Symmetric computations

Suraj Kumar

Inria & ENS Lyon

Email:*suraj.kumar@inria.fr*

CR12: September 2023

https://surakuma.github.io/courses/daamtc.html

Symmetric Rank-K (SYRK) update

$C = A \cdot A^T$, where A is an $n_1 \times n_2$ matrix and C is an $n_1 \times n_1$ symmetric matrix.

- It multiplies a matrix with its transpose
- It requires roughly half of the computation of general matrix multiplication due to symmetry
- $C_{ij} = \sum_{k=0}^{n_2-1} A_{ik} A_{jk}$ for $0 \leq j \leq i \leq n_1$

SYRK pseudo code:

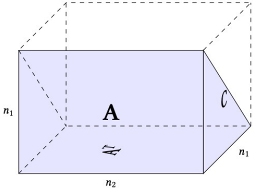
for $i = 0{:}n_1 - 1$

  for $j = 0{:}i$

    for $k = 0{:}n_2 - 1$

      $C[i][j] += A[i][k] * A[j][k]$

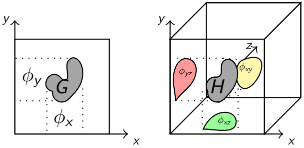
SYRK iteration space:



Here we assume that each $C[i][j]$ is initialized to 0 in the beginning.

Loomis-Whitney inequalitiy

- For the 2d object G, $Area(G) \leq \phi_x \phi_y$
- For the 3d object H, $Volume(H) \leq \sqrt{\phi_{xy}\phi_{yz}\phi_{xz}}$



Lemma

Let $V \in \mathbb{Z}^3$ and $\phi_{ij}(V)$ be the projection of V on the $i-j$ plane, i.e., $\phi_{ij}(V) = \{(i,j) : \exists k, (i,j,k) \in V\}$. Similarly $\phi_{jk}(V)$ and $\phi_{ik}(V)$ are defined. Then

$$|V| \leq |\phi_{ij}(V)|^{\frac{1}{2}} |\phi_{jk}(V)|^{\frac{1}{2}} |\phi_{ik}(V)|^{\frac{1}{2}}.$$

Extension of Loomis-Whitney inequality

Theorem (Lemma 3, Ballard et al., SPAA 2023.)

Let $V = \{(i,j,k) \in \mathbb{Z}^3 : j < i\}$ and $\phi_{ij}(V)$ be the projection of V on the $i-j$ plane, i.e., $\phi_{ij}(V) = \{(i,j) : \exists k, (i,j,k) \in V\}$. Similarly $\phi_{jk}(V)$ and $\phi_{ik}(V)$ are defined. Then

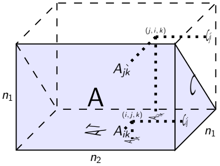
$$2|V| \leq |\phi_{jk}(V) \cup \phi_{ik}(V)| \left(2|\phi_{ij}(V)|\right)^{\frac{1}{2}}.$$

Proof: Let $\tilde{V} = \{(i,j,k) \in \mathbb{Z}^3 : (j,i,k) \in V\}$.

$$|V| = |\tilde{V}| \text{ and } |V \cup \tilde{V}| = 2|V|$$

$$\phi_{ij}(\tilde{V}) = \{(i,j) : (j,i) \in \phi_{ij}(V)\} \text{ and}$$
$$|\phi_{ij}(V) \cup \phi_{ij}(\tilde{V})| = 2|\phi_{ij}(V)|$$

$$\phi_{jk}(\tilde{V}) = \phi_{ik}(V),\ \phi_{ik}(\tilde{V}) = \phi_{jk}(V) \text{ and}$$
$$\phi_{ik}(V \cup \tilde{V}) = \phi_{jk}(V \cup \tilde{V}) = \phi_{jk}(V) \cup \phi_{ik}(V)$$



Applying the previous lemma (Loomis-Whitney inequality) on $V \cup \tilde{V}$ yields the mentioned inequality.

We focus on the computation of the entries below the diagonal of C.

$$
\begin{array}{l}
\text{for } i = 0{:}n_1 - 1 \\
\quad \text{for } j = 0{:}i - 1 \\
\quad\quad \text{for } k = 0{:}n_2 - 1 \\
\quad\quad\quad C[i][j] += A[i][k] * A[j][k]
\end{array}
$$

- The computation is load balanced – each processor performs $\frac{n_1(n_1-1)n_2}{2P}$ loop iterations
- One copy of data is in the system – there exists a processor whose input data at the start plus output data at the end must be at most $\frac{n_1(n_1-1)/2+n_1n_2}{P}$ words
  - Will analyze data transfers for this processor
  - F be the set of indices (i, j, k) associated with loop iterations performed on this processor

Optimization problem to compute communication lower bound

$$\textit{Minimize } |\phi_{ik}(F) \cup \phi_{jk}(F)| + |\phi_{ij}(F)| \text{ s.t.}$$

$$(2|\phi_{ij}(F)|)^{\frac{1}{2}} \, |\phi_{ik}(F) \cup \phi_{jk}(F)| \geq 2|F| = \frac{n_1(n_1-1)n_2}{P}$$

$|\phi_{ik}(F) \cup \phi_{jk}(F)|$: number of accessed entries of A

$|\phi_{ij}(F)|$: number of accessed entries of C

- Using Lagrange multipliers, optimal value obtained when $|\phi_{ik}(F) \cup \phi_{jk}(F)| = 2|\phi_{ij}(F)| = \left(\frac{n_1(n_1-1)n_2}{P}\right)^{\frac{2}{3}}$
- Lower bound $= |\phi_{ik}(F) \cup \phi_{jk}(F)| + |\phi_{ij}(F)| -$ data owned by the processor
  $= \frac{3}{2}\left(\frac{n_1(n_1-1)n_2}{P}\right)^{\frac{2}{3}} - \frac{n_1(n_1-1)/2+n_1n_2}{P}$

Symmetric Rank-2k (SYR2K) update

$C = A \cdot B^T + B \cdot A^T$, where A and B are $n_1 \times n_2$ matrices. C is an $n_1 \times n_1$ symmetric matrix.

SYR2K pseudo code:

for $i = 0{:}n_1 - 1$
  for $j = 0{:}i$
    for $k = 0{:}n_2 - 1$
      $C[i][j] += A[i][k] * B[j][k] + A[j][k] * B[i][k]$

Assignment 3 – deadline Oct 5

Question: Obtain parallel memory-independent communication lower bound for SYR2K. Assume that all the operations of each loop iteration are performed on the same processor, i.e, $A[i][k] * B[j][k]$ and $A[j][k] * B[i][k]$ are computed on one processor.