Лекции по Численным методам

Лектор: Тараканов Александр

Весна 2024

Содержание

1 Прямые методы решения систем линейных уравнений

Постановка задачи

Рассмотрим систему уравнений:

$$\begin{cases} A_{11}x_1 + \dots A_{1n}x_n = y_1 \\ \dots \\ A_{n1}x_1 + \dots + A_{nn}x_n = y_n \end{cases} \tag{1}$$

Система уравнений (1) называется линейной системой уравнений (СЛУ). A_{ij} — матрица коэффициентов, y_i — правая часть и x_i — неизвестные

Про СЛУ естественно говорить на языке линейных пространств и операторов: пусть задан лиценый оператор $A : X \longrightarrow Y$, вектор $y \in Y$ и требуется найти его прообраз $x \in X : Ax = y$

Предположения

В общем случае при решении СЛУ возможны несколько случаев:

- Решение существует и единственно $\ker A = 0$ и $y \in \operatorname{Im} A$
- Решение существует, но не единственно $\ker A \neq 0$ и $y \in \operatorname{Im} A$
- Решения не существует $y \notin \operatorname{Im} A$

В данном курсе мы будем рассматривать только частный случай: $\ker A = 0$ и $\operatorname{Im} A = Y$ или $\operatorname{coker} A = 0$

Иными словами: будем считать, что X и Y — линейные пространства одинаковой размерности n, матрица СЛУ A является несингулярной (квадратной и невырожденной)

В такой постановке задача $Ax = y$ определена корректно и решение существует и единственно для любой правой части

Далее в курсе будут разбираться различные алгоритмы поиска решения в зависимости от свойств матрицы A

1.1 Нижние и верхние треугольные матрицы

Определение 1. Матрица L называется *нижней треугольной* матрицей, если $L_{ij} = 0$, если $j > i$

Определение 2. Матрица U называется *верхней треугольной* матрицей, если $U_{ij} = 0$, если $j < i$

Нижние и верхние треугольные матрицы обладают следующим важным свойством

Утверждение 3. *Пусть A и B нижние (верхние) треугольные матрицы, то и матрица $C = AB$ является нижней (верхней) треугольной матрицей*

Доказательство. Для доказательства вычислим значение матричного элемента C_{ij} при $j > i$ для нижних треугольных матриц:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj} = \sum_{k=1}^{i} A_{ik} \cdot B_{kj} + \sum_{k=i+1}^{n} A_{ik} \cdot B_{kj} = \sum_{k=1}^{i} A_{ik} \cdot 0 + \sum_{k=i+1}^{n} 0 \cdot B_{kj} = 0$$

□

Решение системы с нижней треугольной матрицей

Рассмотрим СЛУ с нижней треугольной матрицей L:

$$\begin{cases} L_{11}x_1 & = y_1 \\ L_{21}x_1 + L_{22}x_2 & = y_2 \\ \dots & \\ L_{n1}x_1 + \dots + L_{nn}x_n & = y_n \end{cases}$$

Тогда решение можно найти, исключая неизвестные:

$$
\begin{aligned}
&x_1 = y_1/L_{11} \\
&x_2 = (y_2 - L_{21}x_1)/L_{22} \\
&\dots \\
&x_i = \left(y_i - \sum_{j=1}^{i-1} L_{ij}x_j\right)/L_{ii} \\
&\dots
\end{aligned}
$$

Решение системы с верхней треугольной матрицей

Рассмотрим СЛУ с верхней треугольной матрицей U:

$$
\begin{cases}
U_{11}x_1 + \ldots + U_{1n}x_n & = y_1 \\
\dots & \\
U_{(n-1)(n-1)}x_{n-1} + \ldots + U_{(n-1)n}x_n & = y_{n-1} \\
U_{nn}x_1 & = y_n
\end{cases}
$$

Тогда решение можно найти, исключая неизвестные:

$$
\begin{aligned}
&x_n = y_n/U_{nn} \\
&x_{n-1} = (y_{n-1} - U_{(n-1)n}x_{n-1})/U_{(n-1)(n-1)} \\
&\dots \\
&x_i = \left(y_i - \sum_{j=i+1}^{n} U_{ij}x_j\right)/U_{ii} \\
&\dots
\end{aligned}
$$

1.2 Решение системы уравнений с помощью LU разложения

Пусть есть СЛУ $Ax = y$

Идея алгоритма состоит в следующем:

- Представить матрицу A в виде произведения: $A = LU$, где L — нижняя треугольная матрица, U — верхняя треугольная
- Решить систему $Lz = y$
- Решиить систему $Ux = z$

Доказательство. Пусть x — решение исходной системы. Тогда

$$Ax = LUx = L(Ux) = Lz = y$$

□

Замечание

- В общем случае LU разложение не определено однозначно: если D — невырожденная диагональная матрица, то можно построить другое LU разложение по уже имеющемуся:

  $$A = LU = LDD^{-1}U = (LD)(D^{-1}U) = L'U'$$

  Данную неопределенность можно решить, зафиксировав, что $L_{ii} = 1$ или $U_{ii} = 1$

Алгоритм построения LU разложения

Рассмотрим случай, когда диагональ верхней треугольной матрицы $U_{ii} = 1$. Будем вычислять элементы матриц L и U построчно:

- Первая строка: $L_{11}U_{11} = A_{11}$ и $U_{11} = 1$, поэтому $L_{11} = A_{11}/U_{11}$ (перемножили 1-ую строку и 1-ый столбец). Далее вычислим U_{1i}: (перемножаем 1-ую строку и i-ый столбец)

$$L_{11}U_{1i} = A_{1i} \iff U_{1i} = A_{1i}/L_{11}$$

- Вторая строка: $L_{21}U_{11} = A_{21}$ и $U_{11} = 1$, поэтому $L_{21} = A_{21}/U_{11}$ (перемножили 1-ую строку и 2-ый столбец). Далее вычислим L_{22}: (перемножим 2-ую строку и 2-ый столбец)

$$L_{21}U_{12} + L_{22}U_{22} = A_{22} \iff L_{22} = (A_{22} - L_{21}U_{12})/U_{22}$$

  Теперь вычислим U_{2i}: (перемножим 2-ую строку и i-ый столбец)

$$L_{21}U_{12} + L_{22}U_{2i} = A_{2i} \iff U_{2i} = (A_{2i} - L_{21}U_{12})/L_{22}$$

  И так далее

- В итоге получаем, что вычислить i-ую строку матрицы L, можно следующим образом: ($j \leqslant i$)

$$L_{ij} = \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik}U_{kj}\right)/U_{jj}$$

  А i-ая строка матрицы U вычисляется так: ($j > i$)

$$U_{ij} = \left(A_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj}\right)/L_{ii}$$

Замечание Если мы задаем диагональные элементы нижней треугольной матрицы L, то задача сводится к уже решенной:

$$A^\top = L'U' \iff A = (U')^\top (L')^\top = LU$$

Данный алгоритм позволяет найти LU разложение для матрицы A единственным образом, если зафиксировать диагональные элементы матрицы U или L. Однако если на m-ом шаге окажется, что $L_{mm} = 0$, то алгоритм позволяет найти только лишь LU разложение главного минора порядка m матрицы A:

$$[A]_m = [L]_m [U]_m,$$

где $[L]_m$ и $[U]_m$ — нижняя и верхняя треугольные матрицы, вычисленные за m шагов

В общем случае, данный алгоритм не гарантирует сходимости к LU разложению для матрицы A, однако существует класс матриц A, для которых алгоритм корректно находит LU разложение

Класс матриц

Необходимо проверить, есть ли нули на диагонали матрицы L, тогда вышеописанный алгоритм будет работать корректно

Утверждение 4. *Если* $\forall m \in \{1, \dots n\} : \det[A]_m \neq 0$, *то* $L_{mm} \neq 0$

Доказательство. Докажем от противного. Зафиксируем $m \in \{1, \dots, n\}$ и $\det[A]_m \neq 0$. Пусть $L_{ii} \neq 0$ при $i < m$ и $L_{mm} = 0$. Тогда верно, что

$$[A]_m = [L]_m[U]_m \implies \det[A]_m = \det[L]_m \cdot \det[U]_m = \prod_{i=1}^{m} L_{ii} \cdot \prod_{i=1}^{m} U_{ii} = 0$$

Получили противоречие, что определитель главного минора $[A]_m$ не равен 0 □

В общем случае данное утверждение сложно проверить. Если матрица является симметричной положительно определенной (SPD), то тогда оно выполнено по критерию Сильвестра

Помимо SPD матриц, часто встречаются матрицы со следующим особым свойством

Определение 5. Матрица A называется *матрицей с диагональным преобладанием*, если $\forall\, i \in \{1, \dots, n\}$

$$|A_{ii}| - \sum_{j \neq i} |A_{ij}| > 0$$

Видно, что главный минор такой матрицы также является матрицей с диагональным преобладанием. Докажем следующее утверждение

Утверждение 6. *Матрица с диагональным преобладанием является несингулярной*

Доказательство. Докажем от противного. Пусть матрица вырожденная. Тогда $\exists\, x \in \mathbb{R}^n : Ax = 0$. Найдем максимальный по модулю элемент в векторе $x : |x_i| = \max_j |x_j|$ и рассмотрим i-ую строку Ax:

$$0 = \left| \sum_j A_{ij} x_j \right| = |x_i| \cdot \left| \sum_j A_{ij} \frac{x_j}{|x_i|} \right| = |x_i| \cdot \left| A_{ii} + \sum_{j \neq i} A_{ij} \cdot \frac{x_j}{|x_i|} \right| \geqslant |x_i| \cdot \left| A_{ii} - \sum_{j \neq i} |A_{ij}| \cdot \frac{|x_j|}{|x_i|} \right| > 0$$

Предпоследнее неравенство верно по обратному неравенства треугольника. Последнее неравенство верно, так как матрица A является матрицей с диагональным преобладанием, а отношение $|x_j|/|x_i| < 1$ при $j \neq i$.

Получили противоречие □

1.3 Решение системы уравнений с помощью разложения Холецкого

Пусть есть СЛУ $Ax = y$ и матрица A является SPD матрицей. Тогда существует специальный вид (причем единственный) LU разложения — разложение Холецкого:

$$A = LL^\top,$$

где L — нижняя треугольная матрица с положительными элементами на диагонали

Утверждение 7. *Существование разложения Холецкого*

Доказательство. Пусть задано какое-то LU разложение: $A = LU$ (существование его доказывалось ранее). Тогда верно следующее:

$$LU = A = A^\top = U^\top L^\top$$

Домножим слева равенство на L^{-1}:

$$U = L^{-1} U^\top L^\top$$

Теперь домножим справа равенство на $(L^\top)^{-1}$:

$$U (L^\top)^{-1} = L^{-1} U^\top = D$$

Получим, что слева у нас верхняя треугольная матрица, а справа нижняя треугольная матрица, поэтому и справа, и слева диагональная матрица D

Рассмотрим исходное LU разложение:

$$A = LU = L \cdot (DL^\top) = LD^{1/2} D^{1/2} L^\top = (LD^{1/2})(LD^{1/2})^\top = L' L'^\top$$

Причем диагональные элементы D могут быть только положительными, так как A является SPD матрицей и L — матрица перехода от одного базиса к другому

□

Утверждение 8. *Единственность разложения Холецкого*

Доказательство. Единственность доказывается вместе с построением алгоритма вычисления, аналогичному для LU разложения

- $L_{11}L_{11} = A_{11} \iff L_{11} = \sqrt{A_{11}}$
- i-ая строка при $j < i$:

$$\sum_{k=1}^{j-1} L_{ik}L_{kj} + L_{ij}L_{jj} = A_{ij} \iff L_{ij} = \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{kj}\right)/L_{jj}$$

- Диагональные элементы L_{ii}:

$$L_{ii} = \sqrt{A_{ii} - \sum_{k=1}^{j-1} L_{ik}L_{kj}}$$

$\square$

1.4 Решение систем уравнений с помощью алгоритма Гаусса

Пусть есть СЛУ $Ax = y$

Будем решать ее алгоритмом Гаусса: сначала применять прямой алгоритм Гаусса, потом обратный. Данный алгоритм является одним из способов построения LU разложения

Во время прямого алгоритма Гаусса мы будем приводить матрицу A к ступенчатому виду, выполняя операции над строками:

- 1 тип: $i \mapsto i \cdot \lambda, \lambda \neq 0$
- 2 тип: $i \leftrightarrow j$
- 3 тип: $i \mapsto i + j \cdot \lambda$

Во время обратного хода мы теми же действиями будем приводить матрицу A к улучшенному ступенчатому виду, чтобы главные коэффициенты были равны 1

Каждой операции над строками однозначно сопоставляется умножение слева на матрицу

Сложность алгоритма Гаусса составляет $O(n^3)$, где n — разномерность матрицы A. Поэтому данные алгоритмы не используются для решения СЛУ больших размерностей

2 Норма и анализ сходимости

Определение 9. Пусть задано линейное векторное пространство V над полем $\mathbb{R}$.
Функцию $\|\cdot\| : V \longrightarrow \mathbb{R}$ будем называть *нормой*, если выполнены следующие свойства:

- $\forall\, x \in V : \|x\| \geqslant 0$
- $\|x\| = 0 \iff x = 0$
- $\|\alpha x\| = |\alpha| \cdot \|x\|$
- $\forall\, x, y \in V : \|x + y\| \leqslant \|x\| + \|y\|$ — неравенство треугольника

Пространство V с нормой $\|\cdot\|$ называется *нормированным пространством*

2.1 Векторные l_p нормы

Важным примером норм является l_p норма. Пусть $V = \mathbb{R}^n$. Тогда для $x \in V$, который будем записывать в виде вектор-столбца $x = [x_1, \ldots, x_n]^\top$, определим l_p норму:

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

Существуют особые виды l_p нормы:

- $p = 1 : \|x\|_1 = |x_1| + \ldots + |x_n|$
- $p = 2 : \|x\|_2 = \sqrt{|x_1|^2 + \ldots + |x_n|^2}$
- $p = \infty : \|x\|_\infty = \max\limits_i |x_i|$

Утверждение 10. *Докажем, что приведенные функции являются нормами*

Доказательство. Разберем каждый случай по отдельности:

1. Случай $p = 1$
   - $\|x\|_1 \geqslant 0$ очевидно. Пусть $\|x\| = 0$. Тогда $|x_1| + \ldots + |x_n| = 0 \iff x_1 = \ldots = x_n = 0 \iff x = 0$. В обратную сторону очевидно
   - $\|\alpha x\|_1 = |\alpha x_1| + \ldots + |\alpha x_n| = |\alpha| \cdot (|x_1| + \ldots + |x_n|) = |\alpha| \cdot \|x\|_1$
   - Зафиксируем $x, y \in V$. Тогда

   $$\|x + y\|_1 = \sum_{i=1}^{n} |x_i + y_i| \leqslant \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |y_i| = \|x\|_1 + \|y\|_1$$

2. Случай $p = 2$
   - Первые три свойства проверяются аналогично. Докажем только неравенство треугольника. Зафиксируем $x, y \in V$ и воспользуемся неравенством Коши-Буняковского:

   $$\|x+y\|_2^2 = \sum_{i=1}^{n} |x_i+y_i|^2 = \sum_{i=1}^{n} |x_i|^2 + \sum_{i=1}^{n} |y_i|^2 + 2\sum_{i=1}^{n} |x_i| \cdot |y_i| \leqslant \|x\|_2^2 + \|y\|_2^2 + 2\|x\|_2 \cdot \|y\|_2 = (\|x\|_2 + \|y\|_2)^2$$

3. Случай $p \in \mathbb{R} : p > 1$
   - Первые три свойства проверяются аналогично. Докажем неравенство треугольника

- Предположим, что $x, y \in V : x \neq 0, y \neq 0$ и $\forall\, i \in \{1, \dots, n\} : x_i \geqslant 0$ без ограничения общности рассуждений.
  Зафиксируем x и будем искать максимум $f(y) = \|x+y\|_p$ по y при условии, что $\|y\|_p = C = \text{const}$
  Из курса математического анализа известно, что непрерывная функция на компакте достигает своего максимума. Пусть $f(y)$ достигает максимума в точке $y^\star$
  Тогда запишем уравнение касательной плоскости к поверхности $\|y\|_p = C$:
  ($dy = [dy_1, \dots, dy_n]^\top$ — вектор приращений)

$$\sum_{i=1}^{n} \frac{\partial}{\partial y_i} \|y\|_p^p \, dy_i = \sum_{i=1}^{n} p|y_i|^{p-1} \, dy_i = 0 = \left\langle \nabla \|y\|_p^p, \, dy \right\rangle \tag{2}$$

- Так как $y^\star$ — точка экстремума функции, то найдем производную $f(y)^p$:

$$\frac{d}{dy} f(y^\star)^p = \sum_{i=1}^{n} p|x_i + y^\star|^{p-1} \, dy_i = 0 = \left\langle \nabla f(y^\star)^p, \, dy \right\rangle \tag{3}$$

- Из (2) в точке $y^\star$ и (3) следует, что векторы

$$\nabla \|y^\star\|_p^p = \left[|y_1^\star|^{p-1}, \dots, |y_n^\star|^{p-1} \right]^\top$$

  и

$$\nabla f(y^\star)^p = \left[|x_1 + y_1^\star|^{p-1}, \dots, |x_n + y_n^\star|^{p-1} \right]^\top$$

  перпендикулярны вектору приращений dy, а значит коллинеарны:

$$|x_i + y_i^\star| = \lambda |y_i^\star|$$

- Так как $y^\star$ — точка максимума, то в знаки x_i и $y_i^\star$ должны совпадать. То есть $y_i = kx_i$ для некоторого $k > 0$, которое можно найти следующим образом:

$$k = \frac{\|y\|_p}{\|x\|_p} = \frac{C}{\|x\|_p}$$

$$\|x+y\|_p \leqslant \|x + y^\star\|_p = \|x + kx\|_p = \|x\|_p + \|kx\|_p = \|x\|_p + \|y\|_p$$

4. Случай $p = \infty$

- Рассмотрим следующий предел:

$$\lim_{p \to \infty} \|x\|_p$$

- Пусть дан вектор $x \in V : \|x\|_\infty = |x_k|$. Тогда

$$\|x\|_\infty = |x_k| = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \leqslant \|x\|_p \leqslant \left(\sum_{i=1}^{n} |x_k|^p \right)^{1/p} = n^{1/p} \|x\|_\infty = n^{1/p} |x_k|$$

- Отсюда и из $n^{1/p} \longrightarrow 1$ при $p \longrightarrow \infty$ видно, что $\|x\|_\infty = \lim\limits_{p \to \infty} \|x\|_p$
- Теперь мы можем доказать, что $\|x\|_\infty$ является нормой по предельным переходам

□

Определение 11. Пусть задано линейное векторное пространство V над полем $\mathbb{R}$.
Функцию $\rho : V \times V \longrightarrow \mathbb{R}$ будем называть *метрикой*, если выполнены следующие свойства:

- $\forall\, x, y \in V : \rho(x, y) \geqslant 0$
- $\rho(x, y) = 0 \iff x = y$

- $\forall x, y \in V : \rho(x, y) = \rho(y, x)$
- $\forall x, y, z \in V : \rho(x + z) \leqslant \rho(x, y) + \rho(y, z)$ — неравенство треугольника

Пространство V с метрикой ρ называется *метрическим пространством*

Заметим, что любая норма на линейном пространстве задает метрику:

$$\rho(x, y) = \|x - y\|$$

2.2 Сходимость по норме

Определение 12. Две нормы $\|\cdot\|_a$ и $\|\cdot\|_b$ (необязательно l_p нормы) на нормированном пространстве V называются эквивалентными, если $\exists\, C_1, C_2 > 0 : \forall\, x \in V$

$$C_1 \cdot \|x\|_a \leqslant \|x\|_b \leqslant C_2 \cdot \|x\|_b$$

Известно, что на конечномерных пространствах все нормы являются эквивалентными. Будем говорить, что последовательность векторов $\{x_k\}$ сходится к x по норме, если $\|x_k - x\| \longrightarrow 0$ при $k \longrightarrow \infty$. Так как все нормы являются эквивалентными, то для исследования сходимости можно использовать любую норму. Также для конечномерных пространств верно, что из покоордиантной сходимости следует сходимость по норме и наоборот

2.3 Норма линейного оператора

Определение 13. Пусть задано нормированное пространство V с нормой $\|x\|$. Пусть задан линейный оператор $A : V \longrightarrow V$. Определим норму линейного оператора следующим образом:

$$\|A\| = \sup_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|}$$

Утверждение 14. *Докажем, что это действительно норма, и перечислим свойства*

- Первые три свойства нормы выполняются
- Проверим неравенство треугольника:

$$\|A + B\| = \sup_{\|x\| \neq 0} \frac{\|Ax + Bx\|}{\|x\|} \leqslant \sup_{\|x\| \neq 0} \left(\frac{\|Ax\|}{\|x\|} + \frac{\|Bx\|}{\|x\|} \right) \leqslant \sup_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} + \sup_{\|x\| \neq 0} \frac{\|Bx\|}{\|x\|} = \|A\| + \|B\|$$

- Видно из определения нормы, что

$$\|Ax\| \leqslant \|A\| \cdot \|x\|$$

- Оценим сверху норму композиции операторов BA:

$$\|BAx\| = \|B(Ax)\| \leqslant \|B\| \cdot \|Ax\| \leqslant \|B\| \cdot \|A\| \cdot \|x\|$$

  Поэтому $\|BA\| \leqslant \|B\| \cdot \|A\|$

- Из линейности оператора следует, что

$$\sup_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\| \neq 0} \left\| A \cdot \left(\frac{x}{\|x\|} \right) \right\| = \sup_{\|x\| = 1} \|Ax\|$$

  По этой причине норма любого линейного оператора на конечномерном линейном пространстве с l_p нормой существует и конечна

Примеры

- Норма диагонального оператора $D_{n\times n} = D$ с помощью l_p нормы:

$$\sup_{\|x\|=1} \|Dx\|_p = \left(\sum_{i=1}^{n} |D_{ii}|^p \cdot |x_i|^p\right)^{1/p}$$

Пусть $D_{kk} = \max_i |D_{ii}|$. Тогда

$$\left(\sum_{i=1}^{n} |D_{ii}|^p \cdot |x_i|^p\right)^{1/p} \leqslant \left(\sum_{i=1}^{n} |D_{kk}|^p \cdot |x_i|^p\right)^{1/p} = |D_{kk}| \left(\sum_{i=1}^{n} |x_i|\right)^{1/p}$$

Получается, что

$$\sup_{\|x\|=1} \|Dx\|_p \leqslant \max_i |D_{ii}|$$

Пример, на котором достигается максимум легко построить: возьмем $x = [0, \dots, 1, \dots, 0]^\top$ — ненулевая координата только на k-ой позиции

- Рассмотрим l_2 норму на конечномерном линейном пространстве V. Пусть есть некоторый оператор A. Известно, что любой оператор можно разложить в виде композиции поворотов, отражений и растяжений вдоль осей с положительными коэффициентами — SVD:

$$A = U_1 D U_2,$$

где U_1, U_2 — ортогональные матрицы, которые сохраняют расстояние $\|U_1 x\| = \|U_2 x\| = \|x\|$

$$\|A\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|=1} \|U_1 D U_2 x\| = \sup_{\|x\|=1} \|D U_2 x\| = \sup_{\|x\|=1} \|Dx\| = \max_i D_{ii}$$

- Рассмотрим l_2 норму на конечномерном линейном пространстве V. Пусть есть некоторый самосопряженный оператор A (заданный SPD матрицей). Тогда представим его в следующем виде:

$$A = UDU^\top,$$

где U — ортогональная матрица, D — диагональная матрица из собственный значений λ_i. Аналогично предыдущему пункту:

$$\|A\| = \max_i \lambda_i$$

- Рассмотрим l_∞ норму на конечномерном линейном пространстве V и произвольный оператор A

$$\|A\| = \sup_{\|x\|=1} \|Ax\|_\infty = \max_i \left|\sum_j A_{ij} x_j\right| = \left|\sum_j A_{kj} x_j\right|$$

Заметим, что A_{kj} и x_j должны быть одного знака, иначе можно поменять знак координаты j вектора x на противоположный и значение увеличится, что противоречит максимальности выражения. Так как $\|x\|_\infty = 1$, тогда есть координата $|x_m| = 1$. Тогда заменим все x_j на 1, от этого норма не изменится. В итоге получили, что

$$\|A\| = \max_i \left|\sum_j A_{ij}\right|$$

2.4 Число обусловленности

Определение 15. Пусть на нормированном конечномерном пространстве V задан невырожденный линейный оператор $A : V \longrightarrow V$. *Числом обусловленности* линейного оператора будем называть следующее выражение:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

Видно, что $\kappa(A) = \kappa(A^{-1})$

Утверждение 16. *Число обусловленности* $\kappa(A) \geqslant 1$

Доказательство.

$$\|x\| = \|A^{-1}Ax\| \leqslant \|A^{-1}\| \cdot \|Ax\| \leqslant \|A^{-1}\| \cdot \|A\| \cdot \|x\| = \kappa(A) \cdot \|x\|$$

□

Утверждение 17. *Пусть A — самосопряженный линейный оператор и задана l_2 норма. Тогда число обусловленности равно*

$$\kappa(A) = \frac{\max_i \lambda_i}{\min_i \lambda_i},$$

где λ_i — собственное значение матрицы A

Доказательство. Докажем, что

$$\kappa(A) = \frac{\sup_{\|x\|=1}\|Ax\|}{\inf_{\|x\|=1}\|Ax\|}$$

- Так как A — самосопряженный оператор, то $A = U^\top DU$ и $A^{-1} = U^\top D^{-1}U$
- Найдем $\|A^{-1}\|$

$$\|A^{-1}\| = \sup_{\|x\|\neq 0} \frac{\|A^{-1}x\|}{\|x\|} = \sup_{\|y\|\neq 0} \frac{\|A^{-1}Ay\|}{\|Ay\|} = \sup_{\|y\|\neq 0} \frac{\|y\|}{\|Ay\|} = \frac{1}{\inf_{\|x\|=1}\|Ax\|}$$

- Отсюда и так как задана l_2 норма получаем нужное равенство через собственные значения

□

В общем случае, число обусловленности показывает, насколько матрица близка к сингулярной: чем больше число обусловленности, тем ближе к сингулярности

2.5 Число обусловленности и устойчивость решения системы уравнений

Рассмотрим СЛУ $Ax = b$. Допустим, что правая часть b известна с точностью до ошибок Δb. Тогда мы решаем систему $Ax^\star = b + \Delta b$

Пусть погрешность тогда равна $\Delta x = x^\star - x$. Оценим относительную ошибку:

$$\frac{\|\Delta x\|}{\|x\|} = \frac{\|A^{-1}\Delta b\|}{\|x\|} \leqslant \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|b\|} \cdot \frac{\|b\|}{\|x\|} = \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|b\|} \cdot \frac{\|Ax\|}{\|x\|} \leqslant \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\Delta b\|}{\|b\|} = \kappa(A) \cdot \frac{\|\Delta b\|}{\|b\|}$$

Рассмотрев $A^{-1}b = x$ аналогично можно получить следующую оценку:

$$\frac{1}{\kappa(A)} \cdot \frac{\|\Delta b\|}{\|b\|} \leqslant \frac{\|\Delta x\|}{\|x\|} \leqslant \kappa(A) \cdot \frac{\|\Delta b\|}{\|b\|}$$

СЛУ, где матрица A имеет большое число обусловленности, могут иметь неустойчивые решения, которые могут сильно отличаться от аналитического решения

3 Итерационные методы

Постановка задачи

Рассмотрим СЛУ $Ax = y$. Предположим, что матрица A является матрицей с диагональным преобладанием

3.1 Метод Якоби

Описание алгоритма

Рассмотрим i-ую строку:

$$\sum_{k=1}^{n} A_{ik}x_k = b_i \iff x_i = \left(b_i - \sum_{k\neq i} A_{ik}x_k\right)/A_{ii}$$

Будем теперь находить решение СЛУ итерационно, сперва проинициализировав $x_i^{(1)}$: (t — номер итерации)

$$x_i^{(t+1)} = \left(b_i - \sum_{k\neq i} A_{ik}x_k^{(t)}\right)/A_{ii}$$

Пусть D — матрица диагональных элементов матрицы A. Тогда итерационный процесс можно записать в матричной форме:

$$x^{(t+1)} = (I - D^{-1}A)x^{(t)} + D^{-1}b$$

Сходимость метода Якоби

Утверждение 18. *Метод Якоби сходится для любой стартовой точки*

Доказательство. Докажем, что $x^{(t)}$ сходится по критерию Коши

- Рассмотрим $\Delta_t = x^{(t+1)} - x^{(t)}$.

  $$\Delta_t = (I - D^{-1}A)x^{(t)} + D^{-1}b - (I - D^{-1}A)x^{(t-1)} - D^{-1}b = (I - D^{-1}A)(x^{(t)} - x^{(t-1)}) = (I - D^{-1}A)\Delta_{t-1}$$

  Пусть $G = (I - D^{-1}A)$ — будем называть *итерационным оператором*. Покажем, что $\|G\| < 1$ при l_p норме, где $p = \infty$

  $$\|G\|_\infty = \max_i \sum_j |G_{ij}| = \max_i \left(\sum_j \frac{|A_{ij}|}{|A_{ii}|} - 1\right) = \max_i \sum_{i\neq j} \frac{|A_{ij}|}{|A_{ii}|} < 1,$$

  так как A матрица с диагональным преобладанием

- Теперь оценим $\|\Delta_t\|_\infty$:

  $$\|\Delta_t\|_\infty = \|G\Delta_{t-1}\|_\infty \leqslant \|G\|_\infty \|\Delta_{t-1}\|_\infty \leqslant \ldots \leqslant \|G\|_\infty^{t-1} \cdot \|\Delta_1\|_\infty$$

- Рассмотрим теперь критерий Коши:

  $$\|x^{(t+q)} - x^{(t)}\|_\infty = \left\|x^{(1)} + \sum_{i=1}^{t+q-1} \Delta_i - x^{(1)} - \sum_{i=1}^{t-1} \Delta_i\right\|_\infty = \left\|\sum_{i=t}^{t+q-1} \Delta_i\right\|_\infty \leqslant \sum_{i=t+1}^{t+q} \|\Delta_i\| \leqslant \sum_{i=t+1}^{t+q} \|G\|_\infty^{i-1} \cdot \|\Delta_1\|_\infty =$$
  $$= \|G\|_\infty^t \cdot \frac{1 - \|G\|_\infty^{t+q}}{1 - \|G\|_\infty} \cdot \|\Delta_1\|_\infty$$

Так как $\|G\|_\infty < 1$, то при $t \longrightarrow \infty$

$$\|x^{(t+q)} - x^{(t)}\|_\infty \leqslant \|G\|_\infty^t \cdot \frac{1}{1 - \|G\|_\infty} \cdot \|\Delta_1\|_\infty \longrightarrow 0$$

- Так как последовательность $x^{(t)}$ фундаментальная, то она сходится к некоторому $x^\star$, что и будет решением СЛУ. Переходя к пределу в равенстве для $x^{(t+1)}$, получаем, что

$$x^\star = (I - D^{-1}A)x^\star + D^{-1}b \iff Ax^\star = b$$

□

3.2 Общий вид итерационного алгоритма

Рассмотрим СЛУ $Ax = y$. Пусть Q — обратимая матрица (матрица расщепления или splitting matrix). Тогда можем преобразовать СЛУ следующим образом:

$$Qx = (Q - A)x + y$$

Тогда легко видеть, что решение x вычисляется следующим образом:

$$x = (I - Q^{-1}A)x + Q^{-1}y$$

Тогда построим итерационную последовательность $x^{(t)}$:

$$x^{(t+1)} = (I - Q^{-1}A)x^{(t)} + Q^{-1}y$$

В общем виде последовательность имеет вид:

$$x^{(t+1)} = Gx^{(t)} + c$$

3.3 Спектральный радиус и сходимость

Определение 19. *Спектральным радиусом* линейного оператора A называется следующая величина:

$$\rho(A) = \sup\{|\lambda| : \lambda \in \operatorname{spec}(A)\},$$

где $\operatorname{spec}(A)$ — множество собственных значений (*спектр*) оператора A

Сходимость нашей итерационной последовательности определяется спектральным радиусом матрицы G

Утверждение 20. *Процесс сходится из любой стартовой точки, если $\rho(A) < 1$*

Доказательство. Докажем сначала, что при $\rho(A) \geqslant 1$ итерационный процесс расходится, а потом докажем сходимость в обратном случае

- Пусть v — собственный вектор с собственным значением $|\lambda| \geqslant 1$. Тогда запустим два итерационных процесса из разных точек: $x^{(1)}$ и $\xi^{(1)} = x^{(1)} + v$. Рассмотрим их разность на t-ой итерации:

$$\|x^{(t)} - \xi^{(t)}\|_\infty = \|G^t(x^{(1)} - \xi^{(1)})\|_\infty = \|G^t v\|_\infty = |\lambda|^t \cdot \|v\|_\infty$$

  Видно, что при $\lambda \geqslant 1$ не могут одновременно сходиться

- Основная идея: если $\rho(G) < 1$, то можно построить такую норму $\|\cdot\|$, что $\|G\| < 1$. Тогда доказать утверждение можно аналогично методу Якоби

  Вспомним, что матрица G разбивается на блоки — жорданова нормальная форма

$$
\begin{pmatrix}
\lambda & 1 & 0 & \cdots & 0 \\
0 & \lambda & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \lambda & 1 \\
0 & 0 & 0 & 0 & \lambda
\end{pmatrix}
$$

Видно, что ограничение матрицы (оператора) G на некоторую жорданову клетку имеет вид линейной комбинации тождественного и нильпотентного оператора:

$$G|_{\text{cell}} = \lambda I + N$$

Тогда и сама матрица G в жордановом базисе имеет вид:

$$
G = \begin{pmatrix}
\lambda_1 I_1 + N_1 & & & & \\
& \lambda_2 I_1 + N_2 & & & \\
& & \ddots & & \\
& & & \ddots & \\
& & & & \lambda_n I_n + N_n
\end{pmatrix}
$$

Рассмотрим одну жорданову клетку. Для нее существует такой вектор v, что набор векторов $v, Nv, N^2v, \dots, N^{k-1}v$ — образуют базис для некоторого подпространства. Более того $N^k v = 0$, так как N — нильпотентный оператор (матрица сдвига). Если жорданова клетка имеет стандартный вид, то тогда $v = [0, \dots, 0, 1, 0, \dots, 0]^\top$ (1 стоит на k-ом месте)

Тогда возьмем следующий базис $w_i = \varepsilon^{-i} N^i v$ для некоторого $\varepsilon > 0$ и $i \in \{0, \dots, k-1\}$. Рассмотрим жорданову клетку в этом базисе:

$$(\lambda I + N)w_i = \lambda w_i + N w_i = \lambda w_i + N\varepsilon^{-i} N^i v = \lambda w_i + \varepsilon \cdot \varepsilon^{-(i+1)} N^{i+1} v = \lambda w_i + \varepsilon w_{i+1}$$

Получается, что жорданова клетка в этом базисе имеет вид:

$$
\begin{pmatrix}
\lambda & \varepsilon & 0 & \cdots & 0 \\
0 & \lambda & \varepsilon & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \lambda & \varepsilon \\
0 & 0 & 0 & 0 & \lambda
\end{pmatrix}
$$

Проделаем так с каждой жордановой клеткой матрицы G. Получаем матрицу S перехода из одного базиса в другой и матрицу G в виде:

$$
S^{-1} G S = \begin{pmatrix}
\lambda_1 I_1 + \varepsilon N_1 & & & & \\
& \lambda_2 I_1 + \varepsilon N_2 & & & \\
& & \ddots & & \\
& & & \ddots & \\
& & & & \lambda_n I_n + \varepsilon N_n
\end{pmatrix}
$$

Рассмотрим l_∞ норму в построенном новом базисе и получим, что

$$\|G\|_\infty = \max_i \sum_j |G_{ij}| = \max_i (\lambda_j + \varepsilon) = \rho(G) + \varepsilon$$

Так как $\rho(G) < 1$, то можно подобрать такое $\varepsilon > 0$, что $\|G\|_\infty < 1$. Далее доказываем аналогично методу Якоби

$\square$

3.4 Метод Гаусса-Зейделя

В отличие от прошлого метода, в методе Гаусса-Зейделя чтобы посчитать все координаты вектора $x^{(t+1)}$, можно использовать не только $x^{(t)}$, но и уже посчитанные координаты этого вектора на этой же итерации

Описание алгоритма

Пусть $x^{(t)}$ — решение на t шаге. Тогда на следующем шаге решение вычисляется следующим образом:

$$x_i^{(t+1)} = \left(b_i - \sum_{j<i} A_{ij} x_j^{(t+1)} - \sum_{j>i} A_{ij} x_j^{(t)}\right) / A_{ii}$$

Пусть матрица $A = L + U^\star$, где L — нижняя треугольная матрица, $U^\star$ — строго верхняя треугольная матрица (нулевая диагональ). Тогда уравнение можно переписать в виде:

$$Ax = y \iff Lx = b - U^\star x \implies x^{(t+1)} = L^{-1}(b - U^\star x^{(t)})$$

Или в ином виде:

$$x^{(t+1)} = (I - L^{-1}A)x^{(t)} + L^{-1}y$$

Замечание Метод Гаусса-Зейделя дает небольшой выигрыш по памяти, так как можем перезаписывать значения вектора $x^{(t)}$, и имеет может иметь чуть лучшую сходимость и точность, так как мы переиспользуем уже вычисленные значения

Сходимость метода Гаусса-Зейделя

Утверждение 21. *Метод Гаусса-Зейделя сходится для любой стартовой точки.*

Доказательство. Согласно 20 достаточно доказать, что $G = I - L^{-1}A$ такая, что $\rho(G) < 1$

Пусть x — собственный вектор с собственным значением λ матрицы G. Тогда

$$Gx = (I - L^{-1}A)x = \lambda x$$

Домножим это равенство справа на матрицу L:

$$L \cdot (I - L^{-1}A)x = (L - A)x = -Ux = \lambda Lx$$

Пусть теперь $i : |x_i| = \max_j |x_j| > 0$ и перепишем верхнее равенство в координатной форме:

$$\lambda A_{ii} x_i + \lambda \sum_{j<i} A_{ij} x_j = -\sum_{j>i} A_{ij} x_j$$

Оценим собственное значение λ по модулю, воспользовавшись обратным неравенством треугольника и поделим на x_i:

$$|\lambda| \cdot \left(A_{ii} - \sum_{j<i} |A_{ij}|\right) \leqslant \sum_{j>i} |A_{ij}|$$

Отсюда и, вспомнив, что A — матрица с диагональным преобладанием, следует, что

$$|\lambda| \leqslant \frac{\sum_{j>i} |A_{ij}|}{A_{ii} - \sum_{j<i} |A_{ij}|} < 1$$

□

3.5 Метод Релаксации. SOR

Теперь пусть матрица A — эрмитовый (самосопряженный) оператор, то есть:

$$A = A^\star,$$

где $A^\star$ — транспонированная комплексно-сопряженная матрица A

Описание алгоритма

Пусть $\alpha > 1/2$ — некоторый параметр, D — диагональ матрицы A и матрица C такая, что $C + C^\star = D - A$.

Тогда матрицы расщепления возьмем $Q = \alpha D - C$ и получаем итерационный процесс:

$$x^{(t+1)} = (I - Q^{-1}A)x^{(t)} + Q^{-1}b,$$

который сходится к решению нашей СЛУ

Сходимость метода релаксации

Утверждение 22. *Полученный итерационный процесс сходится для любой стартовой точки*

Доказательство. Согласно 20 достаточно доказать, что $G = I - Q^{-1}A$ такая, что $\rho(G) < 1$.

Пусть x — собственный вектор с собственным значением λ матрицы G. Тогда

$$Gx = (I - Q^{-1}A)x = \lambda x$$

Введем теперь вектор $y = (I - G)x = x - Gx$

Заметим, что

$$y = (I - G)x = (I - I + Q^{-1}A)x = Q^{-1}Ax \implies (\alpha D - C)y = Ax$$

А также, что

$$(Q - A)y = Ax - Ay = A(x - y) = AGx \iff (\alpha D - D + C^\star)y = AGx$$

Домножим первое и второе равенство на скалярно (эрмитово скалярное произведение) y:

$$\begin{cases} \alpha\langle Dy, y\rangle - \langle Cy, y\rangle = \langle Ax, y\rangle \\ \alpha\langle y, Dy\rangle - \langle y, Dy\rangle + \langle y, C^\star y\rangle = \langle y, AGx\rangle \end{cases}$$

Так как D — тоже эрмитова матрица, то $\langle Dy, y\rangle = \langle y, Dy\rangle$. Также верно, что $\langle Cy, y\rangle = \langle y, C^\star y\rangle$, так как $C^\star$ сопряженный оператор для C. Сложим два уравнения и получим:

$$(2\alpha - 1)\langle Dy, y\rangle = \langle Ax, y\rangle + \langle y, AGx\rangle = \langle Ax, x - Gx\rangle + \langle x - Gx, AGx\rangle = (1 - |\lambda|^2) \cdot \langle Ax, x\rangle$$

Так как $\forall x \neq 0 : \langle Ax\rangle > 0$, то случай $|\lambda| = 1$ невозможен ($y = 0$), поэтому так как слева и справа положительные числа, то $|\lambda| < 1$, что означает, что $\rho(G) < 1$ $\square$

Замечание Можно взять в качестве матрицы C строго нижнюю часть матрицы A. Тогда $C^\star$ — строго верхняя часть матрицы A

4 Методы наискорейшего спуска и сопряженных градиентов

Постановка задачи

Рассмотрим СЛУ $Ax = y$, где A — SPD матрицей, то есть

$$A^\top = A$$

и

$$\forall\, x \neq 0 : x^\top A x > 0$$

Определение 23. *Скалярным произведением* называется функция $\langle \cdot, \cdot \rangle : V \times V \longrightarrow \mathbb{R}$, где V — конечномерное векторное пространство, обладающая следующими свойствами:

- $\forall\, x, y \in V : \langle x, y \rangle = \langle y, x \rangle$
- $\forall\, \alpha, \beta \in \mathbb{R}, x, y, z \in V : \langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$
- $\forall\, x \in V : \langle x, x \rangle > 0$

Иначе говоря, это симметричная положительно определенная билинейная форма

Примеры

- Самый известный пример скалярного произведения:

$$\langle x, y \rangle = x^\top y$$

- В общем случае скалярное произведение задается матрицей Грамма G — матрица попарных скалярных произведений базисных векторов:

$$\langle x, y \rangle = \left\langle \sum_i x_i e_i, \sum_j y_j e_j \right\rangle = \sum_{i,j} x_i y_j \langle e_i, e_j \rangle = x^\top G y$$

- Любая SPD матрица задает скалярное произведение:

$$x^\top A y = \langle x, Ay \rangle = \langle Ax, y \rangle = \langle x, y \rangle_A$$

Определение 24. Рассмотрим линейное векторное пространство V и набор векторов $\{v_i\}_{i=1}^n$. Тогда *линейной оболочкой* будем называть подпространство L:

$$L = \{\lambda_1 v_1 + \ldots \lambda_n v_n \mid \lambda_i \in \mathbb{R}\} = \operatorname{span}(v_1, \ldots, v_n)$$

Определение 25. При наличии скалярного произведения можно говорить об ортогональности между вектором $x \in V$ и подпространством L: если $\forall\, v_i \in L : \langle v_i, x \rangle = 0$. Множество векторов, которые ортогональны подпространству L, называется *ортогональным* и обозначается как $L^\perp$

4.1 Метод наискорейшего спуска

Рассмотрим следующую функцию:

$$F(x) = \frac{1}{2} \langle x, x \rangle_A - b^\top x$$

Данная функция — квадратичная функция с положительно определенным гессианом, поэтому эта функция имеет единственную точку минимума. Найдем ее

- Найдем производную $F'(x)$:

$$\frac{\partial F}{\partial x_k} = \frac{1}{2}\frac{\partial}{\partial x_k}\sum_{i=1}^{n} x_i \cdot (Ax)_i - \frac{\partial}{\partial x_k}\sum_{i=1}^{n} b_i \cdot x_i =$$
$$= \frac{1}{2}\frac{\partial}{\partial x_k}\sum_{i=1}^{n} x_i \sum_{j=1}^{n} A_{ij}x_j - b_k = \frac{1}{2}\frac{\partial}{\partial x_k}\sum_{i,j<n} x_i A_{ij} x_j - b_k =$$
$$= \frac{1}{2}\sum_{i=1}^{n} A_{ik}x_i + \frac{1}{2}\sum_{i=1}^{n} A_{ki}x_i - b_k = \frac{1}{2}\sum_{i=1}^{n}(A_{ik} + A_{ki})x_i - b_k$$

  Получается, что $F'(x)$:

$$F'(x) = \frac{1}{2}(A + A^\top)x - b$$

  Вспомним, что A — симметрическая матрица, тогда $F'(x) = Ax - b$

- Получается, что задача поиска решения $Ax = b$ сводится к поиску минимума функции $F(x)$

Описание алгоритма

- Решение системы будем искать итерационно. Инициализируем стартовую точку (например, $x_1 = 0$)
- k-ую итерацию вычислим рекурсивно:

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

  где $r_k = b - Ax_k$ — вектор невязки (residual)

  Коэффициент α_k находится из минимизации функции $F(x_k + \alpha r_k)$ по α:

$$\frac{\partial}{\partial \alpha}F(x_k + \alpha r_k) = \frac{1}{2}\frac{\partial}{\partial \alpha}\langle x_k + \alpha r_k, x_k + \alpha r_k\rangle_A - \frac{\partial}{\partial \alpha}\langle b, x_k + \alpha r_k\rangle =$$
$$= \frac{1}{2}\frac{\partial}{\partial \alpha}\left(\langle x_k, x_k\rangle_A + \alpha^2\langle r_k, r_k\rangle_A + 2\alpha\langle x_k, r_k\rangle_A\right) - \langle b, r_k\rangle =$$
$$= \alpha\langle r_k, r_k\rangle_A + \langle Ax, r_k\rangle - \langle n, r_k\rangle = \alpha\langle r_k, r_k\rangle_A - \langle r_k, r_k\rangle = 0$$

  Получаем, что

$$\alpha = \frac{\langle r_k, r_k\rangle}{\langle r_k, r_k\rangle_A} = \frac{r_k^\top r_k}{r_k^\top A r_k}$$

- В итоге алгоритм имеет следующий вид:

$$\begin{cases} x_{k+1} = x_k + \alpha_k r_k \\ r_{k+1} = r_k - \alpha_k A r_k \\ \alpha_k = \dfrac{\langle r_k, r_k\rangle}{\langle r_k, r_k\rangle_A} \end{cases}$$

Сходимость алгоритма

Оценим приращение функции за одну итерацию:

$$F(x_{k+1}) - F(x_k) = \frac{1}{2}\alpha_k^2\langle r_k, r_k\rangle_A - \alpha_k\langle r_k, r_k\rangle = [\text{подставим } \alpha_k] = -\frac{1}{2}\frac{\langle r_k, r_k\rangle^2}{\langle r_k, r_k\rangle_A}$$

Предположим, что наш алгоритм не сходится, то есть $\|r_k\|^2 = \langle r_k, r_k\rangle_A > \varepsilon > 0$. Тогда изменение функции за один шаг можно оценить снизу константой, что означает, что за достаточное количество операций мы можем получить сколь угодно маленькое значение функции $F(x)$, что противоречит существованию минимума функции

4.2 Метод сопряженных градинетов

Метод наискорейшего спуска предельно прост и интуитивно понятен. К сожалению, в некоторых случаях скорость сходимости метода может быть не высока (например, когда у матрицы A большее число обусловленности), так как векторы r_k могут быть линейно зависимы. Эта проблема решается в следующем методе: мы будем запоминать направление на предыдущих шагах

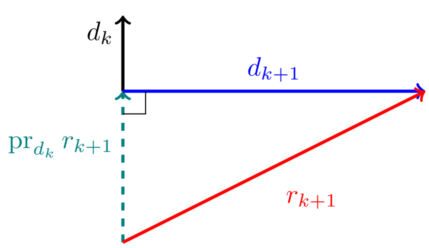
Описание метода

- Инициализация: $x_1 = 0$ (обязательно), $r_1 = b$, $d_1 = r_1$ — вектор направлений
- k-ая итерация:

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ r_{k+1} = r_k - \alpha_k A d_k \\ \alpha_k = \dfrac{\langle d_k, r_k\rangle}{\langle d_k, d_k\rangle_A} \end{cases}$$

  Коэффициент α_k находится аналогично предыдущему методу, минимизируя $F(x_k + \alpha d_k)$ по α. Нетривиально вычисляется вектор d_{k+1}. В предыдущем методы мы всегда вдоль вектора невязки, теперь же мы смещаться вдоль ортогональной проекции вектора невязки r_{k+1} на вектор направлений d_k

$$d_{k+1} = r_{k+1} - \operatorname{pr}_{d_k} r_{k+1} = r_{k+1} - \frac{\langle r_{k+1}, d_k\rangle_A}{\langle d_k, d_k\rangle_A} d_k$$



- В итоге алгоритм имеет следующий вид:

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ r_{k+1} = r_k - \alpha_k A d_k \\ \alpha_k = \dfrac{\langle d_k, r_k\rangle}{\langle d_k, r_k\rangle_A} \\ d_{k+1} = r_{k+1} - \dfrac{\langle r_{k+1}, d_k\rangle_A}{\langle d_k, d_k\rangle_A} d_k \end{cases}$$

Свойства

Утверждение 26. *Все векторы r_k ортогональны между собой:*

$$\forall\, k_1 \neq k_2 : \langle r_{k_1}, r_{k_2} \rangle = 0$$

А также все d_k A-ортогональны между собой:

$$\forall\, k_1 \neq k_2 : \langle d_{k_1}, d_{k_2} \rangle_A = 0$$

Доказательство. Будем доказывать по индукции по номеру итераций.

- База $k = 2$. Проверим, что $\langle r_1, r_2 \rangle = 0$ и $\langle d_1, d_2 \rangle_A = 0$.

$$\langle r_1, r_2 \rangle = \left\langle r_1, r_1 - \frac{\langle d_1, r_1 \rangle}{\langle d_1, d_1 \rangle_A} A d_1 \right\rangle = \langle r_1, r_1 \rangle - \frac{\langle d_1, r_1 \rangle}{\langle d_1, d_1 \rangle_A} \langle r_1, A d_1 \rangle = \langle r_1, r_1 \rangle - \frac{\langle r_1, r_1 \rangle}{\langle r_1, r_1 \rangle_A} \langle r_1, r_1 \rangle_A = 0$$

  $\langle d_1, d_2 \rangle_A = 0$ по определению.

- Предположение и шаг. Пусть вычислены $r_1, \ldots, r_k$ и $d_1, \ldots, d_k$ и выполнены условия ортогональности. Заметим, что

$$\operatorname{span}(r_1, \ldots, r_k) = \operatorname{span}(d_1, \ldots, d_k),$$

  Можно доказать это равенство по индукции используя правила вычисления r_{k+1} и d_{k+1} в процессе алгоритма:

$$\begin{aligned} r_{k+1} &= r_k - \alpha_k A d_k \\ d_{k+1} &= r_{k+1} - \frac{\langle r_{k+1}, d_k \rangle_A}{\langle d_k, d_k \rangle_A} d_k \end{aligned}$$

  1. Докажем ортогональность r_k. Вычислим r_{k+1} по определению:

$$r_{k+1} = r_k - \frac{\langle d_k, r_k \rangle}{\langle d_k, d_k \rangle_A} A d_k$$

     Поэтому вместо проверки $\langle r_{k+1}, r_i \rangle$ для $i \leqslant k$ можно проверить, что $\langle r_{k+1}, d_i \rangle$ для $i \leqslant k$. Эти утверждения равносильны. Случай $i = k$ проверяется построение. Для $i < k$:

$$\langle r_{k+1}, d_i \rangle = \langle r_k, d_i \rangle - \frac{\langle d_k, r_k \rangle}{\langle d_k, d_k \rangle_A} \langle d_i, A d_i \rangle = 0 - 0 = 0 \text{ по предположению индукции}$$

  2. Докажем теперь A-ортогональность d_k. Случай $i = k$ проверяется построением. Для $i < k$:
     По определению r_{i+1}:

$$r_{i+1} = r_i - \alpha_i A d_i \iff A d_i = \beta_i (r_{i+1} - r_i)$$

     По определению d_{k+1}:

$$d_{k+1} = r_{k+1} - \operatorname{pr}_{d_k} r_{k+1} = r_{k+1} - \frac{\langle r_{k+1}, d_k \rangle_A}{\langle d_k, d_k \rangle_A} d_k$$

     Рассмотрим $\langle d_i, d_{k+1} \rangle_A = \langle A d_i, d_{k+1} \rangle$:

$$\begin{aligned} \langle A d_i, d_{k+1} \rangle &= \beta_i \langle r_{i+1} - r_i, r_{k+1} \rangle - \beta_i \frac{\langle r_{k+1}, d_k \rangle_A}{\langle d_k, d_k \rangle_A} \langle r_{i+1} - r_i, d_k \rangle = \\ &= \beta_i \langle r_{i+1} - r_i, r_{k+1} \rangle - \frac{\langle r_{k+1}, d_k \rangle_A}{\langle d_k, d_k \rangle_A} \langle A d_i, d_k \rangle = 0 - 0 \text{ по предположению индукции} \end{aligned}$$

□

Каноническая запись

Найдем теперь $\langle d_k, r_k \rangle$, выразив d_k:

$$\langle d_k, r_k \rangle = \langle r_k - \beta_k d_{k-1}, r_k \rangle = \langle r_k, r_k \rangle - \beta_k \langle r_k, d_{k-1} \rangle = \langle r_k, r_k \rangle - 0 = \langle r_k, r_k \rangle$$

Так как линейные оболочки $\mathrm{span}(r_1, \ldots, r_k) = \mathrm{span}(d_1, \ldots, d_k)$ для всех k, то и $\langle r_k, d_{k-1} \rangle = 0$. Тогда получаем следующее:

$$r_{k+1} = r_k - \frac{\langle r_k, r_k \rangle}{\langle d_k, d_k \rangle_A} A d_k$$

Рассмотрим $\langle r_{k+1}, d_k \rangle_A = \langle r_{k+1}, A d_k \rangle$. Подставим $A d_k$ из выражения выше:

$$\langle r_{k+1}, A d_k \rangle = \left\langle r_{k+1}, \frac{\langle d_k, d_k \rangle_A}{\langle r_k, r_k \rangle} \cdot (r_k - r_{k+1}) \right\rangle = -\langle r_{k+1}, r_{k+1} \rangle \cdot \frac{\langle d_k, d_k \rangle_A}{\langle r_k, r_k \rangle}$$

Получаем, что

$$d_{k+1} = r_{k+1} + \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} d_k$$

Итоговый алгоритм на k-ой итерации:

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ r_{k+1} = r_k - \alpha_k A d_k \\ \alpha_k = \dfrac{\langle r_k, r_k \rangle}{\langle d_k, d_k \rangle_A} \\ \beta_k = \dfrac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} \\ d_{k+1} = r_{k+1} + \beta_k d_k \end{cases}$$

5 Метод бисопряженных градиентов

Напоминание

На прошлой лекции рассматривалась следующая задача:

$$Ax = b,$$

где $A_{n\times n}$ — SPD матрица. Для таких матриц и был придуман метод сопряженных градиентов, который сходится за n шагов в точной арифметике

- Инициализация: $x_1 = 0$ (обязательно), $r_1 = b$, $d_1 = r_1$ — вектор направлений
- k-ая итерация:

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ r_{k+1} = r_k - \alpha_k A d_k \\ \alpha_k = \dfrac{\langle d_k, r_k\rangle}{\langle d_k, d_k\rangle_A} \end{cases}$$

Одной из главных особенностей метода сопряженных градиентов является ортогональность векторов невязок r_k и направлений d_k — ключевое свойство для доказательства сходимости за n шагов. Можно ли построить такой же алгоритм для несимметричной матрицы A?

Двойственность

Определение 27. Пусть V — линейное пространство над полем $\mathbb{F}$. Тогда двойственным (или сопряженным) к нему пространством назовем:

$$V^\star = \{f : V \longrightarrow \mathbb{F} : f \text{ — линейное над } \mathbb{F}\}$$

Скалярное произведение на линейном пространстве V позволяет отождествить само пространство V со множеством линейных функций на V:

$$f(x) = f\left(\sum_i x_i e_i\right) = \sum_i x_i f(e_i) = \sum_i x_i f_i = \langle x, f\rangle,$$

то есть в ортонормированном базисе линейной функции со значениями f_i на базисных векторах e_i ставится в соответствие вектор $[f_1, \ldots, f_n]^\top$

Но если на пространстве V не задано скалярное произведение, то и соответствия между V и пространством его линейных функций можно построить, но будет зависеть от выбора базиса

По этой причине в методе бисопряженных градиентов строятся 4 последовательности векторов: 2 последовательности векторов и 2 последовательности связанных с ними функций

Двойственность и линейные операторы

Пусть $A : V \longrightarrow V$ — линейный оператор, $f : V \longrightarrow \mathbb{R}$ — линейная функция. Возьмем произвольный $x \in V$:

$$f^\top Ax = f(Ax) = \sum_i f_i \sum_j A_{ij} x_j = \sum_j \left(\sum_i A_{ij} f_i\right) x_j = (A^\star f)(x) = (A^\top f)^\top x$$

То есть по A мы можем построить линейный оператор $A^\star : V^\star \longrightarrow V^\star : A^\star = A^\top$

5.1 Алгоритм

- Инициализация x_1. По начальному приближению строится вектор невязки $r_1 = b - Ax_1$.
  Затем задается линейная функция $\hat{r}_1 = \hat{r}_1(r_1) = \hat{r}_1^\top r_1 \neq 0$. Наиболее популярный вариант: $\hat{r}_1 = r_1$
  Векторы направлений строятся как $p_1 = r_1, \hat{p}_1 = \hat{r}_1$
- $k+1$-ая итерация:

$$
\begin{cases}
x_{k+1} = x_k + \alpha_k p_k \\
r_{k+1} = r_k - \alpha_k A p_k \\
\hat{r}_{k+1} = \hat{r}_k - \alpha_k A^\top \hat{p}_k \\
p_{k+1} = r_{k+1} + \beta_k p_k \\
\hat{p}_{k+1} = \hat{r}_{k+1} + \beta_k \hat{p}_k \\
\alpha_k = \dfrac{\hat{r}_k r_k}{\hat{p}_k A p_k} \\
\beta_k = \dfrac{\hat{r}_{k+1} r_{k+1}}{\hat{r}_k r_k}
\end{cases}
$$

5.2 Доказательство алгоритма

Ортогональность

Утверждение 28. *Для любых $i \neq j$ выполняются следующие условия:*

$$
\begin{cases}
\langle \hat{r}_i, r_j \rangle = \hat{r}_i^\top r_j = 0 \\
\langle \hat{p}_i, A p_j \rangle = \hat{p}_i^\top A p_j = 0
\end{cases}
$$

Доказательство. Доказываем по индукции $k = \max(i, j)$

- Заметим, что (доказывается по индукции)

$$
\operatorname{span}(r_1, \dots, r_k) = \operatorname{span}(p_1, \dots, p_k)
$$
$$
\operatorname{span}(\hat{r}_1, \dots, \hat{r}_k) = \operatorname{span}(\hat{p}_1, \dots, \hat{p}_k)
$$

- Пусть $j = k+1, i < k$:

$$
\hat{r}_i^\top r_{k+1} = \hat{r}_i^\top r_k - \alpha_k \hat{r}_i^\top A p_k = 0 - 0 = 0
$$
$$
\hat{p}_i^\top A p_{k+1} = \hat{p}_i^\top \cdot \frac{r_k - r_{k+1}}{\alpha_k} = \frac{1}{\alpha_k} \hat{p}_i^\top r_k - \frac{1}{\alpha_k} \hat{p}_i^\top r_{k+1} = 0 - 0 = 0
$$

- Для $i = k$ ортогональность выполняется из-за выбора коэффициентов α_k и β_k:

$$
\hat{r}_k^\top r_{k+1} = \hat{r}_k^\top r_k - \alpha_k \hat{r}_k^\top A p_k = \hat{r}_k^\top r_k - \alpha_k \hat{p}_k^\top A p_k + \beta_{k-1} \alpha_k \hat{p}_{k-1} A p_k = 0 \,(\text{подставим } \alpha_k)
$$

  Аналогично для векторов направлений:

$$
\hat{p}_k A p_{k+1} = (A^\top \hat{p}_k)^\top p_{k+1} = (A^\top \hat{p}_k)^\top r_{k+1} + \beta_k (A^\top \hat{p}_k)^\top p_k = -\frac{1}{\alpha_k} \hat{r}_{k+1}^\top r_{k+1} + \beta_k (A^\top \hat{p}_k)^\top p_k = 0
$$

□

5.3 Проблемы алгоритма

6 Предобуславливание

7 Собственные векторы и собственные значения

Определение 29. *Собственным вектором* v матрицы $A_{n\times n}$ с соответствующим *собственным значением* λ называется вектор, для которого выполнено:

$$Av = \lambda v$$

Постановка задачи: SPD матрица

Задача нахождения (всех или нескольких) собственных векторов и собственных значений часто возникает на практике, особенно случай, когда A_n является SPD матрицей, который будет рассматриваться сейчас

Для SPD матрицы известно, что существует ортонормированный базис из собственных векторов $e_1, \dots, e_n$ с собственными значениями $\lambda_1, \dots, \lambda_n$, где $\lambda_i > 0$

7.1 Степенной метод

Описание метода

Следующий метод позволяет найти собственный вектор и соответствующее наибольшее собственное значение матрицы A

- Сгенерируем случайный вектор $v_0 : v^\top v = 1$
- Посчитаем $x_1 = Av_0$ и отнормируем его
- Проделаем то же самое с вектором $v_0 = x_1$
- При достаточном количество итераций получим v_0 — собственный вектор с наибольшим собственным значением:

$$\lambda = \frac{v_1^\top v_0}{v_0^\top v_0}$$

Сходимость метода

Введем $\|\cdot\|_2$ норму. Пусть у матрицы A есть базис из ортонормированных собственных векторов $e_1, \dots, e_n$ и соответствующими собственными значениями $\lambda_1 > \lambda_2 \geqslant \dots \geqslant \lambda_n$. Пусть $v_0 = \sum x_i e_i$ — некоторый вектор единичной длины с $x_1 \neq 1$. Определим последовательность

$$v_{k+1} = \frac{Av_k}{\|Av_k\|}$$

$$\|Av_k\| \cdot v_{k+1} = Av_k = A^k v_0 = \sum_{i=1}^n x_i A^k e_i = \sum_{i=1}^n x_i \lambda_i^k e_i = \lambda_1^k \left(x_1 e_1 + \sum_{i=2}^n x_i \cdot \left(\frac{\lambda_i}{\lambda_1}\right)^k e_i \right)$$

Вспомним, что $\lambda_i/\lambda_1 < 1$, поэтому при достаточно больших k:

$$\|Av_k\| \cdot v_{k+1} = Av_k = \lambda_1^k (x_1 e_1 + \varepsilon_k),$$

где $\|\varepsilon_k\| \longrightarrow 0$. Рассмотрим при $k \longrightarrow \infty$

$$\|v_{k+1} - e_1\| = \left\| \frac{Av_k}{\|Av_k\|} - e_1 \right\| = \frac{1}{\|Av_k\|} \cdot \left\| \lambda_1^k (x_1 e_1 + \varepsilon_k) - \|Av_k\| \cdot e_1 \right\| \leqslant \frac{1}{\|Av_k\|} \cdot \left| \lambda_1^k x_1 - \|Av_k\| \right| \cdot \|e_1\| \longrightarrow 0$$

Теперь последовательность:

$$r_{k+1} = \frac{\langle v_{k+1}, v_k \rangle}{\langle v_k, v_k \rangle} = \frac{v_k^\top A v_k}{v_k^\top v_k} \longrightarrow \lambda \cdot \frac{v_k^\top v_k}{v_k^\top v_k} = \lambda$$

Как искать остальные?

Зафиксируем сколь угодно малое $\varepsilon > 0$. Пусть мы нашли собственный вектор v_1 с наибольшим собственным значением λ_1. Построим матрицу $B = A - (\lambda_1 - \varepsilon)v_1 v_1^\top$. Так как A является SPD матрицей, то

$$A = UDU^\top,$$

где U — ортогональная матрица, столбцы которой являются ортонормированными собственными векторами, D — диагональная матрица из собственных значений $\lambda_i > \varepsilon$. Поэтому $A = \sum \lambda_i v_i v_i^\top$. Тогда

$$B = A - (\lambda_1 - \varepsilon)v_1 v_1^\top = \varepsilon v_1 v_1^\top + \sum_{i=2}^{n} \lambda_i v_i v_i^\top$$

Тогда если применить степенной метод для матрицы B, то мы можем получить собственный вектор со следующим по величине собственным значением

Замечание Чтобы найти наименьшее собственное значение и соответствующий собственный вектор, то можем применить алгоритм к матрице A^{-1}. Чтобы вычислить $A^{-1}v_k$, мы можем вместо поиска A^{-1}, решать СЛУ $Az_k = v_k \iff z_k = A^{-1}v_k$

Поиск конкретного собственного значения

Пусть у матрицы A есть собственные значения $\lambda_1, \ldots, \lambda_n$ и мы хотим найти λ_i. Пусть мы знаем некоторое приближение $\lambda_i^\star$. Определим тогда матрицу $B = A - \lambda_i^\star I$ и рассмотрим ее спектр

$$\operatorname{spec}(B) = \{\lambda_k - \lambda_i^\star\} \implies \operatorname{spec}(B^{-1}) = \left\{\frac{1}{\lambda_k - \lambda_i^\star}\right\}$$

Получается, что если мы подобрали хорошее начальное приближение $\lambda_i^\star$, то $1/(\lambda_i - \lambda_i^\star)$ — будем наибольшим собственным значением матрицы B^{-1}, которое мы можем найти с помощью степенного метода

7.2 QR-разложение

Следующий метод позволяет найти сразу несколько m наибольших собственных значений и собственных векторов

Введем $\|\cdot\|_2$ норму. В этом методе нам понадобится процесс ортогонализации Грамма-Шмидта. Пусть у нас есть векторы $v_1, \ldots, v_n$ и мы хотим получить ортонормированный базис $e_1, \ldots e_n$. Процесс описывается следующим образом:

- Пусть $u_1 = v_1$. Теперь будем вычислять векторы $u_2, \ldots, u_n$ следующим образом:

$$u_k = v_k - \sum_{j=1}^{k-1} \operatorname{pr}_{u_j}(v_k),$$

  где $\operatorname{pr}_{u_j}(v_k)$ — проекция вектора v_k на u_j

- Теперь $e_i = u_i / \|u_i\|$

Описание метода

Пусть $V_{n\times m} = V_0 = (v_1 | \ldots | v_m)$ — матрица, чьи столбцы v_i являются ортонормированными векторами. Тогда будем итеративно вычислять матрицу:

$$V_{k+1} = \operatorname{ORT}(AV_k),$$

где ORT — процесс ортогонализации Грамма-Шмидта

При достаточно большом k мы получим, что столбцы матрицы V_k являются собственными векторами $e_1, \ldots, e_m$ с собственными значениями $\lambda_1 \geqslant \ldots \geqslant \lambda_m$

Сходимость метода

Сходимость этого метода доказывается аналогично прошлому методу

Замечание Аналогично прошлому методу мы можем найти первые m наименьших собственных значений, применив метод к матрице A^{-1}