

Стандартные матричные разложения: LDL, LU и QR

Скачков Николай Андреевич

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

21 ноября 2017 г.

Задачи с точки зрения линейной алгебры:

- 1 Быстрое нахождение обратной матрицы
- 2 Нахождение собственных значений и собственных векторов
- 3 Эффективное приближение матрицы другой, обладающей нужными свойствами

Задачи с точки зрения машинного обучения:

- 1 Ускорение обучения
- 2 Увеличение устойчивости алгоритмов обучения
- 3 Уменьшение размерности пространства признаков

Ищем разложение: $\mathbf{A} = \mathbf{LU}$, где L — нижняя треугольная матрица, U — верхняя треугольная.

Применение:

- Решение СЛАУ $Ax = b$, для различных правых частей:
 - 1 Решается система: $Ly = b$
 - 2 Её решение подставляется в следующую: $Ux = y$
- Нахождение обратной матрицы. Решается задача: $AX = I$

Данное разложение существует и единственно \Leftrightarrow главные миноры матрицы A отличны от нуля.

Данное условие накладывает большие ограничения на использование данного разложения.

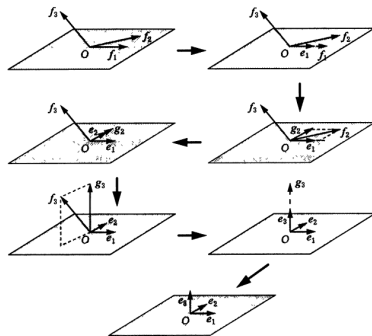
QR-разложение

Ищем разложение: $\mathbf{A} = \mathbf{QR}$, где Q — ортогональная матрица, R — верхняя треугольная.

- Разложение существует и единственно $\Leftrightarrow A$ невырождена.
- Разложение существует для любой квадратной матрицы. И его можно определить для прямоугольной матрицы $n \times m$, где $n > m$.

Разложение находится в процессе алгоритма Грамма-Шмидта:

$$b_i = a_i - \sum_{j < i} \text{proj}_{b_j} a_i$$



QR-алгоритм для симметричных матриц

Для симметричной матрицы A определен QR-алгоритм:

$$A_k = Q_k R_k, \quad A_0 = A,$$

$$A_{k+1} := R_k Q_k = Q_k^T Q_k R_k Q_k = Q_k^T A_k Q_k$$

Как видно, данный алгоритм сохраняет собственные значения матрицы A на каждой итерации.

Можно показать, что матрица A_n сходится к верхней треугольной, с собственными значениями на диагонали, когда $|\lambda_1| > \dots > |\lambda_n|$

Использование:

- + Эффективная реализация метода главных компонент, наряду с SVD
- Необходимость преобразования матрицы к трехдиагональному виду для ускорения алгоритма
- Неустойчивость при работе с плохо обусловленными матрицами
- Неустойчивость при наличии кратных собственных значений

Ищем разложение: $\mathbf{A} = \mathbf{LDL}^T$, где L — верхняя треугольная матрица с единичной диагональю, D — диагональная.

LDL-разложение — обобщение разложения Холецкого на случай произвольной симметричной матрицы.

Свойство:

A — положительноопределенная матрица $\Leftrightarrow d_i > 0$.

Вычисление:

$$D_j = A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2 D_k, \quad L_{ij} = \frac{1}{D_j} \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_k \right)$$

Возможно блочное вычисление матриц разложения.

Задача многомерной оптимизации: $f(x) \rightarrow \min_x$. Решаем итеративно:

$$f(x_k + d_k) \approx f(x_k) + \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k \rightarrow \min_{d_k}$$

$$\nabla_{d_k} = 0 \iff \nabla_{d_k} f(x_k) + \nabla_{d_k}^2 f(x_k) d_k = 0$$

Решение находим в виде:

$$d_k = - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) := -H_k^{-1} g_k$$

Преимущества и недостатки:

- + Значительно быстрее градиентного спуска
- Сходится устойчиво только если матрица H_k — положительноопределенная

Метод Ньютона

Комбинируем с идеей градиентного спуска:

$$x_{k+1} = x_k - \alpha_k H_k^{-1} g_k$$

Решение проблемы положительной определенности: пусть $H_k = LDL^T$, тогда:

$$\hat{d}_i = \begin{cases} d_i, & d_i > \varepsilon, \\ \varepsilon, & d_i \leq \varepsilon \end{cases}$$

Определим $\hat{H}_k := L\hat{D}L^T$, и получим:

$$x_{k+1} = x_k - \alpha_k L_k^{-T} \hat{D}_k^{-1} L_k^{-1} g_k$$

Итог:

- Данный метод сходится устойчиво
- Алгоритм сходится быстрее метода градиентного спуска