Sparse Matrix-Vector Multiplication

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen University

Contents

Outline

Objective and Learning Outcomes

Objective

- Discussion of blocking for sparse matrix-vector multiplication as a nice introductory cases study for handling combinatorial problems in scientific computing.

Learning Outcomes

- You will understand
  - storage schemes for sparse matrices
  - $(1, B)$-blocking method
  - computational complexity of optimal $(1, B)$-blocking
  - approximate solution of $(1, B)$-blocking
- You will be able to
  - convert sparse matrices into compressed storage
  - solve the $(1, B)$-blocking problem heuristically by reduction to Traveling Sales Person.

Outline

For $F : \mathbf{R}^n \to \mathbf{R}^n$ we look for $\mathbf{x}^*$ such that $F(\mathbf{x}^*) = 0$. Newton's method iterates

$$\mathbf{x}^{i+1} = \mathbf{x}^i + F'(\mathbf{x}^i)^{-1} \cdot F(\mathbf{x}^i)$$

for $i = 0, \ldots, n-1$ and a given starting point $\mathbf{x}^0$.

The Newton step $d\mathbf{x}^i = -F'(\mathbf{x}^i)^{-1} \cdot F(\mathbf{x}^i)$ is computed as the solution of the system of linear equations

$$F'(\mathbf{x}^i) \cdot d\mathbf{x}^i = -F(\mathbf{x}^i)$$

for example, using stationary iterative Methods such as Jacobi or Gauss-Seidel. Both involve evaluations of (sparse) matrix-vector products.

Solve $A\mathbf{x} = \mathbf{b}$ iteratively by

$$\mathbf{x}_{k+1} = G\mathbf{x}_k + C$$

for some starting value $\mathbf{x}_0$ that is not too far from the solution. Choose G and C such that the fixed point

$$\mathbf{x} = G\mathbf{x} + C$$

is the solution to $A\mathbf{x} = \mathbf{b}$ that is

$$A(G\mathbf{x} + C) = \mathbf{b} \quad .$$

To ensure convergence the absolute value of the largest Eigenvalue of G (the spectral radius of G) needs to be strictly lower than one, i.e, G needs to be contractive.

Splitting

$$A = D + L + U$$

where D is the diagonal of A and L and U are the corresponding lower and upper triangular matrices, respectively. Assuming that A has no zeros on the diagonal, that is D is nonsingular and hence invertible, we get

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - (L + U) \cdot \mathbf{x}_k)$$

from

$$\begin{aligned} A \cdot \mathbf{x} &= \mathbf{b} \\ D \cdot \mathbf{x} + (L + U) \cdot \mathbf{x} &= \mathbf{b} \\ D \cdot \mathbf{x} &= \mathbf{b} - (L + U) \cdot \mathbf{x} \\ \mathbf{x} &= D^{-1}(\mathbf{b} - (L + U) \cdot \mathbf{x}) \quad \text{(fixed-point iteration).} \end{aligned}$$

$\Rightarrow$ (sparse) matrix-vector product

Outline

Also: Matrix Market format; see `https://sparse.tamu.edu/`

For

$$\begin{pmatrix} a_{0,0} & a_{0,1} & 0 \\ 0 & a_{1,1} & 0 \\ a_{2,0} & 0 & a_{2,2} \end{pmatrix}$$

we get

$$(0,0,a_{0,0}),(0,1,a_{0,1}),\ldots(2,2,a_{2,2})$$

Potential symmetry can be exploited by storing only the lower triangular part.

For

$$\begin{pmatrix} a_{0,0} & a_{0,1} & 0 \\ 0 & a_{1,1} & 0 \\ a_{2,0} & 0 & a_{2,2} \end{pmatrix}$$

we get

$$\begin{aligned} \mathbf{a}^T &= (a_{0,0}, a_{0,1}, a_{1,1}, a_{2,0}, a_{2,2}) \\ \kappa^T &= (0, 1, 1, 0, 2) \\ \rho^T &= (0, 2, 3, 5) \end{aligned}$$

For

$$\begin{pmatrix} a_{0,0} & a_{0,1} & 0 \\ 0 & a_{1,1} & 0 \\ a_{2,0} & 0 & a_{2,2} \end{pmatrix}$$

we get

$$\begin{aligned} \mathbf{a}^T &= (a_{0,0}, a_{2,0}, a_{0,1}, a_{1,1}, a_{2,2}) \\ \rho^T &= (0, 2, 0, 1, 2) \\ \kappa^T &= (0, 2, 4, 5) \end{aligned}$$

Outline

for $i = 0$ **to** $m-1$ **do**

$\quad y_i = 0$

$\quad$ **for** $j = \rho_i$ **to** $\rho_{i+1} - 1$ **do**

$\quad\quad y_i = y_i + a_j \cdot x_{\kappa_j}$

Good temporal locality of an algorithm is characterized by minimal time between accesses to the same data items. Good spatial locality is present if consecutively accessed data items are close in memory.

We observe good spatial locality in $\mathbf{a}$, κ, ρ, and $\mathbf{y}$ Temporal locality is limited to $\mathbf{y}$. Accesses to $\mathbf{x}$ can be irregular (missing temporal and spatial locality; likely cache misses). Three load operations are required for two arithmetic operations in the statement inside the inner loop.

$\mathbf{y} = A\mathbf{x}$ in RCS

Exercise

$$\begin{pmatrix} a_{0,0} & a_{0,1} & 0 \\ 0 & a_{1,1} & 0 \\ a_{2,0} & 0 & a_{2,2} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}$$

$$\mathbf{a}^T = (a_{0,0}, a_{0,1}, a_{1,1}, a_{2,0}, a_{2,2})$$
$$\kappa^T = (0, 1, 1, 0, 2)$$
$$\rho^T = (0, 2, 3, 5)$$

for $i = 0$ **to** 2 **do**
 $y_i = 0$
 for $j = \rho_i$ **to** $\rho_{i+1} - 1$ **do**
  $y_i = y_i + a_j \cdot x_{\kappa_j}$

Outline

Software and Tools for Computational Engineering RWTH AACHEN UNIVERSITY

... exploits distributivity of matrix-vector multiplication, that is

$$(A_1 + A_2) \cdot x = A_1 \cdot x + A_2 \cdot x$$

... aims to maximize the size of dense $(1, B)$-submatrices in order to

1. improve spatial locality in $\mathbf{x}$
2. decrease storage required for A
3. decrease number of load operations in matrix-vector product.

Dense $(1, B)$-blocks allow for storage of single index of first element (instead of B individual indices) and efficient sequential access due to data locality.

$(1, B)$-Blocking

Decomposition of Matrix-Vector Product

Decomposition of A with maximum block size B into a sum of $(1, b)$-blocked matrices for $1 \leq b \leq B$ enables efficient multiplication of individual matrices with the given vector $\mathbf{x}$ e.g,

$$A = A^2_{1,2} + A^2_{1,1} = A^3_{1,3} + A^3_{1,2} + A^3_{1,1}$$

$$\begin{pmatrix} * & * & * & & \\ & * & & * & \\ * & * & & & \\ & & & * & * \\ & & * & * & * \end{pmatrix} = \begin{pmatrix} * & * & & & \\ & & & & \\ * & * & & & \\ & & & * & * \\ & & * & * & \end{pmatrix} + \begin{pmatrix} & & * & & \\ & * & & * & \\ & & & & \\ & & & & \\ & & & & * \end{pmatrix}$$

$$= \begin{pmatrix} * & * & * & & \\ & & & & \\ & & & & \\ & & & & \\ & & * & * & * \end{pmatrix} + \begin{pmatrix} & & & & \\ & & & & \\ * & * & & & \\ & & & * & * \\ & & & & \end{pmatrix} + \begin{pmatrix} & & & & \\ & * & & * & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

$(1, B)$-Blocking

Example

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

$(1, 2)$-Blocking (Exercise: symmetric RCS)

$$\mathbf{a}_2 = (4, -1, -1, 4, 4, -1, -1, -1) \qquad \mathbf{a}_1 = (-1, -1, -1, 4)$$
$$\kappa_2 = (0, 0, 2, 1) \qquad \kappa_1 = (2, 3, 0, 3)$$
$$\rho_2 = (0, 1, 2, 3, 4) \qquad \rho_1 = (0, 1, 2, 3, 4)$$

$(1, 3)$-Blocking (Exercise: symmetric RCS)

$$\mathbf{a}_3 = (4, -1, -1, -1, -1, 4) \qquad \mathbf{a}_2 = (-1, 4, 4, -1) \qquad \mathbf{a}_1 = (-1, -1)$$
$$\kappa_3 = (0, 1) \qquad \kappa_2 = (0, 2) \qquad \kappa_1 = (3, 0)$$
$$\rho_3 = (0, 1, 1, 1, 2) \qquad \rho_2 = (0, 0, 1, 2, 2) \qquad \rho_1 = (0, 0, 1, 2, 2)$$

for $i = 0$ **to** $m - 1$ **do**
    $y_i = 0$
    for $j = \rho_i$ **to** $\rho_{i+1} - 1$ **do**
        $y_i = y_i + \left\langle (a_k)_{k=j\cdot B,\ldots,j\cdot B+B-1}, (x_k)_{k=\kappa_j,\ldots,\kappa_j+B-1} \right\rangle$

where the inner product $\left\langle (a_k)_{k=j\cdot B,\ldots,j\cdot B+B-1}, (x_k)_{k=\kappa_j,\ldots,\kappa_j+B-1} \right\rangle$ can / should be implemented efficiently, for example, using BLAS.[1]

Exercise: $A^3_{1,3} \cdot \mathbf{x}$ for $A^3_{1,3} = (\mathbf{a}_3, \kappa_3, \rho_3)$

[1] `www.netlib.org/blas`

Given an $m \times n$ matrix $A \equiv (a_{i,j})$, find an ordering of columns in A to maximize the number of (i,j) pairs satisfying $a_{i,j} \neq 0$ and $a_{i,j+1} \neq 0$.

Note: Switching two columns in A requires switching of the corresponding entries in $\mathbf{x}$.

Example: Pick a good ordering for

$$\begin{pmatrix} a & b & c & & d \\ & e & f & & g \\ h & & i & & \\ & & & j & k \\ & & & l & m \end{pmatrix}$$

out of $5! = 120$ options ...

BS Problem

Example

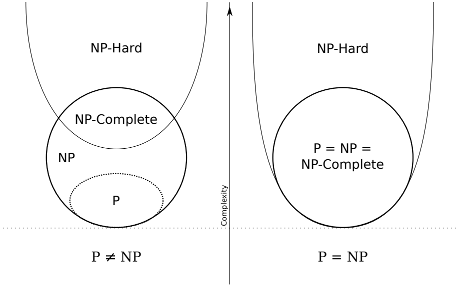
$$\begin{pmatrix} a & b & c & & d \\ & e & f & & g \\ h & & i & & \\ & & & j & k \\ & & & l & m \end{pmatrix} \qquad \begin{pmatrix} a & c & b & d & \\ & f & e & g & \\ h & i & & & \\ & & & k & j \\ & & & m & l \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} 1 & 3 & 2 & 5 & 4 \end{pmatrix}$$

Given an $m \times n$ matrix $A \equiv (a_{i,j})$ and an integer $K > 0$.

Is there an ordering of columns in A such that the number of (i, j) pairs satisfying $a_{i,j} \neq 0$ and $a_{i,j+1} \neq 0$ is greater than or equal to K?

The BS problem is NP-complete.



- P: solvable and verifiable in polynomial[2] time, e.g, sorting
- NP: verifiable in polynomial time, e.g, decision version of BS
- NP-hard: not solvable in polynomial time, e.g, optimization version of BS
- NP-complete: not solvable, but verifiable in polynomial time, decision version of BS

[2] polynomial in the size of the given problem formulation

- Pick known NP-complete problem K, e.g,
  - Hamiltonian Path (HP): Given a graph[3] $G = (V, E)$. Does G contain a Hamiltonian path (a path that contains all vertices in G exactly once)?

- Derive an invertible polynomial construction of an instance of BS for each instance in K.

  A polynomial algorithm for BS would also solve K yielding a contradiction to $P \neq NP \ldots$

- Show that any proposed solution to BS can be validated with polynomial computational cost.

[3] Graphs do not contain parallel edges, that is, all $(i, j) \in E$ are unique.

Reduction from HP:

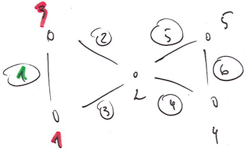
For given $G = (V, E)$ construct $A \in \boldsymbol{R}^{|E|\times|V|}$ such that

$$e_j = (i, k) \in E \quad \Leftrightarrow \quad a_{j,i} \neq 0 \ \vee \ a_{j,k} \neq 0.$$

Two adjacent columns can share nonzeros in at most one row as there are no parallel edges allowed. There can be at most $|V| - 1$ $(1, 2)$-blocks after reordering, achieved when the vertices of consequent columns share an edge in G, which defines a Hamiltonian path in G.

A given solution of BS can be validated in polynomial time by simply counting the number of consecutive nonzero entries. □

Reduction HP $\to$ BS



$\Rightarrow$

	1	2	3	4	5
1	×		×		
2		×	×		
3	×	×			
4		×		×	
5		×			×
6				×	×

$G = (V, E)$

$A \in \mathbb{R}^{|E| \times |V|}$

1, 2, 3, 4, 5 not a Hamilton path

Reorder ...

- Pick a well-studied NP-complete problem K, e.g,
  - Traveling Sales Person (TSP): [4] Given a graph $G = (V, E)$ with weights $w_{i,j}$ on its edges (i, j) find a Hamiltonian path $(v_1, \ldots, v_n)$ that maximizes $\sum_{j=1}^{n-1} w_{v_j,v_{j+1}}$.

  Derive an efficiently invertible construction of an instance of K for each instance in BS.

- Apply known algorithm to instance of K.

- Map solution back to instance of BS.

[4] TSP cannot be "easier" than HP.

For given $A \in \mathbb{R}^{m \times n}$ construct $G = (V, E)$ where

$$V = \{1, \ldots, n\}$$

(Vertices in G represent columns in A) and

$$E = \{(i, j) | \exists k \in \{1, \ldots, n\} : a(k, i) \neq 0 \wedge a(k, j) \neq 0\}$$

(Vertices in G are connected by an edge if the corresponding columns in A have both nonzero entries in the same row.)

A weight $w_{i,j} > 0$ is associated with every edge (i, j) such that

$$w_{i,j} = |\{k \in \{1, \ldots, n\} : a(k, i) \neq 0 \wedge a(k, j) \neq 0\}|$$

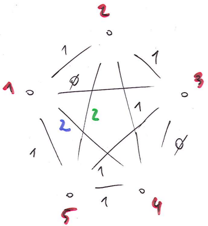
(Edges in G are weighted with the numbers of rows in which the corresponding columns in A have both nonzero entries.)

Solving BS

Reduction to TRAVELING SALES PERSON

Reduction BS → TSP

	1	2	3	4	5
1	×	×		×	×
2		×	×		×
3	×			×	
4				×	
5			×		



Heuristic: Start: 1; Longest Next + Natural Order as Tie-Breaker

1, 4, 2, 5, 3 ⟹ Objective = 6

⟹ 6 consecutive pairs

Greedy heuristic:

1. Pick start vertex π_1 at random.
2. Pick π_{i+1} for $i = 1, \ldots, n-1$ such that $w_{i,i+1}$ is maximized.

Solves example for $\pi_1 = 1$.

Exercise: Try $\pi_1 = 2$ and $\pi_1 = 4$.

Outline

Objective and Learning Outcomes

Motivation

Storage Schemes For Sparse Matrices

$\mathbf{y} = A\mathbf{x}$ in RCS

$(1, B)$-Blocking

- Matrix-Vector Product in $(1, B)$-Blocked RCS
- Block Size Problem
- Block Size is NP-Complete
- Solving Block Size

Summary and Next Steps

Summary

- Block Size turns out to be a nice introductory cases study for handling combinatorial problems in scientific computing.

Next Steps

- Work through the exercises.
- Continue the course to find out more ...