Matrix Sketching over Sliding Windows

Zhewei Wei[1], Xuancheng Liu[1], Feifei Li[2], Shuo Shang[1]
Xiaoyong Du[1], Ji-Rong Wen[1]

[1] School of Information, Renmin University of China

[2] School of Computing, The University of Utah

Matrix data

- Modern data sets are modeled as large matrices.
- Think of $A \in R^{n \times d}$ as n rows in R^d.

Data	Rows	Columns	d	n
Textual	Documents	Words	$10^5 - 10^7$	$>10^{10}$
Actions	Users	Types	$10^1 - 10^4$	$>10^7$
Visual	Images	Pixels, SIFT	$10^5 - 10^6$	$>10^8$
Audio	Songs, tracks	Frequencies	$10^5 - 10^6$	$>10^8$
Machine Learning	Examples	Features	$10^2 - 10^4$	$>10^6$
Financial	Prices	Items, Stocks	$10^3 - 10^5$	$>10^6$

Singular Value Decomposition (SVD)

$$
\overset{A}{\begin{bmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{bmatrix}}
=
\overset{U}{\begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \cdots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix}}
\times
\overset{\Sigma}{\begin{bmatrix} \delta_1 & 0 & \cdots & 0 \\ 0 & \delta_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_d \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}
\times
\overset{V^T}{\begin{bmatrix} v_{11} & \cdots & v_{d1} \\ \vdots & \cdots & \vdots \\ v_{1d} & \cdots & v_{nd} \end{bmatrix}}
$$

- Principal component analysis (PCA)
- K-means clustering
- Latent semantic indexing (LSI)

SVD & Eigenvalue decomposition

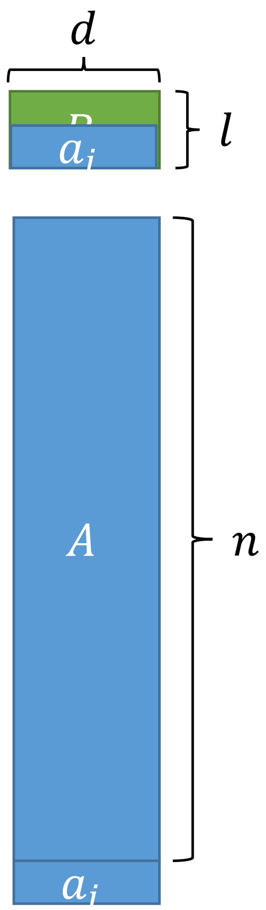
Covariance Matrix
$A^T A$

$$
\overset{A^T}{\begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \cdots & \vdots \\ a_{1d} & \cdots & a_{nd} \end{bmatrix}}
\times
\overset{A}{\begin{bmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{bmatrix}}
$$

$$
= \overset{V}{\begin{bmatrix} v_{11} & \cdots & v_{1d} \\ \vdots & \cdots & \vdots \\ v_{d1} & \cdots & v_{nd} \end{bmatrix}}
\times
\overset{\Sigma^2}{\begin{bmatrix} \delta_1^2 & 0 & \cdots & 0 \\ 0 & \delta_2^2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_d^2 \end{bmatrix}}
\times
\overset{V^T}{\begin{bmatrix} v_{11} & \cdots & v_{d1} \\ \vdots & \cdots & \vdots \\ v_{1d} & \cdots & v_{nd} \end{bmatrix}}
$$

Matrix Sketching

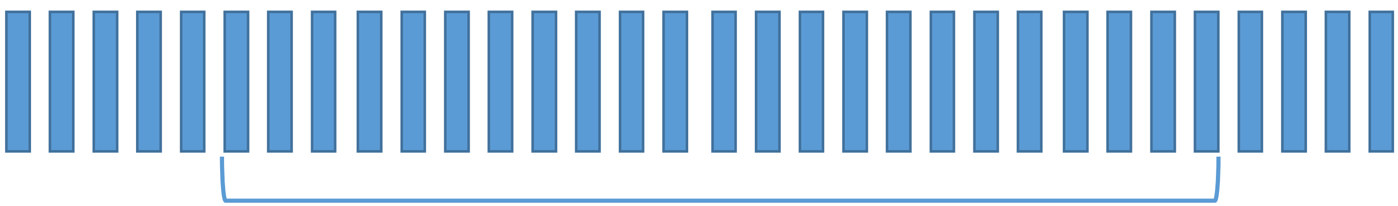
- Computing SVD is slow (and offline).
- Matrix sketching: approximate large matrix $A \in R^{n\times d}$ with $\mathrm{B} \in R^{l\times d}$, $l \ll n$, in an online fashion.
- Row-update stream: each update receives a row.
- Covariance error [Liberty2013, Ghashami2014, Woodruff2016]: $\left\|A^T A - B^T B\right\| / ||A||_F^2 \le \varepsilon$.
- Feature hashing [Weinberger2009], random projection [Papadimitriou2011], …
- Frequent Directions (FD) [Liberty2013]:
  - $\mathrm{B} \in R^{l\times d}$, $l = \frac{1}{\varepsilon}$, s.t. covariance error $\le \varepsilon$.



Matrix Sketching over Sliding Windows

- Each row is associated with a timestamp.
- Maintain B_W for A_W: rows in sliding window W.

  Covariance error: $||A_W^T A_W - B_W^T B_W||/||A_W||_F^2 \leq \varepsilon$
- Sequence-based window: past N rows.



A_W: N rows

- Time-based window: rows in a past time period Δ.

A_W: rows in Δ time units

Motivation 1: Sliding windows vs. unbounded streams

- Sliding window model is a more appropriate model in many real-world applications.

- Particularly so in the areas of data analysis wherein matrix sketching techniques are widely used.

- Applications:
  - Analyzing tweets for the past 24 hours.
  - Sliding window PCA for detecting changes and anomalies [Papadimitriou2006, Qahtan2015].

Motivation 2: Lower bound

- Unbounded stream solution: use $O(d^2)$ space to store $A^T A$.
  - Update: $A^T A \leftarrow A^T A + a_i^T a_i$

Theorem 4.1 An algorithm that returns $A^T A$ for any sequence-based sliding window must use $\Omega(Nd)$ bits space.

- Matrix sketching is necessary for sliding window, even when dimension d is small.
- Matrix sketching over sliding windows requires new techniques.

Three algorithms

- Sampling:
  - Sample a_i w.p. proportional to $||a_i||^2$ [Frieze2004].
  - Priority sampling[Efraimidis2006] + Sliding window top-k.
- LM-FD: Exponential Histogram (Logarithmic method) [Datar2002] + Frequent Directions.
- DI-FD: Dyadic interval techniques [Arasu2004] + Frequent Directions.

Sketches	Update	Space	Window	Interpretable?
Sampling	$\frac{d}{\varepsilon^2}\log\log NR$	$\frac{d}{\varepsilon^2}\log NR$	Sequence & time	Yes
LM-FD	$d\log\varepsilon NR$	$\frac{1}{\varepsilon^2}\log\varepsilon NR$	Sequence & time	No
DI-FD	$\frac{d}{\varepsilon}\log\frac{R}{\varepsilon}$	$\frac{R}{\varepsilon}\log\frac{R}{\varepsilon}$	Sequence	No

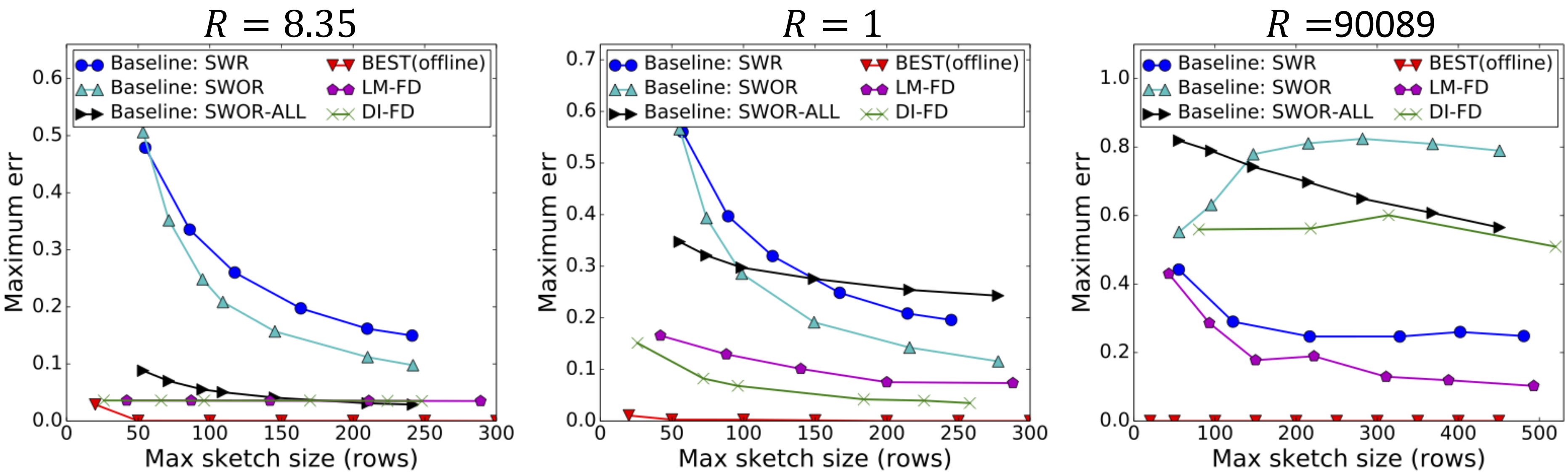
Three algorithms

- Sampling:
  - Sample a_i w.p. proportional to $||a_i||^2$ [Frieze2004].
  - Priority sampling[Efraimidis2006] + Sliding window top-k.
- LM-FD: Exponential Histogram (Logarithmic method) [Datar2002] + Frequent Directions.
- DI-FD: Dyadic interval techniques [Arasu2004] + Frequent Directions.

Sketches	Update	Space	Window	Interpretable?
Sampling	Slow	Large	Sequence & time	Yes
LM-FD	Fast	Small	Sequence & time	No
DI-FD	Slow	Best for small R	Sequence	No

- Interpretable: rows of the sketch B come from A.
- R: ratio between maximum squared norm and minimum squared norms.

Experiments: space vs. error

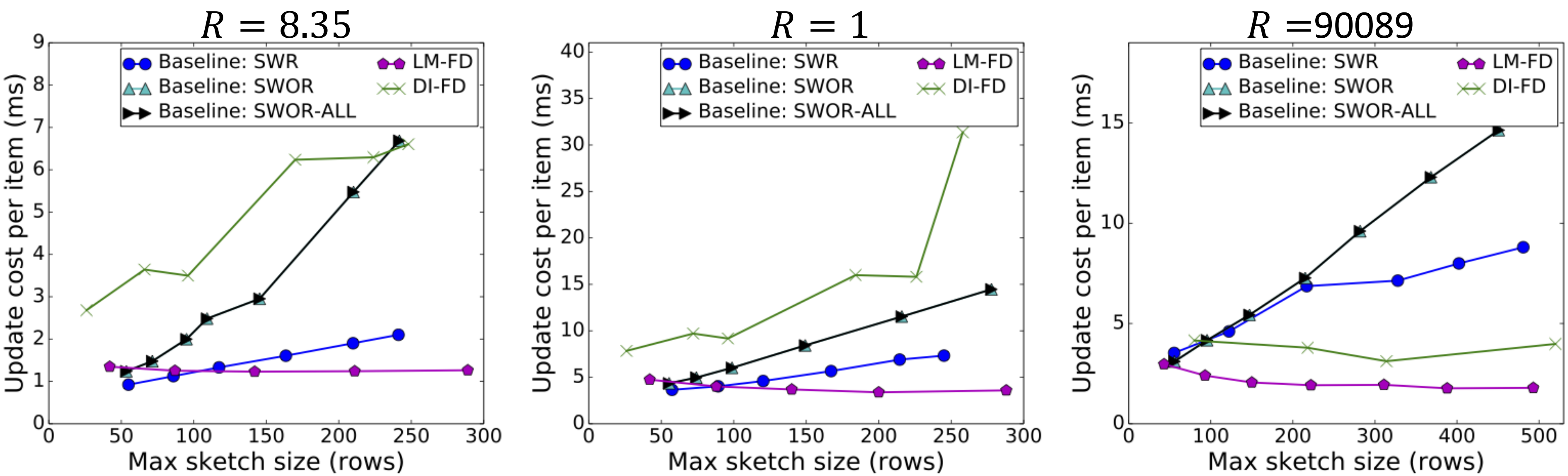


(a) SYNTHETIC (b) BIBD (c) PAMAP

Sketches	Update	Space	Window	Interpretable?
Sampling	Slow	Large	Sequence & time	Yes
LM-FD	Fast	Small	Sequence & time	No
DI-FD	Slow	Best for small R	Sequence	No

- Interpretable: rows of the sketch B come from A.
- R: ratio between maximum squared norm and minimum squared norms.

Experiments: time vs. space



(a) SYNTHETIC (b) BIBD (c) PAMAP

Sketches	Update	Space	Window	Interpretable?
Sampling	Slow	Large	Sequence & time	Yes
LM-FD	Fast	Small	Sequence & time	No
DI-FD	Slow	Best for small R	Sequence	No

- Interpretable: rows of the sketch B come from A.
- R: ratio between maximum squared norm and minimum squared norms.

Conclusions

- First attempt to tackle the sliding window matrix sketching problem.
- Lower bounds show that the sliding window model is different from unbounded streaming model for the matrix sketching problem.
- Propose algorithms for both time-based and sequence-based windows with theoretical guarantee and experimental evaluation.

Thanks!