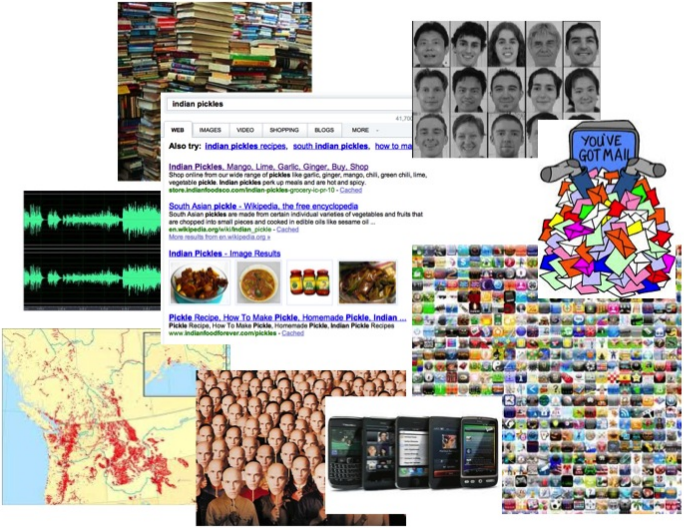
Simple and Deterministic Matrix Sketches

Edo Liberty

Data Matrices

Often our data is represented by a matrix.



Data Matrices

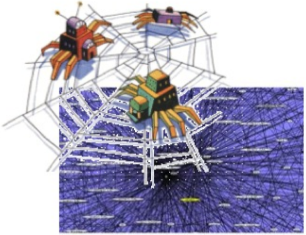
But our data matrix is typically too large to work with on a single machine.

Data	**Columns**	**Rows**	d	n	**sparse**
Textual	Documents	Words	10^5 - 10^7	$> 10^{12}$	yes
Actions	Users	Types	10^1 - 10^4	$> 10^8$	yes
Visual	Images	Pixels, SIFT	10^6 - 10^7	$> 10^9$	no
Audio	Songs, tracks	Frequencies	10^6 - 10^7	$> 10^9$	no
Machine Learning	Examples	Features	10^2 - 10^4	$> 10^5$	no
Financial	Prices	Items, Stocks	10^3 -10^5	$> 10^6$	no

We think of $A \in \mathbb{R}^{d \times n}$ as n column vectors in $\mathbb{R}^d$ and typically $n \gg d$.

Streaming Matrices

Sometimes, we cannot store the entire matrix at all.



YAHOO!
Mail

flickr

Streaming Matrices

Example: can we compute the covariance matrix from the a stream? (enough for PCA for example).

Streaming Matrices

Example: can we compute the covariance matrix from the a stream? (enough for PCA for example).

$$AA^T = \sum_{i=1}^{n} A_i A_i^T$$

Naïve solution

Compute AA^T in time $O(nd^2)$ and space $O(d^2)$.

Think about $1Mp$ images, $d = 10^6$. This solution requires 10^{12} operations per update and $1T$ space.

Matrix Approximation

Matrix sketching or approximation

Efficiently compute a **concisely representable** matrix B such that

$$B \approx A \text{ or } BB^T \approx AA^T$$

Working with B instead of A is often "good enough".

- Dimension reduction
- Signal denoising
- Classification
- Regression
- Clustering
- Approximate matrix multiplication
- Reconstruction
- Recommendation
- ...

Matrix Approximation

Column subset selection algorithms			
Paper	**Space**	**Time**	**Bound**
FKV04	$O(k^4/\varepsilon^6 \max(k^4, \varepsilon^{-2}))$	$O(k^5/\varepsilon^6 \max(k^4, \varepsilon^{-2}))$	P, εA
DV06	$\#C = O(k/\varepsilon + k^2 \log k)$ $O(n(k/\varepsilon + k^2 \log k))$	$O(\text{nnz}(A)(k/\varepsilon + k^2 \log k)+$ $(n+d)(k^2/\varepsilon^2 + k^3 \log(k/\varepsilon) + k^4 \log^2 k))$	P, εR
DKM06 "LinearTimeSVD"	$\#C = O(1/\varepsilon^2)$ $O((n + 1/\varepsilon^2)/\varepsilon^4)$	$O((n + 1/\varepsilon^2)/\varepsilon^4 + \text{nnz}(A))$	P, εL_2
	$\#C = O(k/\varepsilon^2)$ $O((k/\varepsilon^2)(n + k/\varepsilon^2))$	$O((k/\varepsilon^2)^2(n + k/\varepsilon^2) + \text{nnz}(A))$	P, εA
DKM06 "ConstantTimeSVD"	$\#C{+}R = O(1/\varepsilon^4)$ $O(1/\varepsilon^{12} + nk/\varepsilon^4)$	$O((1/\varepsilon^{12} + nk/\varepsilon^4 + \text{nnz}(A))$	P, εL_2
	$\#C{+}R = O(k^2/\varepsilon^4)$ $O(k^6/\varepsilon^{12} + nk^3/\varepsilon^4)$	$O(k^6/\varepsilon^{12} + nk^3/\varepsilon^4 + \text{nnz}(A))$	P, εA
DMM08 "CUR"	$\#C = O(k^2/\varepsilon^2)$ $\#R = O(k^4/\varepsilon^6)$	$O(nd^2)$	C, εR
MD09 "ColumnSelect"	$\#C = O(k \log k/\varepsilon^2)$ $O(nk \log k/\varepsilon^2)$	$O(nd^2)$	$P_{O(k \log k/\varepsilon^2)}$, εR
BDM11	$\#C = 2k/\varepsilon(1 + o(1))$	$O((ndk + dk^3)\varepsilon^{-2/3})$	$P_{2k/\varepsilon(1+o(1))}$, εR

[Relative Errors for Deterministic Low-Rank Matrix Approximations, Ghashami, Phillips 2013]

Sparsification and entry sampling			
Paper	**Space**	**Time**	**Bound**
AM07	$\rho n/\varepsilon^2 + n \cdot \text{polylog}(n)$	nnz $\rho n/\varepsilon^2$ + nnz $n \cdot \text{polylog}(n)$	$\|A - B\|_2 \le \varepsilon\|A\|_2$
AHK06	$(\widetilde{\text{nnz}} \cdot n/\varepsilon^2)^{1/2}$	$\text{nnz}(\widetilde{\text{nnz}} \cdot n/\varepsilon^2)^{1/2}$	$\|A - B\|_2 \le \varepsilon\|A\|_2$
DZ11	$\rho n \log(n)/\varepsilon^2$	nnz $\rho n \log(n)/\varepsilon^2$	$\|A - B\|_2 \le \varepsilon\|A\|_2$
AKL13	$\tilde{n}\ \rho \log(n)/\varepsilon^2$ + $(\rho \log(n)\ \widetilde{\text{nnz}}/\varepsilon^2)^{1/2}$	nnz	$\|A - B\|_2 \le \varepsilon\|A\|_2$

[Near-optimal Distributions for Data Matrix Sampling, Achlioptas, Karnin, Liberty, 2013]

Matrix Approximation

Linear subspace embedding sketches			
Paper	**Space**	**Time**	**Bound**
DKM06 LinearTimeSVD	$\#\mathrm{R} = O(1/\varepsilon^2)$ $O((d + 1/\varepsilon^2)/\varepsilon^4)$	$O((d + 1/\varepsilon^2)/\varepsilon^4 + \mathrm{nnz}(A))$	P, $\varepsilon \mathrm{L}_2$
	$\#\mathrm{R} = O(k/\varepsilon^2)$ $O((k/\varepsilon^2)^2(d + k/\varepsilon^2))$	$O((k/\varepsilon^2)^2(d + k/\varepsilon^2) + \mathrm{nnz}(A))$	P, εA
Sar06 turnstile	$\#\mathrm{R} = O(k/\varepsilon + k \log k)$ $O(d(k/\varepsilon + k \log k))$	$O(\mathrm{nnz}(A)(k/\varepsilon + k \log k) + d(k/\varepsilon + k \log k)^2))$	$\mathrm{P}_{O(k/\varepsilon + k \log k)}$, εR
CW09	$\#\mathrm{R} = O(k/\varepsilon)$	$O(nd^2 + (ndk/\varepsilon))$	$\mathrm{P}_{O(k/\varepsilon)}$, εR
CW09	$O((n + d)(k/\varepsilon))$	$O(nd^2 + (ndk/\varepsilon))$	C, εR
CW09	$O((k/\varepsilon^2)(n + d/\varepsilon^2))$	$O(n(k/\varepsilon^2)^2 + nd(k/\varepsilon^2) + nd^2)$	C, εR

Deterministic sketching algorithms			
Paper	**Space**	**Time**	**Bound**
FSS13	$O((k/\varepsilon) \log n)$	$n((k/\varepsilon) \log n)^{O(1)}$	$\mathrm{P}_{2\lceil k/\varepsilon \rceil}$, εR
Lib13	$\#\mathrm{R} = O(\rho/\varepsilon)$ $O(d\rho/\varepsilon)$	$O(nd\rho/\varepsilon)$	$\mathrm{P}_{O(\rho/\varepsilon)}$, $\varepsilon \mathrm{L}_2$
GP13	$\#\mathrm{R} = \lceil k/\varepsilon + k \rceil$ $O(dk/\varepsilon)$	$O(ndk/\varepsilon)$	P, εR

[Relative Errors for Deterministic Low-Rank Matrix Approximations, Ghashami, Phillips 2013]

Frequent Directions

Goal:

Efficiently maintain a matrix B with only $\ell = 2/\varepsilon$ columns s.t.

$$\|AA^T - BB^T\|_2 \leq \varepsilon \|A\|_f^2$$

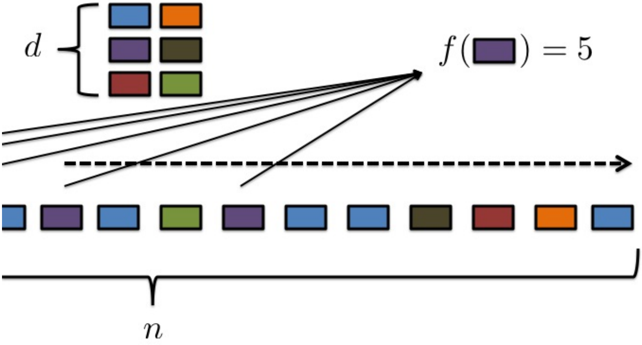
Intuition:

Extend Frequent-items

[Finding repeated elements, Misra, Gries, 1982.]
[Frequency estimation of internet packet streams with limited space, Demaine, Lopez-Ortiz, Munro, 2002]
[A simple algorithm for finding frequent elements in streams and bags, Karp, Shenker, Papadimitriou, 2003]
[Efficient Computation of Frequent and Top-k Elements in Data Streams, Metwally, Agrawal, Abbadi, 2006]

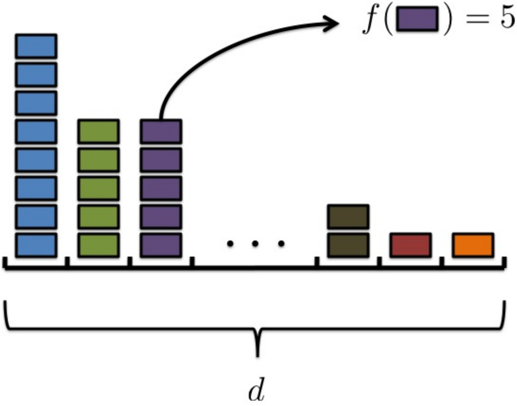
(An algorithm so good it was invented 4 times.)

Frequent Items



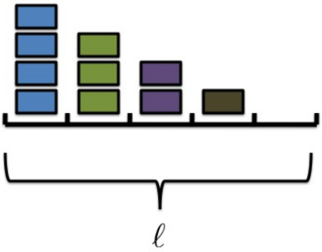
Obtain the frequency $f(i)$ of each item in the stream of items

Frequent Items



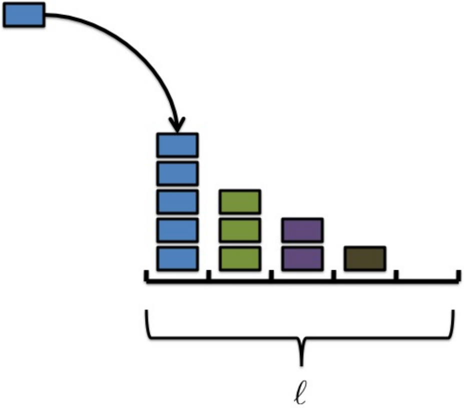
With d counters it's easy but not good enough (IP addresses, queries....)

Frequent Items



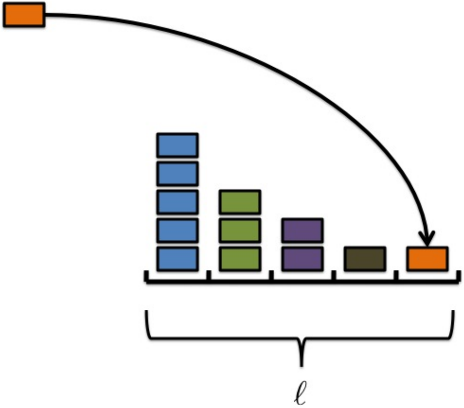
(Misra-Gries) Lets keep **less than** a fixed number of counters ℓ.

Frequent Items



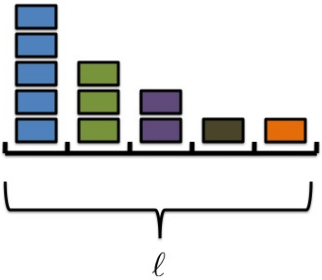
If an item has a counter we add 1 to that counter.

Frequent Items



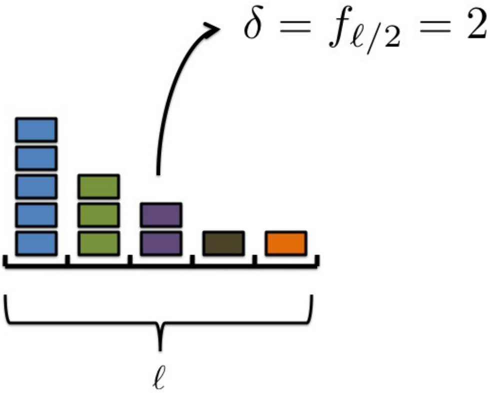
Otherwise, we create a new counter for it and set it to 1

Frequent Items



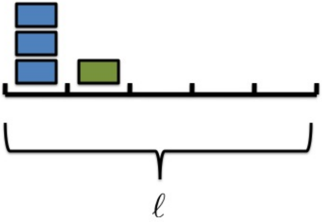
But now we do not have less than ℓ counters.

Frequent Items



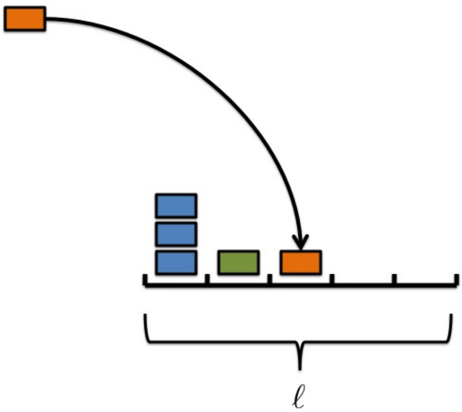
Let δ be the median counter value at time t

Frequent Items



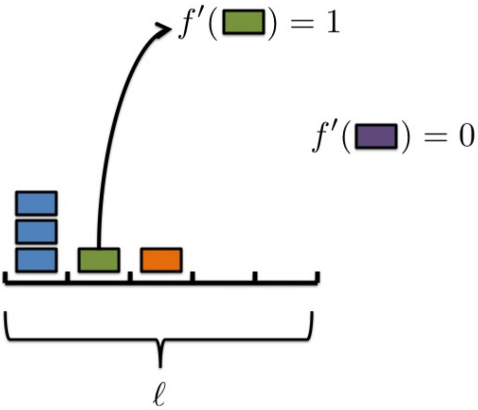
Decrease all counters by δ (or set to zero if less than δ)

Frequent Items



And continue...

Frequent Items



The approximated counts are f'

Frequent Items

- We increase the count by only 1 for each item appearance.

$$f'(i) \leq f(i)$$

- Because we decrease each counter by at most δ_t at time t

$$f'(i) \geq f(i) - \sum_t \delta_t$$

- Calculating the total approximated frequencies:

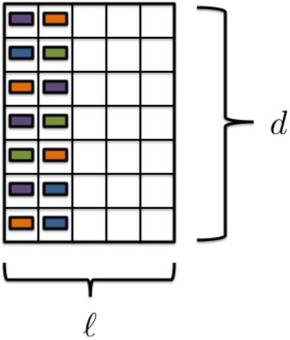
$$0 \leq \sum_i f'(i) \leq \sum_t 1 - (\ell/2) \cdot \delta_t = n - (\ell/2) \cdot \sum_t \delta_t$$

$$\sum_t \delta_t \leq 2n/\ell$$

- Setting $\ell = 2/\varepsilon$ yields

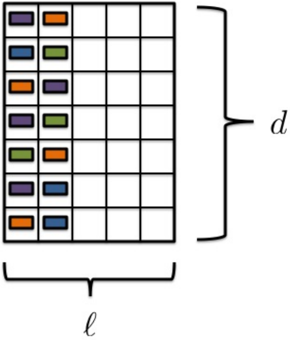
$$|f(i) - f'(i)| \leq \varepsilon n$$

Frequent Directions



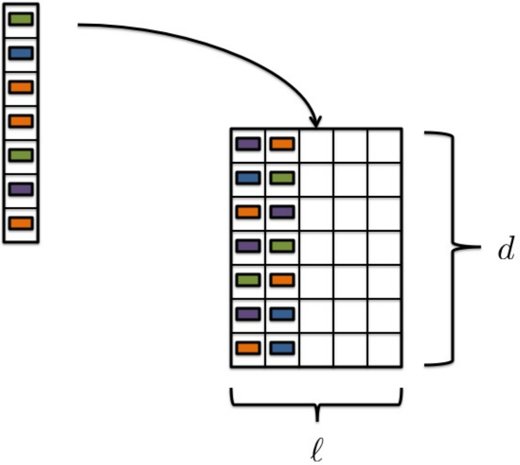
We keep a sketch of at most ℓ columns

Frequent Directions



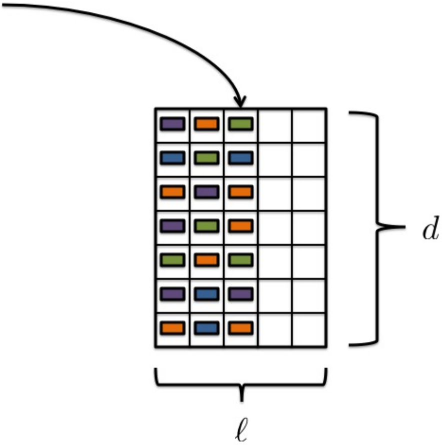
We maintain the invariant that some columns are empty (zero valued)

Frequent Directions



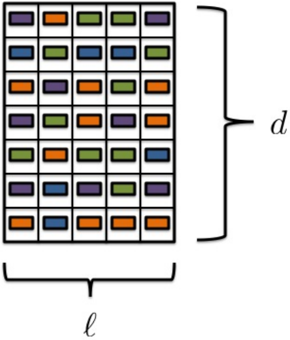
Input vectors are simply stored in empty columns

Frequent Directions



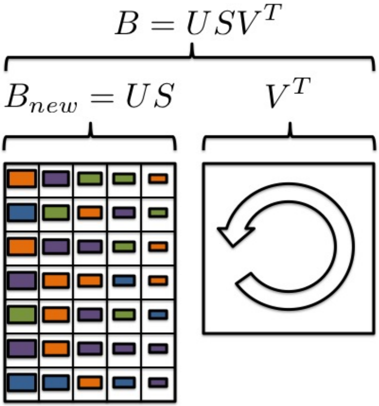
Input vectors are simply stored in empty columns

Frequent Directions



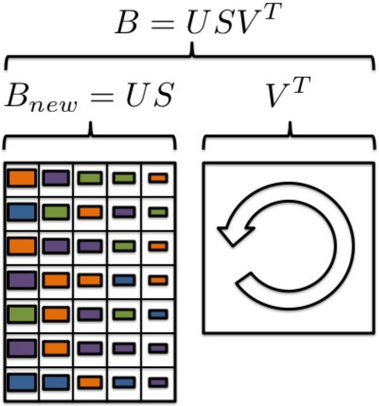
When the sketch is 'full' we need to zero out some columns...

Frequent Directions



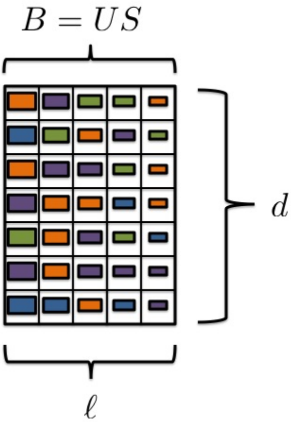
Using the SVD we compute $B = USV^T$ and set $B_{new} = US$

Frequent Directions



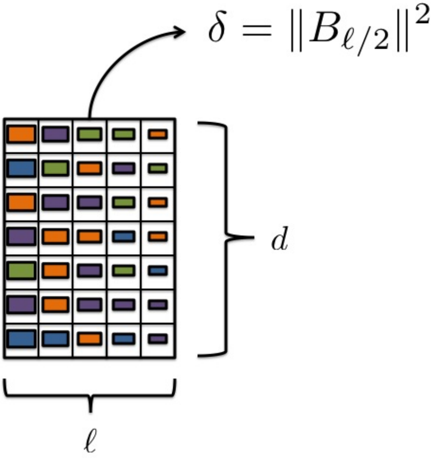
Note that $BB^T = B_{new}B_{new}^T$ so we don't "lose" anything

Frequent Directions



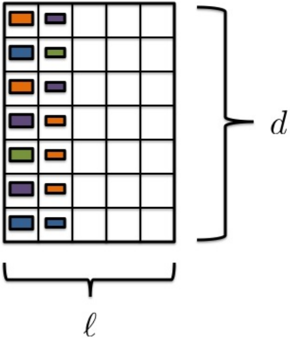
The columns of B are now orthogonal and in decreasing magnitude order

Frequent Directions



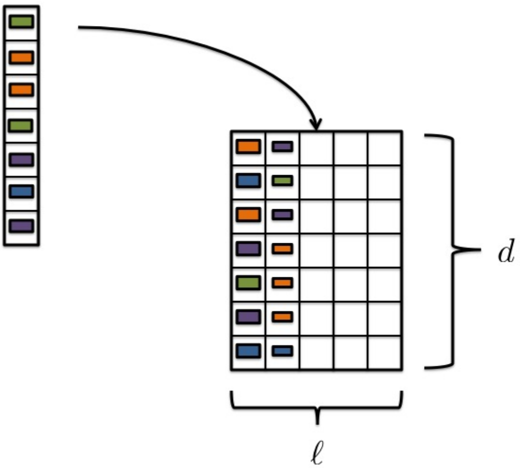
Let $\delta = \|B_{\ell/2}\|^2$

Frequent Directions



Reduce column ℓ_2^2-norms by δ (or nullify if less than δ)

Frequent Directions



Start aggregating columns again...

Frequent Directions

Input: $\ell, \;\; A \in \mathbb{R}^{d \times n}$
$B \leftarrow$ all zeros matrix $\in \mathbb{R}^{d \times \ell}$
for $i \in [n]$ **do**
  Insert A_i into a zero valued column of B
  if B has no zero valued colums **then**
    $[U, \Sigma, V] \leftarrow SVD(B)$
    $\delta \leftarrow \sigma_{\ell/2}^2$
    $\check{\Sigma} \leftarrow \sqrt{\max(\Sigma^2 - I_\ell \delta, 0)}$
    $B \leftarrow U\check{\Sigma}$ # At least half the columns of B are zero.
Return: B

Bounding the error

We first bound $\|AA^T - BB^T\|$

$$\begin{aligned}
\sup_{\|x\|=1} \|xA\|^2 - \|xB\|^2 &= \sup_{\|x\|=1} \sum_{t=1}^{n} [\langle x, A_t \rangle^2 + \|xB^{t-1}\|^2 - \|xB^t\|^2] \\
&= \sup_{\|x\|=1} \sum_{t=1}^{n} [\|xC^t\|^2 - \|xB^t\|^2] \\
&\leq \sum_{t=1}^{n} \|{C^t}^T C^t - {B^t}^T B^t\| \cdot \|x\|^2 \\
&= \sum_{t=1}^{n} \delta_t
\end{aligned}$$

Which gives:

$$\|AA^T - BB^T\| \leq \sum_{t=1}^{n} \delta_t$$

Bounding the error

We compute the Frobenius norm of the final sketch.

$$
\begin{aligned}
0 \le \|B\|_f^2 &= \sum_{t=1}^{n} [\|B^t\|_f^2 - \|B^{t-1}\|_f^2] \\
&= \sum_{t=1}^{n} [(\|C^t\|_f^2 - \|B^{t-1}\|_f^2) - (\|C^t\|_f^2 - \|B^t\|_f^2)] \\
&= \sum_{t=1}^{n} \|A_t\|^2 - tr({C^t}^T C^t - {B^t}^T B^t) \\
&\le \|A\|_f^2 - (\ell/2) \sum_{t=1}^{n} \delta_t
\end{aligned}
$$

Which gives:

$$
\sum_{t=1}^{n} \delta_t \le 2\|A\|_f^2/\ell
$$

Bounding the error

We saw that:

$$\|AA^T - BB^T\| \leq \sum_{t=1}^{n} \delta_t$$

and that:

$$\sum_{t=1}^{n} \delta_t \leq 2\|A\|_f^2/\ell$$

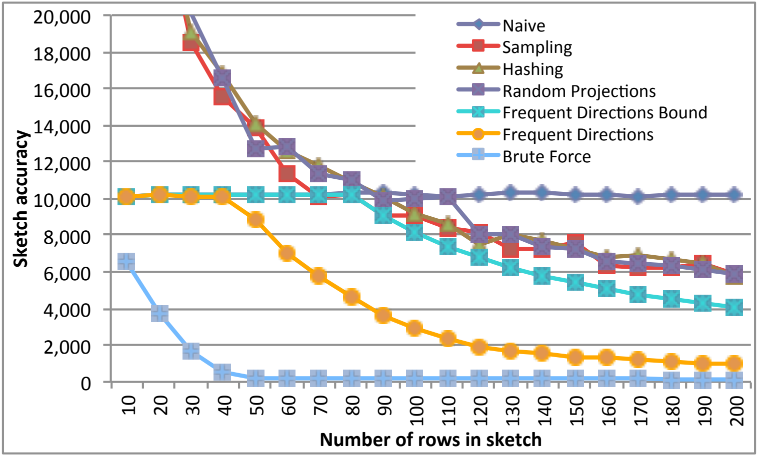
Setting $\ell = 2/\varepsilon$ yields

$$\|AA^T - BB^T\| \leq \varepsilon\|A\|_f^2 \ .$$

The two proofs are (maybe unsurprisingly) very similar...

Experiments

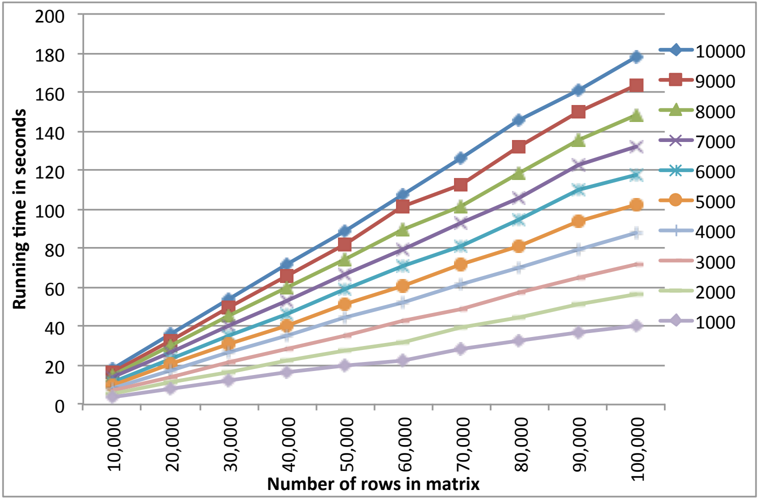
$\|AA^T - BB^T\|$ as a function of the sketch size ℓ



Synthetic input matrix with linearly decaying singular values.

Experiments

Running time in second as a function of n (x-axis) and d (y-axis)



The running time scales linearly in n, d and ℓ as expected.

Thanks