

Матричные разложения и их применение

Доклад представляли:
Кузнецов Дмитрий
Факанов Павел
БПМИ171

О чём пойдет разговор

- 1) Линейная алгебра
- 2) SVD
- 3) SVD в машинном обучении
- 4) Способы построения разложения
- 5) Применение SVD в реальных задачах

Базис

Пусть \mathbf{E} - евклидово пространство размерности n над полем \mathbf{F}

Базисом мы называем набор векторов $e_1, e_2, \dots, e_n \in \mathbf{E}$ т.ч.

$$\exists \alpha_1, \alpha_2, \dots, \alpha_n \in \mathbf{F}, \exists i : \alpha_i \neq 0 : \alpha_1 e_1 + \alpha_2 e_2 + \dots + \alpha_n e_n = 0$$

ОНБ - базис $e_1, e_2, \dots, e_n \in \mathbf{E}$ т.ч.

$$\forall i \neq j : (e_i, e_j) = 0$$

$$\forall i = \overline{1, n} : |e_i| = 1$$

Линейное отображение

Отображение $\phi : E \rightarrow V$ называется *линейным* если:

$$\forall x, y \in E \quad \forall \alpha \in F$$

$$\phi(x + y) = \phi(x) + \phi(y)$$

$$\phi(\alpha x) = \alpha\phi(x)$$

Оператор $\phi : E \rightarrow E$ называется *линейным* если:

$$\forall x, y \in E \quad \forall \alpha \in F$$

$$\phi(x + y) = \phi(x) + \phi(y)$$

$$\phi(\alpha x) = \alpha\phi(x)$$

Матрицы и отображения

Пусть $e \in E$ и $f \in V$ - фиксированные базисы
Тогда $\text{Hom}(E, V)$ изоморфно $\text{Mat}_{n \times m}(F)$

Самосопряженный лин. оператор

Линейный оператор ϕ называется диагонализуемым, если:

Эе - базис, т.ч.

Матрица оператора в этом базисе имеет диагональный вид

Оператор называется самосопряженным, если: $(\phi(x), y) = (x, \phi(y))$

Самосопряженный лин. оператор

Всякий самосопряженный лин. оператор диагонализуем

Более того, для него всегда найдется ОНБ из собственных векторов

Теорема о сингулярных базисах

Пусть $\phi : \mathbf{E} \rightarrow \mathbf{E}'$, $r = rk\phi$

Тогда:

Э ОНБ $\mathbf{e} \in \mathbf{E}$ и ОНБ $\mathbf{f} \in \mathbf{E}'$ т.ч.

$$A(\phi; \mathbf{e}; \mathbf{f}) = \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_r & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}, \text{ где } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: e - ОНБ из собственных векторов ψ

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: \mathbf{e} - ОНБ из собственных векторов ψ $A(\psi; \mathbf{e}) = \text{diag}(s_1, \dots, s_n)$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: e - ОНБ из собственных векторов ψ $A(\psi; e) = \text{diag}(s_1, \dots, s_n)$

$$(\psi(e_i), e_i) = (\phi(e_i), \phi(e_i)) \geq 0$$

$$(\psi(e_i), e_i) = s_i(e_i, e_i) = s_i \geq 0$$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: e - ОНБ из собственных векторов ψ $A(\psi; e) = \text{diag}(s_1, \dots, s_n)$

$$(\psi(e_i), e_i) = (\phi(e_i), \phi(e_i)) \geq 0$$

$$(\psi(e_i), e_i) = s_i(e_i, e_i) = s_i \geq 0$$

Введем базис в E' :

$$f_i := \frac{1}{\sigma_i} \phi(e_i) \in E'$$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: e - ОНБ из собственных векторов ψ $A(\psi; e) = \text{diag}(s_1, \dots, s_n)$

$$(\psi(e_i), e_i) = (\phi(e_i), \phi(e_i)) \geq 0$$

$$(\psi(e_i), e_i) = s_i(e_i, e_i) = s_i \geq 0$$

Введем базис в E' :

И заметим:

$$f_i := \frac{1}{\sigma_i} \phi(e_i) \in E'$$

$$(f_i, f_j) = (\frac{1}{\sigma_i} \phi(e_i), \frac{1}{\sigma_j} \phi(e_j))' = \delta_{ij}$$

Идея доказательства

Рассмотрим $\psi := \phi^* \phi$

Заметим, что:

$$\forall x, y \in E : (\psi(x), y) = (\phi^*(\phi(x)), y) = (\phi(x), \phi(y))' = (x, \psi(y))$$

Следовательно: $\psi = \psi^*$

Тогда имеем: e - ОНБ из собственных векторов ψ $A(\psi; e) = \text{diag}(s_1, \dots, s_n)$

$$(\psi(e_i), e_i) = (\phi(e_i), \phi(e_i)) \geq 0$$

$$(\psi(e_i), e_i) = s_i(e_i, e_i) = s_i \geq 0$$

Введем базис в E' :

И заметим:

А значит:

$$f_i := \frac{1}{\sigma_i} \phi(e_i) \in E'$$

$$(f_i, f_j) = (\frac{1}{\sigma_i} \phi(e_i), \frac{1}{\sigma_j} \phi(e_j))' = \delta_{ij}$$

f_1, \dots, f_k - орт.норм.система в E'

Идея доказательства

Т.к.

$$\phi(e_1), \dots, \phi(e_n) = (f_1, \dots, f_m) A(\phi; \mathbf{e}, \mathbf{f})$$

$$A(\phi; \mathbf{e}; \mathbf{f}) = \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_r & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}, \text{ где } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$$

Идея доказательства

Т.к.

$$\phi(e_1), \dots, \phi(e_n) = (f_1, \dots, f_m) A(\phi; \mathbf{e}, \mathbf{f})$$

$$A(\phi; \mathbf{e}; \mathbf{f}) = \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_r & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}, \text{ где } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$$

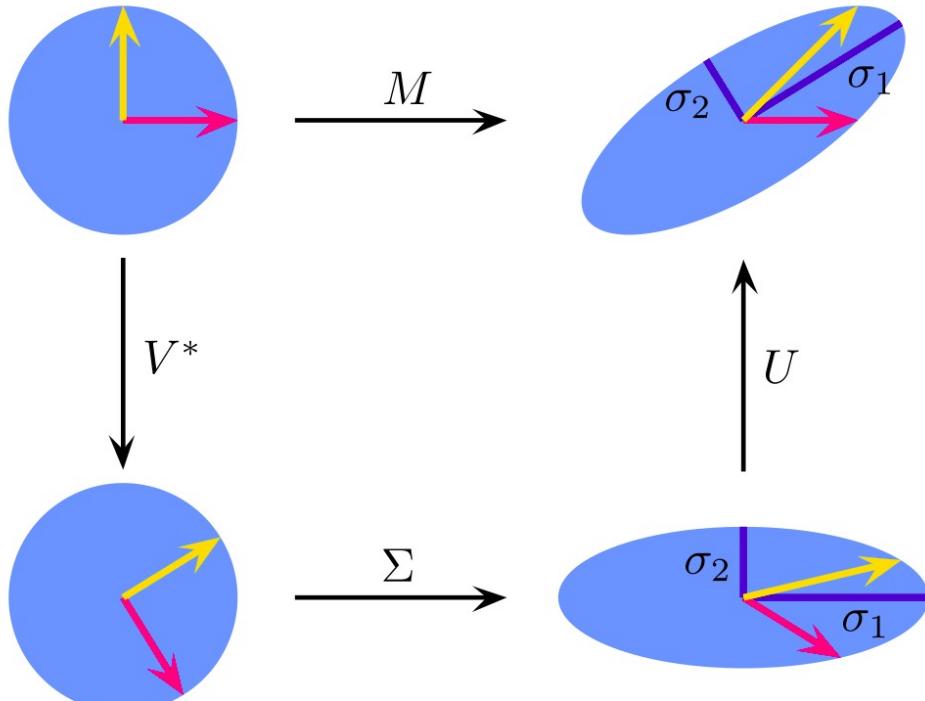


Следствие (SVD)

$$A = U\Sigma V^T, \text{ где } UU^T = E \text{ и } VV^T = E$$

$$\Sigma = \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_r & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}$$

Следствие (SVD)



$$M = U \cdot \Sigma \cdot V^*$$

Геометрический смысл не
дает интуитивного
понимания разложения.
Однако

SVD раскладывает
линейное отображение на
повороты и растяжения

Следствие (SVD)

$$A_{(m \times n)} = U_{(m \times \tau)} \Sigma_{(\tau \times \tau)} V^T_{(\tau \times n)}$$

Данное разложение называется
усеченным сингулярным разложением.
Оно порождают новую матрицу A_k

Теорема о низкоранговом приближении

Пусть $A \in Mat_{m \times n}(\mathbf{R})$

$$A = U\Sigma V^T, \Sigma = \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_r & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \Rightarrow \text{Минимум } ||A - B|| \text{ среди всех } B, rk B \leq k \text{ достигается при } B = U\Sigma_k V^T$$

Пусть $k < r$ и $\Sigma_k =$

$$\begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma_k & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}$$

Почему это все можно было не рассказывать

- Посчитать ковариационную матрицу
- Найти у нее собственные значения и вектора (не самая тривиальная операция)
- Дополнить до ОНБ вторую систему
- Больно? Больно

Почему это все можно было не рассказывать

- Посчитать ковариационную матрицу
- Найти у нее собственные значения и вектора (не самая тривиальная операция) => Ищем другой подход
- Дополнить до ОНБ вторую систему
- Больно? Больно

“SVD”

Мы хотим получить разложение исходной матрицы в таком виде:

Переобозначим матрицы разложения:

И теперь мы хотим найти:

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

$$U = \hat{U}\sqrt{\Sigma}, V = \hat{V}\sqrt{\Sigma}$$

$$\mathbf{X} = UV^T$$

Заметим, что в таком виде разложение уже не единственное

“SVD”

Найдем не точное разложение, а
наиболее близкое по норме
Фробениуса.

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

Решать гладкие оптимизационные
задачи мы уже умеем.

“SVD”

Найдем не точное разложение, а
наиболее близкое по норме
Фробениуса.

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

Решать гладкие оптимизационные
задачи мы уже умеем.

“SVD”

Найдем не точное разложение, а
наиболее близкое по норме
Фробениуса.

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

Решать гладкие оптимизационные
задачи мы уже умеем.

Мотивирующий пример

Сформулируем кратко задачу рекомендации.

Пусть мы знаем оценки фильмов, поставленные некоторыми пользователями нашей рекомендательной площадки.

Хотим для Коли предсказать: а понравится ли ему “Улица Вязов”?

| | 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 |

Мотивирующий пример

Постановка задачи:

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

Почему именно матричные разложения? Причем тут SVD?

По построению SVD: U – базис из собственных векторов ковариационной матрицы $X^T X$
V – матрицы XX^T (причем отсортированные по сингулярным числам)

Т.е. построением SVD мы строим две наиболее близкие с точки зрения нормы Фробениуса матрицы. Матрицы которые сохраняют максимум информации о пользователях и фильмах, соответственно

Мотивирующий пример

Постановка задачи:

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

Как получить новое предсказание?

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

GD

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

Градиент выглядит так:

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

ϵ_j

Обозначим: $\epsilon_{ij} = (\langle u_i, v_j \rangle - x_{ij})$

GD

Шаг GD:

$$u_i^{(t+1)} = u_i^{(t)} - \gamma_t \sum_j \varepsilon_{ij} v_j.$$
$$\varepsilon_{ij} = (\langle u_i, v_j \rangle - x_{ij})$$

На каждой итерации по каждой компоненте нам придется суммировать линейное число слагаемых. Больно

SGD

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

$$\varepsilon_{ij} = (\langle u_i, v_j \rangle - x_{ij})$$

Теперь шаг оптимизации более
легковесный. Но как мы знаем
сходится медленнее

А если не хотим SGD? Есть ALS

Если будем фиксировать одну из матриц, а по второй будем оптимизировать:

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

В таком случае на каждом шаге задача оптимизации вырождается в типичный МНК

А если не хотим SGD? Есть ALS

Если будем фиксировать одну из матриц, а по второй будем оптимизировать:

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

В таком случае на каждом шаге задача оптимизации вырождается в типичный МНК

Разберемся с алгоритмом чуть подробнее

А если не хотим SGD? Есть ALS

На каждом шаге по очереди фиксируем одну из матриц. Пусть, например, V .

Решаем следующую систему относительно другой матрицы и обновляем значения:

На следующем шаге фиксируем U .
Решаем, обновляем:

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

$$\frac{\partial Q}{\partial v_i} = 0.$$

ALS. Подробнее о шаге

Шаг выглядит так:

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

Распишем производную чуть
подробнее:

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

ALS. Подробнее о шаге

Распишем производную чуть
подробнее:

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

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

ALS. Подробнее о шаге

Распишем производную чуть
подробнее:

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

$$\sum_j v_j \langle v_j, u_i \rangle = \sum_j x_{ij} v_j \iff \sum_j v_j v_j^T u_i = \sum_j x_{ij} v_j$$

ALS. Подробнее о шаге

Распишем производную чуть
подробнее:

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

$$\sum_j v_j \langle v_j, u_i \rangle = \sum_j x_{ij} v_j \iff \sum_j v_j v_j^T u_i = \sum_j x_{ij} v_j \iff$$

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

Решаем такие СЛУ на каждой
итерации, пока не сойдемся.

ALS. Подробнее о шаге

Шаг ALS до сих очень дорогой. Но
ALS параллелизируется, что дает ему
победное преимущество перед
SGD, сохраняя достоинства
обычного GD

В источниках есть статья, в
котором приводятся эксперименты
по распараллизации данного
алгоритма.

ALS. Подробнее о шаге

Шаг ALS до сих очень дорогой. Но ALS параллелизируется, что дает ему победное преимущество перед SGD, сохраняя достоинства обычного GD

В источниках есть статья, в которой приводятся эксперименты по распараллизации данного алгоритма.

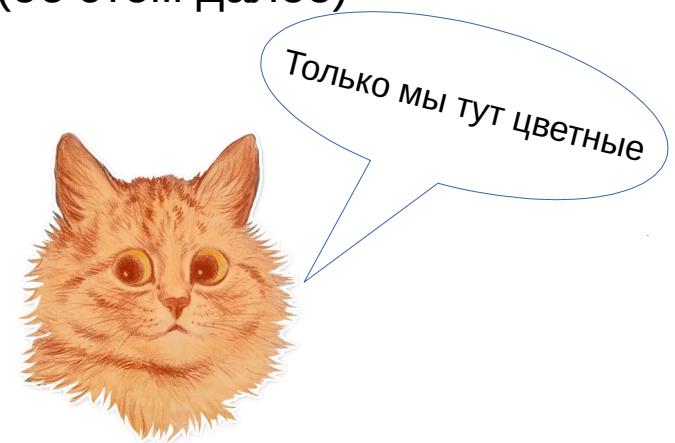
Более того, у данного метода есть модификации, которые дают ему еще больше преимуществ в других задачах (об этом далее)

ALS. Подробнее о шаге

Шаг ALS до сих очень дорогой. Но ALS параллелизируется, что дает ему победное преимущество перед SGD, сохраняя достоинства обычного GD

В источниках есть статья, в котором приводятся эксперименты по распараллеливанию данного алгоритма.

Более того, у данного метода есть модификации, которые дают ему еще больше преимуществ в других задачах (об этом далее)



Еще один пример

Пусть теперь мы рекомендуем не фильмы, а товары.

| | 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 |

Таргет теперь – кол-во товаров, которые купит клиент. Т.е. клетки матрицы – покупки клиента.

Еще один пример

Пусть теперь мы рекомендуем не фильмы, а товары.

| | 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 |

Таргет теперь – кол-во товаров, которые купит клиент. Т.е. клетки матрицы – покупки клиента.

Проблема: где в обучающей выборке объекты отрицательного “класса”? Негативных примеров нет. Это больно

В данной ситуации случаи “товар не понравился” и “пользователь не видел товар” не различимы.

Чуть-чуть понятий из области

Explicit feedback – обратная связь от пользователя, когда имеются и положительные, и отрицательные примеры

Implicit feedback - есть либо положительные, либо отрицательные примеры

Implicit ALS

Идея: занулить все неизвестные значения матрицы.

Если все оставить как есть, то мы внесем некоторую априорную информацию о паре (клиент, товар)

Поэтому предлагается взвесить ошибки предсказания (теперь оптимизация по всем клеткам):

Веса нулевых клеток полагаются несколько меньше регулярных.
Например, можно ввести вес так:

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

$$w_{ij} = 1 + \alpha |x_{ij}|,$$

Контрольные вопросы

1. Укажите геометрическую интерпретацию сингулярного разложения. Сформулируйте интуитивное понимание смысла матриц U, V и Sigma с точки зрения информативности матриц (по норме Фробениуса).

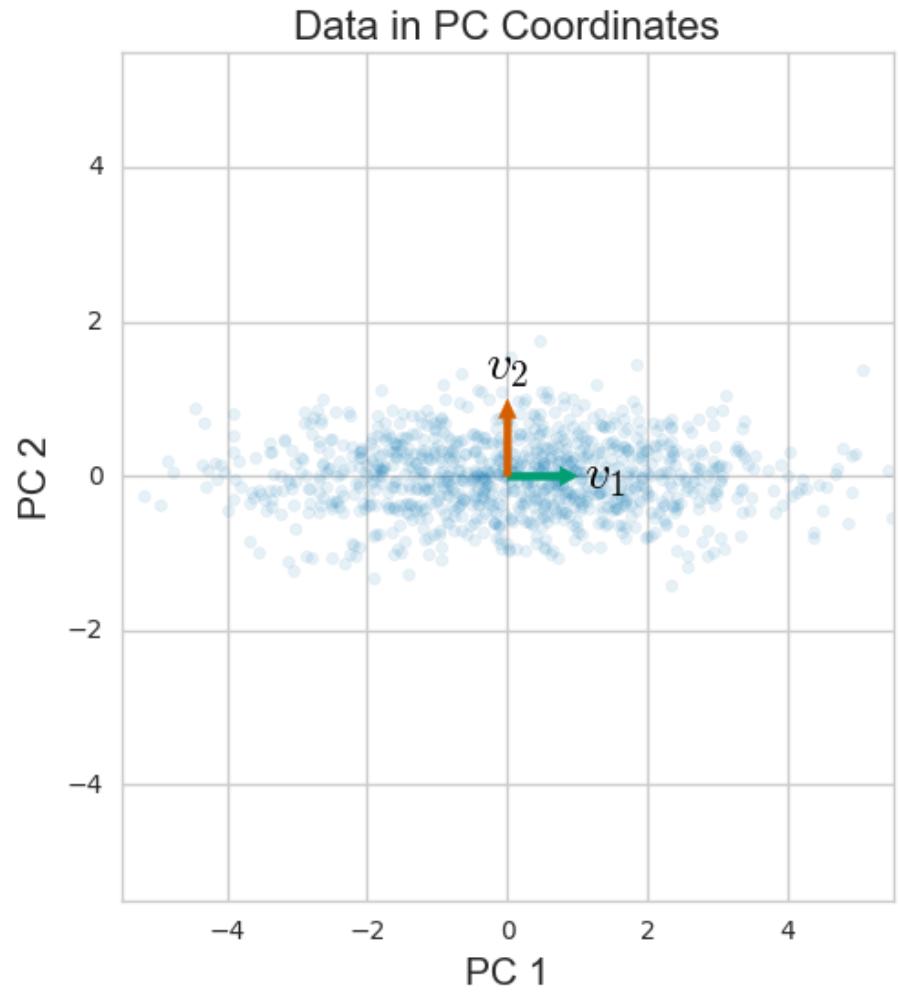
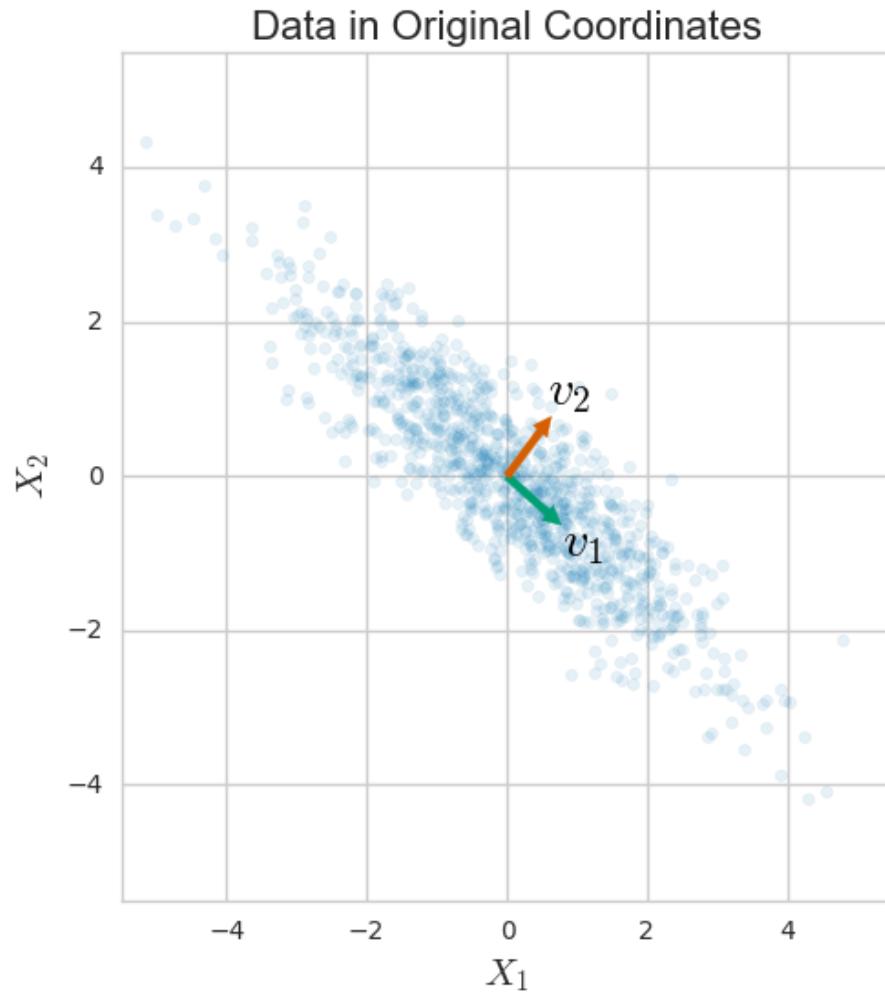
Примечание: Данная особенность раскрывается в теореме о низкоранговом приближении. Рассмотрите теорему при $k=1$ (ранг приближаемой матрицы)

2. Сформулируйте задачу рекомендации товаров. Как матричные разложения помогают решить данную задачу? Какие проблемы возникают в процессе ее решения? Как эти проблемы предлагалось решить?
3. Что такое ALS? В чем концептуальное отличие от GD/SGD? Сформулируйте основные шаги оптимизации методом ALS. В чем преимущества и недостатки?

ИСТОЧНИКИ

1. Курс “Линейная алгебра” ФКН ПМИ 2017/2018 Авдеев Р.С.
2. Изображения взяты: Конспект курса “Матричные разложения” МФТИ
3. Elizabeth Veronica Vera Cervantes “Performance of Alternating Least Squares in a distributed approach using GraphLab and MapReduce”, National University of San Agustin
4. Y Koren, R Bell, and C Volinsky. 2009. “Matrix factorization techniques for recommender systems”.

What is PCA?



PCA

Step 1

- Mean normalize every feature of the dataset.

| Student | Math | English | Art |
|---------|------|---------|-----|
| 1 | 90 | 60 | 90 |
| 2 | 90 | 90 | 30 |
| 3 | 60 | 60 | 60 |
| 4 | 60 | 60 | 90 |
| 5 | 30 | 30 | 30 |

$$\mathbf{A} = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix}$$

$$\bar{\mathbf{A}} = [66 \ 60 \ 60]$$

PCA

Step 2

- Compute the covariance matrix of the dataset

| | <i>Math</i> | <i>English</i> | <i>Art</i> |
|----------------|-------------|----------------|------------|
| <i>Math</i> | 504 | 360 | 180 |
| <i>English</i> | 360 | 360 | 0 |
| <i>Art</i> | 180 | 0 | 720 |

$$cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{x})(Y_i - \bar{y})$$

PCA

Step 3

- Compute Eigenvectors and corresponding Eigenvalues

$$\lambda \approx 44.81966\dots, \lambda \approx 629.11039\dots, \lambda \approx 910.06995\dots$$

Eigenvalues

$$\det \begin{pmatrix} 504 - \lambda & 360 & 180 \\ 360 & 360 - \lambda & 0 \\ 180 & 0 & 720 - \lambda \end{pmatrix}$$

$$\begin{pmatrix} -3.75100\dots \\ 4.28441\dots \\ 1 \end{pmatrix}, \begin{pmatrix} -0.50494\dots \\ -0.67548\dots \\ 1 \end{pmatrix}, \begin{pmatrix} 1.05594\dots \\ 0.69108\dots \\ 1 \end{pmatrix}$$

Eigenvectors

PCA

Step 4

- Choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix W .

$$\lambda \approx 44.81966\dots, \lambda \approx 629.11039\dots, \lambda \approx 910.06995\dots$$

$$W = \begin{bmatrix} 1.05594 & -0.50494 \\ 0.69108 & -0.67548 \\ 1 & 1 \end{bmatrix}$$

Eigenvalues

$$\begin{pmatrix} -3.75100\dots \\ 4.28441\dots \\ 1 \end{pmatrix}, \begin{pmatrix} -0.50494\dots \\ -0.67548\dots \\ 1 \end{pmatrix}, \begin{pmatrix} 1.05594\dots \\ 0.69108\dots \\ 1 \end{pmatrix}$$

Eigenvectors

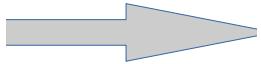
PCA

Step 5

- Transform the samples onto the new subspace

$$A' = AW$$

```
[ [ 24.    0.   30.]  
[ 24.   30.  -30.]  
[ -6.    0.    0.]  
[ -6.    0.   30.]  
[ -36.  -30.  -30. ]]
```



```
[ [ 55.34256  17.88144]  
[ 16.07496 -62.38296]  
[ -6.33564  3.02964]  
[ 23.66436  33.02964]  
[ -88.74624  8.44224] ]
```

SVD and PCA connection

$$XX^T = WDW^T$$

SVD and PCA connection

$$XX^T = WDW^T$$

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

SVD and PCA connection

$$XX^T = WDW^T$$

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

$$XX^T = (U\Sigma V^T)(V\Sigma U^T)$$

SVD and PCA connection

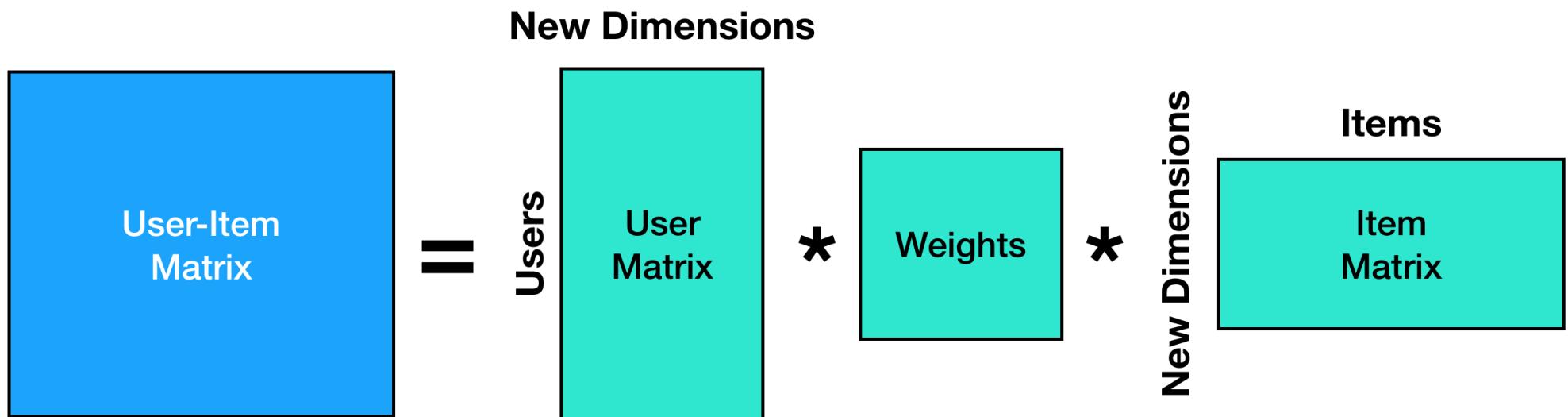
$$XX^T = WDW^T$$

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

$$XX^T = (U\Sigma V^T)(V\Sigma U^T)$$

$$XX^T = U\Sigma^2 U^T$$

SVD. Recommender systems



Problem Statement

- Predict a rating for a user item pair based on the history of ratings given by the user and given to the item

q_i – vector presentation of an item

p_u – vector presentation of a user

expected rating = $\hat{r}_{ui} = q_i^T p_u$

$$\text{minimum}(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u)^2$$

Modifying minimization equation

- Regularization:

$$\text{minimum}(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

Modifying minimization equation

- Regularization:

$$\text{minimum}(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

Bias terms:

$$\hat{r}_{ui} = q_i^T \cdot p_u + \mu + b_i + b_u$$

Modifying minimization equation

- Regularization:

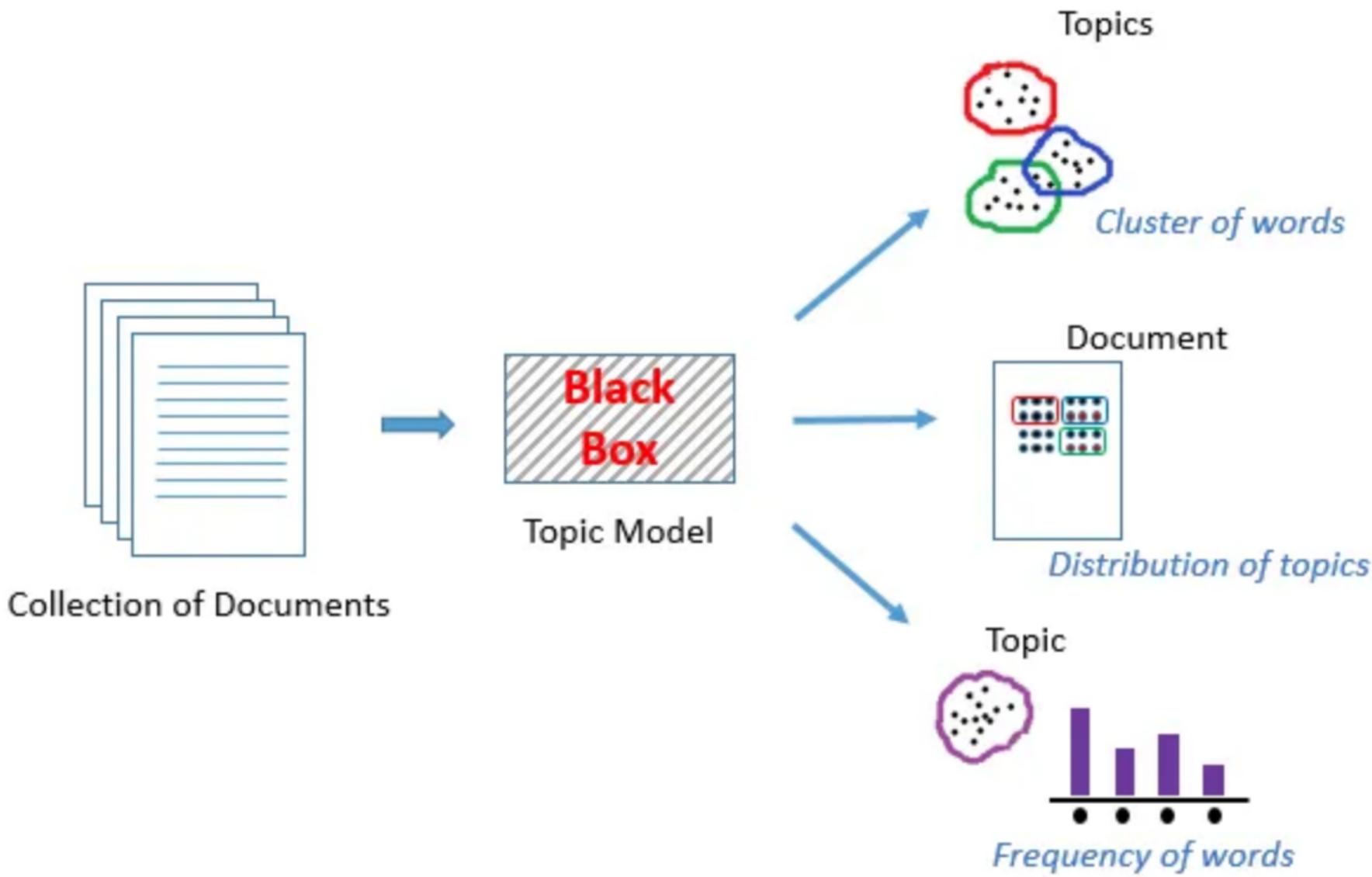
$$\text{minimum}(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

Bias terms:

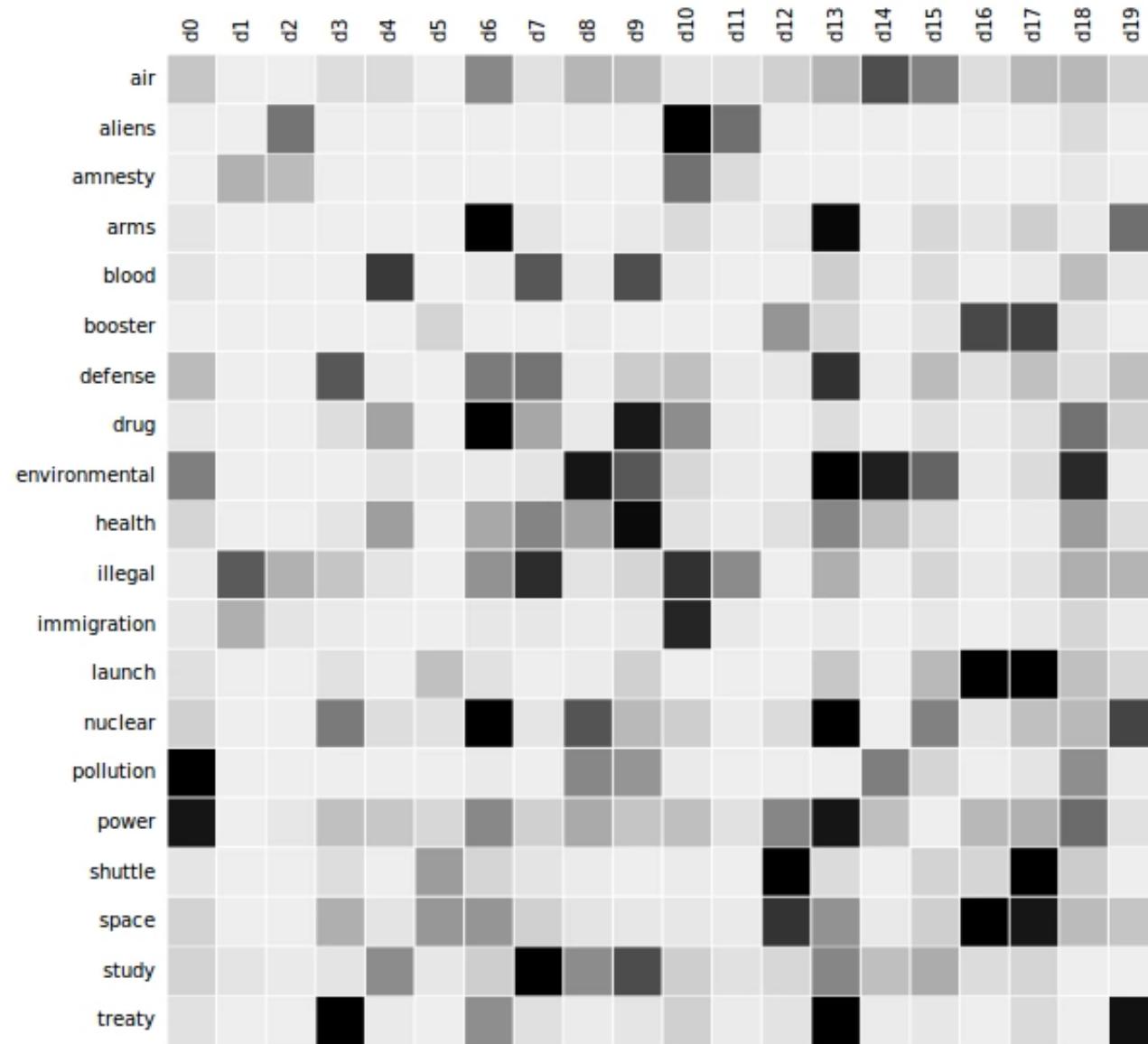
$$\hat{r}_{ui} = q_i^T \cdot p_u + \mu + b_i + b_u$$

$$\text{minimum}(p, q, b_i, b_u) \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u - \mu - b_i - b_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2)$$

Topic modeling



Basic approach. Document-word matrix



SVD approach

Step 1

Generate a document-term matrix of shape $m \times n$ having TF-IDF scores.

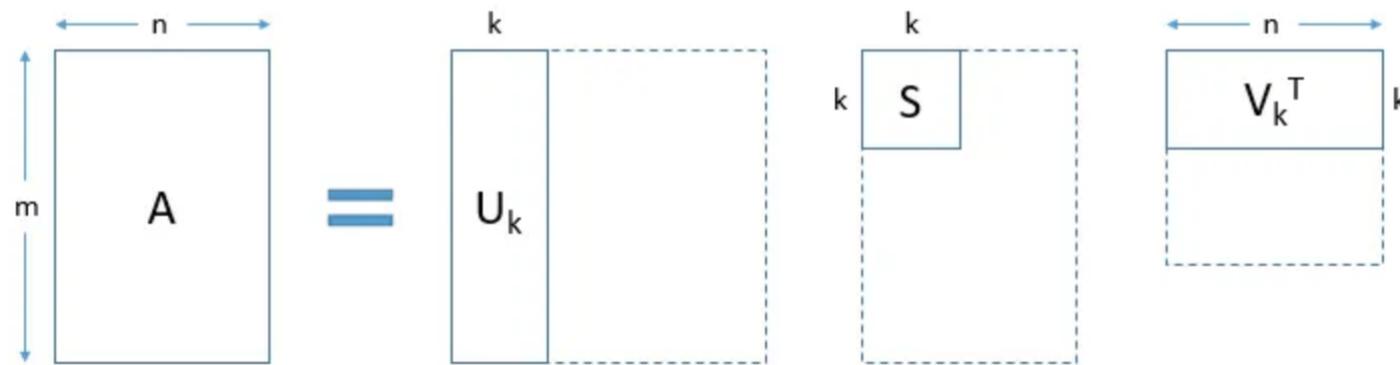
Documents

| | Terms | | | | | |
|-----|-------|-----|-----|-----|-----|-----|
| | T1 | T2 | T3 | ... | Tn | |
| D1 | 0.2 | 0.1 | 0.5 | ... | 0.1 | |
| D2 | 0.1 | 0.3 | 0.4 | ... | 0.3 | |
| D3 | 0.3 | 0.1 | 0.1 | ... | 0.5 | |
| ... | ... | ... | ... | ... | ... | ... |
| Dm | 0.2 | 0.1 | 0.2 | ... | 0.1 | |

SVD approach.

Step 2

$$A = USV^T$$



- K - the number of desired topics
- U row - vector representation of a document
- V column – vector representation of a word

SVD Approach.

Step 3

- Find similar words and similar documents using the cosine similarity method.

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

Background removal

Original image



Image without background



Background removal

Step 1

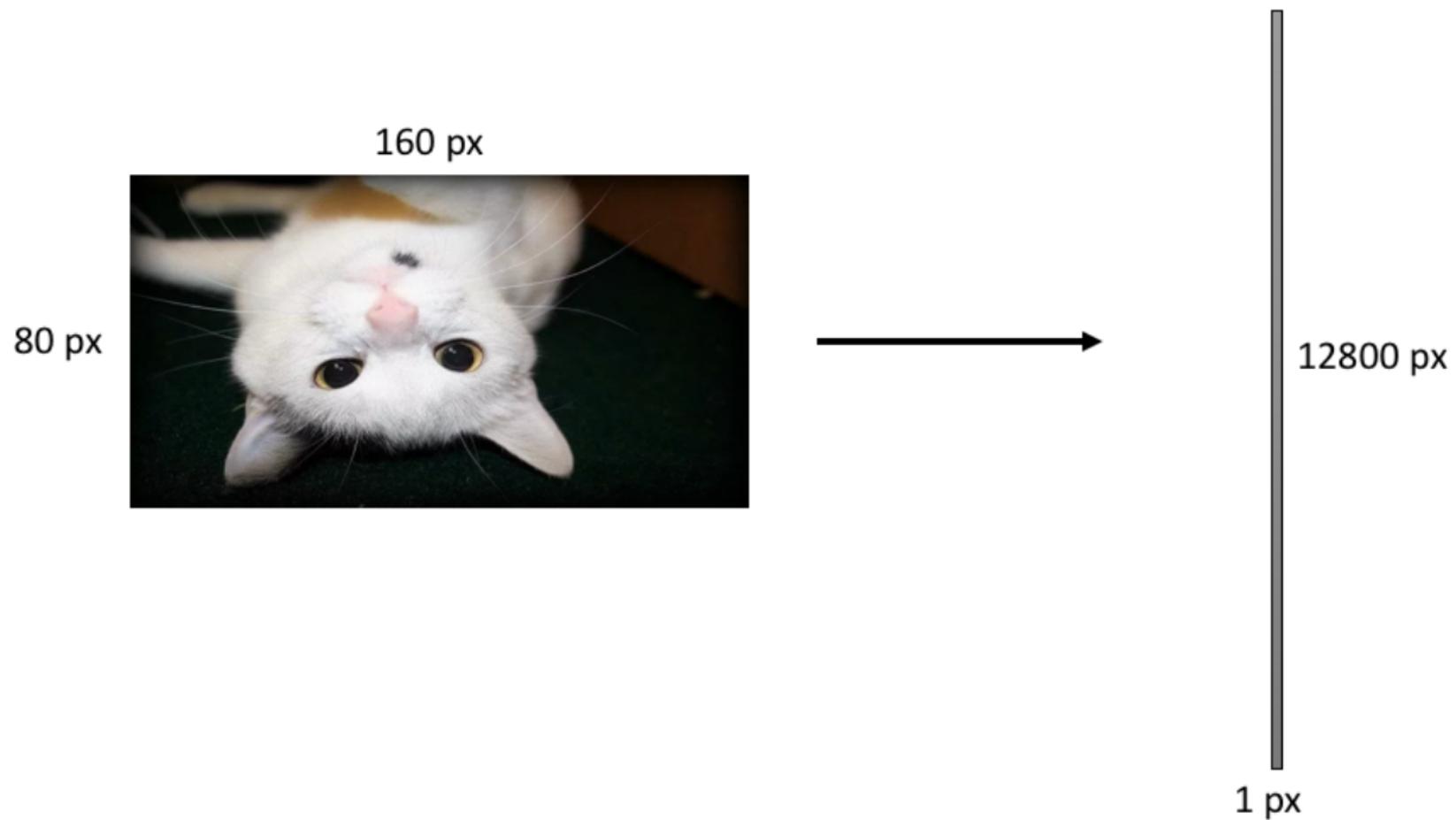
- Split video into frames based on some predefined frame per second



Background removal

Step 2

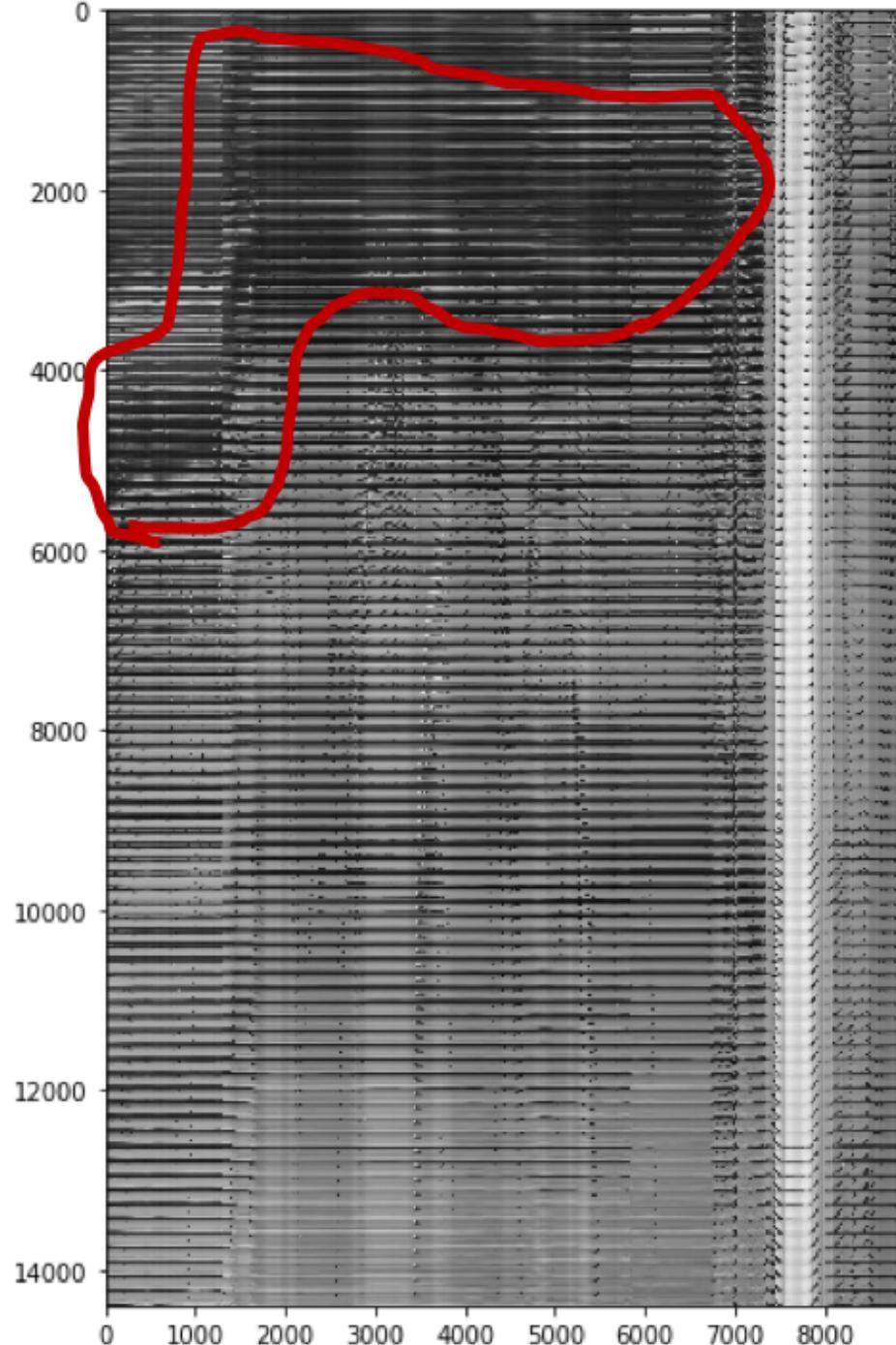
- Flatten each frame (convert to 1-D vector).
Matrix of video with images as 1-D vector



Background removal

Step 3

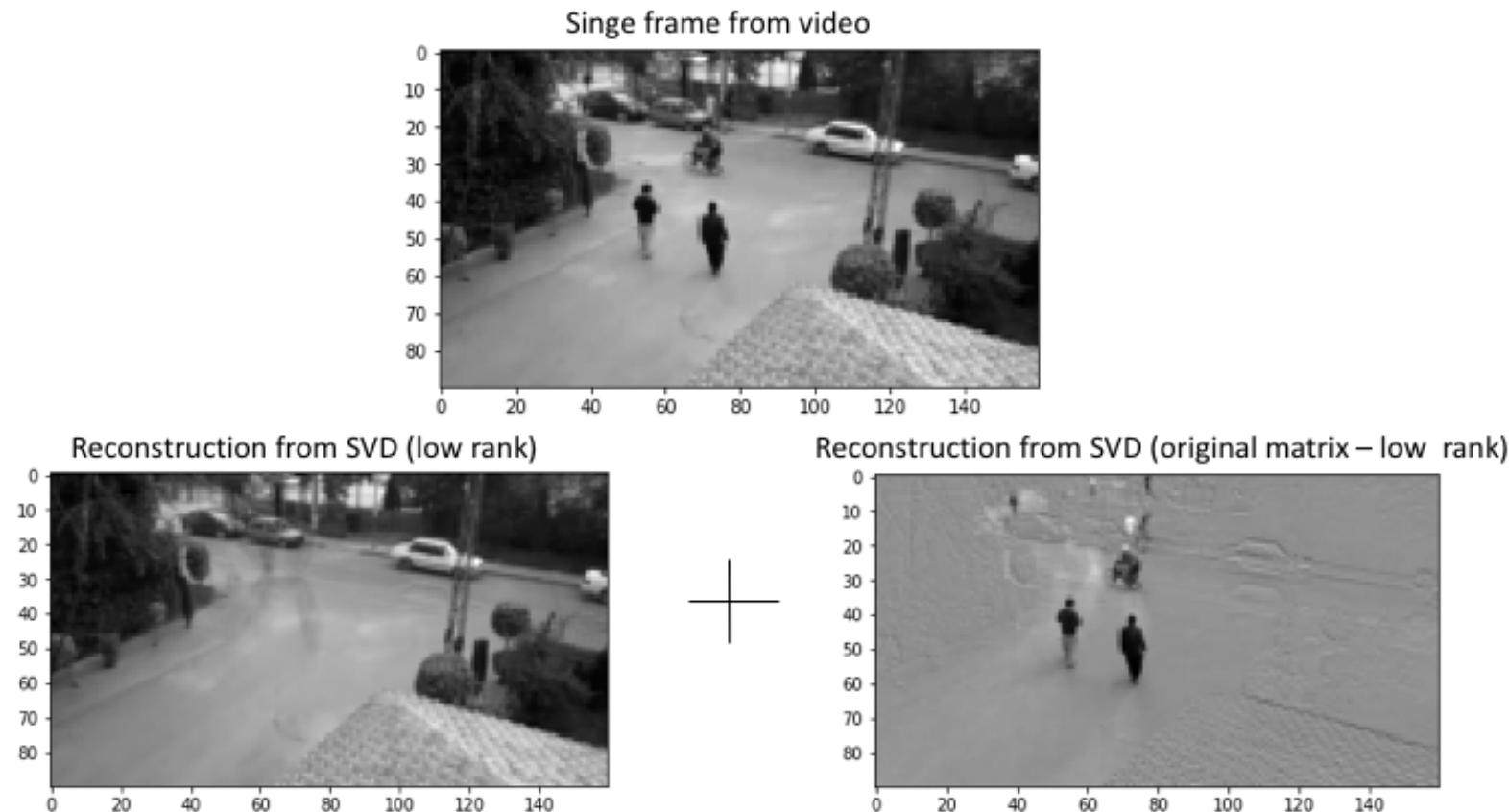
- Do step 2 for all the frames and concatenate resultant 1-D vectors side by side



Background removal

Step 4

- Decompose the matrix from step 3 using SVD. The foreground is the difference between the original matrix and the lower rank matrix



Cholesky decomposition

$$A = LL^*$$

- L is a lower triangular matrix with real and positive diagonal entries
- L^* denotes the Hermitian transpose of L
- The Cholesky decomposition is unique when A is positive definite

The Cholesky–Banachiewicz algorithm

$$\begin{aligned}\mathbf{A} = \mathbf{LL}^T &= \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} L_{11} & L_{21} & L_{31} \\ 0 & L_{22} & L_{32} \\ 0 & 0 & L_{33} \end{pmatrix} \\ &= \begin{pmatrix} L_{11}^2 & & & (\text{symmetric}) \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{pmatrix}\end{aligned}$$

The Cholesky–Banachiewicz algorithm

$$\begin{aligned}\mathbf{A} = \mathbf{LL}^T &= \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} L_{11} & L_{21} & L_{31} \\ 0 & L_{22} & L_{32} \\ 0 & 0 & L_{33} \end{pmatrix} \\ &= \begin{pmatrix} L_{11}^2 & & & (\text{symmetric}) \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{pmatrix}\end{aligned}$$

$$\mathbf{L} = \begin{pmatrix} \sqrt{A_{11}} & 0 & 0 \\ A_{21}/L_{11} & \sqrt{A_{22} - L_{21}^2} & 0 \\ A_{31}/L_{11} & (A_{32} - L_{31}L_{21})/L_{22} & \sqrt{A_{33} - L_{31}^2 - L_{32}^2} \end{pmatrix}$$

The Cholesky–Banachiewicz algorithm

$$L_{j,j} = \sqrt{A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2},$$
$$L_{i,j} = \frac{1}{L_{j,j}} \left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k} L_{j,k} \right) \quad \text{for } i > j.$$

The expression under the square root is always positive if A is real and positive-definite

System of linear equations

$$\begin{pmatrix} A_{11} & A_{21}^* & A_{31}^* & A_{41}^* \\ A_{21} & A_{22} & A_{32}^* & A_{42}^* \\ A_{31} & A_{32} & A_{33} & A_{43}^* \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

System of linear equations

$$\begin{pmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{pmatrix} * \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

$$\begin{pmatrix} L_{11} & L^*_{21} & L^*_{31} & L^*_{41} \\ 0 & L_{22} & L^*_{32} & L^*_{42} \\ 0 & 0 & L_{33} & L^*_{43} \\ 0 & 0 & 0 & L_{44} \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Least-squares regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Least-squares regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$X^T X = LL^T$$

Least-squares regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$X^T X = LL^T$$

$$LL^T \hat{\beta} = X^T Y$$

Least-squares regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$X^T X = LL^T$$

$$LL^T \hat{\beta} = X^T Y$$

1. Solve for $x = L^T \beta$

Least-squares regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

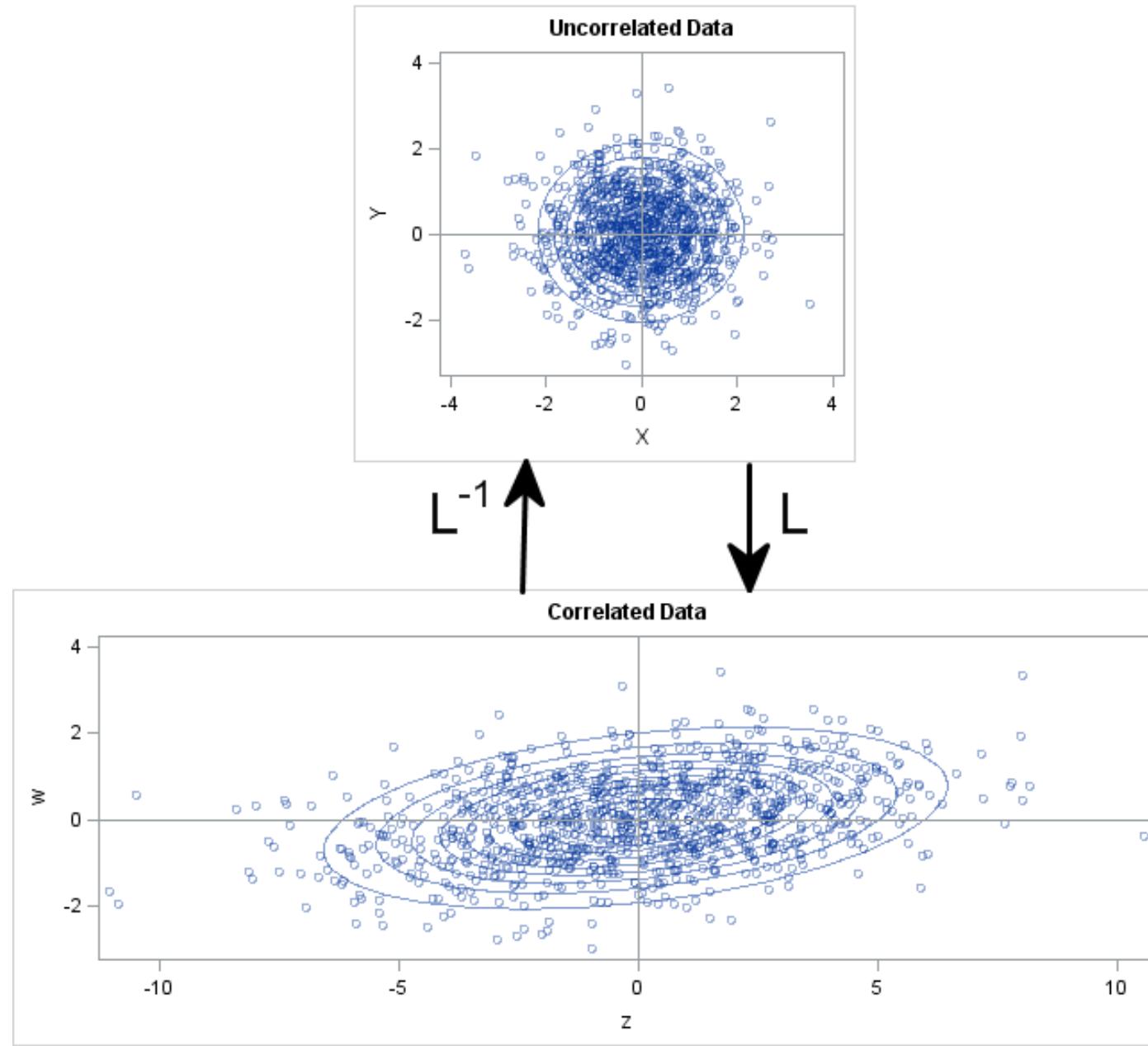
$$X^T X = LL^T$$

$$LL^T \hat{\beta} = X^T Y$$

1. *Solve for $x = L^T \beta$*

2. *Solve for β*

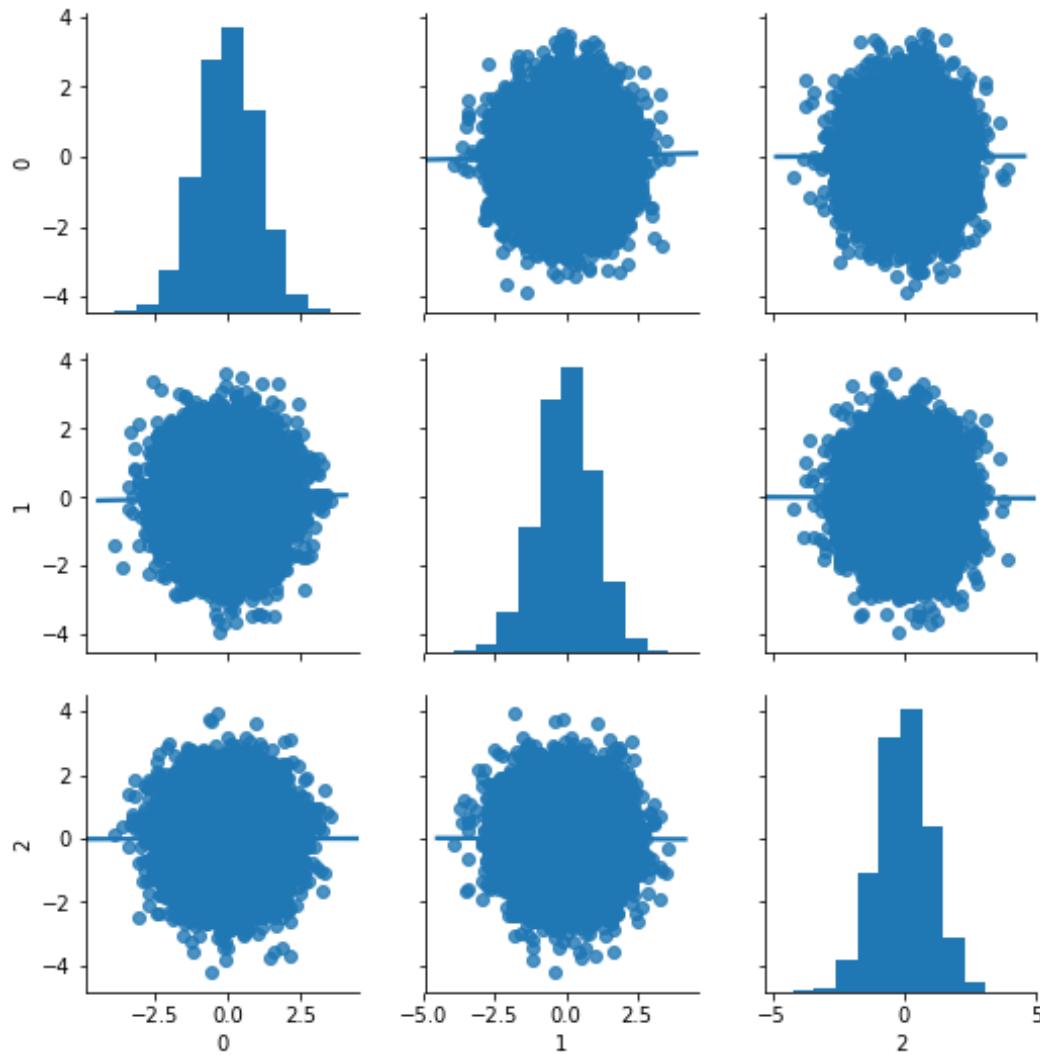
Monte-Carlo Simulation



Monte-Carlo Simulation.

Step 1

- Generate uncorrelated normal variables:



Monte-Carlo Simulation.

Step 2

- Calculate covariance matrix

$$\text{Covariance Matrix} = \begin{bmatrix} 10.0 & -2.0 & 2.0 \\ -2.0 & 20.0 & 0.5 \\ 2.0 & 0.5 & 0.5 \end{bmatrix}$$

Monte-Carlo Simulation.

Step 3

- Apply Cholesky Decomposition to covariance matrix

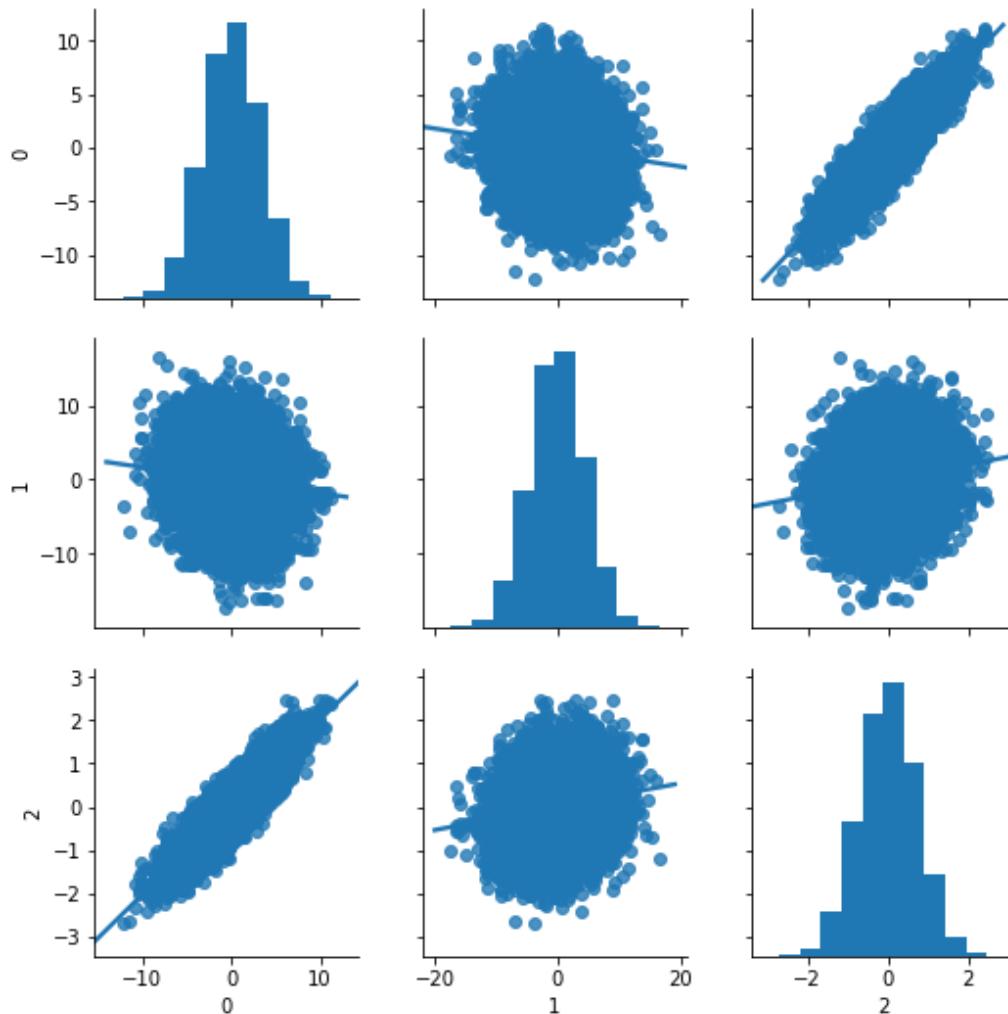
$$\begin{bmatrix} 10. & -2. & 2. \\ -2. & 20. & 0.5 \\ 2. & 0.5 & 0.5 \end{bmatrix}$$
$$\begin{bmatrix} 3.162 & 0. & 0. \\ -0.632 & 4.427 & 0. \\ 0.632 & 0.203 & 0.242 \end{bmatrix}$$
$$\begin{bmatrix} 3.162 & 0. & 0. \\ -0.632 & 4.427 & 0. \\ 0.632 & 0.203 & 0.242 \end{bmatrix}$$

$$\begin{bmatrix} 3.162 & 0. & 0. \\ -0.632 & 4.427 & 0. \\ 0.632 & 0.203 & 0.242 \end{bmatrix}$$
$$\begin{bmatrix} 3.162 & 0. & 0. \\ -0.632 & 4.427 & 0. \\ 0.632 & 0.203 & 0.242 \end{bmatrix}$$
$$\begin{bmatrix} 3.162 & 0. & 0. \\ -0.632 & 4.427 & 0. \\ 0.632 & 0.203 & 0.242 \end{bmatrix}$$

Monte-Carlo Simulation.

Step 4

- Multiply the matrix of uncorrelated random variables and L matrix



Non-negative matrix factorization

$$V \simeq WH$$

$$W, H = \underset{W, H}{\operatorname{argmin}} \|V - WH\|_F$$

$$W \geq 0, \quad H \geq 0$$

The optimization methods mentioned earlier are not suitable for that problem

Problems with finding Non-negative matrix factorization

$$(W^*, H^*) = \underset{W \geq 0, H \geq 0}{\operatorname{argmin}} \|X - WH\|_F^2.$$

GD can't be applied since there is limitation

Problems with finding Non-negative matrix factorization

$$(W^*, H^*) = \underset{W \geq 0, H \geq 0}{\operatorname{argmin}} \|X - WH\|_F^2.$$

GD can't be applied since there is limitation

Idea! $h_{kj} \leftarrow h_{kj} - \nu_{kj} \frac{\partial D_F}{\partial h_{kj}}, w_{ik} \leftarrow w_{ik} - \eta_{ik} \frac{\partial D_F}{\partial w_{ik}}. k = 1, \dots, r,; \quad i = 1, \dots, m,; \quad j = 1, \dots, n$

Problems with finding Non-negative matrix factorization

$$(W^*, H^*) = \underset{W \geq 0, H \geq 0}{\operatorname{argmin}} \|X - WH\|_F^2.$$

GD can't be applied since there is limitation

Idea! $h_{kj} \leftarrow h_{kj} - \nu_{kj} \frac{\partial D_F}{\partial h_{kj}}, w_{ik} \leftarrow w_{ik} - \eta_{ik} \frac{\partial D_F}{\partial w_{ik}}. k = 1, \dots, r,; \quad i = 1, \dots, m,; \quad j = 1, \dots, n$

Problem: The limitations are not fulfilled

Deriving Multiplicative Update Rules for NMF

$$\frac{\partial D_F}{\partial h_{kj}} = \sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij},$$

$$D_F(X, \hat{X}) = \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - \hat{x}_{ij})^2 \equiv \|X - WH\|_F^2$$

Deriving Multiplicative Update Rules for NMF

$$\frac{\partial D_F}{\partial h_{kj}} = \sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij},$$

$$\nu_{kj} = \frac{h_{kj}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}}$$

$$D_F(X, \hat{X}) = \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - \hat{x}_{ij})^2 \equiv \|X - WH\|_F^2$$

Deriving Multiplicative Update Rules for NMF

$$\frac{\partial D_F}{\partial h_{kj}} = \sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij},$$

$$\nu_{kj} = \frac{h_{kj}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}}$$
$$h_{kj} \leftarrow h_{kj} - \frac{h_{kj}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}} \left(\sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij} \right)$$

$$D_F(X, \hat{X}) = \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - \hat{x}_{ij})^2 \equiv \|X - WH\|_F^2$$

Deriving Multiplicative Update Rules for NMF

$$\frac{\partial D_F}{\partial h_{kj}} = \sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij},$$

$$\nu_{kj} = \frac{h_{kj}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}}$$

$$h_{kj} \leftarrow h_{kj} - \frac{h_{kj}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}} \left(\sum_{i=1}^m w_{ik} \hat{x}_{ij} - \sum_{i=1}^m w_{ik} x_{ij} \right)$$

$$h_{kj} \leftarrow h_{kj} \frac{\sum_{i=1}^m w_{ik} x_{ij}}{\sum_{i=1}^m w_{ik} \hat{x}_{ij}}.$$

If the origin matrices are properly initialized, then they will remain positive

Other methods of optimization

ALS:

$$H \leftarrow \max \left(\underset{H}{\operatorname{argmin}} \|X - WH\|_F^2, 0 \right) = \max \left((W^T W)^{-1} W^T X, 0 \right)$$

Other methods of optimization

ALS:

$$H \leftarrow \max \left(\operatorname{argmin}_H \|X - WH\|_F^2, 0 \right) = \max \left((W^T W)^{-1} W^T X, 0 \right)$$

ANLS:

$$H \leftarrow \operatorname{argmin}_{H \geq 0} \|X - WH\|_F^2$$

Other methods of optimization

ALS:

$$H \leftarrow \max \left(\operatorname{argmin}_H \|X - WH\|_F^2, 0 \right) = \max \left((W^T W)^{-1} W^T X, 0 \right)$$

ANLS:

$$H \leftarrow \operatorname{argmin}_{H \geq 0} \|X - WH\|_F^2$$

HALS:

$$h_k \leftarrow \operatorname{argmin}_{h_k \geq 0} \|X - WH\|_F^2 = \max \left(0, \frac{w_k^T X - \sum_{l \neq k} w_k^T w_l h_l}{w_k^T w_k} \right)$$

NMF

Loss functions

- L1-norm

$$d_1(x, \hat{x}) = |x - \hat{x}|$$

NMF

Loss functions

- L1-norm

$$d_1(x, \hat{x}) = |x - \hat{x}|$$

- Frobenius norm

$$d_F(x, \hat{x}) = (x - \hat{x})^2$$

NMF

Loss functions

- L1-norm

$$d_1(x, \hat{x}) = |x - \hat{x}|$$

- Frobenius norm

$$d_F(x, \hat{x}) = (x - \hat{x})^2$$

- Kullback–Leibler divergence

$$d_{KL}(x, \hat{x}) = x \ln \frac{x}{\hat{x}} - x + \hat{x}$$

NMF

Loss functions

- L1-norm

$$d_1(x, \hat{x}) = |x - \hat{x}|$$

- Frobenius norm

$$d_F(x, \hat{x}) = (x - \hat{x})^2$$

- Kullback–Leibler divergence

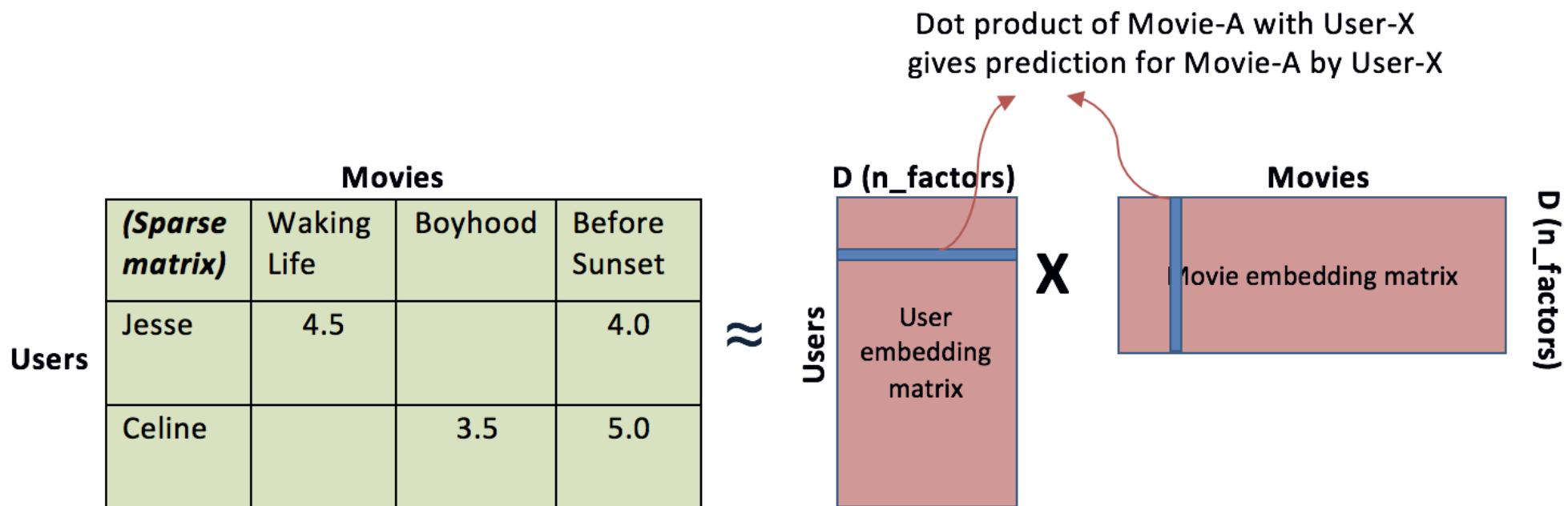
$$d_{KL}(x, \hat{x}) = x \ln \frac{x}{\hat{x}} - x + \hat{x}$$

- Hellinger distance

$$d_H(x, \hat{x}) = \left(\sqrt{\tilde{x}} - \sqrt{\tilde{\hat{x}}} \right)^2$$

NMF

Recommender systems

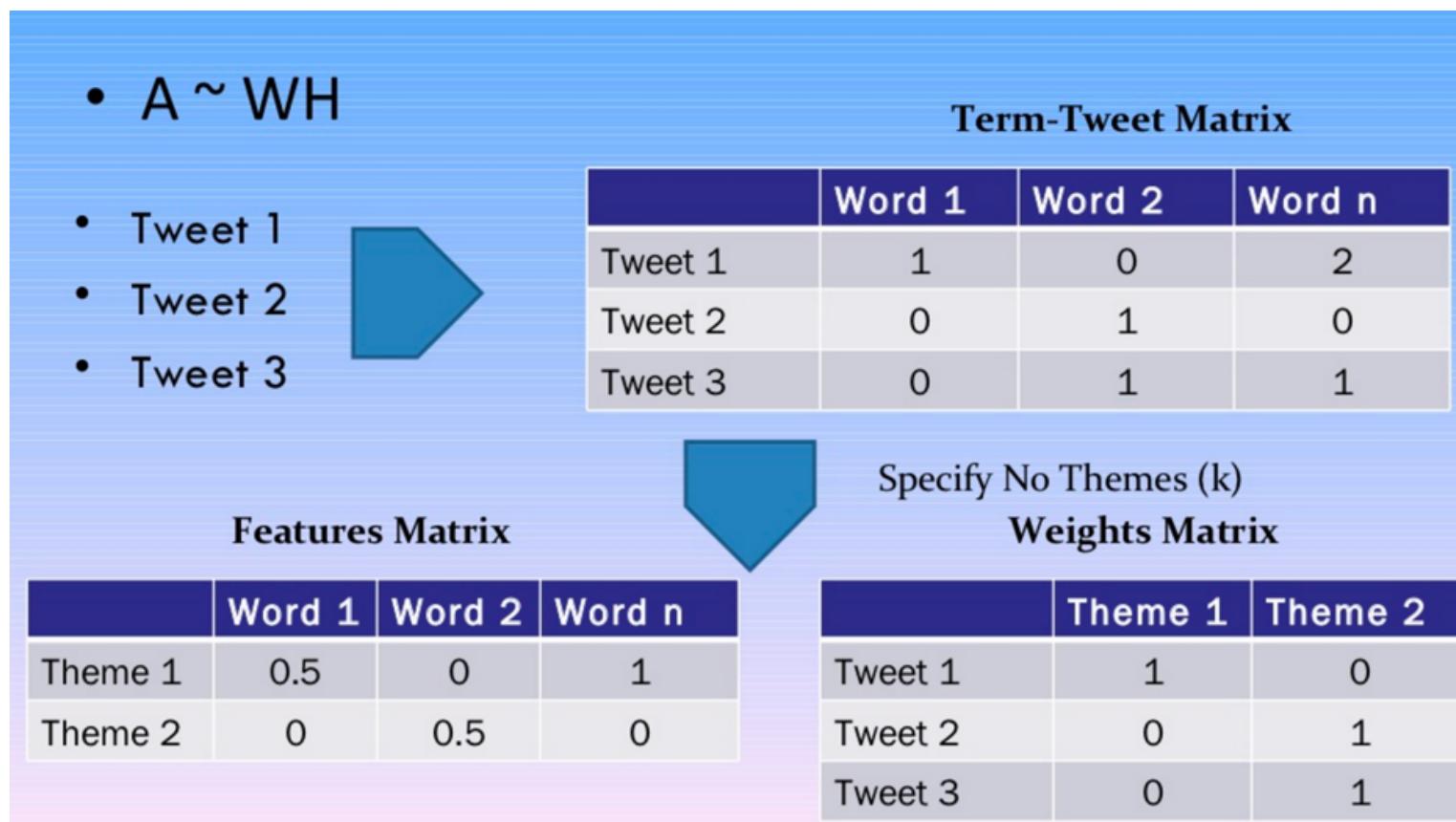


The key advantage of the approach is the interpretability of results

NMF

Topic modeling

- Bag of words text presentation
- Tf-idf text presentation
- NMF



NMF

Topic modeling results

NMF Topics:

Topic 0: people don think like know time right good did say

Topic 1: windows file use dos files window using program problem card

Topic 2: god jesus bible christ faith believe christian christians church sin

Topic 3: drive scsi drives hard disk ide controller floppy cd mac

Topic 4: game team year games season players play hockey win player

Topic 5: key chip encryption clipper keys government escrow public use algorithm

Topic 6: thanks does know mail advance hi anybody info looking help

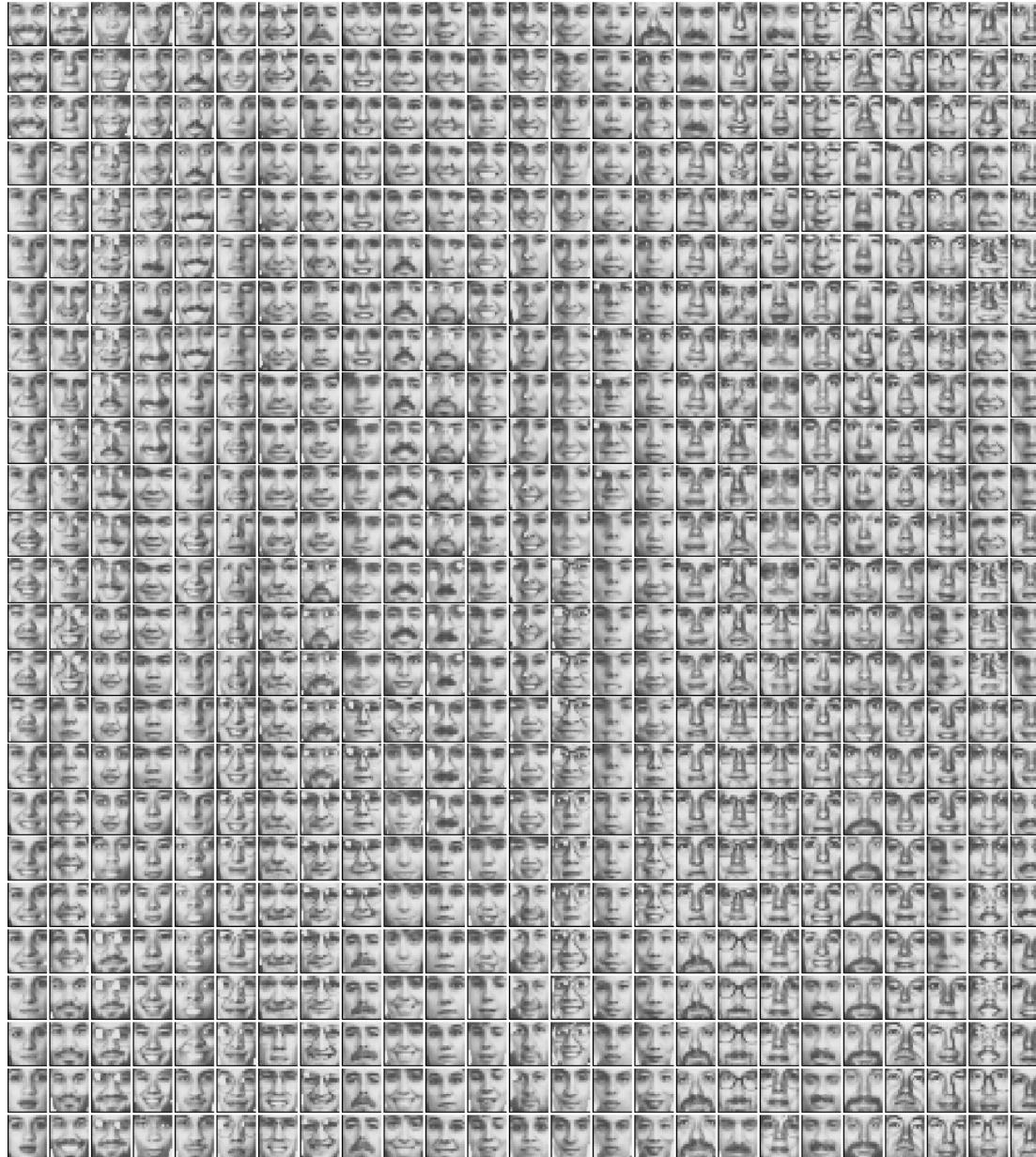
Topic 7: car new 00 sale price 10 offer condition shipping 20

Topic 8: just like don thought ll got oh tell mean fine

Topic 9: edu soon cs university com email internet article ftp send

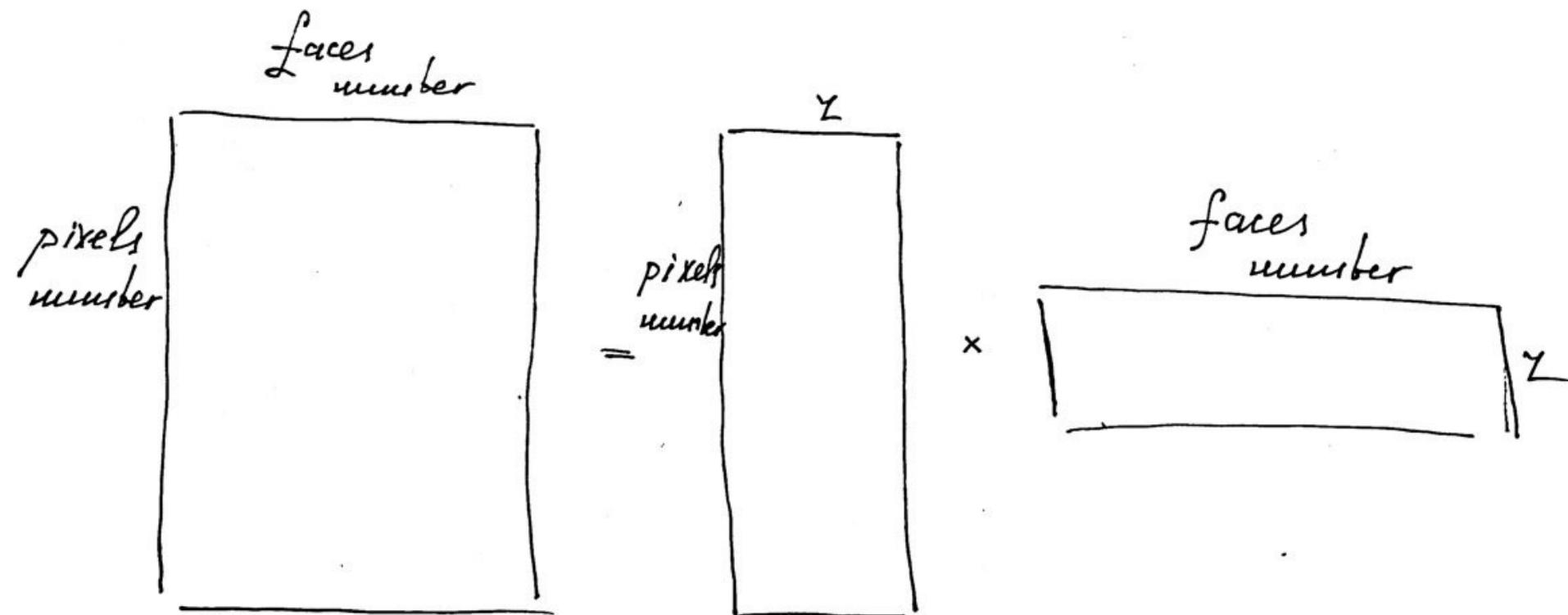
NMF

Decomposing face images



NMF

Decomposing face images



NMF

Decomposing face images

$$\underbrace{X(:, j)}_{j\text{th facial image}} \approx \sum_{k=1}^r \underbrace{W(:, k)}_{\text{facial features}} \underbrace{H(k, j)}_{\substack{\text{importance of features} \\ \text{in } j\text{th image}}} = \underbrace{WH(:, j)}_{\substack{\text{approximation} \\ \text{of } j\text{th image}}} .$$


Контрольные вопросы

- Как связаны между собой SVD и PCA? Как связаны собственные значения матрицы ковариации и сингулярные значения исходной матрицы?
- Укажите асимптотику нахождения разложения Холецкого
- Что такое Non-negative matrix factorization и как его получить? Опишите смысл применения NMF на примере разложения матриц из изображений лиц: что из себя представляют столбец матрицы признаков?