

1. Введение

В данной статье представлена реализация алгоритма обновления SVD с использованием ортонормированных μ -вращений. Ортонормированный μ -поворот - это поворот на угол из заданного набора углов μ -поворота (например, углов $\Phi_i = \arctan 2^{-i}$), которые выбраны таким образом, что вращение может быть реализовано с помощью небольшого количества операций добавления сдвига. Используется версия, в которой все вычисления полностью основаны на оценке и применении нормальных вращений. Для всех приближений используется одинаковая точность. Оценка поворота также может быть выполнена путем выполнения μ -поворотов.

2.

Ортонормированное вращение

В этом разделе представлены разложения матрицы (QRD, SVD), необходимые для алгоритма обновления SVD, определены и рассмотрены их вычисления в соответствии с алгоритмом обновления SVD (QRD-обновление и вычисление SVD с использованием алгоритма Когбетлянца).

Вращение в ортонормированной плоскости (заданное вращение) $G_{pq}(\Phi) \in R^{n \times n}$ определяется углом поворота Φ и плоскостью (p, q) , в которой происходит вращение, т.е. соотношением $\cos \Phi$ и $\sin \Phi$ в (pp, pq, qp, qq) позиции единичной матрицы $n \times n$. $G_{pq}(\Phi)$ - это ортонормированное вращение, поскольку $G_{pq}(\Phi)^T G_{pq}(\Phi) = I$. Не теряя общности, мы лишь подробно рассмотрим оценку и применение ортонормальных вращений 2×2 .

$$G(\Phi) = \begin{bmatrix} \cos \Phi & \sin \Phi \\ -\sin \Phi & \cos \Phi \end{bmatrix} = \frac{1}{\sqrt{1 + (\tan \Phi)^2}} \begin{bmatrix} 1 & \tan \Phi \\ \tan \Phi & 1 \end{bmatrix} \quad (1)$$

в следующем.

CORDIC

Процедура CORDIC использует следующее представление для угла поворота Φ :

$$\Phi = \sum_{i=0}^w \tau_i \Phi_i, \quad (2)$$

где $\Phi_i = \arctan 2^{-i}$ образуют базовые углы, а методы $\tau_i \in \{+1, -1\}$ являются цифрами представления.

Следовательно, $\tan \Phi_i = 2^{-i}$ выполняется для базисных углов таким образом, что с помощью (1) получается вращение CORDIC

$$G(\Phi) = \frac{1}{K_w} \prod_{i=0}^w \begin{bmatrix} 1 & \tau_i 2^{-k} \\ -\tau_i 2^{-k} & 1 \end{bmatrix} \quad (3)$$

где коэффициент масштабирования - не зависит от угла поворота тени:

$$\frac{1}{K_w} = \prod_{i=0}^w \frac{1}{\sqrt{1 + 2^{-2i}}} \quad (4)$$

Предпринимались различные попытки устранить масштабирующий фактор или, по крайней мере, привести его к простому двоичному представлению. Делорм предложил метод вычисления переменного коэффициента масштабирования в режиме реального времени. Этот случай возникает для переменной итерации, связанной в (3). Вместо работы с базисными углами Φ_i базисные углы получаются путем двукратного выполнения поворота на Φ_{i+1} . Относительный двойной поворот $G_d(\bar{\Phi})$ задается

$$G_d(\bar{\Phi}) = \frac{1}{K_w^2} \prod_{i=1}^{w+1} \begin{bmatrix} 1 - 2^{-2i} & \tau_i 2^{-i+1} \\ -\tau_i 2^{-i+1} & 1 - 2^{-2i} \end{bmatrix} \quad (5)$$

Базовые углы двойного поворота задаются соотношением

$$\bar{\Phi}_i = \arctan \frac{2^{-i+1}}{1 - 2^{-2i}} = 2\Phi_{i+1}$$

Теперь для каждого шага рекурсии требуется четыре (вместо двух) операции добавления сдвига, но коэффициент масштабирования не содержит квадратного корня. Чтобы избежать деления в коэффициенте масштабирования, можно использовать следующее простое тождество:

$$\frac{1}{1 + 2^{-2k}} = (1 - 2^{-2k})(1 + 2^{-4k})(1 + 2^{-8k}) \dots \quad (6)$$

QR-декомпозиция

QR-декомпозиция матрицы $X \in R^{m \times n} (m \geq n)$ определяется

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad (7)$$

где $Q \in R^{m \times m}$ ортонормировано ($Q^T Q = I$), а $R \in R^{n \times n}$ является верхним треугольником.

QRD подзадача 2×1

Вектор $[x, y]^T$ поворачивается на угол Φ с помощью

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = G(\Phi) \begin{bmatrix} x \\ y \end{bmatrix} \quad (8)$$

Вычисление Φ таким образом, что $y' = 0$ решает подзадачу 2×1 QRD подзадачу, т.е. вычисляет

$$\Phi = \arctan \frac{y}{x} \quad (9)$$

Вычисление QR-кода

Треугольная матрица R получается путем решения последовательности подзадач 2×1 QRD, т.е. применения последовательности ортонормированных вращений $G_{pq}(\Phi)$ к матрицам X (исходный X перезаписан), где $G_{pq}(\Phi)$ аннулирует мгновенный элемент x_{pq} для $1 < q \leq n$ и $q + 1 < p \leq m$, т.е.,

$$X \leftarrow G_{pq}(\Phi) \cdot X \quad (10)$$

где $\Phi = \arctan \frac{x_{qp}}{x_{pp}}$, такой, что $Q = \prod_{p,q} G_{pq}(\Phi)$ и X перезаписываются R .

QRD-Обновление

Альтернативой триангуляции X по столбцам является выполнение триангуляции строка за строкой. Это приводит к рекурсивному обновлению QRD. $X_{[k-1]}$ - матрица данных $k - 1 \times n$, доступная на временном шаге $k - 1$, а $x_{[k]}^T$ - новый вектор данных, измеренный на временном шаге k , который получается из

$$X_{[k]} = \begin{bmatrix} \lambda X_{[k-1]} \\ x_{[k]}^T \end{bmatrix} \quad (11)$$

где λ - фактор забывания.

Учитывая QRD $X_{[k-1]}$

$$X_{[k-1]} = Q_{[k-1]} \begin{bmatrix} R_{[k-1]} \\ 0 \end{bmatrix} \quad (12)$$

верхний треугольный коэффициент $R_{[k]}$ получен путем добавления нового вектора данных $x_{[k]}^T$ к взвешенной матрице $\lambda R_{[k-1]}$ и использования последовательности поворотов Гивенса $G_{pq}(\Phi)$ ($p = k$; $1 \leq q \leq n$) для уничтожения присоединенной строки, т.е.,

$$\begin{bmatrix} R_{[k]} \\ 0^T \end{bmatrix} \leftarrow \prod_{q=1}^n G_{kq}(\Phi) \begin{bmatrix} \lambda R_{[k-1]} \\ x_{[k]}^T \end{bmatrix} \quad (13)$$

Разложение по сингулярным значениям

SVD матрицы $X \in R^{m \times n}$ определяется

$$X = U \Sigma V^T \quad (14)$$

где $U \in R^{m \times m}$ и $V \in R^{n \times n}$ - ортонормированные матрицы ($U^T U = I, V^T V = I$), а $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ диагональная матрица $m \times n$, содержащая сингулярные значения.

Подзадача SVD 2×2

Дана матрица 2×2 $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ применяются вращения $G(\Phi^U)$ и $G(\Phi^V)$ слева и справа от A:

$$\begin{bmatrix} a_{11}' & a_{12}' \\ a_{21}' & a_{22}' \end{bmatrix} = G(\Phi^U)^T \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} G(\Phi^V) \quad (15)$$

Вычисление Φ^U и Φ^V таким образом, что $a_{12}' = 0$ и $a_{21}' = 0$ выполняется, решается подзадача SVD 2×2 . Углы Φ^U и Φ^V могут определяться по двум углам Φ^R и Φ^S которые могут быть вычисляются независимо путем решения двух подзадач 2×1 QRD. С помощью

$$\begin{aligned} x_1 &= (a_{11} + a_{22})/2 & x_2 &= (a_{22} - a_{11})/2 \\ y_1 &= (a_{21} - a_{12})/2 & y_2 &= (a_{21} + a_{12})/2 \end{aligned} \quad (16)$$

мы определяем два поворота $G(\Phi^R)$ и $G(\Phi^S)$:

$$\begin{bmatrix} x_1' \\ y_1' \end{bmatrix} = G(\Phi^R) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} x_2' \\ y_2' \end{bmatrix} = G(\Phi^S) \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (18)$$

это дает $y_1' = 0$ ($\Phi^R = \arctan \frac{y_1}{x_1}$) и $y_2' = 0$ ($\Phi^S = \arctan \frac{y_2}{x_2}$). Затем, используя

$$\Phi^U = \frac{1}{2}(\Phi^S - \Phi^R) \leftrightarrow G(\Phi^U) = G\left(-\frac{\Phi^R}{2}\right) G\left(\frac{\Phi^S}{2}\right) \quad (19)$$

$$\Phi^V = \frac{1}{2}(\Phi^S + \Phi^R) \leftrightarrow G(\Phi^V) = G\left(\frac{\Phi^R}{2}\right) G\left(\frac{\Phi^S}{2}\right) \quad (20)$$

в (15) получается $a'_{12} = a'_{21} = 0$.

Алгоритм Когбетлянца для SVD

Будем рассматривать алгоритм Когбетлянца для квадратной матрицы $A \in R^{n \times n}$.

$$V = I; U = I;$$

для $l = 0, 1, 2, \dots$

для всех пар индексов (p, q)

$$\begin{aligned} A &\leftarrow G_{p,q,l}^T(\Phi^U) \cdot A \cdot G_{p,q,l}^T(\Phi^V) \\ V &\leftarrow V \cdot G_{p,q,l}(\Phi^V) \\ U &\leftarrow U \cdot G_{p,q,l}(\Phi^U) \end{aligned}, \quad (21)$$

где $G_{p,q,l}(\Phi^U)$ и $G_{p,q,l}(\Phi^V)$ - вращения плоскости в (p, q) - плоскости 1-й итерации. Для пар индексов (p, q) используется схема циклического упорядочения по строкам, т.е.

$$(p, q) = (1, 2), (1, 3), \dots, (1, n)(2, 3), \dots (n-1, n) \quad (22)$$

Повороты плоскости $G_{p,q,l}(\Phi^U)$ и $G_{p,q,l}(\Phi^V)$ получены путем решения соответствующей подзадачи $(p, q) 2 \times 2$ SVD для каждого преобразования (21). Следовательно, недиагональное количество

$$S = \sqrt{\|A\|_F^2 - \sum_{i=0}^n a_{ii}^2} \quad (23)$$

уменьшается при каждом преобразовании (21) таким образом, что матрица A сходится к диагональной матрице, содержащей сингулярные значения A (т.е. Σ).

3. SVD-Алгоритм обновления

Алгоритм обновления SVD основан на матрице умножение векторов, этап обновления QRD и вычисление SVD по алгоритму Когбетлянца.

Пусть $X_{[k-1]} = U_{[k-1]}\Sigma_{[k-1]}V_{[k-1]}$ будет SVD из $X_{[k-1]}$ на временном шаге $k - 1$ и $x_{[k]}^T$ будут новым вектором данных. Для того чтобы объединить QRD-обновление и SVD-вычисление, необходимо спроецировать новый вектор данных $x_{[k]}^T$ на уже вычисленную матрицу правильного единственного числа векторы $V_{[k-1]}$:

$$\tilde{x}_{[k]}^T \leftarrow x_{[k]}^T V_{[k-1]} \quad (24)$$

Затем выполняется QRD-обновление с использованием $\tilde{x}_{[k]}^T$ в качестве добавленного вектора:

$$\begin{bmatrix} \tilde{R}_{[k]} \\ O^T \end{bmatrix} \leftarrow \prod_{q=1}^n G_{kq}(\Phi) \begin{bmatrix} \lambda R_{[k-1]} \\ \tilde{x}_{[k]}^T \end{bmatrix} \quad (25)$$

Теперь SVD $\tilde{R}_{[k]}$ вычисляется с использованием алгоритма Когбетлянца. Чтобы уменьшить сложность алгоритма Когбетлянца используется одна развертка или даже часть развертки алгоритма Когбетлянца. Аннигилируем только элементы матрицы $\tilde{r}_{i,i+1}$, ($i = 1, \dots, n - 1$) после каждого обновления, т.е.

$$R_{[k]} \leftarrow \prod_{i=1}^{n-1} G_{i,i+1,[k]}^T(\Phi^U) \cdot \tilde{R}_{[k]} \cdot \prod_{i=1}^{n-1} G_{i,i+1,[k]}(\Phi^V) \quad (26)$$

$$V_{[k]} \leftarrow V_{[k-1]} \prod_{i=1}^{n-1} G_{i,i+1,[k]}(\Phi^V) \quad (27)$$

В таком виде алгоритм обновления SVD требует переориентации матриц $V_{[k]}$. Этой переортогонализации можно избежать, параметризуя $V_{[k]}$ в терминах $n(n - 1)/2$ ортогональных вращений

$$V_{[k-1]} = \prod_{i=1}^n \prod_{j=i+1}^n G_{i,j,[k-1]}(\Phi) \quad (28)$$

и обновляет соответствующие углы поворота, применяя повороты $\prod_{i=1}^{n-1} G_{i,i+1,[k]}(\Phi^V)$ к этой факторизации. Теперь умножение матрицы на вектор (24) также может быть выполнено путем применения вращений. Таким образом, эта форма алгоритма обновления SVD полностью основана на оценке и применении ортонормированных вращений.