

2. Преобразование признаков

Преобразование признаков

1. Задача понижения размерности
2. Метод главных компонент и SVD
3. Manifold learning

Как выглядит обучающая выборка

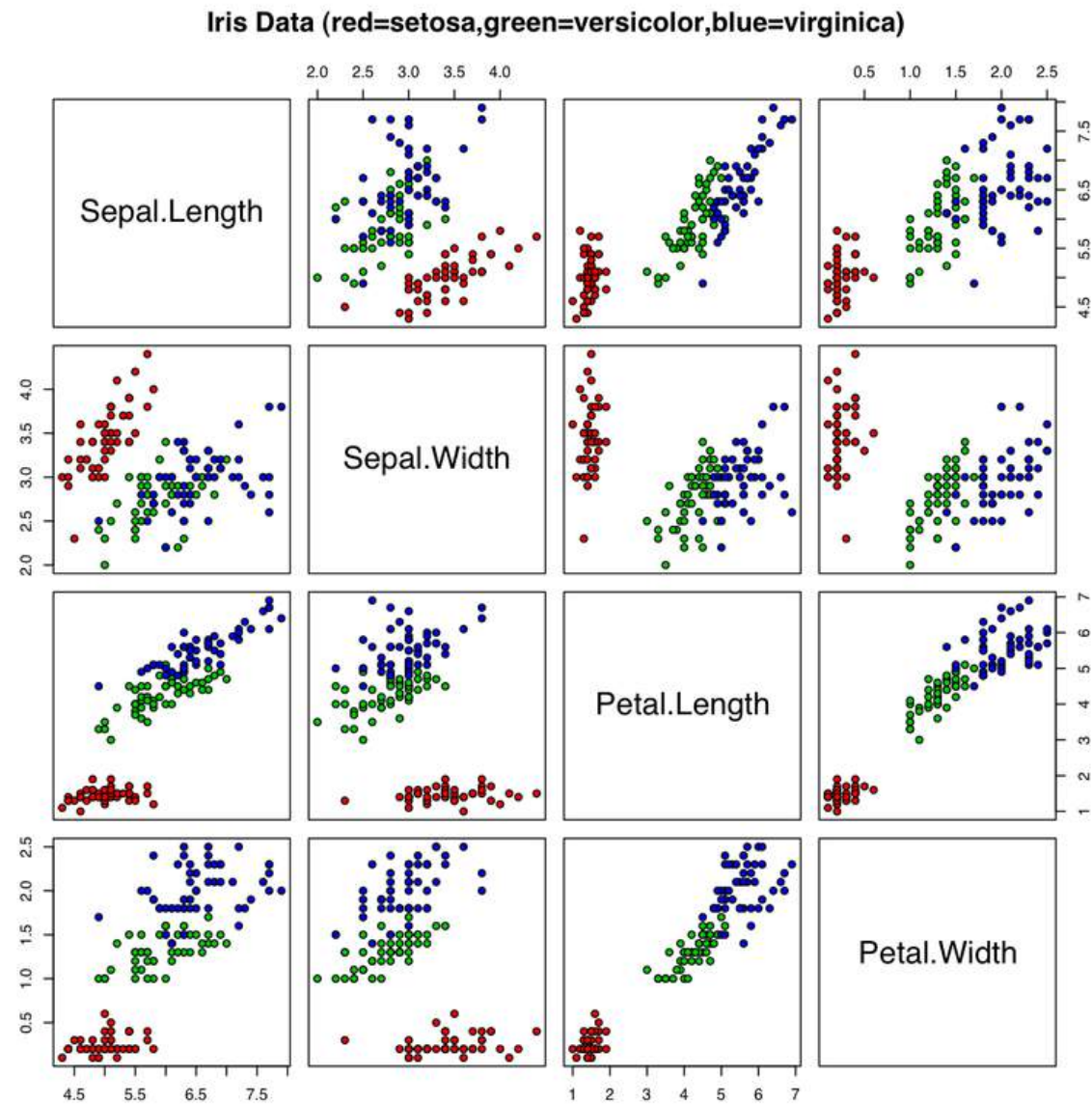
Fisher's Iris Data

Sepal length ⇅	Sepal width ▲	Petal length ⇅	Petal width ⇅	Species ⇅
5.0	2.0	3.5	1.0	<i>I. versicolor</i>
6.0	2.2	5.0	1.5	<i>I. virginica</i>
6.2	2.2	4.5	1.5	<i>I. versicolor</i>
6.0	2.2	4.0	1.0	<i>I. versicolor</i>
6.3	2.3	4.4	1.3	<i>I. versicolor</i>
5.5	2.3	4.0	1.3	<i>I. versicolor</i>
5.0	2.3	3.3	1.0	<i>I. versicolor</i>
4.5	2.3	1.3	0.3	<i>I. setosa</i>
5.5	2.4	3.8	1.1	<i>I. versicolor</i>
5.5	2.4	3.7	1.0	<i>I. versicolor</i>
4.9	2.4	3.3	1.0	<i>I. versicolor</i>
6.7	2.5	5.8	1.8	<i>I. virginica</i>
5.7	2.5	5.0	2.0	<i>I. virginica</i>
6.3	2.5	5.0	1.9	<i>I. virginica</i>
6.3	2.5	4.9	1.5	<i>I. versicolor</i>
4.9	2.5	4.5	1.7	<i>I. virginica</i>

Что хотелось бы уметь

- Визуализировать обучающую выборку, когда признаков больше трёх
- Уменьшать количество признаков, переходя к новым, более информативным

Визуализируем выборку

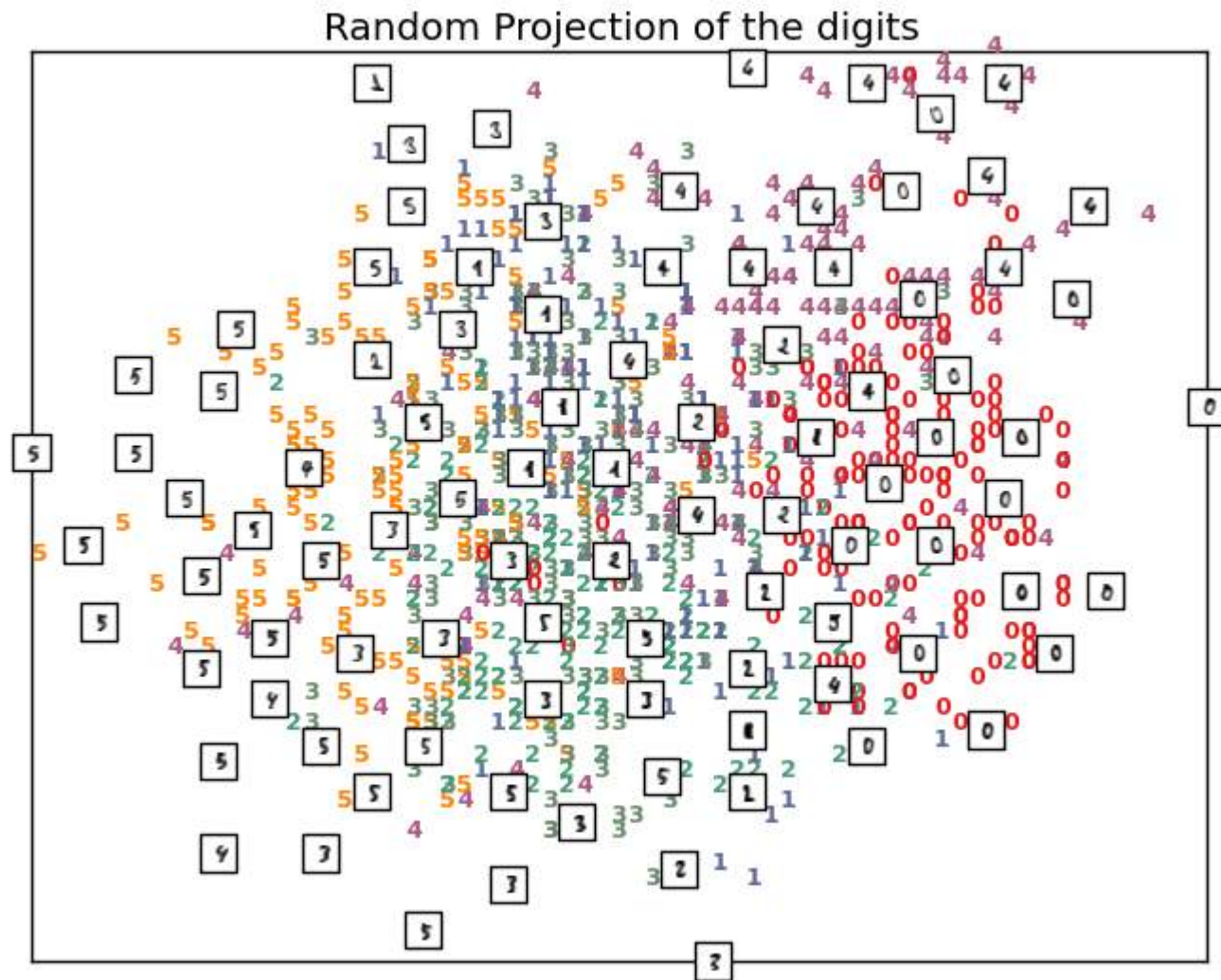


Более сложный случай

Что делать, если признаков еще больше?

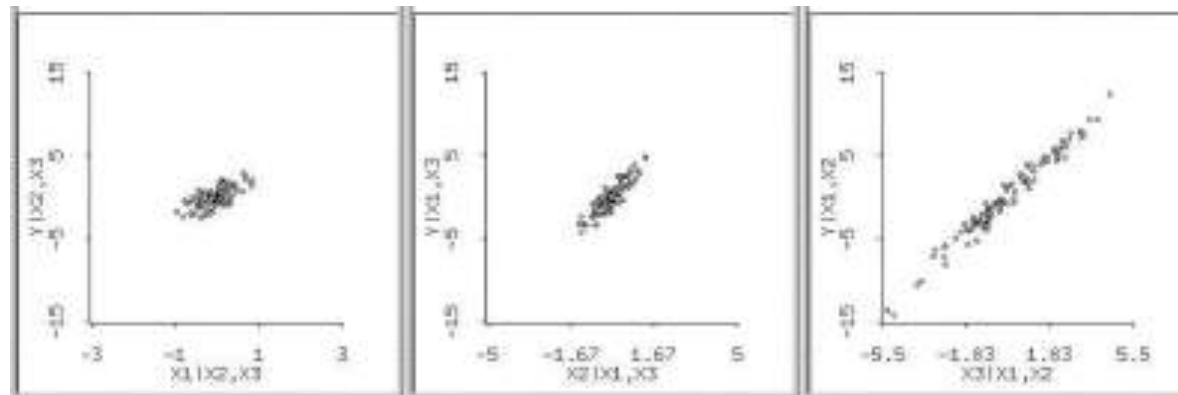
1	6	4	12	5	5	3	4	1	67	3	2	1	2	1	0	0	1	0	0	1	0	0	1	1
2	48	2	60	1	3	2	2	1	22	3	1	1	1	1	0	0	1	0	0	1	0	0	1	2
4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	1	0	0	1	0	1	0	1
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	1	1
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	1	0	0	0	0	0	1	2
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	1	0	0	0	0	1	0	1
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	1	0	0	1	0	0	1	1
2	36	2	69	1	3	3	2	3	35	3	1	1	2	1	0	1	1	0	1	0	0	0	0	1
4	12	2	31	4	4	1	4	1	61	3	1	1	1	1	0	0	1	0	0	1	0	1	0	1
2	30	4	52	1	1	4	2	3	28	3	2	1	1	1	1	0	1	0	0	1	0	0	0	2
2	12	2	13	1	2	2	1	3	25	3	1	1	1	1	1	0	1	0	1	0	0	0	1	2
1	48	2	43	1	2	2	4	2	24	3	1	1	1	1	0	0	1	0	1	0	0	0	1	2
2	12	2	16	1	3	2	1	3	22	3	1	1	2	1	0	0	1	0	0	1	0	0	1	1
1	24	4	12	1	5	3	4	3	60	3	2	1	1	1	1	0	1	0	0	1	0	1	0	2
1	15	2	14	1	3	2	4	3	28	3	1	1	1	1	1	0	1	0	1	0	0	0	1	1
1	24	2	13	2	3	2	2	3	32	3	1	1	1	1	0	0	1	0	0	1	0	1	0	2
4	24	4	24	5	5	3	4	2	53	3	2	1	1	1	0	0	1	0	0	1	0	0	1	1
1	30	0	81	5	2	3	3	3	25	1	3	1	1	1	0	0	1	0	0	1	0	0	1	1
2	24	2	126	1	5	2	2	4	44	3	1	1	2	1	0	1	1	0	0	0	0	0	0	2
4	24	2	34	3	5	3	2	3	31	3	1	2	2	1	0	0	1	0	0	1	0	0	1	1
4	9	4	21	1	3	3	4	3	48	3	3	1	2	1	1	0	1	0	0	1	0	0	1	1
1	6	2	26	3	3	3	3	1	44	3	1	2	1	1	0	0	1	0	1	0	0	0	1	1
1	10	4	22	1	2	3	3	1	48	3	2	2	1	2	1	0	1	0	1	0	0	1	0	1
2	12	4	18	2	2	3	4	2	44	3	1	1	1	1	0	1	1	0	0	1	0	0	1	1
4	10	4	21	5	3	4	1	3	26	3	2	1	1	2	0	0	1	0	0	1	0	0	1	1
1	6	2	14	1	3	3	2	1	36	1	1	1	2	1	0	0	1	0	0	1	0	1	0	1
4	6	0	4	1	5	4	4	3	39	3	1	1	1	1	0	0	1	0	0	1	0	1	0	1
3	12	1	4	4	3	2	3	1	42	3	2	1	1	1	0	0	1	0	1	0	0	0	1	1
2	5	2	24	1	3	2	2	2	24	3	1	1	1	1	0	0	0	0	0	1	0	0	1	1

Случайная проекция для рукописных цифр



Проблемы «лишних признаков»

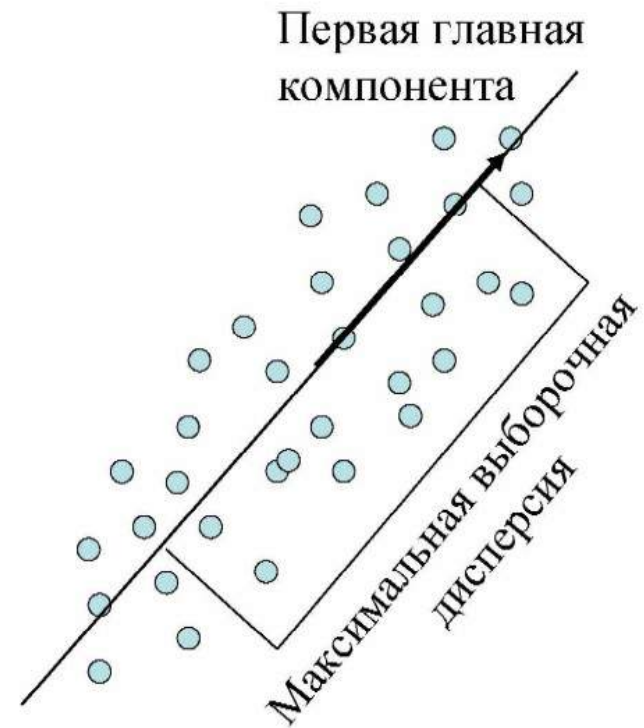
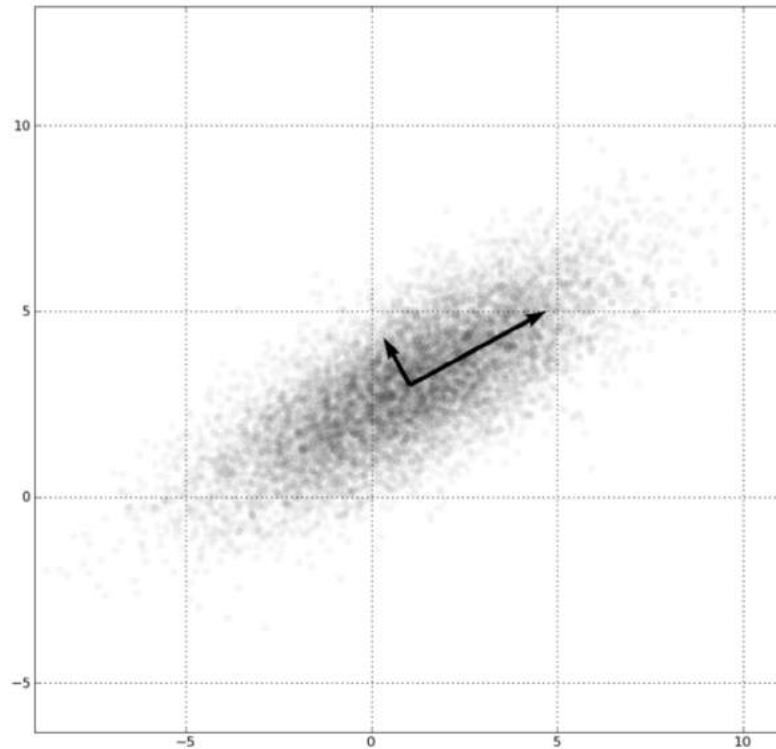
Если признаки сильно коррелированы, то у многих методов машинного обучения будут проблемы (например, из-за неустойчивости обращения матрицы ковариаций, где это нужно)



Principal Component Analysis 1

Идея 1: давайте выделять в пространстве признаков направления, вдоль которых разброс точек наибольший (они кажутся наиболее информативными)

РСА (интерпретация 1)

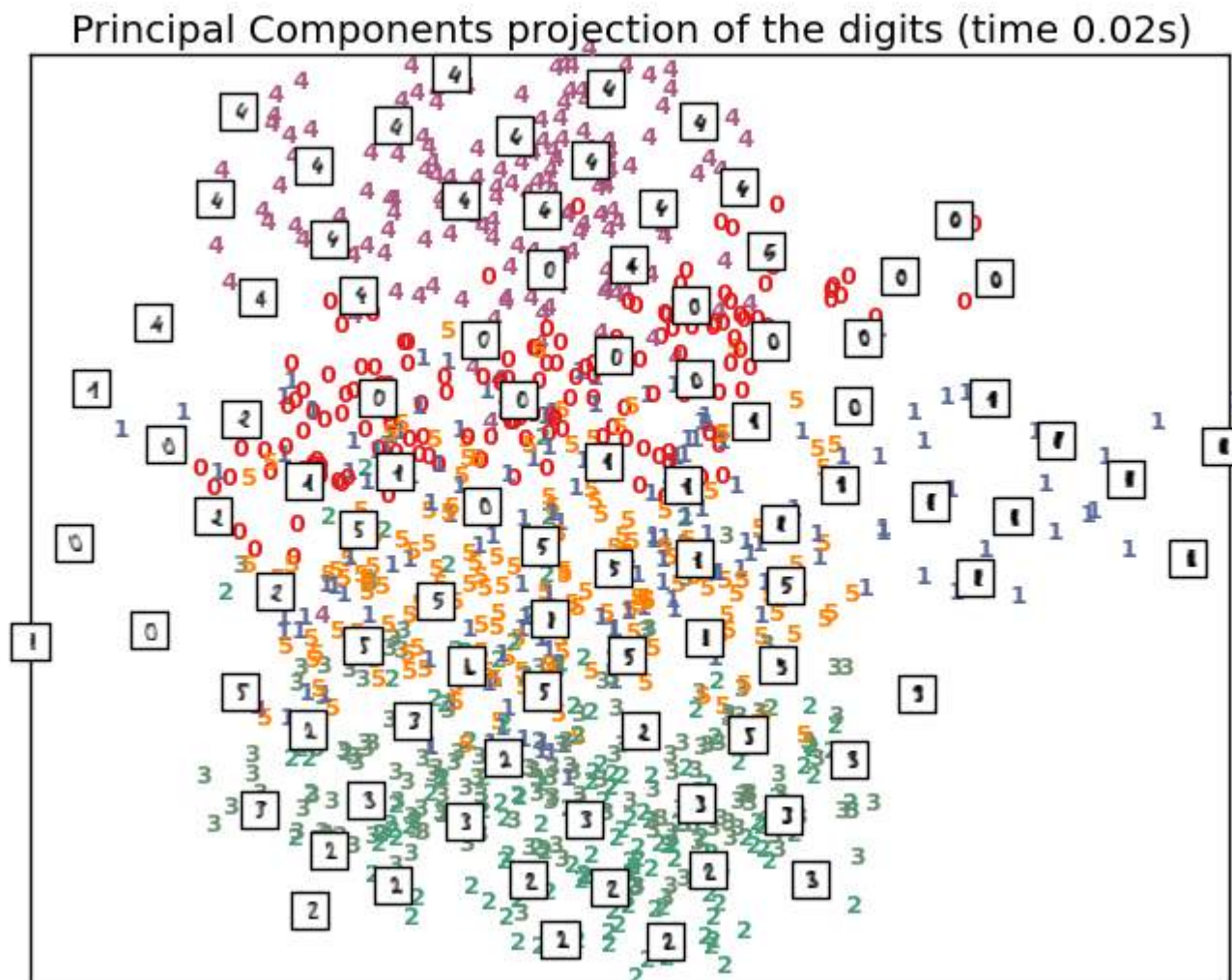


Пример: eigenfaces




$$= \text{mean} + 0.9 * \img alt="Eigenface 1: A grayscale image showing the first principal component of face variation, primarily capturing the left-right asymmetry." data-bbox="382 742 487 930"/> - 0.2 * \img alt="Eigenface 2: A grayscale image showing the second principal component of face variation, primarily capturing the top-bottom asymmetry." data-bbox="584 742 689 930"/> + 0.4 * \img alt="Eigenface 3: A grayscale image showing the third principal component of face variation, primarily capturing the nose and mouth area." data-bbox="796 742 901 930"/> + ...$$

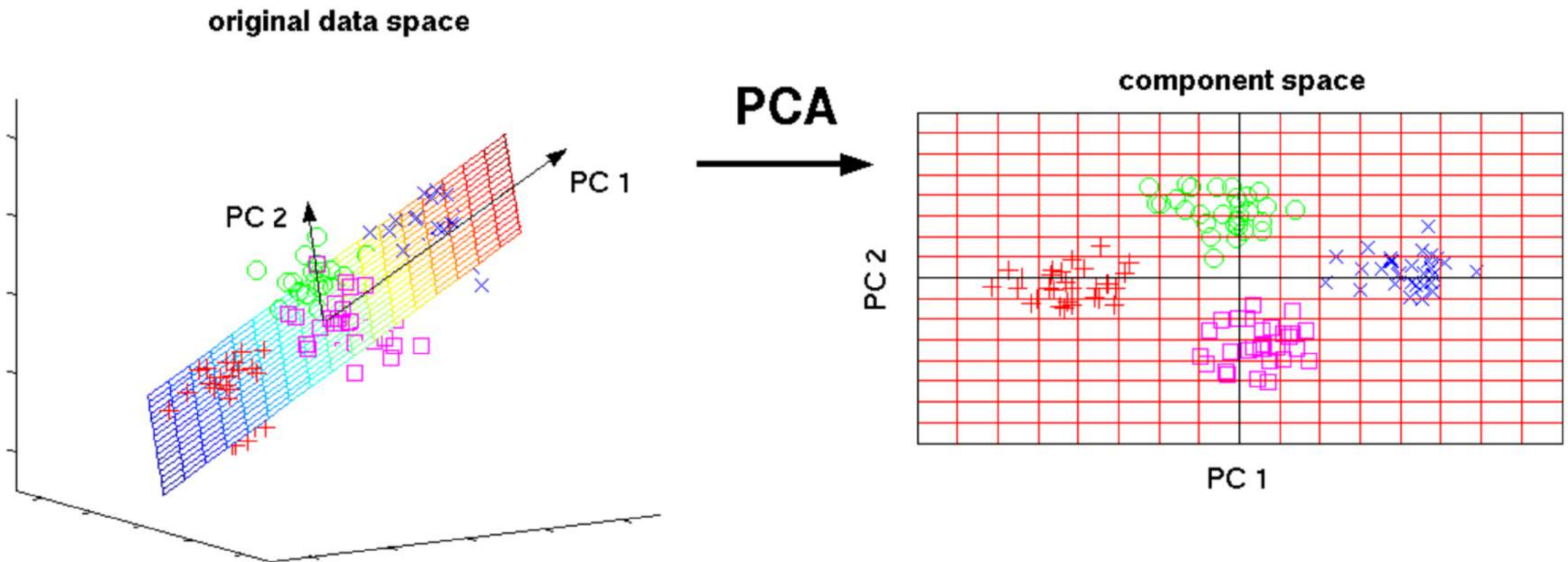
Рукописные цифры: проекция на главные компоненты



РСА (интерпретация 2)

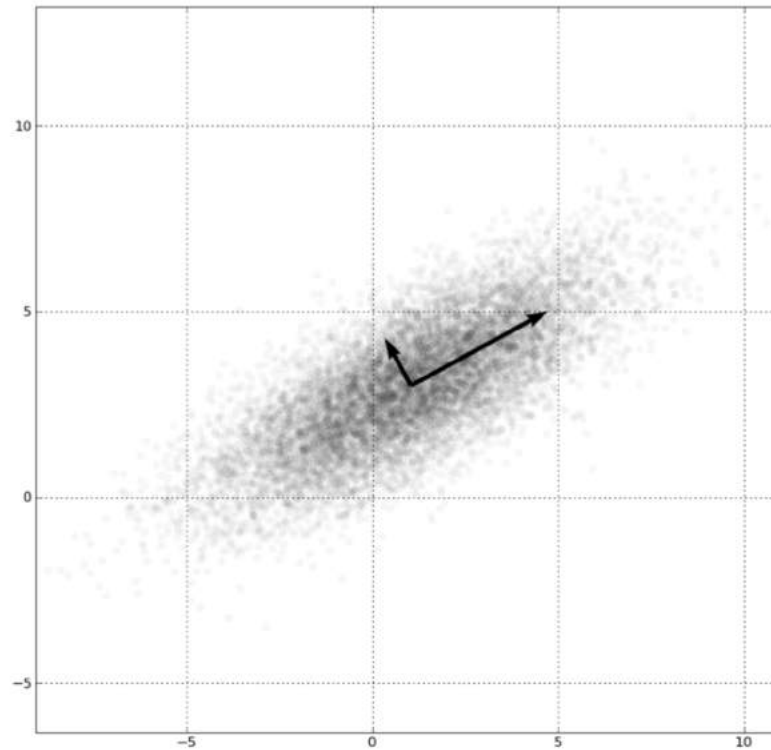
Идея 2: давайте строить проекцию выборки на линейное подпространство меньшей размерности. А выбирать его так, чтобы квадраты отклонений точек от проекций были минимальны.

РСА (интерпретация 2)



РСА (интерпретация 3)

Идея 3. Переход в базис, в котором матрица ковариаций диагональна



РСА (интерпретация 4)

Приблизим исходную матрицу признаков произведением двух матриц:

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

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

РСА: как сделать?

- Центрируем выборку (из каждого признака вычитаем среднее значение), получаем матрицу X с новыми значениями признаков
- Делаем SVD-разложение матрицы X :

$$X \approx A \cdot \Lambda \cdot B^T$$

Выбираем $U = A \cdot \Lambda$, $V = B$

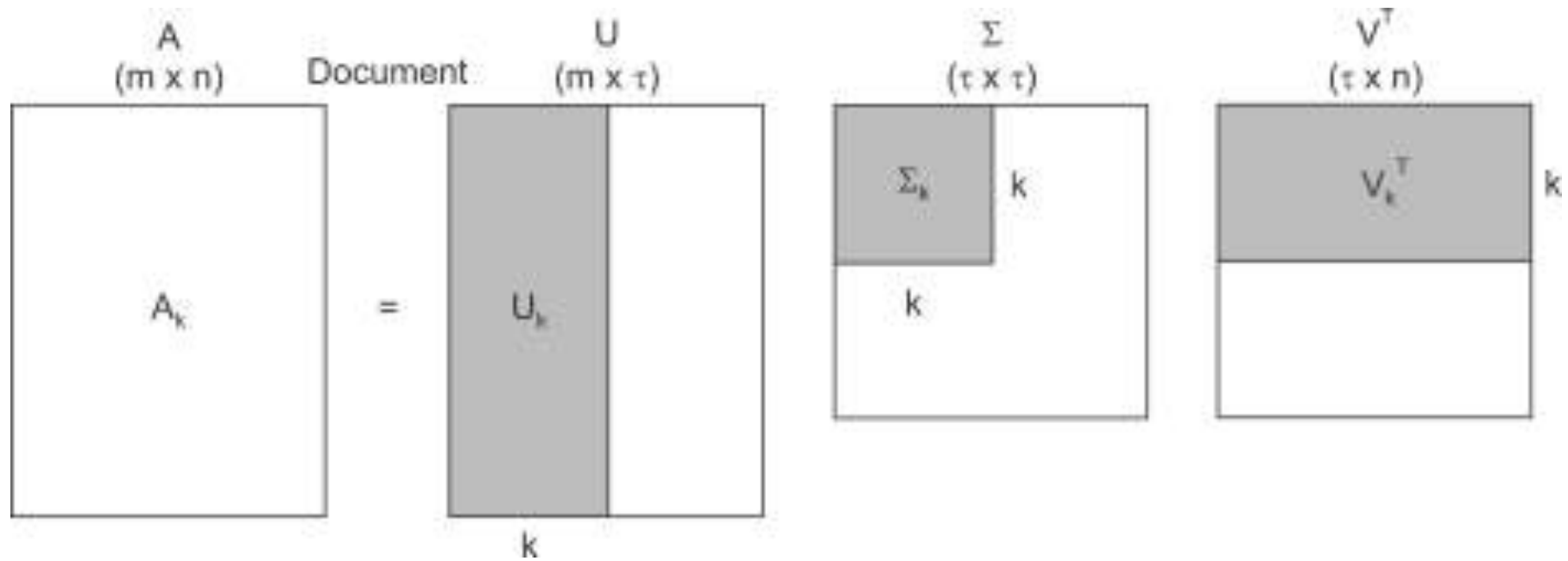
SVD

SVD = Singular Vector Decomposition (сингулярное разложение матриц)

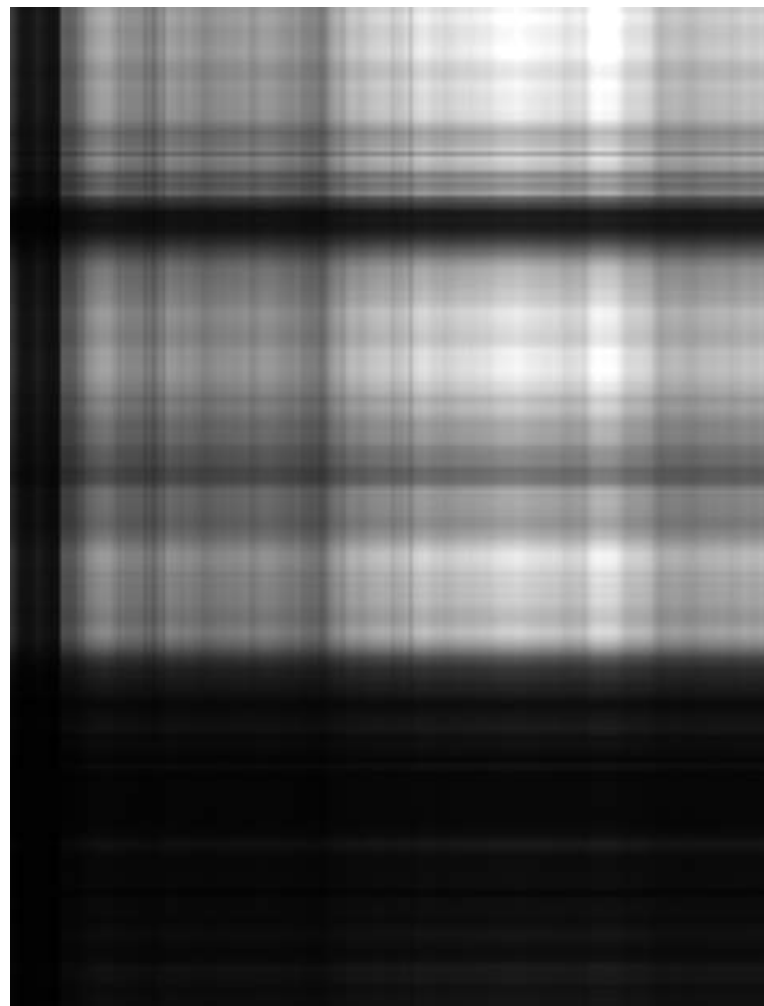
Позволяет получить наилучшее приближение исходной матрицы X матрицей X' ранга k .

Применяется для снижения размерности пространства признаков.

SVD



SVD: пример



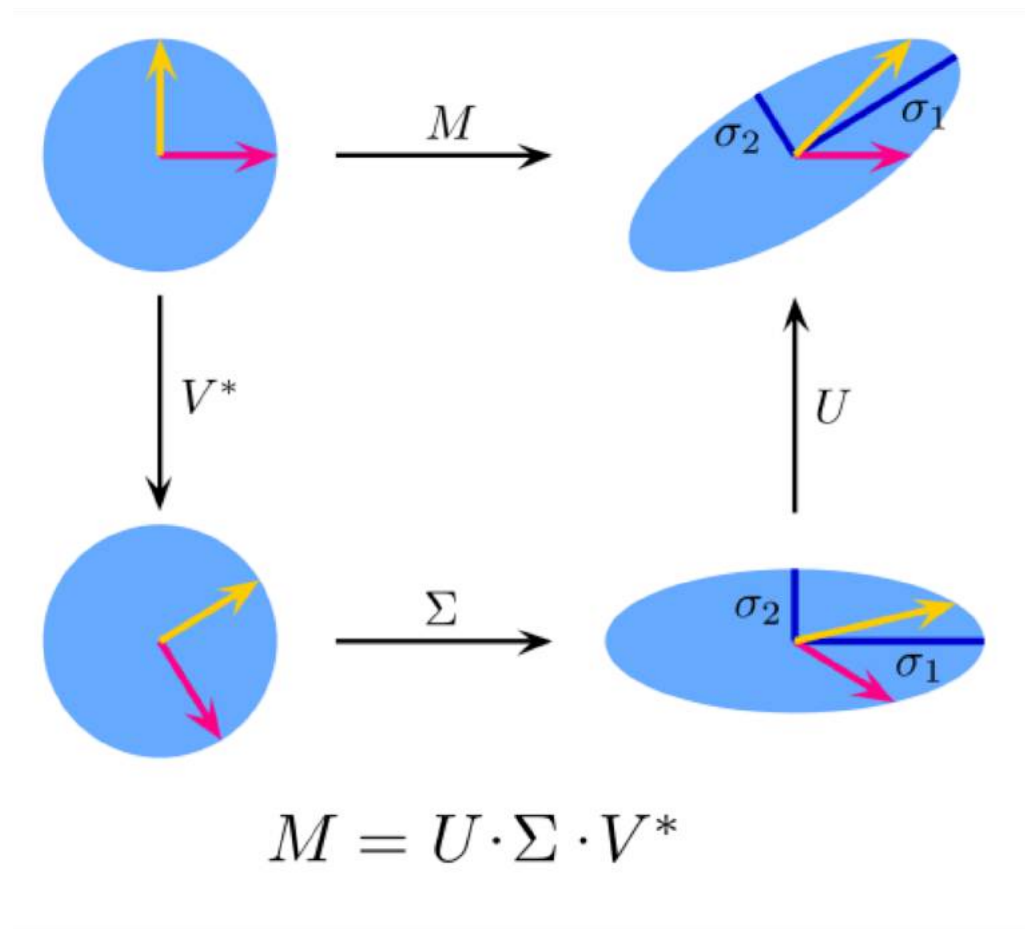
SVD: пример



SVD: пример

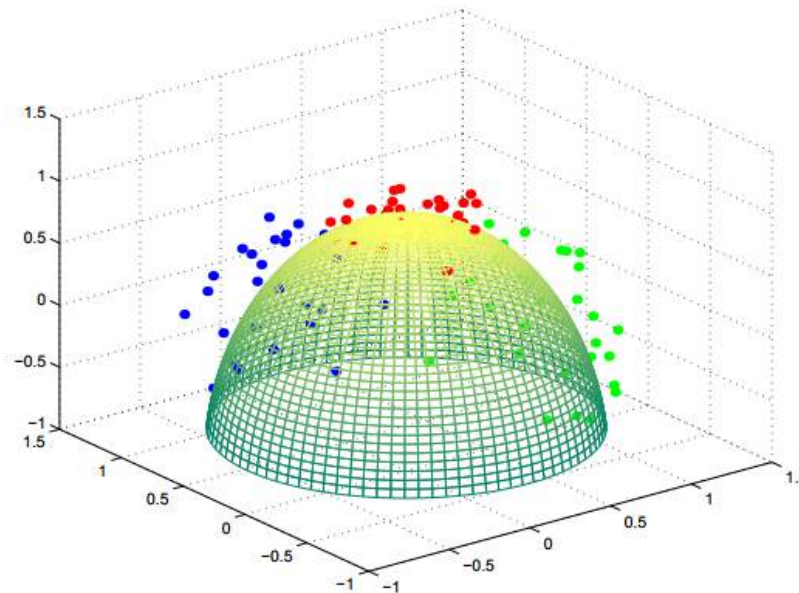


Геометрический смысл SVD

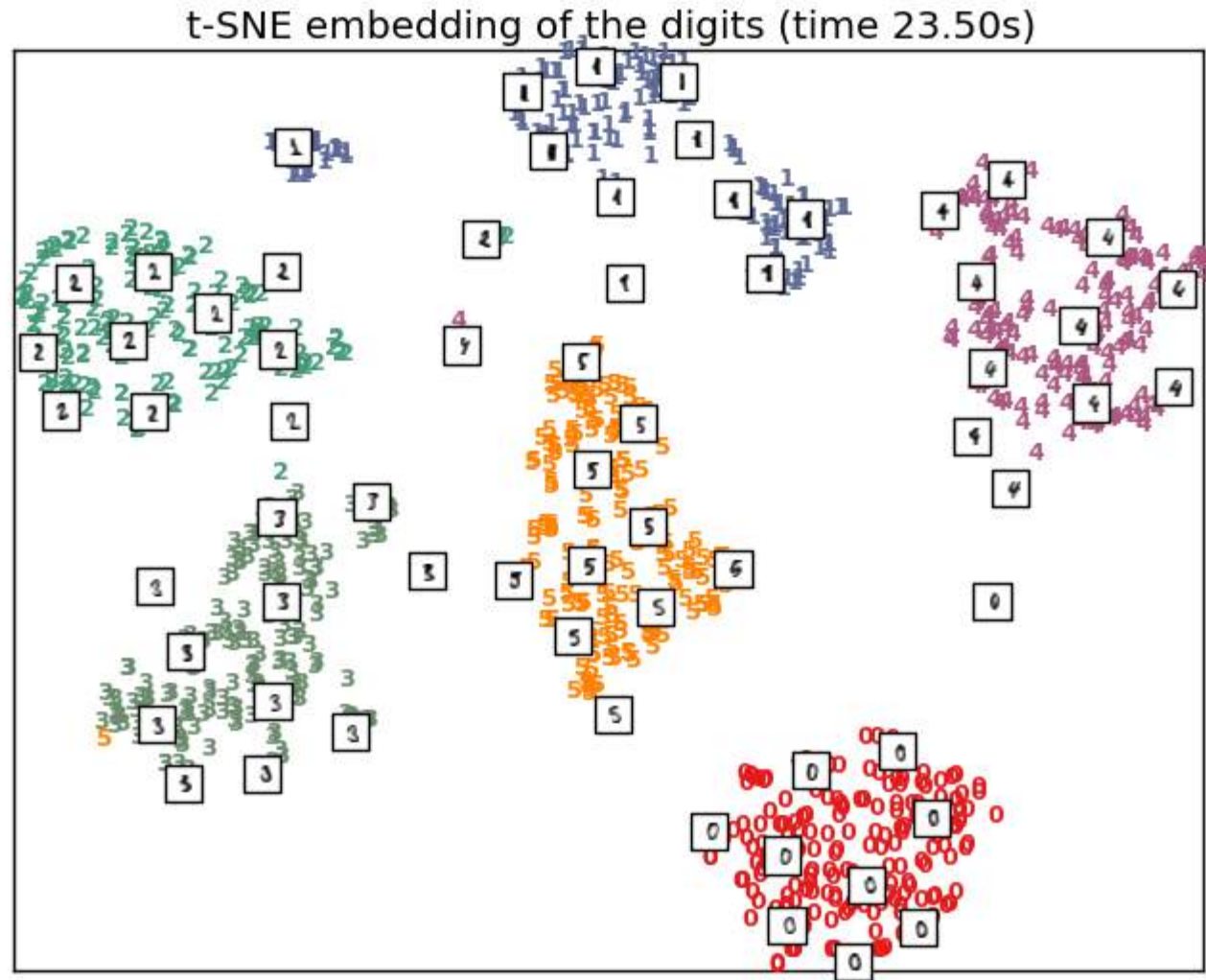


А что, если линейных преобразований признаков мало?

- Идея 1: объекты могут лежать в пространстве признаков на поверхности малой размерности.
- Идея 2: эта поверхность может быть нелинейной.



Manifold learning: t-SNE



3. Матричные разложения

Матрица рейтингов

	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

Матрица рейтингов

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

Матрица рейтингов

	<i>j</i>			
	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

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

u_i - «интересы пользователей»

v_j - «параметры фильмов»

"SVD" в машинном обучении

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

u_i - «профили» объектов

v_j - «профили» исходных признаков

Матрица частот слов и 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

Матрица частот слов и SVD

j

	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
i 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

Матрица частот слов и SVD

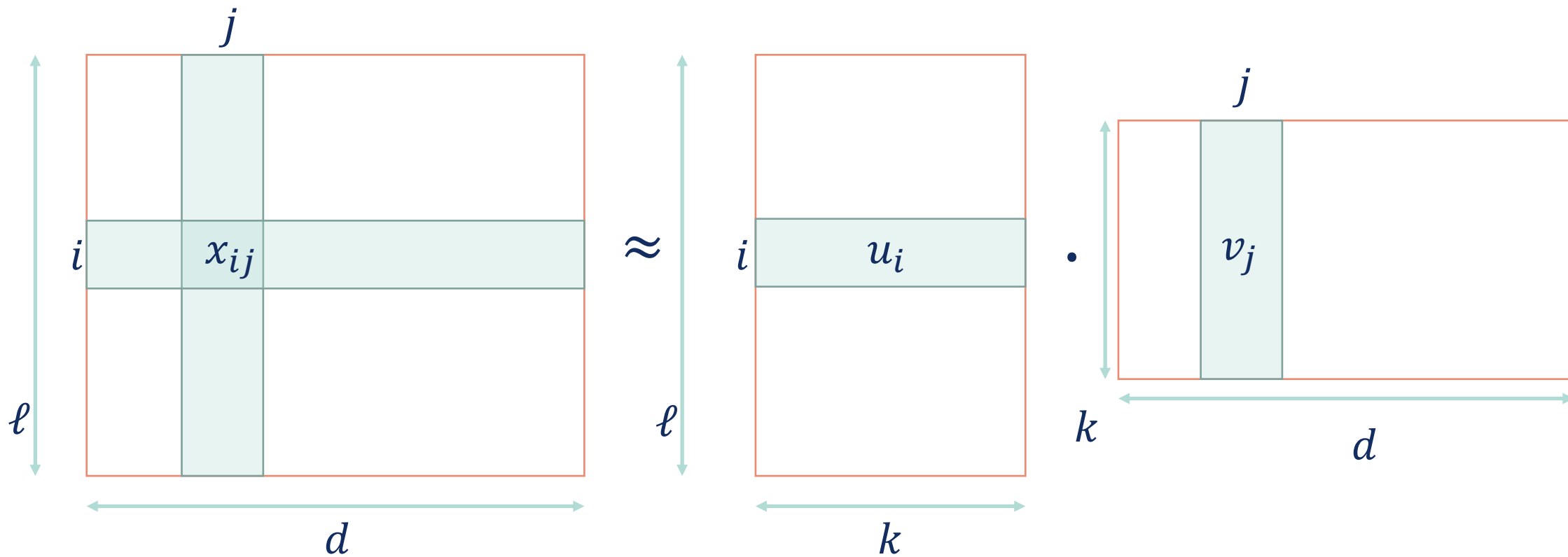
	<i>j</i>					
	database	SQL	index	regression	likelihood	linear
<i>i</i>	d1	24	21	9	0	3
	d2	32	10	5	0	3
	d3	12	16	5	0	0
	d4	6	7	2	0	0
	d5	43	31	20	0	3
	d6	2	0	0	18	7
	d7	0	0	1	32	12
	d8	3	0	0	22	4
	d9	1	0	0	34	27
	d10	6	0	0	17	4

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

u_i - «темы» документов

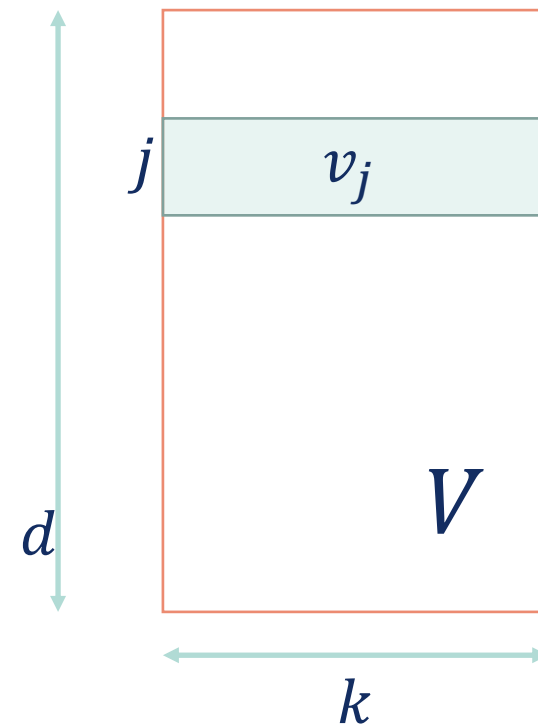
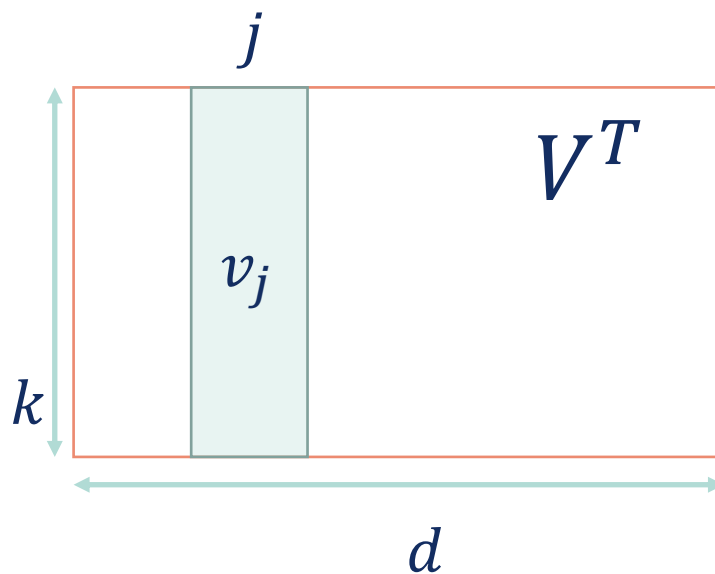
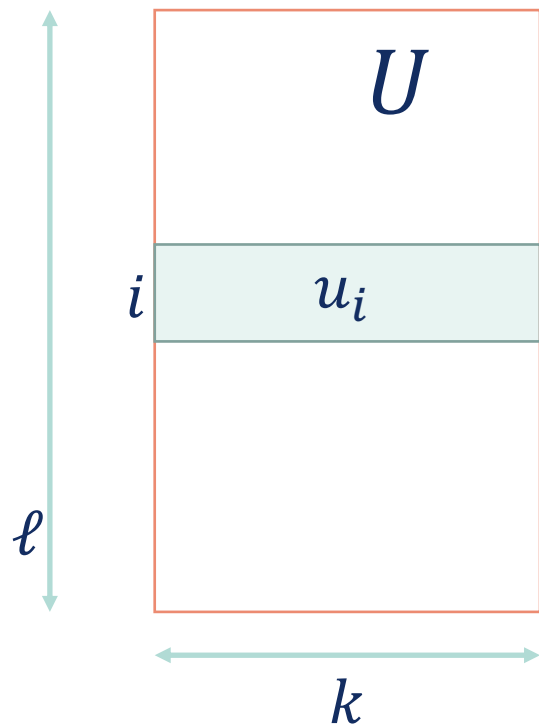
v_j - «темы» слов

Еще немного про обозначения



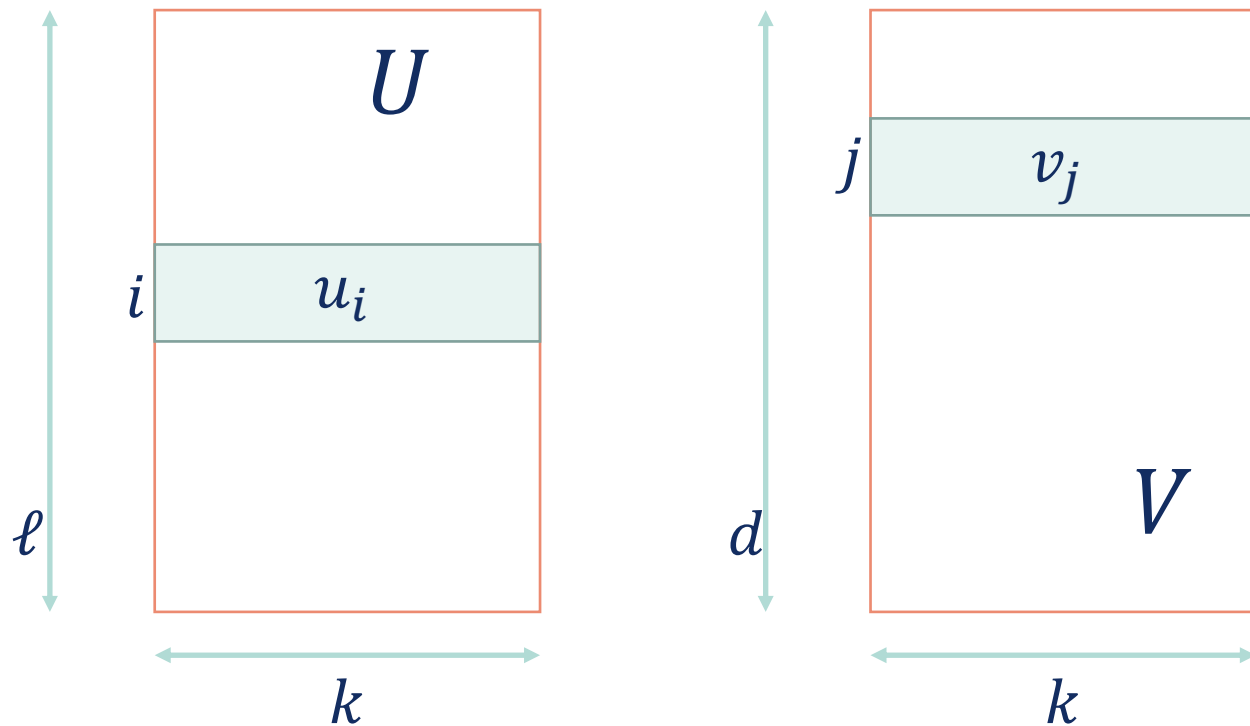
$$x_{ij} \approx \langle u_i, v_j \rangle$$

Еще немного про обозначения



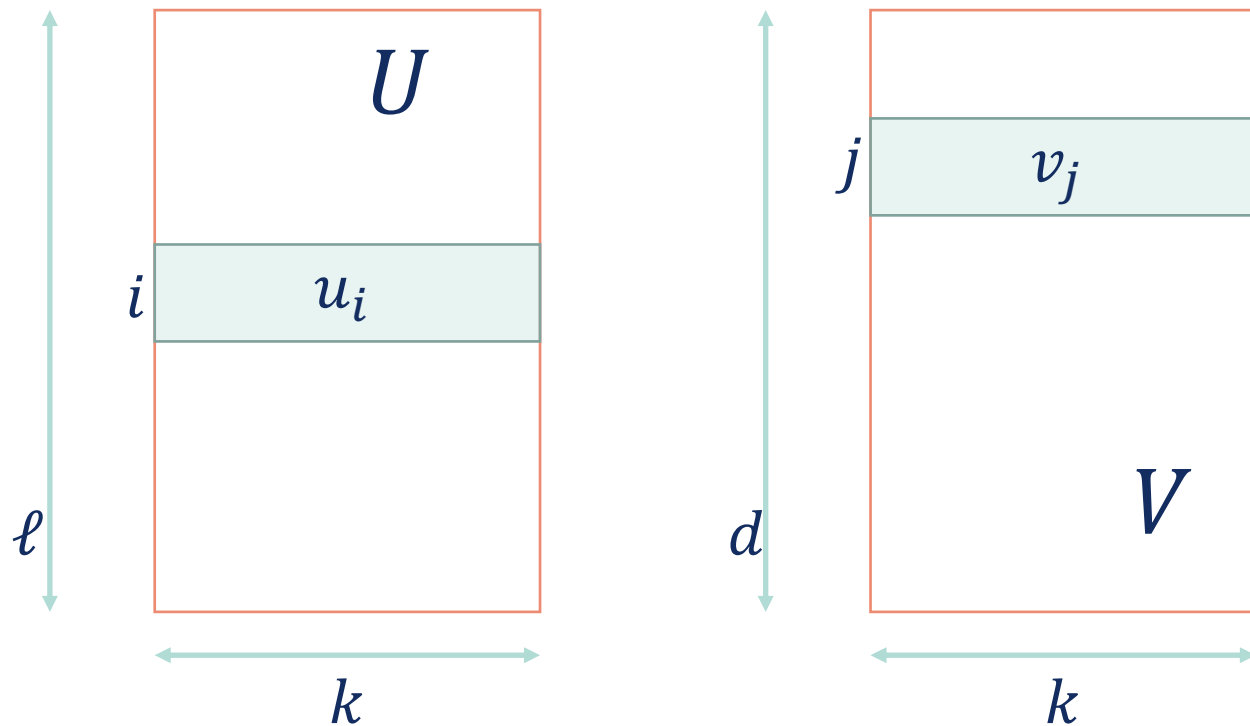
$$x_{ij} \approx \langle u_i, v_j \rangle$$

Еще немного про обозначения



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

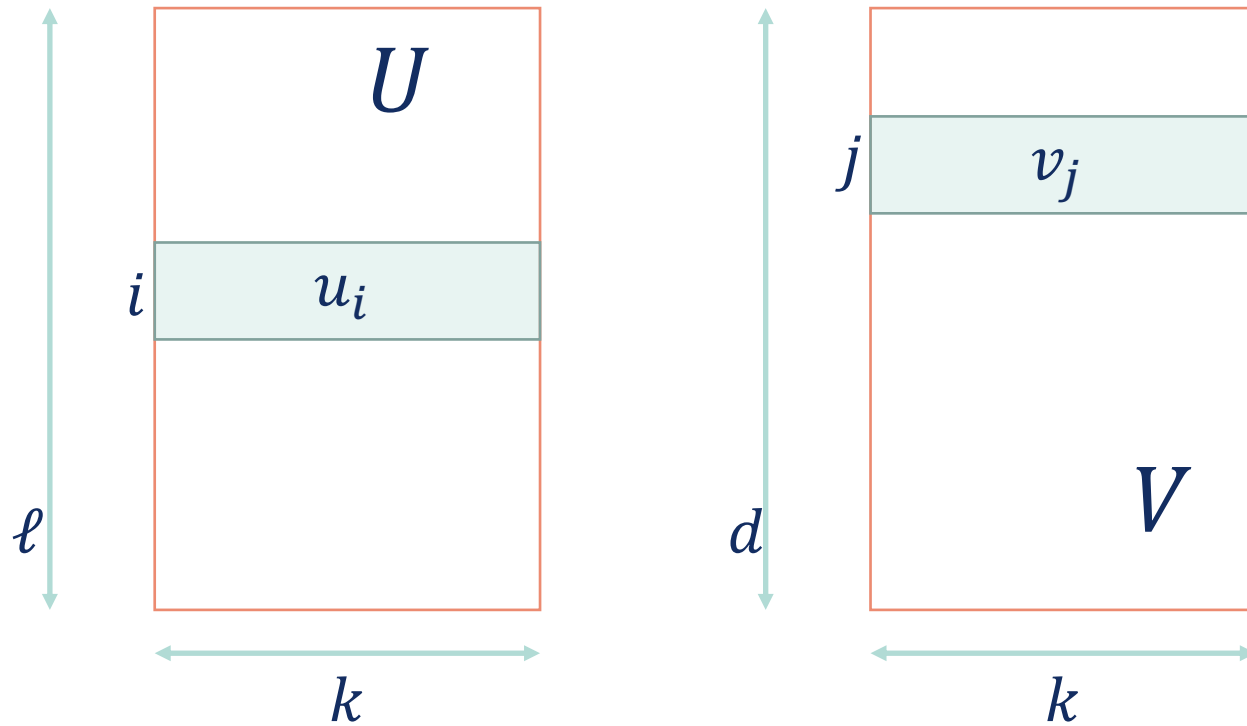
Еще немного про обозначения



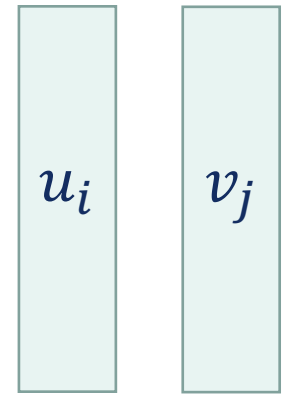
$$x_{ij} \approx \langle u_i, v_j \rangle$$

Below the equation, two vertical light green bars represent the vectors u_i and v_j , with labels u_i and v_j placed inside them.

Еще немного про обозначения

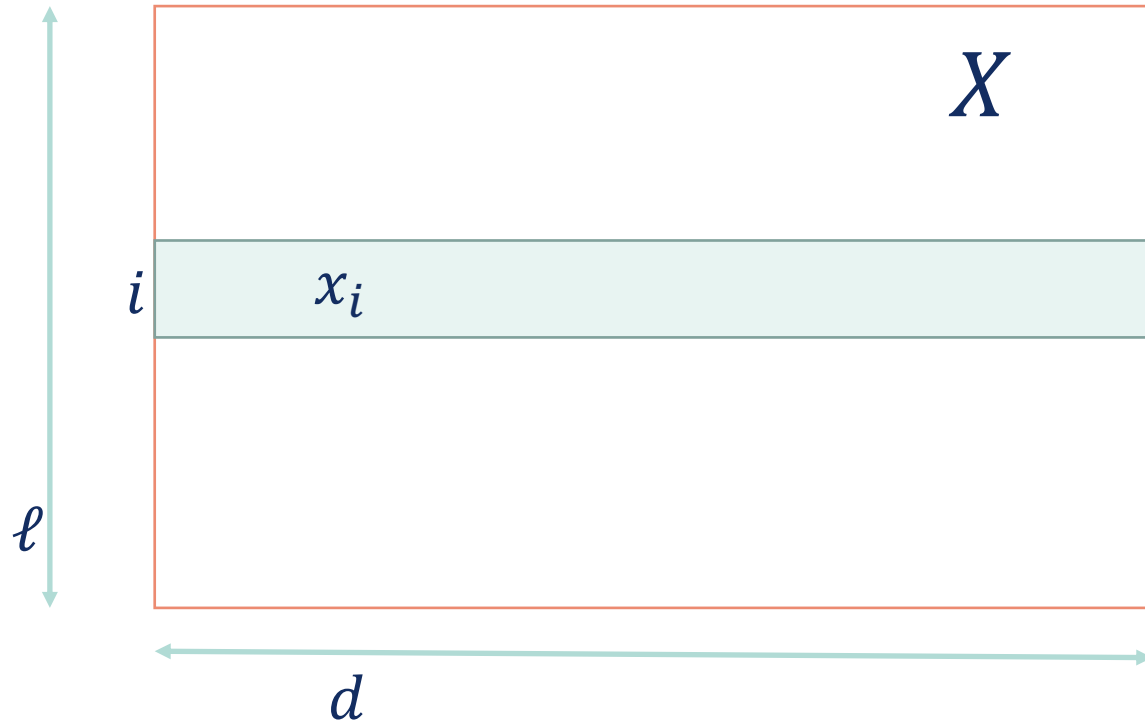


$$x_{ij} \approx \langle u_i, v_j \rangle$$



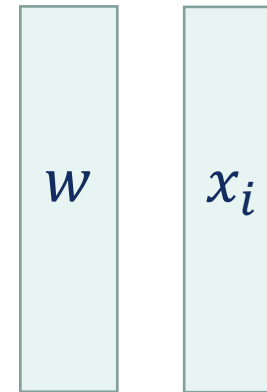
$$x_{ij} \approx u_i^T v_j$$

Аналогия с матрицей признаков



В линейных моделях:

$$\langle w, x_i \rangle = w^T x_i$$



Какие обозначения встречаются

$$X \approx UV^T$$

$$X \approx PQ^T$$

$$X \approx WH$$

$$X \approx \Phi\Theta$$

Оптимизационная задача

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

Градиентный спуск (GD)

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

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

$$= \sum_j 2(\langle u_i, v_j \rangle - x_{ij}) v_j \quad \varepsilon_{ij} = (\langle u_i, v_j \rangle - x_{ij}) - \text{ошибка на } x_{ij}$$

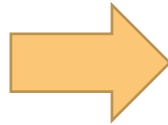
$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \sum_j \varepsilon_{ij} v_j$$

Стохастический градиентный спуск (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:

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

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

Для случайных i, j

Плюсы и минусы SGD

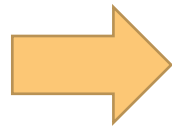
- + Простота реализации
- + Сходимость
- Медленно сходится
- Сложность выбора шага градиентного спуска (γ_t и η_t)
- При константном шаге сходится очень медленно

Идея ALS

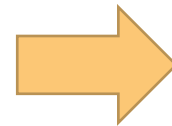
$$Q \rightarrow \min_{u_i, v_j}$$

Повторяем до сходимости:

$$\frac{\partial Q}{\partial u_i} = 0$$


$$u_i$$

$$\frac{\partial Q}{\partial v_j} = 0$$


$$v_j$$

Выписываем шаг в ALS

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

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

$$\sum_j v_j v_j^T u_i = \sum_j x_{ij} v_j$$
$$\underbrace{\left(\sum_j v_j v_j^T \right)}_A u_i = \underbrace{\sum_j x_{ij} v_j}_b$$

ALS: итоговый алгоритм

Повторяем по случайным i, j до сходимости:

$$\left(\sum_j v_j v_j^T \right) u_i = \sum_j x_{ij} v_j \quad \Rightarrow \quad u_i \quad \text{(решение системы линейных уравнений)}$$

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

Регуляризация

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

α и β - небольшие положительные числа (0.001, 0.01, 0.05)

Модель прогнозирования

		<i>j</i>			
		Пила	Улица Вязов	Ванильное небо	1+1
<i>i</i>	Маша	5	4	1	2
	Юля	5	5	2	
	Вова			3	5
	Коля	3	?	4	5
	Петя				4
	Ваня		5	3	3

Модель прогнозирования

i	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

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

u_i - «интересы пользователей»

v_j - «параметры фильмов»

Отличия в задаче рекомендаций

j

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

Почему нужно что-то менять

j

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

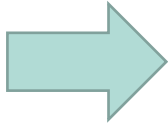
$$x_{ij} = 1 \approx \langle u_i, v_j \rangle$$

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

Почему нужно что-то менять

	j			
	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

$$x_{ij} = 1 \approx \langle u_i, v_j \rangle$$

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


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

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

Explicit и implicit

Explicit feedback


Есть положительные и отрицательные пример (например, низкие и высокие оценки фильмов, лайки и дислайки и т.д.)

Implicit feedback

Есть только положительные (покупки, просмотры, лайки) или только отрицательные примеры (дислайки)

Implicit matrix factorization

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



Сумма по всем индексам (не только по известным элементам матрицы)

w_{ij} принимает большие значения для $x_{ij} \neq 0$
и значительно меньшие для $x_{ij} = 0$

Популярный метод: Implicit ALS

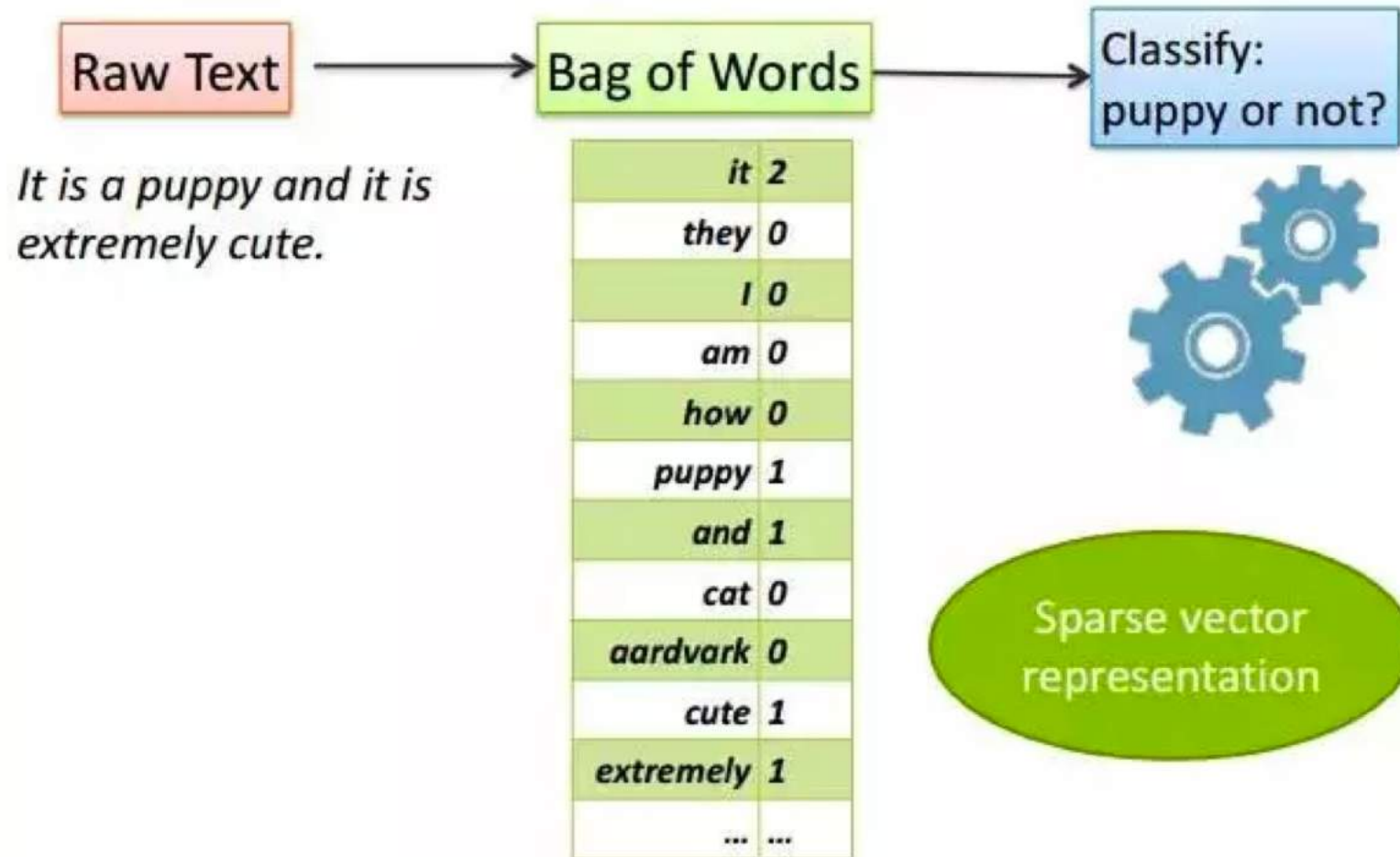
$$\sum_{i,j} w_{ij} (\langle u_i, v_j \rangle - x_{ij})^2 \rightarrow \min$$

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

u_i, v_j оцениваем с помощью ALS

4. Векторные представления (embeddings)

Bag of words



Проблемы WoW

- Огромное число признаков
- Семантически близкие «cute» и «lovely» не связаны
- Нужно много данных, чтобы выучить хорошую модель

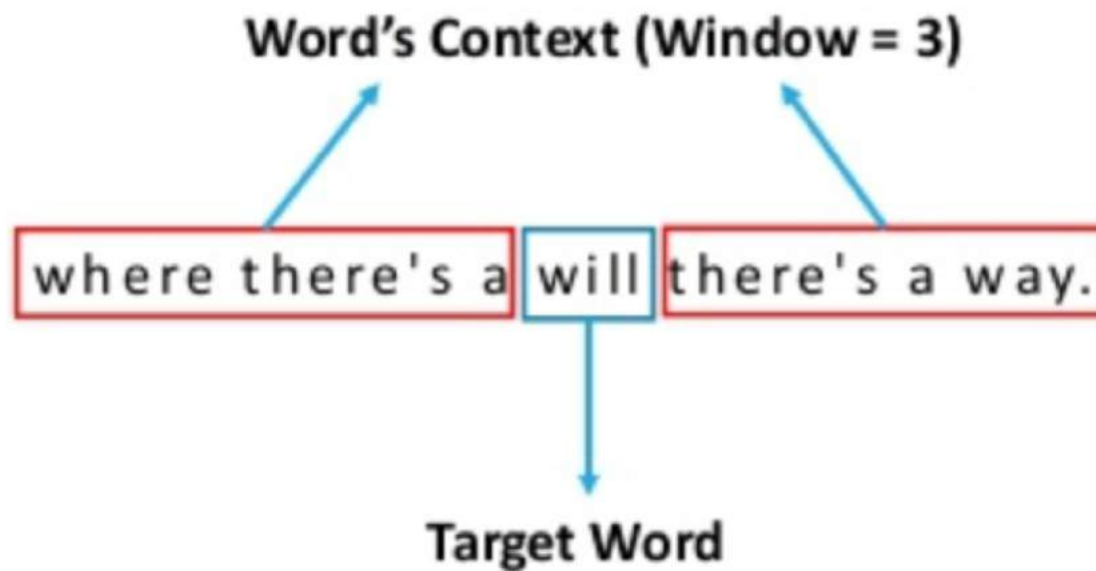
Похожие слова

- «cute» и «lovely» — синонимы
- Для компьютера это разные строки
- Как понять, что они похожи?

Похожие слова

- «cute» и «lovely» — синонимы
 - Для компьютера это разные строки
 - Как понять, что они похожи?
-
- На основе данных
 - Слова со схожим смыслом часто идут в паре с одними и теми же словами
 - У них похожие контексты

Дистрибутивная семантика



Term-context matrix

	C1	C2	C3	C4	C5	C6	C7
dog	5	0	11	2	2	9	1
cat	4	1	7	1	1	7	2
bread	0	12	0	0	9	1	9
pasta	0	8	1	2	14	0	10
meat	0	7	1	1	11	1	8
mouse	4	0	8	0	1	8	1

Term-context matrix

	dog	cat	computer	animal	mouse
dog	0	4	0	2	1
cat	4	0	0	3	5
computer	0	0	0	0	3
animal	2	3	0	0	2
mouse	1	5	3	2	0

Векторные представления слов

Хотим каждое слово представить как вещественный вектор:

$$w \rightarrow \vec{w} \in \mathbb{R}^d$$

Какие требования?

- Размерность d должна быть не очень велика
- Похожие слова должны иметь близкие векторы

word2vec

$$\sum_{i=1}^n \sum_{\substack{j=-k \\ j \neq 0}}^k \log p(w_{i+j} | w_i) \rightarrow \max,$$

где вероятность вычисляется через soft-max:

$$p(w_i | w_j) = \frac{\exp(\langle \vec{w}_i, \vec{w}_j \rangle)}{\sum_w \exp(\langle \vec{w}, \vec{w}_j \rangle)}$$

(сумма в знаменателе — по всем словам из словаря)

word2vec

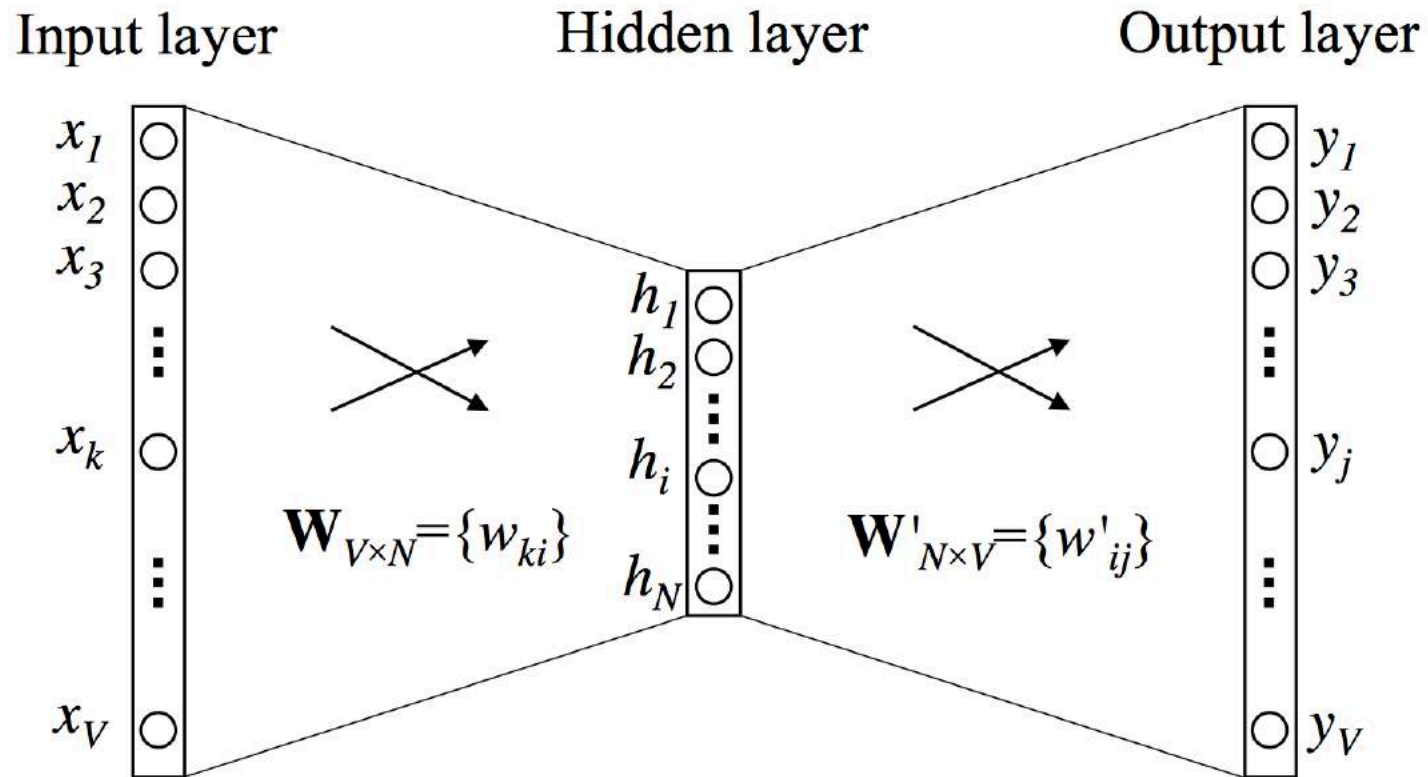
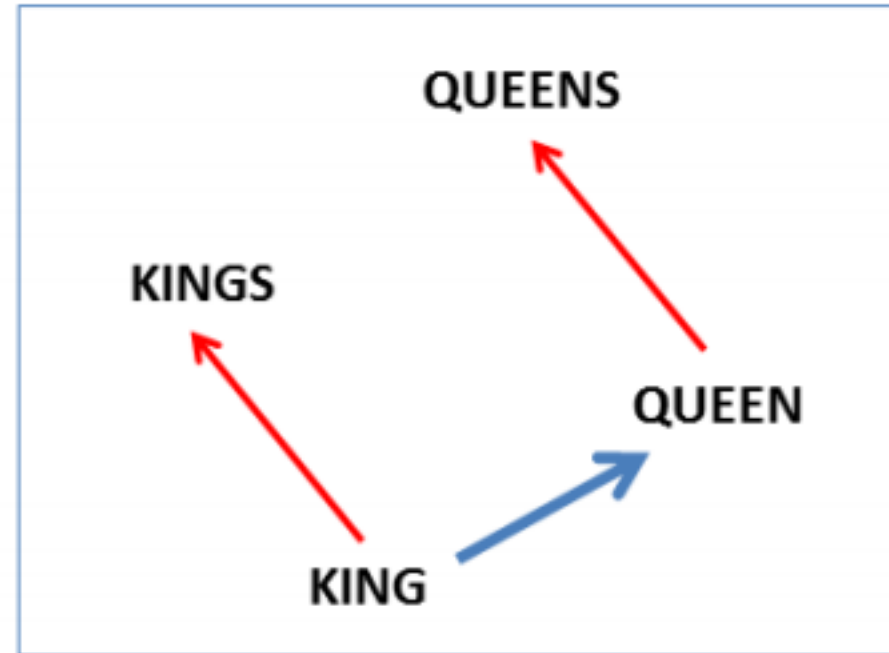
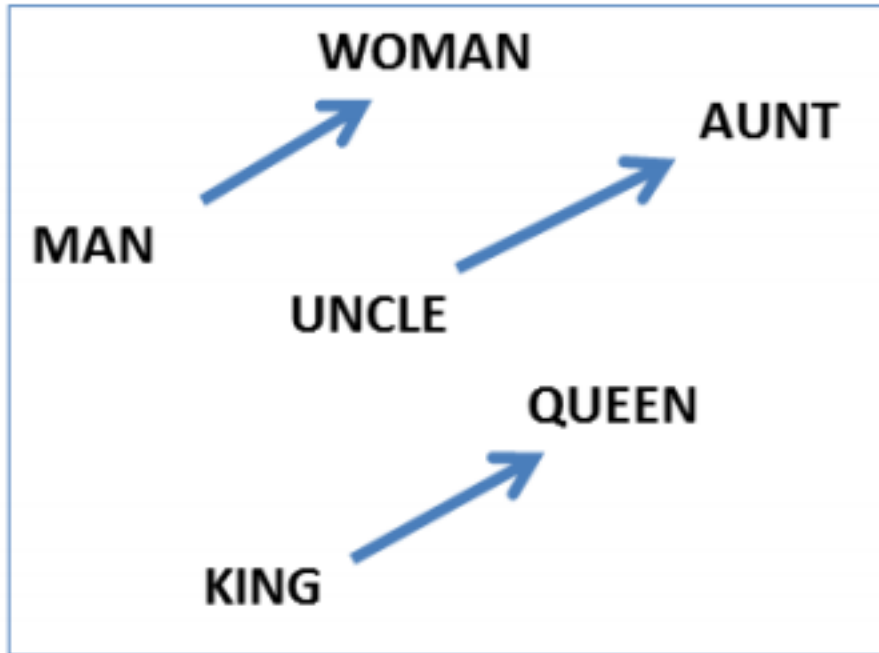


Figure 1: A simple CBOW model with only one word in the context

Самый популярный пример с word2vec



(Mikolov et al., NAACL HLT, 2013)

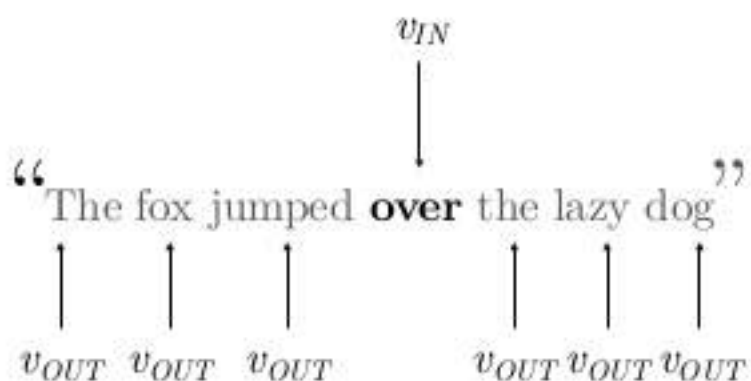
Особенности представлений

- Скалярное произведение векторов хорошо отражает похожесть слов по контекстам, в которых встречаются
- king – man \approx queen – woman
- Moscow – Russia \approx London – England
- Перевод: one – uno + four \approx quattro
- Хорошее признаковое описание текста – среднее арифметическое представлений слов текста

CBOW и Skip-gram

SkipGram

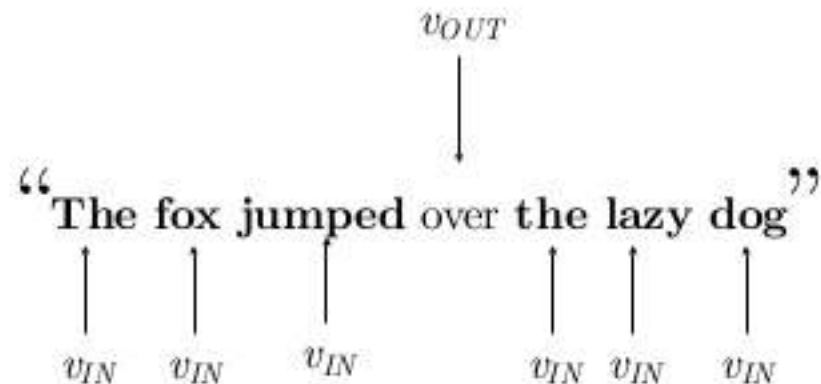
Guess the context
given the word



Better at syntax.
(this is the one we went over)

CBOW

Guess the word
given the context

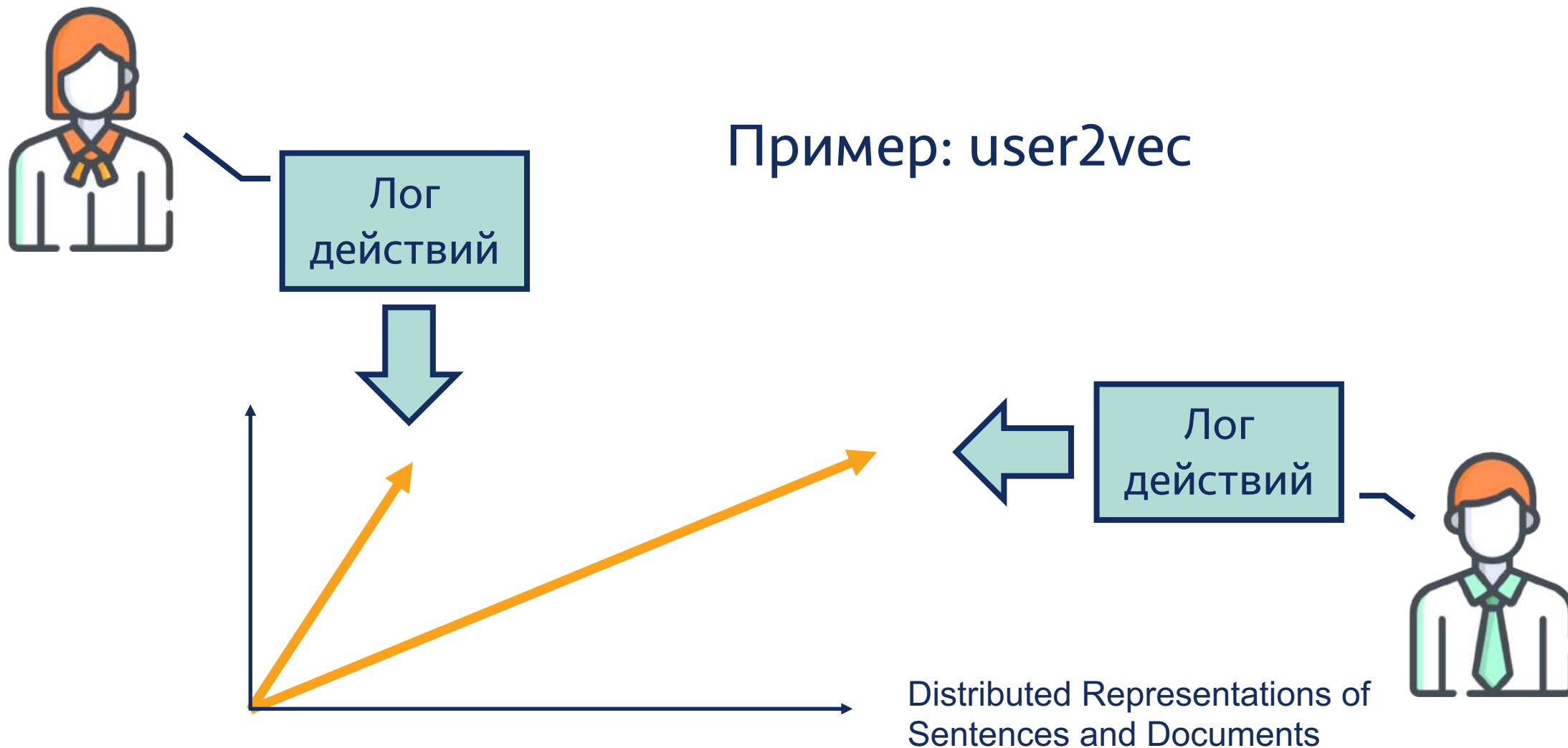


~20x faster.
(this is the alternative.)

Word2Vec и обучение с учителем

- Проблема мешка слов — слишком большое количество признаков
- Средний word2vec-вектор позволяет получить компактное признаковое описание
- При размерности вектора 100-500 можно обучать композиции деревьев

Word2Vec → Everything2Vec



Word2Vec и матричные разложения

$PMI(w_i, c_j)$ - совместная встречаемость w_i и c_j

Word2Vec и матричные разложения

$PMI(w_i, c_j)$ - совместная встречаемость w_i и c_j

Измеряется так:

$$PMI(w_i, c_j) = \ln \frac{p(w_i)p(c_j)}{p(w_i, c_j)}$$

Word2Vec и матричные разложения

$PMI(w_i, c_j)$ - совместная встречаемость w_i и c_j

Измеряется так:

$$PMI(w_i, c_j) = \ln \frac{p(w_i)p(c_j)}{p(w_i, c_j)}$$

Оказывается (Levi, NIPS 2014), Word2Vec выполняет матричное разложение матрицы, заполненной числами $PMI(w_i, c_j) - \ln k$ (k – количество примеров в Negative Sampling)

Общая идея эмбедингов

1. Есть объекты, для которых вам нужно обучить векторные представления v_i
2. Из каких соображений обучать представления – формулируется *оптимизационной задачей*, составленной из неких разумных соображений
3. Оптимизационная задача решается некоторым методом численной оптимизации (например, SGD)

План

1. Задача кластеризации

2. Понижение размерности

3. Матричные разложения

4. Векторные представления