OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

Проверить, идет ли запись!

Меня хорошо видно && слышно?

Ставьте + , если все хорошо

Напишите в чат, если есть проблемы

Матричные разложения. SVD

Легкоступ Виктор Валерьевич

Научный сотрудник

ОАО «АЛЕВКУРП»

Преподаватель

Легкоступ Виктор

- 5 лет работы научным сотрудником на предприятии, осуществляющим проектирование БЛА.
- Специализация: фильтрация данных, оценивание параметров систем, системы автоматического управления, обработка сигналов, численные методы, аэродинамика, параллельные вычисления.
- Базовые инструменты: Matlab/Simulink, Mathematica, Python, C++
- Профессиональные интересы: БЛА, системы управления и измерения, моделирование на C++, Python, Matlab

Правила вебинара

Активно участвуем

Задаем вопрос в чат или голосом

Off-topic обсуждаем в Slack #канал группы или #general

Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара

Матричные разложения. SVD

Рекомендации

ALS

Практика в Python

Цели вебинара | После занятия вы сможете

1. Рассказать, что такое SVD, его область применения, описать основные свойства

2. Рассказать смысл матричного приближения, ALS

3. Увидеть, как применить SVD в некоторых задачах

Смысл | Зачем вам это уметь

1 Разработка рекомендательных систем

2 Получение пространства признаков,
снижение размерности данных

3 Так или иначе, но SVD выходит далеко за рамки Data Science.
Большая часть всех разложений сводится к SVD.
SVD находит применение в:
методе наименьших квадратов, методе главных компонент,
сжатии данных, получении псевдообратных матриц и т.д.

Начало

Матричные разложения

Матричное разложение (англ. factorization/decomposition) – представление матрицы в виде произведения других более простых матриц (обычно это 2 или 3 матрицы), обладающих особенными или полезными свойствами.

Виды:

- SVD
- QR
- LU
- Cholesky

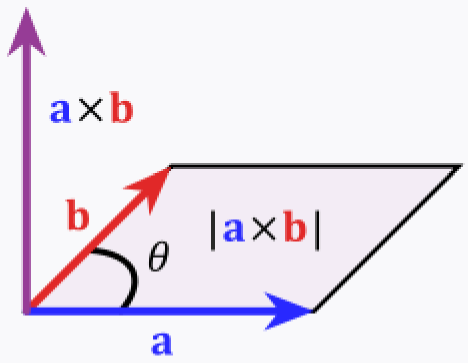
и другие

Singular Value Decomposition (SVD)

Чем оно полезно?

1) Выделение существенных характеристик данных
2) Снижение размерности данных (сжатие)
3) Фильтрация
4) Предсказание неизвестных данных

Разложение вектора на произведение векторов



Разложение матрицы на произведение матриц. SVD

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

U – ортогональная матрица левых собственных векторов XX^T

V – ортогональная матрица правых собственных векторов $X^T X$

Σ – диагональная матрица сингулярных чисел $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$

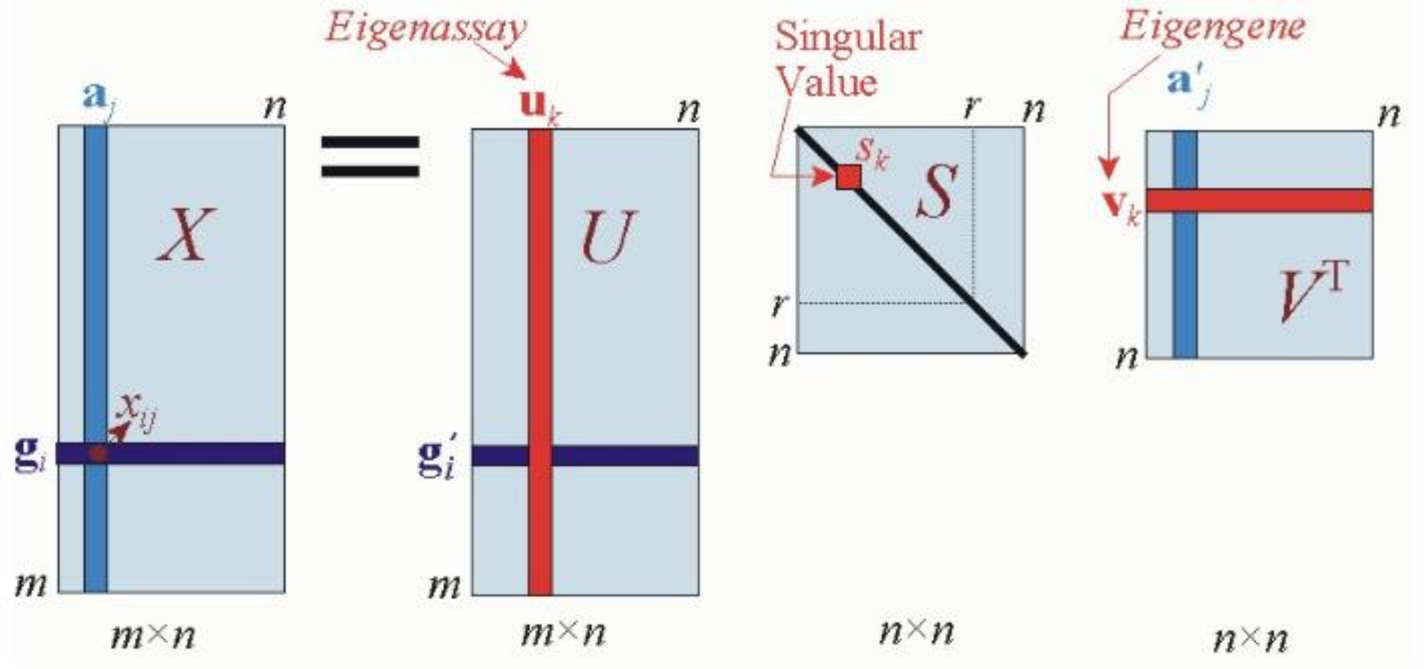
являющихся корнями из собственных чисел матрицы XX^T

$$U^T U = I$$

$$V^T V = I$$

$$\underset{m\times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m\times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r\times r}{\overset{\Sigma}{\begin{pmatrix} \sigma_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & \sigma_{rr} \end{pmatrix}}} \underset{r\times n}{\overset{V^\mathsf{T}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

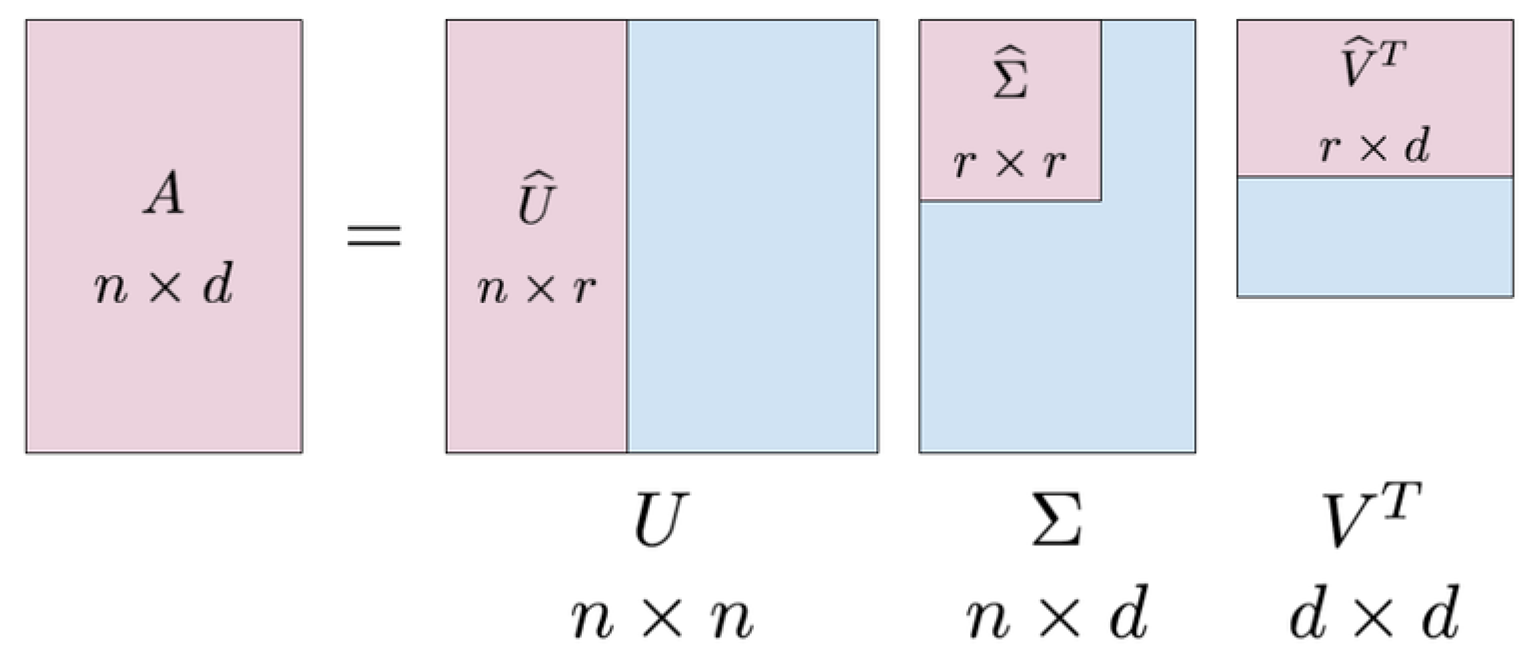
SVD



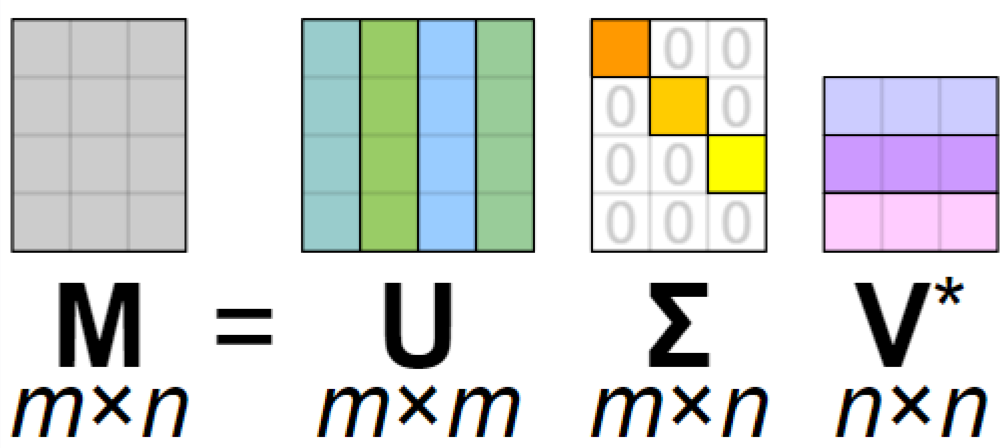
- Столбцы матрицы *U* – собственные векторы XX^T
- Столбцы матрицы *V* – собственные векторы $X^T X$
- На диагонали матрицы S находятся сингулярные числа – корни из собственных значений матрицы XX^T или $X^T X$

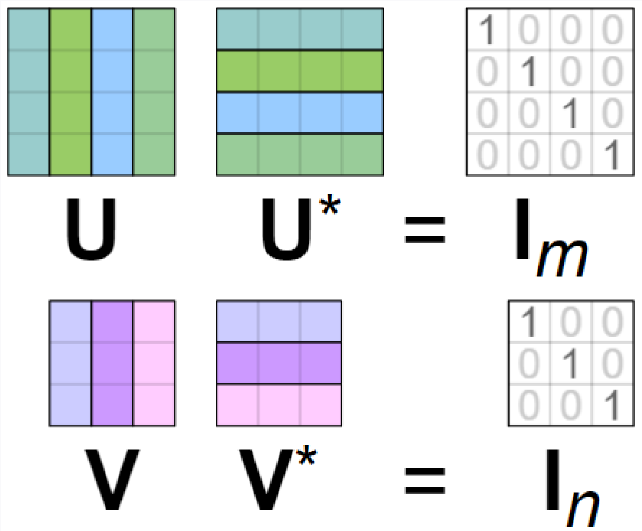
Усечённое SVD. Низкоранговое приближение.

- Приближаем матрицу *A* матрицей меньшего ранга *r*, учитывая то, что за основную часть информации матрицы *A* отвечают первые сингулярные числа.
- Таким образом получается сохранить наибольшую часть информации, задействовав меньше памяти. При этом отброшенные сингулярные числа считаем «шумовыми».
- Это является аналогом низкочастотной фильтрации данных.



Памятка о виде матриц в SVD





Пример SVD разложения матрицы

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

$$AA^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}$$

$$A^T A = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}$$

$\sigma_1 = \sqrt{25} = 5$ $\sigma_2 = \sqrt{9} = 3$

eigenvalues: $\lambda_1 = 25$, $\lambda_2 = 9$

eigenvectors

$$u_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad u_2 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

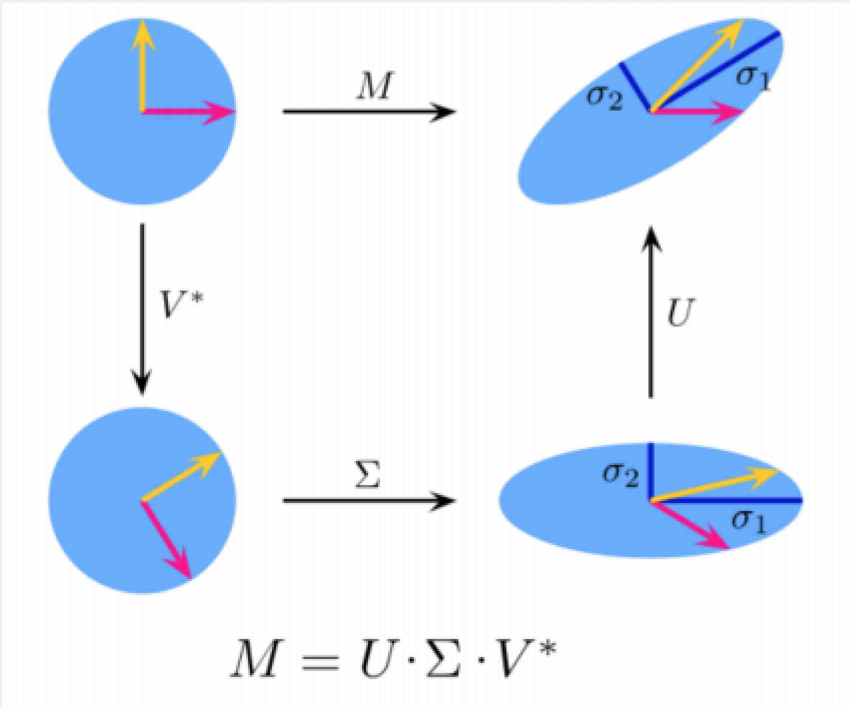
eigenvalues: $\lambda_1 = 25$, $\lambda_2 = 9$, $\lambda_3 = 0$

eigenvectors

$$v_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 1/\sqrt{18} \\ -1/\sqrt{18} \\ 4/\sqrt{18} \end{pmatrix} \quad v_3 = \begin{pmatrix} 2/3 \\ -2/3 \\ -1/3 \end{pmatrix}$$

$$A = USV^T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}$$

Геометрический смысл SVD



Связь SVD с Методом Главных Компонент (PCA)

Необходимо диагонализировать ковариационную матрицу $C = \dfrac{X^T X}{n-1}$

где X – центрированная матрица исследуемых данных

Представим ковариационную матрицу через SVD:

$$\boxed{C = \frac{V\Sigma U^T U \Sigma V^T}{n-1} = V \frac{\Sigma^2}{n-1} V^T}$$

$\lambda_i = \dfrac{\sigma_i^2}{n-1}$ – собственные значения матрицы $X^T X$

Столбцы матрицы U соответствуют собственным векторам матрицы $X^T X$

Это считается быстрее, чем вычисление ковариационной матрицы в лоб с последующим получением собственных векторов

Как применить SVD для рекомендации фильмов

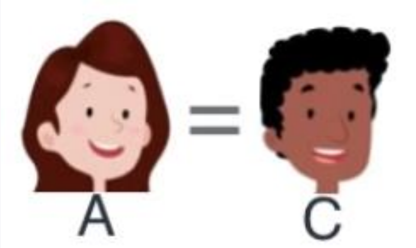
Четыре пользователя оценили пять фильмов

Кластеризация пользователей и фильмов

Как применить SVD для рекомендации фильмов

	M 1	M 2	M 3	M 4	M 5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

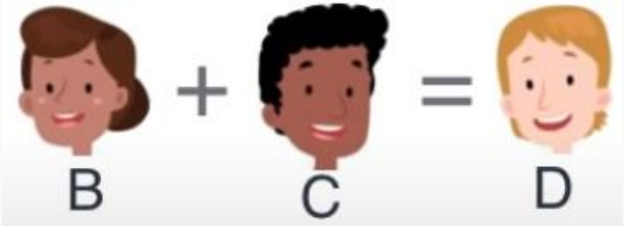
	M1	M2	M3	M4	M5
	3	1	1	3	1
	3	1	1	3	1



Как применить SVD для рекомендации фильмов

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4



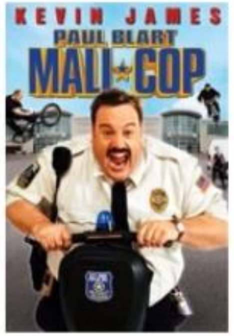
Как применить SVD для рекомендации фильмов

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

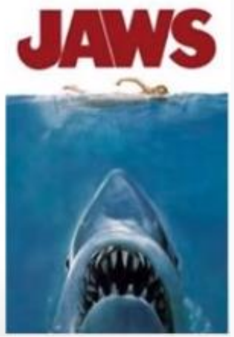
	M1	M2	M3	M4	M5
	3			3	
	1			1	
	3			3	
	4			4	

	M1	M2	M3	M4	M5
		1	1		1
		2	4		3
		1	1		1
		3	5		4

M1 = M4



M5 = Average(M2, M3)



Как применить SVD для рекомендации фильмов

Что, если некоторых оценок нет и мы хотим их спрогнозировать?

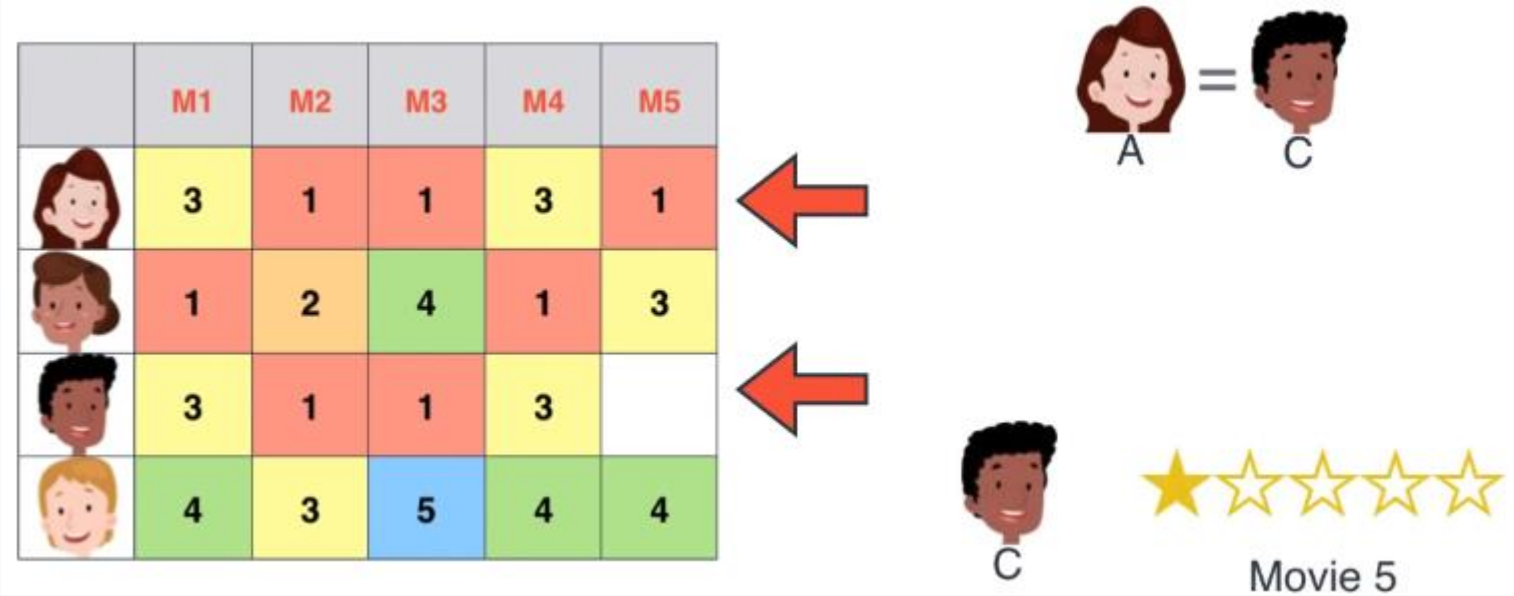
Вопрос: на основании чего мы можем это сделать?

	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	?	3
	3	3	3	3	3

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	
	4	3	5	4	4

Как применить SVD для рекомендации фильмов

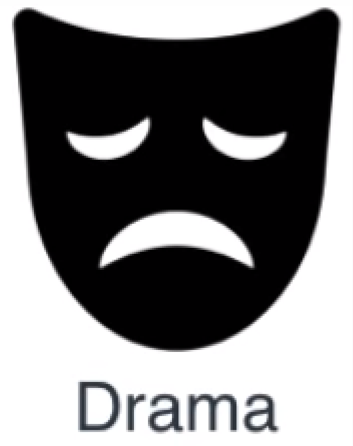
Что, если некоторых оценок нет и мы хотим их спрогнозировать?



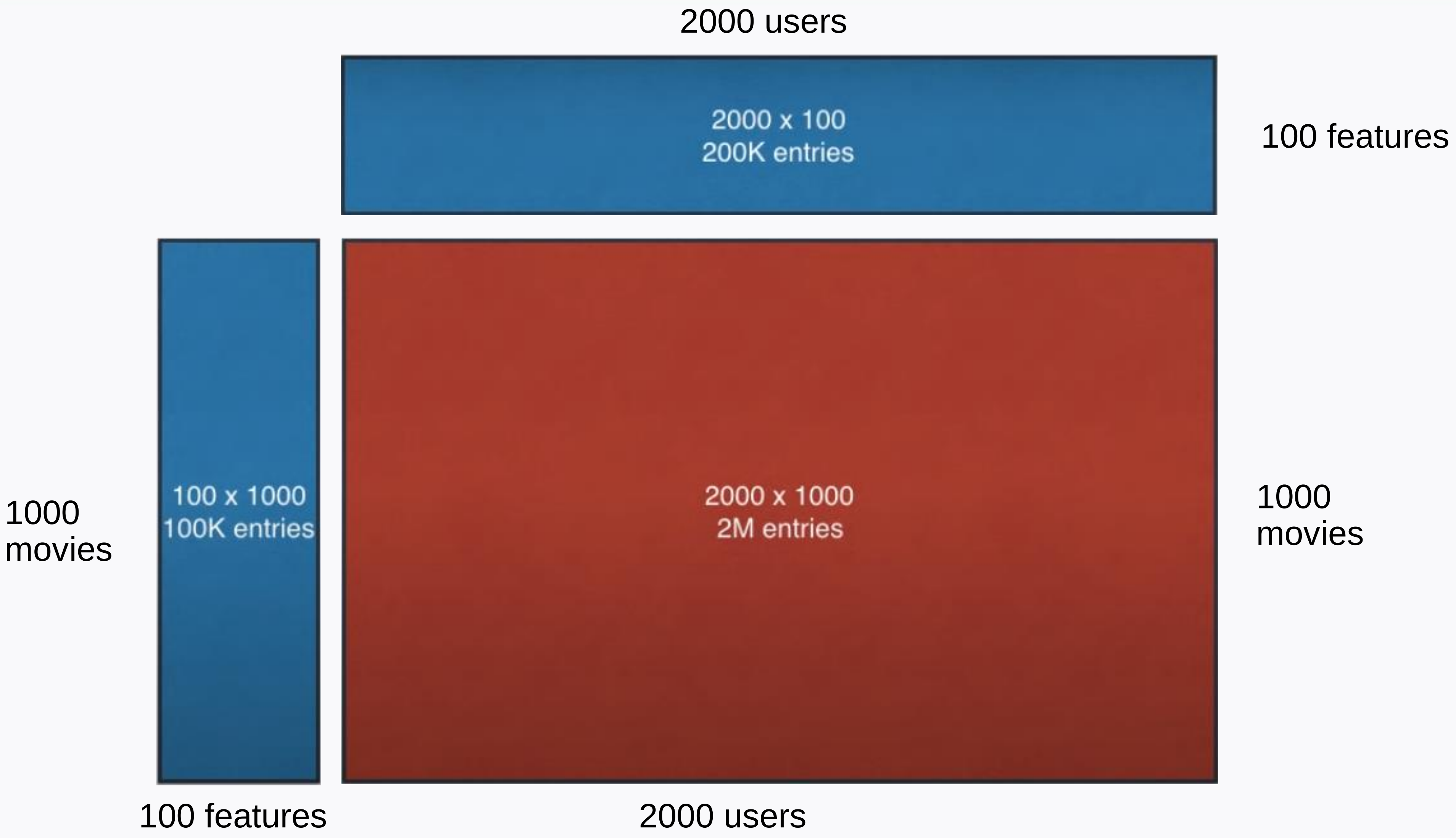
Как применить SVD для рекомендации фильмов

Вместо того, чтобы хранить информацию в изначальном виде с «закодированными признаками», мы можем попытаться выделить некоторое пространство признаков, осуществляющие связь между пространствами пользователей и фильмов

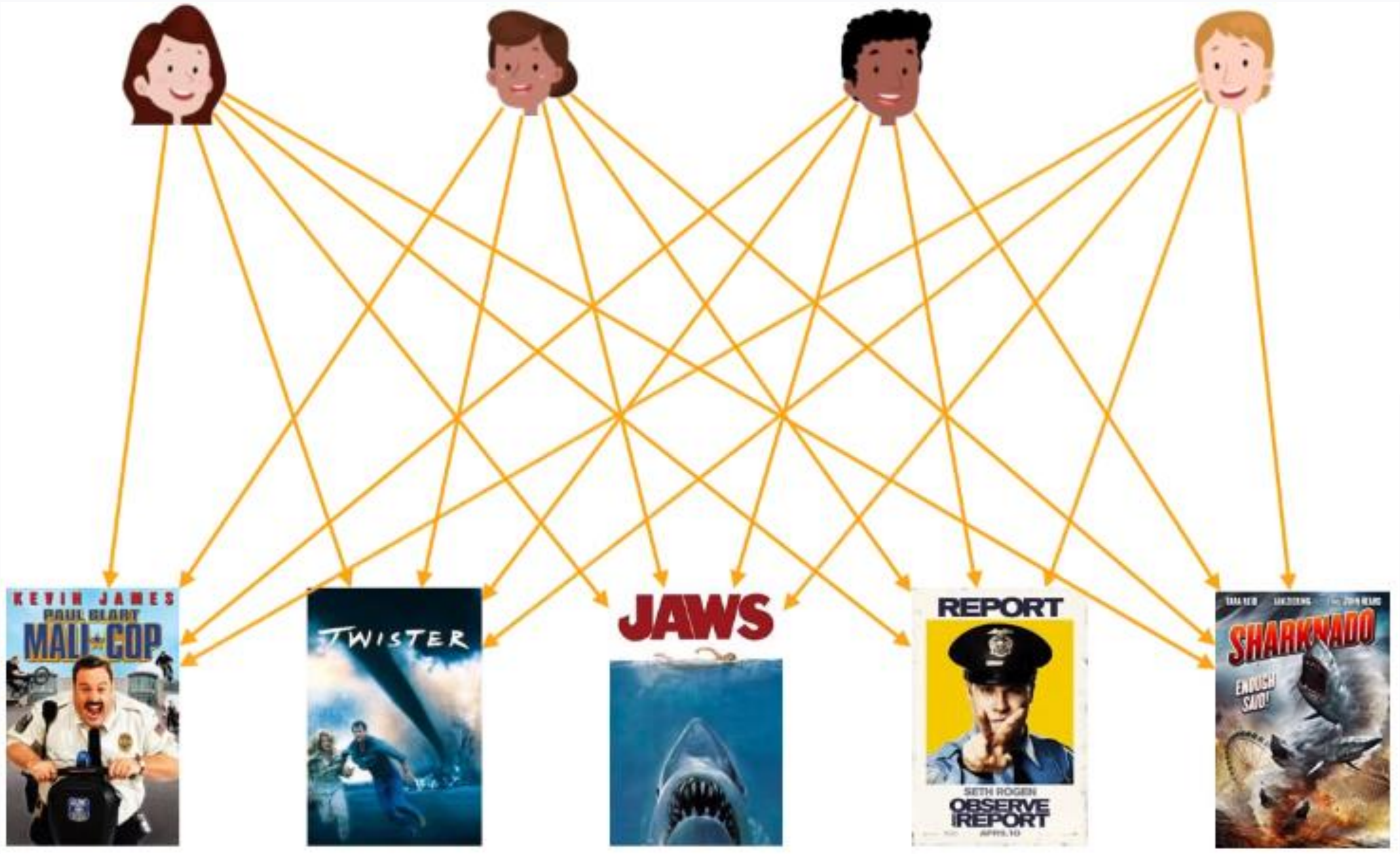
Features



Как применить SVD для рекомендации фильмов

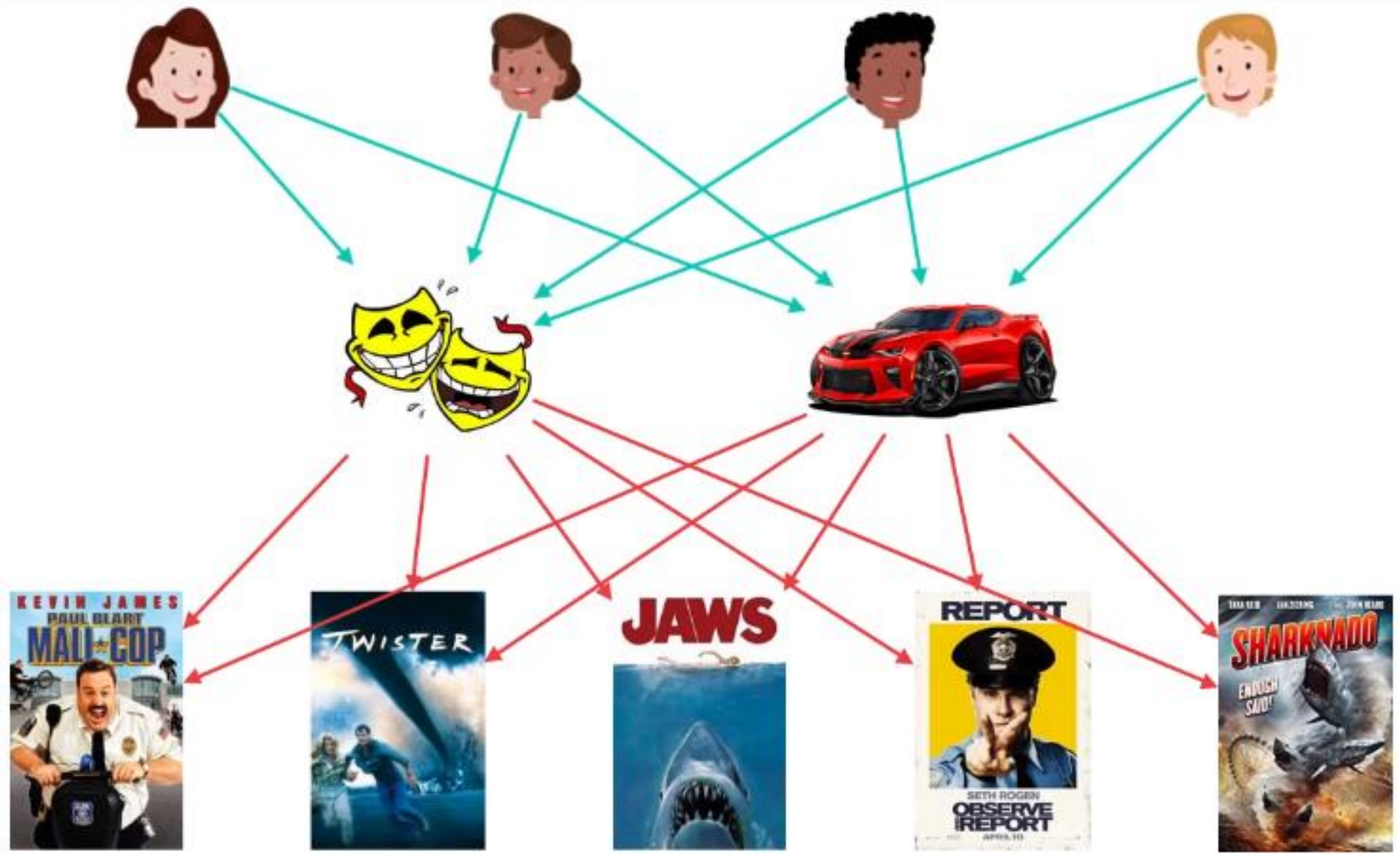


Как применить SVD для рекомендации фильмов



$$5 \times 4 = 20$$

Как применить SVD для рекомендации фильмов



$$8+10=18$$

Как применить SVD для рекомендации фильмов

Степень присутствия двух признаков в пространствах пользователей и фильмов

	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

	Comedy	Action
A	✓	✗
B	✗	✓
C	✓	✗
D	✓	✓

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Как применить SVD для рекомендации фильмов

Что, если некоторых оценок нет и мы хотим их спрогнозировать?

Вопрос: как применить SVD?

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	
	4	3	5	4	4

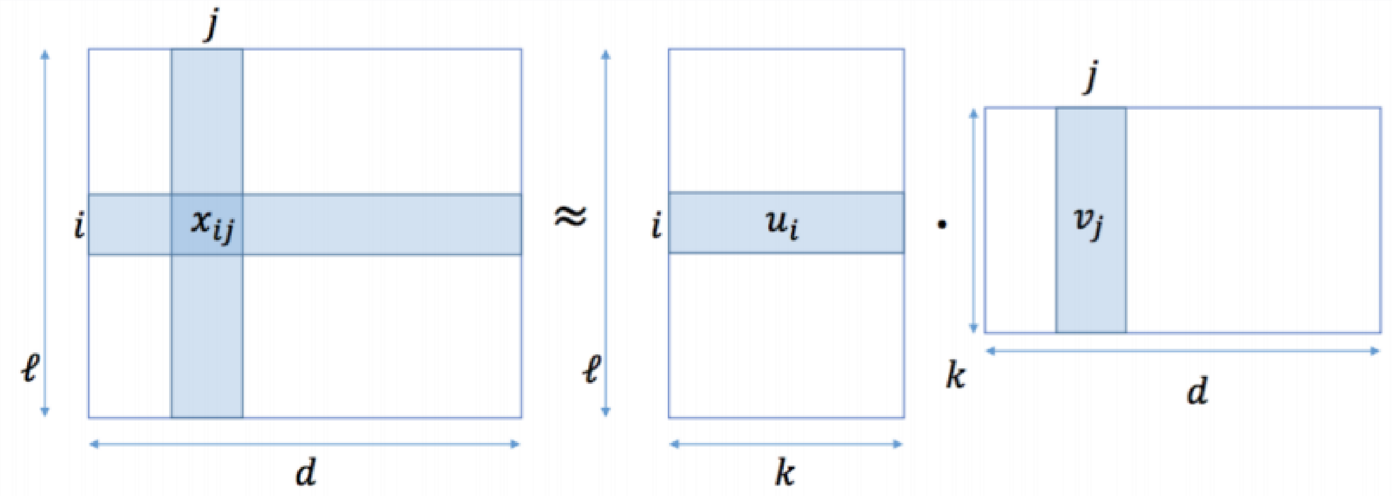
Alternative Least Squires (ALS)

Попытаемся приблизить заданную матрицу X

$$X_{l,d} \approx U_{l,k} \cdot V_{k,d}^T,$$

В качестве меры схожести используем норму Фробениуса по существующим элементам

$$Q = \sum_{i,j} \left(\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \rangle - x_{ij} \right)^2 \to \min_{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j}$$



От k зависит то, насколько точно мы сможем приблизить исходную матрицу произведением наших матриц

Alternative Least Squires (ALS)

Будем использовать метод Градиентного Спуска (GD)

$$\hat{\mathbf{u}}_i^{(t+1)} = \hat{\mathbf{u}}_i^{(t)} - \gamma^{(t)} \frac{\partial Q}{\partial \hat{\mathbf{u}}_i}$$

$$\frac{\partial Q}{\partial \hat{\mathbf{u}}_i} = 2\sum_j \left(\left\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \right\rangle - x_{ij} \right) \hat{\mathbf{v}}_j = 2\sum_j \varepsilon_{ij} \hat{\mathbf{v}}_j^{(t)}$$

$$\varepsilon_{ij} = \left\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \right\rangle - x_{ij}$$ – *невязка*

$$\boxed{\begin{aligned} \hat{\mathbf{u}}_i^{(t+1)} &= \hat{\mathbf{u}}_i^{(t)} - \gamma_u^{(t)} \sum_j \varepsilon_{ij} \hat{\mathbf{v}}_j^{(t)} \\ \hat{\mathbf{v}}_j^{(t+1)} &= \hat{\mathbf{v}}_j^{(t)} - \gamma_v^{(t)} \sum_i \varepsilon_{ij} \hat{\mathbf{u}}_i^{(t)} \end{aligned}}$$

Alternative Least Squires (ALS)

Так как в задаче могут использоваться большие матрицы с произвольной структурой, возможны проблемы со сходимостью.

Используем метод Стохастического Градиентного Спуска (SGD)

$$\boxed{\begin{aligned}\hat{\mathbf{u}}_i^{(t+1)} &= \hat{\mathbf{u}}_i^{(t)} - \gamma_u^{(t)} \varepsilon_{ij} \hat{\mathbf{v}}_j^{(t)} \\ \hat{\mathbf{v}}_j^{(t+1)} &= \hat{\mathbf{v}}_j^{(t)} - \gamma_v^{(t)} \varepsilon_{ij} \hat{\mathbf{u}}_i^{(t)}\end{aligned}}$$

i, j выбираются случайным образом

Alternative Least Squires (ALS)

Так как в задаче могут использоваться большие матрицы с произвольной структурой, возможны проблемы со сходимостью.

Используем метод Стохастического Градиентного Спуска (SGD)

$$\hat{\mathbf{u}}_i^{(t+1)} = \hat{\mathbf{u}}_i^{(t)} - \gamma_u^{(t)} \varepsilon_{ij} \hat{\mathbf{v}}_j^{(t)}$$

$$\hat{\mathbf{v}}_j^{(t+1)} = \hat{\mathbf{v}}_j^{(t)} - \gamma_v^{(t)} \varepsilon_{ij} \hat{\mathbf{u}}_i^{(t)}$$

i, j выбираются случайным образом

Какие проблемы существуют при использовании этого метода? Как выбрать шаг?

Alternative Least Squires (ALS)

Выведем Альтернативный Метод Наименьших Квадратов

$$\frac{\partial Q}{\partial \hat{\mathbf{u}}_i} = 2\sum_j \left(\left\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \right\rangle - x_{ij} \right) \hat{\mathbf{v}}_j = 0$$

$$\sum_j \left\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \right\rangle \hat{\mathbf{v}}_j = \sum_j x_{ij} \hat{\mathbf{v}}_j$$

Это уже система линейных уравнений

$$\boxed{\begin{cases} \left(\sum_j \hat{\mathbf{v}}_j \hat{\mathbf{v}}_j^T \right) \hat{\mathbf{u}}_i = \sum_j x_{ij} \hat{\mathbf{v}}_j & \Rightarrow \quad \hat{\mathbf{u}}_i \\ \left(\sum_i \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i^T \right) \hat{\mathbf{v}}_j = \sum_i x_{ij} \hat{\mathbf{u}}_i & \Rightarrow \quad \hat{\mathbf{v}}_j \end{cases}}$$

i, j выбираются случайным образом до тех пор, пока не сойдется с нужной точностью

Alternative Least Squires (ALS)

Добавим L2 регуляризацию к функционалу для предотвращения переобучения, так как задача может иметь множество решений

$$Q = \sum_{i,j}\left(\langle \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j \rangle - x_{ij}\right)^2 + \alpha \sum_i \left\| \hat{\mathbf{u}}_i \right\|^2 + \beta \sum_j \left\| \hat{\mathbf{v}}_j \right\|^2 \to \min_{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j}$$

α, β малые – 0.1, 0.01, и т.д.


LIVE

SVD в Python

- Максимально просто сделать лишь разложение SVD
- Не так просто осмыслить результат

Код Python

```
import numpy as np
U, S, V = np.linalg.svd(X)                # Получили сразу три матрицы разложения
```

Слайд с заданием

1. Как выглядит SVD
2. SVD в рекомендательных системах
3. SVD в сжатии изображений

Тайминг: 30 минут

Слайд с тезисами

1 SVD дает произведение двух ортогональных матриц собственных векторов и диагональной матрицы сингулярных чисел

2 SVD применим в выделении признаков, в снижении размерностей данных, в сжатии, в PCA

3 Сингулярные числа отсортированы по убыванию. Наибольшее значение имеют первые числа

4 SVD применимо только к полным матрицам. В случае пропусков применяем ALS

5 Чистое SVD – это линейное и не всесильное разложение. И часто не самое быстрое.

Рефлексия

С какими основными мыслями и инсайтами уходите с вебинара?

Достигли ли вы цели вебинара?

Следующий вебинар

Тема: «Матричные производные»

19 июня

Ссылка на вебинар будет в ЛК за 15 минут

Материалы к занятию в ЛК — можно изучать

Обязательный материал обозначен красной лентой

Список материалов для изучения

- Очень много инфы по SVD и регрессии на русском: http://www.machinelearning.ru/wiki/images/b/b9/Strijov08ln.pdf
- Дипломная по рекомендациям: http://www.machinelearning.ru/wiki/images/9/93/2014_517_RomovPA.pdf
- Основы на русском: https://www.coursera.org/lecture/unsupervised-learning/matrichnyie-razlozhieniia-DUBp9
- Мега курс от Стива Брантона!!!: https://www.youtube.com/playlist?list=PLMrJAkhIeNNSVjnsviglFoY2nXildDCcv
- Как связать SVD и рекомендации фильмов: https://www.youtube.com/watch?v=ZspR5PZemcs
- Крутой ресурс «Связь с Методом Главных Компонент»: https://medium.com/@jonathan_hui/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491
- Видос: https://www.youtube.com/watch?v=FgGjc5oabrA
- Хорошая статейка «Collaborative Filtering for Implicit Feedback Datasets»: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=34AEEE06F0C2428083376C26C71D7CFF?doi=10.1.1.167.5120&rep=rep1&type=pdf

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары

Легкоступ Виктор Валерьевич