Разложение Шура. Форма Гессенберга. SVD

Даня Меркулов

МФТИ. AI360

Разложение Шура

Разложение Шура

Для произвольной квадратной матрицы $A \in \mathbb{R}^{n \times n}$ существует разложение:

$$A = UTU^*,$$

где U - унитарная матрица, T - верхняя треугольная матрица с собственными числами матрицы A на главной диагонали.

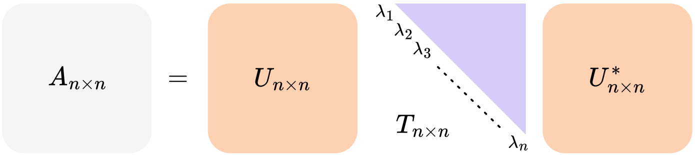


Рис. 1: Разложение Шура матрицы A

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:
    - Вычисляем QR-разложение: $A_k = Q_k R_k$

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \dots$:
   - Вычисляем QR-разложение: $A_k = Q_k R_k$
   - Формируем следующую итерацию: $A_{k+1} = R_k Q_k$

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:
    - Вычисляем QR-разложение: $A_k = Q_k R_k$
    - Формируем следующую итерацию: $A_{k+1} = R_k Q_k$

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:
    - Вычисляем QR-разложение: $A_k = Q_k R_k$
    - Формируем следующую итерацию: $A_{k+1} = R_k Q_k$

Для симметричных матриц этот процесс сходится к диагональной матрице, содержащей собственные числа. Для произвольных матриц он сходится к верхнетреугольной матрице, где диагональные элементы являются собственными числами исходной матрицы.

$$A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^{\top} A_k Q_k,$$

- Алгоритм сходится последовательно от старших собственных чисел к младшим, 2-3 итерации на одно собственное число.

QR-алгоритм

QR-алгоритм выглядит следующим образом:

1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:
   - Вычисляем QR-разложение: $A_k = Q_k R_k$
   - Формируем следующую итерацию: $A_{k+1} = R_k Q_k$

Для симметричных матриц этот процесс сходится к диагональной матрице, содержащей собственные числа. Для произвольных матриц он сходится к верхнетреугольной матрице, где диагональные элементы являются собственными числами исходной матрицы.

$$
A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^\top A_k Q_k,
$$

- Алгоритм сходится последовательно от старших собственных чисел к младшим, 2-3 итерации на одно собственное число.
- 1 итерация = подсчёт QR-разложения $\mathcal{O}(n^3)$ + умножение матриц $\mathcal{O}(n^3)$. Наивная сложность QR-алгоритма составляет $\mathcal{O}(n^4)$.

QR-алгоритм

QR-алгоритм выглядит следующим образом:

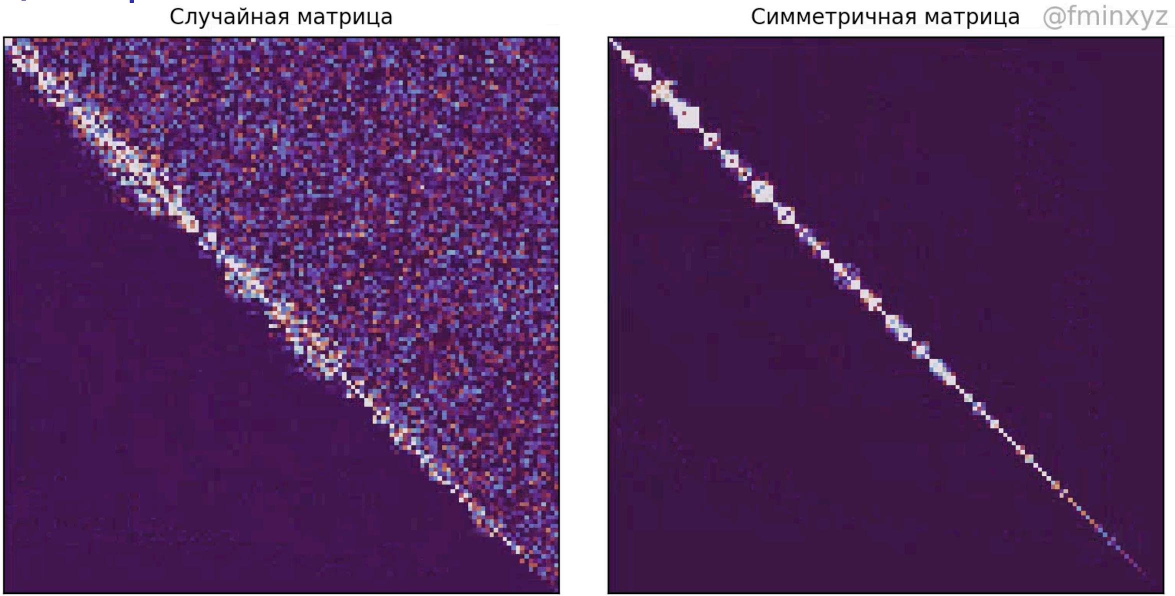
1. Начинаем с матрицы $A_0 = A$
2. Для $k = 0, 1, 2, \ldots$:
   - Вычисляем QR-разложение: $A_k = Q_k R_k$
   - Формируем следующую итерацию: $A_{k+1} = R_k Q_k$

Для симметричных матриц этот процесс сходится к диагональной матрице, содержащей собственные числа. Для произвольных матриц он сходится к верхнетреугольной матрице, где диагональные элементы являются собственными числами исходной матрицы.

$$
A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^\top A_k Q_k,
$$

- Алгоритм сходится последовательно от старших собственных чисел к младшим, 2-3 итерации на одно собственное число.
- 1 итерация $=$ подсчёт QR-разложения $\mathcal{O}(n^3)$ $+$ умножение матриц $\mathcal{O}(n^3)$. Наивная сложность QR-алгоритма составляет $\mathcal{O}(n^4)$.
- На практике используются различные стратегии ускорения сходимости до $\mathcal{O}(n^3)$. Например, приведение матрицы к форме Гессенберга ($\mathcal{O}(n^3)$), QR разложение которой строится за $\mathcal{O}(n^2)$ итераций. Кроме того, используются сдвиги, которые позволяют ускорить сходимость.

QR-алгоритм



Форма Гессенберга

Матрица A представлена в форме Гессенберга, если

$$a_{ij} = 0, \quad \text{если } i \geq j + 2.$$

Матрица в форме Гессенберга имеет вид:

$$H = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}.$$

Приведение матрицы к форме Гессенберга

- Применение отражений Хаусхолдера позволяет привести любую матрицу к форме Гессенберга

$$U^* A U = H$$

Приведение матрицы к форме Гессенберга

- Применение отражений Хаусхолдера позволяет привести любую матрицу к форме Гессенберга

$$U^* A U = H$$

- Единственное отличие от разложения Шура состоит в том, что мы должны преобразовать первый столбец в вектор с двумя ненулевыми элементами, и первый элемент не изменится.

Приведение матрицы к форме Гессенберга

- Применение отражений Хаусхолдера позволяет привести любую матрицу к форме Гессенберга

$$U^* A U = H$$

- Единственное отличие от разложения Шура состоит в том, что мы должны преобразовать первый столбец в вектор с двумя ненулевыми элементами, и первый элемент не изменится.
- Стоимость такого приведения составляет $\mathcal{O}(n^3)$ операций.

Приведение матрицы к форме Гессенберга

- Применение отражений Хаусхолдера позволяет привести любую матрицу к форме Гессенберга

$$U^* A U = H$$

- Единственное отличие от разложения Шура состоит в том, что мы должны преобразовать первый столбец в вектор с двумя ненулевыми элементами, и первый элемент не изменится.
- Стоимость такого приведения составляет $\mathcal{O}(n^3)$ операций.
- Вычисление одной итерации QR-алгоритма в форме Гессенберга требует $\mathcal{O}(n^2)$ операций (например, используя вращения Гивенса, как?), и форма Гессенберга сохраняется при QR-итерации (проверьте почему).

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Решение:

1. В матрице в форме Гессенберга в каждом столбце есть только один ненулевой элемент под диагональю.

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Решение:

1. В матрице в форме Гессенберга в каждом столбце есть только один ненулевой элемент под диагональю.
2. Для обнуления этого элемента достаточно одного вращения Гивенса, которое затрагивает только две строки матрицы.

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Решение:

1. В матрице в форме Гессенберга в каждом столбце есть только один ненулевой элемент под диагональю.
2. Для обнуления этого элемента достаточно одного вращения Гивенса, которое затрагивает только две строки матрицы.
3. Применение вращения Гивенса к двум строкам матрицы требует $\mathcal{O}(n)$ операций.

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Решение:

1. В матрице в форме Гессенберга в каждом столбце есть только один ненулевой элемент под диагональю.
2. Для обнуления этого элемента достаточно одного вращения Гивенса, которое затрагивает только две строки матрицы.
3. Применение вращения Гивенса к двум строкам матрицы требует $\mathcal{O}(n)$ операций.
4. Всего нужно применить $n-1$ вращений (по одному для каждого столбца, кроме последнего).

Упражнение

Почему вычисление QR-разложения матрицы в форме Гессенберга требует $\mathcal{O}(n^2)$ операций, а не $\mathcal{O}(n^3)$?

Решение:

1. В матрице в форме Гессенберга в каждом столбце есть только один ненулевой элемент под диагональю.
2. Для обнуления этого элемента достаточно одного вращения Гивенса, которое затрагивает только две строки матрицы.
3. Применение вращения Гивенса к двум строкам матрицы требует $\mathcal{O}(n)$ операций.
4. Всего нужно применить $n-1$ вращений (по одному для каждого столбца, кроме последнего).
5. Следовательно, общая сложность составляет $\mathcal{O}(n) \cdot (n-1) = \mathcal{O}(n^2)$ операций. Это значительно быстрее, чем $\mathcal{O}(n^3)$ операций для произвольной матрицы, где в каждом столбце может быть до $n-1$ ненулевых элементов под диагональю.

Упражнение

Объясните, почему при применении QR-алгоритма к матрице в форме Гессенберга сохраняется форма Гессенберга?

Упражнение

Таким образом, QR-алгоритм является эффективным практическим инструментом для вычисления собственных значений матрицы. Практические трюки, которые используются для ускорения сходимости QR-алгоритма, основаны на приведении матрицы к форме Гессенберга, а так же использовании сдвигов так, чтобы собственные значения матрицы A были как можно более отделены друг от друга.

Выполните упражнение на QR-алгоритм. Code.

SVD

Сингулярные значения и собственные числа

Для произвольной матрицы A существует сингулярное разложение (SVD):

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

где U и V - унитарные матрицы, Σ - диагональная матрица с сингулярными значениями.

Это разложение можно рассматривать как приведение матрицы к диагональному виду с помощью двусторонних унитарных преобразований:

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

С помощью двусторонних преобразований Хаусхолдера любую матрицу можно привести к **бидиагональной форме** B.

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:

$$T = B^* B$$

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:
$$T = B^*B$$
2. Работа с расширенной матрицей:
$$T = \begin{bmatrix} 0 & B \\ B^* & 0 \end{bmatrix}$$

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:
$$T = B^* B$$
2. Работа с расширенной матрицей:
$$T = \begin{bmatrix} 0 & B \\ B^* & 0 \end{bmatrix}$$
3. Важно отметить, что первый способ применим только если матрица T не формируется явно!

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:
$$T = B^* B$$
2. Работа с расширенной матрицей:
$$T = \begin{bmatrix} 0 & B \\ B^* & 0 \end{bmatrix}$$
3. Важно отметить, что первый способ применим только если матрица T не формируется явно!
4. Таким образом, задача вычисления сингулярных значений сводится к задаче вычисления собственных значений симметричной трехдиагональной матрицы.

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:
$$T = B^* B$$
2. Работа с расширенной матрицей:
$$T = \begin{bmatrix} 0 & B \\ B^* & 0 \end{bmatrix}$$
3. Важно отметить, что первый способ применим только если матрица T не формируется явно!
4. Таким образом, задача вычисления сингулярных значений сводится к задаче вычисления собственных значений симметричной трехдиагональной матрицы.

Вычисление сингулярных значений

Явный QR-алгоритм (со сдвигами) позволяет вычислять собственные значения и форму Шура. Однако его нельзя напрямую применить к бидиагональной матрице, так как она в общем случае не диагонализуема.

Задача вычисления сингулярных значений может быть сведена к задаче вычисления собственных значений симметричной матрицы двумя способами:

1. Работа с трехдиагональной матрицей:
$$T = B^* B$$
2. Работа с расширенной матрицей:
$$T = \begin{bmatrix} 0 & B \\ B^* & 0 \end{bmatrix}$$
3. Важно отметить, что первый способ применим только если матрица T не формируется явно!
4. Таким образом, задача вычисления сингулярных значений сводится к задаче вычисления собственных значений симметричной трехдиагональной матрицы.

Выполните упражнение Disentangled Representations на вычисление сингулярных значений. Code.