Numerical Studies on PSVD Algorithms

Zheng Wang

April 21, 2014

We tested the following PSVD solvers on matrices from LSI and PCA applications, and compare their efficiency and accuracy.

1. Randomized Block Power method (rsvd_power): $q = k$, $p = 2$;
2. Randomized Block Krylov method (rsvd_krylov): $q = 2$, $p = 2$;
3. PROPACK;
4. svds (MATLAB).
5. LMSVD;
6. Block Chebyshev-Davidson method (bchdav_svd);

All the implementations are in MATLAB 7.8.0 (64-bit), and all experiments are run on the SMUHPC high-memory node with 8 cores and 144GB RAM. The compared quantities include

1. CPU time in seconds;
2. relative low-rank approximation error in F-norm

$$err\_mat = \frac{\|\mathbf{A} - \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}}\|_F}{\|\mathbf{A}\|_F}; \tag{0.1}$$

3. relative error of each computed singular triplet in 2-norm

$$err\_vec = \frac{\|\mathbf{A}\mathbf{v}_i - \sigma_i \mathbf{u}_i\|_2}{\|\mathbf{A}\|_2}. \tag{0.2}$$

0.1 Comparison with varying number of computed singular triplets

- Experiment on the News20 matrix:
  In bchdav_svd, $polym = 6$, $blk = 50$, $vimax = 250$. Convergence tolerance (tol) is 10^{-8} for all algorithms. We gradually increase the number of computed singular triplets (k) from 400 to 2000 with a stride length 400. Fig.(0.1) shows that our PSVD solver is the fastest with accurate results.

0.2 Comparison with varying matrix dimension

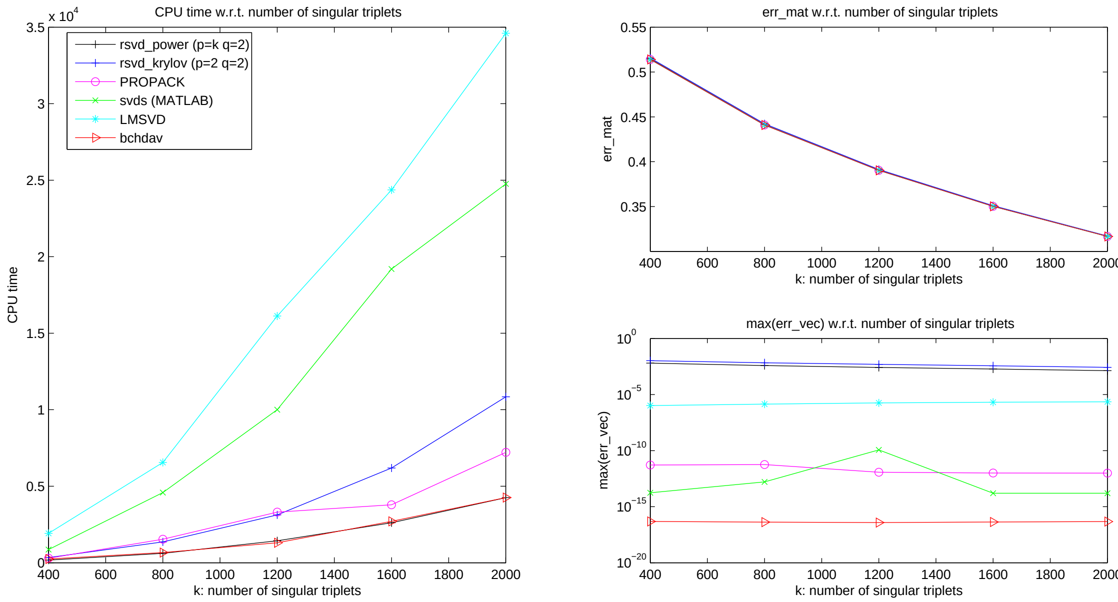

Figure 0.1: Comparison of CPU time, err_mat and max(err_vec) with varying number of computed singular triplets on sparse matrix News20 $(53,975 \times 11,269)$.