APPM 4/5720, 01-29-2016 Notes

1 Recall the RSVD

1. G = randn(n,k+p)
2. $Y = AG$, where A is $m \times n$, G is $n \times (k+p)$
3. $[Q, \sim, \sim]$ = qr(Y,0), where Y is $m \times (k+p)$
4. $B = Q^* A$
5. $[Uhat, D, V]$= svd(B,'econ')
6. $U = QUhat$, where $Uhat$ is $(k+p) \times (k+p)$
7. (Truncate)

2 Computational Costs

2.1 Environment 1: A fits in RAM

Traditional flop count is (sort of) a relevant measure. We make the following estimates:

- Cost of matrix-matrix multiply of matrices of dimensions i, j, k is

$$T \approx C_{mm} ijk$$

  $C = AB$, where C is $i \times k$, A is $i \times j$ and B is $j \times k$.

- Cost of QR/SVD Factorization of matrix of size $i \times j$ is

$$T_{qr} \approx C_{qr} ij \min(i, j)$$

$$T_{svd} \approx C_{svd} ij \min(i, j)$$

Assume the number of extra samples p is small, and can be ignored. Then, the cost of the steps of the RSVD are:

2. $\sim nk$ small so ignore!
3. $C_{mm} mnk$
4. $C_{qr} mk^2$
5. $C_{mm} mnk$

6. $C_{svd} nk^2$

7. $C_{mm} mk^2$

So $T_{rsvd} \approx C_{mm}(2mnk + mk^2) + C_{qr} mk^2 + C_{svd} nk^2$. The asymptotically dominant term is mnk since $k < \min(m,n)$, so crudely $T_{rsvd} \sim mnk$.

2.2 Environment 2: A is dense and stored on a hard drive

A is stored "out of core." Assume k is small enough that G, Y, Q, B, U, V, D all fit in RAM. We have $O(k(m+n))$ RAM but not $O(mn)$, so A does not fit.

In this case, the time required to read A from disk dominates. So a relevant estimate would be

$$T_{rsvd} \approx 2 * (\text{cost of reading } A) + C_{mm} mk^2 + C_{qr} mk^2 + C_{svd} nk^2$$

where the cost of reading A depends on the bandwidth of the machine. The CPU will be idle most of the time as data is being moved for computation.

Note: The two reads come from

1. $Y = AG$

2. $B = Q^* A$

2.3 Environment 3: A and A^* can rapidly be applied to a vector or matrix

Let A be a $m \times n$ matrix. Let X_1 be $m \times r$ and X_2 be $n \times r$. Suppose that the cost of evaluating AX_1 is $\approx C_1 r$ and $A^* X_2$ is $\approx C_2 r$.

In Environment 1, we had $C_1 = C_{mm} mn$. If A is *sparse*, then $C_1 \ll C_{mm} mn$. In this case,

$$T_{rsvd} \approx (C_1 + C_2)k + C_{mm} mk^2 + C_{qr} mk^2 + C_{svd} nk^2.$$

Examples of Environment 3 include:

1. A is sparse

2. A has internal structure. For instance A is a convolution matrix:

$$\begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_n \\ b_n & b_1 & b_2 & \dots & b_{n-1} \\ b_{n-1} & b_n & b_1 & \dots & b_{n-2} \\ \vdots & & \vdots & \ddots & \vdots \\ b_2 & b_3 & b_4 & \dots & b_1 \end{bmatrix}$$

   Then A can be applied to a vector in $O(n \log n)$ operations using the Fast Fourier Transform. $y = F^{-1}(FAF^{-1})Fx$ where F is the $n \times n$ discrete Fourier Transform.

   The "FFT" algorithm applies F or F^{-1} in $O(n \log n)$ operations.

3. Applying A could consist of solving a PDE using a spare solver such as Multigrid methods.