Frequent Direction Algorithms for Approximate Matrix Multiplication with Applications in CCA

Qiaomin Ye Luo Luo Zhihua Zhang

Department of Computer Science and Engineering
Shanghai Jiao Tong University

IJCAI, 2016

Outline

Introduction

Approximate matrix multiplication

we are given A, B each with a large number of rows n, and the goal is to compute some matrix X, s.t.

$$\|A^T B - X\|_X$$

is small, for some matrix norm $\|\cdot\|_X$.

Introduction

Approximate matrix multiplication

we are given A, B each with a large number of rows n, and the goal is to compute some matrix X, s.t.

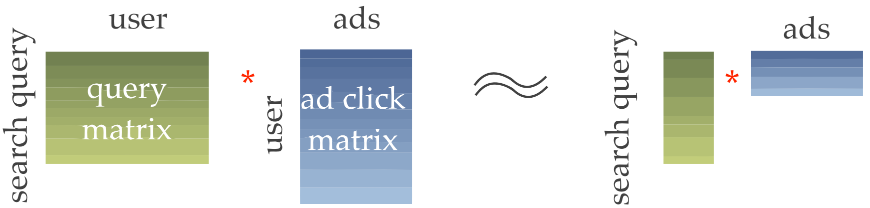
$$\|A^T B - X\|_X$$

is small, for some matrix norm $\|\cdot\|_X$.

- Image processing
- Information retrieval
- Approximate leverage scores
- Large-scale k-means clustering

Introduction

Approximate matrix multiplication



- Image processing
- Information retrieval
- Approximate leverage scores
- Large-scale k-means clustering

Streaming Matrices

- Matrix too large to be stored.
- Streaming setting:
  - Use a small amount of memory.
  - Make a single pass over the data.

Streaming Matrices

- Matrix too large to be stored.
- Streaming setting:
  - Use a small amount of memory.
  - Make a single pass over the data.

$$A^T B = \sum_i^n a_i^T b_i$$

Naive solution

Compute $A^T B$ in time $\mathcal{O}(nm_1 m_2)$.

Outline

Random Selection

- Random Selection. It randomly picks ℓ rows of A and B with probability

$$p_i = \|a_i\|\|b_i\| / \sum_{i=1}^{n} \|a_i\|\|b_i\|,$$

and with high probability,

$$\|A^T B - C^T D\|_2 \leq \frac{\mathcal{O}(1)}{\sqrt{\ell}} \|A\|_F \|B\|_F.$$

Random Selection

Get C and D in time $\mathcal{O}(n(m_1 + m_2)))$.

Outline

Random Projection

- Random Projection. Let $\Pi \in \mathbb{R}^{\ell \times n}$ be some subspace embedding, then

$$\|A^T B - (\Pi A)^T (\Pi B)\|_2 \leq \sqrt{\frac{\mathcal{O}(r)}{\ell}} \|A\|_2 \|B\|_2$$

holds with high probability, where $r = rank(A) + rank(B)$. For fast or sparse subspace embedding, ΠA and ΠB can be computed quickly.

Random Projection

For dense Π, get C and D in time $\mathcal{O}(n\ell(m_1 + m_2))$.
For sparse Π, get C and D in time $\mathcal{O}(nnz(A) + nnz(B))$.

Outline

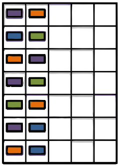
Frequent Directions

FD (Edo Liberty 2013)

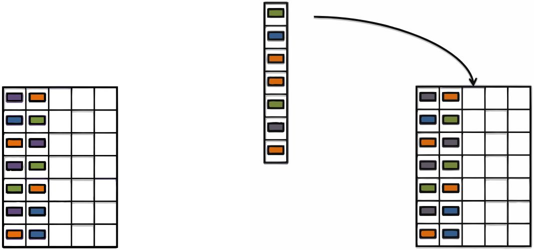
Give $A \in \mathbb{R}^{d \times n}$, FD efficiently maintains a matrix B with only ℓ columns s.t.

$$\|AA^T - BB^T\|_2 \leq 2\|A\|_F^2/\ell,$$

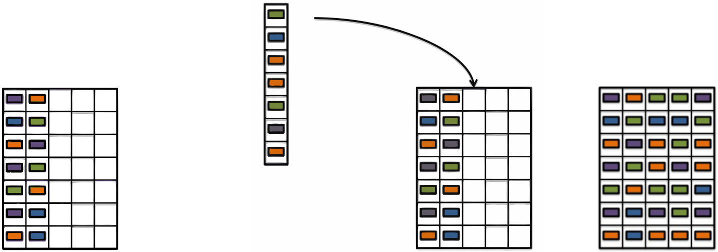
The FD algorithm runs in time $\mathcal{O}(nd\ell)$.



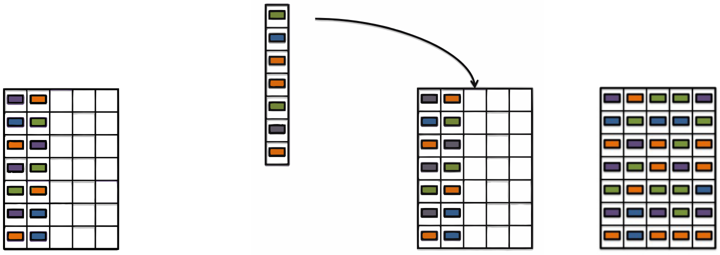
We keep a sketch of ℓ columns.

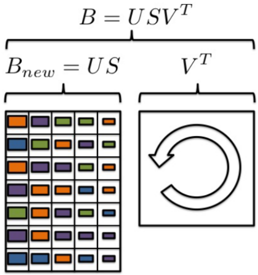


Input vectors are simply stored in empty columns.

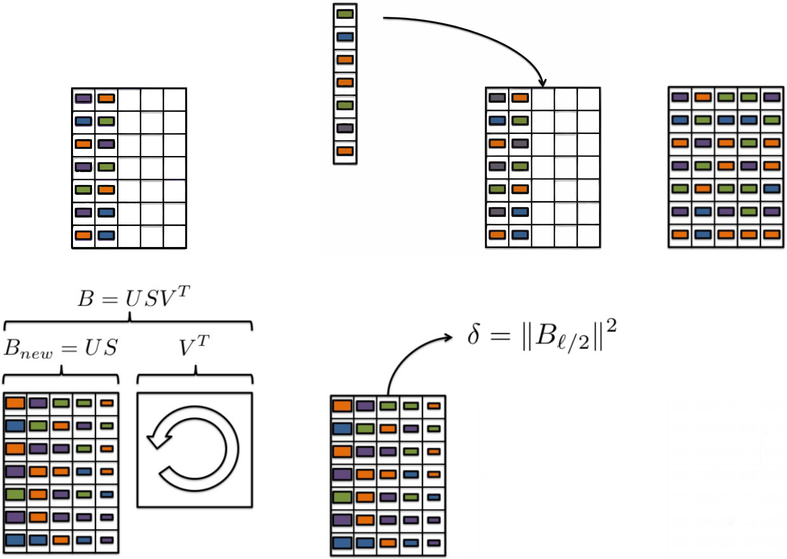


When the sketch is 'full' we need to zero out some columns.

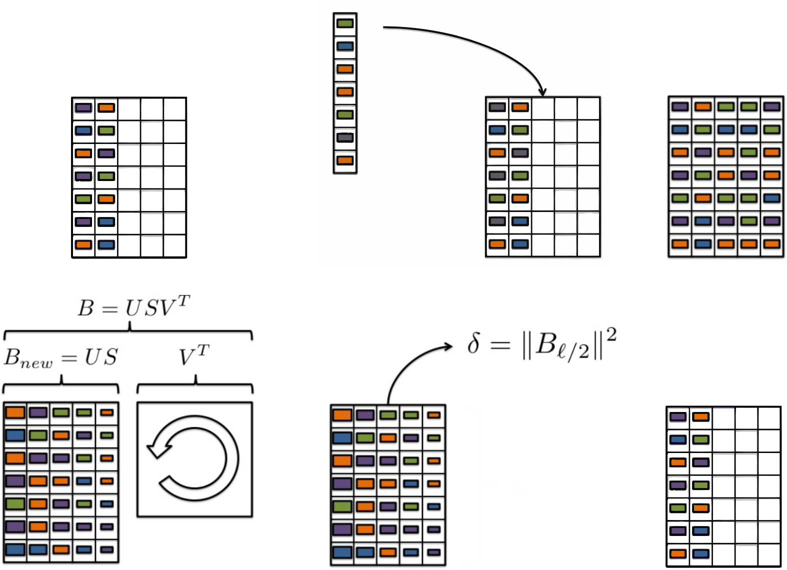




Using the SVD we compute $B = USV^T$ and set $B_{new} = US$.



Let $\sigma = \|B_{\ell/2}\|^2$.


$B = USV^T$
$B_{new} = US$
V^T
$\delta = \|B_{\ell/2}\|^2$

Outline

FD-AMM

Algorithm 2 FD-AMM

Input: ℓ, $A \in R^{n\times m_1}$, $B \in \mathbb{R}^{n\times m_2}$

$G = [A \ B]$

$H = \mathrm{FD}(\ell, G)$

$C = H_{[\ell],1:m_1}$

$D = H_{[\ell],m_1+1:m_1+m_2}$

return C, D

FD-AMM

Algorithm 2 FD-AMM

Input: ℓ, $A \in R^{n\times m_1}$, $B \in \mathbb{R}^{n\times m_2}$

$G = [A\ B]$

$H = \mathrm{FD}(\ell, G)$

$C = H_{[\ell],1:m_1}$

$D = H_{[\ell],m_1+1:m_1+m_2}$

return C, D

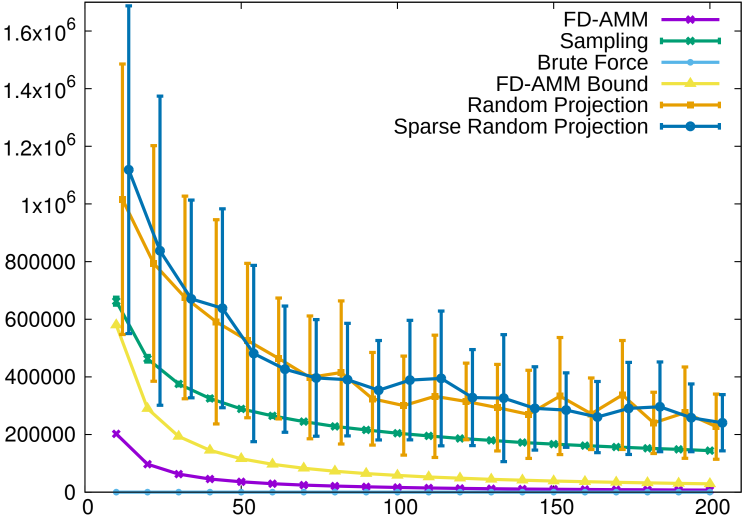
Theorem

If X is the result of applying the FD-AMM algorithm to matrices A, B, and sketch size ℓ, then

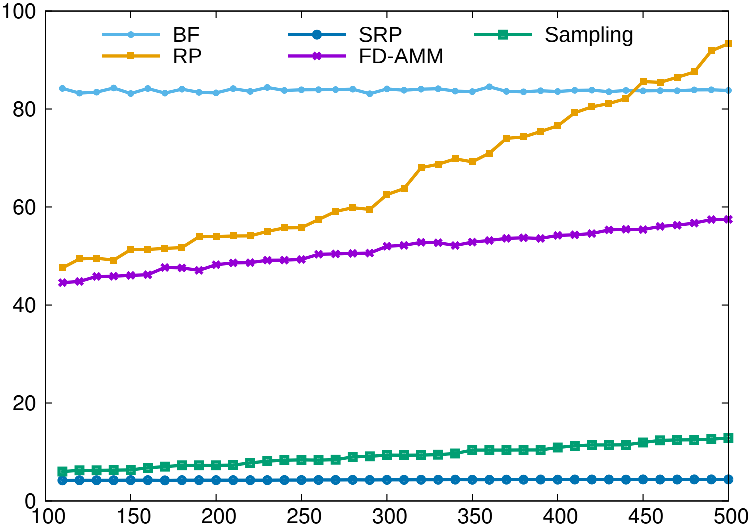
$$\|A^T B - C^T D\|_2 \leq (\|A\|_F^2 + \|B\|_F^2)/\ell.$$

Experiments

$\|A^T B - C^T D\|_2$ as a function of the sketch size ℓ.



Running time in second as a function of ℓ.



Thank You