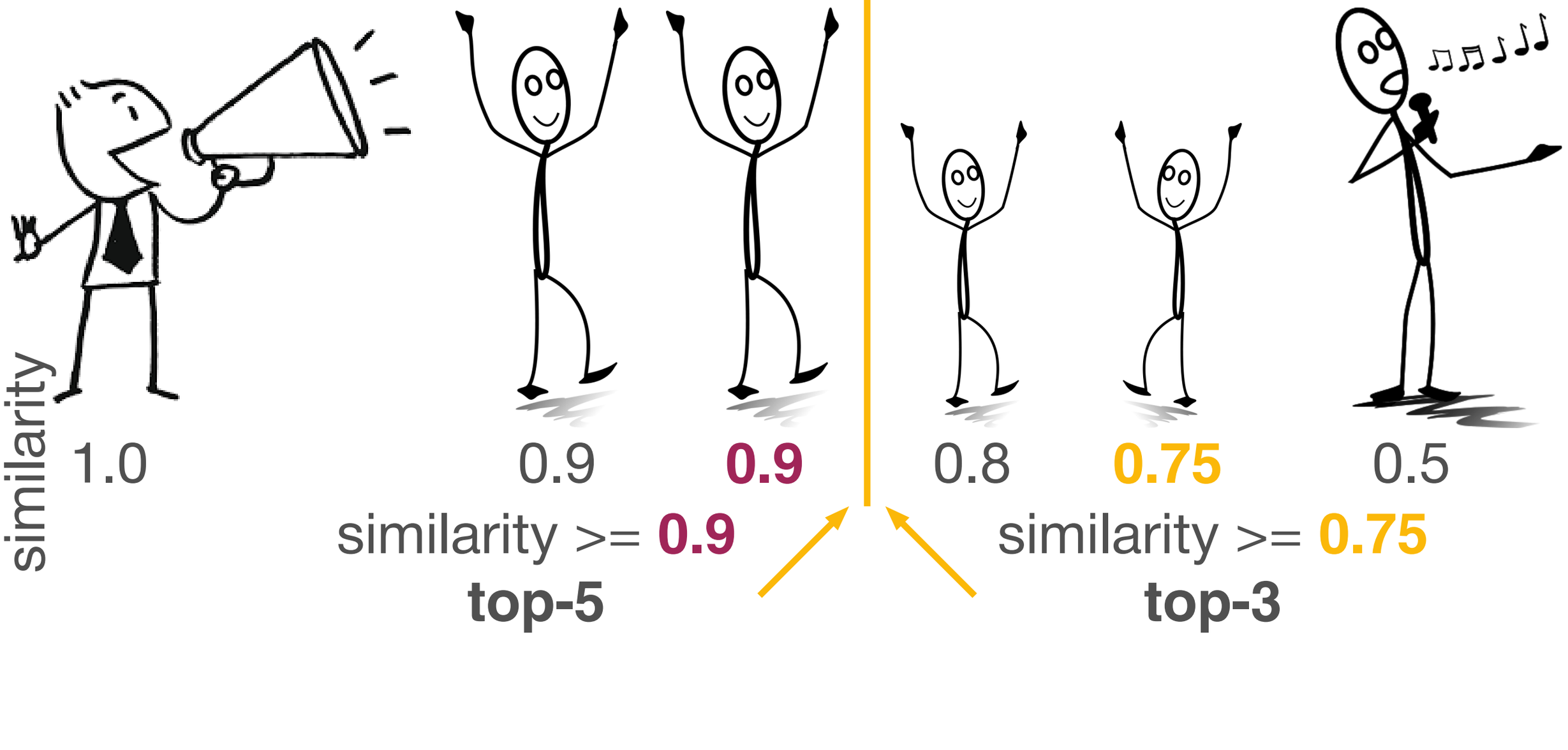
1. All
2. Top-N
3. **Similarity thresh**

Main components:

1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

Strategies:

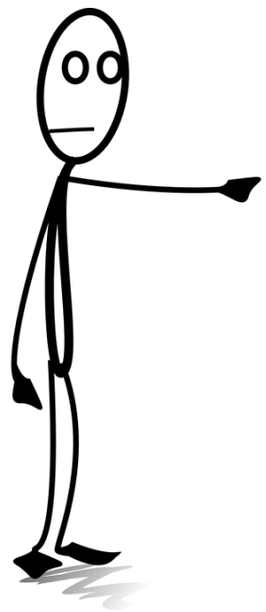
1. All
2. Top-N
3. **Similarity thresh**



Main components:

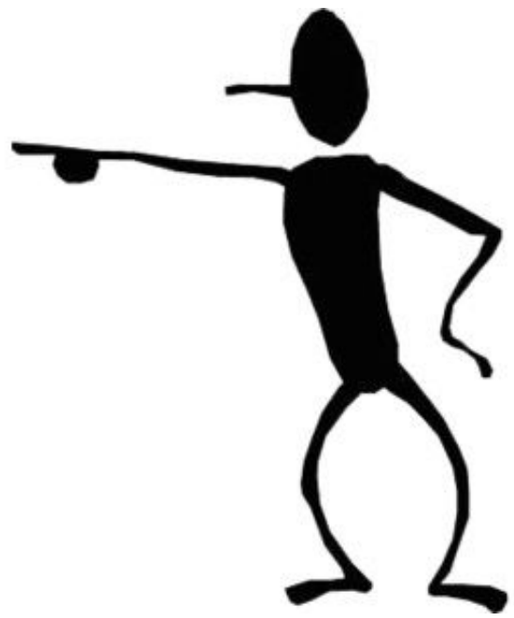
1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

similarity



1.0

???



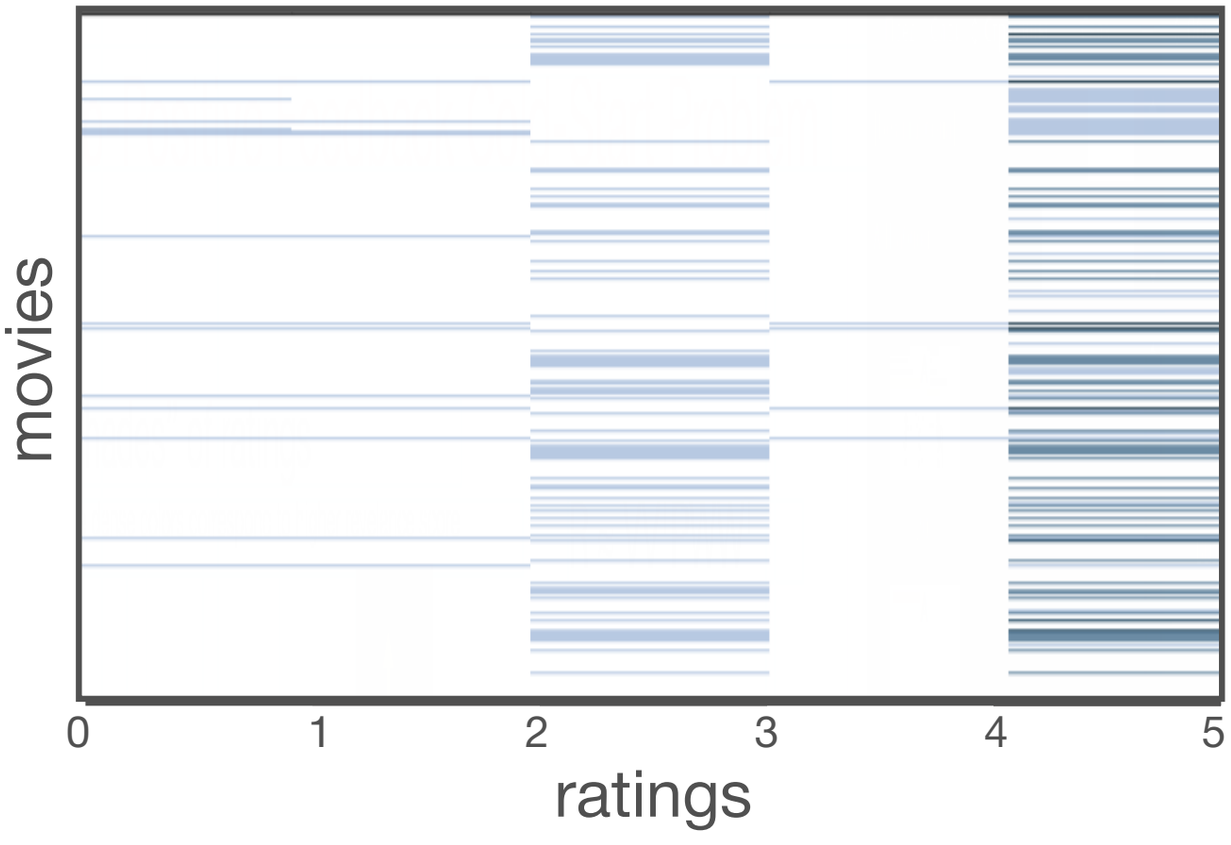
-0.5

negative correlation

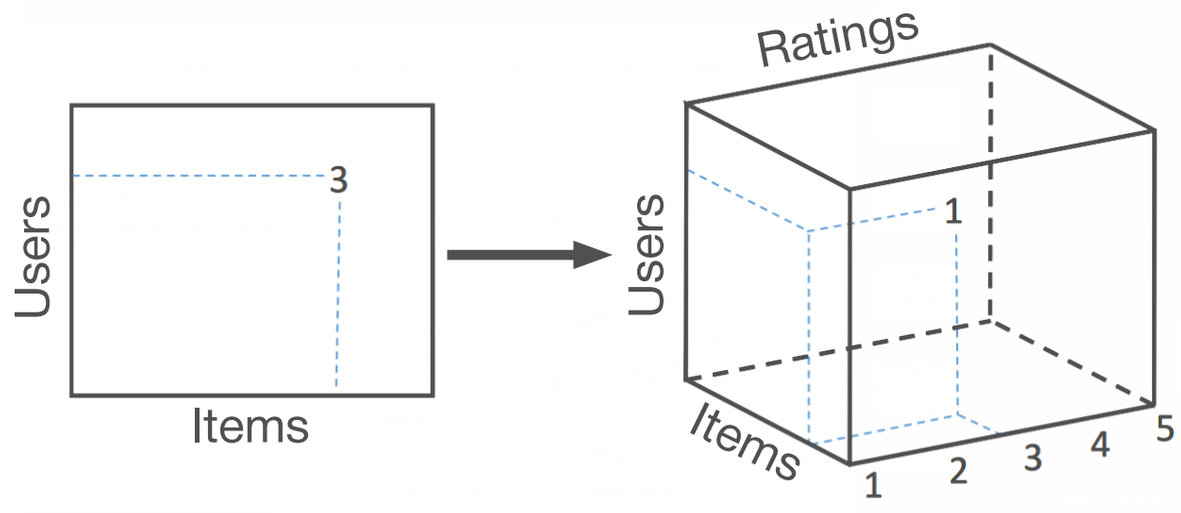
No-Positive Feedback Cold-Start Problem

“Shades” of ratings

More dense colors correspond to higher revelance score



$$R \approx VV^T PWW^T$$



From a matrix to a third order tensor

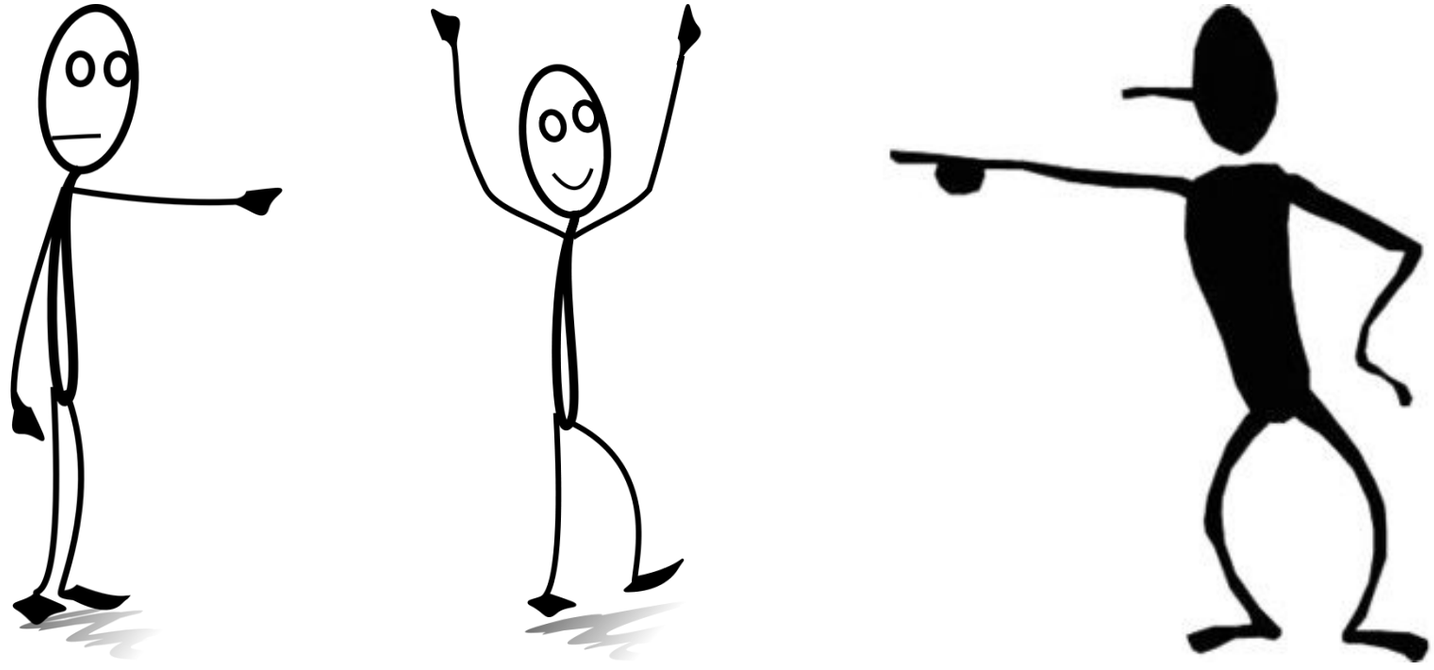
see: http://dl.acm.org/citation.cfm?doid=2959100.2959170

Main components:

1. Normalization
2. Similarity measure
3. **Neighbourhood selection**

Strategies:

1. All
2. Top-N
3. Similarity thresh
4. **Random**



Prediction (User-User CF)

$$p_{u_i} = \frac{\sum_{v \in u_i \cap N_u} W_{uv} V_{v_i}}{\sum_{v \in u_i \cap N_u} W_{uv}}$$

$\mathrm{u}_i = users\ who\ rated\ item\ i$

$\mathrm{W}_{uv} = sim(u, v)$

$\mathrm{N}_u = neighbourhood\ of\ user\ u$

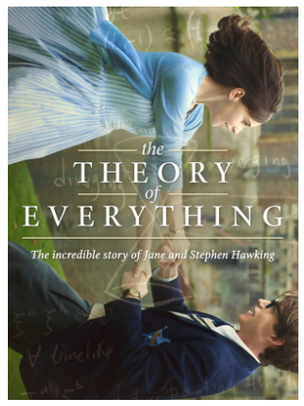
Prediction (Item-Item CF)

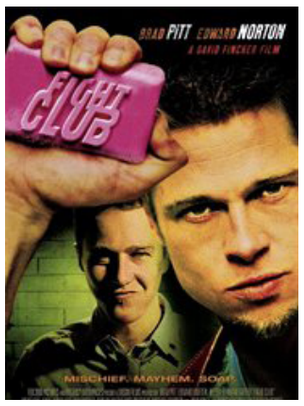
$$p_{u_i} = \frac{\sum_{j \in N_i \cap I_u} W_{ij} V_{u_j}}{\sum_{j \in W_i \cap I_u} W_{ij}}$$

Prediction (Item-Item CF)

you might like item

because you have
provided high positive score for





★★★★★ ★★★★★ ★★★★★

CF Efficiency

CF Efficiency

$n = |U|$

$m = |I|$

	Space
UU	$O(n^2)$
II	$O(m^2)$

CF Efficiency

$n = |U|$

$m = |I|$

	Space		Prediction
UU	$O(n^2)$		$O(k)$
II	$O(m^2)$		$O(k)$

$k = $ #neighbours

CF Efficiency

$n = |U|$

$m = |I|$

$p = \max_u |I_u|$

$q = \max_i |U_i|$

$k = $ #neighbours

	Space	Preprocessing	Prediction
UU	$O(n^2)$	$O(n^2 p)$	$O(k)$
II	$O(m^2)$	$O(m^2 q)$	$O(k)$

CF Efficiency

$n = |U|$

$m = |I|$

$p = \max_u |I_u|$

$q = \max_i |U_i|$

$k =$ #neighbours

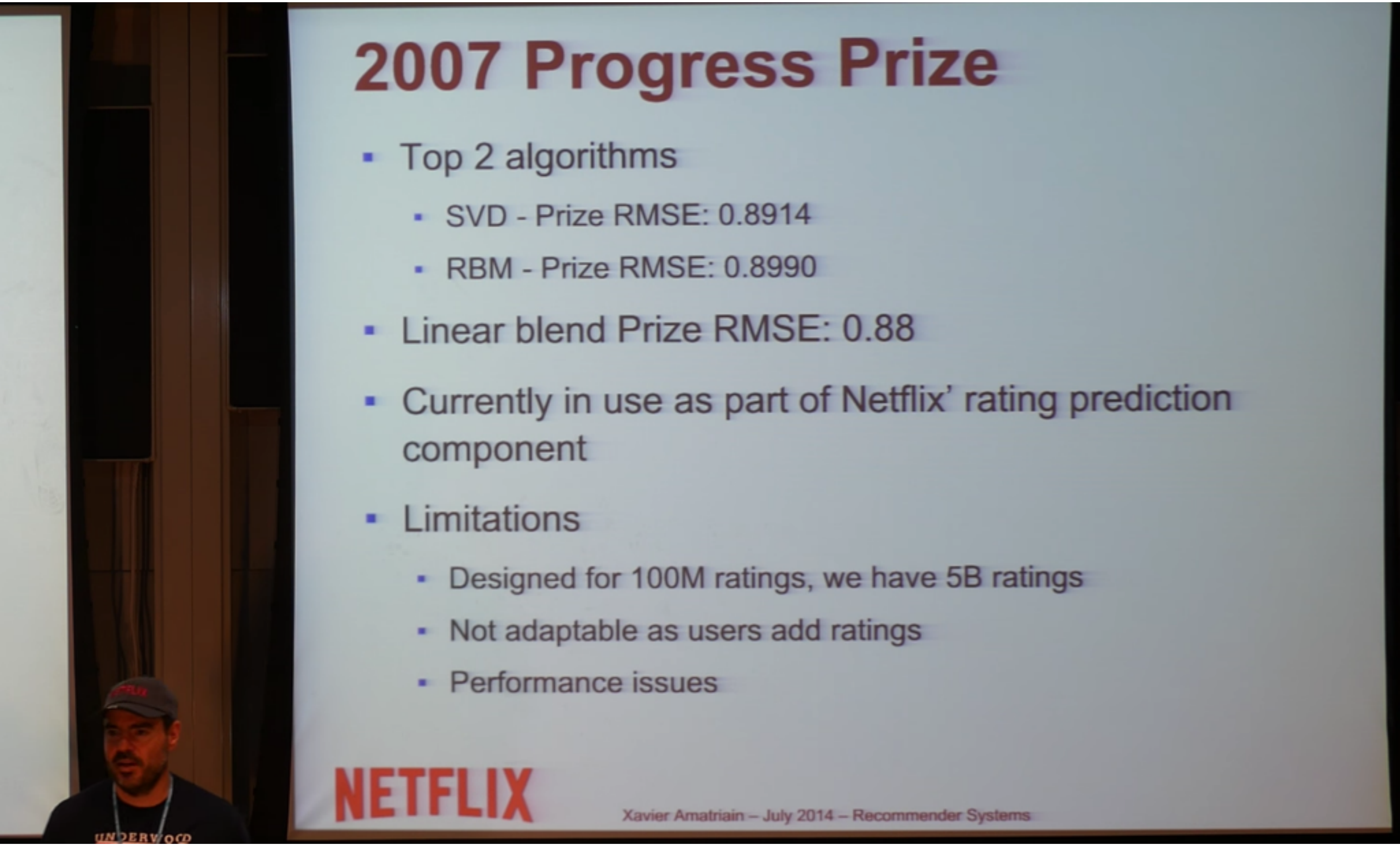
$R =$ total #ratings

$$s = \frac{R}{n} \; t = \frac{R}{m}$$

	Space	Preprocessing	Prediction
UU	$O(n^2)$	$O(n^2 p)$	$O(k)$
II	$O(m^2)$	$O(m^2 q)$	$O(k)$

	Avg. #neighbours	Avg. #ratings
UU	$(n-1)\left(1-\left(\frac{m-s}{m}\right)^s\right)$	$\frac{s^2}{m}$
II	$(m-1)\left(1-\left(\frac{n-t}{n}\right)^t\right)$	$\frac{t^2}{n}$

Matrix Factorization



credit to: Xavier Amatrian

Recap: Item-based or user-based

Key advantages:

- easy to implement
- intuitive explanations
- good baseline

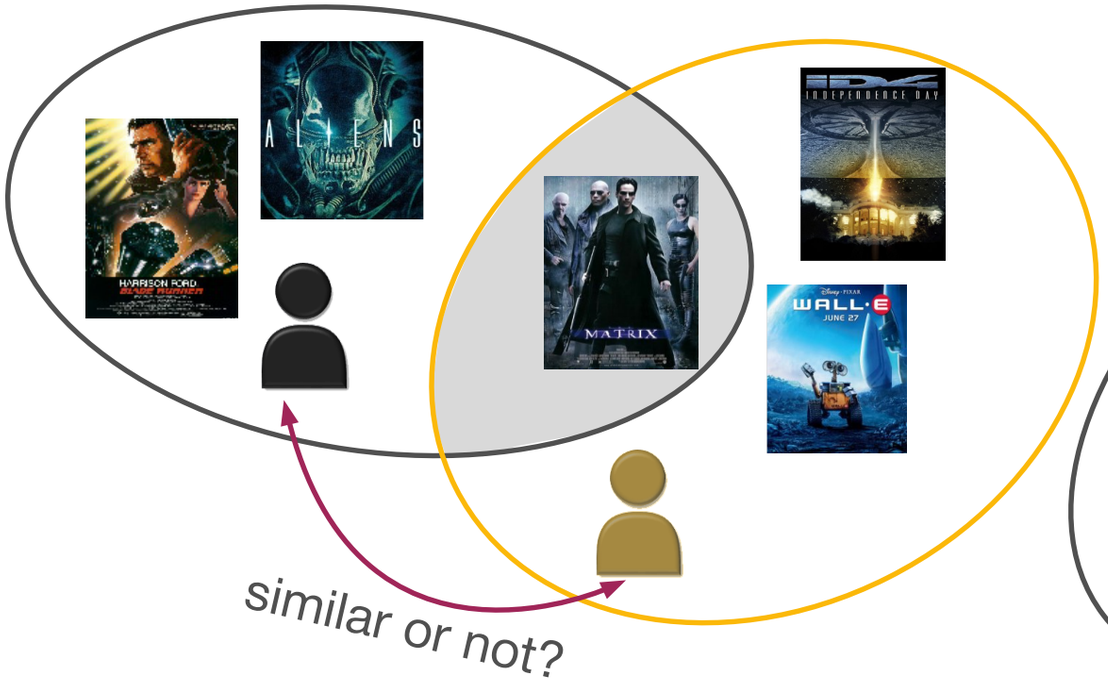
Scalability:

$O(n^2)$ or $O(m^2)$ complexity in the worst case

- due to sparsity real complexity is close to linear
- could store only limited number of neighbors
- make incremental updates

Limited coverage problems

Unreliable correlations



Weak generalization

never recommended together

Dimensionality reduction

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)

Dimensionality reduction

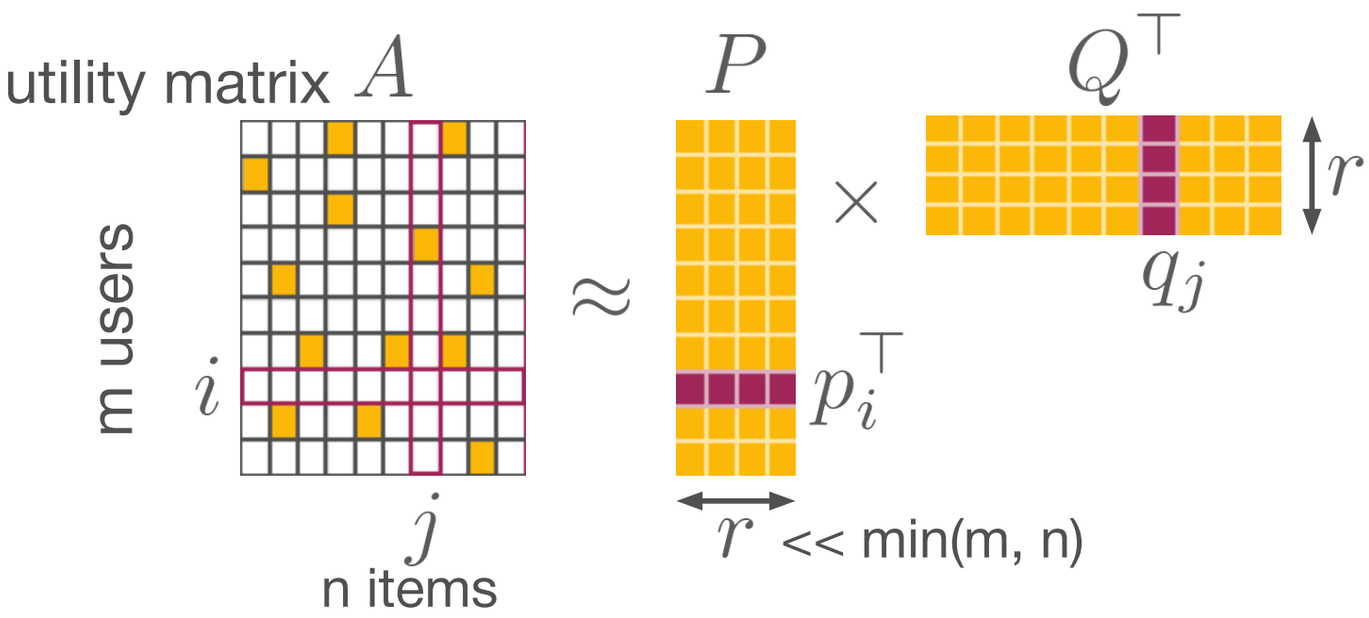
Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)

Dimensionality reduction

Martix factorization

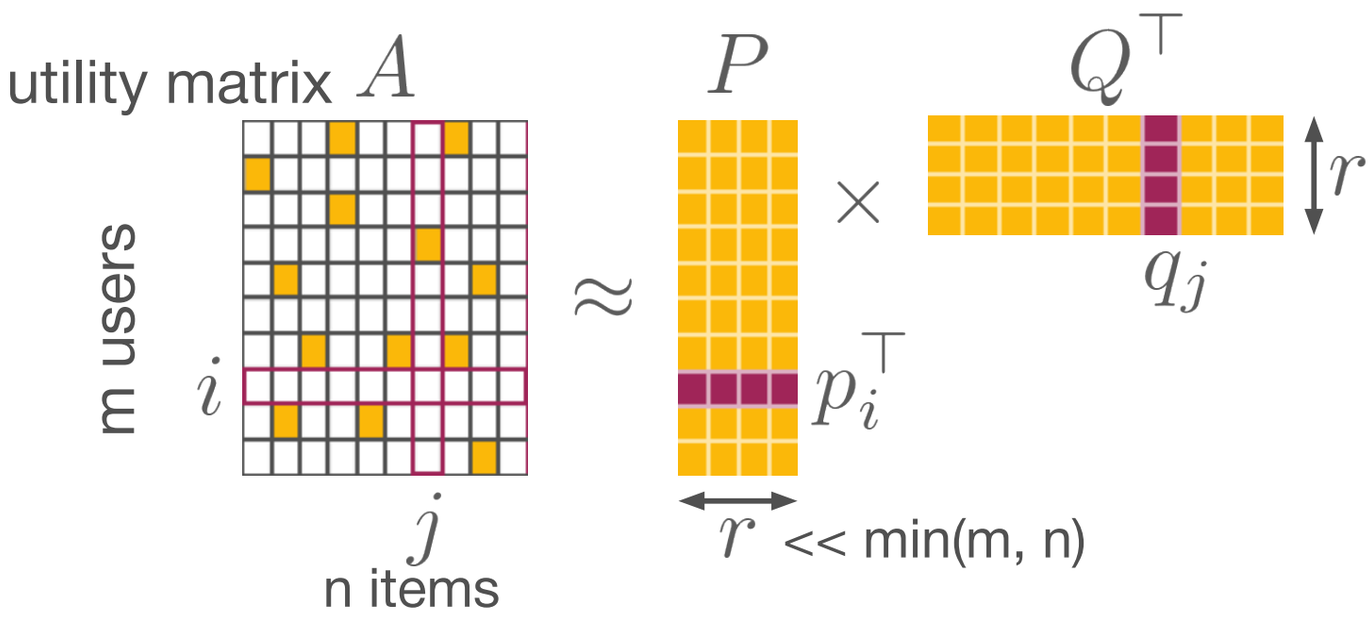
- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)



Dimensionality reduction

Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)



predicted utility of item j for user i

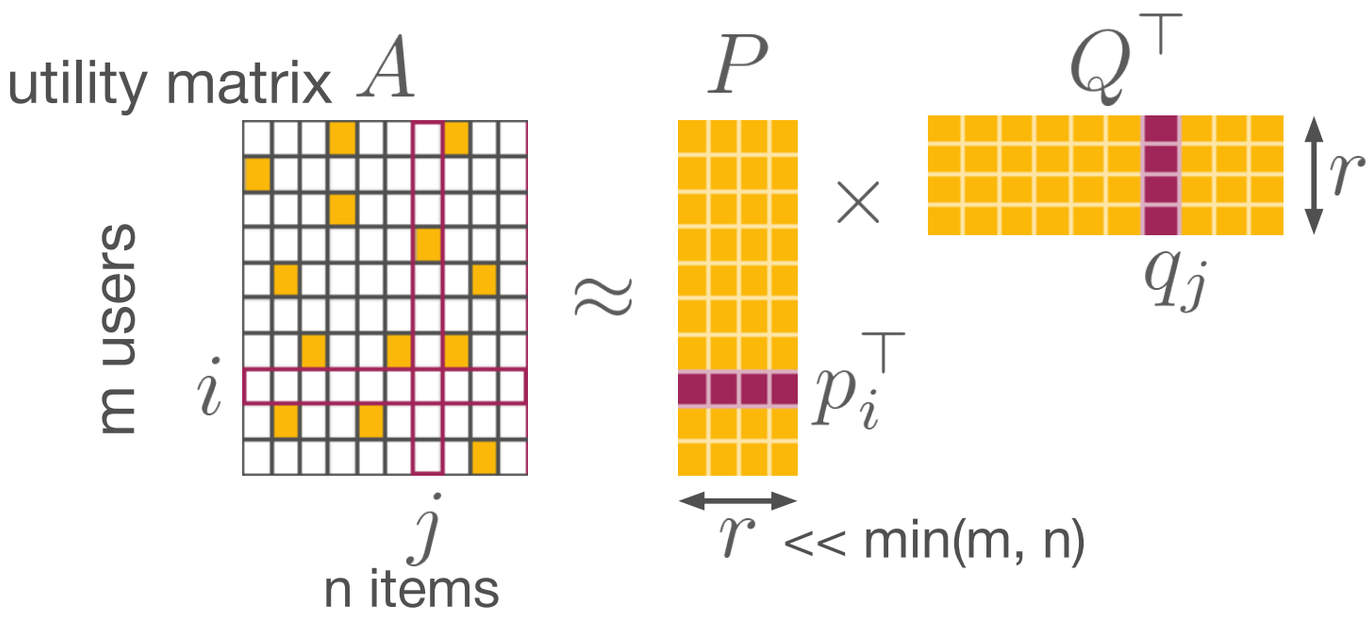
$$a_{ij} \approx p_i^\top q_j = \sum_{k=1}^{r} p_{i_k} q_{j_k}$$

Mind: Index Notation

Dimensionality reduction

Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)



predicted utility of item j for user i

$$a_{ij} \approx p_i^\top q_j = \sum_{k=1}^{r} p_{i_k} q_{j_k}$$

top-n recommendations task:

$$toprec(i, n) := \arg\max_{j}^{n} a_{ij}$$

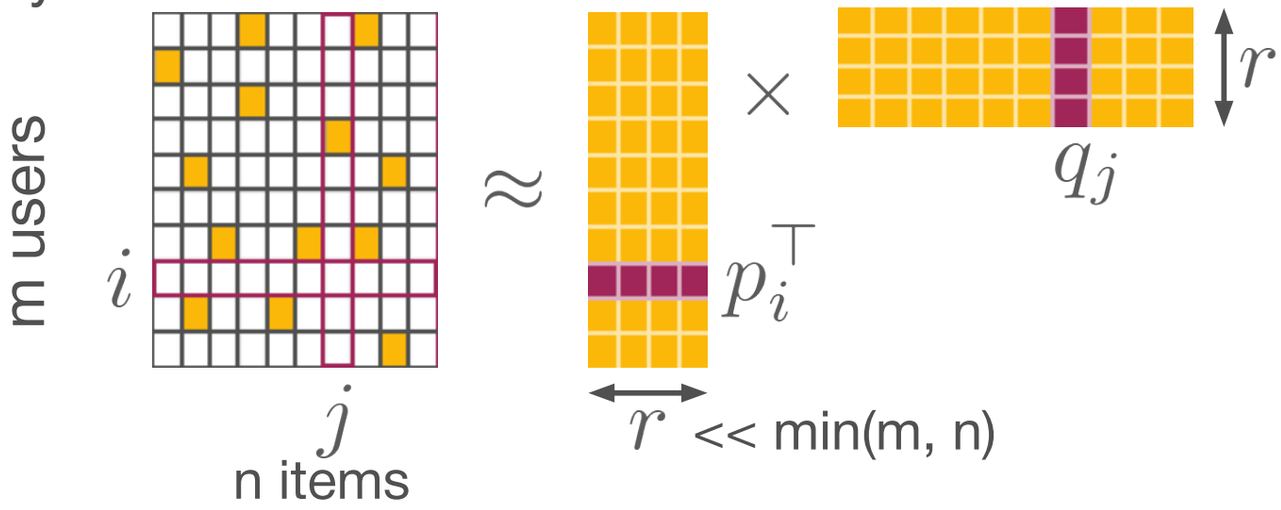
Mind: Index Notation

Dimensionality reduction

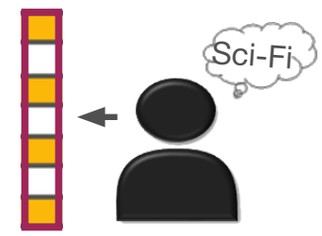
Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)

utility matrix A



Simplistic view: latent features $\leftrightarrow$ genres



predicted utility of item j for user i

$$a_{ij} \approx p_i^\top q_j = \sum_{k=1}^{r} p_{i_k} q_{j_k}$$

top-n recommendations task:

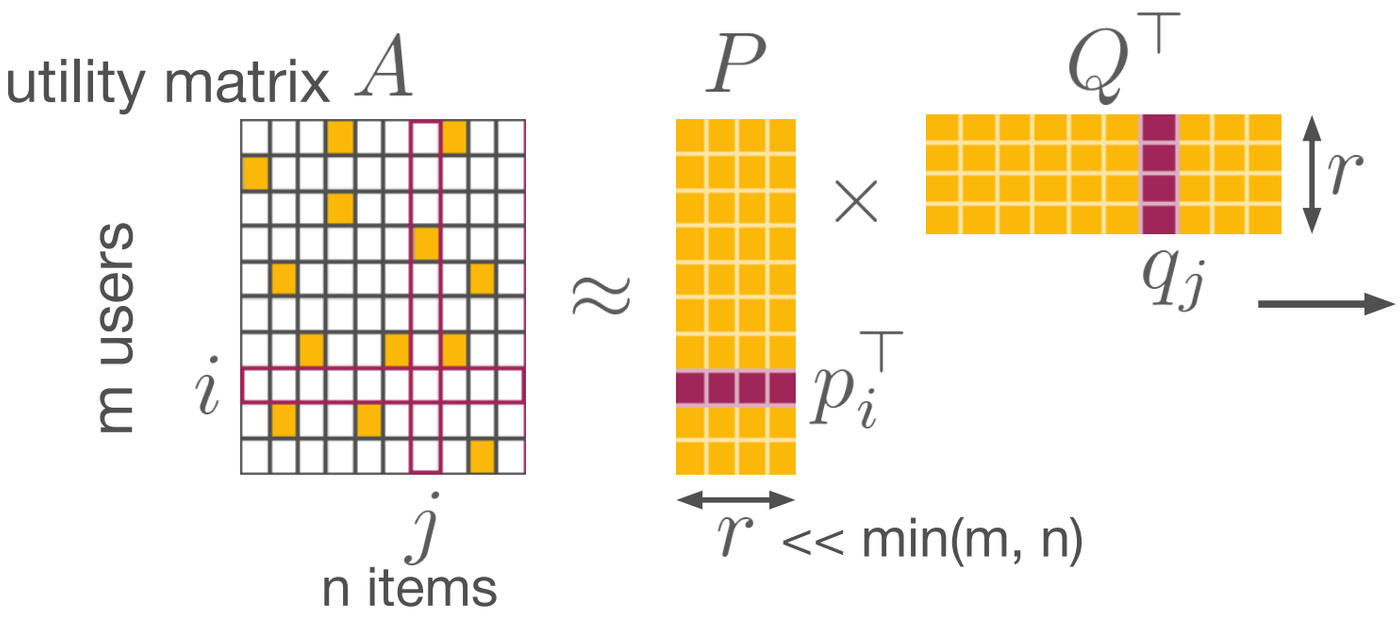
$$toprec(i, n) := \arg\max_{j}^{n} a_{ij}$$

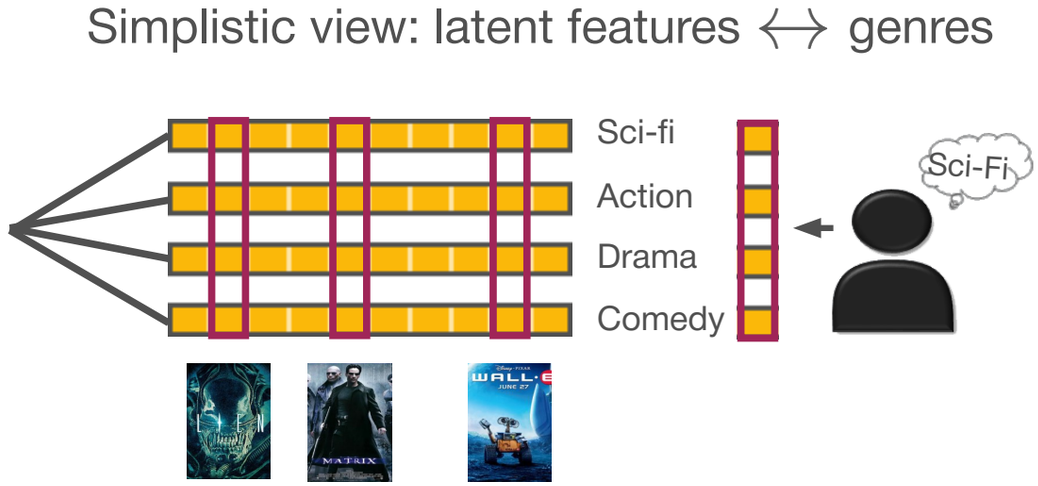
Mind: Index Notation

Dimensionality reduction

Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)





predicted utility of item j for user i

$$a_{ij} \approx p_i^\top q_j = \sum_{k=1}^{r} p_{i_k} q_{j_k}$$

top-n recommendations task:

$$toprec(i, n) := \arg\max_{j}^{n} a_{ij}$$

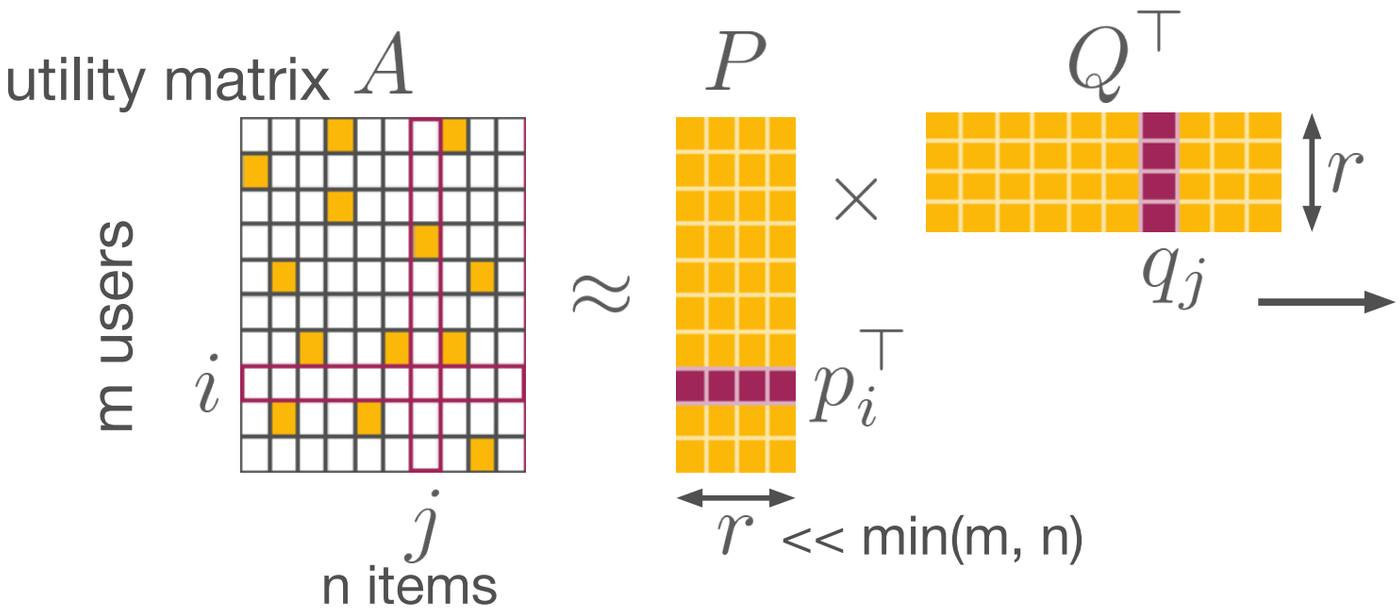
Mind: Index Notation

Dimensionality reduction

Martix factorization

- gives a small set of parameters to describe users and items
- find common behavioral patterns (hidden structure of data)

utility matrix A



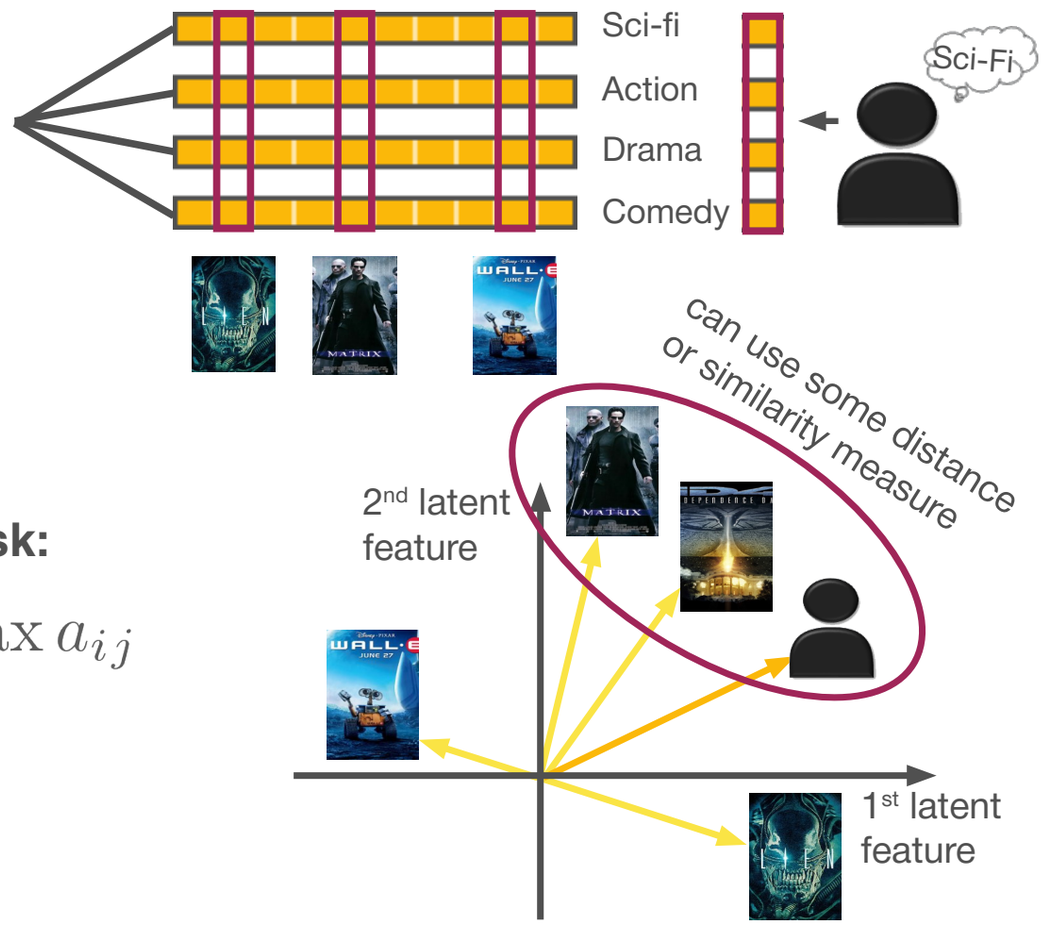
predicted utility of item j for user i

$$a_{ij} \approx p_i^\top q_j = \sum_{k=1}^{r} p_{i_k} q_{j_k}$$

top-n recommendations task:

$$toprec(i, n) := \arg\max_{j}^{n} a_{ij}$$

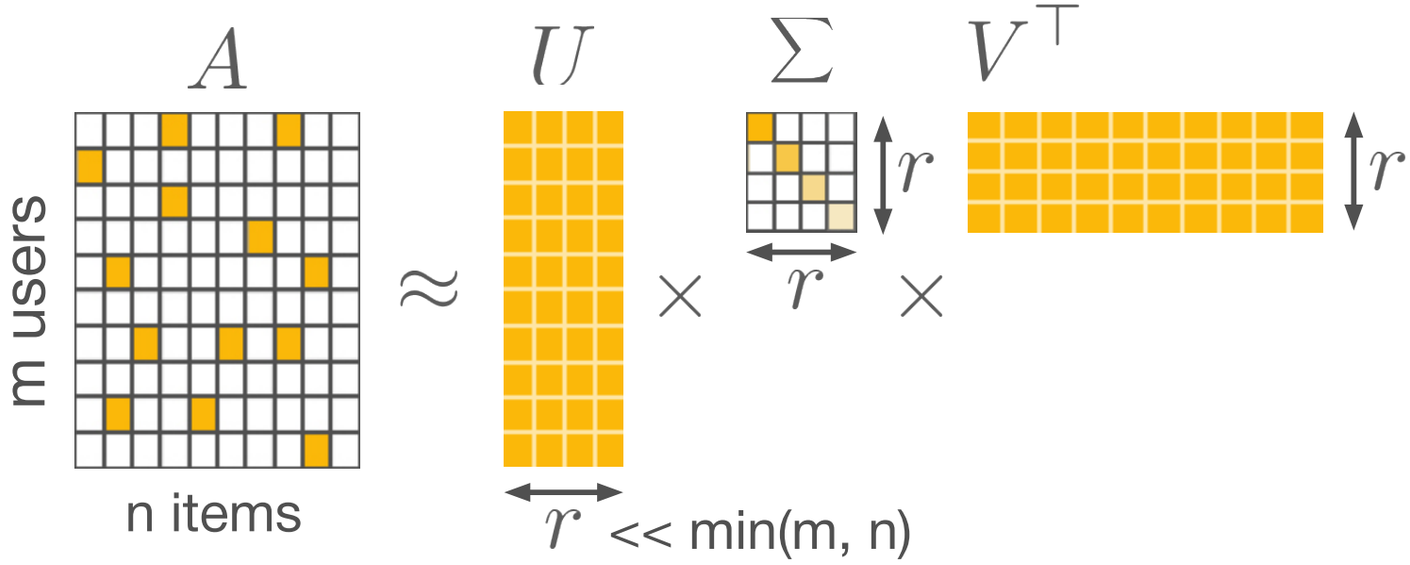
Simplistic view: latent features ↔ genres



Singular Value Decomposition

Used in LSA/LSI, PCA…

Truncated SVD of rank r



columns are orthonormal:

$$U^\top U = V^\top V = I$$

$$||A - A_r||_F^2 \to \min$$

$$A_r = U\Sigma V^\top$$

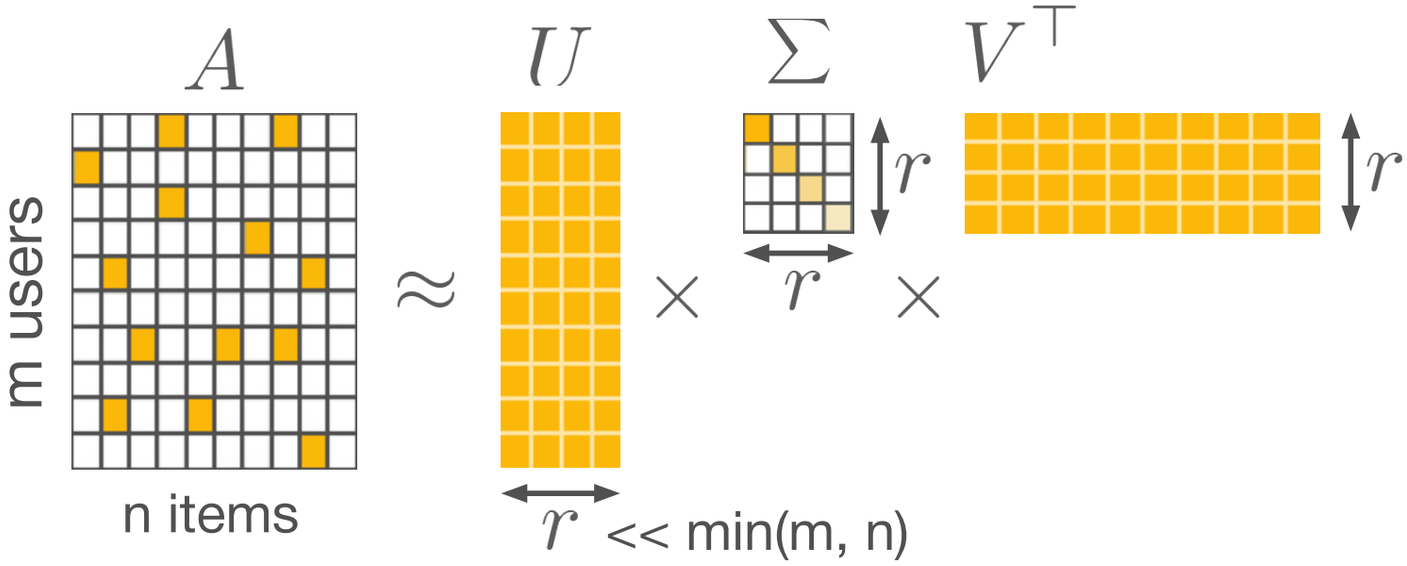
$$||X||_F^2 = \sum_{ij} x_{ij}^2$$

Mind: Index Notation

Singular Value Decomposition

Used in LSA/LSI, PCA…

Truncated SVD of rank r



columns are orthonormal:

$$U^\top U = V^\top V = I$$

$$||A - A_r||_F^2 \to \min$$

$$A_r = U\Sigma V^\top$$

$$||X||_F^2 = \sum_{ij} x_{ij}^2$$

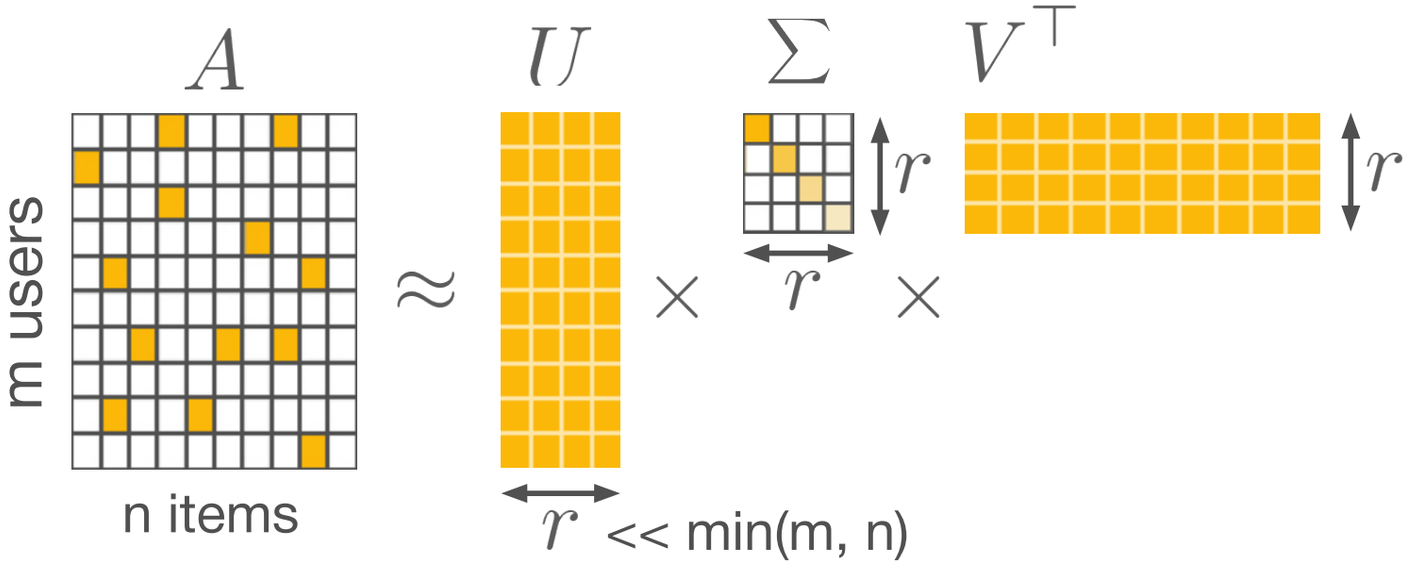
Undefined for incomplete matrix!

Mind: Index Notation

Singular Value Decomposition

Used in LSA/LSI, PCA…

Truncated SVD of rank r



columns are orthonormal:

$$U^\top U = V^\top V = I$$

$$||A - A_r||_F^2 \to \min$$

$$A_r = U\Sigma V^\top$$

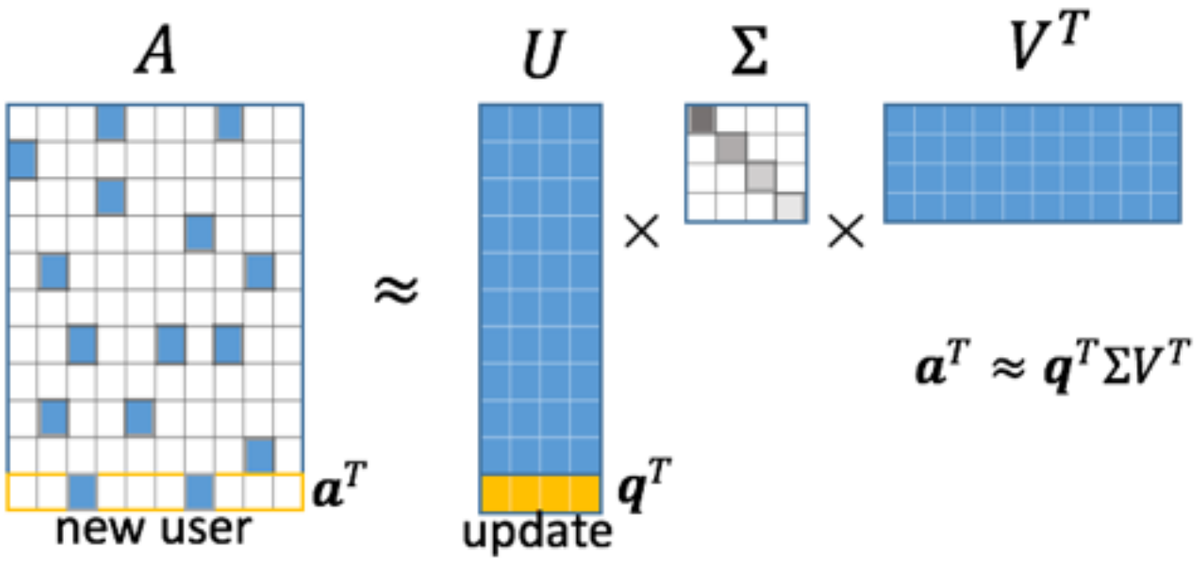
$$||X||_F^2 = \sum_{ij} x_{ij}^2$$

Undefined for incomplete matrix!

Let's impute zeros - PureSVD model.

- values are highly biased towards 0
- not good for rating prediction
- its not a big problem for ranking task

PureSVD – recommending online



$V^T V = I$ orthogonality of columns

$a^T \approx q^T \Sigma V^T \rightarrow a^T V \approx q^T \Sigma V^T V = q^T \Sigma$

*folding-in technique**

$$q^T \approx a^T V \Sigma^{-1}$$

new user embedding

$$r^T = q^T \Sigma V^T \approx a^T V \Sigma^{-1} \Sigma V^T = a^T V V^T$$

allows for real-time recommendations

$O(nr)$ complexity

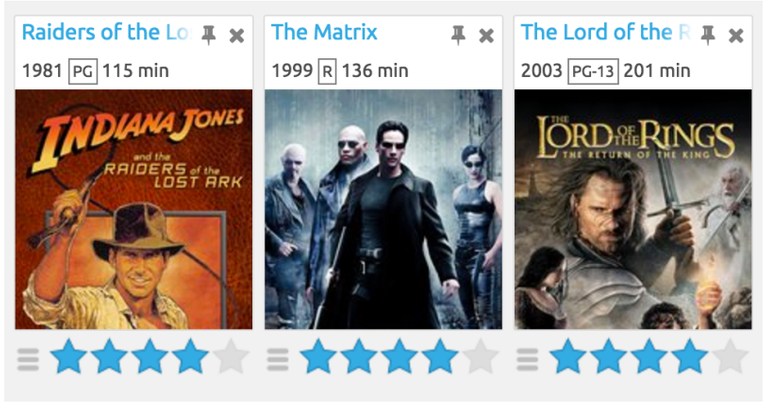
vector of predicted item scores

$$r \approx V V^T a$$

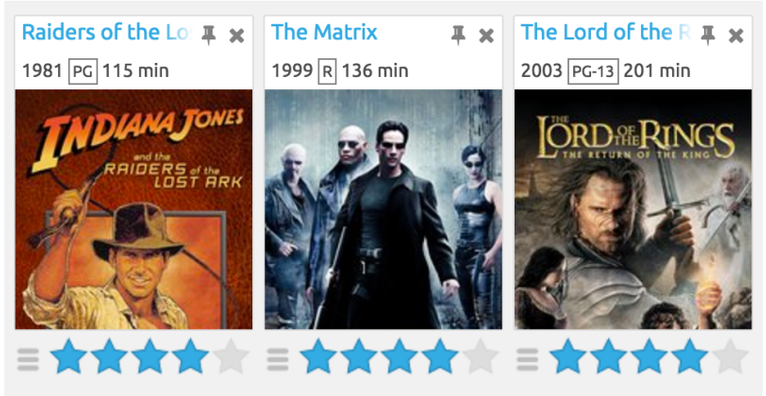
Mind: Index Notation

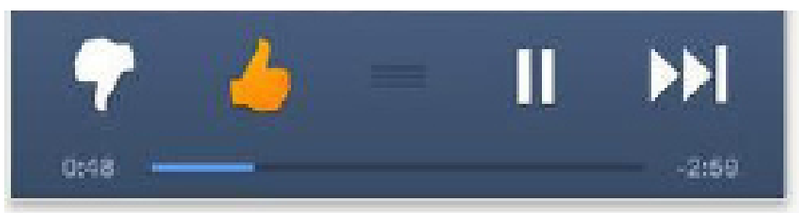
Explicit and Implicit Feedback

Explicit vs Implicit

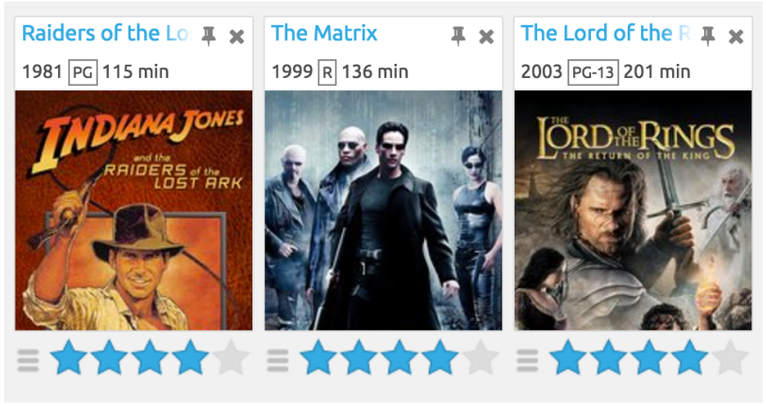


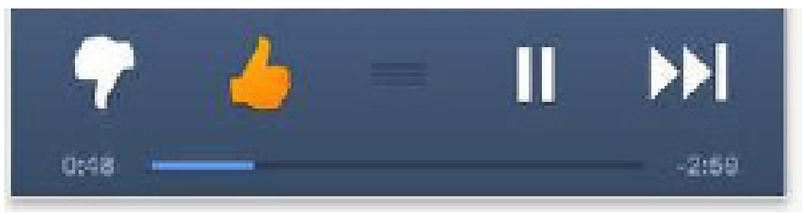
Explicit vs Implicit

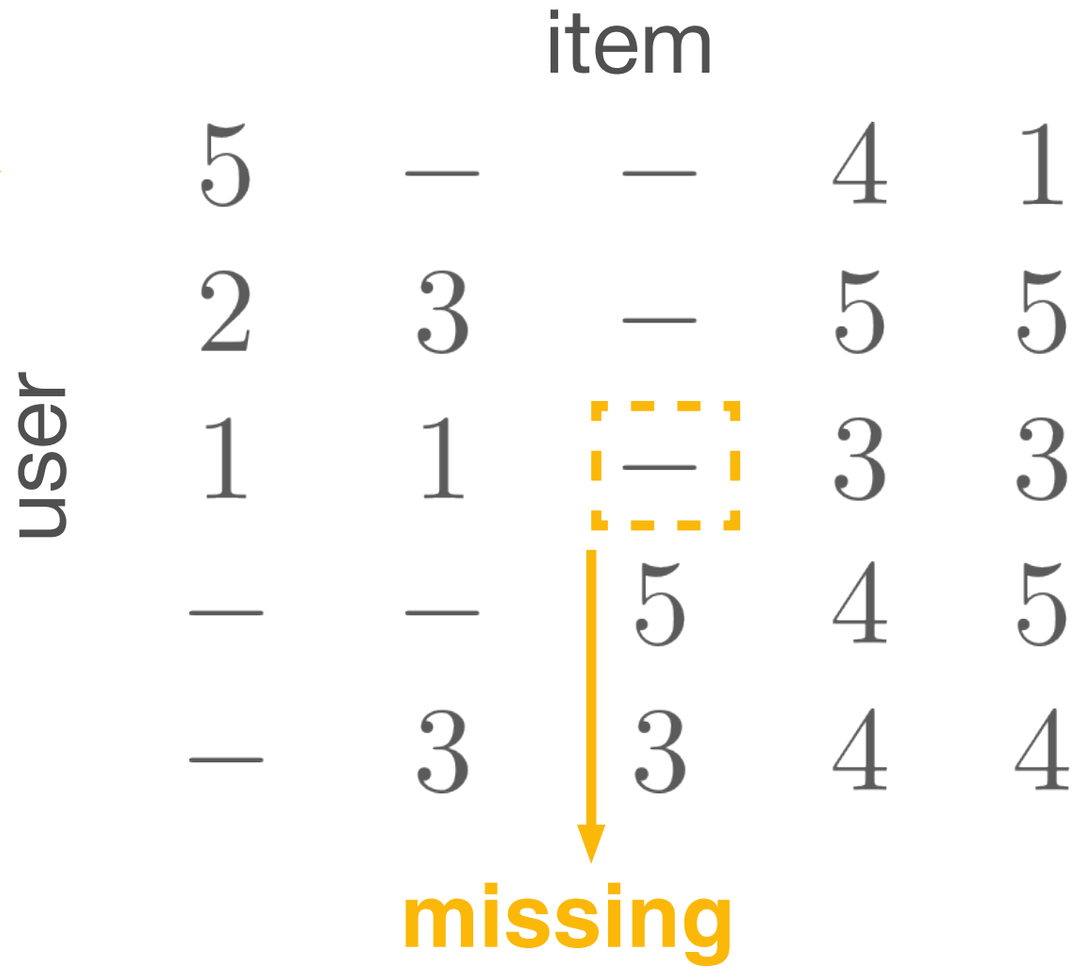




Explicit vs Implicit







≈ 55% (or 0.55)

≈ 55% (or 0.55)

10%

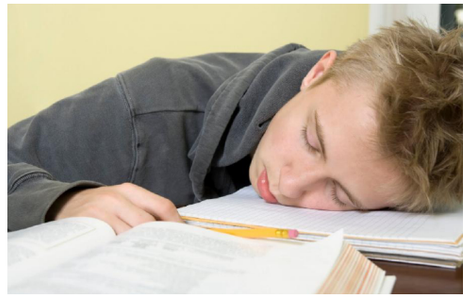
90%

Hu Yifan, Yehuda Koren, and Chris Volinsky.
“Collaborative filtering for implicit feedback datasets”

1. No true positive / true negative feedback

Hu Yifan, Yehuda Koren, and Chris Volinsky.
"Collaborative filtering for implicit feedback datasets"

1. No true positive / true negative feedback
2. Implicit feedback is noisy

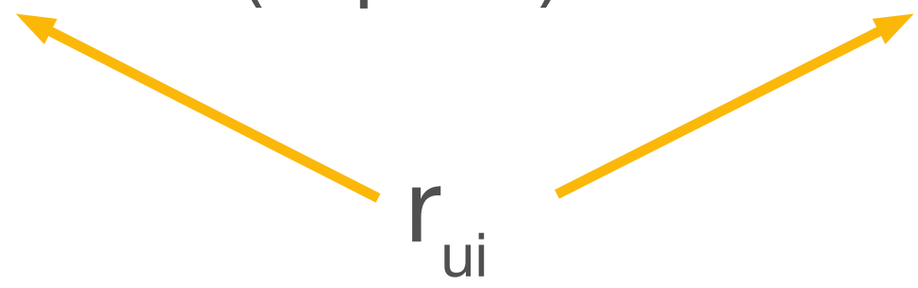


Hu Yifan, Yehuda Koren, and Chris Volinsky.
"Collaborative filtering for implicit feedback datasets"

1. No true positive / true negative feedback
2. Implicit feedback is noisy
3. Evaluation of implicit feedback RecSys must have its own metrics

Hu Yifan, Yehuda Koren, and Chris Volinsky.
"Collaborative filtering for implicit feedback datasets"

1. No true positive / true negative feedback
2. Implicit feedback is noisy
3. Evaluation of implicit feedback RecSys must have its own metrics
4. Preference (explicit) vs confidence (implicit)



ALS / iALS

Alternating Least Squares (ICDM)

Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights

Authors: Robert M. Bell
Yehuda Koren

2007 Article

Published in:
· Proceeding
ICDM '07 Proceedings of the 2007 Seventh IEEE International Conference on Data Mining
Pages 43-52

October 28 - 31, 2007
IEEE Computer Society Washington, DC, USA ©2007
table of contents ISBN:0-7695-3018-4 doi>10.1109/ICDM.2007.90

Bibliometrics

· Citation Count: 100
· Downloads (cumulative): 0
· Downloads (12 Months): 0
· Downloads (6 Weeks): 0

Explicit vs Implicit Feedback

(ALS)
Alternating Least Squares

iALS
(i = implicit)

Collaborative Filtering for Implicit Feedback Datasets

Authors: Yifan Hu
Yehuda Koren
Chris Volinsky

2008 Article

Published in:
· Proceeding
ICDM '08 Proceedings of the 2008 Eighth IEEE International Conference on Data Mining
Pages 263-272

December 15 - 19, 2008
IEEE Computer Society Washington, DC, USA ©2008
table of contents ISBN: 978-0-7695-3502-9 doi>10.1109/ICDM.2008.22

Bibliometrics
· Citation Count: 331
· Downloads (cumulative): 0
· Downloads (12 Months): 0
· Downloads (6 Weeks): 0

$$r_{u_i} = \frac{watched(movie)}{length(movie)}$$ - observations

item

user				
	0.5	0.9	0.2	0.7
	0.6	0.1	0.3	2.0
	0.7	0.2	0.1	1.0
	0.1	0.5	3.0	0.9

$$r_{u_i} = \frac{watched(movie)}{length(movie)}$$ - observations

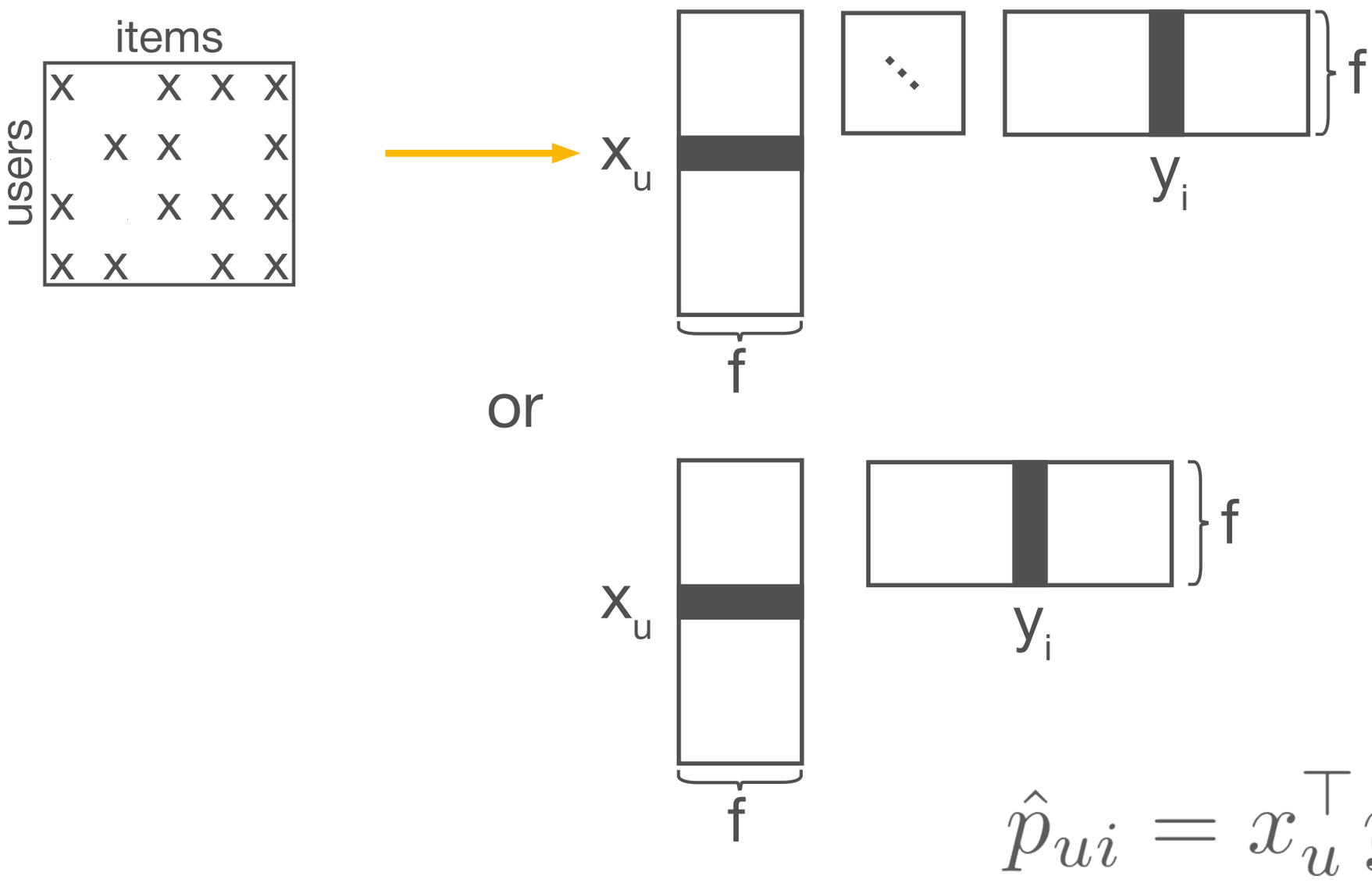
$$p_{u_i} = \begin{cases} 0, & r_{u_i} = 0 \\ 1, & r_{u_i} > 0 \end{cases}$$ - preference

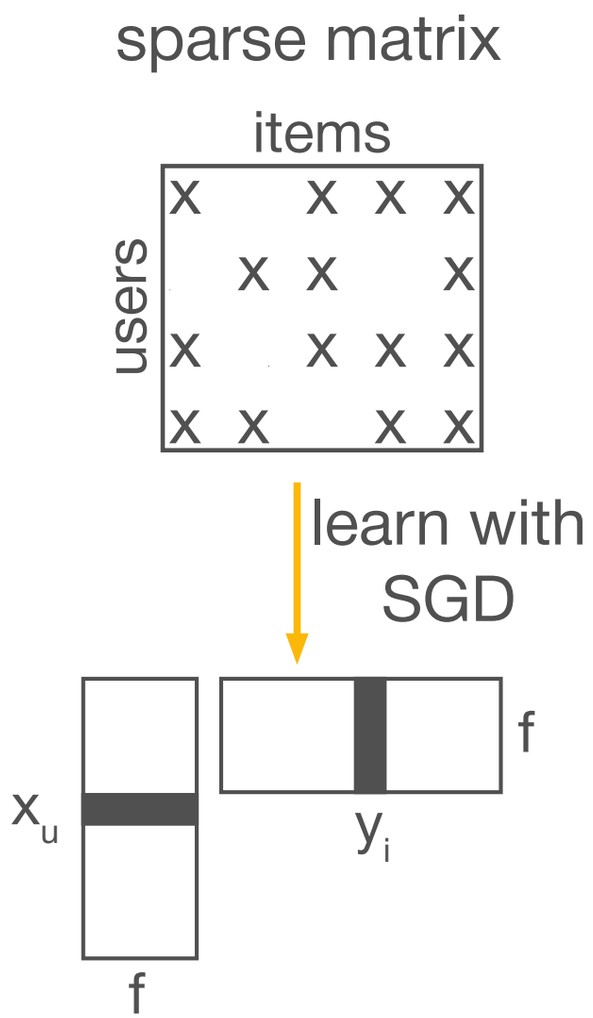
$$\begin{aligned} c_{u_i} &= 1 + \alpha r_{u_i}; \\ c_{u_i} &= 1 + \alpha log(1 + \frac{r_{u_i}}{\varepsilon}) \end{aligned}$$ - confidence

item

user				
	0.5	0.9	0.2	0.7
	0.6	0.1	0.3	2.0
	0.7	0.2	0.1	1.0
	0.1	0.5	3.0	0.9

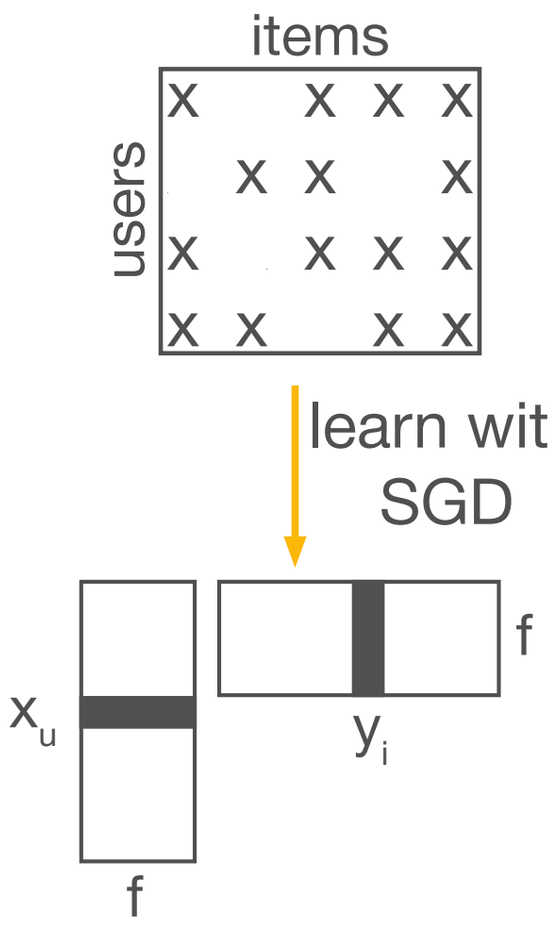
SVD-like / matrix decomposition





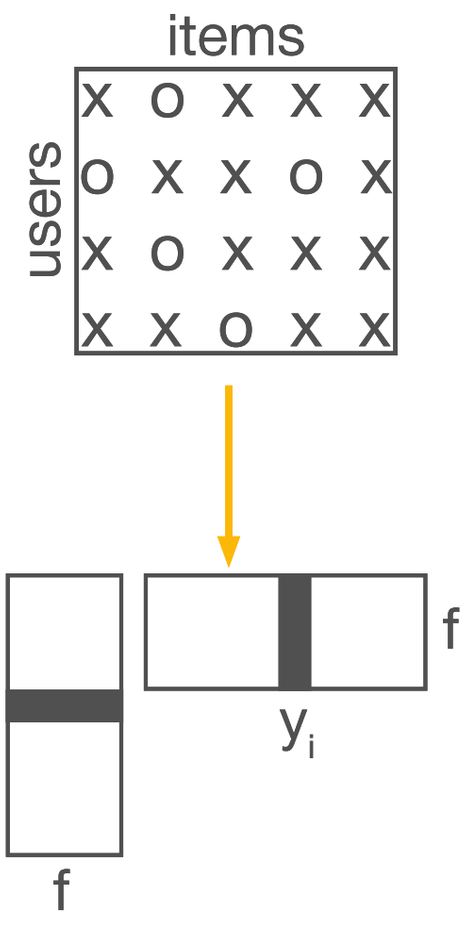
$$\min_{x_*, y_*} \sum_{r_{ui} \text{ is known}} (r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

sparse matrix



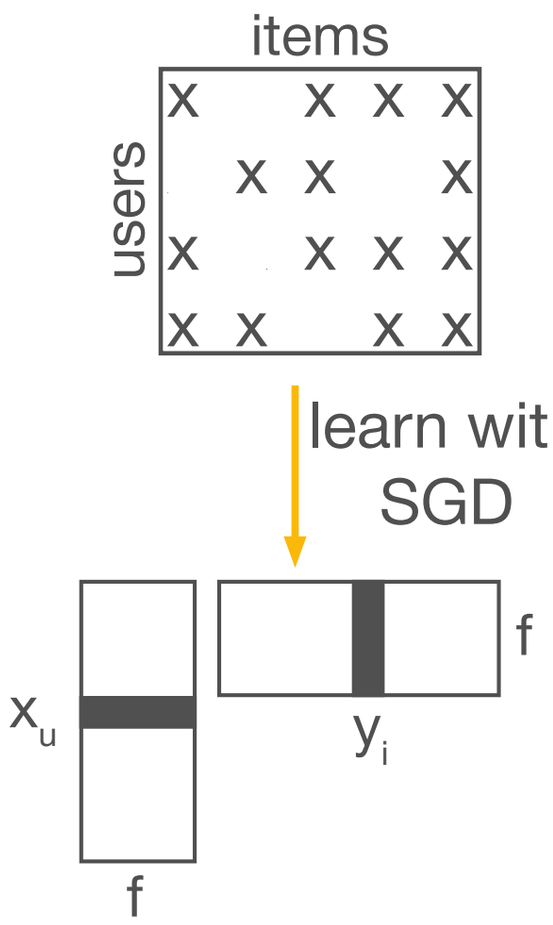
$$\min_{x_*, y_*} \sum_{r_{ui} \text{ is known}} (r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

dense matrix



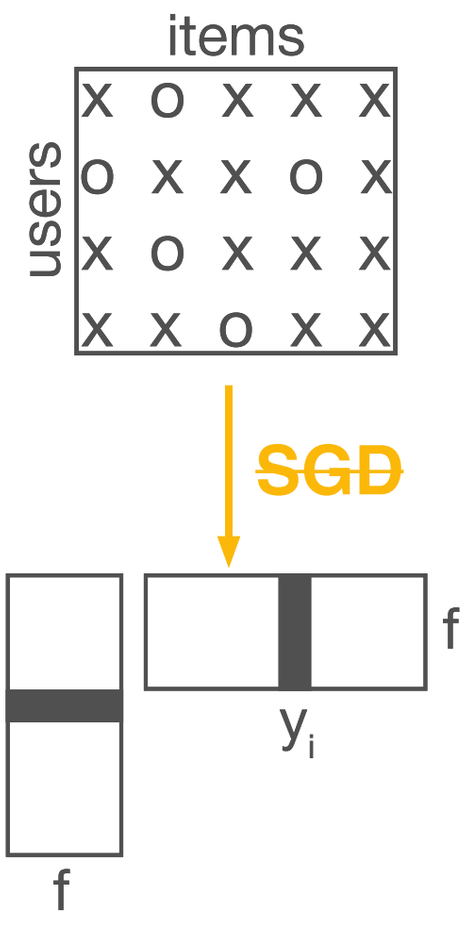
$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

sparse matrix



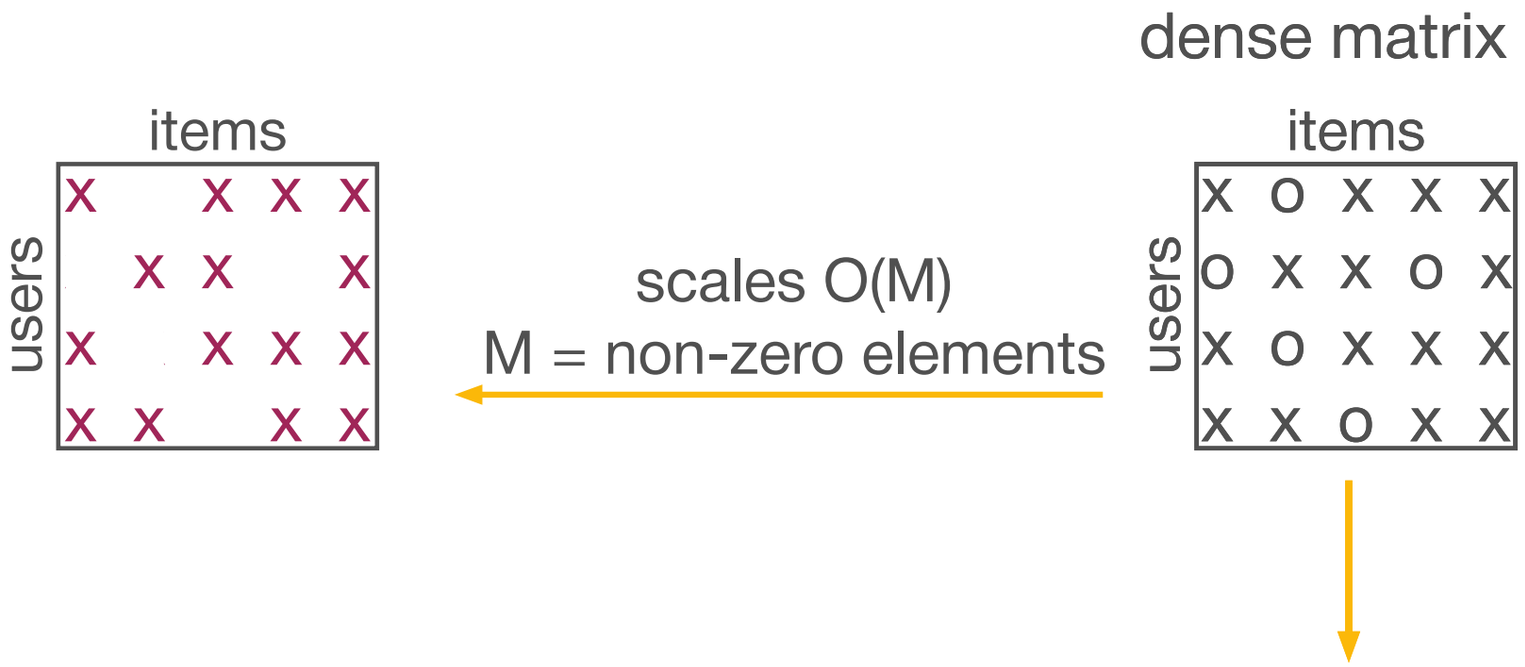
$$\min_{x_*, y_*} \sum_{r_{ui} \text{ is known}} (r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

dense matrix



$O(n \times m)$ terms

$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$



$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

quadratic equation

y_i = ... (analytical formula)

$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

quadratic equation

x_u = ... (analytical formula)

$$\min_{x_*, y_*} \sum_{r_{ui}} C_{ui}(r_{ui} - x_u^\top y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

- least squares problem

$y_i = ...$

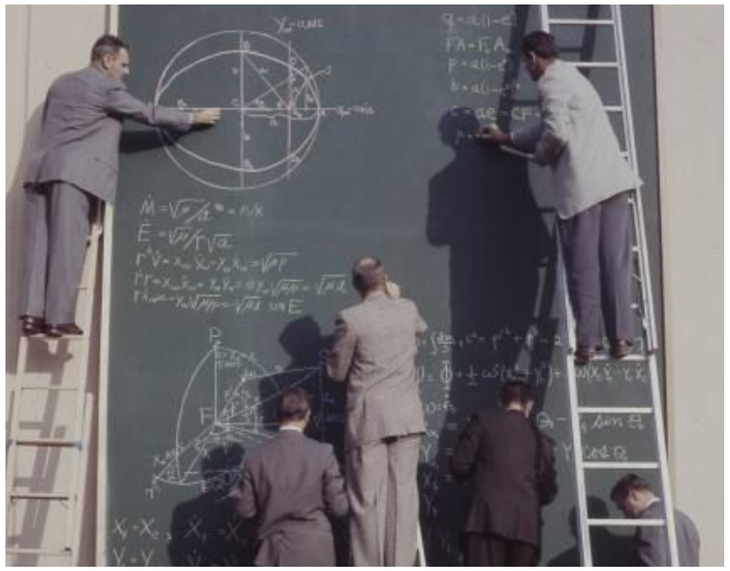
- least squares problem

$x_u = ...$

$y_i = ...$

$x_u = ...$

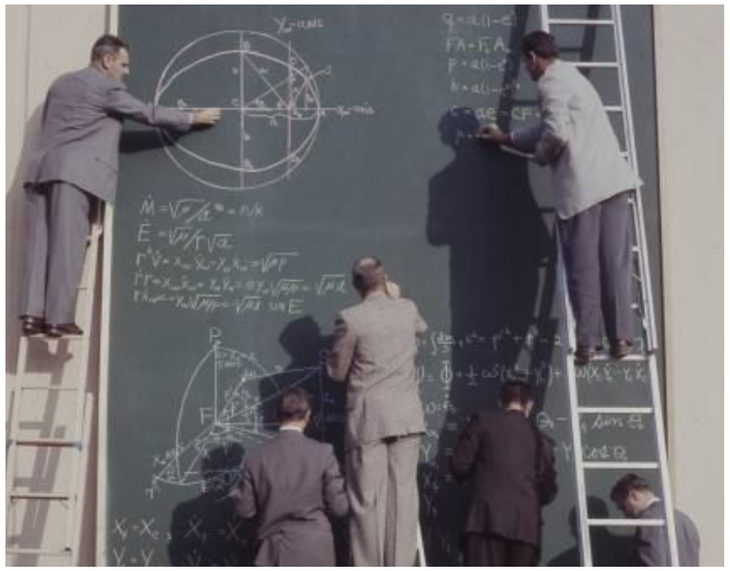
...



vs

Mathematician

Data Engineer



vs

Mathematician

Data Engineer

$$\min_{x_*, y_*} \sum_{u,i} C_{u_i} (r_{u_i} - x_u^\top y_i)^2 + \lambda (\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$Y =$  $y_i \quad m \times f$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$\mathrm{p(u)} = \mathrm{p}_{u_i}, \ i = 1..m$$

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

for each user

(1) left-hand side

(2) right-hand side

(3) [matrix] x [vector]

Data Engineer

$$Y = \boxed{} \; y_i \quad m \times f \qquad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

[f x m] x diag[m x m] x [m x f]

Data Engineer

$$Y = \boxed{}\, y_i \quad m \times f \qquad \mathrm{p(u)} = \mathrm{p}_{u_i},\ i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (\underline{Y^\top C^u Y} + \lambda I)^{-1} Y^\top C^u p(u)$$

[f x m] x diag[m x m] x [m x f]

O(f m^2) → major coefficient

[f x m] x [m x f] → [f x f]

O(f^2m)

Data Engineer

$$Y = \boxed{}\ y_i \quad m \times f \qquad \mathrm{p(u)} = \mathrm{p}_{u_i},\ i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (\underline{Y^\top C^u Y} + \lambda I)^{-1} Y^\top C^u p(u)$$

[f x m] x diag[m x m] x [m x f]

~~O(f m^2)~~ → O(f m)

[f x m] x [m x f] → [f x f]

$O(f^2 m)$

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (\underbrace{Y^\top C^u Y}_{\text{O(f}^2\text{ m)}} + \underbrace{\lambda I}_{\text{O(f)}})^{-1} Y^\top C^u p(u)$$

[f x f] [f x f]

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$(Y^\top C^u Y + \lambda I)$: $O(f^2\, m)$, [f x f]

$(Y^\top C^u Y + \lambda I)^{-1}$: $O(f^3)$, [f x f]

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$(Y^\top C^u Y + \lambda I)^{-1}$ → $O(f^2\, m + f^3)$ → $[f \times f]$

$Y^\top C^u p(u)$ → $[f \times m] \times \mathrm{diag}[m \times m] \times [m \times 1]$

$\mathrm{diag}[m \times m] \times [m \times 1]$ → $O(f\, m)$ → $[f \times m] \times [m \times 1]$

$[f \times m] \times [m \times 1]$ → $O(f\, m)$ → $[f \times 1]$

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$O(f^2\, m + f^3)$ → $[f \times f]$

$O(f^2)$

$O(f\, m)$ → $[f \times 1]$

$[f \times f] \times [f \times 1] \longrightarrow [f \times 1]$

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$O(f^2\, m + f^3)$ → $[f \times f]$

$O(f^2)$

$O(f\, m)$ → $[f \times 1]$

$[f \times f] \times [f \times 1] \longrightarrow [f \times 1]$

overall: $O(f^2 m + f^3 + f^2 + fm) \longrightarrow O(f^2 m + f^3)$

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

for each user

$O(f^2\, m + f^3)$ → [f x f]

$O(f^2)$

$O(f\, m)$ → [f x 1]

[f x f] x [f x 1] → [f x 1]

overall: $O(f^2 m + f^3 + f^2 + fm) \longrightarrow O(f^2 m + f^3)$

overall: $O(f^2 \mathbf{mn} + f^3 n)$

Data Engineer

$$Y = \boxed{} \; y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

O(f² m)

[f x f]

for each user

overall: O(f²**mn** + f³n)

Data Engineer

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

for each user

overall: O(f²**mn** + f³n)

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

$$\mathrm{c}_{u_i} = 1 + \alpha r_{u_i};$$

$$\mathrm{c}_{u_i} = 1 + \alpha log(1 + \frac{r_{u_i}}{\varepsilon})$$

=0 if r_{ui} =0

for each user

overall: O(f²**mn** + f³n)

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

for each user

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

m_u non-zero elements (m_u << m)

$$\mathrm{c}_{u_i} = 1 + \alpha r_{u_i};$$

$$\mathrm{c}_{u_i} = 1 + \alpha log(1 + \frac{r_{u_i}}{\varepsilon})$$

=0 if r_{ui} =0

overall: O(f²**mn** + f³n)

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

for each user

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

$\mathbf{m_u}$ non-zero elements ($m_u << m$)

O(f² m_u)

[f x f]

overall: O(f²**mn** + f³n)

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

for each user

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

$\mathbf{m_u}$ non-zero elements ($m_u << m$)

O(f² m_u)

[f x f]

overall: O(f²**mn** + f³n) -> O(f²**m** + f²**M** + f³n)

$$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix} y_i \quad m \times f \quad \mathrm{p(u)} = \mathrm{p}_{u_i}, \; i = 1..m$$

$$\mathrm{C}^u = diag[C^u_{ii} = C_{ui}]$$

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

Data Engineer

for each user

O(f² m)

[f x f]

$$Y^\top C^u Y = Y^\top Y + Y^\top (C^u - I_{m \times m}) Y$$

O(f² m)

[f x f]

$\mathbf{m_u}$ non-zero elements ($m_u << m$)

O(f² m_u)

[f x f]

overall: O(f²**mn** + f³n) -> O(f²**M** + f³n)

$$x_u = (Y^\top C^u Y + \lambda I)^{-1} Y^\top C^u p(u)$$

$$\mathrm{y}_i = (X^\top C^i X + \lambda I)^{-1} X^\top C^i p(i)$$

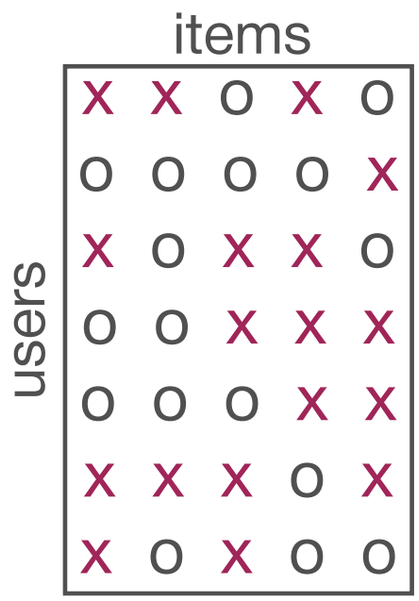
for each user

overall: $O(f^2M + f^3n)$

for each item

overall: $O(f^2M + f^3m)$

1 iALS iteration: $O(f^2M + f^3(n + m))$



Summary

From CF to MF RS: Goals

- **build** User-User (**UU**) and Item-Item (**II**) Collaborative Filtering Recommender System algorithms;
- **explain** the difference between **explicit** and **implicit** feedback;
- **optimize** learning with the help of **ALS** / **iALS**;
- **use** matrix factorization (**MF**) for recommendations;
- **explain** why SVD-like and **not** SVD.

Summary

- you can **explain** the usage of **normalization**, **similarity measure** and **neighbourhood selection** in Collaborative Filtering algorithms (and consequently you can **build UU and II CF**)

- you can **analyze** complexity of CF algorithms and **estimate** its influence on the accuracy of predictions

Summary

- you can **explain** the difference between explicit and implicit feedback datasets (**no negative feedback / noise / evaluation / preference vs confidence**)
- you can **compare** complexity of SGD versus ALS
- you can **calculate iALS** faster than O(**mn**) complexity (precise - $O(f^2\mathbf{M} + f^3(\mathbf{n} + \mathbf{m}))$)

Summary

- you know the benefits of **dimensionality reduction** approach
- you understand the concept of latent features
- you can explain what **matrix factorization** is
- dimensionality reduction and neighborhood models can be combined

Summary

- You know what SVD is and what its key features are
- You understand the role it plays in the **PureSVD** model
- You can explain how **folding-in** approach works and how to use it for online recommendations

Thank you! Questions?

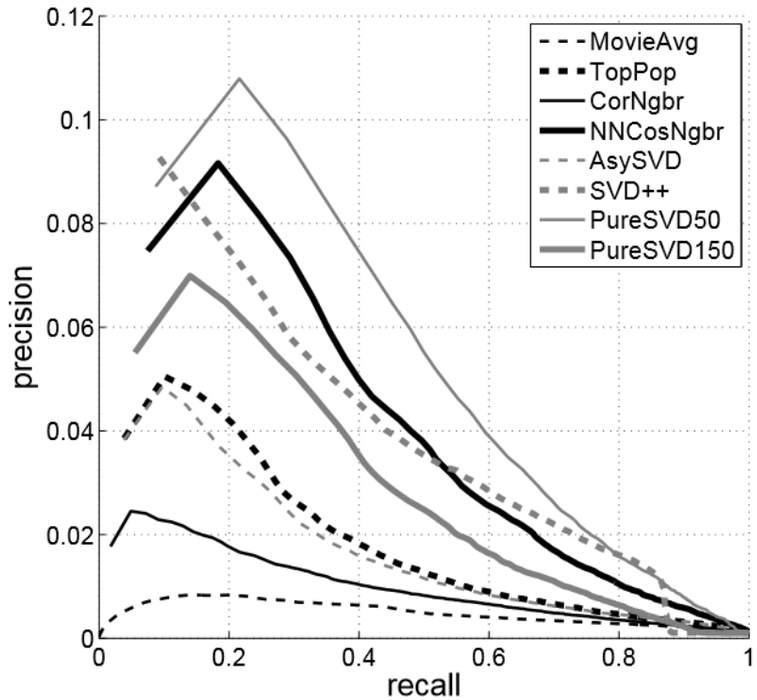
Feedback

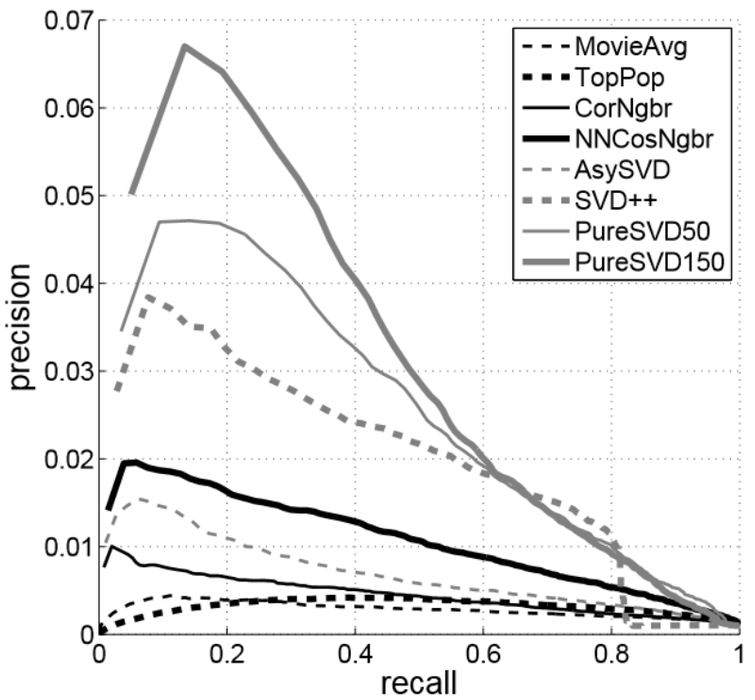
лекции: https://goo.gl/forms/09CxzkyxWrJ2hLlw1

семинары: https://goo.gl/forms/SIa0Elv5PrrEnwwa2

Appendix

PureSVD – quality of recommendations





Netflix data: complete dataset (left) and “long-tail” (right).

P. Cremonesi, Y.Koren, R.Turrin, “Performance of Recommender Algorithms on Top-N Recommendation Tasks“,Proceedings of the 4th ACM conference on Recommender systems, 2011.

Note: Funk SVD, SVD++, TimeSVD++, Asymmetric SVD … **are not** the SVD!

Контрольные вопросы

1. Запишите формулы для User-User и Item-Item коллаборативной фильтрации.

2. Каким образом происходит обучение ALS алгоритма?

3. Приведите примеры explict / implicit feedback для различных рекомендательных систем

4. Каким образом происходит обучение iALS алгоритма (учет неявной информации, целевая функция для оптимизации, сложность алгоритма "в лоб" и с оптимизацией)

References

Теоретический минимум:

1. лекция Introduction to Machine Learning 10-701 CMU 2015 (Alex Smola);
2. лекция Воронцова про коллаборативную фильтрацию;

Учебная программа:

1. лекции по RS на Coursera (последняя неделя в рамках курса "Big Data Applications: Machine Learning at Scale");
2. Xavier Amatriain Lectures 1+2, Lectures 3+4 (2014);

References (Deep Dive)

1. специализация про RS на Coursera (University of Minnesota)

2. книги:

   Recommender System Handbook (2015, Ricci, Francesco, Rokach, Lior, Shapira, Bracha (Eds.))

   Recommender Systems (2016, Aggarwal, Charu C.)

   Practical Recommender Systems (2015-2017, Kim Falk)

3. Блог https://buildingrecommenders.wordpress.com/

4. материалы конференций ACM RecSys

Outline (Homeworks)

- [1 week] Building simple RS (Non-personalized, CB)

- [2 weeks] RS contest (expected 2 tracks: prediction / recommendation task)