Recommender systems

Part 1

The formulation of the problem

First view

- There is a list of users and a list of items (products, movies, songs)
- We have a feedback from users (ratings of items, clicks, purchases, likes or dislikes)
- We need to recommend every user items he would like

Movies recommendations: possible problem formalization

- There are ratings that have been chosen by user for movies that he have already watched
- We need to:
  - Predict ratings, that could be chosen by user for other movies
  - Recommend movies that user will like more (according to our predictions)

Movies recommendations

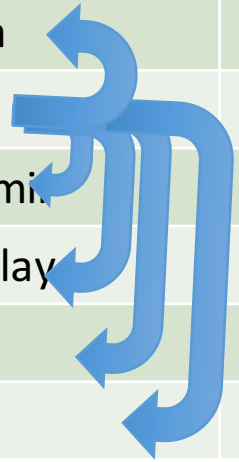
	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	
Vladimir			3	5
Nikolay	3		4	5
Peter				4
Ivan		5	3	3

Movies recommendations

	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	?
Vladimir			3	5
Nikolay	3		4	5
Peter				4
Ivan		5	3	3

User-based kNN

	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	
Vladimir			3	5
Nickolay	3		4	5
Peter				4
Ivan		5	3	3



User-based kNN

	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	?
Vladimir			3	5
Nikolay	3		4	5
Peter				4
Ivan		5	3	3

Item-based kNN

	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	?
Vladimir			3	5
Nikolay	3		4	5
Peter				4
Ivan		5	3	3

Item-based kNN

	Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
Maria	5	4	1	2
Julia		5	2	?
Vladimir			3	5
Nikolay	3		4	5
Peter				4
Ivan		5	3	3

Matrix factorizations

j

i		Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
	Maria	5	4	1	2
	Julia	5	5	2	
	Vladimir			3	5
	Nikolay	3	?	4	5
	Peter				4
	Ivan		5	3	3

u_i - “user interests”

v_j - “movies parameters”

$$x_{ij} \approx \langle u_i, v_j \rangle = \sum_{k=1}^{K} u_{ik} v_{jk}$$

Matrix factorizations: fitting model

$$x_{ij} \approx \langle u_i, v_j \rangle$$

$$\sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to min$$

Measuring the quality

Is the recommendations quality == rating predicting quality?

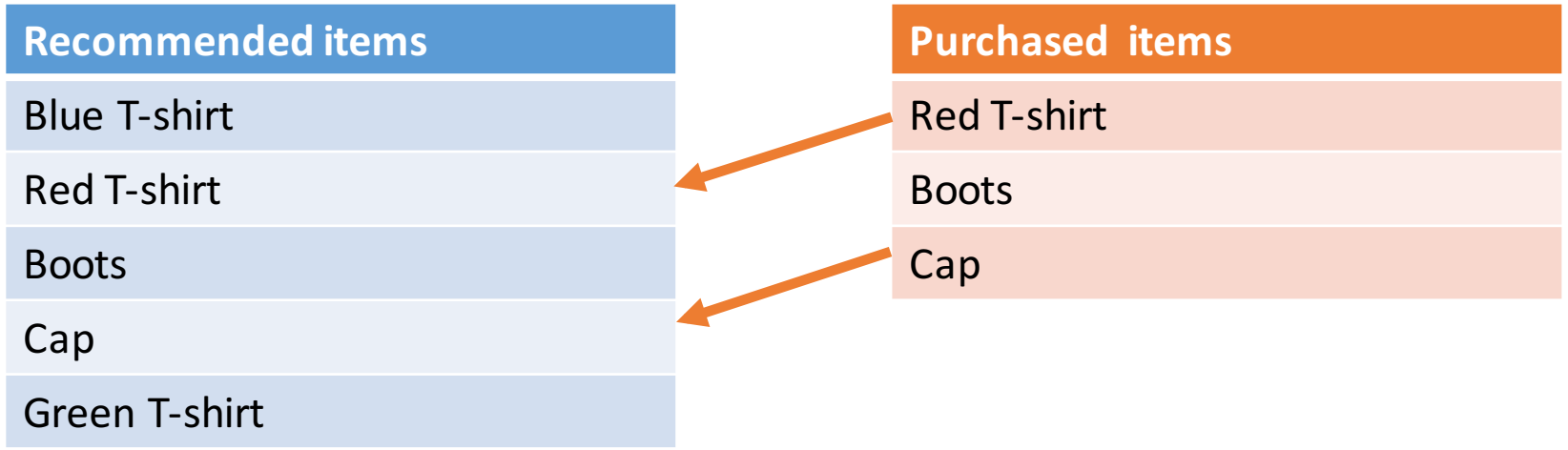
Variants:

- Root mean square error (RMSE) of ratings predictions
- Mean absolute error (MAE) of ratings predictions

Do we measure quality in a right way?

- We are measuring: the quality of ratings predictions
- What should be measured: the quality of recommendations

Recall@k



k – count of the recommended items

$$\text{Recall@k} = \frac{purchased\ from\ recommended}{purchased\ items}$$

AverageRecall@k - average in user sessions Recall@k

Precision@k

Recommended items
Blue T-shirt
Red T-shirt
Boots
Cap
Green T-shirt

Purchased items
Red T-shirt
Boots
Cap

k – count of the recommended items

$$\text{Precision@k} = \frac{purchased\ from\ recommended}{k}$$

AveragePrecision@k - average in user sessions Precision@k

Recommendations in retail

j

i

	Dress	Boots	Jeans	T-shirt
Maria	1		1	
Julia	1	1		1
Vladimir		1	1	
Nikolay	1	?	1	
Peter		1	1	
Ivan			1	1

Difference from movie recommendations

- No negative examples
- Simple connection with revenue

Possible solution

- Predict which items will be purchased by user
- Maximize the revenue

Maximizing income

Item 1	**Item 2**	**Item 3**	**Item 4**

Maximizing income

Item 1	Item 2	Item 3	Item 4

Probability:	p_1	p_2	p_3	p_4
Price:	c_1	c_2	c_3	c_4

Maximizing income

Puma
Ветровка
3 490 руб.

Crocs
Сланцы
1 990 руб.

Tony-p
Слипоны
~~1 999 руб.~~ 1 590 руб.

Champion
Брюки спортивные
~~3 599 руб.~~ 1 970 руб.

Probability:	**0.05**	**0.02**	**0.015**	**0.009**
Price:	3490	1990	1590	1970

Maximizing revenue

Puma
Ветровка
3 490 руб.

Crocs
Сланцы
1 990 руб.

Tony-p
Слипоны
~~1 999 руб.~~ 1 590 руб.

Champion
Брюки спортивные
~~3 599 руб.~~ 1 970 руб.

Probability:	**0.05**	**0.02**	**0.015**	**0.009**
Price:	3490	1990	1590	1970
Marginality:	0.1	0.4	0.4	0.2

Predicting probabilities

- Examples: tuples (user, item, timestamp)
- Classes: 1 – item will be purchased by user in this session, 0 – item won’t be purchased
- Features: user parameters, item parameters, timestamp parameters and interactions of these parameters

Negative samples

- Add all other items from catalogue as negative examples for every positive example (unreal)
- Random with uniform distribution
- Random with probabilities proportional to items popularities
- Most popular and not purchased
- Recommendations from other algorithm (not purchased)

Candidates selection

- Popular items
- Popular items from already seen categories
- Items that have high PMI (Pointwise Mutual Information) with already seen items
- Items from custom candidates list

PMI

$$\operatorname{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$

Example of items with high PMI in user sessions:
Laptop + Mouse

Examples with medium PMI:
T-shirt + Boots
Dress + Handbag
Dress + Shoes

Example with low PMI:
Shoes + Mouse

Online metrics

The quality on the historical data is high

What can we say about the quality of recommendations in production?

Online metrics

Допустим, на исторических данных качество алгоритма высокое, а будет ли оно высоким в реальности?

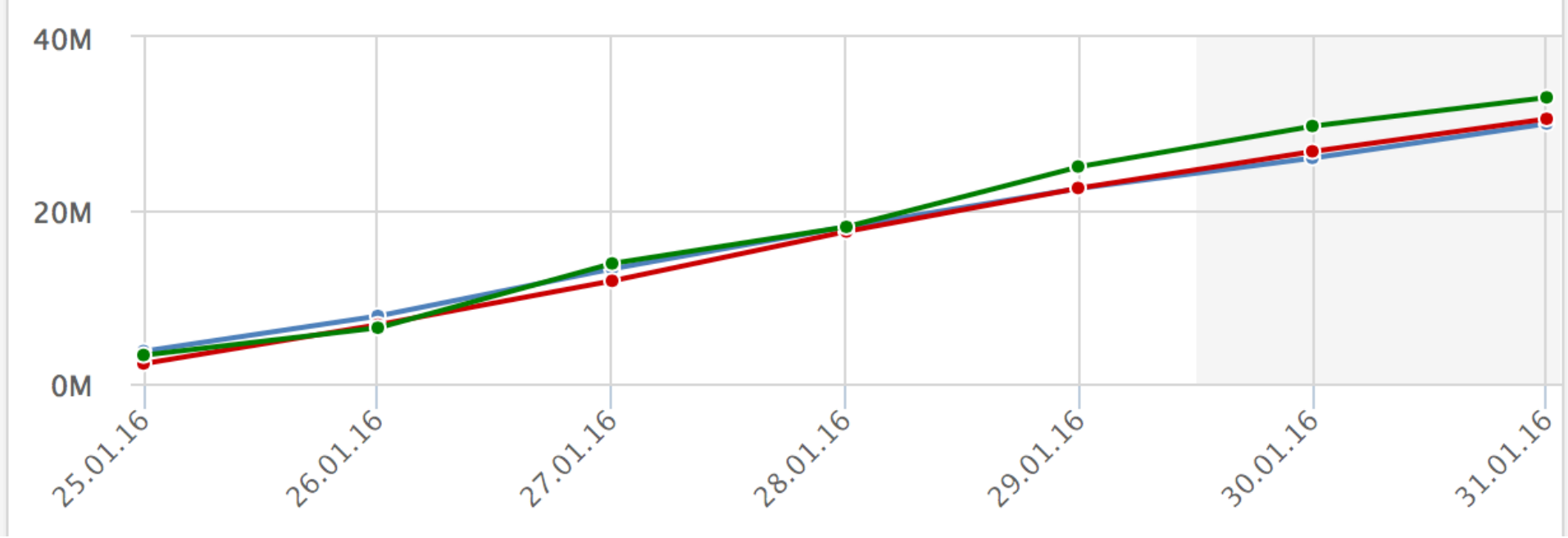
Ideas:

1. A/B test
2. Statistical significance test

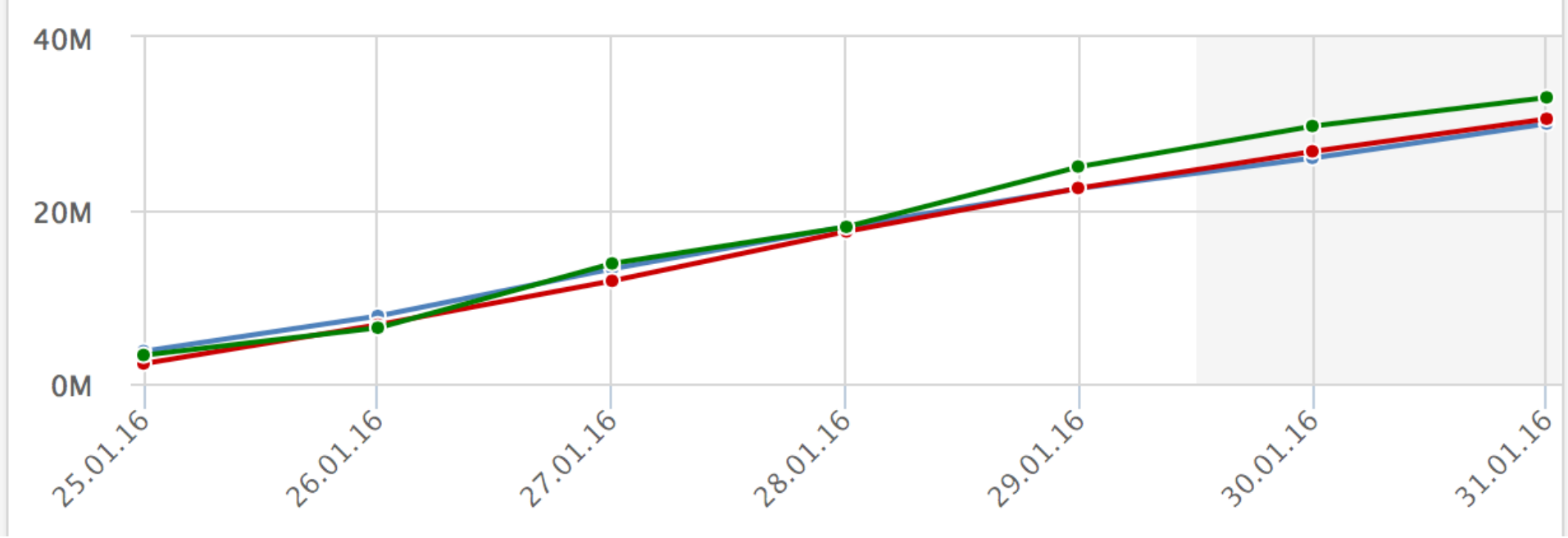
A/B test

1. Split users randomly on two groups
2. Measure online metrics (for example, purchases or revenue) in both groups for rather long period of time
3. Result is two numbers (f. ex. 2.3 M $ in group A and 2.2 M $ in group B)
4. What decision can we make?

Statistical significance: example



Statistical significance: example



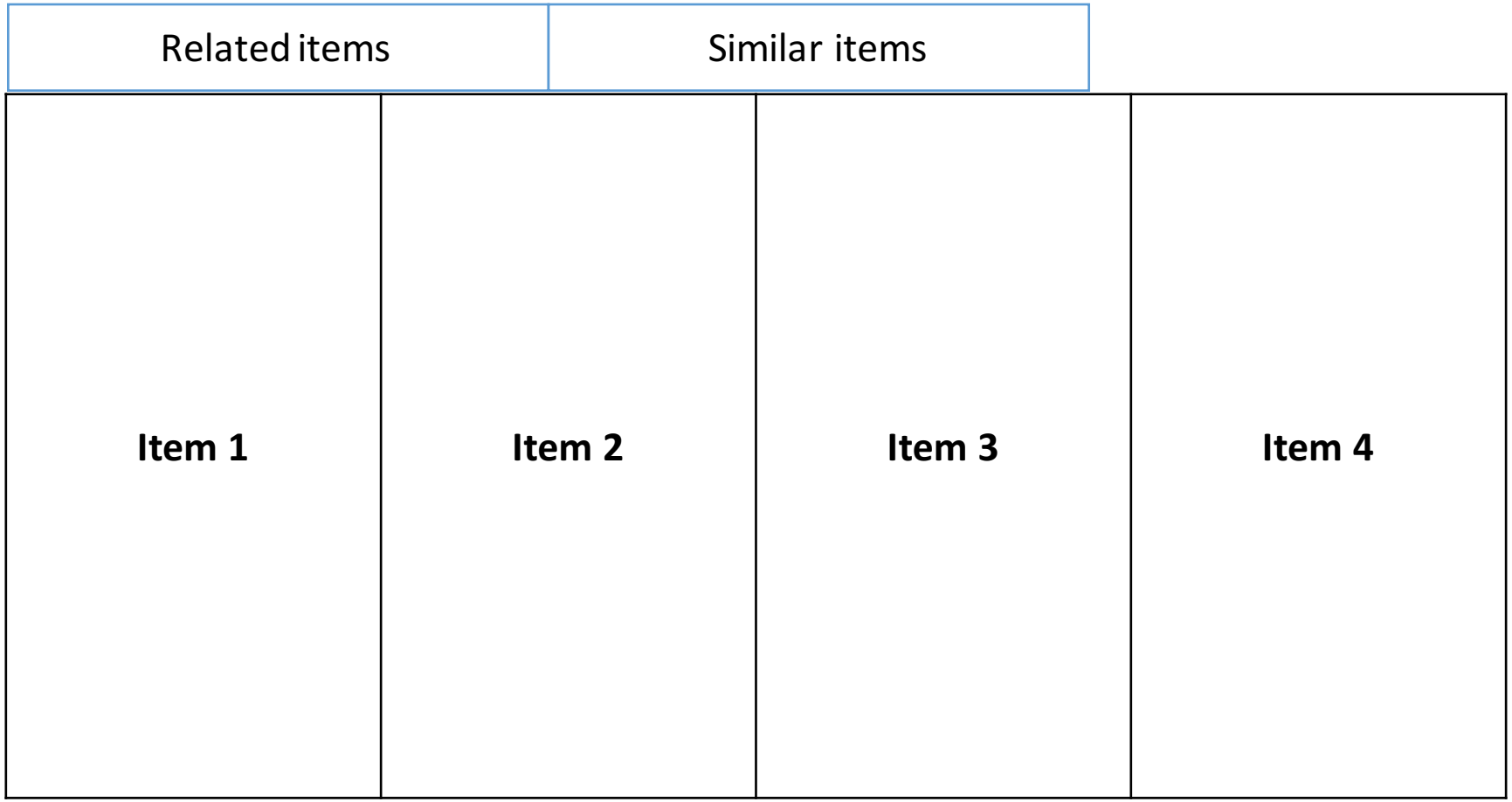
This plot is for random split on three groups

Story 1: bad split in A/B test

- Proposed variant:
  - Group = hash(user_id) % 2

- Released:
  - Group = hash(user_id + user_email) % 2

Story 2: design



Story 3: comparing two solutions

- Compared their solution with recommendations developed in another company
- Offline quality was exactly the same
- Decided not to use this recommender engine
- Some months later discovered the reason of such result

Recap

1. Don’t solve problem until the formulation of the problem is clear
2. Problem formalization should be connected with economics
3. Problem formalization includes the procedure of offline and online quality measuring
4. Good formalization and good model is less valuable than not doing nonsense

Recommender systems

Part 2

Matrix factorizations

Matrix factorization

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

Matrix factorizations

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \rightarrow min$$

Matrix factorizations

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

$$\|A\|_F = \sqrt{\sum_{i,j} {a_{ij}}^2}$$

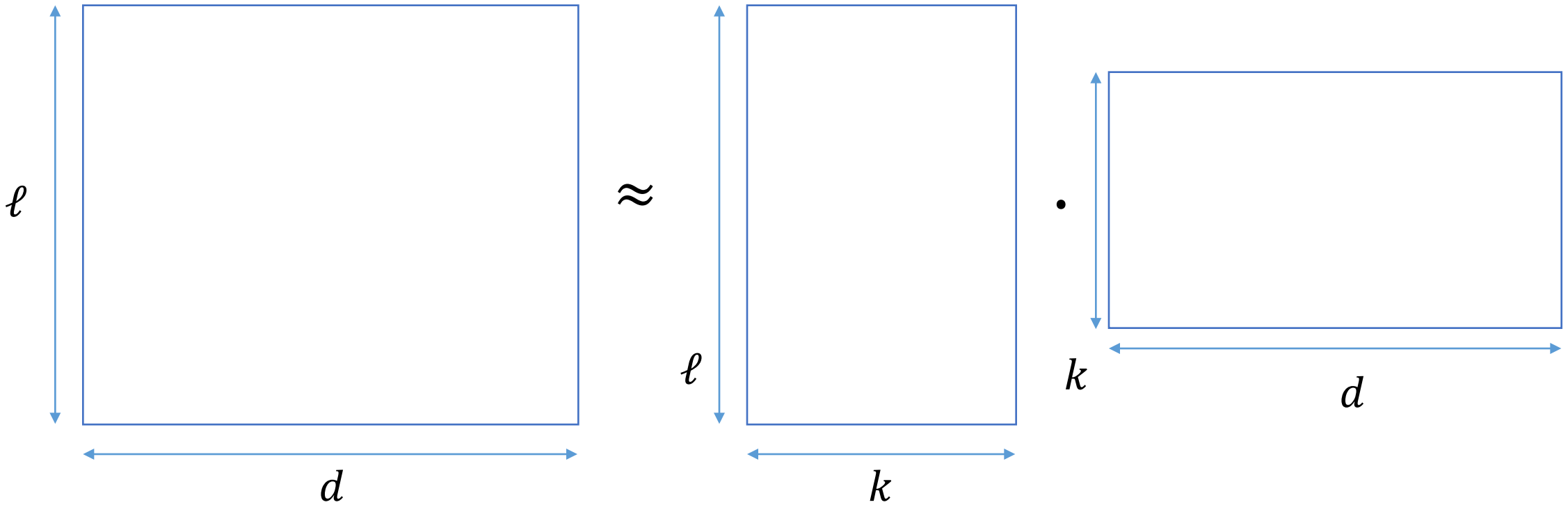
Matrix factorizations

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

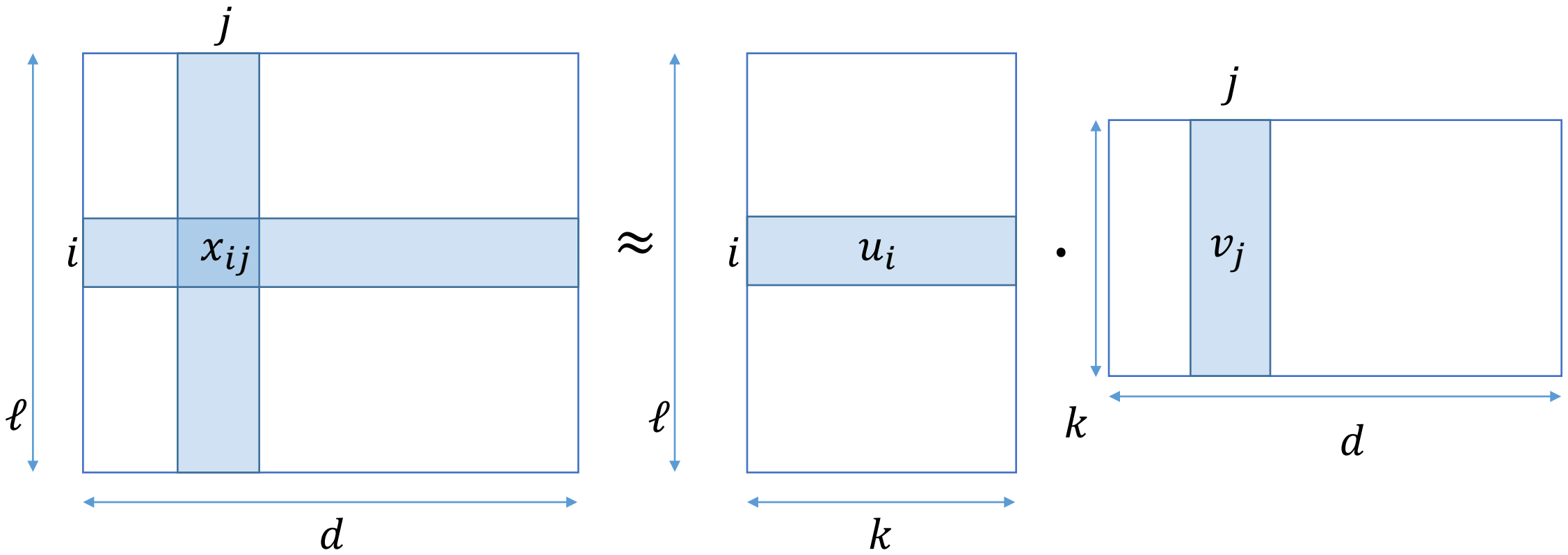
$$\sum_{i,j} \left(x_{ij} - \langle u_i, v_j \rangle\right)^2 \to min$$

Notation



$$X \approx U \cdot V^T$$

Notation



$$x_{ij} \approx < u_i, v_j >$$

Singular Vector Decomposition in algebra

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

U - *orthogonal*

Σ - *diagonal*

V - *orthogonal*

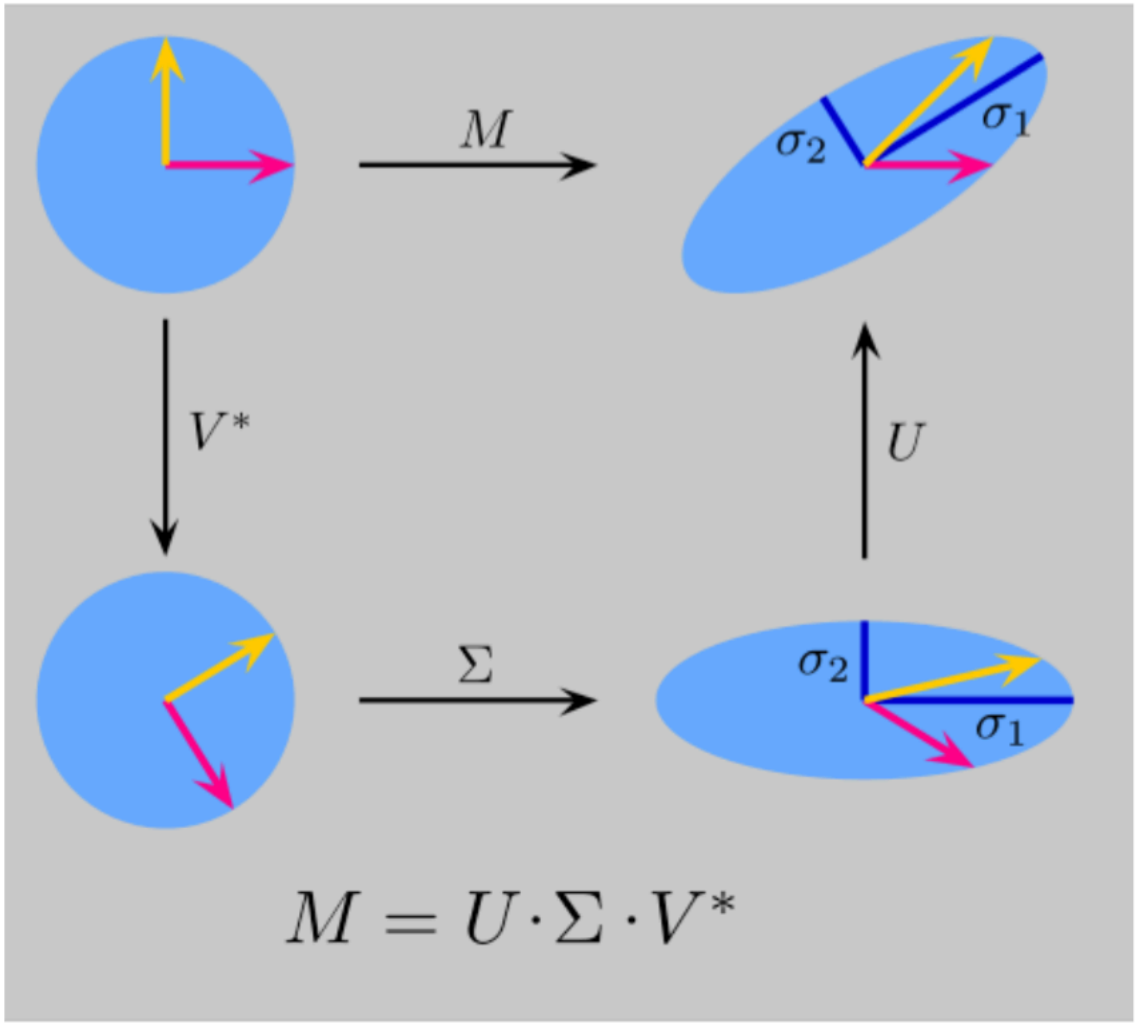
Singular Vector Decomposition in algebra

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

U - orthogonal

Σ - diagonal

V - orthogonal



SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \rightarrow min$$

SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \rightarrow min$$

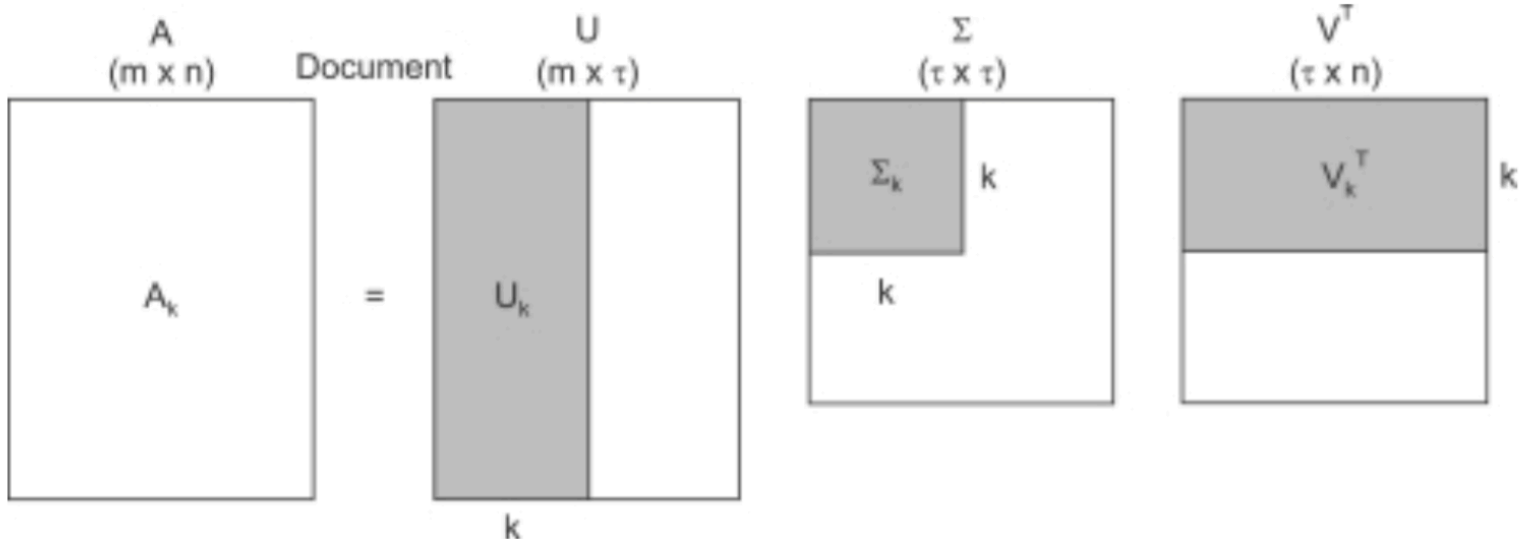
$$X = \tilde{U}\Sigma\tilde{V}^T$$

SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

$$X = \tilde{U}\Sigma\tilde{V}^T$$



SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

$$X = \tilde{U}\Sigma\tilde{V}^T$$

$\tilde{U}_k, \Sigma_k, \tilde{V}_k$- truncated SVD matrixes

$$U = \tilde{U}_k\Sigma_k, \quad V = \tilde{V}_k$$

SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

$$X = \tilde{U}\Sigma\tilde{V}^T$$

$\tilde{U}_k, \Sigma_k, \tilde{V}_k$- truncated SVD matrixes

$$U = \tilde{U}_k, \quad V = \tilde{V}_k\Sigma_k$$

SVD for matrix approximation

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\left|\left|X - U \cdot V^T\right|\right| \to min$$

$$X = \tilde{U}\Sigma\tilde{V}^T$$

$\tilde{U}_k, \Sigma_k, \tilde{V}_k$- truncated SVD matrixes

$$U = \tilde{U}_k\sqrt{\Sigma_k}, \quad V = \tilde{V}_k\sqrt{\Sigma_k}$$

”SVD” in Machine Learning

$$\underset{l \times n}{X} \approx \underset{l \times k}{U} \cdot \underset{k \times n}{V^T}$$

$$\sum_{i,j} \left(x_{ij} - \langle u_i, v_j \rangle\right)^2 \to min$$

u_i - samples “descriptions”
v_j - features “descriptions”

Movie ratings and SVD

	Пила	Улица Вязов	Ванильное небо	1+1
Маша	5	4	1	2
Юля	5	5	2	
Вова			3	5
Коля	3		4	5
Петя				4
Ваня		5	3	3

Movie ratings and SVD

j

	Пила	Улица Вязов	Ванильное небо	1+1
Маша	5	4	1	2
Юля	5	5	2	
Вова			3	5
Коля	3		4	5
Петя				4
Ваня		5	3	3

i

Movie ratings and SVD

j

i		Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
	Maria	5	4	1	2
	Julia	5	5	2	
	Vladimir			3	5
	Nikolay	3	?	4	5
	Peter				4
	Ivan		5	3	3

u_i - “user interests”

v_j - “movies parameters”

$$x_{ij} \approx \langle u_i, v_j \rangle = \sum_{k=1}^{K} u_{ik} v_{jk}$$

Word frequencies and SVD

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

Word frequencies and SVD

j

i

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

Word frequencies and SVD

j

i

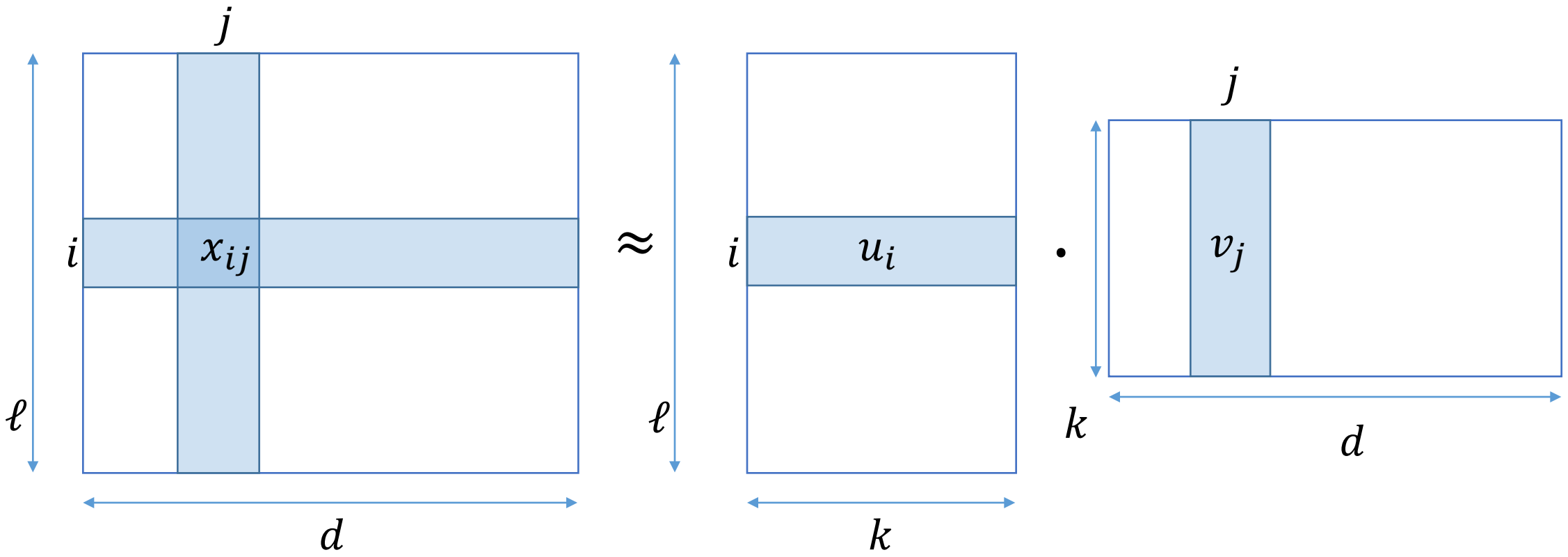
	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

$$x_{ij} \approx < u_i, v_j >$$

u_i - «темы» документов

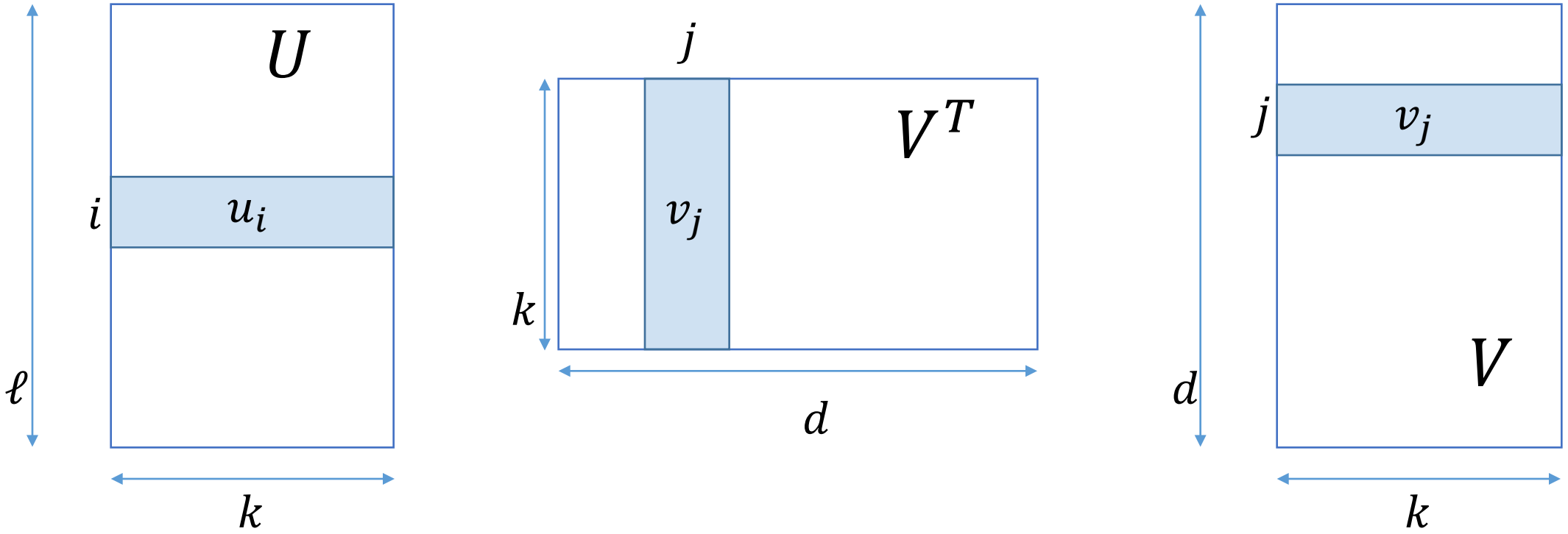
v_j - «темы» слов

A little bit more about notations



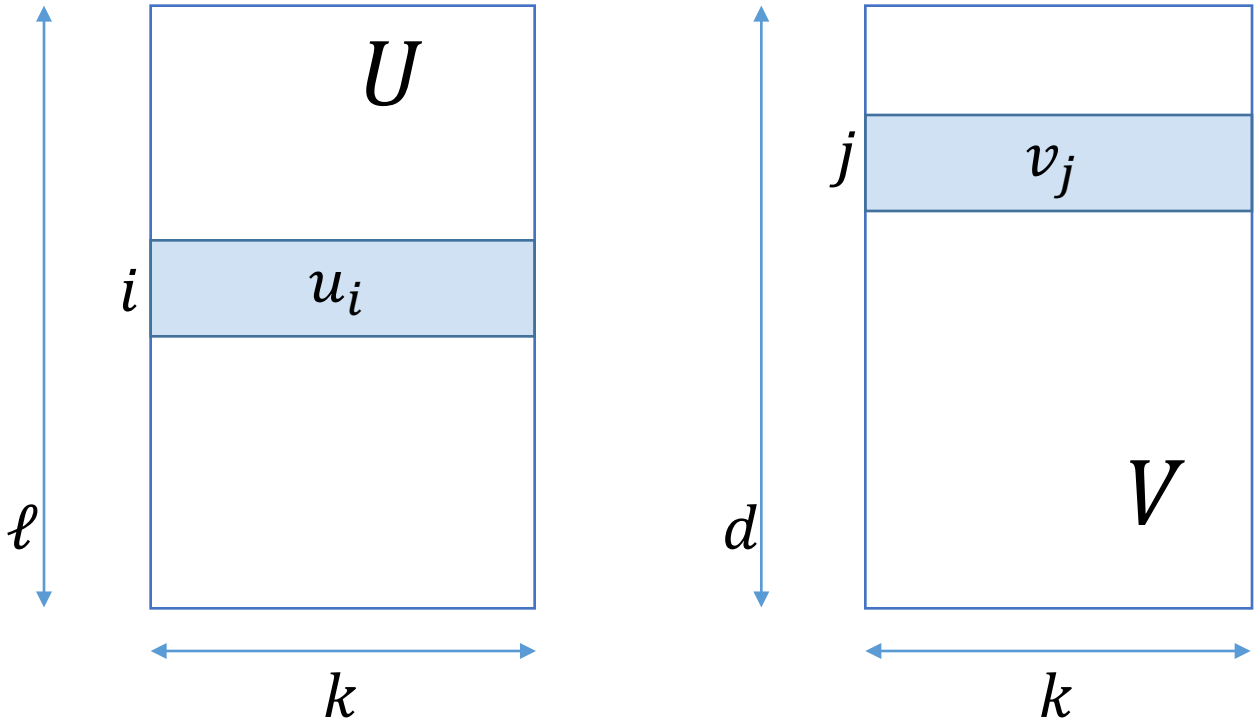
$$x_{ij} \approx < u_i, v_j >$$

A little bit more about notations



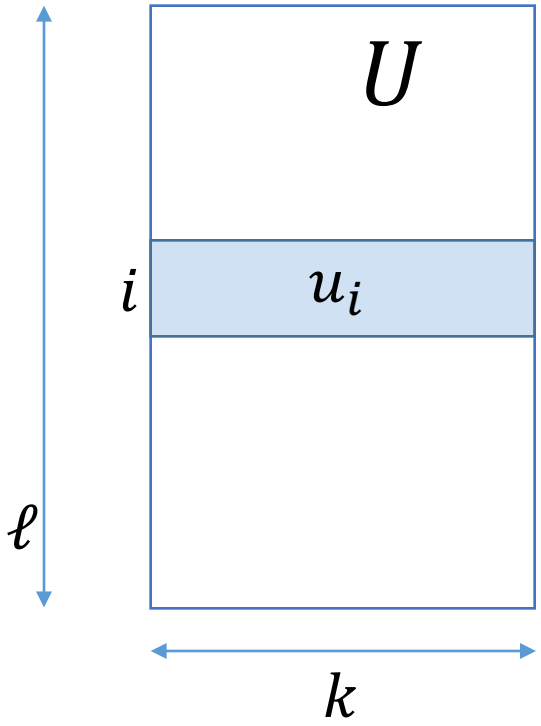
$$x_{ij} \approx < u_i, v_j >$$

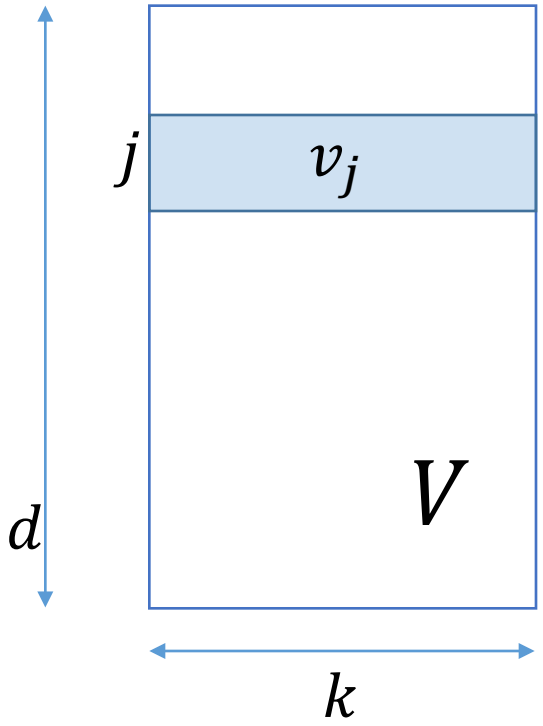
A little bit more about notations



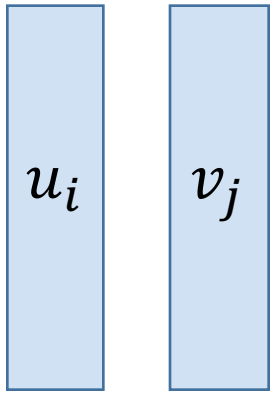
$$x_{ij} \approx < u_i, v_j >$$

A little bit more about notations

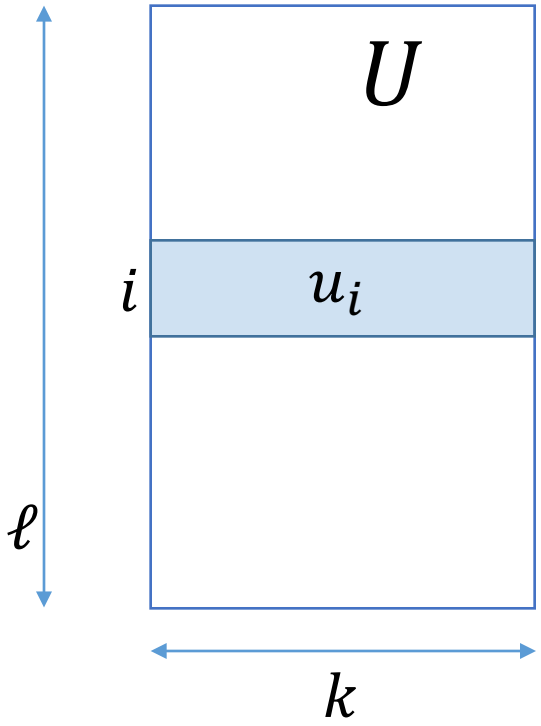


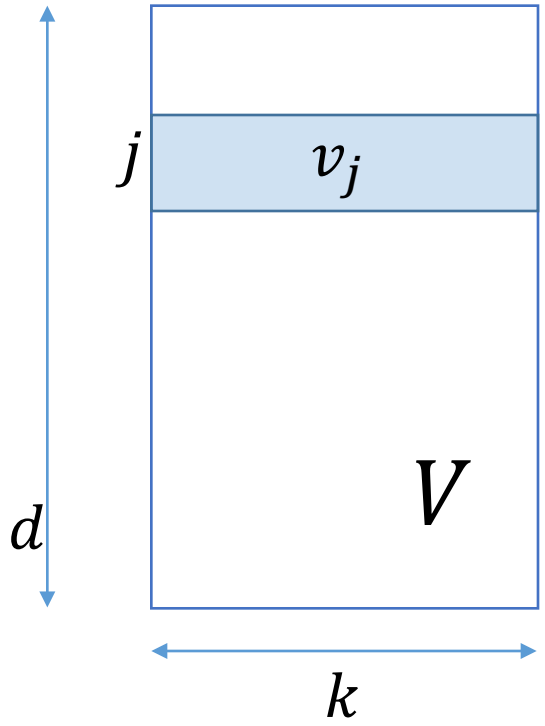


$$x_{ij} \approx < u_i, v_j >$$

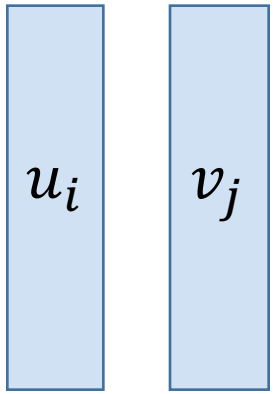


A little bit more about notations



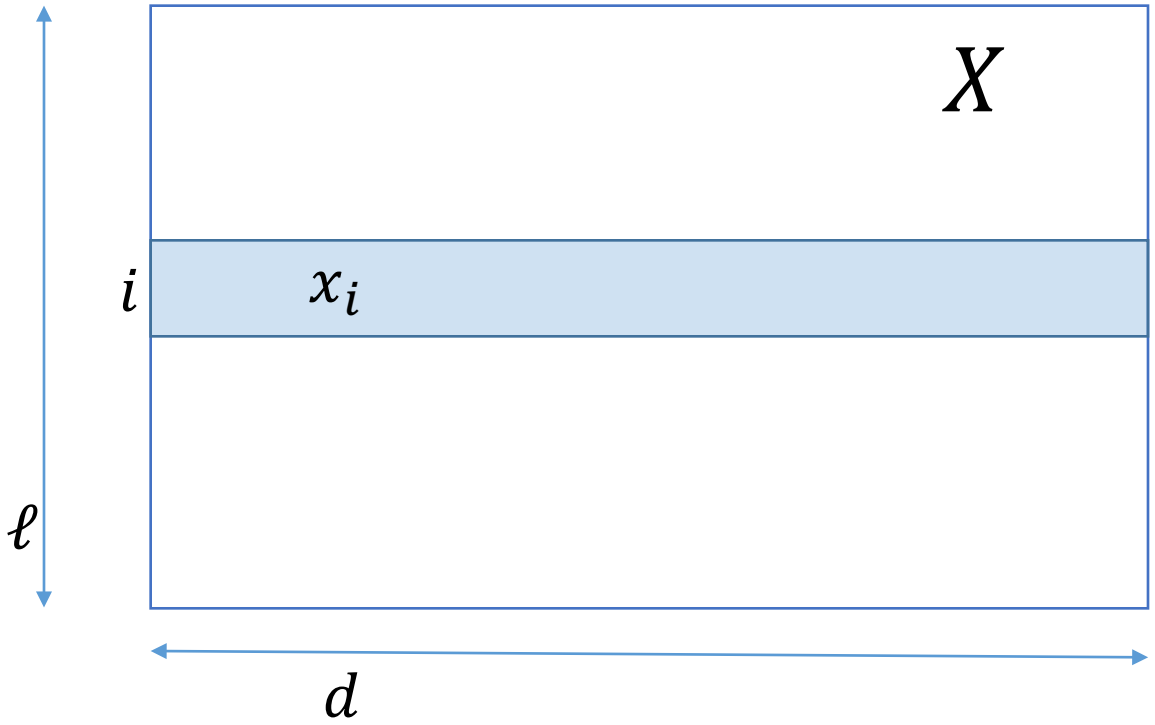


$$x_{ij} \approx < u_i, v_j >$$



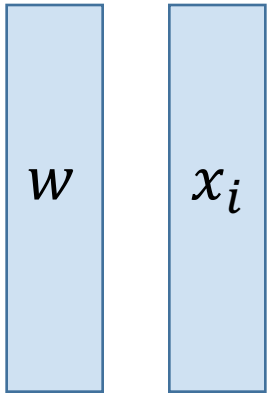
$$x_{ij} \approx u_i^T v_j$$

A little bit more about notations



In linear models:

$$< w, x_i > = w^T x_i$$



Popular notations

$$X \approx UV^T$$

$$X \approx PQ^T$$

$$X \approx WH$$

$$X \approx \Phi\Theta$$

Problem formulation in SVD

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to \min_{u_i, v_j}$$

Gradient Decent (GD)

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to \min_{u_i, v_j}$$

$$\frac{\partial Q}{\partial u_i} = \sum_{\tilde{i},j} \frac{\partial}{\partial u_i} \left(\langle u_{\tilde{i}}, v_j \rangle - x_{\tilde{i}j} \right)^2 = \sum_j 2 \left(\langle u_i, v_j \rangle - x_{ij} \right) \frac{\partial \langle u_i, v_j \rangle}{\partial u_i} =$$

$$= \sum_j 2 \left(\langle u_i, v_j \rangle - x_{ij} \right) v_j \qquad \varepsilon_{ij} = \left(\langle u_i, v_j \rangle - x_{ij} \right) \text{ - error on } x_{ij}$$

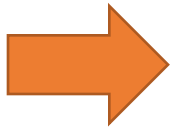
$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \sum_j \varepsilon_{ij} v_j$$

Stochastic Gradient Decent (SGD)

GD:

$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \sum_j \varepsilon_{ij} v_j$$

$$v_j^{(t+1)} = v_j^{(t)} - \eta_t \sum_i \varepsilon_{ij} u_i$$



SGD:

$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \varepsilon_{ij} v_j$$

$$v_j^{(t+1)} = v_j^{(t)} - \eta_t \varepsilon_{ij} u_i$$

For random i, j

SGD: pros and cons

+Simple

+Converge (usually)

- Slow
- Needs good steps choosing (γ_t and η_t)
- Extremely slow with constant step

Alternating Least Squares (concept)

$$Q \rightarrow \min_{u_i, v_j}$$

Iteratively repeat:

$$\frac{\partial Q}{\partial u_i} = 0 \quad \Rightarrow \quad u_i \qquad \frac{\partial Q}{\partial v_j} = 0 \quad \Rightarrow \quad v_j$$

First step in ALS

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to \min_{u_i, v_j}$$

$$\frac{\partial Q}{\partial u_i} = \sum_j 2\left(\langle u_i, v_j \rangle - x_{ij} \right) v_j = 0 \qquad \sum_j v_j \langle v_j, u_i \rangle = \sum_j x_{ij} v_j$$

$$\sum_j v_j {v_j}^T u_i = \sum_j x_{ij} v_j \qquad \underbrace{\left(\sum_j v_j {v_j}^T \right)}_{A} u_i = \underbrace{\sum_j x_{ij} v_j}_{b}$$

ALS: algorithm

Repeat for random (i, j) until converge:

$$\left(\sum_j v_j {v_j}^T\right) u_i = \sum_j x_{ij} v_j \quad \Rightarrow \quad u_i \qquad \text{(Least Squares method)}$$

$$\left(\sum_i u_i {u_i}^T\right) v_j = \sum_i x_{ij} u_i \quad \Rightarrow \quad v_j$$

Regularization

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 + \alpha \sum_i \|u_i\|^2 + \beta \sum_j \|v_j\|^2 \to \min_{u_i, v_j}$$

α and β - small positive constants (0.001, 0.01, 0.05)

Prediction model

j

i		Saw	Nightmare on Elm Street	Vanilla Sky	The Intouchables
	Maria	5	4	1	2
	Julia	5	5	2	
	Vladimir			3	5
	Nikolay	3	?	4	5
	Peter				4
	Ivan		5	3	3

u_i - “user interests”

v_j - “movies parameters”

$$x_{ij} \approx \langle u_i, v_j \rangle = \sum_{k=1}^{K} u_{ik} v_{jk}$$

Minimizing function

$$x_{ij} \approx < u_i, v_j >$$

$$\sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to min$$

Bias

$$x_{ij} \approx \mu + < u_i, v_j >$$

$$\sum_{i,j} \left(\mu + \langle u_i, v_j \rangle - x_{ij} \right)^2 \rightarrow min$$

Baseline predictors

$$x_{ij} \approx \mu + b_i^u + b_j^v + < u_i, v_j >$$

$$\sum_{i,j} \left(\mu + b_i^u + b_j^v + \langle u_i, v_j \rangle - x_{ij}\right)^2 \to min$$

Regularization

$$\sum_{i,j}\left(\mu + b_i^u + b_j^v + \langle u_i, v_j\rangle - x_{ij}\right)^2 + \alpha\sum_i \|u_i\|^2 + \beta\sum_j \|v_j\|^2 +$$

$$+\gamma\sum_i {b_i^u}^2 + \delta\sum_j {b_j^v}^2 \to min$$

Recommendations

j

i		Вечернее платье	Поднос для писем	iPhone 6s	Шуба D&G
	Маша	1		1	
	Юля	1	1		1
	Вова		1	1	
	Коля	1	?	1	
	Петя		1	1	
	Ваня			1	1

Recommendations in retail

j

i	Dress	Boots	Jeans	T-shirt
Maria	1		1	
Julia	1	1		1
Vladimir		1	1	
Nikolay	1	?	1	
Peter		1	1	
Ivan			1	1

Почему нужно что-то менять

j

i	Dress	Boots	Jeans	T-shirt
Maria	1		1	
Julia	1	1		1
Vladimir		1	1	
Nikolay	1	?	1	
Peter		1	1	
Ivan			1	1

$$x_{ij} = 1 \approx < u_i, v_j >$$

$$\sum_{i,j:x_{ij} \neq 0} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \rightarrow min$$

$$u_i = \frac{1}{\sqrt{d}}(1 \quad \cdots \quad 1)$$

$$v_j = \frac{1}{\sqrt{d}}(1 \quad \cdots \quad 1)$$

Explicit и implicit

- **Explicit feedback:** positive and negative examples (f.ex. high and low movie ratings, likes and dislikes and so on)
- **Implicit feedback:** only positive feedback (purchases, clicks, likes) or only negative feedback

Implicit matrix factorization

$$\sum_{i,j} w_{ij}\left(\langle u_i, v_j\rangle - x_{ij}\right)^2 \to min$$

Sum over the all indices (not only indices of known matrix elements – suppose unknown elements are equal to zero)

w_{ij} is high for $x_{ij} \neq 0$
and rather low for $x_{ij} = 0$

Implicit ALS

$$\sum_{i,j} w_{ij}\left(\langle u_i, v_j\rangle - x_{ij}\right)^2 \to min$$

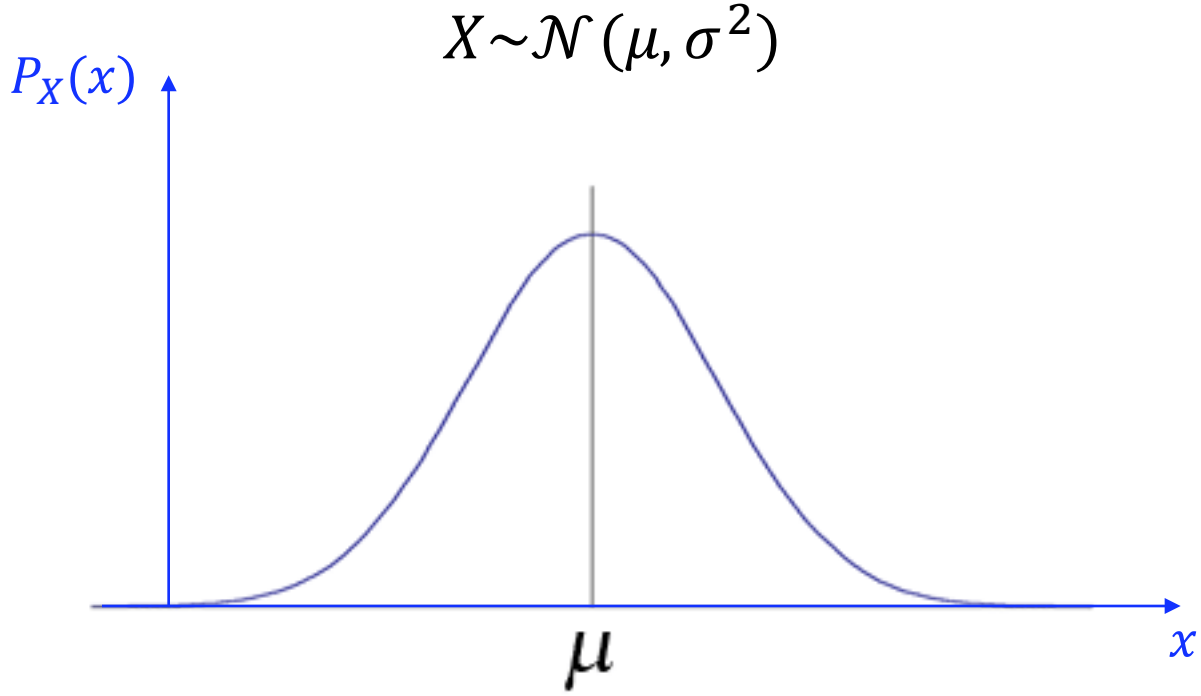
$$w_{ij} = 1 + \alpha\left|x_{ij}\right| \qquad \alpha = 10, 100, 1000$$

Fitting u_i, v_j with ALS

Problem formulation in SVD

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to \min_{u_i, v_j}$$

Normal distribution



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

SVD and normal distribution

$$x_{ij} = \langle u_i, v_j \rangle + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$x_{ij} \sim \mathcal{N}\left(\langle u_i, v_j \rangle, \sigma^2\right)$$

$$\prod_{i,j} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(x_{ij} - \langle u_i, v_j \rangle\right)^2}{2\sigma^2}} \to max$$

$$\sum_{i,j} \frac{\left(x_{ij} - \langle u_i, v_j \rangle\right)^2}{2\sigma^2} - \frac{1}{2} \ln 2\pi\sigma^2 \to min$$

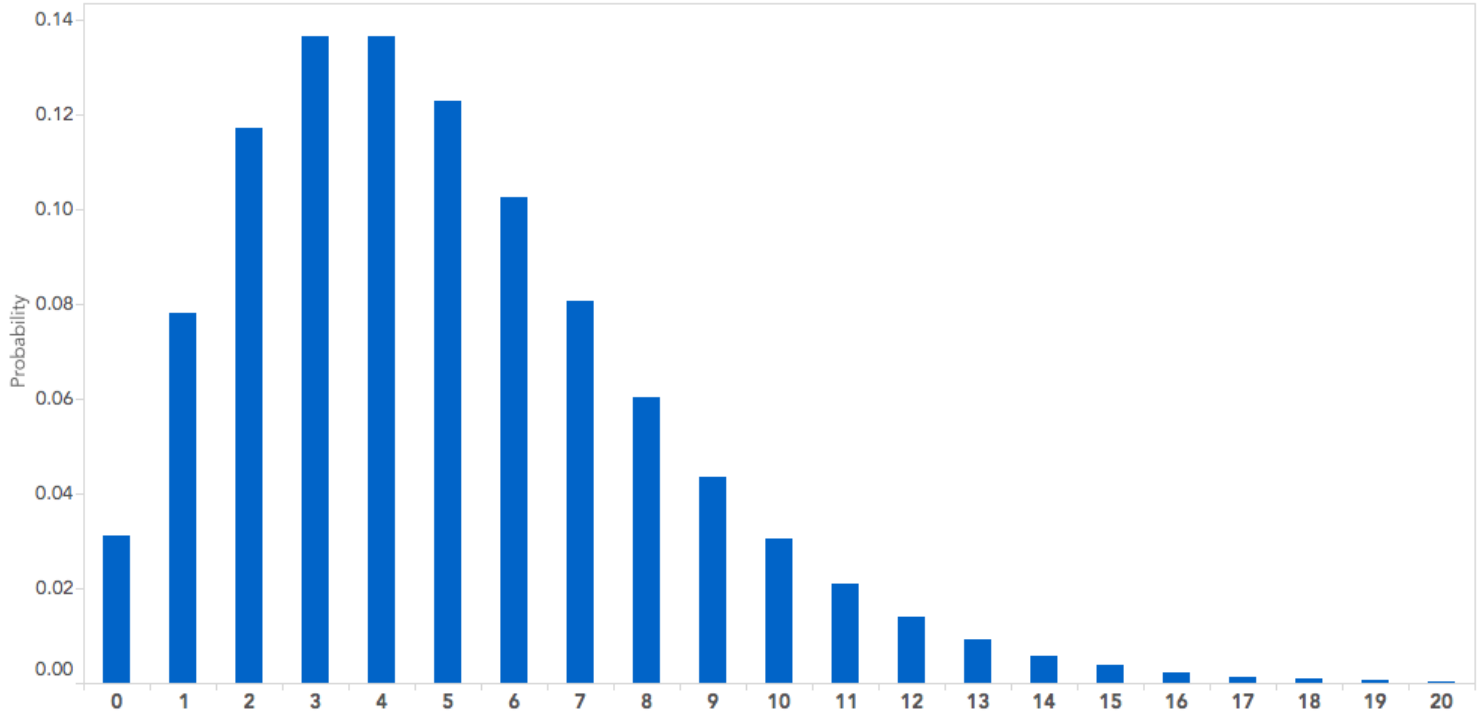
$$\sum_{i,j} \left(x_{ij} - \langle u_i, v_j \rangle\right)^2 \to min$$

What distribution describes this data better?

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

Poisson distribution

$$X \sim Poiss(\lambda)$$



$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda} \qquad \mathbb{E}X = \lambda$$

Poisson distribution and matrix factorization

$$x_{ij} \sim Poiss(\langle u_i, v_j \rangle) \qquad P(x_{ij}) = \frac{\langle u_i, v_j \rangle^{x_{ij}}}{x_{ij}!} e^{-\langle u_i, v_j \rangle}$$

$$\prod_{i,j} \frac{\langle u_i, v_j \rangle^{x_{ij}}}{x_{ij}!} e^{-\langle u_i, v_j \rangle} \to max$$

$$\sum_{i,j} \langle u_i, v_j \rangle - x_{ij} \ln \langle u_i, v_j \rangle + \ln x_{ij}! \to min$$

$$\sum_{i,j} \langle u_i, v_j \rangle - x_{ij} \ln \langle u_i, v_j \rangle \to min$$

SGD for NMF (Non-negative matrix factorization)

$$Q = \sum_{i,j} \langle u_i, v_j \rangle - x_{ij} \ln \langle u_i, v_j \rangle \to min$$

$$\frac{\partial Q}{\partial u_i} = \sum_j v_j - \frac{x_{ij}}{\langle u_i, v_j \rangle} v_j = \sum_j \underbrace{\frac{\langle u_i, v_j \rangle - x_{ij}}{\langle u_i, v_j \rangle}} v_j \to min$$

$\tilde{\varepsilon}_{ij}$ - relative error

SGD:

$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \tilde{\varepsilon}_{ij} v_j$$

$$v_j^{(t+1)} = v_j^{(t)} - \eta_t \tilde{\varepsilon}_{ij} u_i$$

Another non-negative factorizations

One can use for NMF Frobenius norm with restrictions on U and V:

$$Q = \sum_{i,j} \left(\langle u_i, v_j \rangle - x_{ij} \right)^2 \to \min_{\substack{u_i, v_j: \\ u_{ik} \geq 0 \\ v_{jk} \geq 0}}$$

Recap

- Matrix factorization
- SGD and ALS
- Implicit matrix factorizations
- Probabilistic interpretation