Streams Processing

Matrix Sketching

Dimensionality reduction

- Linear
  - Principal Component Analysis: SVD-based, PPCA, GLRM
  - **Approx PCA: Matrix sketching**
  - Compressed sensing
- Non-linear
  - Kernel PCA
  - Isometric mapping

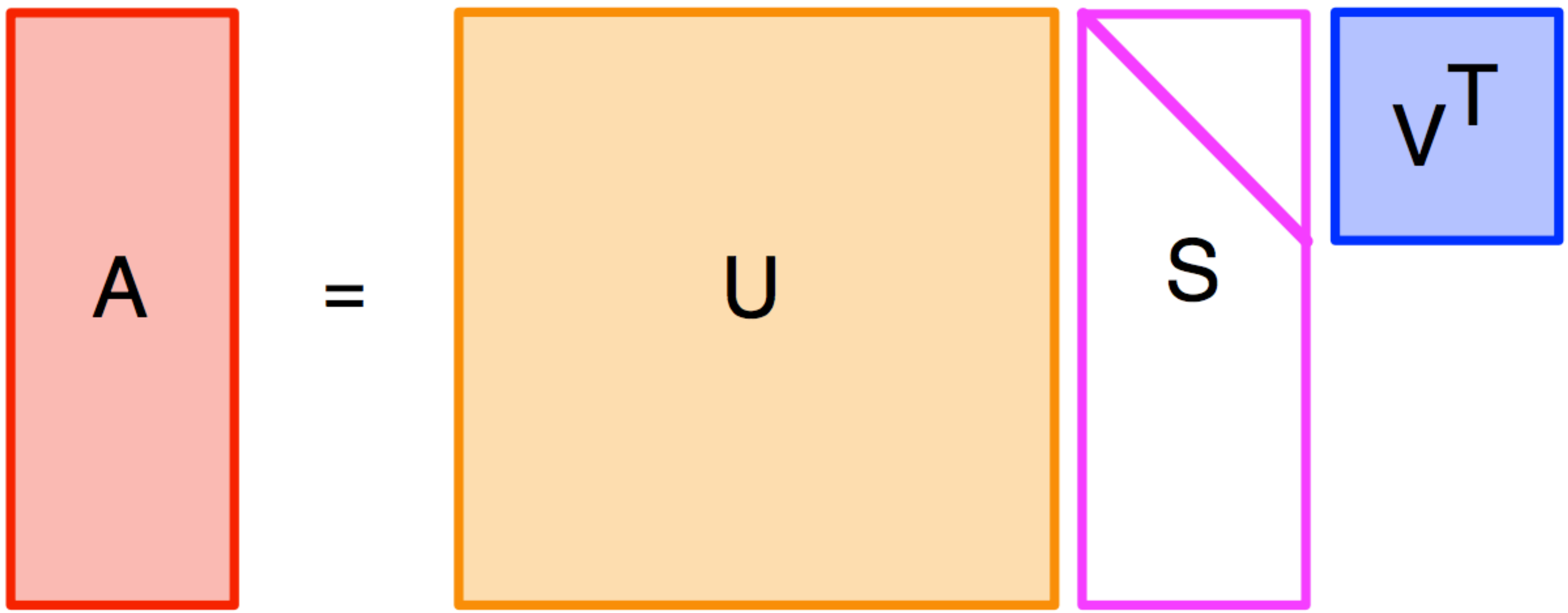
Matrix sketching

Online, interpretable PCA

We are going to see a way to “improve” PCA and the SVD

Matrix sketching: the SVD

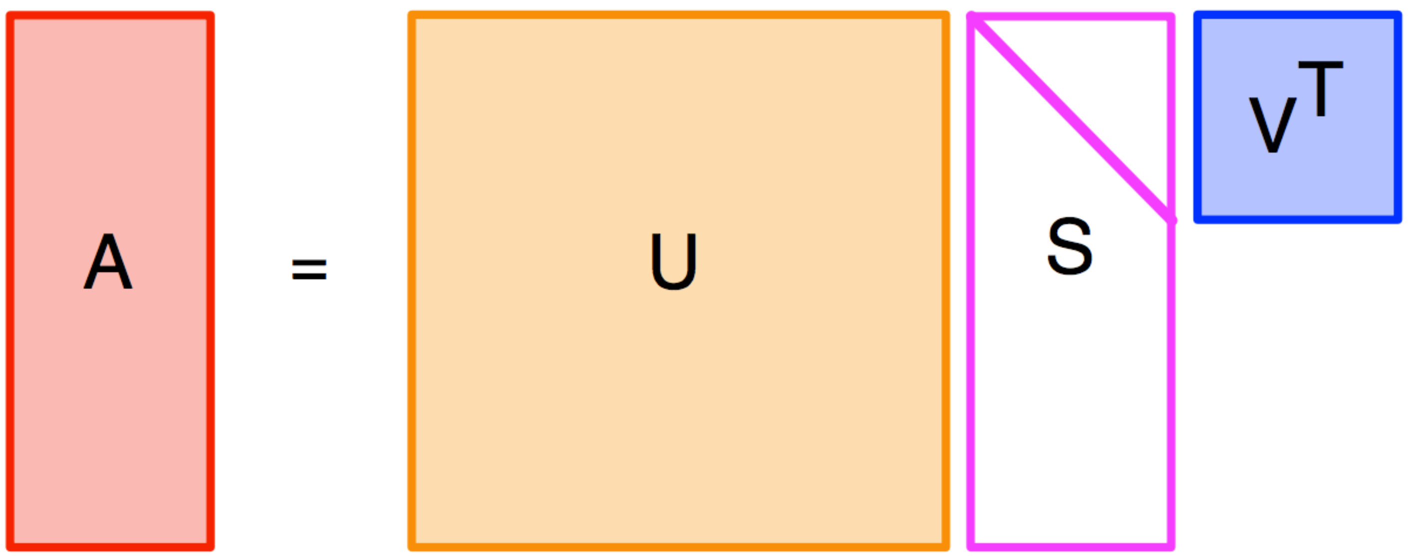
$$[U, S, V] = \mathrm{svd}(A)$$



Jeff M. Phillips, University of Utah

Matrix sketching: the SVD

$[U, S, V] = \mathrm{svd}(A)$



$$U = [u_1, \ldots, u_n]$$

$$S = \text{diag}(\sigma_1, \ldots, \sigma_d)$$

$$V = [v_1, \ldots, v_d]$$

Jeff M. Phillips, University of Utah

$$A = \sum_{j=1}^{d} \sigma_j u_j v_j^T$$

PCA: $A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T$

Approximation of *A* by SVD truncation

Interpretability of V (Principal Components)

Columns of V are linear combinations of features

What is a linear combination of purchased books?

What is a linear combination between number of calls and being male or female?

Approximation of *A* by SVD truncation

Computational efficiency

Computing the SVD demands loading all A into memory and requires time $O(\min\{nd^2, n^2d\})$

What about **really big data**?

What about **streamed** data?

Data stream *A*: each row of the input matrix can be processed only once and storage is severely limited

What about **distributed** data?

Matrix sketch

Sketch of a matrix *A*: Small matrix *B*, that approximates *A* well.

$$A \in \mathbb{R}^{n \times m}$$

we want to find $B \in \mathbb{R}^{\ell \times m}$ with $\ell \ll n$

such that $A^T A \approx B^T B$

Row sampling

Interpretability: choose V so that columns of V are columns of A

Different notions of **importance**

Example

$$S = \{(a_1, w_1), \ldots, (a_n, w_n)\}$$

Define representative sample,
e.g., a sample representative of the total weight of S, in expectation

Importance sampling

Input: $A \in \mathbb{R}^{d \times n}, 1 \leq c \leq n$
Output: $B \in \mathbb{R}^{d \times c}$
$B \leftarrow$ all zeros matrix $\in \mathbb{R}^{d \times c}$
for $i \in [n]$ **do**
    Compute probability p_i for row $A_{:,i}$
    for $j \in [c]$ **do**
        Insert (and rescale) $A_{:,i}$ into $B_{:,j}$ with probability p_i
return B

Row sampling

How it works:

For each row $a_i \in A$

Compute $w_i = \|a_i\|^2$

Select *t* rows and form *R,* with probability proportional to w_i

t rows will stand for V_k^T

$$R = \begin{bmatrix} w_{i1} a_{i1} \\ \cdots \\ w_{it} a_{it} \end{bmatrix}$$

$$t = (k/\epsilon)^2 \log(1/\delta)$$

Row sampling

However

V_k has orthogonal columns but *R* does not…

Solution: orthogonalize *R*, using a projection matrix

$$\Pi_R = R^T(RR^T)^{-1}R$$

Projection of *A* onto the subspace of *R*

$$A_R = A\Pi_R$$

Row sampling

Remember:

$$t = (k/\epsilon)^2 \log(1/\delta) \qquad R = \begin{bmatrix} w_{i1} a_{i1} \\ \cdots \\ w_{it} a_{it} \end{bmatrix}$$

$$\Pi_R = R^T (RR^T)^{-1} R$$

It can be proven:

$$\mathbb{P}\left(\|A - A\Pi_R\|_F \leq \|A - A_k\|_F + \epsilon \|A\|_F \right) \geq 1 - \delta$$

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on. IEEE, 1998

Row sampling

We can sample columns or columns and rows

$$A \approx CUR$$

Sampled rows

Sampled columns

Row sampling: CUR decomposition

$$A \approx CUR$$

$$\begin{bmatrix} \\ A \\ \\ n \times m \end{bmatrix} \approx \begin{bmatrix} \text{Sample} \\ \text{columns} \\ \\ n \times s \end{bmatrix} \underset{s \times r}{\begin{bmatrix} \text{Multi} \\ \text{plier} \end{bmatrix}} \begin{bmatrix} \text{Sample rows} \\ r \times m \end{bmatrix}$$

Avrim Blum, John Hopcroft, and Ravindran Kannan, Foundations of Data Science

Frequent directions

Intuition: frequent items algorithm

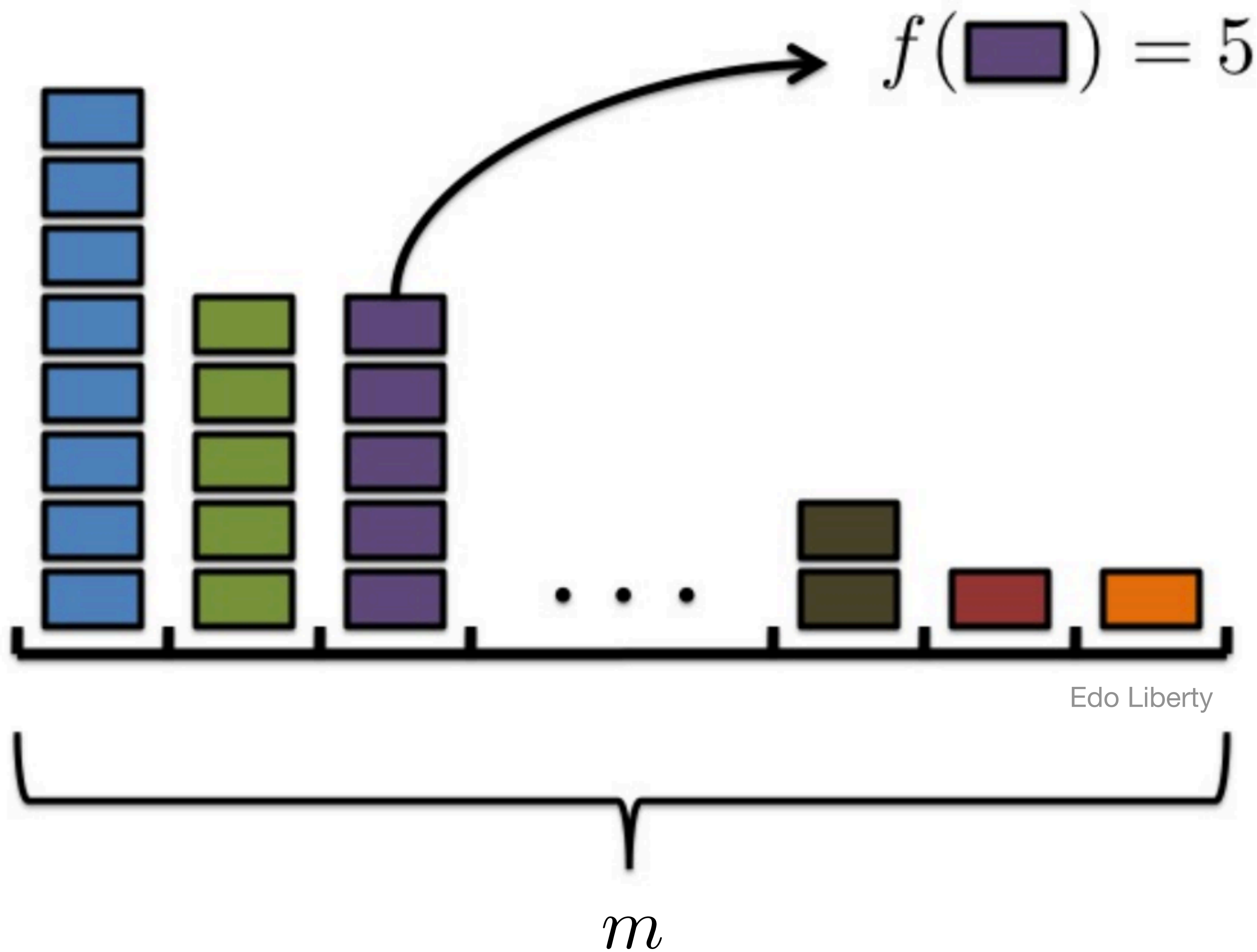
Finding repeated elements, Misra, Gries, 1982

Goal: estimate the frequency of each item in a stream of items

There are *m* different items, and a stream of *n* items appearing

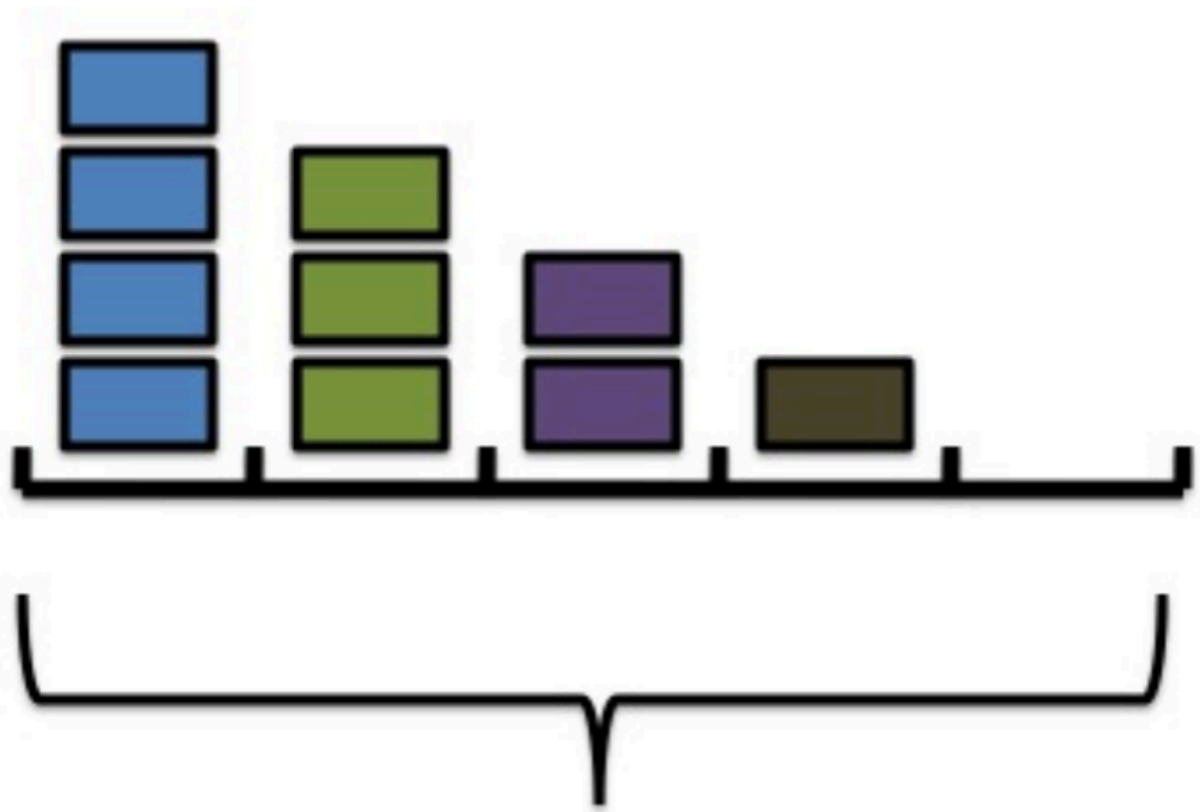
The frequency f_i is the number of times item *i* appeared on the stream

Frequent items



But, in general, we cannot use m counters…

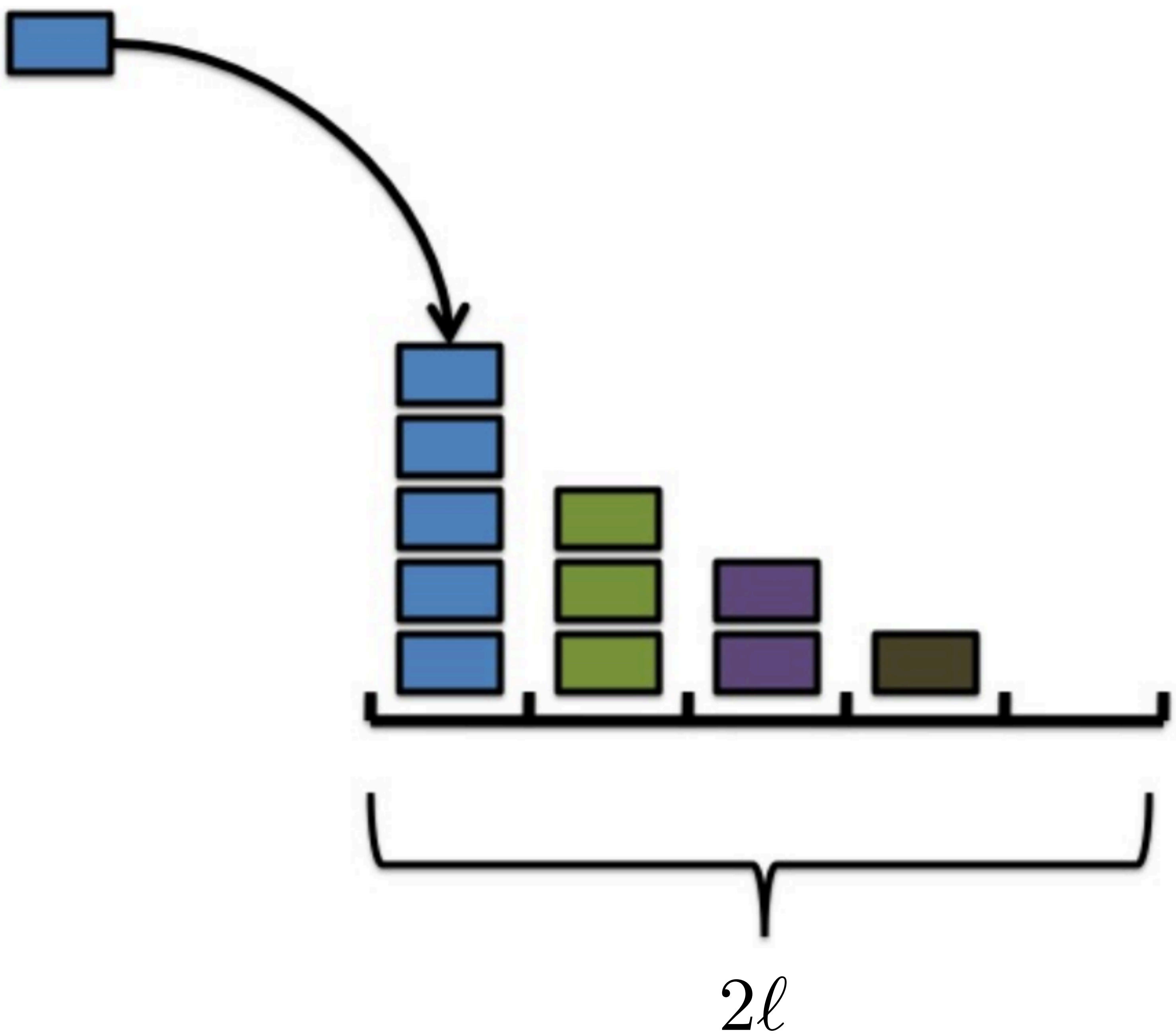
Frequent items



2ℓ

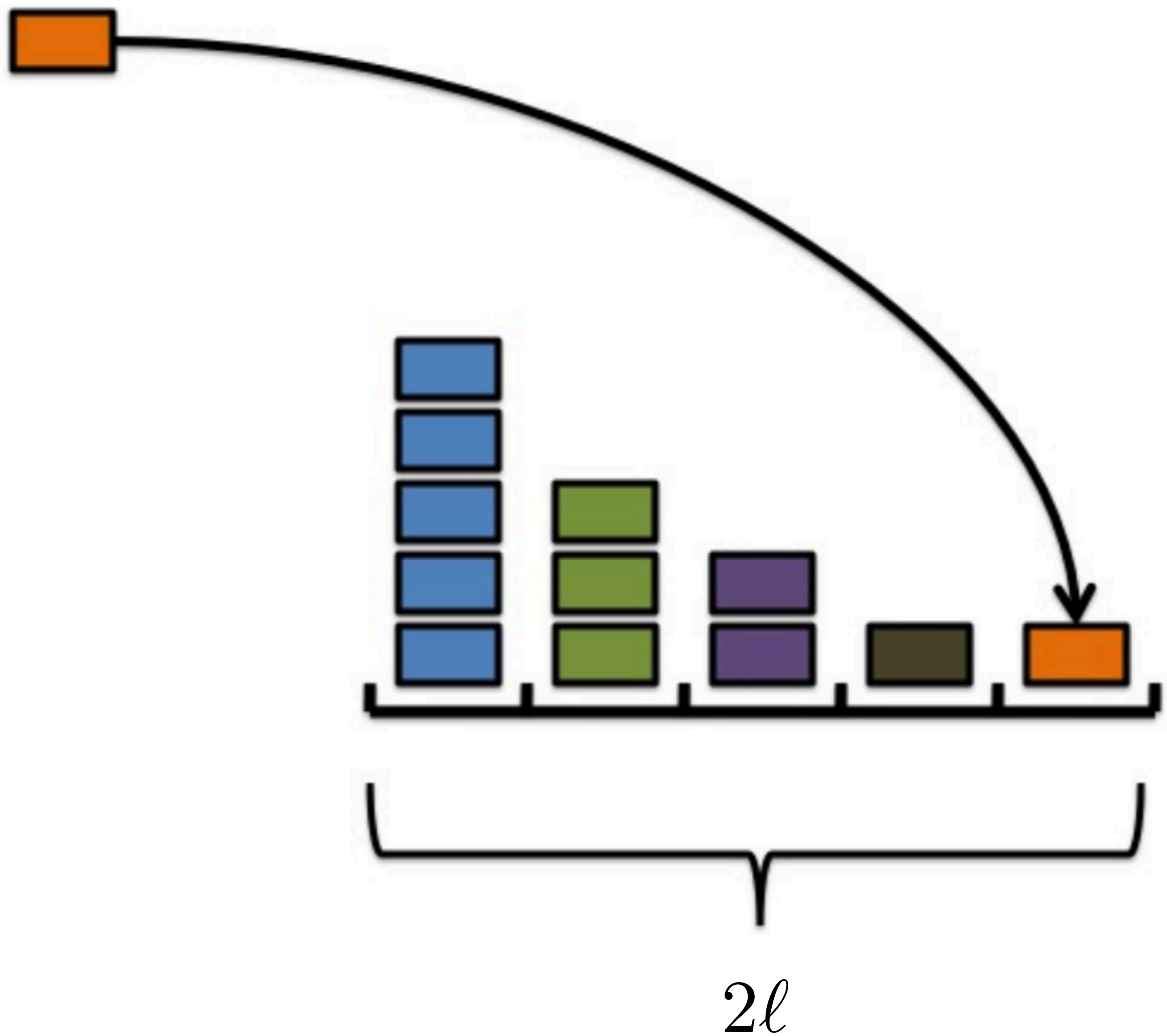
Keep less than a fixed # of counters

Frequent items



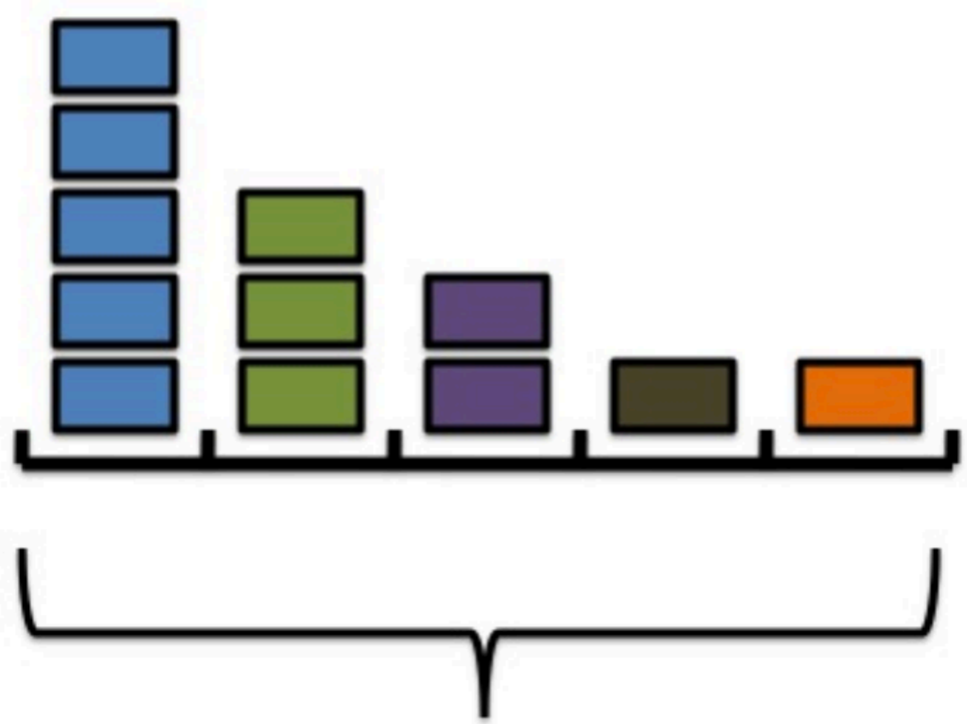
if an item has a counter, increment it

Frequent items



If not, use a free counter and increasing it

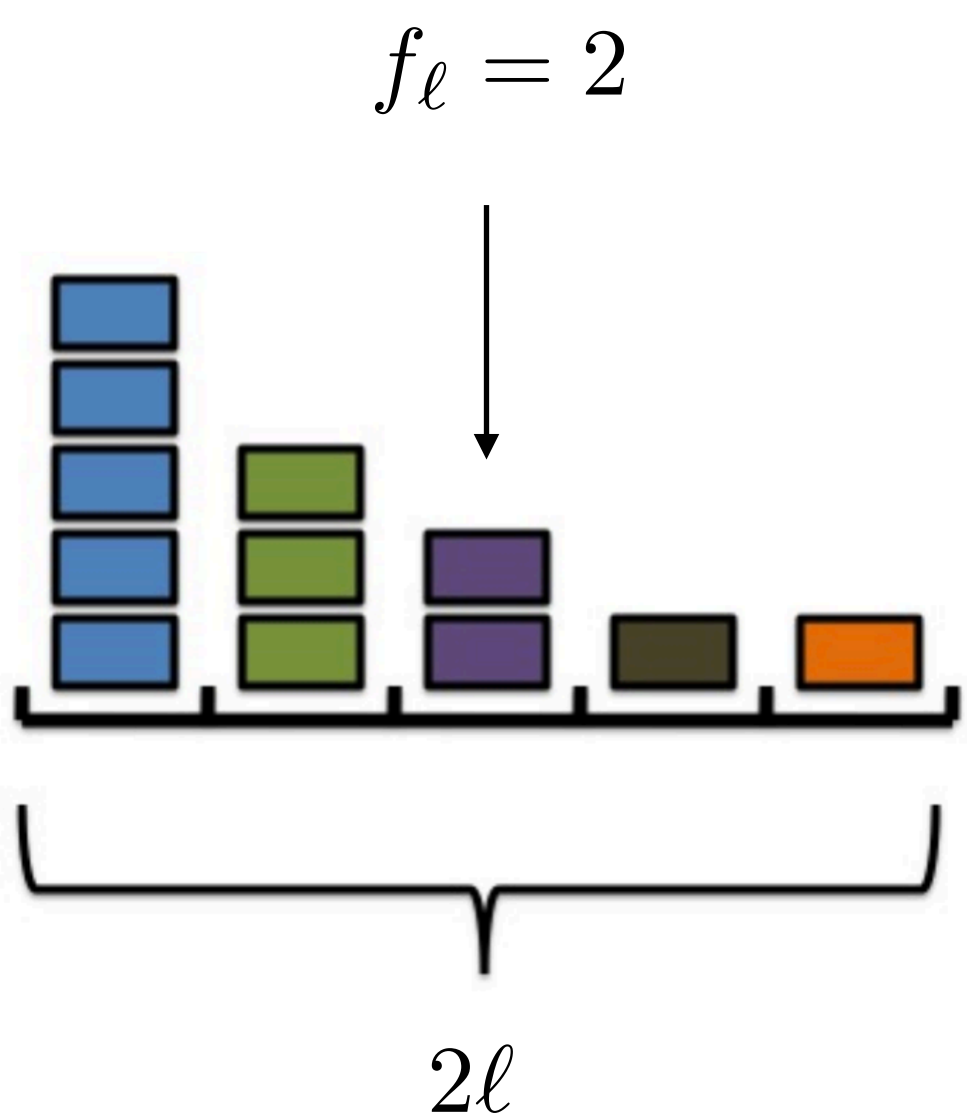
Frequent items



2ℓ

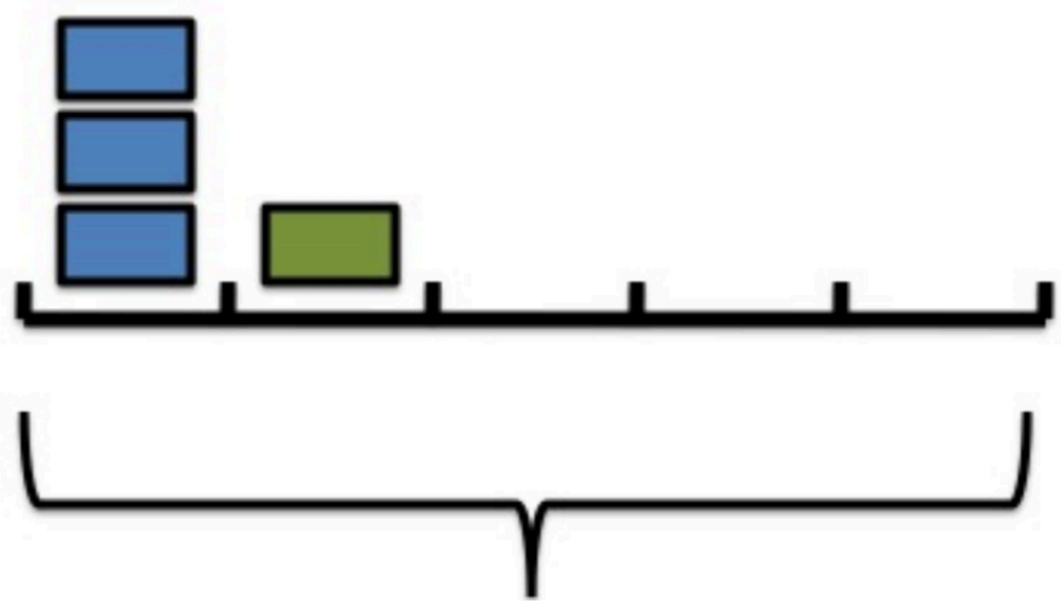
Now we have no slot available

Frequent items



Compute the median

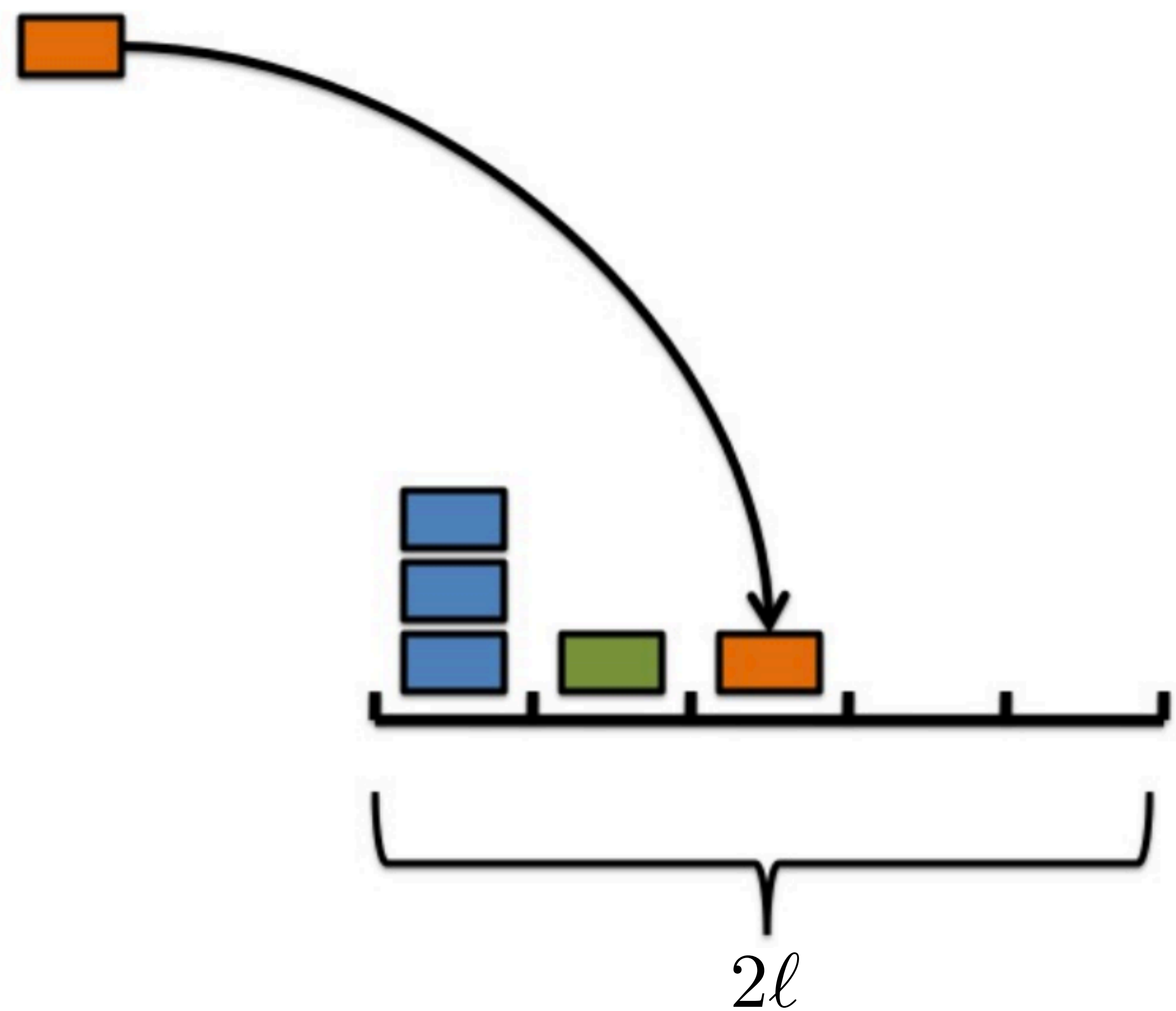
Frequent items



2ℓ

Decrease all items by f_ℓ

Frequent items



And continue…

Frequent items

Approximated counts: f_i'

$$|f_i - f_i'| \leq \frac{n}{\ell}$$

Finding repeated elements, Misra, Gries, 1982

Frequent directions

$B \in \mathbb{R}^{2\ell \times m}$ with only 2ℓ rows (directions)

Take the first 2ℓ rows from *A* as *B*

$$[U, S, V] = \texttt{svd}(B)$$

$$S = \text{diag}(\sigma_1, \ldots, \sigma_{2\ell})$$

if $\sigma_{2\ell} > 0$ subtract $\delta = \sigma_\ell^2$ to each squared entry in *S*

$$S' = \text{diag}(\sqrt{\sigma_1^2 - \delta}, \ldots, \sqrt{\sigma_{\ell-1}^2 - \delta}, 0, \ldots, 0)$$

$$B = S'V^T$$

Frequent directions

For any direction (unit norm) x

$$0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \|A - A_k\|_F^2/(\ell - k)$$

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. SICOMP, 2016.

Frequent directions

For any direction (unit norm) x

$$0 \leq ||Ax||^2 - ||Bx||^2 \leq ||A - A_k||_F^2/(\ell - k)$$

Why does it work?

When some mass is deleted from one counter it is also deleted from all counters, and none can be negative

Squared mass can be summed along orthogonal directions independently

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. SICOMP, 2016.