CUR Decomposition

Review …

- Last time
  - Matrix Sketching
- Today
  - CUR Decompositions
- Next class
  - Social Networks – Graph Algorithms

Matrix Approximations ($n \times m$)

- UV decomposition
  - Optimization problem
  - Dense U, V matrices of size $n \times d$ and $d \times n$
- SVD
  - Very powerful algorithm
  - Low-rank approximation using random projections
    - Dense U, V matrices of size $n \times d$ and $d \times n$
- Sketching
  - Streaming algorithm
  - Maintain the dense sketch of size $n \times d$

CUR Decomposition

- In most applications, the matrix A is sparse
- Although the previously covered approaches achieve dimensionality reduction, we are still left to deal with either a large n or large m
- In CUR decomposition, we decompose the matrix A as,

$$A = C\ U\ R$$

where, C, R are sparse matrices of size $n \times d$ and $d \times m$ and U is a dense matrix of size $d \times d$

- Is always an approximation irrespective of the choice of d

CUR Decomposition

- Consider matrix A of size $n \times m$
- Choose the number of "concepts" r to be used for the decomposition
  - Remember the Utility matrix for recommendation systems
  - Similar to the idea of classes of users/items
- A CUR-decomposition of A is
  - a randomly chosen set of r columns of A, which form the $n \times r$ matrix C,
  - a randomly chosen set of r rows of A , which form the $r \times m$ matrix R, and
  - a specially constructed $r \times r$ matrix U

Constructing the U matrix

- Let W be the $r \times r$ matrix at the intersection of the chosen rows and columns (C, R) of A
- Compute the SVD of W; $W = X\Sigma Y^T$
- Compute Σ^+, the pseudoinverse of Σ
- $U = Y(\Sigma^+)^2 X^T$

Choosing the right rows & columns

- Although we choose the rows & columns randomly, we maintain a bias towards the more important rows & columns
- The measure of importance is square of the Frobenius norm

$$f = \sum_{i,j} a_{ij}^2$$

- We choose rows with probability $p_i = \sum_j a_{ij}^2 / f$, and
- We choose columns with probability $q_j = \sum_i a_{ij}^2 / f$.
- Scale each selected row/column by dividing its elements by the square root of the expected number of times this row/column would be picked – i.e., divide elements by $\sqrt{rq_j}$ or $\sqrt{rp_i}$