Матричные разложения

(и их применения)

Агапова Ольга, Тимофей Сеньченко

Что такое матричное разложение?

- $A = XY$

Что такое матричное разложение?

- A = XY
- X, Y могут обладать какими-нибудь полезными свойствами

Какие они бывают?

- Их очень много
- Мы расскажем про SVD, NMF, LU, QR, Choletsky

Какие они бывают?

- Их очень много
- Мы расскажем про SVD, NMF, LU, QR, Choletsky
- Можно классифицировать по назначению (часто встречаются МР, полезные для решения СЛУ или для работы с собственными значениями)

SVD

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

(ортогональная) (диагональная) (ортогональная)

SVD

$$A_{(m\times n)} = U_{(m\times m)}\Sigma_{(m\times n)}V^{T}_{(n\times n)}$$

(ортогональная) (диагональная) (ортогональная)

Либо:

$$A_{(m\times n)} = U_{(m\times k)}\Sigma_{(k\times k)}V^{T}_{(k\times n)}$$

$$A = \underbrace{\begin{pmatrix} u_{11} & \dots & u_{1k} \\ \vdots & & \vdots \\ u_{m1} & \dots & u_{mk} \end{pmatrix}}_{=U} \cdot \underbrace{\begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{pmatrix}}_{=\Sigma} \cdot \underbrace{\begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & & \vdots \\ v_{1k} & \dots & v_{nk} \end{pmatrix}}_{=V^T}$$

$$= u_1 \sigma_1 v_1^T + u_2 \sigma_2 v_2^T + u_3 \sigma_3 v_3^T + \dots$$

$$A = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & & \vdots \\ u_{m1} & \cdots & u_{mk} \end{pmatrix}}_{=U} \cdot \underbrace{\begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{pmatrix}}_{=\Sigma} \cdot \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{n1} \\ \vdots & & \vdots \\ v_{1k} & \cdots & v_{nk} \end{pmatrix}}_{=V^T}$$

$$A_r = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & & \vdots \\ u_{m1} & \cdots & u_{mr} \end{pmatrix}}_{=U_r} \cdot \underbrace{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix}}_{=\Sigma_r} \cdot \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{n1} \\ \vdots & & \vdots \\ v_{1r} & \cdots & v_{nr} \end{pmatrix}}_{=V_r^T}$$

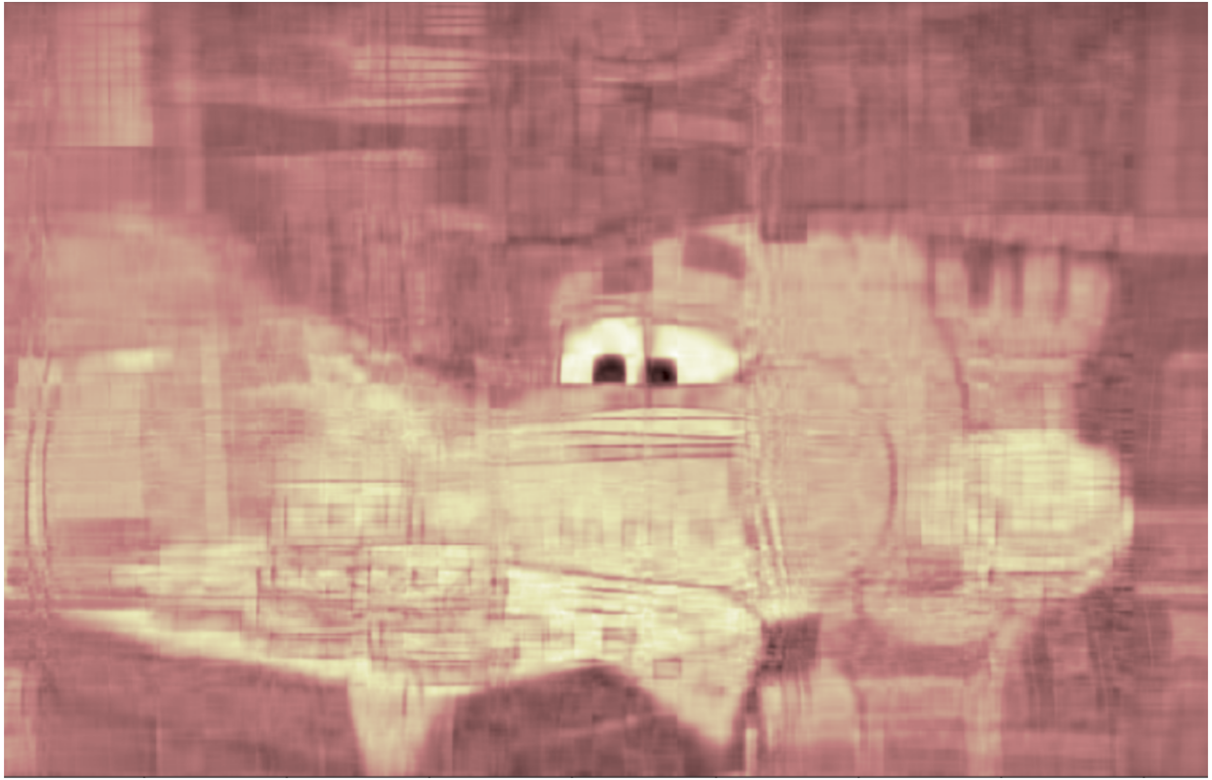
$$A_r = \underbrace{\begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & & \vdots \\ u_{m1} & \dots & u_{mr} \end{pmatrix}}_{=U_r} \cdot \underbrace{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix}}_{=\Sigma_r} \cdot \underbrace{\begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & & \vdots \\ v_{1r} & \dots & v_{nr} \end{pmatrix}}_{=V_r^T}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T$$

это - сумма матриц ранга 1

Вспомним пример из лабы в конце 1 курса:

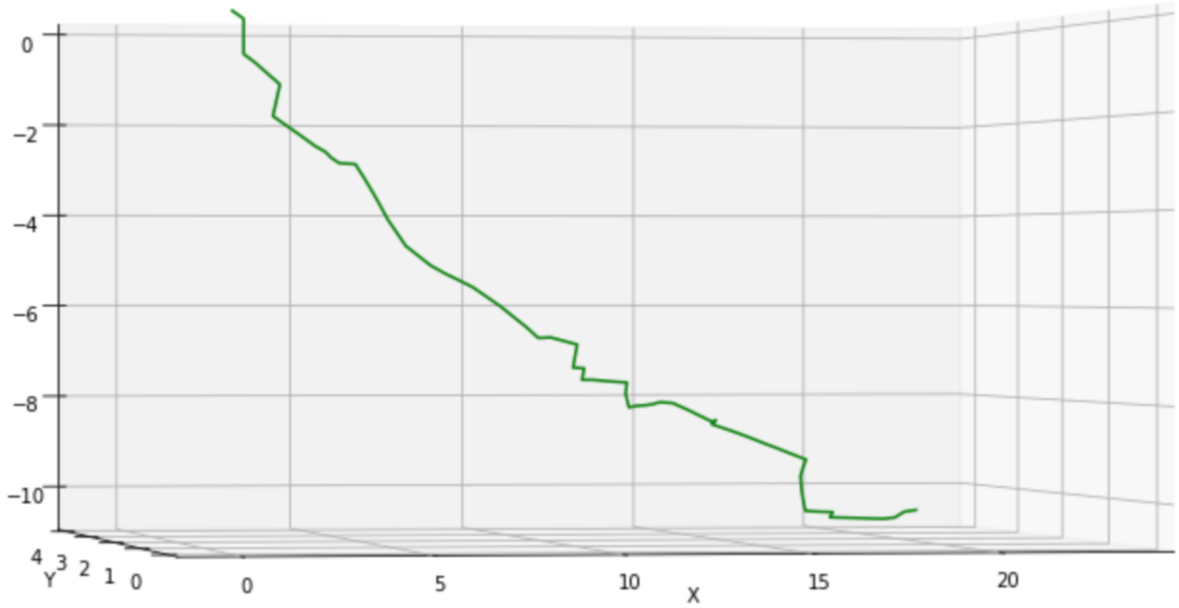
$$A_{20} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_{20} u_{20} v_{20}^T$$



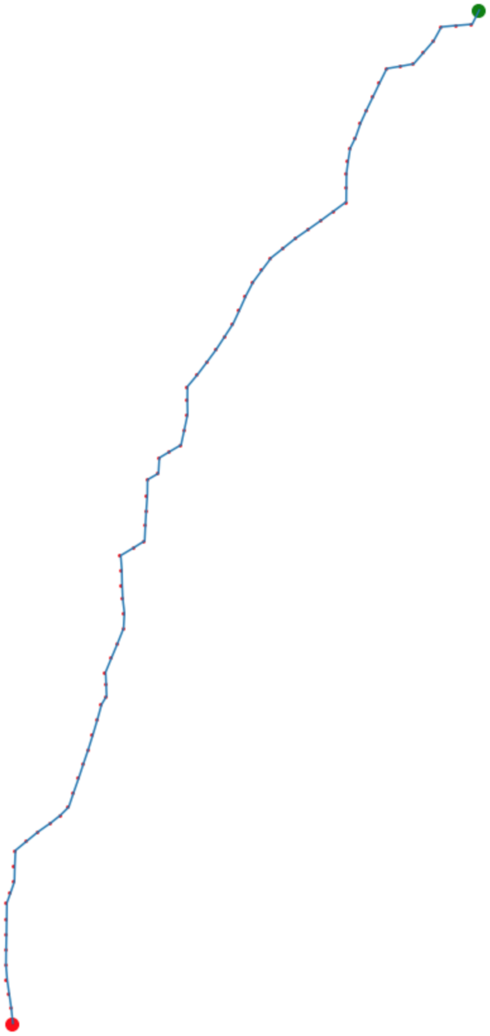
$$A_{40} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_{40} u_{40} v_{40}^T$$

$$A_{100} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_{100} u_{100} v_{100}^T$$

Траектория, не ориентированная в пространстве:

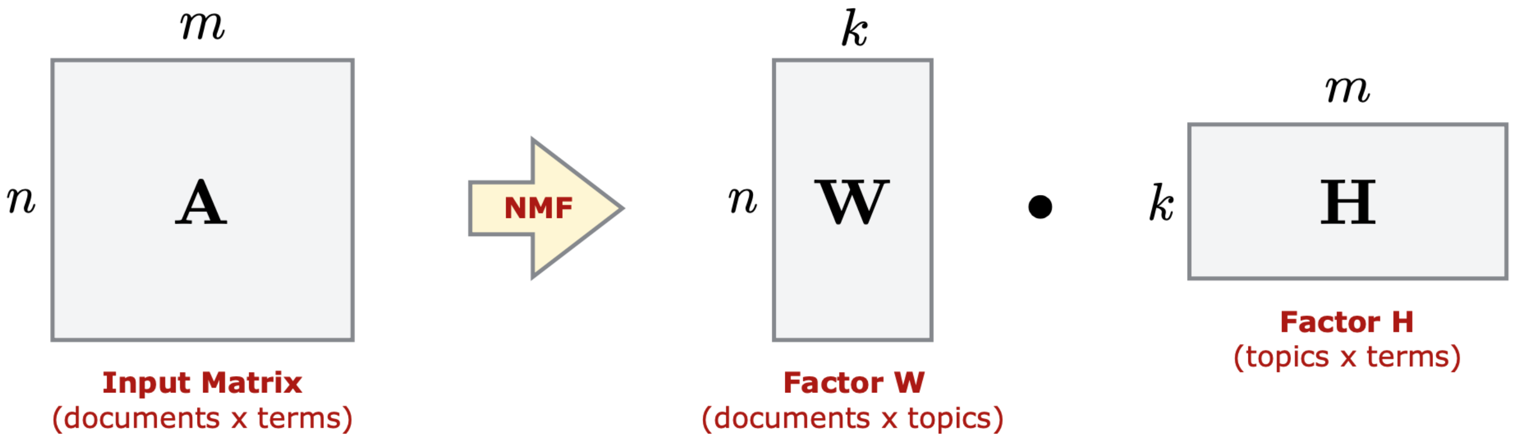


хотим правильно выбрать две оси, вдоль которых происходит изменение (здесь это X и Z, но это не соответствует ground truth)



- Точки в пространстве - объекты
- Оси координат - признаки
- SVD на матрицу признаки-объекты
- Первые два сингулярных значения соответствовали основным осям

Non-Negative Matrix Factorization (NNMF)



Non-Negative Matrix Factorization (NNMF)

- ***k*** задается в зависимости от задачи

Non-Negative Matrix Factorization (NNMF)

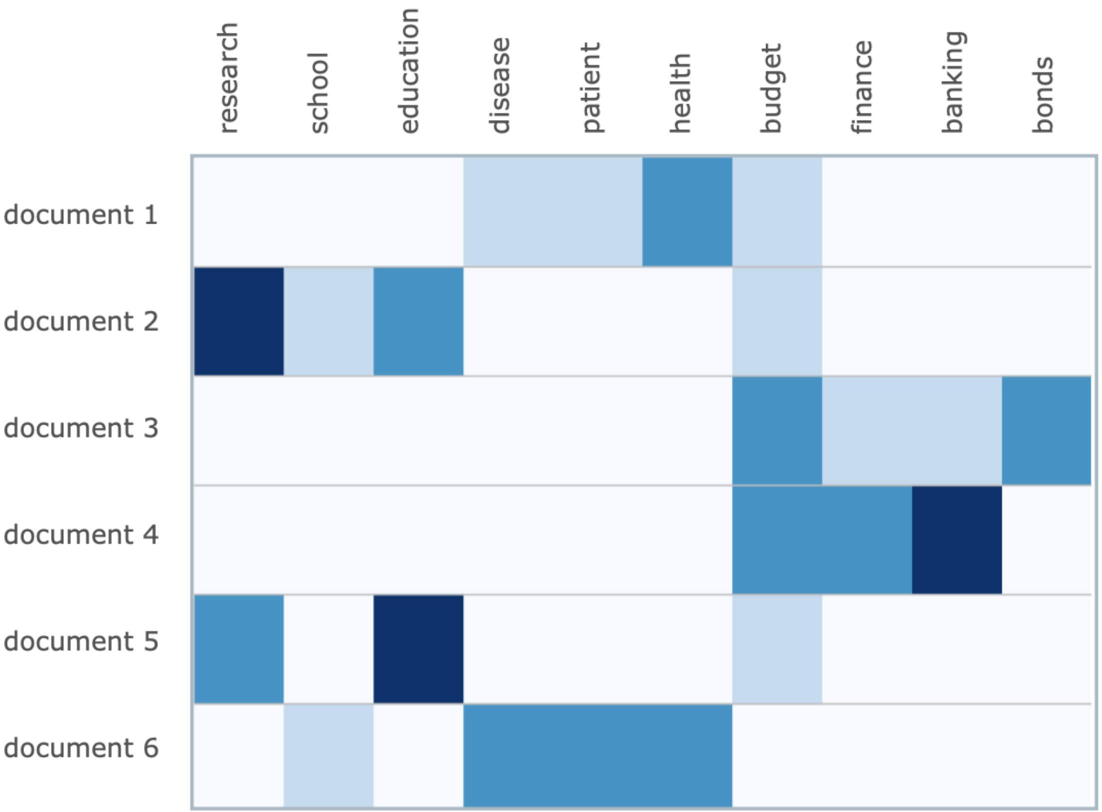
- ***k*** задается в зависимости от задачи
- Хорошее уточнение: столбцы **W** могут быть *выпуклыми комбинациями* входных векторов (объектов)

Non-Negative Matrix Factorization (NNMF)

- ***k*** задается в зависимости от задачи
- Хорошее уточнение: столбцы **W** могут быть *выпуклыми комбинациями* входных векторов (объектов)
- Вычисление зависит от способа минимизации нормы Фробениуса

Non-Negative Matrix Factorization (NNMF)

- ***k*** задается в зависимости от задачи
- Хорошее уточнение: столбцы **W** могут быть *выпуклыми комбинациями* входных векторов (объектов)
- Вычисление зависит от способа минимизации нормы Фробениуса
- Не единственно (существует подходящая обратимая матрица **B**, чтобы было $\mathbf{WB} \times \mathbf{B}^{-1}\mathbf{H}$)



6 Documents x 10 Terms

Factor **W**
Weights for 6 documents relative to 3 topics

	Topic 1	Topic 2	Topic 3
document 1	0.0	1.0	1.0
document 2	0.0	0.0	1.0
document 3	0.7	0.0	0.0
document 4	0.7	0.0	0.0
document 5	0.0	0.0	1.0
document 6	0.0	1.0	1.0

6 Rows x 3 Columns

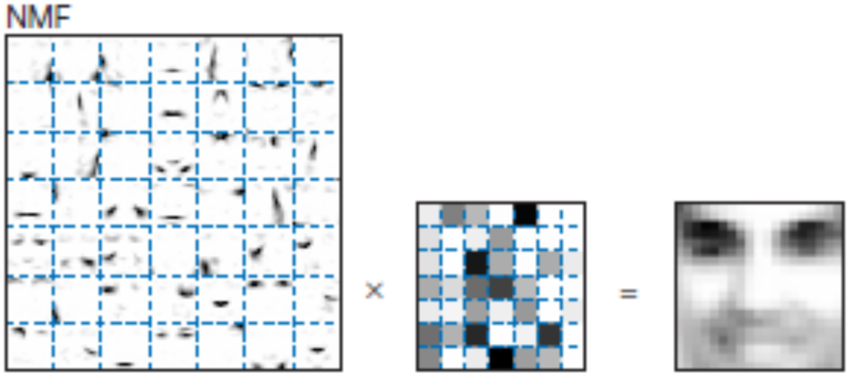
Factor **H**
Weights for 10 terms relative to 3 topics

	Topic 1	Topic 2	Topic 3
research	0.0	0.0	1.0
school	0.0	0.1	0.1
education	0.0	0.0	1.0
disease	0.0	0.6	0.0
patient	0.0	0.7	0.0
health	0.0	1.0	0.0
budget	0.3	0.1	0.2
finance	0.6	0.0	0.0
banking	0.7	0.0	0.0
bonds	0.3	0.0	0.0

10 Rows x 3 Columns

Где еще можно применить:

- В распознавании лиц (вместо, например, PCA)



Где еще можно применить:

- В распознавании лиц (вместо, например, PCA)
- Везде, где признаки можно интерпретировать как неотрицательные функции (интенсивность пикселей, набранные очки, возраст)

Вопросы на данный момент:

1. Запишите в общем виде представление матрицы как суммы матриц ранга 1.

2. Как сжать изображение с помощью SVD?

3. Единственно ли разложение NMF?

QR-разложение

$$A_{(n \times n)} = Q_{(n \times n)} R_{(n \times n)}$$

Верхнетреугольная матрица

Ортогональная матрица

$$A_{(m \times n)} = Q_{(m \times m)} R_{(m \times n)}$$

Алгоритм Грама-Шмидта

A = [a1, …., an]

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{u}_2 &= \mathbf{a}_2 - \mathrm{proj}_{\mathbf{u}_1}\,\mathbf{a}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{u}_3 &= \mathbf{a}_3 - \mathrm{proj}_{\mathbf{u}_1}\,\mathbf{a}_3 - \mathrm{proj}_{\mathbf{u}_2}\,\mathbf{a}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
&\ \vdots & &\ \vdots \\
\mathbf{u}_k &= \mathbf{a}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j}\,\mathbf{a}_k, & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}
\end{aligned}
$$

Алгоритм Грама-Шмидта

Q = [e1, …., en]

$$Q^{\mathsf{T}} Q = I.$$

$$Q^{\mathsf{T}} A = Q^{\mathsf{T}} Q R$$

$$R = Q^{\mathsf{T}} A$$

Алгоритм Хаусхолдера

По сравнению с алгоритмом Грама-шмидта:

- Обладает вычислительной устойчивостью
- Более ресурсозатратный

Метод наименьших квадратов

Один из методов решения СЛУ – метод наименьших квадратов:

$$Ax = b$$

$$A^T Ax = A^T b \quad \Rightarrow$$

$$x = (A^T A)^{-1} A^T b$$

Метод наименьших квадратов

$$A^T A x = A^T b$$

$$(QR)^T (QR) x = (QR)^T b$$

$$R^T Q^T Q R x = R^T Q^T b$$

$$R x = Q^T b$$

LU-разложение

$$A_{(n \times n)} = L_{(n \times n)}\, U_{(n \times n)}$$

невырожденная матрица

Верхнетреугольная матрица

Нижнетреугольная матрица

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Нахождение матриц L и U

Первый шаг:

Элементы на диагонали L равны 1

1. $u_{1j} = a_{1j},\ j = 1 \ldots n$
2. $l_{j1} = \dfrac{a_{j1}}{u_{11}},\ j = 1 \ldots n\ (u_{11} \neq 0)$

Далее для i = 2, …., n:

1. $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj},\ j = i \ldots n$
2. $l_{ji} = \dfrac{1}{u_{ii}}(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki}),\ j = i \ldots n$

Решение СЛУ

1. $Ax = b$
2. $A = LU$
3. $LUx = b$
4. $Ly = b$
5. $Ux = y$

Разложение Холецкого

Разложение всегда существует и единственно для любой симметричной матрицы

$$A = L\ L^T$$

Нижнетреугольная матрица

Симметричная матрица

или же:

$$A = U^T\ U$$

$(U = L^T)$

Верхнетреугольная матрица

Алгоритм построения L

Элементы матрицы L вычисляются начиная с верхнего левого угла матрицы:

$$L_{ii} = \sqrt{A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2},$$

$$L_{ij} = \frac{1}{L_{jj}} \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right), \text{ если } j < i$$

вычисление происходит сверху вниз, слева направо

Применения

Данное разложение может использоваться для решения СЛУ $Ax = b$, в случае если A – симметричная положительно определенная матрица.

Разложение Холецкого также применяется в методах Монте-Карло для генерации коррелированных случайных величин.

Вопросы:

1 Каковы различия алгоритма Грамма-Шмидта и Хаусхолдера для построения QR-разложения (преимущества и недостатки)?
2 Какое разложение стоит использовать в случае симметричных матриц?
3 Чем отличается QR от LU и что лучше использовать для СЛУ?

Заключение

- Матрицы удобно представлять в виде произведения матриц с полезными свойствами (ортогональность, диагональность, etc.)
- Это делается для решения самых разных задач (любых, где могут быть полезны свойства матриц-множителей)
- Мы рассмотрели применение МР к решению СЛУ, понижению размерности матриц, выделению тем (топиков) в текстах

Источники

PCA and SVD explained with numpy by Zichen Wang

A Gentle Introduction to Matrix Factorization for Machine Learning by Jason Brownlee

Applications of Matrix Decompositions for Machine Learning by Prince Grover

Non-negative matrix factorization from Wikipedia, the free encyclopedia

Источники

Статья про разложение Холецкого на википедии

Статья про разложение Холецкого на алговики

Статья про LU-разложение на википедии

Примеры QR-разложения

Статья на гитхабе про QR-разложение

Статья про QR-разложение на википедии