Виноградова Арина КМБО-01-20

[arina.airina@yandex.ru](mailto:arina.airina@yandex.ru)

@ari\_grape

Введение. В данной статье представлена реализация алгоритма обновления SVD с использованием ортонормированных -вращений. Ортонормированный -поворот - это поворот на угол из заданного набора углов -поворота (например, углов ), которые выбраны таким образом, что вращение может быть реализовано с помощью небольшого количества операций добавления сдвига. Используется версия алгоритма обновления SVD, в которой все вычисления полностью основаны на оценке и применении нормальных вращений. Следовательно, в таком виде алгоритм обновления SVD поддается к реализации, использующей ортонормированные -вращения, т.е. каждое вращение, выполняемое в алгоритме обновления SVD, будет аппроксимироваться ортонормированными -вращениями. Для всех приближений используется одинаковая точность, т.е. только r<<w (w: длина слова) ортонормированные -вращения используются для аппроксимации точного вращения. Оценка поворота также может быть выполнена путем выполнения -поворотов таким образом, что полный алгоритм обновления SVD может быть выражен в терминах ортонормированных -поворотов. Моделирование показывает эффективность алгоритма обновления SVD, основанного на ортонормированных -поворотах.

Вступление. Вычисляя разложение по сингулярным значениям (SVD) матрицы данных , можно извлечь подпространства данных signal и noise. Знание этих подпространств необходимо во многих приложениях, например, для оценки DOA [1], идентификации системы в пространстве состояний [2], коммуникации [3]. На практике, когда проблемы обычно изменяются во времени, важно иметь возможность отслеживать эти подпространства. Поэтому в последние годы были предложены различные алгоритмы отслеживания подпространства. Эти алгоритмы основаны на разложениях, выявляющих ранг [4, 5], алгоритме Ланцоша [6] или алгоритм обновления SVD [7]. Алгоритм обновления SVD включает в себя новый вектор данных путем умножения матричного вектора, шаг обновления QRD и часть развертки алгоритма SVD Когбетлянца [7, 8]. Эти типы вычислений хорошо подходят для параллельной реализации, и в работе [9] было показано, что все вычисления могут быть легко объединены, что приводит к созданию систолического алгоритма и архитектуры для обновления SVD. Однако эта оригинальная версия алгоритма обновления SVD [7] демонстрирует численные проблемы, поскольку накопление ошибок округления разрушает ортогональность сингулярных векторов. Шаги переортогонализации могут избежать этой проблемы, но не подходят для систолической реализации. В [10] эта задача была решена с помощью параметризации ортонормированной матрицы правых сингулярных векторов на повороты плоскости и обновление соответствующего поворота в-глес. В этом виде алгоритм обновления SVD полностью основан на оценке и применении поворотов плоскости. Эти вращения плоскости могут быть реализованы стандартным способом с использованием квадратных корней и делений или трансцендентных функций. Для упрощения реализации были представлены различные возможности модификации вращений в ортогональной плоскости, например, вращения без квадратного корня или вращения без квадратного корня и деления (см., например, [11]). В отношении эффективного Реализация ASIC (специализированной интегральной схемы) другим широко используемым методом является CORDIC алгоритм [12, 13], т.е. представляющий угол поворота Ф в базисе "" (мы предполагаем конечную длину слова из w бит):

образуют базовые углы, а методы являются цифрами представления. Поворот CORDIC на угол Ф может быть выполнен с помощью w+1 рекурсий, каждая из которых состоит из двух операций добавления сдвига и процедуры масштабирования (это масштабирование также может быть выполнено с помощью операций добавления сдвига). В работах [14, 15] было показано, что идея CORDIC может быть применена для получения приближенных вращений, т.е. поворот на приблизительный угол поворота . Пусть , – набор возможных углов поворота (поворот на один конкретный Ф определяется как ортонормированный -поворот [16, 17]), аппроксимация точного угла поворота на уровне r Ф является Ф = где i1 > i2 > … > ir, и is. Это соответствует представлению приблизительного угла поворота следующим образом:

, где . (2)

Это представление состоит только из определенных углов полной последовательности CORDIC (). Поэтому, в отличие от алгоритма CORDIC, необходимо определить конкретное значение i. Но, с другой стороны, используются только действительно необходимые повороты последовательности CORDIC (сравните, например, два представления (1) и (2) для небольшого угла θ). В [16] был разработан метод, позволяющий оценить оптимальный угол поворота (т.е. ) с использованием также -вращения. Углы -образного поворота i2,i3, … могут быть определены путем итеративного применения та же процедура [14]. Элементарная архитектура для оценки и применения ортонормированных -вращений была представлена в [17]. В [18] было показано, что использование приближенных вращений целесообразно для того, чтобы избежать вычислений квадратного корня или вычислений квадратного корня и деления без снижения производительности алгоритма обновления SVD. В этой статье мы демонстрируем эффективность приближенных вращений, основанных на ортонормированных -вращениях, для алгоритма обновления SVD. Это требует распространения идей, представленных в [14], на SVD, т.е. использования ортонормированных -вращение обсуждается для алгоритма SVD Когбетлянца и применяется к алгоритму обновления SVD, как указано в [10]. Обратите внимание, что только эта численно стабильная версия алгоритма обновления SVD подходит для реализации всего алгоритма, основанного на ортонормированных вращениях, поскольку только эта версия полностью основана на оценке и применении плоских вращений. Моделирование показывает, что очень грубые приближения, т.е. использование r «нормированных -вращений (во всех наших примерах мы используем r = 1) для каждого поворота плоскости, работают следующим образом а также использование точных поворотов (т.е. r = w для точного CORDIC). В разделе 2 мы приводим некоторые предварительные данные. Прежде всего, дается определение вращений в ортогональной плоскости и их реализация с использованием CORDIC. Затем рассматриваются алгоритмы линейной алгебры, составляющие алгоритм обновления SVD, т.е. QRD-обновление и SVD с использованием алгоритма Когбетлянца. В разделе 3 представлен алгоритм обновления SVD, приведенный в [10] требующий только применения и оценки плоскостных вращений по всему алгоритму. В разделе 4 мы описываем вычисление ортонормированных -вращений для подзадач 2 × 1 QRD и 2 × 2 SVD. Показано, как оценка оптимальных ортонормальных -вращений для подзадачи QRD может быть отнесена к выполнению -вращений. Ортонормированные -вращения для подзадачи SVD могут быть вычислены с использованием ортонормированных -вращений, определенных для двух независимых подзадач QRD. Представленная процедура оценки ортонормированных вращений значительно улучшает методы, представленные в [15, 19]. Приближенные вращения, основанные на ортонормированных -вращениях, применяются к алгоритму обновления SVD в разделе 5. В разделе 6 моделирование показывает эффективность представленного алгоритма, а раздел 7 завершает статью.

2. Предварительные замечания

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). Обсуждаемые версии этих алгоритмов полностью основаны на оценке и применении ортонормированных поворотов плоскости. Также обсуждается реализация этих ортонормированных вращений плоскости с использованием CORDIC.

Определение [Ортонормированное вращение плоскости]. Вращение в ортонормированной плоскости (заданное вращение) определяется углом поворота Ф и плоскостью (p, q), в которой происходит вращение, т.е. соотношением и в (pp, pq, qp, qq) позиции единичной матрицы . – это ортонормированное вращение, поскольку . Не теряя общности, мы лишь подробно рассмотрим оценку и применение ортонормальных вращений 2 x 2.

в следующем.

2.2. CORDIC

Процедура CORDIC [12, 13] использует представление (1) для угла поворота Ф. Следовательно, выполняется для базисных углов таким образом, что с помощью (3) получается вращение CORDIC

где коэффициент масштабирования - не зависит от угла поворота тени:

Предпринимались различные попытки устранить масштабирующий фактор или, по крайней мере, привести его к простому двоичному представлению. Делорм [20] предложила метод вычисления переменного коэффициента масштабирования в режиме реального времени. Этот случай возникает для переменной итерации, связанной в (4). Вместо работы с базисными углами базисные углы получаются путем двукратного выполнения поворота на . Относительный двойной поворот задается

Базовые углы двойного поворота задаются соотношением 2-1+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 на, можно использовать следующее простое тождество:

Мы подробнее остановимся на этом, когда будем обсуждать приблизительные ротации.

2.3. QR-декомпозиция

Определение 2 [QR-декомпозиция]. QR-декомпозиция матрицы определяется

, где ортонормировано (), а является верхним треугольником.

QRD подзадача 2 × 1. Вектор поворачивается на угол Ф с помощью

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

Вычисление QR-кода. Треугольная матрица R получается путем решения последовательности подзадач 2 × 1 QRD, т.е. применения последовательности ортонормированных вращений к матрицам X (исходный X перезаписан), где аннулирует мгновенный элемент для , т.е.,

где , такой, что и X перезаписываются R.

QRD-Обновление. Альтернативой триангуляции X по столбцам является выполнение триангуляции строка за строкой. Это приводит к рекурсивному обновлению QRD. Позволь - матрица данных , доступная на временном шаге , а новый вектор данных, измеренный на временном шаге k, который получается

где - фактор забывания.

Учитывая QRD

верхний треугольный коэффициент получен путем добавления нового вектора данных к взвешенной матрице и использования последовательности поворотов Гивенса ) для уничтожения присоединенной строки, т.е.,

2.4. Разложение по сингулярным значениям

Определение 3 [SVD]. SVD матрицы определяется

где и - ортонормированные матрицы (), а диагональная матрица , содержащая сингулярные значения.

Подзадача SVD 2 × 2. Дана матрица 2 × 2 применяются вращения и слева и справа от A:

Вычисление и таким образом, что выполняется, решается подзадача SVD 2 × 2. Вработах [21-23] было показано, что углы и могут бытьопределяется по двум углам и которые могут быть вычисляется независимо путем решения двух подзадач 2 × 1 QRD. С

мы определяем два поворота и что

дает () и (). Затем, используя

в (16) получается .

Алгоритм SVD Когбетлянца. Без ограничения общности мы будем рассматривать только Когбетлянца Алгоритм SVD для квадратной матрицы (хорошо известно, что выгодно применять Алгоритм SVD Когбетлянца к верхней треугольной матрице R, полученной с помощью подготовительного QRD). Алгоритм SVD Когбетлянца приведен следующим образом:

для

для всех пар индексов (p,q)

где и - вращения плоскости в (p,q) - плоскости l-й итерации. Для

пар индексов (p,q) используется схема циклического упорядочения по строкам, т.е.

Эта схема упорядочения может быть сопоставлена со схемой параллельного упорядочения, что делает алгоритм Когбетлянца высокоэффективным подходит для параллельной реализации [24]. Выполнение всех n(n - 1)/2 пар из (23) называется разверткой (l-й разведкой).

Повороты плоскости и получены путем решения соответствующей подзадачи (p, q) 2 × 2 SVD для каждого преобразования (22). Следовательно, недиагональное количество

уменьшается при каждом преобразовании (22) таким образом, что матрица A сходится к диагональной матрице, содержащей сингулярные значения A (т.е. ).

2.5. Замечания по параллельным реализациям. Обсуждаемые алгоритмы (QRD, алгоритм SVD Когбетлянца) хорошо подходят для параллельной реализации. По существу, это происходит из-за представления ортогональных матриц в терминах ортонормированных поворотов плоскости. Такая параметризация (представление) ортогональных матриц обеспечивает параллельную реализацию алгоритмов на систолических массивах [25, 26], а также использование кордической арифметики для реализации процессорных элементов [21] (это также основная идея, лежащая в основе избегания переортогонализации шаги в алгоритме обновления SVD, который обсуждается в следующем разделе). Интересно отметить, что эта параметризация является результатом Эйлера (1770) (можно найти в [27]) в то время, когда ни параллельные реализации, ни компьютерная арифметика не были проблемой.

3. SVD-Алгоритм обновления. Важный результат работы Moonen et al. [7, 9] состояли в том, чтобы подтвердить, что параллельная реализация базовых алгоритмов (QRD-обновление, алгоритм Когбетлянца) может быть хорошо объединена с алгоритмом SVD-обновления. Алгоритм обновления SVD основан на матрице умножение векторов, этап обновления QRD и вычисление SVD по алгоритму Когбетлянца. Пусть будет SVD из на временном шаге k - 1 и будут новым вектором данных. Для того чтобы объединить QRD-обновление и SVD-вычисление, необходимо спроецировать новый вектор данных на уже вычисленную матрицу правильного единственного числа векторы :

Затем выполняется QRD-обновление с использованием в качестве добавленного вектора:

Теперь SVD вычисляется с использованием алгоритма Когбетлянца. Чтобы уменьшить сложность алгоритма Когбетлянца, в работах [7, 8] было показано, что одна развертка или даже часть развертки алгоритма Когбетлянца Алгоритма SVD достаточно для отслеживания подпространства достаточно медленных процессов, изменяющихся во времени. Уничтожая только элементы матрицы после каждого обновления, т.е.

также обеспечивает регулярную реализацию SVD-обновления на систолическом массиве [9]. В таком виде (25)-(28) алгоритм обновления SVD требует переориентации матриц . Этой переортогонализации можно избежать, параметризуя в терминах n(n - 1)/2 ортогональных вращений

(например, QRD ] дает эту факторизацию) и обновляет соответствующие углы поворота, применяя повороты к этой факторизации [10]. Теперь умножение матрицы на вектор (25) также может быть выполнено путем применения вращений. Таким образом, эта форма алгоритма обновления SVD полностью основана на оценке и применении ортонормированных вращений.

4. Ортонормированные -вращения. Все вычисления вращения в алгоритме обновления SVD заменяются приблизительными вращениями далее, где используются ортонормированные -вращения, приводится набор доступных приблизительных вращений.

4. 1. Подзадача QRD. В то время как выполнение точного поворота, как описано в (9) (10), гарантирует , приблизительный поворот , определяемый приблизительным углом поворота , обеспечивает только

при .

Предположим, мы использовали приблизительный угол , уравнение (9) дает:

Представляя это уравнение (31) в соответствии с (32), используя , получаем

Очевидно, что для точного поворота, при котором получается d = 0.

На этом этапе, определив приблизительный поворот, мы используем идею CORDIC [12], т.е. в отношении простой реализации поворота мы ограничиваемся набором приблизительных углов в виде

где . Поэтому мы допускаем только повороты формы

где коэффициент масштабирования, а (не масштабированное) -вращение. называется ортонормированным -вращением.

Поскольку ортонормированное –вращение определяется индексом угла i, вычисляющим оптимальное ортонормированное -вращение соответствует нахождению CORDIC угла , который наиболее близок к точному вращению угол , т. е.

Направление вращения определяется значением . Следовательно, определившись , мы можем работать с чтобы оценить .

Выбор оптимального приблизительного угла в соответствии с (35) эквивалентен

Эта минимизация может быть осуществлена путем определения угловых интервалов таким образом, что является оптимальным углом всякий раз, когда . Границы интервалов следуют из решения уравнения

т. е. угол , при котором выбор приводит к тому же коэффициенту уменьшения , что и при выборе . Решение (37) доходность:

Поскольку ортонормированное p-вращение в (34) определяется одним конкретным шагом рекурсии исходной последовательности CORDIC (4), мы определяем ортонормированное двойное -вращение одним конкретным шагом вращения последовательности двойного вращения (6):

где коэффициент масштабирования, а (не масштабированное) двойное -вращение. Коэффициент масштабирования может быть рекурсивно вычислен с помощью операций сдвига и сложения

при . Поскольку ортонормированный двойной -вращения обеспечивают компенсацию коэффициента масштабирования (40), а поскольку пределы интервалов могут быть легко определяемый для набора ортонормированных двойных -вращений, отныне мы ограничиваем наш набор приближенных вращений ортонормированными двойными -вращениями. Пределы интервалов для выбора оптимального двойного угла поворота теперь задаются как

Поэтому, учитывая вектор (т.е. ) мы используем если и если . Поскольку это решение может быть принято через два не масштабируемое -поворотов и ,т. е. вычислить

Таким образом, мы получаем

Чтобы найти оптимальное значение , еще одно немасштабированное требуется -вращение. Пусть и , соответственно, обозначают мантиссу и показатель степени двузначного числа с плавающей запятой . Поскольку мы можем получить оценку оптимального значение путем вычисления значение , и поскольку получается . Следовательно, можно определить оптимальное значение следующим образом. Вычислять:

Тогда

Эта процедура приводит к оптимальному ортонормированному двойному -вращению , такому что с .

4. 2. Подзадача SVD. Оптимальные ортонормированные двойные -вращения для двух подзадач QRD (18) и (19), т.е. и , может быть определен с помощью процедуры, описанной выше. Учитывая эти вращения, вращения SVD обрабатываются в соответствии с (20) и (21). Поскольку , мы получаем приблизительные вращения для подзадачи SVD следующим образом:

Используя вместо для вращений SVD, мы фактически используем и в качестве приблизительных углов для подзадачи QRD (вместо и ). Следовательно, аппроксимация подзадачи QRD изменяется соответствующим образом, и мы получаем с .

Остается показать, что при гарантируется для каждой подзадачи, чтобы соответствовать требованиям к сходимости алгоритма Когбетлянца [18]. Для двух подзадач QRD (18) и (19) получаем

так что [21]

Следовательно, с помощью (17) получаем

такой, что

Кроме того, вместо мы используем , что гарантирует , а также приводит к улучшенной аппроксимации точных углов. Обратите внимание, что этот метод улучшает коэффициенты уменьшения (и, следовательно, производительность всех алгоритмов), приведенные в [14, 15, 19].

Как обсуждалось в [14, 16], всегда можно повысить точность приблизительных вращений, используя ортонормированных двойных -вращений для приблизительного соответствия точному вращению.

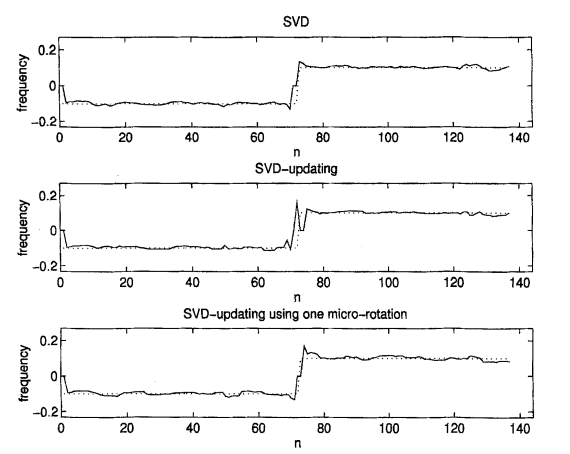


Рис. 1. Оценка частоты для s1 (t) с использованием точного SVD (вверху), с использованием алгоритма обновления SVD с точными поворотами (посередине) и алгоритма обновления SVD, основанного на одном ортонормированном двойном -повороте (внизу).

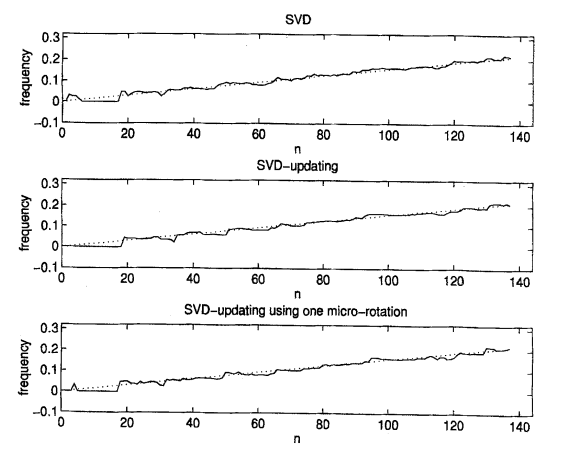


Рис. 2. Оценка частоты для s2(t) с использованием точного SVD (вверху), с использованием алгоритма обновления SVD с точными поворотами (посередине) и алгоритма обновления SVD, основанного на одном ортонормированном двойном -образном повороте (внизу).