Frequent *

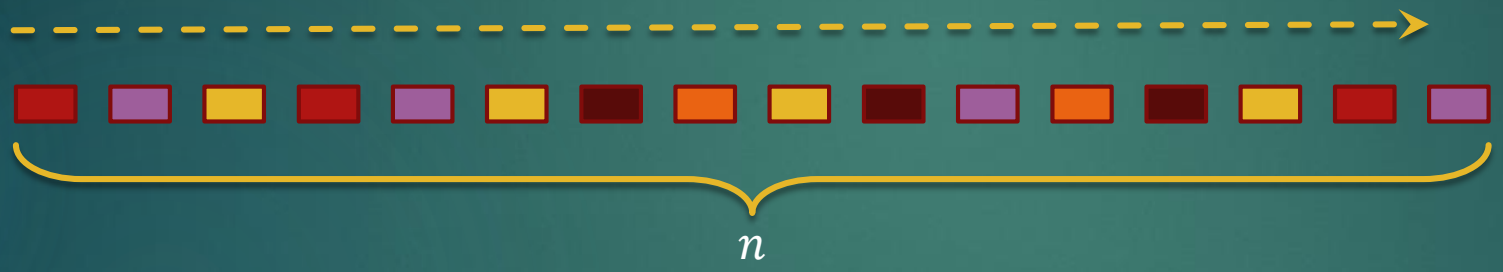
STREAMING ALGORITHMS FOR APPROXIMATIONS

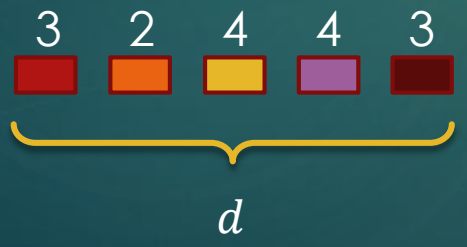
Reminder

- Please start testing your code on Elephant
- Start thinking about your final project
  - Teams
  - Topics
  - Proposals due before fall break
- Review material
  - Mid-term
  - Review class

Frequent Items

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





$f(\square) = 4$

What if d is very large?

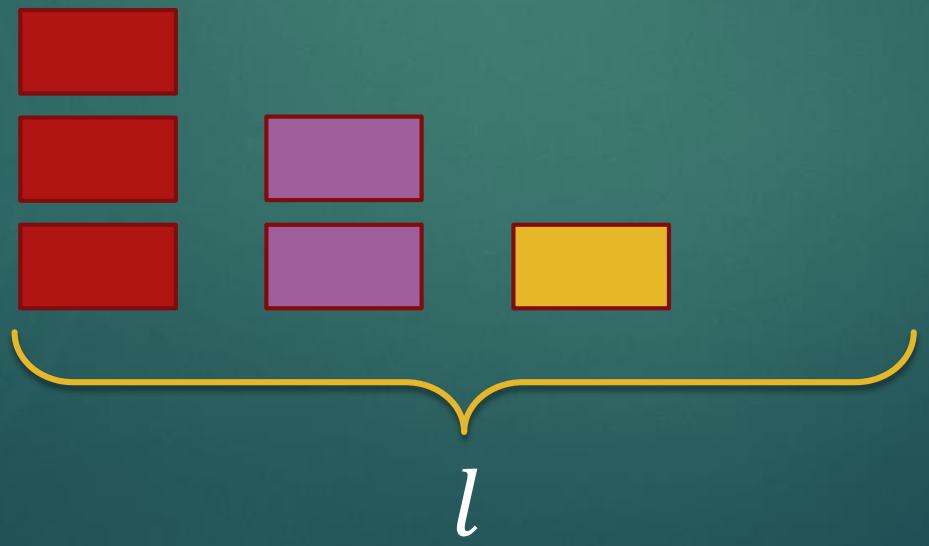
Think

EXACT VS. APPROXIMATIONS

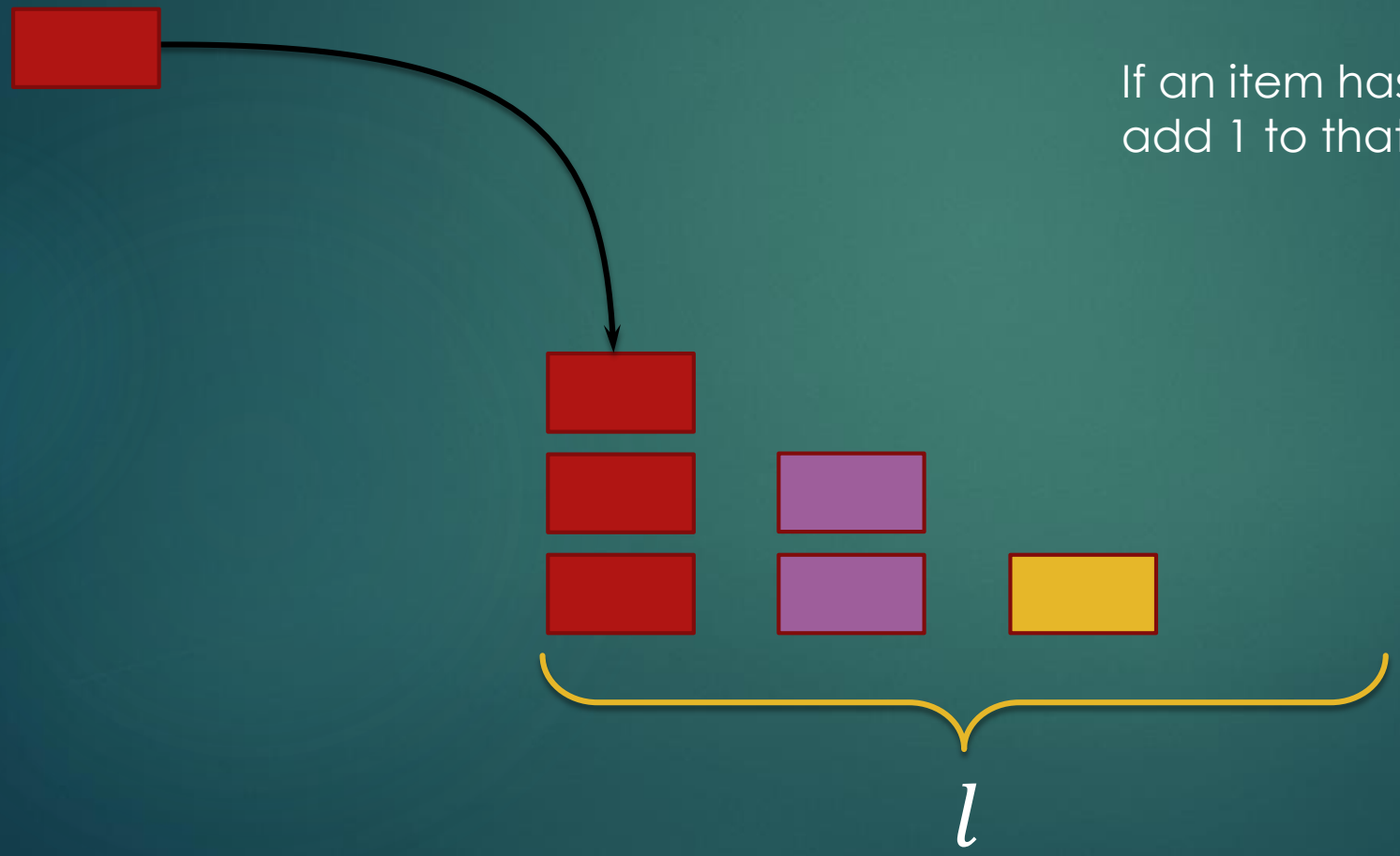
Frequent Items

Lets maintain less than $l < d$ counters (Misra-Gries)

With at least 1 empty counter

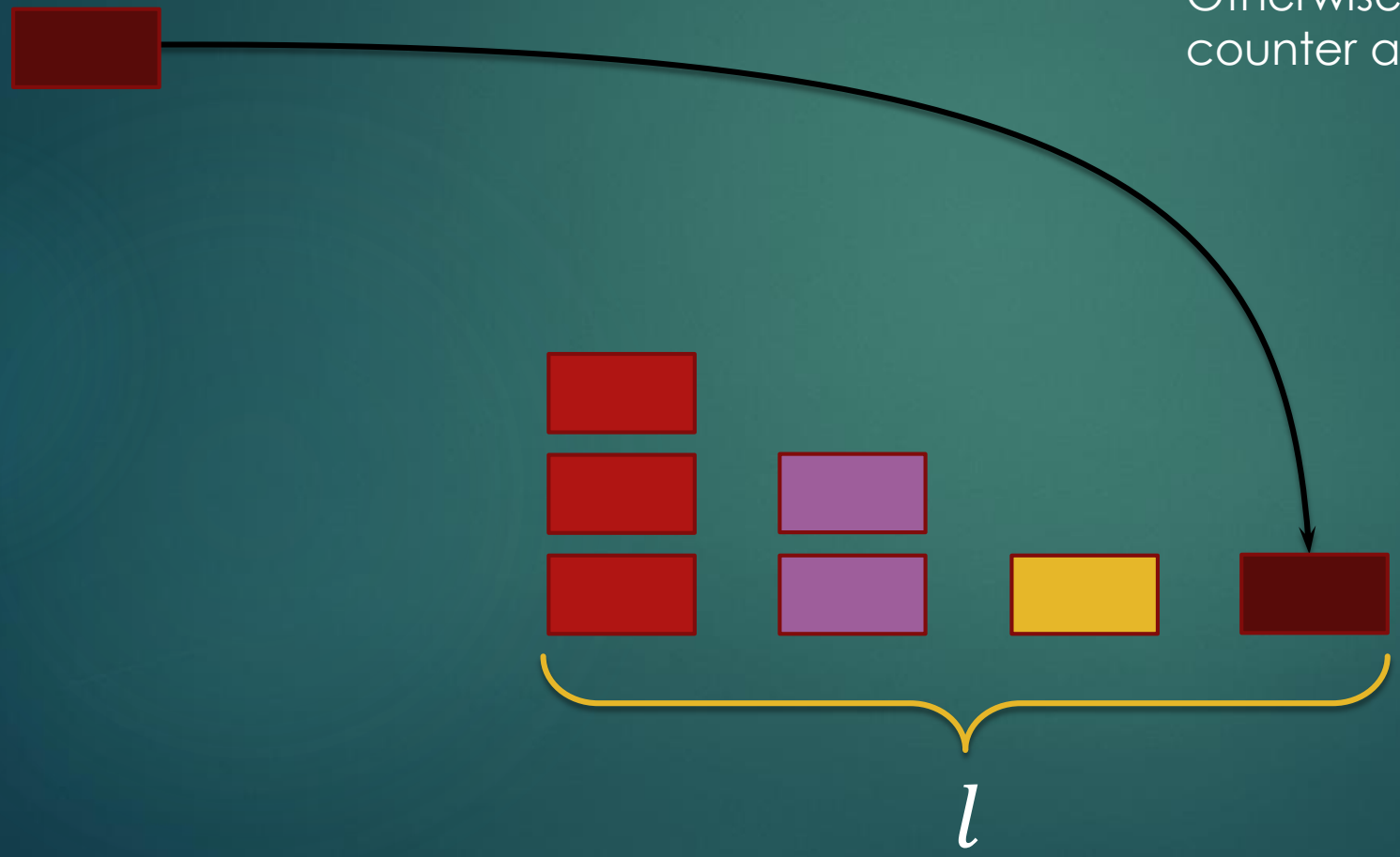


Frequent Items



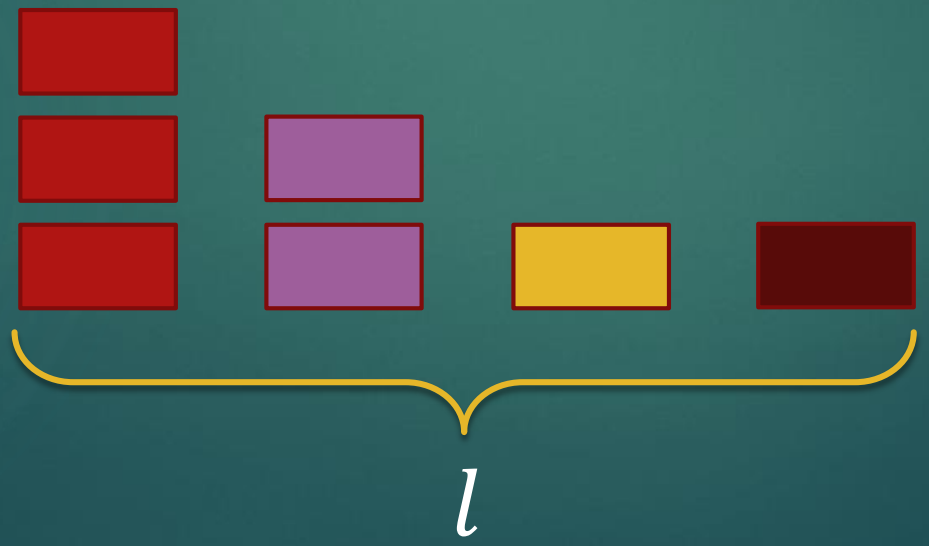
If an item has a counter,
add 1 to that counter

Frequent Items



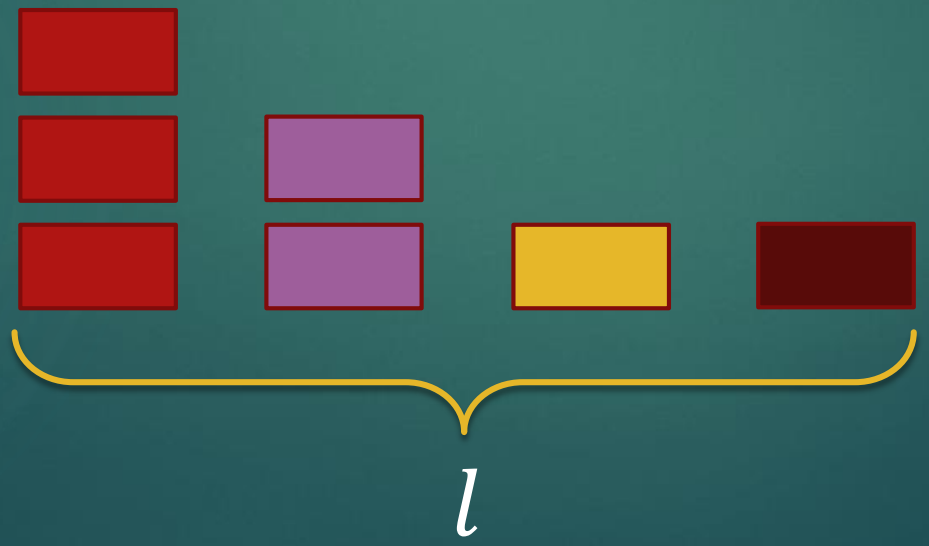
Frequent Items

But now we don't have less than l counters



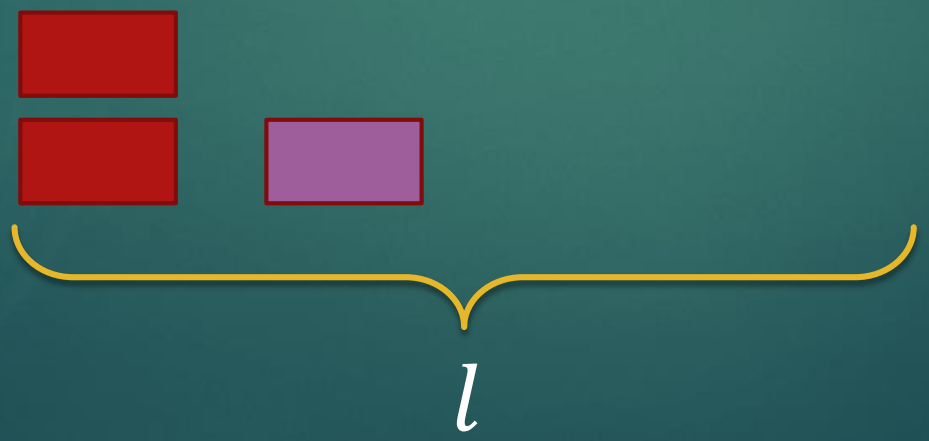
Frequent Items

Let δ be the median counter value at time t

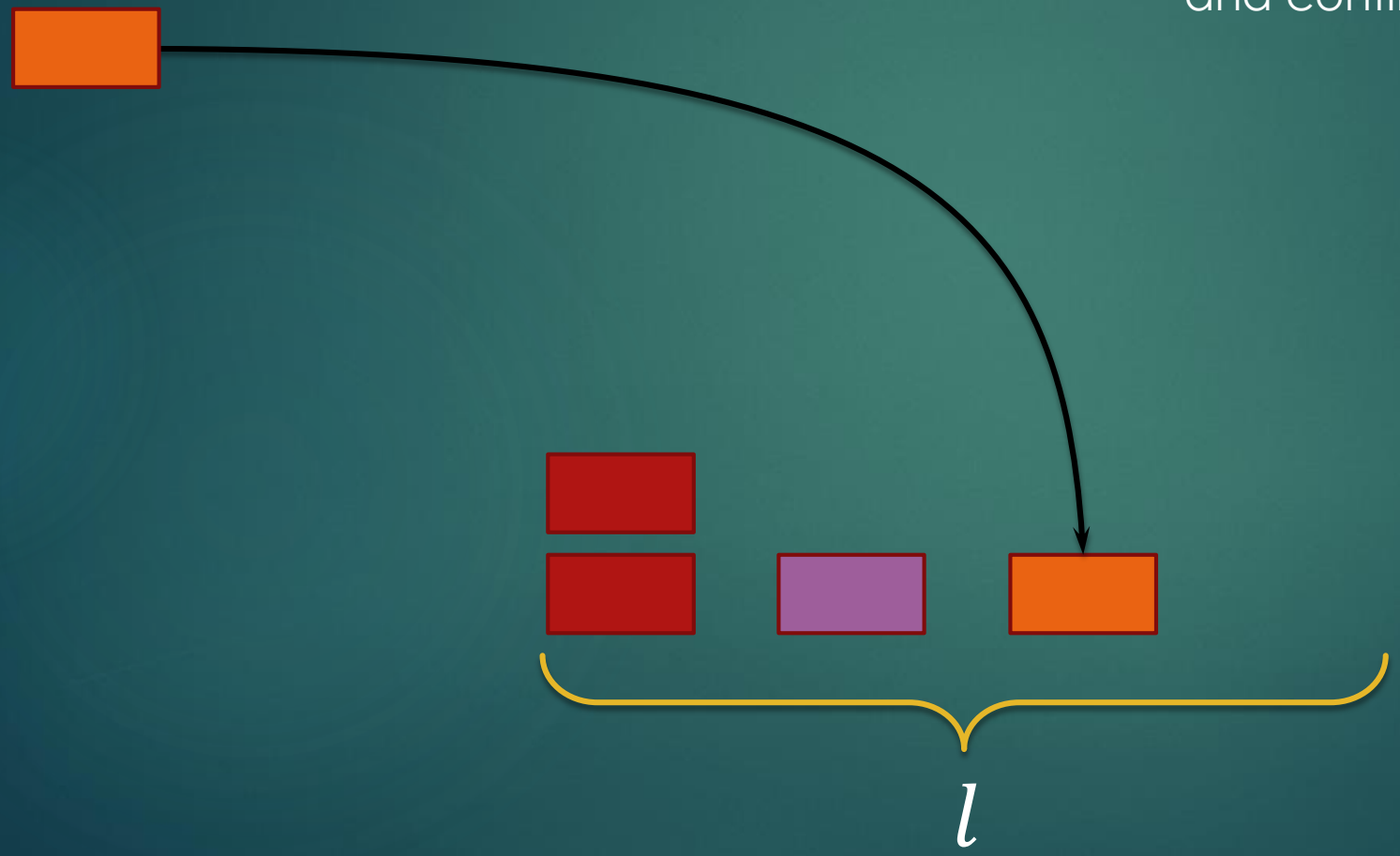


Frequent Items

Decrease all counters by δ

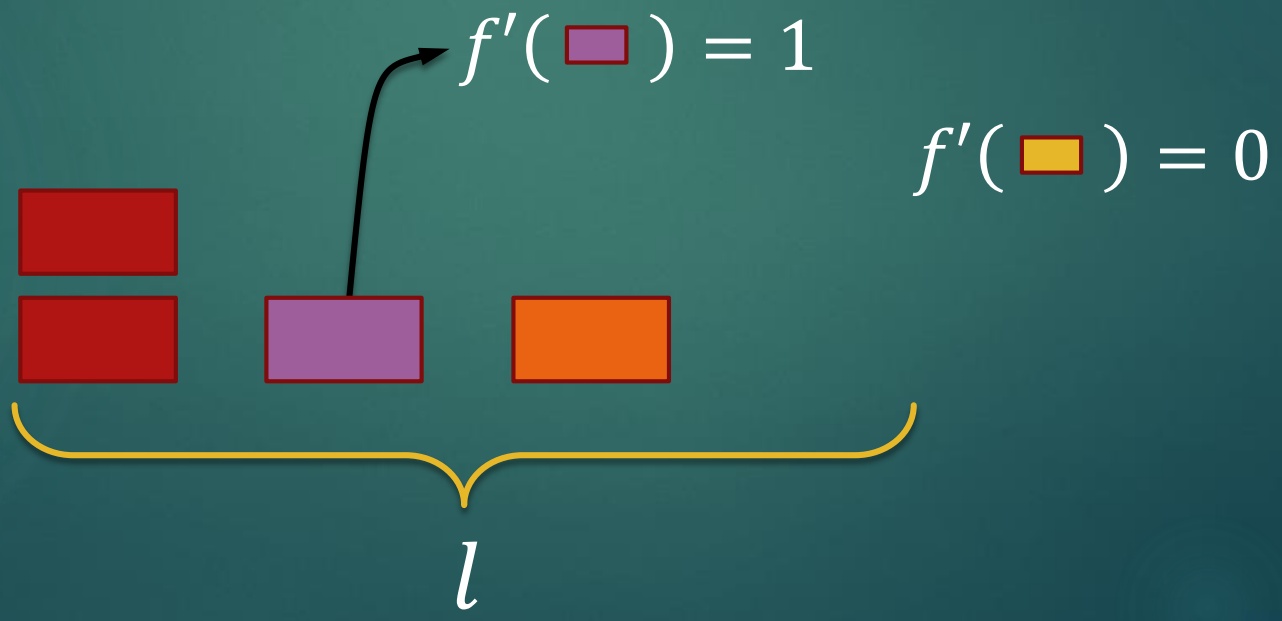


Frequent Items



Frequent Items

The approximated counts are f'



Frequent Items

- We increase the count by only 1 for each item appearance
- Because we decrease each counter by at most δ_t at time t
- Calculating the total approximate frequencies:
- Setting $l = 2/\epsilon$ gives us

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 approximate frequencies:

$$0 \leq \sum_i f'(i) \leq \sum_t 1 - \frac{l}{2}\delta_t = n - \frac{l}{2}\sum_t \delta_t$$

$$\sum_t \delta_t \leq \frac{2n}{l}$$

- Setting $l = 2/\epsilon$ gives us

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

What about higher dimensions?

Matrix Sketching

- Data is usually represented as a matrix
- For most Big Data applications, this matrix is too large for one machine
- In many cases, the matrix is too large to even fit in distributed memory
- Need to optimize for data access
- Streaming algorithm
  - Generate approximation by accessing (streaming) data once

Matrix Sketching

- Efficiently compute a concisely representable matrix B such that

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

- Working with B is often "good enough"
  - Dimension reduction
  - Classification
  - Regression
  - Matrix Multiplication (approximate)
  - Recommendation systems

Frequent Directions (Edo Liberty '13)

- Efficiently maintain a matrix B with only $l = 2/\epsilon$ columns such that,

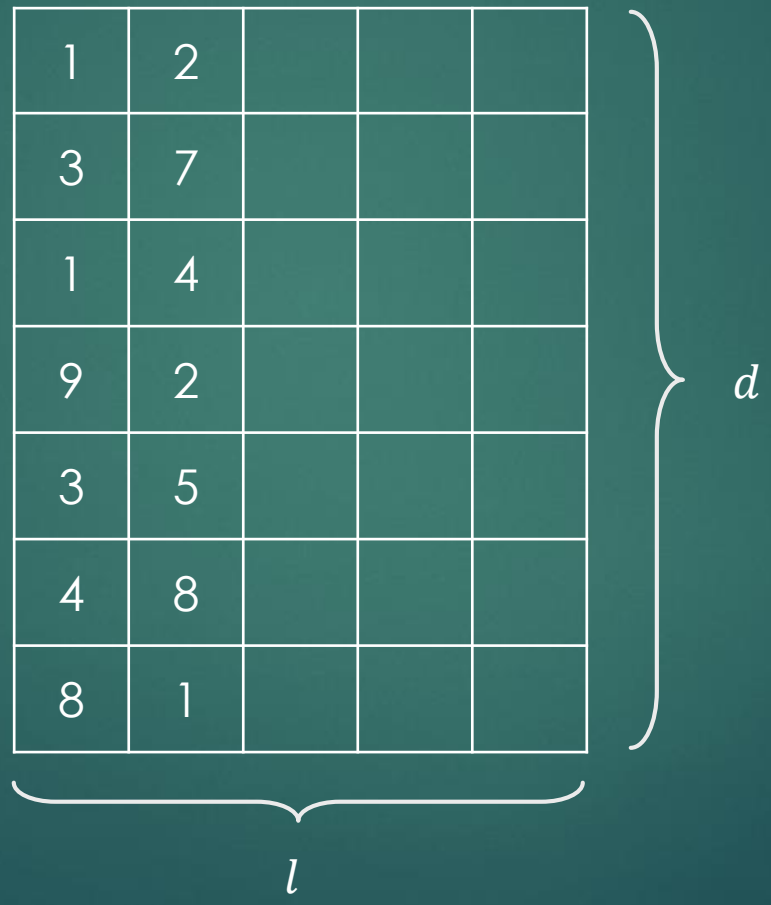
$$\|AA^T - BB^T\|_2 \leq \epsilon \|A\|_f^2$$

- Intuitive approach → extend frequent items
  - How to estimate the frequency of (frequent) items in a streaming fashion?

Frequent Directions (Liberty 2013)

maintain a sketch of at most l columns

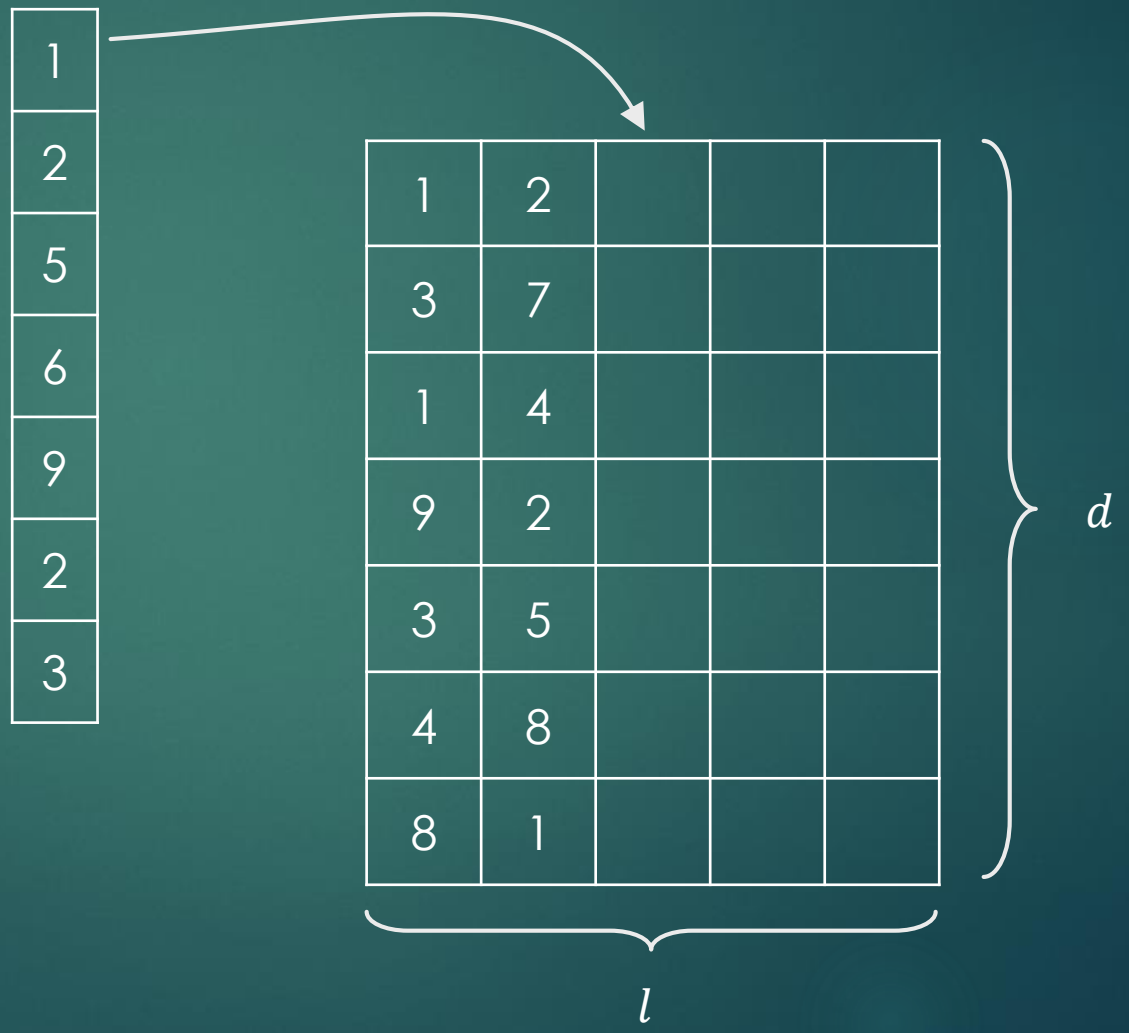
maintain the invariant that some columns are empty (zero-valued)



Frequent Directions

Stream in matrix A one column at a time

Input vectors are simply stored In empty columns



Frequent Directions

When the sketch is ‘full’
We need to zero out some columns

1	2	3	8	6
3	7	5	7	0
1	4	4	3	3
9	2	2	3	7
3	5	6	6	1
4	8	1	5	4
8	1	2	6	2

d

l

Singular Value Decomposition

SWISS ARMY KNIFE OF LINEAR ALGEBRA

CS 5965/6965 - Big Data Systems - Fall 2014

Goal: Given a $m \times n$ matrix A,

$$A = U \Sigma V^* = \sum_{j=1}^{n} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^*$$

$m \times n$ $m \times n$ $n \times n$ $n \times n$

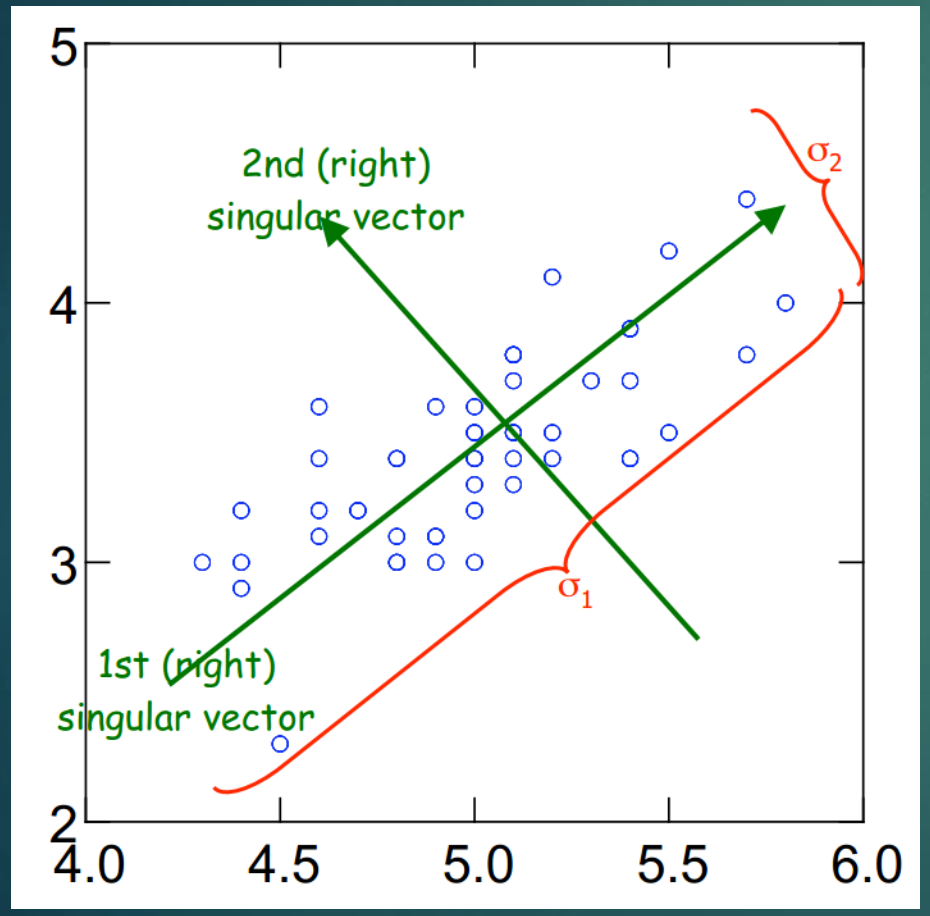
$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are the singular values of A
$\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_n}$ are orthonormal, the left singular vectors of A, and
$\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_n}$ are orthonormal, the right singular vectors of A.

Singular Value Decomposition

- Closely related problems:
  - Eigenvalue decomposition $A \approx V \Lambda V^*$
  - Spanning columns or rows $A \approx C\ U\ R$

- Applications:
  - **Principal Component Analysis**: Form an empirical covariance matrix from some collection of statistical data. By computing the singular value decomposition of the matrix, you find the directions of maximal variance
  - **Finding spanning columns or rows**: Collect statistical data in a large matrix. By finding a set of spanning columns, you can identify some variables that "explain" the data. (Say a small collection of genes among a set of recorded genomes, or a small number of stocks in a portfolio)
  - **Relaxed solutions to k-means clustering**: Relaxed solutions can be found via the singular value decomposition
  - **PageRank**: primary eigenvector

Singular values, intuition



- Blue circles are m data points in $2D$
- The SVD of the $m \times 2$ matrix
  - V_1: 1st (right) singular vector: direction of maximal variance,
  - σ_1: how much of data variance is explained by the first singular vector
  - V_2: 2nd (right) singular vector: direction of maximal variance, after removing projection of the data along first singular vector.
  - σ_2: measures how much of the data variance is explained by the second singular vector

SVD - Interpretation

$M = U\Sigma V^*$ - example:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

v1

SVD - Interpretation

$M = U\Sigma V^*$ - example:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

variance ('spread') on the v1 axis

SVD - Interpretation

$\boldsymbol{M} = \boldsymbol{U\Sigma V}^*$ - example:

- $U\Sigma$ gives the coordinates of the points in the projection axis

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Dimensionality reduction

set the smallest eigenvalues to zero:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 9.64 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

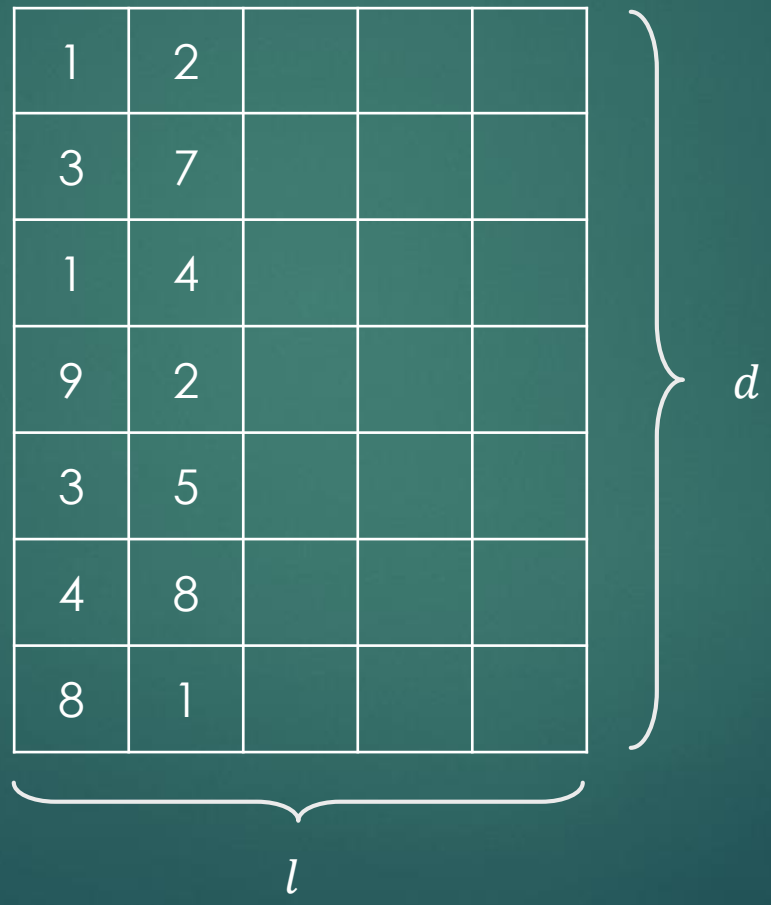
Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Frequent Directions (Liberty 2013)

maintain a sketch of at most l columns

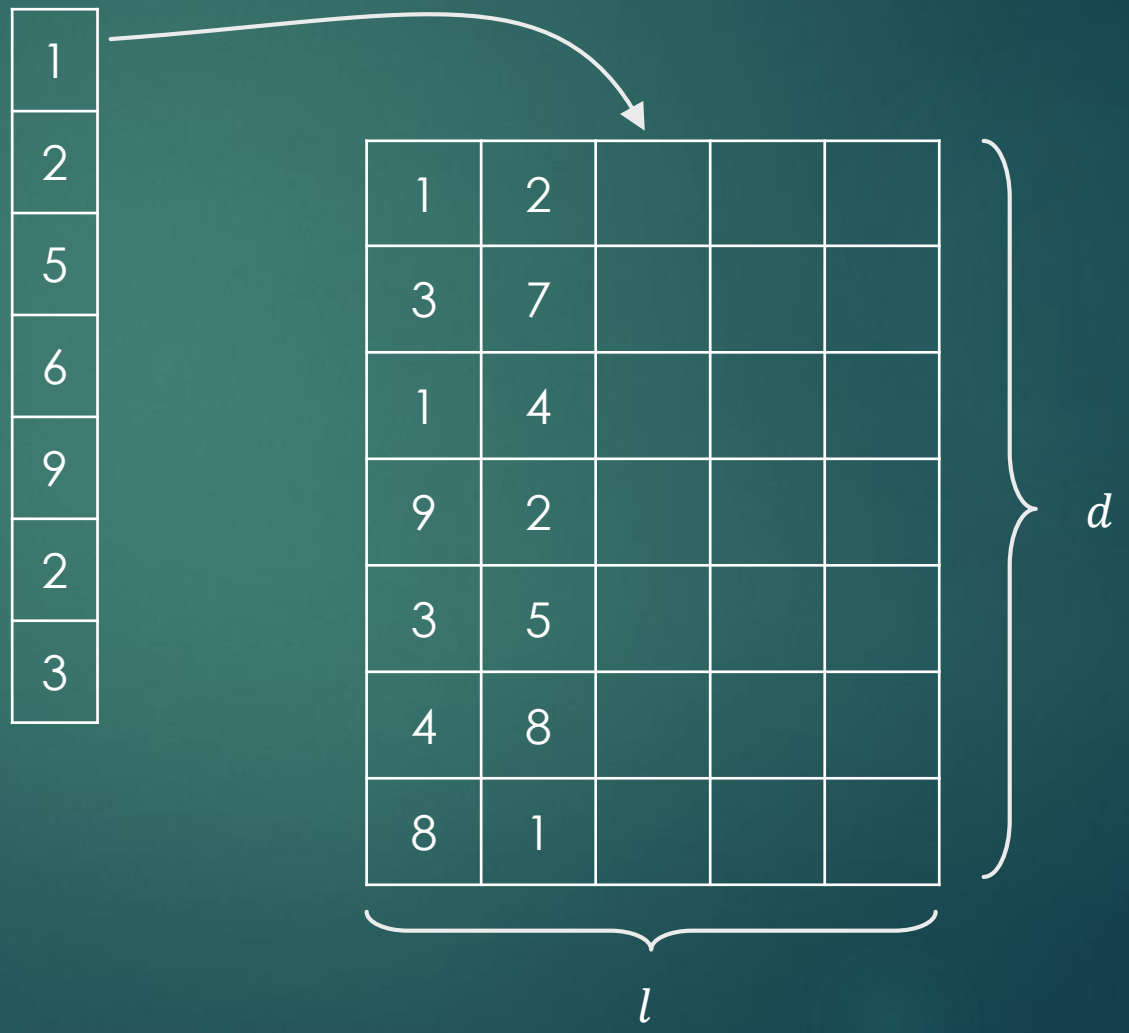
maintain the invariant that some columns are empty (zero-valued)



Frequent Directions

Stream in matrix A one column at a time

Input vectors are simply stored In empty columns



Frequent Directions

When the sketch is ‘full’
We need to zero out some columns

1	2	3	8	6
3	7	5	7	0
1	4	4	3	3
9	2	2	3	7
3	5	6	6	1
4	8	1	5	4
8	1	2	6	2

d

l

Frequent Directions

Using SVD, compute

$$B = U\Sigma V^T$$

And set

$$B_{new} = U\Sigma$$

$$BB^T = B_{new}B_{new}^T$$

$B = U\Sigma V^T$

$B_{new} = U\Sigma$ | V^T

-9.2	-0.4	-5.1	1.5	0.9
-10.4	-4.4	2.0	0.5	0.3
-6.5	-1.9	-1.1	-0.9	-1.9
-9.9	6.7	0.1	-1.1	-1.4
-9.6	-3.2	1.1	1.5	-1.4
-10.1	-0.8	0.3	-4.0	1.6
-9.0	3.8	2.1	2.7	1.3

Frequent Directions

The columns of B are now orthogonal and in decreasing magnitude

$B_{new} = U\Sigma$

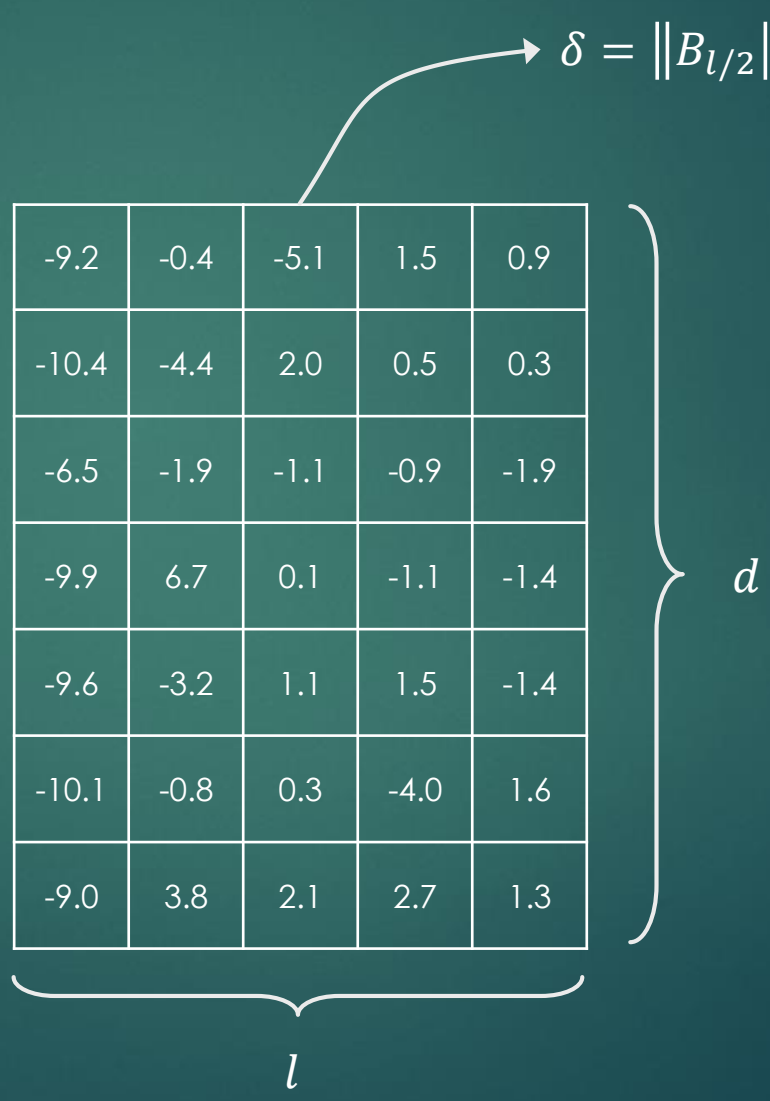
-9.2	-0.4	-5.1	1.5	0.9
-10.4	-4.4	2.0	0.5	0.3
-6.5	-1.9	-1.1	-0.9	-1.9
-9.9	6.7	0.1	-1.1	-1.4
-9.6	-3.2	1.1	1.5	-1.4
-10.1	-0.8	0.3	-4.0	1.6
-9.0	3.8	2.1	2.7	1.3

d

l

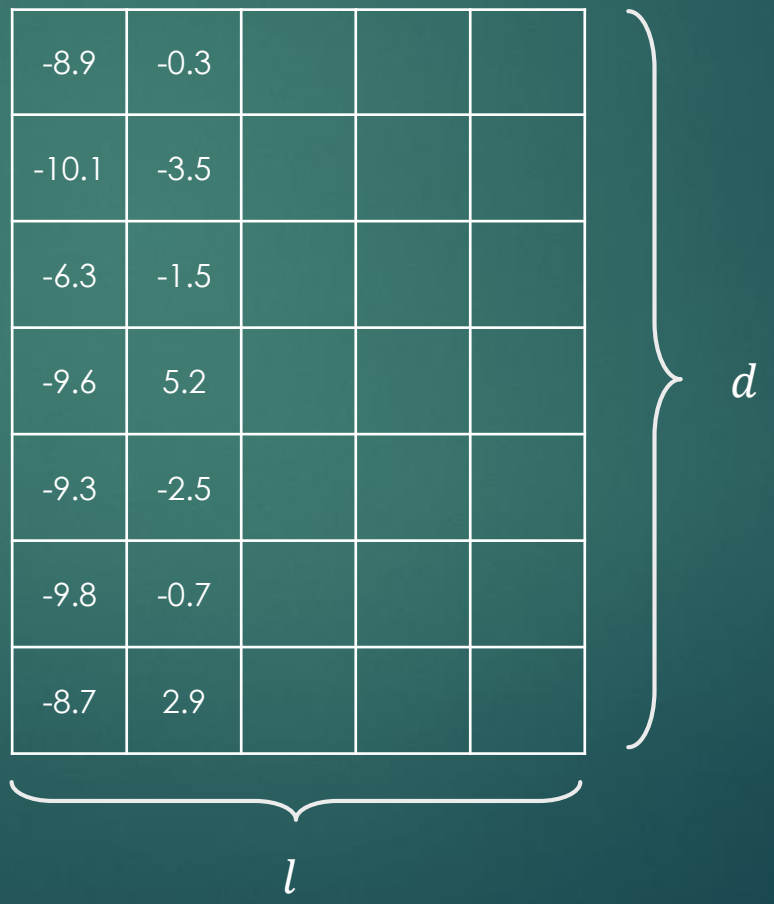
Frequent Directions

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

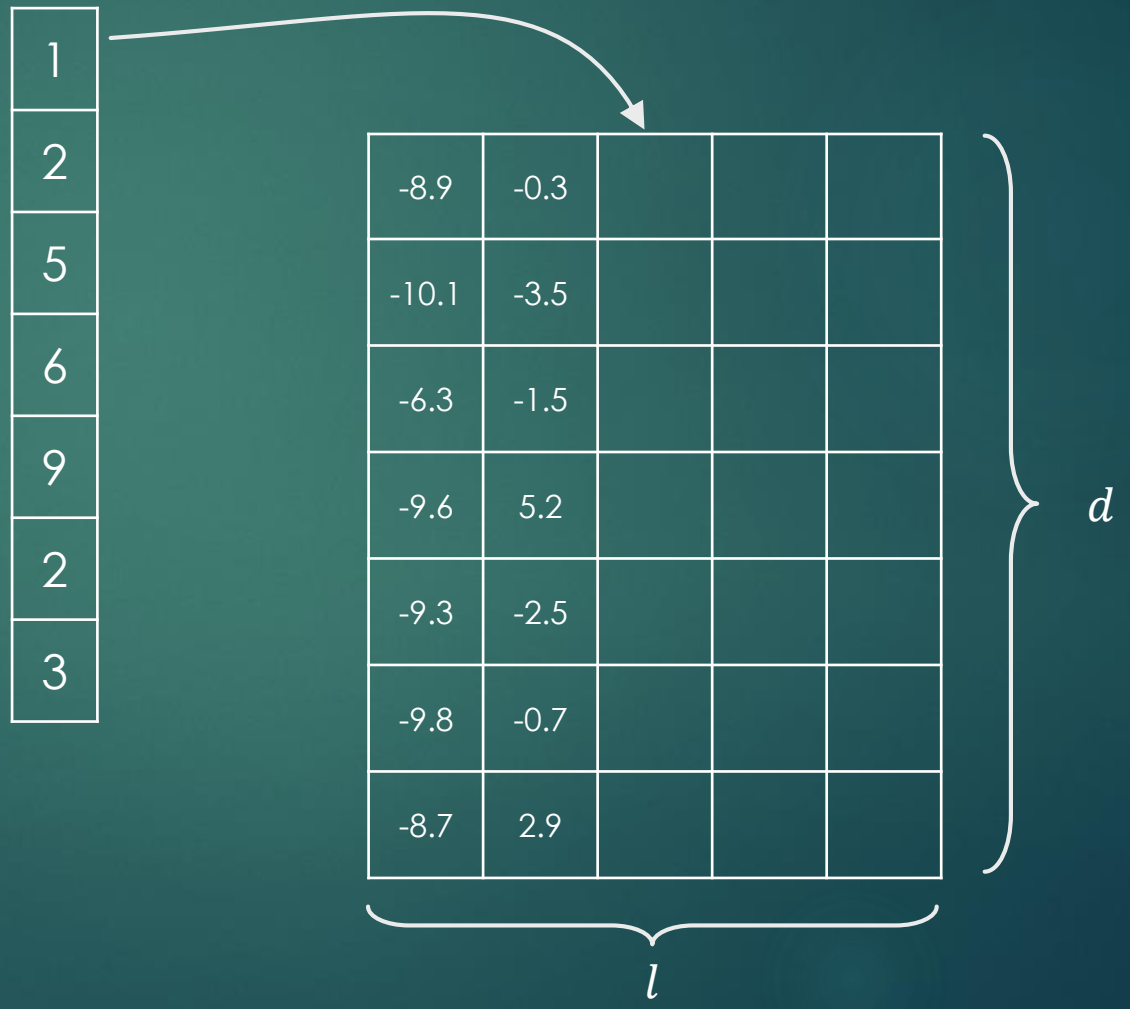


Frequent Directions

Reduce column l_2^2-norms by δ



Frequent Directions



Start aggregating columns again

Frequent Directions

Input: $l,\ A \in \mathbb{R}^{d \times n}$

$B \leftarrow$ all zeros matrix $\in \mathbb{R}^{d \times l}$

for $i \in [n]$ **do**

Insert A_i into a zero valued column of B

if B has no zero valued columns **then**

$[U, \Sigma, V] \leftarrow SVD(B)$

$\delta \leftarrow \sigma_{l/2}^2$

$\hat{\Sigma} \leftarrow \sqrt{\max(\Sigma^2 - I_l \delta, 0)}$

$B \leftarrow U\hat{\Sigma}$

Return: B

Bounding the error

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

$$\sup_{\|x\|=1} \|xA\|^2 - \|xB\|^2 = \sup_{\|x\|=1} \sum_{t=1}^{n} [\langle x, A_t \rangle^2 + \|xB^{t-1}\|^2 - \|xB^t\|^2]$$

$$= \sup_{\|x\|=1} \sum_{t=1}^{n} [\|xC^t\|^2 - \|xB^t\|^2]$$

$$\leq \sum_{t=1}^{n} \left\| C^{t^T} C^t - B^{t^T} B^t \right\| \cdot \|x\|^2$$

$$= \sum_{t-1}^{n} \delta_t$$

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,

$$0 \leq \|B\|_f^2 = \sum_{t=1}^{n} \left[\|B^t\|_f^2 - \|B^{t-1}\|_f^2 \right]$$

$$= \sum_{t=1}^{n} \left[\left(\|C^t\|_f^2 - \|B^{t-1}\|_f^2 \right) - \left(\|C^t\|_f^2 - \|B^t\|_f^2 \right) \right]$$

$$= \sum_{t=1}^{n} \|A_t\|^2 \; - tr\left(C^{t^T} C^t - B^{t^T} B^t \right)$$

$$\leq \|A\|_f^2 - \frac{l}{2} \sum_{t=1}^{n} \delta_t$$

Which gives,

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

Bounding the error

We saw that:

$$\|AA^T - BB^T\| \leq \sum \delta_t$$

and,

$$\sum \delta_t \leq \frac{2\|A\|_f^2}{l}$$

setting $l = \frac{2}{\epsilon}$ gives us,

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

Divide & Conquer

- Sketching can be implemented in a divide & conquer fashion as well
- Let $A = [A_1; A_2]$
- Compute the sketches B_1, B_2 of matrices A_1, A_2
- Compute the sketch C of the matrix $[B_1; B_2]$
- It can be shown that

$$\|AA^T - CC^T\| \leq \frac{2\|A\|_f^2}{l}$$