Math-Net.Ru

Общероссийский математический портал

М. А. Черепнев, Блочный алгоритм типа Ланцоша решения разреженных систем линейных уравнений, *Дискрет. матем.*, 2008, том 20, выпуск 1, 145–150

DOI: <https://doi.org/10.4213/dm997>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 52.205.19.152

26 апреля 2023 г., 09:20:55

Блочный алгоритм типа Ланцоша решения разреженных систем линейных уравнений

© 2008 г. М. А. Черепнев

В статье предлагается новый блочный алгоритм решения разреженных систем линейных уравнений над $GF(2)$ вида $Ax = b$, $A \in F(N \times N)$, $b \in F(N \times 1)$, где A — симметричная матрица, $F = GF(2)$ — поле из двух элементов. Алгоритм построен с использованием матричных аппроксимаций Паде. Время работы алгоритма при параллельной организации вычислений оценивается величиной $\max\{O(dN^2/n), O(N^2)\}$, где d — максимальное по строкам матрицы A число ненулевых элементов. Если $d < Cn$ для некоторой абсолютной константы C , то эта оценка лучше оценки времени работы известного алгоритма Монгмери.

Пусть требуется решить систему линейных уравнений

$$Ax = b, \quad A \in F(N \times N), \quad b \in F(N \times 1), \quad (1)$$

где A — симметричная матрица, $F = GF(2)$ — поле из двух элементов.

Для матричного многочлена $R(\lambda) \in F(n \times n)[\lambda]$, $n \in \mathbf{N}$, обозначим R_i коэффициент при λ^i , то есть

$$R(\lambda) = \sum_{i=0}^{\deg R(\lambda)} R_i \lambda^i.$$

Пусть также

$$R(A, B) = \sum_{i=0}^{\deg R(\lambda)} A^i B R_i,$$

где $B \in F(N \times n)$, $n \leq N$, а n — длина машинного слова (в дальнейшем будем считать, что $n = 64$). Обозначим $W(B) \in F(N \times n')$, $n' \leq n$, подсовокупность вектор-столбцов матрицы B такую, что матрица $(W(B))^T A W(B)$ невырождена. Эта подсовокупность может быть построена например при помощи алгоритма из §8 статьи [1]. Пусть далее O_n , I_n — соответственно нулевая и единичная матрицы из $F(n \times n)$.

Выберем произвольно матрицу $B \in F(N \times n)$, $n \leq N$. Пусть

$$\alpha(\lambda) = \sum_{i=0}^{2N/(n-1)} \alpha_i \lambda^{-i}, \quad \alpha_i = B^T A^i B. \quad (2)$$

Скалярное произведение $(\varphi(\lambda), \psi(\lambda)) \in F(n \times n)$ двух матричных многочленов определим как коэффициент при λ^{-1} в произведении $\varphi(\lambda)^T \alpha \psi(\lambda)$. Обозначим

$(C, D) = C^T A D \in F(n \times n)$ матрицу A -скалярных произведений вектор-столбцов матриц $C, D \in F(N \times n)$. Из этого определения и симметричности матрицы A следует в частности, что $(C, D) = (D, C)^T$.

Лемма 1. *Справедливо равенство $(\varphi(A, B), \psi(A, B)) = (\varphi(\lambda), \psi(\lambda))$.*

Доказательство. Заметим, что согласно введенным определениям правая и левая части линейны по φ и ψ , поэтому достаточно доказать лемму при $\varphi = \lambda^i, \psi = \lambda^j, i, j \in \mathbb{N} \cup \{0\}$. В этом случае правая часть равна $B^T A^{i+j+1} B$, а левая — $(A^i B)^T A A^j B$, что, ввиду симметричности матрицы A , одно и то же. Лемма доказана.

Пусть имеются матричные приближения Паде

$$\begin{aligned} \alpha(\lambda) Q^{(t-1)}(\lambda) - P^{(t-1)}(\lambda) &= \sum_{i=t}^{N/n} \tau_i \lambda^{-i}, \\ \alpha(\lambda) Q^{(t)}(\lambda) - P^{(t)}(\lambda) &= \sum_{i=t+1}^{N/n} \rho_i \lambda^{-i}, \\ \deg Q^{(s)}(\lambda) \leq s, \quad \deg P^{(s)}(\lambda) \leq s, \quad s \in \{t-1, t\}, \end{aligned} \quad (3)$$

Здесь λ — переменная, $\tau_i, \rho_i \in F(n \times n)$ — матрицы, а $Q^{(s)}(\lambda), P^{(s)}(\lambda) \in F(n \times n)[\lambda]$ — матричные многочлены, $Q^{(-1)} = 0_n, P^{(-1)} = I_n, Q^{(0)} = I_n, P^{(0)} = \alpha_0$.

Рассмотрим задачу построения следующего $(t+1)$ -го приближения с аналогичными оценками на степень и порядок. По аналогии с одномерным случаем рассмотрим матричный многочлен $b_{t+1}(\lambda) = v_1 \lambda + v_0, v_0, v_1 \in F(n \times n)$. Домножим второе из матричных уравнений (3) на этот многочлен справа и сложим с первым. Введем обозначения

$$\begin{aligned} Q^{(t+1)}(\lambda) &= Q^{(t)}(\lambda) b_{t+1}(\lambda) + Q^{(t-1)}(\lambda), \\ P^{(t+1)}(\lambda) &= P^{(t)}(\lambda) b_{t+1}(\lambda) + P^{(t-1)}(\lambda). \end{aligned} \quad (4)$$

Попытаемся обратить в ноль первые коэффициенты получившегося справа ряда

$$\alpha(\lambda) Q^{(t+1)}(\lambda) - P^{(t+1)}(\lambda) = \sum_{i=t+2}^{N/n} \gamma_i \lambda^{-i}. \quad (5)$$

Тогда получим систему линейных матричных уравнений

$$\begin{aligned} \tau_t &= -\rho_{t+1} v_1, \\ \tau_{t+1} &= -\rho_{t+1} v_0 - \rho_{t+2} v_1. \end{aligned} \quad (6)$$

Решив эту систему, получим следующее выражение для первого ненулевого коэффициента $(t+1)$ -го матричного приближения:

$$\gamma_{t+2} = \tau_{t+2} + \rho_{t+2} v_0 + \rho_{t+3} v_1. \quad (7)$$

Для обеспечения разрешимости относительно v_1, v_0 системы (6) в случае, когда $\text{rank } \rho_{t+1} < n$, нужно применять дополнительные вычисления, чтобы сделать шаг алгоритма. Найдем такую матрицу D , что $\rho_{t+1} D$ примет вид, в котором первые слева столбцы линейно независимы, а последние $k = \text{corank } \rho_{t+1}$ равны нулю. Умножим на D

справа второе из равенств (3), сохраняя обозначения. Отметим, что k последних столбцов матрицы $Q^{(t)}(\lambda)$ удовлетворяют ограничениям на степень и порядок для приближения с номером $t+1$. Умножение последних k столбцов матриц $Q^{(t)}(\lambda)$, $P^{(t)}(\lambda)$ на λ присоединяет к линейно независимым столбцам матрицы ρ_{t+1} последние k столбцов матрицы ρ_{t+2} .

Допустим, что ранг полученной матрицы стал полным. Ранг старшего коэффициента $Q_{t+2}^{(t+1)}$ получившегося матричного многочлена $Q^{(t+1)}(\lambda)$ в силу (4) будет не больше k . Найдем такую матрицу E , что матрица $Q_{t+2}^{(t+1)}E$ имеет вид, в котором первые слева k столбцов линейно независимы, а последние равны нулю.

Заменяем первые k столбцов матрицы $Q^{(t+1)}(\lambda)$ на последние k столбцов матрицы $Q^{(t)}(\lambda)$. Теперь матричный многочлен, получившийся в результате этой замены, имеет степень $t+1$ и порядок приближения $t+2$.

В общем случае, если умножение на λ соответствующих столбцов матриц $Q^{(t)}(\lambda)$, $P^{(t)}(\lambda)$ не привело к полному рангу для матрицы $\tilde{\rho}_{t+1}$ первого коэффициента разложения, можно снова привести эту матрицу к виду, в котором первые слева столбцы линейно независимы, а последние l равны нулю. Далее соответствующие столбцы снова умножаются на λ . Полученные $k+l$ хороших приближений дадут возможность сделать шаг алгоритма аналогичным способом. Приближения большего порядка в принципе можно использовать в нескольких следующих шагах. К сожалению, повторное использование хороших приближений в следующем шаге приводит к появлению чисто нулевых приближений, после чего алгоритм быстро вырождается.

Причину, а также способ устранения этого явления помогает найти следующая конструкция.

Лемма 2. Ранг по столбцам над полем матричного полинома $Q_t^{(t)}(\lambda)$ равен рангу матрицы

$$\begin{pmatrix} Q_t^{(t)} \\ Q_{t-1}^{(t)} \\ \vdots \\ Q_0^{(t)} \end{pmatrix}. \quad (8)$$

Доказательство. Элементарным преобразованиям столбцов матричного полинома $Q^{(t)}$ соответствуют те же преобразования столбцов матрицы (8). Приведем матрицу старшего коэффициента $Q_t^{(t)}$ к виду, в котором первые слева столбцы линейно независимы, а остальные нулевые. Отвечающие нулевым столбцам $Q_t^{(t)}$ столбцы матрицы $Q_{t-1}^{(t)}$ также приведем к виду, в котором слева стоят линейно независимые столбцы, а остальные нулевые, и так далее. Из того, что столбцы разной степени линейно независимы над полем, следует, что коранг обеих матриц равен числу чисто нулевых столбцов, получившихся после окончания описанной процедуры. Лемма доказана.

Систему уравнений (6) и преобразование (4) запишем в следующем матричном виде:

$$\begin{pmatrix} \rho_{t+1} & O_n & \tau_t \\ \rho_{t+2} & \rho_{t+1} & \tau_{t+1} \\ Q_t^{(t)} & O_n & O_n \\ Q_{t-1}^{(t)} & Q_t^{(t)} & O_n \\ Q_{t-2}^{(t)} & Q_{t-1}^{(t)} & Q_{t-1}^{(t-1)} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} v_1 \\ v_0 \\ I_n \end{pmatrix} = \begin{pmatrix} O_n \\ O_n \\ Q_{t+1}^{(t+1)} \\ Q_t^{(t+1)} \\ Q_{t-1}^{(t+1)} \\ \vdots \end{pmatrix}. \quad (9)$$

Верхние блочные строки левой матрицы в (9), содержащие коэффициенты рядов, будем называть верхней частью, остальное — нижней частью. Если предположить, что приближение с номером t имеет по столбцам нетривиальный коранг $l > 0$, а приближение с номером $t - 1$ — коранг $l' \geq 0$, то после некоторого невырожденного преобразования столбцов вся матрица имеет $2l + l'$ нулевых столбцов. Верхняя часть ввиду того, что число строк на $n = 64$ меньше числа столбцов, как правило, имеет полный строчный ранг. Если привести ее к треугольному виду, то при помощи Гауссовых исключений можно в нижней части сделать нулевыми еще $2 \cdot 64$ столбцов, стоящих под лидирующими элементами верхней части. Ранг, преобразованной таким образом нижней части, а значит, и ранг результата будет не больше $64 - (2l + l')$.

Рассмотрим аналогичные (9) преобразования с большим числом $k + 3$ блочных столбцов при $t \geq k \geq 0$

$$\begin{pmatrix} O_n & O_n & O_n & \dots & \sigma_{t-k} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ O_n & O_n & O_n & \dots & \sigma_{t-1} \\ \rho_{t+1} & O_n & \tau_t & \dots & \sigma_t \\ \rho_{t+2} & \rho_{t+1} & \tau_{t+1} & \dots & \sigma_{t+1} \\ Q_t^{(t)} & O_n & O_n & \dots & O_n \\ Q_{t-1}^{(t)} & Q_t^{(t)} & O_n & \dots & O_n \\ Q_{t-2}^{(t)} & Q_{t-1}^{(t)} & Q_{t-1}^{(t-1)} & \dots & O_n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Q_{t-k-2}^{(t)} & Q_{t-k-1}^{(t)} & Q_{t-k-1}^{(t-1)} & \dots & Q_{t-k-1}^{(t-k-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_0 \\ \nu_{-1} \\ \vdots \\ \nu_{-k-1} \end{pmatrix} = \begin{pmatrix} O_n \\ \vdots \\ O_n \\ Q_{t+1}^{(t+1)} \\ \vdots \\ Q_{t-k-1}^{(t+1)} \\ \vdots \end{pmatrix}. \quad (10)$$

Ввиду сильно выраженной прямоугольной формы верхней части она, как правило, имеет полный строчный ранг равный $(k + 2)64$, в то время как ранг нижней части не превосходит $(k + 3)64$. Таким образом, для получения полного ранга в столбце справа достаточно, чтобы разность между рангами верхней и нижней частей равнялась 64. Для этого достаточно добиться того, чтобы нижняя часть левой матрицы имела полный ранг, например, чтобы диагональные блоки, кроме последнего, были невырождены. С этой целью можно применить преобразования, не меняющие структуру рассматриваемой матрицы: элементарные преобразования с отдельными приближениями i -го шага, приводящими верхний блок к виду, в котором первые слева столбцы линейно независимы, а остальные нулевые, а затем умножение отдельных приближений i -го шага, имеющих меньшую степень на λ . Если за одну итерацию полного ранга достичь не удалось, можно повторить эту процедуру не более $i - t + k + 1$ раз при $i = t$, так же как и при $i = t - 1$, не более k раз. В результате умножения на λ содержимое соответствующего столбца в нижней и верхней части сдвигается на одну блочную позицию вверх.

Численные эксперименты, проведенные И. А. Поповяном, показывают, что указанное увеличение числа блочных столбцов последовательно на единицу в случае невозможности достижения нужного ранга с меньшим числом столбцов достаточно эффективно. Алгоритм, позволяющий наращивать число столбцов до 9, успешно обрабатывает квадратную симметричную матрицу размера $N = 2000$ с вероятностью около 0,75, при этом с увеличением размера матрицы при сохранении значения вероятности, максимальное число используемых столбцов растет логарифмически. Кроме того среднее число столбцов

по всем шагам оказывается меньше 5. Заметим также, что для успешной работы алгоритма достаточно преобразовывать только те столбцы, которые участвуют в нетривиальной линейной комбинации столбцов, равной нулю, а для подсчета ранга нижней части можно ограничиться верхним квадратом плюс еще несколько блочных строк.

Указанные процедуры, как и в алгоритме Монтгомери [1], повторяем до полного вырождения последовательности матриц $W(Q^{(i)}(A, B))$.

В случае если алгоритм на каком-то шаге не срабатывает и если при этом решение системы (1) не лежит в линейной оболочке уже построенных векторов, можно начать сначала с другой матрицей B или рандомизировать матрицу A , применяя соображения описанные в [2, 3].

Так как равен нулю коэффициент при λ^{-1} в

$$(Q^{(j)}(\lambda))^T \alpha(\lambda) Q^{(t+1)}(\lambda) = (Q^{(j)}(\lambda))^T \sum_{i=t+2}^{N/n} \gamma_i \lambda^{-i},$$

$(Q^{(j)}(A, B), Q^{(t+1)}(A, B)) = O_n$ при любых $j, t, 0 \leq j \leq t \in \{0, 1, \dots, N/n - 1\}$. Поэтому столбцы матриц $Q^{(i)}(A, B)$ лежат в попарно A -ортогональных подпространствах, а их число, или, точнее сказать, число среди них таких, которые дают невырожденную матрицу попарных A -скалярных произведений, должно быть приблизительно равно размерности всего пространства. Для того чтобы этого добиться, описанные процедуры нужно повторять несколько более чем N/n раз. Положим $W_t = W(Q^{(t)}(A, B))$. Ожидаемый ранг матрицы (W_t, W_t) предположительно будет лишь чуть меньше n . Если этот ранг в среднем оказался меньшим на единицу, то при решении невырожденной системы линейных уравнений (1) можно считать достаточным число указанных итераций, приблизительно равное $N/n + N/n^2 + \dots = N/(n - 1)$. В случае вырожденной системы следует заменить N на $\text{rank } A$.

Решение системы (1) можно получить по формуле (8) из [1]

$$X = \sum_{i=0}^{m-1} W_i (W_i^T A W_i)^{-1} W_i^T b,$$

где m чуть больше N/n .

Для того, чтобы доказать, что оценка числа операций в основной части алгоритма есть $O(dN^2/n)$, где d — максимальное по строкам матрицы A число ненулевых элементов, достаточно для числа операций на одном шаге получить оценку $O(dN)$, то есть такую же по порядку, как для умножения на матрицу A . При использовании на шаге алгоритма трех последовательных приближений

$$Q^{(t+1)}(\lambda) = \lambda Q^{(t)}(\lambda) \tilde{v}_1 + Q^{(t)}(\lambda) \tilde{v}_0 + Q^{(t-1)}(\lambda) \tilde{v}_{-1} + Q^{(t-2)}(\lambda) \tilde{v}_{-2},$$

это будет следовать из равенств

$$Q^{(t+1)}(A, B) = A Q^{(t)}(A, B) \tilde{v}_1 + Q^{(t)}(A, B) \tilde{v}_0 + Q^{(t-1)}(A, B) \tilde{v}_{-1} + Q^{(t-2)}(A, B) \tilde{v}_{-2}.$$

Так же, как и в алгоритме Монтгомери, здесь только одна сложная операция умножения на матрицу A и столько же слагаемых, как и в соответствующей формуле Монтгомери. Число элементов α_i , которые нужно хранить для вычисления первых элементов в правых частях равенств (3) на шаге t , не превосходит $t+1 \leq N/(n - 1)$, а число блочных операций

для этих вычислений во всем алгоритме есть $O((N/n)^2)$. Численный эксперимент показывает, что среднее число θ по всем шагам дополнительных умножений на A примерно равно 1,5. Еще примерно N/n умножений плотных блоков на матрицу A и $2N/n$ умножений плотных блоков друг на друга потребуется для вычисления всех необходимых α_i , $i = 1, 2, \dots, 2N/(n-1)$, в равенстве (2). Важно отметить, что эти вычисления могут быть проделаны параллельно друг другу и основной части алгоритма. Тем самым окончательная оценка сложности имеет главный член вида $\max\{(1+\theta)NA(d, N, n)/n, 2NM(N, n)/n\}$, где $A(d, N, n)$ — число операций для умножения матрицы A на плотный блок, а $M(N, n)$ — число операций для умножения плотных блоков друг на друга. При помощи объединения отдельных столбцов в блоки и резервирования в основном блоке места для дополнительных умножений на матрицу A , величину θ можно уменьшить и возможно приблизить к нулю. Для сравнения, в алгоритме Монтгомери скалярные произведения плотных блоков делаются в шаге алгоритма, в каждом по 2, что дает в оценке главный член вида $(N/n)A(d, N, n) + (2N/n)M(N, n)$, где $A(d, N, n) = O(dN)$, $M(N, n) = O(Nn)$ (см. [1]). Тем самым после несложных преобразований получаем, что предложенный алгоритм имеет лучшую оценку при $A(d, N, n) < (2/\theta)M(N, n)$, другими словами, если $A(d, N, n) < C_1 dN$, $M(N, n) > C_2 Nn$, то при $d < 2C_1 n/(\theta C_2)$.

Список литературы

1. Montgomery P. L., A block Lanczos algorithm for finding dependencies over $GF(2)$. *Lect. Notes Computer Sci.* (1995) **921**, 106–120.
2. LaMacchia B. A., Odlyzko A. M., Solving large sparse linear systems over finite fields. *Lect. Notes Computer Sci.* (1995) **537**, 109–133.
3. Eberly W., Kaltofen E., On randomized Lanczos algorithms. In: *Proc. ISSAC'97* (Küchlin W. W., ed.). ACM Press, New York, 1997, pp. 176–183.

Статья поступила 18.04.2007.