

# Матричные вычисления

По лекциям Максима Рахубы

## Содержание

<b>1</b>	<b>О курсе</b>	<b>3</b>
<b>2</b>	<b>Основы матричного анализа</b>	<b>3</b>
2.1	Векторные нормы . . . . .	3
2.1.1	Разреженность в L1-норме . . . . .	3
2.1.2	Скалярное произведение . . . . .	4
2.1.3	Унитарная инвариантность L2-нормы . . . . .	4
2.2	Матричные нормы . . . . .	4
2.3	Разложение Шура . . . . .	5
2.4	Нормальные матрицы . . . . .	6
<b>3</b>	<b>Малоранговое приближение матриц</b>	<b>6</b>
3.1	Разделение переменных и скелетное разложение . . . . .	6
3.2	SVD . . . . .	7
3.3	Ортопроектор . . . . .	8
3.4	Простейший рандомизированный алгоритм . . . . .	9
<b>4</b>	<b>Малоранговое приближение матриц — 2</b>	<b>9</b>
4.1	Скелетная аппроксимация матриц . . . . .	9
4.2	ALS алгоритм . . . . .	10
<b>5</b>	<b>Малоранговая аппроксимация многомерных массивов (тензоров)</b>	<b>11</b>
5.1	Кронекерово произведение . . . . .	11
5.2	Внешнее (тензорное) произведение . . . . .	13
5.3	Разложение Таккера . . . . .	14
<b>6</b>	<b>QR-разложение и метод наименьших квадратов</b>	<b>15</b>
6.1	QR-разложение . . . . .	15
6.1.1	Ортогонализация Грамма-Шмидта . . . . .	15
6.1.2	Отражения Хаусхолдера . . . . .	15
6.1.3	Вращения Гивенса . . . . .	17
6.2	Метод наименьших квадратов . . . . .	17
6.2.1	Полноранговый случай . . . . .	17
6.2.2	$\text{rank } A \leq n$ . . . . .	18

<b>7</b>	<b>Быстро умножаем векторы на матрицы</b>	<b>18</b>
7.1	Быстрое преобразование Фурье . . . . .	18
7.2	Циркулянты . . . . .	19
7.3	Двухуровневый циркулянт . . . . .	20
7.4	Матрицы Тёплица . . . . .	20
<b>8</b>	<b>Умножение матриц и вычислительная устойчивость</b>	<b>20</b>
8.1	Сложность матричного умножения . . . . .	20
8.2	Метод Штрассена . . . . .	21
8.2.1	Вывод метода Штрассена . . . . .	21
8.3	Иерархия памяти . . . . .	21
8.4	Машинные числа . . . . .	21
8.5	Машинные числа: обусловленность . . . . .	21
<b>9</b>	<b>Примеры решения линейных систем с плотными матрицами</b>	<b>22</b>
9.1	LU-разложение . . . . .	22
9.1.1	Связь LU-разложения и метода исключения Гаусса . . . . .	24
9.2	Выбор ведущего элемента . . . . .	24
9.3	Разложение Халецкого . . . . .	25
<b>10</b>	<b>Прямые методы решения линейных систем с большими разреженными матрицами</b>	<b>25</b>
10.1	Формула Шермана-Моррисона . . . . .	25
10.2	Форматы представления разреженных матриц . . . . .	26
10.2.1	COO формат . . . . .	26
10.2.2	lil (list of lists) . . . . .	26
10.2.3	CSR (compressed sparse row) . . . . .	26
10.3	Заполнение в L и U . . . . .	27
10.4	Алгоритм поиска P . . . . .	27
10.4.1	Алгоритм Катхилла-Макки . . . . .	27
10.4.2	Minimal degree ordering . . . . .	27
<b>11</b>	<b>Итерационные методы решения линейных уравнений (?)</b>	<b>27</b>
11.1	Метод простой итерации . . . . .	27
11.2	Метод наискорейшего спуска . . . . .	28
11.3	Итерационный метод Чебышева . . . . .	29
11.3.1	Оценка ошибок . . . . .	29
<b>12</b>	<b>???</b>	<b>30</b>
12.1	Оптимизация на подпространствах Крылова . . . . .	30
12.2	Метод сопряжённых градиентов . . . . .	31

# 1 О курсе

Большую часть сказанного можно найти в [ВИКИ](#).

Правда, кроме указанных на вики источников, было упомянуто ещё два:

1. Gilbert (неразборчиво) — Matrix Methods in Data Science (скорее всего, Gilbert Strang — Matrix Methods in Data Analysis, Signal Processing, and Machine Learning)
2. [Ivan Oseledets @ github](#). Скорее всего, имеются в виду репозитории с названиями pla20XX.

## 2 Основы матричного анализа

### 2.1 Векторные нормы

**Определение 2.1.** Векторная норма — функция  $f : \mathbb{F}^n \rightarrow \mathbb{R}$  такая, что:

- $f(x) \geq 0$ ;  $f(x) = 0 \Leftrightarrow x = 0$ ;
- $f(\alpha x) = |\alpha| f(x)$ ;
- $f(x + y) \leq f(x) + f(y)$ .

Обозначается  $\|x\|$ .

Примеры:

- $L_1$ -норма (Единичная окружность — ромб, TODO: нарисовать):

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

- $L_2$ -норма (Единичная окружность — окружность):

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2} = \sqrt{x^* x}$$

- $A$ -норма:

$$\|x\|_A = \sqrt{x^* A x}, \quad A = A^*, \quad \forall x \neq 0 : x^* A x > 0$$

- $L_\infty$ -норма (Единичная окружность — квадрат):

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

- $L_p$ -норма:

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad p \geq 1.$$

#### 2.1.1 Разреженность в $L_1$ -норме

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad m < n$$

Минимизируем  $x$  по  $L_2$ - и  $L_1$ -норме, в случае  $L_1$  получим **разреженное** решение (с большим кол-вом нулей) (TODO: нарисовать).

### 2.1.2 Скалярное произведение

**Определение 2.2.** Скалярное произведение  $(x, y) = x^*y$ .

**Теорема 2.1.** (Неравенство Коши-Буняковского-Шварца).

$$|(x, y)| \leq \|x\| \cdot \|y\|.$$

**Теорема 2.2.** (Неравенство Гёльдера).

$$|(x, y)| \leq \|x\|_p \cdot \|y\|_q \Leftrightarrow \begin{cases} p, q \geq 1; \\ \frac{1}{p} + \frac{1}{q} = 1. \end{cases}$$

### 2.1.3 Унитарная инвариантность L2-нормы

**Определение 2.3.** Унитарная матрица  $U$  —  $U \in \mathbb{C}^{n \times n}$ :

$$U^{-1} = U^* \quad (\Leftrightarrow I = U^*U = UU^*)$$

*Утверждение 2.1.* Если  $U$  — унитарная матрица, то  $\|Ux\|_2 = \|x\|_2$ .

*Доказательство.*

$$\|Ux\|_2 = \sqrt{(Ux)^*Ux} = \sqrt{x^*U^*Ux} = \sqrt{x^*x} = \|x\|_2.$$

□

## 2.2 Матричные нормы

**Определение 2.4.** Норма  $\|\cdot\|$  называется матричной, если

1.  $\|\cdot\|$  — векторная норма на пространстве матриц;
2.  $\|AB\| \leq \|A\| \cdot \|B\|$  (субмультипликативность).

Примеры:

- Норма Фробениуса:

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\text{trace}(A^*A)}.$$

- Операторные нормы. Если  $\|\cdot\|_*$ ,  $\|\cdot\|_{**}$  — векторные нормы, то соответствующей им операторной нормой будет

$$\|A\|_{*,**} = \sup_{x \neq 0} \frac{\|Ax\|_*}{\|x\|_{**}} = \sup_{\|y\|_{**}=1} \|Ay\|_*.$$

- Например, операторной нормой, соответствующей  $L_2$ -норме, является

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^*A)} = \sigma_1(A).$$

*Утверждение 2.2.* Для любой матрицы  $A$  и для любых унитарных матриц  $U, V$  верно

$$\begin{aligned} \|UAV\|_F &= \|A\|_F \\ \|UAV\|_2 &= \|A\|_2 \end{aligned}$$

*Доказательство.* Для  $\|\cdot\|_2$ :

$$\|UAV\|_2 = \sup_{x \neq 0} \frac{\|UAVx\|_2}{\|x\|_2} = \sup_{x \neq 0} \frac{\|U^*(UAVx)\|_2}{\|Vx\|_2}$$

Заменим  $Vx$  на  $y$ . В силу обратимости  $V$  это будет равно

$$\sup_{y \neq 0} \frac{\|Ay\|_2}{\|y\|_2} = \|A\|_2.$$

□

## 2.3 Разложение Шура

Собственное разложение (существует не всегда):

$$A = S\Lambda S^{-1}, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$$

Жорданова форма (всегда существует, но неустойчива при вычислениях):

$$A = PJP^{-1}$$

Для вычислений используют разложение Шура.

**Теорема 2.3.** (О разложении Шура)

Для всякой  $A \in \mathbb{C}^{n \times n}$  существуют такие унитарная  $U$  и верхнетреугольная  $T$ , что  $A = UTU^*$ .

*Доказательство.* Индукцией по размерности  $A$ .

**База.**  $n = 1$ :  $U = I$ ,  $T = A$ .

**Переход.**  $(n - 1) \rightarrow n$ .

Т.к.  $\mathbb{C}$  алгебраически замкнуто, у характеристического многочлена  $A$  есть хотя бы один корень, т.е. у  $A$  есть хотя бы одно собственное значение  $\lambda_1$ , т.е. всегда найдётся хотя бы один ненулевой собственный вектор  $v_1$  единичной длины.

Дополним  $v_1$  до ортонормированного базиса  $v_1, \dots, v_n$  и положим  $U_1 = (v_1 | \dots | v_n)$ .

$v_1$  — собственный вектор, так что  $Av_1 = \lambda_1 v_1$ . Тогда, в силу ортогональности  $v_i$  и  $v_j$ ,

$$v_i^* A v_1 = \begin{cases} \lambda, & i = 1; \\ 0, & i \neq 1. \end{cases}$$

Поумножаем пару матриц:

$$U_1^* A U_1 = \begin{pmatrix} v_1^* \\ \vdots \\ v_n^* \end{pmatrix} \cdot \begin{pmatrix} A v_1 & \dots & A v_n \end{pmatrix} = \begin{pmatrix} \lambda & v_1^* A v_2 & \dots \\ 0 & & \\ \vdots & A_1 & \\ 0 & & \end{pmatrix}$$

По индукции разложим  $A_1$  как  $V_1 T_1 V_1^*$ . Запишем  $U_1^* A U_1$  с помощью блочного умножения:

$$\begin{pmatrix} 1 & 0 \\ 0 & V_1 \end{pmatrix} \cdot \begin{pmatrix} \lambda & \dots \\ 0 & T_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & V_1^* \end{pmatrix}$$

В силу обратимости  $V_1^*$  мы можем так сделать (иначе вектора-строки для  $\dots$  над  $T_1$  могло бы и не существовать).

Получаем, что

$$T = \begin{pmatrix} \lambda & \cdots \\ 0 & T_1 \end{pmatrix};$$

$$U = U_1 \cdot \begin{pmatrix} 1 & 0 \\ 0 & V_1 \end{pmatrix}.$$

Действительно,  $T$  верхнетреугольная по построению, а  $U$  унитарна как произведение двух унитарных матриц.  $\square$

## 2.4 Нормальные матрицы

**Определение 2.5.** Матрица  $A$  называется нормальной, если  $A^*A = AA^*$ .

*Утверждение 2.3.* Матрица диагонализуема в унитарном базисе тогда и только тогда, когда она является нормальной.

*Доказательство.* .

- $(\Rightarrow)$ :

$$A^*A = U\Lambda^*U^*U\Lambda U^* = U\Lambda^*\Lambda U^* = U\Lambda\Lambda^*U^* = AA^*.$$

- $(\Leftarrow)$ : Разложение Шура для  $A$ :  $UTU^*$ .

$$A^*A = AA^* \Rightarrow T^*T = TT^*.$$

Оставшаяся часть доказательства  $(\Leftarrow)$  будет в качестве упражнения в ДЗ.

$\square$

## 3 Малоранговое приближение матриц

### 3.1 Разделение переменных и скелетное разложение

**Определение 3.1.** Функция с разделенными переменными — такая функция  $f : X \times Y \rightarrow Z$ , что существуют  $u, v$  такие, что  $f(x, y) = u(x)v(y)$ .

Для приближения функций используют сумму функций с разделенными переменными:

$$f(x, y) \approx \sum_{i=1}^r u_i(x)v_i(y)$$

Например, разложения в ряд Тейлора и в ряд Фурье:

$$f(x, y) \approx \sum_{i,j=0}^p c_{ij}x^i y^j$$

$$f(x, y) \approx \sum_{i,j=1}^r c_{ij} \sin \pi i x \sin \pi j y \quad ((x, y) \in (0, 1)^2)$$

Как это относится к матричным вычислениям? Возьмём матрицу  $A \in \mathbb{R}^{m \times n}$ .  $a_{ij}$  — функция дискретных переменных  $i, j$ .

Если  $a_{ij} = u_i v_j$ , то  $A = uv^T$  (обратное тоже верно).

Раз все строки (столбцы) матрицы коллинеарны,  $rk A \leq 1$ .

Примеры:  $a_{ij} = \sin i \cos j$ ,  $a_{ij} = i$ .

**Определение 3.2.** Скелетное разложение, rank decomposition — разложение  $A = UV^T$  такое, что новая размерность  $U, V$  минимальна.

Замечания:

1. Storage:  $mn$  Vs.  $(m + n)r$
2. Разложение единственно с точностью до умножения на обратимую матрицу:

$$UV^T = (US)(S^{-1}V^T)$$

Утверждения:

1.  $A = UV^T \Rightarrow rk(A) \leq r$ ;
2.  $rk(A) = r \Rightarrow \exists U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{r \times n} : A = UV^T$ .

*Доказательство.* .

1. очев
2. очев

□

## 3.2 SVD

**Теорема 3.1.** Пусть  $A \in \mathbb{C}^{m \times n}$ ,  $rk(A) = r$ , тогда найдутся унитарные  $U \in \mathbb{C}^{m \times m}$ ,  $V \in \mathbb{C}^{n \times n}$  и  $\sigma_1 \geq \dots \geq \sigma_r > 0$  — сингулярные числа, что

$$A = U\Sigma V^*,$$

где  $\Sigma$  — диагональная матрица с  $\sigma_1, \dots, \sigma_r$  на диагонали.

*Доказательство.* Заметим, что  $A^*A \geq 0$  и  $(A^*A)^* = A^*A$ . Из этого следует, что

$$\exists V : V^*A^*AV = \text{diag}(\sigma_1^2, \dots, \sigma_n^2), \quad \sigma_1 \geq \dots \geq \sigma_n \geq 0. \quad (1)$$

Рассмотрим  $V_r = [v_1, \dots, v_r]$  и  $\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$ , где  $\sigma_{r+1} = 0$  ( $r$  пока неизвестно).

$$\begin{aligned} V_r^*A^*AV_r &= \Sigma_r^2; \\ (\Sigma_r^{-1}V_r^*A^*) (AV_r\Sigma_r^{-1}) &= I \end{aligned}$$

Получается, что  $Av_i = \sigma_i u_i$  при  $i \in [1, r]$ .

А при  $i \in [r + 1, n]$  —  $Av_i = 0$ .

Достраиваем  $U_r$  до унитарной  $U$ :

$$AV = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow U^*AV = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}.$$

Из этого, в частности, следует, что  $r = rk(A)$ .

□

Замечание:

1.  $u_i$  — левые сингулярные векторы;  
 $v_i$  — правые сингулярные векторы;  
 $\sigma_{r+1} = \dots = \sigma_{\min(m,n)} = 0$  — нулевые сингулярные числа.
2. Сингулярные числа определены однозначно.
3. Сингулярные векторы определяются однозначно с точностью до множителя  $C : |C| = 1$  при  $\sigma_1 > \dots > \sigma_r > 0$ .
- 4.

$$\begin{aligned} \text{Im}(A) &= \langle u_1, \dots, u_r \rangle; \\ \text{Ker}(A) &= \langle v_{r+1}, \dots, v_n \rangle. \end{aligned}$$

5. SVD  $\rightarrow$  скелетное разложение:

$$A = U\Sigma V^* = \hat{U}V^*.$$

Переход в другую сторону будет на семинаре.

**Теорема 3.2.** (Эккорта-Янга-Мирского).

Пусть  $k < \text{rk}(A)$  и  $A_k = U_k \Sigma_k V_k^*$ , тогда

$$\min_{\text{rk}(B) \leq k} \|A - B\| = \|A - A_k\|$$

для любой унитарно инвариантной нормы  $\|\cdot\|$ , причем  $\|A - A_k\|_2 = \sigma_{k+1}$ ,  $\|A - A_k\|_F = \sqrt{\sum_{i=k+1}^r \sigma_i^2}$ .

**Определение 3.3.** Нормы Шаттена:

$$\|A\|_{p, \text{Shatten}} = \left( \sum_{i=1}^r \sigma_i^p \right)^{1/p}, \quad 1 \leq p \leq \infty$$

Нормы Шаттена унитарно инвариантны!

Примеры:

1.  $\|\cdot\|_{2, \text{Shatten}} = \|\cdot\|_F$ ;
2.  $\|\cdot\|_{\infty, \text{Shatten}} = \|\cdot\|_2$ ;
3.  $\|\cdot\|_{1, \text{Shatten}} = \|\cdot\|_*$  — ядерная (nuclear) норма.

### 3.3 Ортопроектор

**Определение 3.4.**  $P$  — ортопроектор на  $L$ , если

1.  $\text{Im}(P) = L$
2.  $P^2 = P$
3.  $P^* = P$



Утверждение 3.1.

$$\forall x \in \mathbb{C}^n : Px \perp (I - P)x$$

Доказательство.

$$(Px, (I - P)x) = (x, P^*(I - P)x) = (x, P(I - P)x) = (x, 0) = 0.$$

□

Утверждение 3.2. Если  $U \in \mathbb{C}^{n \times k}$ ,  $U^*U = I_k$ , то  $UU^*$  — ортопроектор на  $\langle u_1, \dots, u_k \rangle$ .

Ортопроекторы, связанные с SVD.

$$A = U\Sigma V^*, U = [U_r | \hat{U}_r], V = [V_r | \hat{V}_r]$$

$U_r U_r^*$  — ортопроектор на  $Im(A)$ .

$\hat{V}_r \hat{V}_r^*$  — ортопроектор на  $Ker(A)$ .

### 3.4 Простейший рандомизированный алгоритм

Хотим найти  $Q \in \mathbb{R}^{m \times r}$  с ортогональными столбцами такую, что для  $A \in \mathbb{R}^{m \times n}$

$$A \approx QQ^T A \quad (\text{если } r = rk(A), Q = U_r, \text{ то точное равенство}).$$

Если мы нашли  $Q$ , то

$$Q(Q^T A) = Q(W\Sigma V^T) = U\Sigma V^T \quad \text{— SVD.}$$

Как выбрать  $Q$ ?

1.  $\Omega = [\omega_1, \dots, \omega_r]$  — случайная матрица;
2.  $Y = A \cdot \Omega$ ;
3. Ортогонализация столбцов  $Y$ . Например, с помощью Грамма-Шмидта. (= QR-разложение)

## 4 Малоранговое приближение матриц — 2

### 4.1 Скелетная аппроксимация матриц

Посмотрим, какие алгоритмы мы уже рассмотрели.

- `pr.linalg.svd` —  $O(mn \min(n, m))$ , при  $m = n$  —  $O(n^3)$ . НО! Гарантированная точность для любой матрицы.
- Рандомизированные алгоритмы —  $O(mnr)$  из-за умножения на матрицу. Для разреженных матриц сложность ещё меньше. НО! Выигрыш для  $r \ll \min(m, n)$ ; не всегда точно.

Есть ли алгоритм со сложностью  $O(\# \text{ эл-в разложения}) = O((m + n)r)$ ?

Значит, мы не должны использовать все элементы раскладываемой матрицы  $A$ .

Скелетное разложение:

$$A = CV^T, \quad C \text{ — базисные столбцы } A;$$

Или:

$A = UR$ ,  $R$  — базисные строки  $A$ .

**Теорема 4.1.** Любая  $A \in \mathbb{C}^{m \times n}$  ранга  $r$  представляется в виде

$$A = C \hat{A}^{-1} R$$

( $\hat{A}$  — любая невырожденная подматрица  $r \times r$ )

*Доказательство.*  $A = [a_1 \dots a_n]$ . Тогда  $a_i = C \cdot x_i$ . В свою очередь,  $\hat{a}_i = \hat{A} x_i$ .

$$R = [\hat{a}_1 \dots \hat{a}_n] = \hat{A} [x_1 \dots x_n] \Rightarrow [x_1 \dots x_n] = \hat{A}^{-1} R;$$

$$A = [C x_1 \dots C x_n] = C [x_1 \dots x_n] = C \hat{A}^{-1} R.$$

□

**Теорема 4.2.** Пусть для  $A$  существует  $B$  ранга  $r$ :  $\|A - B\|_2 \leq \varepsilon$ . Пусть  $\hat{A} \in \mathbb{C}^{r \times r}$  — подматрица максимального по модулю определителя. Тогда

$$\left\| A - C \hat{A}^{-1} R \right\|_c \leq (r + 1) \varepsilon$$

Также называют CGR-разложением, CUR-разложением или псевдоскелетной аппроксимацией.

(Дальше по мотивам презы)

Пример — разложение матрицы Гильберта  $a_{ij} = 1/(i + j - 1)$ .

При  $r \approx 15$  ошибка внезапно подскакивает. В чём дело? Матрица  $\hat{A}$  оказывается близка к вырожденной, а мы её обращаем.

Для устойчивости необходимо регуляризовать вычисление  $\hat{A}^{-1}$ , например, с помощью SVD. Для этого в numpy есть функция `np.linalg.pinv`.

Метод неполной крестовой аппроксимации...

(Преза закончилась)

## 4.2 ALS алгоритм

(Alternating least squares / Alternating linear scheme)

Рассмотрим  $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ . Хотим  $f(X) \rightarrow \min_{\text{rank}(X) \leq r}$ .

Например, задача о наилучшем приближении ранга  $r$ :

$$f(X) = \|A - X\|_F^2;$$

Или задача matrix completion (пример использования — рекомендательная система):

$$f(X) = \|P_\Omega \circ (A - X)\|_F^2, \quad (P_\Omega)_{ij} = (i, j) \in \Omega$$

Можно также  $f(X) = \|X\|_*$ .

Вернёмся к алгоритму.

$$\min_{\text{rank } X \leq r} f(X) = \min_{U, V} f(UV^T)$$

Алгоритм 1: ALS vanilla

$$U_{k+1} = \arg \min_U f(UV_k^T);$$

$$V_{k+1} = \arg \min_V f(U_{k+1} V^T).$$

У этого алгоритма есть существенные недостатки: вектора одной матрицы начнут расти, а другой — уменьшаться; вектора внутри одной матрицы могут становиться почти линейно зависимыми.

Алгоритм 2: ALS с ортогонализацией

$$\begin{aligned} U &:= \arg \min_U f(U V_k^T); \\ U &= Q_1 R_1 \\ V &:= \arg \min_V f(Q_1 V^T); \\ V &= Q_2 R_2 \\ U_{k+1} &= Q_1 R_2^T; \quad V_{k+1} = Q_2 \end{aligned}$$

Как решать  $f(UV^T) \rightarrow \min$ , где  $U^T V = I$  и  $f(X) = \|A - UV^T\|_F^2$ :

1.  $(A, B)_F = \text{trace}(A^T B)$
2.  $\frac{\partial}{\partial X} \text{trace}(XA) = A^T$

*Доказательство.*

$$\left[ \frac{\partial f}{\partial x_{ij}} \right]_{i,j=1}^{m,n} = \frac{\partial f}{\partial X}$$

□

3.  $f(UV^T) = \|A - UV^T\|_F^2 = (A - UV^T, A - UV^T)_F = (A, A) - 2(A, UV^T) + (UV^T, UV^T) = (A, A) + \dots$
4. Получаем, что оптимум  $V_* = A^T U$ , и аналогично  $U_* = AV$ .

## 5 Малоранговая аппроксимация многомерных массивов (тензоров)

### 5.1 Кронекерово произведение

**Определение 5.1.**

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}$$

— называется Кронекеровым произведением.

(Если  $A \in \mathbb{F}^{m \times n}$ ,  $B \in \mathbb{F}^{p \times q}$ , то  $A \otimes B \in \mathbb{F}^{mp \times nq}$ .)

*Утверждение 5.1.* .

1.  $(A + B) \otimes C = A \otimes C + B \otimes C; \quad A \otimes (B + C) = A \otimes B + A \otimes C;$

2.

$$(A \otimes B)^* = A^* \otimes B^*;$$

3.

$$A \otimes B = (A \otimes I)(I \otimes B) = (I \otimes B)(A \otimes I)$$

4.

$$(AB) \otimes (CD) = (A \otimes C)(B \otimes D)$$

5.

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

*Доказательство.* .

1. очевидно

2. очевидно

3. доказывается пристальным взглядом:

$$(A \otimes I)(I \otimes B) = \begin{pmatrix} a_{11}I & \dots & a_{1n}I \\ \vdots & & \vdots \\ a_{m1}I & \dots & a_{mn}I \end{pmatrix} \begin{pmatrix} B & & \\ & B & \\ & & \ddots \\ & & & B \end{pmatrix}$$

4. Проверим, что

$$I \otimes CD = (I \otimes C)(I \otimes D); \quad AB \otimes I = (A \otimes I)(B \otimes I).$$

Тогда по 3)

$$AB \otimes CD = (AB \otimes I)(I \otimes CD) = (A \otimes I)(B \otimes I)(I \otimes C)(I \otimes D) = OOF.$$

Снова по 3)

$$OOF = (A \otimes I)(I \otimes C)(B \otimes I)(I \otimes D) = (A \otimes C)(B \otimes D).$$

5. следует из 4).

□

**Определение 5.2.**  $\text{vec}(\cdot) : \mathbb{C}^{m \times n} \rightarrow \mathbb{C}^{mn}$  действует следующим образом:

$$\text{vec} \left( \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \right) = \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \\ a_{12} \\ \vdots \\ a_{m2} \\ \vdots \\ a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}$$

В Python — `np.reshape(A, [m + n, 1], order='f')` либо `np.flatten(A, order='f')`.  
Пример:  $\text{vec}(uv^T) = v \otimes u$ .

*Утверждение 5.2.*

$$\text{vec}(AXB) = B^T \otimes A \cdot \text{vec}(X)$$

*Доказательство.* Пусть  $X = uv^T$ .

$$\text{vec}(Auv^T B) = \text{vec}((Au)(B^T v)^T) = (B^T v) \otimes (Au) = (B^T \otimes A)(v \otimes u).$$

Но  $\text{vec}$  и матричное умножение линейны, а любую матрицу можно представить как сумму матриц ранга 1.  $\square$

*Утверждение 5.3.*

$$\langle A, B \rangle_F = \text{trace}(A^T B) = \text{vec}(A)^T \text{vec}(B).$$

## 5.2 Внешнее (тензорное) произведение

**Определение 5.3.** Для  $A \in \mathbb{R}^{n_1 \times \dots \times n_d}$ ,  $B \in \mathbb{R}^{m_1 \times \dots \times m_D}$  тензор  $(A \otimes_o B) \in \mathbb{R}^{n_1 \times \dots \times n_d \times m_1 \times \dots \times m_D}$  называется внешним (тензорным) произведением:

$$(A \otimes_o B)_{ij} = A_i B_j \quad (i \in \mathbb{N}^d, j \in \mathbb{N}^D)$$

Пример.  $a_{ij} = u_i v_j$ ;  $A = uv^T = u \otimes_o v$ ;  $\text{vec}(u \otimes_o v) = v \otimes_K u$ .

Замечание. Мы будем писать  $\otimes$  вместо  $\otimes_o$ , если из контекста понятно, о чем идёт речь. Иначе будем писать  $\otimes_o$  и  $\otimes_K$ .

Скелетное разложение:

$$A = UV^T = u_1 \otimes v_1 + \dots + u_r \otimes v_r$$

SVD:

$$A = \sum_{\alpha=1}^r \sigma_{\alpha} u_{\alpha} \otimes v_{\alpha};$$

$$\text{vec } A = V \otimes_K U \text{vec } \Sigma.$$

Обобщим скелетное разложение на большее количество размерностей:

$$a_{ijk} = \sum_{\alpha=1}^R u_{i\alpha} v_{j\alpha} w_{k\alpha};$$

$$A = \sum_{\alpha=1}^R u_{\alpha} \otimes v_{\alpha} \otimes w_{\alpha};$$

$$A = \sum_{\alpha=1}^R \bigotimes_{k=1}^d u_{\alpha}^{(k)} \quad \text{для } A \in \mathbb{R}^{n_1 \times \dots \times n_d}.$$

- Единственность (при некоторых условиях) в отличие от 2D;
- Проблемы при вычислениях.

### 5.3 Разложение Таккера

$$A = \sum_{\alpha=1}^{R_1} \sum_{\beta=1}^{R_2} \sum_{\gamma=1}^{R_3} g_{\alpha\beta\gamma} u_{\alpha} \otimes v_{\beta} \otimes v_{\gamma}.$$

$G = \{g_{\alpha\beta\gamma}\}_{\alpha,\beta,\gamma=1}^{R_1,R_2,R_3}$  — ядро Таккера;

$$\left. \begin{aligned} U &= [u_1 \dots u_{R_1}] \in \mathbb{C}^{m \times R_1} \\ V &= [v_1 \dots v_{R_2}] \in \mathbb{C}^{n \times R_2} \\ W &= [w_1 \dots w_{R_3}] \in \mathbb{C}^{l \times R_3} \end{aligned} \right\} \text{факторы}$$

Минимальные  $(R_1, R_2, R_3)$  — Таккеровский (мультилинейный) ранг.

Также пишут  $A = [G; U, V, W]$ .

Storage:  $n^3 \gg R^3 + 3nR$  при  $R \ll n$ .

**Утверждение 5.4.**

$$\text{vec } A = W \otimes_K V \otimes_K U \cdot \text{vec } G.$$

*Доказательство.* Аналогично похожему док-ву через rank-1 члены. □

**Определение 5.4.**

$$A_{(1)} = [A[:, :, 0], A[:, :, 1], \dots, A[:, :, -1]]$$

$$A_{(2)} = \text{permute}(A, [1, 0, 2])_{(1)}$$

(В Питоне `permute` — `np.transpose`)

$$A_{(3)} = \text{permute}(A, [2, 0, 1])_{(1)}$$

—  $A_{(p)}$  — матризация по моде  $p$ .

Заметим, что  $\text{vec}(A) = \text{vec}(A_{(1)})$ . Тогда

$$\text{vec}(A) = \text{vec}(A_{(1)}) = W \otimes V \otimes U \cdot \text{vec}(G) = (W \otimes V) \otimes U \cdot \text{vec}(G_{(1)}) = \text{vec}(UG_{(1)}(W \otimes V))$$

Получаем, что

$$\begin{aligned} A_{(1)} &= \dots \\ A_{(2)} &= \dots \\ A_{(3)} &= WG_{(3)}(V \otimes U)^T \end{aligned}$$

**Теорема 5.1.**  $(R_1, R_2, R_3) = (\text{rank } A_{(1)}, \text{rank } A_{(2)}, \text{rank } A_{(3)})$ .

*Доказательство.* .

1.

$$A_{(1)}$$

2. Пусть  $A_{(1)} = U_1 \Sigma_1 V_1^T$  — compact SVD.  $U_1 U_1^T A_{(1)} = A_{(1)}$ ;

$$\text{vec}(A_{(1)}) = \text{vec}(U_1 U_1^T U G_{(1)} (W^T \otimes V^T))$$

Чё так быстро-то (((

□

HOSVD алгоритм.

$U_k$  —  $r_k$  левых сингулярных векторов  $A_{(k)}$ , где  $r_k : \|A_{(k)} - U_k U_k^T A_{(k)}\|_F \leq \varepsilon$ .

$G = [A, U_1^T, U_2^T, U_3^T]$ .

Тогда  $A_{HOSVD} = [G, U_1, U_2, U_3]$  и  $\|A - A_{HOSVD}\|_F \leq \sqrt{3}\varepsilon$ .

## 6 QR-разложение и метод наименьших квадратов

### 6.1 QR-разложение

#### 6.1.1 Ортогонализация Грамма-Шмидта

$a_1, \dots, a_n \in \mathbb{C}^m$  — линейно независимы.  $A := [a_1, \dots, a_n]$ .

Рассмотрим процесс ортогонализации Грамма-Шмидта.

$$\begin{aligned} \tilde{q}_1 &= a_1, & q_1 &= \frac{\tilde{q}_1}{\|\tilde{q}_1\|_2}; \\ \tilde{q}_2 &= a_2 - q_1 q_1^* a_2, & q_2 &= \frac{\tilde{q}_2}{\|\tilde{q}_2\|_2}; \\ \tilde{q}_3 &= a_3 - [q_1 q_2] [q_1 q_2]^* a_3, \dots \\ &\vdots \\ [a_1 a_2 \dots a_n] &= [q_1 q_2 \dots q_n] \begin{bmatrix} \|a_1\|_2 & q_1^* a_2 & \dots & \dots \\ 0 & \|\tilde{q}_2\|_2 & \dots & \dots \\ \vdots & 0 & \dots & \dots \\ \vdots & \vdots & \dots & \dots \\ 0 & 0 & \dots & \dots \end{bmatrix} \end{aligned}$$

Т.е.  $A = QR$ , где  $Q$  — унитарная, а  $R$  — верхнетреугольная.

ГШ неустойчив (будет в дз).

#### 6.1.2 Отражения Хаусхолдера

Хотим для любого  $x$  найти унитарную  $U$  такую, что

$$Ux = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Рассмотрим матрицу  $H(v) = I - 2vv^*$ , где  $\|v\|_2 = 1$ . Эта матрица называется матрицей Хаусхолдера. Она унитарна (было в ДЗ 1) и эрмитова (очев).

TODO: график

**Утверждение 6.1.**  $\forall a, b \in \mathbb{C}^n : \|a\|_2 = \|b\|_2 \quad \exists \gamma \in \mathbb{C} : |\gamma| = 1 \quad \wedge \quad \exists v \in \mathbb{C}^n : H(v) \cdot a = \gamma b$ .

Доказательство.

$$H(v) \cdot a = \gamma b \Rightarrow a - 2v^*av = \gamma b.$$

Если  $a$  коллинеарен  $b$ , то  $v = \frac{a}{\|a\|_2}$ .

Иначе положим

$$v = \frac{a - \gamma b}{\|a - \gamma b\|_2} \quad \text{и} \quad 2v^*av = a - \gamma b.$$

Подставим:

$$\begin{aligned} 2 \frac{(a^* - \bar{\gamma}b^*)a}{\|a - \gamma b\|_2} \cdot \frac{a - \gamma b}{\|a - \gamma b\|_2} &= a - \gamma b; \\ 2(a^*a - \bar{\gamma}b^*a) &= \|a - \gamma b\|_2^2 \quad \left( = \|a\|_2^2 + \|b\|_2^2 - 2\Re(\bar{\gamma}b^*a) \right) \\ -2\bar{\gamma}b^*a &= -2\Re(\bar{\gamma}b^*a); \\ \bar{\gamma}b^*a &\in \mathbb{R}. \end{aligned}$$

Если  $b \perp a$ , то подойдёт любое  $\gamma : |\gamma| = 1$ .

Иначе подойдёт  $\gamma = \pm \frac{b^*a}{|b^*a|}$ . □

Следствие.  $\forall a \in \mathbb{C}^n \exists v \in \mathbb{C}^n :$

$$\begin{aligned} H(v)a &= \gamma \begin{bmatrix} \|a\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, |\gamma| = 1; \\ v &= \frac{a - \gamma \|a\|_2 e_1}{\|a - \gamma \|a\|_2 e_1\|_2}. \end{aligned}$$

Замечание. Т.к.  $a - \gamma \|a\|_2 e_1 = (a_1 - \gamma \|a\|_2, a_2, \dots, a_n)^T$ , лучше выбрать  $\gamma < 0$ , т.к. возможно  $a_1 \approx \|a\|_2$  и будет вычитание двух близких чисел.

**Теорема 6.1.** Для любой  $A \in \mathbb{C}^{m \times n}$  существуют  $Q \in \mathbb{C}^{m \times m}$ ,  $R \in \mathbb{C}^{m \times n}$  ( $Q$  — унитарная,  $R$  — верхнетреугольная), что  $A = QR$ .

Доказательство.

$$A = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \xrightarrow{H_1} \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \xrightarrow{H_2 = \begin{bmatrix} 1 & \cdot \\ \cdot & \tilde{H}_2 \end{bmatrix}} \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix} \xrightarrow{H_3 = \dots} \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Получаем  $H_3 H_2 H_1 A = R$ . Но  $H_i$  — унитарная и эрмитова, так что  $A = (H_1 H_2 H_3) R$ .  $H_1 H_2 H_3$  — унитарная, что и требовалось. □

Замечания:

1. Доказательство даёт алгоритм построения  $QR$ .
2. Если  $m \geq n$ :

$$A = [Q_1 Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1 \text{ — thin QR (будет по умолчанию)}$$



Для вычисления thin QR не надо явно считать  $H_1 H_2 \dots H_n$ . Можно

$$Q_1 = \left( H_1 \left( H_2 \dots \left( H_n \begin{bmatrix} I_n \\ 0 \end{bmatrix} \right) \right) \right)$$

Сложность алгоритма —  $2mn^2 - \frac{2}{3}n^3$  (у Грамма-Шмидта —  $2mn^2$ ).

### 6.1.3 Вращения Гивенса

$G_{ij} \in \mathbb{R}^{m \times m}$  — матрица вращения. Единичная кроме подматрицы  $(i, j)^2$ , где имеет вид

$$J(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}.$$

Заметим, что  $J(\varphi)[a_1, a_2]^T = [\alpha, 0]^T$  при правильном подборе  $\varphi(a_1, a_2)$ .

Тогда алгоритм на основе вращений работает так: в каждом столбце, снизу вверх вращаем значения.

Сложность  $3mn - n^3$ . Больше, чем у Хаусхолдера, но параллелизуется.

## 6.2 Метод наименьших квадратов

### 6.2.1 Полноранговый случай

Пусть  $A \in \mathbb{R}^{m \times n}$ ,  $m \geq n$ ,  $\text{rank}(A) = n$ .

$$\|Ax - b\|_2^2 \rightarrow \min_{x \in \mathbb{R}^n}.$$

Решение:  $\nabla J(x) = 2A^T Ax - 2A^T b = 0$ ;  $A^T Ax = A^T b$ . Получаем  $x = (A^T A)^{-1} A^T b$ . Как считать? Не хочется напрямую считать  $A^T A$ .

Способ вычисления:

#### 1. Через QR.

$$A = QR, A^T A = R^T Q^T QR = R^T R.$$

Получаем  $R^T Rx = R^T Q^T b$ , сокращаем на  $R^T$ .

$$Rx = Q^T b \equiv f.$$

$R$  — верхнетреугольная! Можно решать Гауссом снизу.

Сложность  $O(n^2)$ .

Итог:  $2mn^2 - \frac{2}{3}n^3 + O(n^2)$  flops.

#### 2. Через compact SVD. Сложность будет $2mn^2 + 11n^3$ flops.

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

$$A^T A = V \Sigma^2 V^T$$

$$V \Sigma^2 V^T x = V \Sigma U^T b$$

$$x = V \Sigma^{-1} U^T b$$

Сложность через SVD > сложность через QR.

Но SVD может быть полезна, если сингулярные числа маленькие.

### 6.2.2 $\text{rank } A \leq n$

**Теорема 6.2.** Пусть  $A = U\Sigma V^T$  — полное SVD от  $A$ . Тогда

$$x_* = V\Sigma^+ U^T b, \quad \text{где } \Sigma^+ = \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Решает поставленную задачу минимизации и  $x_*$  имеет минимальную вторую норму среди всех решений.

*Доказательство.*

$$\|Ax - b\|_2^2 = \|U\Sigma V^T x - b\|_2^2 = \|U^T U \Sigma V^T x - U^T b\|_2^2 = \|\Sigma \alpha - U^T b\|_2^2 = \left\| \begin{bmatrix} \Sigma_r \alpha_r \\ 0 \end{bmatrix} - \begin{bmatrix} U_r^T b \\ U_\perp^T b \end{bmatrix} \right\|_2^2 = \|\Sigma_r \alpha_r - U_r^T b\|_2^2$$

$\alpha_r = \Sigma_r^{-1} U_r^T b$ ,  $\alpha_\perp$  — любое.

$\|\alpha\|_2^2 = \|x\|_2^2 \Rightarrow \min \text{ норма } x$ , если  $\alpha_\perp = 0$ ;

$\Rightarrow x^* = V_r \Sigma_r^{-1} U_r^T b = V \Sigma^+ U^T b$ . □

**Определение 6.1.**  $A^+ = V\Sigma^+ U^T$  — псевдообратная матрица.

## 7 Быстро умножаем векторы на матрицы

### 7.1 Быстрое преобразование Фурье

((TODO: пропустил начало)))

1.

$$\omega_{2n}^{2pq} = e^{-\frac{2\pi i}{2n} 2pq} = \omega_n^{pq}$$

2.

$$w_{2n}^{2p(n+q)} = w_n^{pq} w_n^{pq}$$

3. ...

$$P_{2n} F_{2n} = \begin{pmatrix} \dots \end{pmatrix}$$

$$F_{2n} X = P_n^{-1} \begin{pmatrix} F_n & 0 \\ 0 & F_n \end{pmatrix} \begin{pmatrix} I_n & 0 \\ 0 & W_n \end{pmatrix} \begin{pmatrix} I_n & I_n \\ I_n & -I_n \end{pmatrix} X$$

— если умножать справа налево по ассоциативности, получится быстро:  $O(n)$  на шаг рекурсии, рекурсия от вдвое меньшего  $n$ . Сложность  $O(n \log n)$ .

## 7.2 Циркулянты

**Определение 7.1.** Матрица называется циркулянт, если её элементы записываются в следующем виде:

$$C = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ c_2 & . & . & . & . \\ \vdots & . & . & . & . \\ c_{n-1} & c_{n-2} & . & \dots & c_0 \end{pmatrix}$$

$y = Cx$  — циклическая свёртка, т.е.

$$y_p = \sum_{q=0}^{n-1} c_{(p-q) \% n} \cdot x_q.$$

Цель: посчитать  $Cx$  за  $o(n^2)$ .

Пусть  $P$  — матрица циклического сдвига на 1 (влево? вправо?). Тогда

$$C = c_0 P^0 + c_1 P + c_2 P^2 + \dots + c_{n-1} P^{n-1}.$$

**Теорема 7.1.** Пусть  $C \in \mathbb{C}^{n \times n}$  — циркулянт, тогда

$$C = F_n^{-1} \text{diag}(F_n C) F_n.$$

*Доказательство.* Покажем, что  $P_n$  диагоналізується преобразованием Фурье. Тогда теорема будет доказана.

$$P_n \begin{pmatrix} 1 \\ \bar{\omega}_n^q \\ \bar{\omega}_n^{2q} \\ \vdots \\ \bar{\omega}_n^{(n-1)q} \end{pmatrix} = \begin{pmatrix} \bar{\omega}_n^{(n-1)q} \\ 1 \\ \bar{\omega}_n^q \\ \vdots \end{pmatrix} = \bar{\omega}_n^{-q} \begin{pmatrix} \bar{\omega}_n^{nq} \\ \bar{\omega}_n^q \\ \vdots \end{pmatrix}$$

$$P_n = F_n^* \Lambda (F_n^*)^{-1}, \quad \Lambda = \text{diag}(1, \omega_n, \dots, \omega_n^{n-1}).$$

$$\Lambda = \text{diag}(F_n e_1); \Lambda^k = \text{diag}(F_n e_k);$$

$$C = c_0 I + \dots = F_n \text{diag}(F_n C) F_n.$$

□

Из этого следует

**Теорема 7.2.** дискретная теорема свёртки:

$$Cx = F_n^{-1}((F_n C) \circ (F_n x)).$$

### 7.3 Двухуровневый циркулянт

**Определение 7.2.**

$$C = \begin{pmatrix} C_0 & C_{n-1} & \dots & C_1 \\ C_1 & C_0 & . & \vdots \\ \vdots & C_1 & . & \vdots \\ C_{n-1} & . & . & . \end{pmatrix}, C_k \in \mathbb{C}^{m \times m}$$

( $C_k$  — циркулянты)

**Теорема 7.3.** Пусть  $F_n \otimes F_m = F$ , тогда

$$Cx = F^{-1} \text{diag}(FC)F.$$

*Доказательство.*

$$C = I_n \otimes C_0 + P \otimes C_1 + P^2 \otimes C_2 + \dots$$

Далее пользуемся дискретной теоремой свёртки и свойствами Кронекерова произведения.  $\square$

### 7.4 Матрицы Тёплица

**Определение 7.3.** Тёплицева матрица —

$$T = t_{i-j},_{i,j=1}^n, \quad t_k \in \mathbb{C}.$$

$y = Tx$  или  $y_p = \sum_{q=1}^n t_{p-q} x_q$  — дискретная свёртка.

Любую Тёплицеву матрицу  $n \times n$  можно вложить в циркулянт  $(2n-1) \times (2n-1)$ .

Например, для  $n = 3$ :

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} \\ t_1 & t_0 & t_{-1} \\ t_2 & t_1 & t_0 \end{pmatrix};$$

$$C = \left( \begin{array}{ccc|cc} t_0 & t_{-1} & t_{-2} & t_2 & t_1 \\ t_1 & t_0 & t_{-1} & t_{-2} & t_2 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ - & - & - & - & - \\ t_{-2} & t_2 & t_1 & t_0 & t_{-1} \\ t_{-1} & t_{-2} & t_2 & t_1 & t_0 \end{array} \right).$$

Если дополнить  $x$  нулями до  $x^*$ , получим  $Tx = Cx^*$ .

Итого,

Fast matvec: `ifft(fft(c) * fft([x, 0].T))`

## 8 Умножение матриц и вычислительная устойчивость

### 8.1 Сложность матричного умножения

$O(n^3), O(n^{2.3\dots}), \dots$

## 8.2 Метод Штрассена

Позволяет уменьшить кол-во умножений за счёт сложений.

По мастер-теореме сложность  $O(n^{\log_2 7})$ . Более точно, константа где-то 7, так что выгоднее обычного умножения только при  $n \geq 500$ .

### 8.2.1 Вывод метода Штрассена

Вернёмся к блочному умножению.

$$\begin{aligned} C_1 &= A_1 B_1 + A_2 B_3; \\ C_2 &= A_1 B_2 + A_2 B_4; \\ C_3 &= A_3 B_1 + A_4 B_3; \\ C_4 &= A_3 B_2 + A_4 B_4. \end{aligned} \Rightarrow C_k = \sum_{i,j=1}^4 x_{ijk} A_i B_j.$$

Оказывается, что канонический ранг тензора  $x$  равен 7. Получив каноническое разложение, получаем метод Штрассена.

Точно так же (но увеличивая разбиение) получают асимптотически более эффективные алгоритмы, но константы там гигантские.

## 8.3 Иерархия памяти

1. Регистры;
2. Кеш;
3. RAM;
4. Жесткий диск.

(Отсортировано по убыванию быстродействия, но по возрастанию места.)

Оказывается, что большие матрицы эффективнее всего хранить не по столбцам и не по строкам, а по блокам, чтобы при выполнении алгоритмов память эффективно перемещалась по иерархии.

Обратно, стоит формулировать алгоритмы через операции с блоками, чтобы можно было воспользоваться особенностями иерархии.

((Обзор LAPACK и вариаций BLAS с оптимизациями под различные архитектуры)))

## 8.4 Машинные числа

(см. презу)

## 8.5 Машинные числа: обусловленность

Пусть  $f : X \rightarrow Y$ .

$$\begin{aligned} f(x + \Delta x) &\approx f(x) + f'(x)\Delta x; \\ \frac{f(x + \Delta x) - f(x)}{\|f(x)\|} &= \frac{f'(x)}{\|f(x)\|} \cdot \|x\| \cdot \frac{\Delta x}{\|x\|}. \end{aligned}$$

Определим число обусловленности (меру обусловленности):

$$\text{Cond}(f, x) = \frac{\|f'(x)\|}{\|f(x)\|} \cdot \|x\|$$

Пример 8.1.  $f(x) = Ax$ ,  $f'(x) = A$ .

$$\text{Cond} = \frac{\|A\|}{\|Ax\|} \|x\| = \frac{\|A\|}{\|y\|} \|A^{-1}y\| \leq \frac{\|A\|}{\dots}$$

Посчитаем обусловленность  $f : (A, b) \rightarrow A^{-1}b$ , то есть  $Ax = b$ .

**Лемма 8.1.** Пусть  $\|\cdot\|$  — матричная норма. Пусть  $\|A\| < 1$ . Тогда  $(I - A)^{-1}$  существует и

1. Ряд Неймана:

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

2.

$$\|(I - A)^{-1}\| \leq \frac{\|I\|}{1 - \|A\|}$$

Доказательство. ...

□

Теперь посчитаем обусловленность.

$$(A + \triangle A)(x + \triangle x) = b + \triangle b;$$

...

## 9 Примеры решения линейных систем с плотными матрицами

$$Ax = b, \quad A \in \mathbb{C}^{n \times n}, \det(A) \neq 0, \quad b \in \mathbb{C}^n$$

$$A = QR, \quad QRx = b; Rx = Q^*b, \quad \frac{4}{3}n^3 + O(n^2)$$

$$A = U\Sigma V^*, \quad x = V\Sigma^{-1}U^*b$$

### 9.1 LU-разложение

**Определение 9.1.**  $A = LU$  — LU-разложение матрицы  $A$ , где  $L$  — нижнетреугольная с 1 на диагонали, а  $U$  — верхнетреугольная.

$$LUx = b \Leftrightarrow \begin{cases} Ly = b; \\ Ux = y. \end{cases}$$

Сложность —  $O(n^2)$  на решение системы,  $O(n^3)$  на разложение (но константа меньше, чем в QR).

**Теорема 9.1.** Пусть  $\det(A) \neq 0$ . Тогда  $A$  имеет LU-разложение, что эквивалентно тому, что все ведущие подматрицы невырождены.

Доказательство. .

- $\Rightarrow A = LU$

$$0 \neq \det(A) = \det(LU) = \det(L)\det(U) = u_{11} \cdot \dots \cdot u_{nn}$$

Из чего следует, что  $\forall k : u_{kk} \neq 0$ .

$$A = \begin{pmatrix} L_k & 0 \\ * & * \end{pmatrix} \begin{pmatrix} U_k & * \\ 0 & * \end{pmatrix} = \begin{pmatrix} L_k U_k & * \\ * & * \end{pmatrix}$$

$$\det(L_k U_k) = u_{11} \cdot \dots \cdot u_{kk} \neq 0.$$

- $\Leftarrow$  — по индукции.

$$A = \begin{pmatrix} a & c^\top \\ b & D \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ -\frac{b}{a} & I \end{pmatrix} \begin{pmatrix} a & c^\top \\ b & D \end{pmatrix} = \begin{pmatrix} a & c^\top \\ 0 & D - \frac{b}{a}c^\top \end{pmatrix}$$

У  $A_1$  все ведущие нормы невырождены (ДЗ).

По индукции  $A_1 = L_1 U_1$ .

$$\begin{pmatrix} 1 & 0 \\ \frac{1}{a}b & L_1 \end{pmatrix} \begin{pmatrix} a & c^\top \\ 0 & U_1 \end{pmatrix} = \begin{pmatrix} a & c^\top \\ \dots & \end{pmatrix} = \begin{pmatrix} a & c^\top \\ b & D \end{pmatrix}$$

□

**Утверждение 9.1.** LU-разложение определяется единственным образом.

*Доказательство.*

$$A = L_1 U_1 = L_2 U_2 \Leftrightarrow L_2^{-1} L_1 = U_2 U_1^{-1} = I \Leftrightarrow L_1 = L_2, U_1 = U_2.$$

□

**Утверждение 9.2. (LDL-разложение)** Пусть у  $A$  все ведущие подматрицы невырождены,  $A = A^*$ ,  $\exists L$  — нижняя унитреугольная и  $D$  — диагональная:

$$A = LDL^*.$$

*Доказательство.*

$$A = LU = LDD^{-1}U = A^* = U^* D^{-*} D^* L^*$$

Из единственности следует  $L = U^* D^{-*}$ ,  $U^* = LD^*$ ,  $U = DL^*$ .

□

### 9.1.1 Связь LU-разложения и метода исключения Гаусса

LU:

$$\begin{pmatrix} 1 & 0 \\ -\frac{1}{a}b & I \end{pmatrix} \begin{pmatrix} a & c^\top \\ b & D \end{pmatrix} = \begin{pmatrix} a & c^\top \\ 0 & D - \frac{1}{a}bc^\top \end{pmatrix}$$

Гаусс:

$$\begin{pmatrix} 1 & 0 & 0 \dots 0 \\ 0 & 1 & 0 \dots 0 \\ 0 & v & I \end{pmatrix} \begin{pmatrix} * & * & \dots & * \\ 0 & * & . & \vdots \\ 0 & * & . & \vdots \\ \vdots & \vdots & . & \vdots \\ 0 & * & \dots & * \end{pmatrix} = \begin{pmatrix} * & * & \dots & * \\ 0 & * & . & \vdots \\ 0 & 0 & . & \vdots \\ \vdots & \vdots & . & \vdots \\ 0 & 0 & * \dots & * \end{pmatrix}$$

Сложность LU-разложения —  $\frac{2}{3}n^3 + O(n^2)$ .

На практике лучше использовать блочное разложение:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B \\ 0 & S \end{pmatrix}, \quad S = D - BA^{-1}C$$

( $S$  — дополнение по Шуру блока  $D$ )

## 9.2 Выбор ведущего элемента

**Теорема 9.2.** Пусть  $\exists$  LU-разложение  $A$  и не возникает ????. Тогда

$$|A - \tilde{L}\tilde{U}| \leq 3n\varepsilon_{\text{machine}}(|A| + |L||U|) + O(\varepsilon_{\text{machine}}^2)$$

Пример 9.1.

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{pmatrix} \begin{pmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{pmatrix}$$

- Выбор ведущего элемента (partial pivoting)

$$PA = LU$$

(TODO: матрица  $PA$ )

$P$  (перестановка) выбирается так, чтобы  $a_k$  был максимальным по модулю в 1-м столбце  $A_k$ . Тогда  $\|L\|_c \leq 1$ . Но элементы в  $U$  ещё могут расти:

$$A = \begin{pmatrix} 1 & 0 & 0 \dots 0 & 1 \\ -1 & \ddots & 0 \dots 0 & 1 \\ \vdots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & 1 \end{pmatrix}, \quad \frac{\|U\|_c}{\|A\|_c} = 2^{n-1}$$

- Полный выбор (full pivoting)

$$PAQ = LU$$

Выбираем перестановку такую, чтобы  $a_k$  был максимальным по модулю во всей  $A_k$ .



### 9.3 Разложение Халецкого

**Определение 9.2.**  $A = LL^*$  — разложение Халецкого, где  $L$  — нижнетреугольная матрица.

**Теорема 9.3.**  $A$  имеет разложение Халецкого  $\Leftrightarrow A = A^* > 0$ .

*Доказательство.* .

- $(\Rightarrow) A = LL^* = A^*, (x, LL^*x) = (L^*x, L^*x) > 0$ .
- $(\Leftarrow) A = LDL^* = LD^{1/2}D^{1/2}L^* = (LD^{1/2})(LD^{1/2})^*$ .

□

Алгоритм вычисления:

$$\begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{23} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix}$$

Решаем матричное уравнение.

Пример разобрали, в целом алгоритм следующий:

```
for k = 1, n:
    e[k,k] = sqrt(a[k,k] - e[k,1]^2 - ... - e[k,k-1]^2)
    for i = k + 1, n:
        e[i,k] = (a[i,k] - e[i,1]e[k,1] - ... - e[i,k-1]e[k,k-1]) / e[k,k]
```

**Теорема 9.4.**

$$|A - \tilde{L}\tilde{L}^*| \leq \varepsilon_{\text{machine}}(n+1) \begin{pmatrix} \sqrt{a_{11}} \\ \vdots \\ \sqrt{a_{nn}} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \dots & \sqrt{a_{nn}} \end{pmatrix} + O(\varepsilon_{\text{machine}}^2)$$

## 10 Прямые методы решения линейных систем с большими разреженными матрицами

### 10.1 Формула Шермана-Моррисона

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^n.$$

Пусть  $b \rightarrow \tilde{b}$ . Тогда решение пересчитывается за  $O(n^2)$ , ведь LU-разложение уже есть.

Что делать, если изменилась  $A$ ? На случай, если она изменилась каким-то управляемым способом, и существует формула Шермана-Моррисона.

**Утверждение 10.1.** Пусть  $\det(A) \neq 0$  и  $u, v \in \mathbb{R}^n$ . Тогда

$$1. A + uv^T \text{ обратима} \Leftrightarrow 1 + v^T A^{-1}u \neq 0$$

$$2. (A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}$$

**Лемма 10.1.**

$$\det(I + ab^T) = 1 + b^T a, \quad a, b \in \mathbb{R}^n$$

Доказательство.

$$(I + ab^\top)w = w, \quad w \perp b, \quad \lambda_1 = \dots = \lambda_{n-1} = 1.$$

$$(I + ab^\top)a = a(1 + b^\top a), \quad \lambda_n = 1 + b^\top a.$$

□

Доказательство. (Утверждение 1)

1.

$$\det(A + uv^\top) = \det(A)(1 + v^\top A^{-1}u)$$

2. Проверим, перемножив  $A + uv^\top$  и предполагаемое обратное:

$$(A + uv^\top) \left( A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u} \right) = I - \frac{uv^\top A^{-1}}{1 + \gamma} + uv^\top A^{-1} - \frac{u\gamma v^\top A^{-1}}{1 + \gamma} =$$

$$= I - \left( \frac{1}{1 + \gamma} - 1 + \frac{\gamma}{1 + \gamma} \right) uv^\top A^{-1} = I.$$

□

Подставим формулу из утверждения 1 в решение уравнения  $x = (A + uv^\top)^{-1}b$ , чтобы на практике не обращать матрицу:

$$x = A^{-1}b - \frac{(A^{-1}u)v^\top(A^{-1}b)}{1 + v^\top(A^{-1}u)} = x_1 - \frac{x_2(v^\top x_1)}{1 + v^\top x_2}.$$

Замечание: Есть также формула для поправки ранга  $r \geq 1$  (формула Шермана-Вудбери-Моррисона).

## 10.2 Форматы представления разреженных матриц

### 10.2.1 COO формат

3 массива: значения `val`, индексы столбцов `col`, индексы строк `row`.

Удобно добавлять элементы, но неэффективен для операций, например, `matvec` ( $y = Ax$ )

```
for i in range(nnz(A)):
    y[row[i]] += val[i] * x[col[i]]
```

### 10.2.2 lil (list of lists)

`col` нет, есть только `row`, который теперь список списков, и в нём для каждой строки хранятся индексы ненулевых элементов.

`val` тоже список списков, форма совпадает с формой `row`.

### 10.2.3 CSR (compressed sparse row)

`col` и `val` те же, что в COO, но в `row[i]` теперь находится самый первый индекс элемента из  $i$ -й строки в массивах `col`, `val`.

Matvec:

```
for i in range(n):
    p, q = row[i], row[i + 1]
    y[i] = dot(val[p : q], x[col[p : q]])
```

## 10.3 Заполнение в L и U

Матрице  $A \in \mathbb{R}^{n \times n}$  можно поставить в соответствие ориентированный граф:  $a_{ij} \neq 0$  задаёт наличие ребра из  $i$  в  $j$ .

Для симметричных матриц граф неориентированный.

Модель выше помогает заметить, что правильное сопряжение перестановкой может сделать  $L$  и  $U$  в LU-разложении разреженными (при условии разреженности исходной матрицы).

Для ленточных матриц это вообще неверно.

## 10.4 Алгоритм поиска P

### 10.4.1 Алгоритм Катхилла-Макки

Нужно для  $G = (V, E)$  найти такую перестановку номеров вершин  $\sigma$ , что метрика  $b$  минимальна:

$$b = \max_{(x,y) \in E} |\sigma(x) - \sigma(y)|.$$

Идея состоит в том, чтобы выбрать какую-нибудь вершину первой, а дальше нумеровать вершины в порядке обхода bfs.

### 10.4.2 Minimal degree ordering

## 11 Итерационные методы решения линейных уравнений (?)

### 11.1 Метод простой итерации

$$x_{k+1} = x_k + P(b - Ax_k) \equiv Gx_k + C$$

**Теорема 11.1.** Пусть  $\|\cdot\|$  — матричная и  $\|G\| < 1$ , тогда итерационная формула выше сходится к  $x_*$ :  $Ax_* = b$  при  $k \rightarrow \infty$  геометрически.

$$\|x_{k+1} - x_*\| = \|G(x_k - x_*)\| \leq \|G\| \|x_k - x_*\| \leq \dots \leq \|G\|^{k+1} \cdot \|x_0 - x_*\|.$$

Случай 1:  $P = D^{-1}$ ,  $D = \text{diag}(A)$  — метод Якоби.

**Теорема 11.2.** Пусть  $A$  обладает свойством диагонального преобладания, т.е.

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|.$$

Тогда метод Якоби сходится.

*Доказательство.*

$$G = I - PA = I - D^{-1}A = D^{-1}(D - A)$$

$$\|D^{-1}(D - A)\|_{\infty} = \max_i \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} < 1$$

□

Случай 2:  $P = (L + D)^{-1}$  — метод Гаусса-Зейделя. Сходимость для  $A = A^\top > 0$  будет доказана на семинаре.

Случай 3:  $P = \tau I, \tau > 0$  — итерация Рундсона.

**Теорема 11.3.** Пусть  $A = A^\top > 0$ . Тогда итерация Рундсона сходится при  $0 < \tau < \frac{2}{\lambda_{\max}(A)}$ . Более того, если

$$\tau = \tau_{\text{opt}} = \frac{2}{\lambda_{\min}(A) + \lambda_{\max}(A)},$$

тогда

$$\|x_{k+1} - x_*\|_2 \leq \frac{\text{cond}_2(A) - 1}{\text{cond}_2(A) + 1} \cdot \|x_k - x_*\|_2$$

*Доказательство.*

$$\|e_{k+1}\|_2 \leq \|G\|_2 \|e_k\|_2;$$

1.

$$\|G\|_2 = \|I - \tau A\|_2 = \max_i |\lambda_i(I - \tau A)| = \max_i |1 - \tau \lambda_i(A)| < 1.$$

Из того, что  $1 - \tau \lambda_{\min}(A) < 1$ , следует, что  $\tau \lambda_{\min} > 0$ . В силу положительной определённости матрицы получаем  $\tau > 0$ .

С другой стороны,

$$1 - \tau \lambda_{\max}(A) > -1;$$

$$2 > \tau \lambda_{\max}(A);$$

$$\tau < \frac{2}{\lambda_{\max}(A)}.$$

2.

$$\begin{aligned} \|I - \tau A\|_2 &= \max\{|1 - \tau \lambda_{\min}(A)|, |1 - \tau \lambda_{\max}(A)|\} = \\ &= \max\left\{\left|\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right|, \left|\frac{\lambda_{\min} - \lambda_{\max}}{\lambda_{\min} + \lambda_{\max}}\right|\right\} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\text{cond}(A)_2 - 1}{\text{cond}(A)_2 + 1}. \end{aligned}$$

□

## 11.2 Метод наискорейшего спуска

Делаем градиентный спуск по функционалу  $J(x)$ .

1.  $J(x) = \|x - x_*\|_2^2$  — вычисление равносильно решению уравнения.

2.  $J(x) = \|Ax - b\|_2^2$  — для общего случая, но для  $A = A^\top > 0$  можно лучше.

3.  $J(x) = \|x - x_*\|_A^2$  — для  $A = A^\top > 0$ . На первый взгляд зависит от решения, но

$$\|x - x_*\|_A^2 = (x - x_*)^\top A (x - x_*) = (x - x^*)^\top (Ax - b) = x^\top Ax - 2x^\top b + \text{const}$$

$J(x) = \frac{1}{2}x^\top Ax - x^\top b$  — функционал энергии.

$$\nabla J(x) = Ax - b$$

$$x_{k+1} = x_k - \tau_k \nabla J = x_k - \tau_k (Ax_k - b).$$

$$\begin{aligned} J(x_k + \tau r_k) &= \frac{1}{2} (x_k + \tau r_k)^\top A (x_k + \tau r_k) - (x_k + \tau r_k)^\top b = \\ &= \tau r_k^\top A x_k + \frac{1}{2} \tau^2 r_k^\top A r_k - \tau r_k^\top b + \text{const}(\tau) \rightarrow \min_{\tau} \\ \tau_k r_k^\top A r_k - r_k^\top r_k &= 0; \\ \tau_k &= \frac{r_k^\top r_k}{r_k^\top A r_k}. \end{aligned}$$

**Теорема 11.4.** Пусть  $A = A^\top > 0$ . Тогда метод наискорейшего спуска сходится и

$$\|x_{k+1} - x_*\|_A \leq \frac{\text{cond}_2(A) - 1}{\text{cond}_2(A) + 1} \|x_k - x_*\|_A.$$

*Доказательство.* 1.

$$\tau_k = \arg \min_{\tau} \sqrt{x_k + \tau r_k} = \arg \min_{\tau} \|x_k + \tau r_k - x_*\|_A$$

2.

$$\begin{aligned} \|x_{k+1} - x_*\|_A &= \min_{\tau} \|x_k + \tau r_k - x_*\|_A = \min_{\tau} \|(I - \tau A)e_k\|_A \leq \\ &\leq \min_{\tau} \|(I - \tau A)\sqrt{A}e_k\|_2 \leq \min_{\tau} \|I - \tau A\|_2 \cdot \|e_k\|_A. \end{aligned}$$

□

## 11.3 Итерационный метод Чебышева

$$\begin{aligned} x_{k+1} &= x_k + \tau_k (b - Ax_k); \\ e_{k+1} &= (I - \tau_k A)e_k = (I - \tau_k A) \dots (I - \tau_0 A)e_0 = \mathbb{P}_{k+1}(A) \cdot e_0 \\ \|e_{k+1}\|_2 &\leq \|p_{k+1}(A)\|_2 \|e_0\|_2, \quad p_{k+1}(0) = 1. \end{aligned}$$

### 11.3.1 Оценка ошибок

[...]

Многочлены Чебышёва — многочлены, наименее отклоняющиеся от нуля на отрезке  $[-1; 1]$ :

$$T_k(x) = \begin{cases} \cos(k \arccos(x)), & |x| \leq 1; \\ \cosh(k \text{arccosh}(x)), & |x| > 1. \end{cases}$$

Корни  $T_k$  легко считаются, через них получаем оценку на  $\tau_i = 1/\eta_i$ , где  $\eta_i$  — эти самые корни.

## 12 ???

### 12.1 Оптимизация на подпространствах Крылова

В прошлый раз мы рассмотрели несколько функционалов  $J$ , для которых верно

$$x_* = \arg \min_x J(x) \Leftrightarrow Ax_* = b.$$

Расширим концепцию и рассмотрим последовательность  $L_1 \subset L_2 \subset \dots \subset L_n$ , где  $\dim(L_k) = k$ .

$$x_k = \arg \min_{x \in x_0 + L_k} J(x)$$

Как выбрать  $L_k$ ?

$$L_k = \mathcal{K}_k(A, f) = \{f, Af, A^2f, \dots, A^{k-1}f\},$$

( $f$  — либо  $b$ , либо  $b - Ax_0$ )

Варианты для  $J$ :

1.  $\|Ax - b\|_2$  — методы minres, qmres
2.  $\|x - x_*\|_A$ , т.е. то же самое, что  $J(x) = \frac{1}{2}x^\top Ax - b^\top x$  — метод CG
3. QMR, bicgstab

При больших  $k$   $A^*f$  становятся почти линейно зависимыми, так что нужно будет их ортогонализировать. Если делать это с помощью Грамма-Шмидта, получающийся базис будет называться «векторы Арнольда».

**Утверждение 12.1.** (Соотношение Арнольда)

$$AQ_k = Q_{k+1}\hat{H}_k, \quad \hat{H}_k = \begin{bmatrix} H_k & & \\ 0 & \dots & 0 & h_{k+1,k}q_{k+1} \end{bmatrix},$$

где  $H_k$  — верхнехессенбергова матрица.

*Доказательство.*

$$\mathfrak{S}(AQ_{k-1}) \subset Q_k,$$

$$A\mathcal{K}_k = A[f, \dots, A^{k-1}f] = [Af, \dots, A^k f] \subset \mathcal{K}_{k+1}.$$

$$Aq_k \in \mathfrak{S}(Q_{k+1}).$$

$$q_{k+1} \cdot \text{const} = (I - Q_k Q_k^\top) Aq_k = Aq_k - (q_1^\top Aq_k)q_1 - \dots - (q_k^\top Aq_k)q_k.$$

□

Следствия:  $Q_k^\top AQ_k$  — верхнехессенбергова,  $Q_k^\top AQ_k$  — трёхдиагональная (???).

## 12.2 Метод сопряжённых градиентов

Обозначения:

$$(x, y)_A = x^\top Ay = (x, Ay);$$

$$x \perp_A y \Leftrightarrow (x, y)_A = 0.$$

Утверждение 12.2.

$$x_k = \arg \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|x_* - x\|_A \Leftrightarrow x_* - x_k \perp_A \mathcal{K}_k \Leftrightarrow r_k \perp \mathcal{K}_k$$

Доказательство.

$$\|x_* - x\|_A = \|x_* - x_0 - y\|_A \rightarrow \min_{y \in \mathcal{K}_k}$$

$P^A$  —  $A$ -ортогональный проектор на  $\mathcal{K}_k$ :  $(P^A)^2 = P^A$ ;  $(P^A x, y)_A = (x, P^A y)_A$ .

$$x_* - x_0 = P^A(x_* - x_0) + (I - P^A)(x_* - x_0)$$

$$\|x_* - x_0 - y\|_A^2 = \left\| P^A(x_* - x_0) - y + (I - P^A)(x_* - x_0) \right\|_A^2 = \left\| P^A(x_* - x_0) - y \right\|_A^2 + \text{const} \rightarrow \min_y$$

$$y = P^A(x_* - x_0).$$

□

Из утверждения 12.1 следует, что  $Q_k^\top A Q_k$  — трёхдиагональная. Если  $A$  ещё и положительно определённая, то мы можем применить к  $Q_k^\top A Q_k$  разложение Холецкого.

$$Q_k^\top A Q_k = R_k^\top R_k$$

$$(Q_k R_k^{-1} D_k)^\top A Q_k R_k^{-1} D_k = D_k^2$$

$$P_k^\top A P_k = D_k^2, \quad p_i \perp_A p_j.$$

$$x_k = x_0 + P_k a_k, \quad a_k = [\alpha_1, \dots, \alpha_k]^\top$$

$$r_k = r_0 - A P_k a_k, \quad r_k \perp \mathfrak{S}(P_k)$$

$$0 = P_k^\top r_k = P_k^\top r_0 - P_k^\top A P_k a_k$$

$$D_k^2 a_k = P_k^\top r_0$$

$$x_k = x_0 + P_k a_k = x_0 + P_{k-1} a_{k-1} + \alpha_k P_k$$

$$x_k = x_{k-1} + \alpha_k p_k$$

$$r_k = r_{k-1} - \alpha_k A p_k$$

Как искать  $\alpha_k$  и  $p_k$ ?

$$r_k \perp p_k, p_k^\top r_k = p_k^\top r_{k-1} - \alpha_k p_k^\top A p_k$$

$$\alpha_k = \frac{p_k^\top r_{k-1}}{p_k^\top A p_k}$$

$$r_k = r_0 - A p_k a_k \in \mathcal{K}_{k+1}$$

$$r_k \perp \mathfrak{S}(P_k) = \mathcal{K}_k$$

$$r_k \parallel q_{k+1}$$

$$\begin{aligned}
q_{k+1} &= \frac{r_k}{\|r_k\|_2} \\
P_k &= Q_k R_k^{-1} D_k \\
Q_k &= P_k D_k^{-1} R_k \\
q_k &= p_k \frac{\gamma_k}{d_k} + p_{k-1} \frac{\delta_{k-1}}{d_{k-1}} \\
q_{k+1} &= p_{k+1} \frac{\gamma_{k+1}}{d_{k+1}} + p_k \frac{\delta_k}{d_k} \\
p_{k+1} &= \frac{d_{k+1}}{\gamma_{k+1}} q_{k+1} - \frac{d_{k+1}}{\gamma_{k+1}} \frac{\delta_k}{d_k} p_k = r_k + \beta_k p_k \\
p_{k+1} \perp_A p_k &\quad \Rightarrow \quad \beta_k = \dots
\end{aligned}$$

Итог:

$$\begin{aligned}
\alpha_k &= \frac{(r_{k-1}, p_k)}{(Ap_k, p_k)} \\
x_k &= x_{k-1} + \alpha_k p_k \\
r_k &= r_{k-1} - \alpha_k Ap_k \\
\beta_k &= -\frac{(r_k, Ap_k)}{(p_k, Ap_k)} \\
p_{k+1} &= r_k + \beta_k p_k
\end{aligned}$$