CUR: How it Works

- **Sampling columns (similarly for rows):**
- **ColumnSelect algorithm:**

Input: matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, sample size c

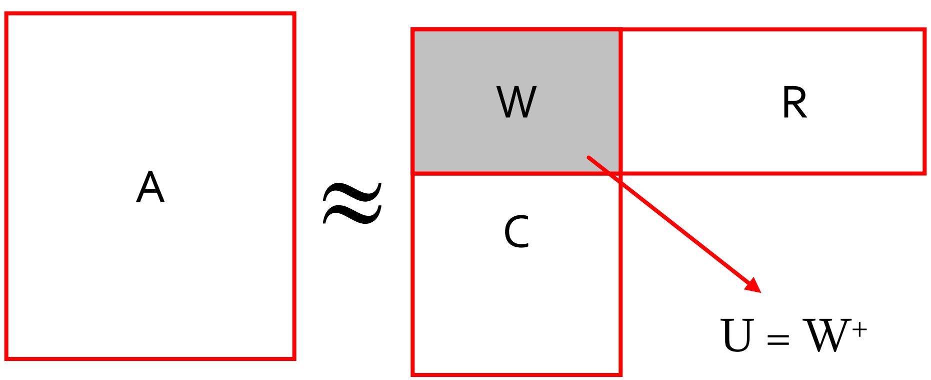
Output: $\mathbf{C}_d \in \mathbb{R}^{m \times c}$

1. for $x = 1 : n$ [column distribution]
2. $P(x) = \sum_i \mathbf{A}(i, x)^2 / \sum_{i,j} \mathbf{A}(i, j)^2$
3. for $i = 1 : c$ [sample columns]
4. Pick $j \in 1 : n$ based on distribution $P(x)$
5. Compute $\mathbf{C}_d(:, i) = \mathbf{A}(:, j) / \sqrt{cP(j)}$

Note this is a randomized algorithm, same column can be sampled more than once

Computing U

- Let **W** be the "intersection" of sampled columns **C** and rows **R**
  - Let SVD of $\mathbf{W} = \mathbf{X\ Z\ Y}^T$
- **Then: $\mathbf{U} = \mathbf{W}^+ = \mathbf{Y\ Z}^+\ \mathbf{X}^T$**
  - Z^+: **reciprocals of non-zero singular values:** $Z^+_{ii} = 1/\ Z_{ii}$
  - W^+ is the "**pseudoinverse**"



Why pseudoinverse works?
$W = X\ Z\ Y$ then $W^{-1} = X^{-1}\ Z^{-1}\ Y^{-1}$
Due to orthonomality
$X^{-1}=X^T$ and $Y^{-1}=Y^T$
Since Z is diagonal $Z^{-1} = 1/Z_{ii}$
Thus, if **W** is non-singular, pseudoinverse is the true inverse

CUR: Provably good approx. to SVD

- For example:
  - Select $c = O\left(\frac{k \log k}{\varepsilon^2}\right)$ columns of A using ColumnSelect algorithm
  - Select $r = O\left(\frac{k \log k}{\varepsilon^2}\right)$ rows of A using ColumnSelect algorithm
  - Set $U = W^+$
- **Then:** $\|A - CUR\|_F \leq (2 + \epsilon) \|A - A_k\|_F$ with probability 98%