An Efficient, Sparsity-Preserving Online Algorithm for Data Approximation: Supplementary Material

1. The Singular Value Decomposition (SVD)

For any real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ there exist orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{U}^T \mathbf{A} \mathbf{V} = \text{diag}\left(\sigma_1, \cdots, \sigma_p\right) \stackrel{\text{def}}{=} \Sigma$$

such that $p = \min(m, n)$ and $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$. The decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ is known as the Singular Value Decomposition (Golub & van Loan, 2013).

For a given matrix $\mathbf{A}$ with rank ρ and a target rank k, rank-k approximation using the SVD achieves the minimal residual error in both spectral and Frobenius norms:

Theorem (Eckart-Young (Eckart & Young, 1936; Golub & van Loan, 2013)).

$$\min_{rank(\mathbf{B}) \leq k} ||\mathbf{A} - \mathbf{B}||_\xi^2 = ||\mathbf{A} - \mathbf{A}_k||_\xi^2 = \sum_{j=k+1}^{\rho} \sigma_j\left(\mathbf{A}\right)$$

where $\xi = F$ *or* 2.

2. Further Discussion of Rank-Revealing Algorithms

An important class of algorithms against which we test SRLU is **rank-revealing** algorithms for low-rank approximation:

Definition 1. *An LU factorization is **rank-revealing** (Miranian & Gu, 2003) if*

$$\sigma_k\left(\mathbf{A}\right) \geq \sigma_{\min}\left(\mathbf{L}_{11}\mathbf{U}_{11}\right) \gg \sigma_{\max}\left(\mathbf{S}\right) \geq \sigma_{k+1}\left(\mathbf{A}\right) \approx 0.$$

Several drawbacks exists to the above definition, including that $\mathbf{L}_{11}\mathbf{U}_{11}$ is not a low-rank approximation of the original data matrix, and that only certain singular values are bounded. Stronger algorithms were developed in (Miranian & Gu, 2003) by modifying the definition above to create strong rank-revealing algorithms:

Definition 2. *An LU factorization is **strong rank-revealing** if*

1.

$$\begin{aligned} \sigma_i\left(\mathbf{A}_{11}\right) &\geq \frac{\sigma_i\left(\mathbf{A}\right)}{q_1\left(k, m, n\right)}, \\ \sigma_j\left(\mathbf{S}\right) &\leq \sigma_{k+j}\left(\mathbf{A}\right) q_1\left(k, m, n\right), \end{aligned}$$

2.

$$\left|\left(\mathbf{A}_{21}\mathbf{A}_{11}^{-1}\right)|ij\right| \quad \leq \quad q_2\left(k, n, m\right),$$

3.

$$\left|\left(\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right)|ij\right| \quad \leq \quad q_3\left(k, n, m\right),$$

where $1 \leq i \leq k, 1 \leq j \leq n-k$, and $q_1(k,m,n)$, $q_2(k,m,n)$, and $q_3(k,m,n)$ are functions bounded by low-degree polynomials of k, m, and n.

Strong rank-revealing algorithms bound all singular values of the submatrix $\mathbf{A}_{11}$, but, as before, do not produce a low-rank approximation. Furthermore, they require bounding approximations of the left and right null spaces of the data matrix, which is both costly and not strictly necessary for the creation of a low-rank approximation. No known algorithms or numeric experiments demonstrate that strong rank-revealing algorithms can indeed be implemented efficiently in practice.

3. Updating R

The goal of TRLUCP is to access the entire matrix once in the initial random projection, and then choose column pivots at each iteration without accessing the Schur complement. Therefore, a projection of the Schur complement must be obtained at each iteration without accessing the Schur complement, a method that first appeared in (Melgaard & Gu, 2015). Assume that s iterations of TRLUCP have been performed and denote the projection matrix

$$\mathbf{\Omega} = \begin{pmatrix} \overset{sb}{\mathbf{\Omega}_1} & \overset{b}{\mathbf{\Omega}_2} & \overset{n-(s+1)b}{\mathbf{\Omega}_3} \end{pmatrix}.$$

Then the current projection of the Schur complement is

$$\mathbf{R}^{\text{cur}} = \begin{pmatrix} \overset{b}{\mathbf{R}_1^{\text{cur}}} & \overset{n-(s+1)b}{\mathbf{R}_2^{\text{cur}}} \end{pmatrix} = \begin{pmatrix} \mathbf{\Omega}_2 & \mathbf{\Omega}_3 \end{pmatrix} \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix},$$

where the right-most matrix is the current Schur complement. The next iteration of TRLUCP will need to choose columns based on a random projection of the Schur complement, which we wish to avoid accessing. We can write:

$$\begin{aligned}
\mathbf{R}^{\text{update}} &= \mathbf{\Omega}_3 \left(\mathbf{A}_{33} - \mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \right) \\
&= \mathbf{\Omega}_3\mathbf{A}_{33} + \mathbf{\Omega}_2\mathbf{A}_{23} - \mathbf{\Omega}_2\mathbf{A}_{23} - \mathbf{\Omega}_3\mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \\
&= \mathbf{\Omega}_3\mathbf{A}_{33} + \mathbf{\Omega}_2\mathbf{A}_{23} - \mathbf{\Omega}_2\mathbf{L}_{22}\mathbf{U}_{23} - \mathbf{\Omega}_3\mathbf{L}_{32}\mathbf{U}_{23} \\
&= \mathbf{R}_2^{\text{current}} - \left(\mathbf{\Omega}_2\mathbf{L}_{22} + \mathbf{\Omega}_3\mathbf{L}_{32} \right) \mathbf{U}_{23}. \qquad (1)
\end{aligned}$$

Here the current $\mathbf{L}$ and $\mathbf{U}$ at stage s have been blocked in the same way as Ω. Note equation (1) no longer has the term $\mathbf{A}_{33}$. Furthermore, $\mathbf{A}_{22}^{-1}$ has been replaced by substituting in submatrices of $\mathbf{L}$ and $\mathbf{U}$ that have already been calculated, which helps eliminate potential instability.

When the block size $b = 1$ and TRLUCP runs fully ($k = \mathbf{min}(m,n)$), TRLUCP is mathematically equivalent to the Gaussian Elimination with Randomized Complete Pivoting (GERCP) algorithm of (Melgaard & Gu, 2015). However, TRLUCP differs from GERCP in two very important aspects: TRLUCP is based on the Crout variant of the LU factorization, which allows efficient truncation for low-rank matrix approximation; and TRLUCP has been structured in block form for more efficient implementation.

4. Proofs of Theorems

Theorem 1. *For any truncated LU factorization*

$$\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\widehat{\mathbf{U}}\| \quad = \quad \|\mathbf{S}\|$$

for any norm $\|\cdot\|$. Furthermore,

$$\|\Pi_1 \mathbf{A} \Pi_2^T - \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}} \right)_s \|_2 \quad \leq \quad 2\|\mathbf{S}\|_2 + \sigma_{s+1}(\mathbf{A})$$

where $(\cdot)_s$ is the rank-s truncated SVD for $s \leq k \ll m, n$.

Proof. The equation simply follows from $\Pi_1 \mathbf{A} \Pi_2^T = \widehat{\mathbf{L}}\widehat{\mathbf{U}} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S} \end{pmatrix}$. For the inequality:

$$\begin{aligned}
&\|\Pi_1 \mathbf{A} \Pi_2^T - \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)_s \|_2 \\
&= \|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\widehat{\mathbf{U}} + \widehat{\mathbf{L}}\widehat{\mathbf{U}} - \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)_s \|_2 \\
&\leq \|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\widehat{\mathbf{U}}\|_2 + \|\widehat{\mathbf{L}}\widehat{\mathbf{U}} - \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)_s \|_2 \\
&= \|\mathbf{S}\|_2 + \sigma_{s+1}\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right) \\
&= \|\mathbf{S}\|_2 + \sigma_{s+1}\left(\Pi_1 \mathbf{A} \Pi_2^T - \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S} \end{pmatrix}\right) \\
&\leq \|\mathbf{S}\|_2 + \sigma_{s+1}\left(\mathbf{A}\right) + \|\mathbf{S}\|_2.
\end{aligned}$$

□

Theorem 2. *For a general rank-k truncated LU decomposition*

$$\sigma_j\left(\mathbf{A}\right) \leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \left(1 + \frac{\|\mathbf{A}\|_2}{\sigma_k\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\right)\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right).$$

Proof.

$$\begin{aligned}
&\sigma_j\left(\mathbf{A}\right) \\
&\leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \frac{\|\mathbf{S}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\right) \\
&= \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \frac{\sigma_j\left(\mathbf{A}\right)}{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right) \\
&\leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \frac{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right) + \|\mathbf{S}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right) \\
&= \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \left(1 + \frac{\|\mathbf{S}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\right)\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right) \\
&\leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left(1 + \left(1 + \frac{\|\mathbf{S}\|_2}{\sigma_k\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\right)\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right).
\end{aligned}$$

□

Note that the relaxation in the final step serves to establish a universal constant across all j, which leads to fewer terms that need bounding when the global SRLU swapping strategy is developed.

Theorem 3.

$$\begin{aligned}
\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_2 &\leq 2\|\mathbf{S}\|_2, \\
\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_F &\leq \|\mathbf{S}\|_F.
\end{aligned}$$

Proof. First

$$
\begin{aligned}
&\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_2 \\
&= \left\| \begin{pmatrix} 0 & \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \right\|_2 \\
&\leq \left\| \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \right\|_2 \\
&\quad + \left\| \begin{pmatrix} \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \right\|_2 \\
&= \left\| \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \right\|_2 \\
&\quad + \left\| \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \begin{pmatrix} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \right\|_2 \\
&\leq 2\|\mathbf{C}\|_2 \\
&= 2\|\mathbf{S}\|_2.
\end{aligned}
$$

Also

$$
\begin{aligned}
&\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_F \\
&= \left\| \begin{pmatrix} \mathbf{Q}_1^L & \mathbf{Q}_2^L \end{pmatrix} \begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \end{pmatrix} \mathbf{A} \begin{pmatrix} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1^U \\ \mathbf{Q}_2^U \end{pmatrix} - \mathbf{Q}_1^L \left(\mathbf{Q}_1^L\right)^T \mathbf{A} \left(\mathbf{Q}_1^U\right)^T \mathbf{Q}_1^U \right\|_F \\
&= \left\| \mathbf{Q}_1^L \left(\mathbf{Q}_1^L\right)^T \mathbf{A} \left(\mathbf{Q}_2^U\right)^T \mathbf{Q}_2^U + \mathbf{Q}_2^L \left(\mathbf{Q}_2^L\right)^T \mathbf{A} \left(\mathbf{Q}_1^U\right)^T \mathbf{Q}_1^U + \mathbf{Q}_2^L \left(\mathbf{Q}_2^L\right)^T \mathbf{A} \left(\mathbf{Q}_2^U\right)^T \mathbf{Q}_2^U \right\|_F \\
&= \left\| \mathbf{Q}_1^L \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \mathbf{Q}_2^U + \mathbf{Q}_2^L \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T \mathbf{Q}_1^U + \mathbf{Q}_2^L \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \mathbf{Q}_2^U \right\|_F \\
&= \left\| \begin{pmatrix} \mathbf{Q}_1^L & \mathbf{Q}_2^L \end{pmatrix} \begin{pmatrix} 0 & \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1^U \\ \mathbf{Q}_2^U \end{pmatrix} \right\|_F \\
&= \left\| \begin{pmatrix} 0 & \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \right\|_F \\
&\leq \left\| \begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T \mathbf{C} \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \right\|_F \\
&= \left\| \begin{pmatrix} \mathbf{Q}_1^L & \mathbf{Q}_2^L \end{pmatrix} \begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T \\ \left(\mathbf{Q}_2^L\right)^T \end{pmatrix} \mathbf{C} \begin{pmatrix} \left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1^U \\ \mathbf{Q}_2^U \end{pmatrix} \right\|_F \\
&= \|\mathbf{C}\|_F \\
&= \|\mathbf{S}\|_F.
\end{aligned}
$$

□

Theorem 4. *SRP produces a rank-k SRLU factorization with*

$$
\begin{aligned}
\|\Pi_1 \mathbf{A} \Pi_2^T - \widehat{\mathbf{L}}\widehat{\mathbf{U}}\|_2 &\leq \gamma \sigma_{k+1}(\mathbf{A}), \\
\|\Pi_1 \mathbf{A} \Pi_2^T - \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)_j \|_2 &\leq \sigma_{j+1}(\mathbf{A}) \left(1 + 2\gamma \frac{\sigma_{k+1}(\mathbf{A})}{\sigma_j(\mathbf{A})}\right),
\end{aligned}
$$

where $j \leq k$ *and* $\gamma = O\left(fk\sqrt{mn}\right)$.

Proof. Note that the definition of α implies

$$\|\mathbf{S}\|_2 \leq \sqrt{(m-k)(n-k)}|\alpha|.$$

From (Pan, 2000):

$$\sigma_{\min}\left(\overline{\mathbf{A}}_{11}\right) \leq \sigma_{k+1}\left(\mathbf{A}\right).$$

Then:

$$\begin{aligned}\sigma_{k+1}^{-1}\left(\mathbf{A}\right) &\leq \|\overline{\mathbf{A}}_{11}^{-1}\|_2 \\ &\leq (k+1)\|\overline{\mathbf{A}}_{11}^{-1}\|_{\max} \\ &\leq (k+1)\frac{f}{|\alpha|}.\end{aligned}$$

Thus

$$|\alpha| \leq f(k+1)\sigma_{k+1}\left(\mathbf{A}\right).$$

The theorem follows by using this result with Theorem 1, with

$$\gamma \leq \sqrt{mn}f(k+1).$$

□

Theorem 5. *Assume the condition of SRLU (equation (2)) is satisfied. Then for $1 \leq j \leq k$:*

$$\frac{\sigma_j\left(\mathbf{A}\right)}{1+\tau\frac{\sigma_{k+1}(\mathbf{A})}{\sigma_j(\mathbf{A})}} \leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right) \leq \sigma_j\left(\mathbf{A}\right)\left(1+\tau\frac{\sigma_{k+1}\left(\mathbf{A}\right)}{\sigma_j\left(\mathbf{A}\right)}\right),$$

where $\tau \leq O\left(mnk^2f^3\right)$.

Proof. After running k iterations of rank-revealing LU,

$$\Pi_1\mathbf{A}\Pi_2^T = \widehat{\mathbf{L}}\widehat{\mathbf{U}} + \mathbf{C},$$

where $\mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S}\end{pmatrix}$, and $\mathbf{S}$ is the Schur complement. Then

$$\begin{aligned}\sigma_j\left(\mathbf{A}\right) &\leq \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right) + \|\mathbf{C}\|_2 \\ &= \sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)\left[1+\frac{\|\mathbf{C}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right)}\right].\end{aligned} \tag{2}$$

For the upper bound:

$$\begin{aligned}\sigma_j\left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}\right) &= \sigma_j\left(\mathbf{A}-\mathbf{C}\right) \\ &\leq \sigma_j\left(\mathbf{A}\right) + \|\mathbf{C}\|_2 \\ &= \sigma_j\left(\mathbf{A}\right)\left[1+\frac{\|\mathbf{C}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right] \\ &= \sigma_j\left(\mathbf{A}\right)\left[1+\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\mathbf{A}\right)}\right].\end{aligned}$$

The final form is achieved using the same bound on γ as in Theorem 4. □

Theorem 6.

$$\begin{aligned}\|\Pi_1\mathbf{A}\Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_2 &\leq 2\gamma\sigma_{k+1}\left(\mathbf{A}\right), \\ \|\Pi_1\mathbf{A}\Pi_2^T - \widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\|_F &\leq \omega\sigma_{k+1}\left(\mathbf{A}\right),\end{aligned}$$

where $\gamma = O\left(fk\sqrt{mn}\right)$ is the same as in Theorem 4, and $\omega = O\left(fkmn\right)$.

Proof. Note that the definition of α implies

$$\|\mathbf{S}\|_F \leq (m-k)(n-k)|\alpha|.$$

The rest follows by using Theorem 3 in a manner similar to how Theorem 4 invoked Theorem 1. □

Theorem 7. *If $\sigma_j^2(\mathbf{A}) > 2\|\mathbf{S}\|_2^2$ then*

$$\sigma_j(\mathbf{A}) \geq \sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right) \geq \sigma_j(\mathbf{A})\sqrt{1-2\gamma\left(\frac{\sigma_{k+1}(\mathbf{A})}{\sigma_j(\mathbf{A})}\right)^2},$$

where $\gamma = O\left(mnk^2f^2\right)$, and f is an input parameter controlling a tradeoff of quality vs. speed as before.

Proof. Perform QR and LQ decompositions $\widehat{\mathbf{L}} = \mathbf{Q}_L\mathbf{R}_L =: \begin{pmatrix}\mathbf{Q}_1^L & \mathbf{Q}_2^L\end{pmatrix}\begin{pmatrix}\mathbf{R}_{11}^L & \mathbf{R}_{12}^L \\ & \mathbf{R}_{22}^L\end{pmatrix}$ and $\widehat{\mathbf{U}} = \mathbf{L}_U\mathbf{Q}_U =: \begin{pmatrix}\mathbf{L}_{11}^U & \\ \mathbf{L}_{21}^U & \mathbf{L}_{22}^U\end{pmatrix}\begin{pmatrix}\mathbf{Q}_1^U \\ \mathbf{Q}_2^U\end{pmatrix}$. Then

$$\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}} = \mathbf{Q}_1^L\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\mathbf{Q}_1^U.$$

Note that

$$\begin{aligned}
\mathbf{A}^T\mathbf{Q}_2^L &= \left(\widehat{\mathbf{L}}\widehat{\mathbf{U}}+\mathbf{C}\right)^T\mathbf{Q}_2^L \\
&= \left(\mathbf{Q}_1^L\mathbf{R}_{11}^L\mathbf{L}_{11}^U\mathbf{Q}_1^U+\mathbf{C}\right)^T\mathbf{Q}_2^L \\
&= \left(\mathbf{Q}_1^U\right)^T\left(\mathbf{L}_{11}^U\right)^T\left(\mathbf{R}_{11}^L\right)^T\left(\mathbf{Q}_1^L\right)^T\mathbf{Q}_2^L+\mathbf{C}^T\mathbf{Q}_2^L \\
&= \mathbf{C}^T\mathbf{Q}_2^L. \qquad (3)
\end{aligned}$$

Analogously

$$\mathbf{A}\left(\mathbf{Q}_2^U\right)^T = \mathbf{C}\left(\mathbf{Q}_2^U\right)^T. \qquad (4)$$

Then

$$
\begin{aligned}
\sigma_j\left(\mathbf{A}\right) &= \sigma_j\begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T\mathbf{A}\left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \\
&= \sigma_j\begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \\
&= \sqrt{\lambda_j\left(\begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \end{pmatrix}^T \begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T \end{pmatrix}\right)} \\
&= \sqrt{\begin{aligned}&\lambda_j\Big(\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right) \\ &+\left(\left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\left(\left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)\Big)\end{aligned}} \\
&\leq \sqrt{\begin{aligned}&\lambda_j\left(\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)\right) \\ &+\left\|\left(\left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_2^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)\right\|_2\end{aligned}} \\
&\leq \sqrt{\lambda_j\left(\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)\right)+\|\mathbf{C}\|_2^2} \\
&= \sqrt{\lambda_j\left(\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T \quad \left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\right)+\|\mathbf{C}\|_2^2} \\
&= \sqrt{\lambda_j\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\right)^T+\left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\right)+\|\mathbf{C}\|_2^2} \\
&\leq \sqrt{\lambda_j\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\right)^T\right)+\left\|\left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{C}\left(\mathbf{Q}_2^U\right)^T\right)^T\right\|_2+\|\mathbf{C}\|_2^2} \\
&\leq \sqrt{\lambda_j\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\right)^T\right)+2\|\mathbf{C}\|_2^2} \\
&\leq \sqrt{\sigma_j^2\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\right)+2\|\mathbf{C}\|_2^2} \\
&= \sqrt{\sigma_j^2\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)+2\|\mathbf{C}\|_2^2} \\
&= \sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)\sqrt{1+2\left(\frac{\|\mathbf{C}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)}\right)^2} \\
&= \sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)\sqrt{1+2\left(\frac{\|\mathbf{S}\|_2}{\sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)}\right)^2}.
\end{aligned}
$$

Solve for $\sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right)$ for the lower bound. The upper bound:

$$
\begin{aligned}
\sigma_j\left(\mathbf{A}\right) &= \sigma_j\begin{pmatrix} \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_2^U\right)^T \\ \left(\mathbf{Q}_2^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T & \left(\mathbf{Q}_2^L\right)^T\mathbf{A}\left(\mathbf{Q}_2^U\right)^T \end{pmatrix} \\
&\geq \sigma_j\left(\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\right) \\
&= \sigma_j\left(\mathbf{Q}_1^L\left(\mathbf{Q}_1^L\right)^T\mathbf{A}\left(\mathbf{Q}_1^U\right)^T\mathbf{Q}_1^U\right) \\
&= \sigma_j\left(\widehat{\mathbf{L}}\mathbf{M}\widehat{\mathbf{U}}\right).
\end{aligned}
$$

□

5. Analysis of the Choice of Block Size for SRLU

A heuristic for choosing a block size for TRLUCP is described here, which differs from standard block size methodologies for the LU decomposition. Note that a key difference of SRLU and TRLUCP from previous works is the size of the random projection: here the size is relative to the block size, not the target rank k ($2pmn$ flops for TRLUCP versus the significantly larger $2kmn$ for others). This also implies a change to the block size also changes the flop count, and, to our knowledge, this is the first algorithm where the choice of block size affects the flop count. For problems where LAPACK chooses $b = 64$, our experiments have shown block sizes of 8 to 20 to be optimal for TRLUCP. Because the ideal block size depends on many parameters, such as the architecture of the computer and the costs for various arithmetic, logic, and memory operations, guidelines are sought instead of an exact determination of the most efficient block size. To simplify calculations, only the matrix multiplication operations are considered, which are the bottleneck of computation. Using standard communication-avoiding analysis, a good block size can be calculated with the following model: let M denote the size of cache, f and m the number of flops and memory movements, and t_f and t_m the cost of a floating point operation and the cost of a memory movement. We seek to choose a block size to minimize the total calculation time T modeled as

$$T = f \cdot t_f + m \cdot t_m.$$

Choosing $p = b + c$ for a small, fixed constant c, and minimizing implies

$$\begin{aligned} T &= \left[(m+n-k)\left(k^2 - kb\right) - \frac{4}{3}k^3 + 2bk^2 - \frac{2}{3}b^2 k\right] \cdot t_f \\ &\quad + \left[(m+n-k)\left(\frac{k^2}{b} - k\right) - \frac{4}{3}\frac{k^3}{b} + 2k^2 - \frac{2}{3}bk\right] \cdot \frac{M}{\left(\sqrt{b^2+M} - b\right)^2} \cdot t_m. \end{aligned}$$

Given hardware-dependent parameters M, t_f, and t_m, a minimizing b can easily be found.

This result is derived as follows: we analyze blocking by allowing different block sizes in each dimension. For matrices $\mathbf{\Omega} \in \mathbb{R}^{p \times m}$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$ consider blocking in the form

$$\mathbf{\Omega} \cdot \mathbf{R} = {}^{s}\overset{\ell}{\begin{pmatrix} * & * & * \\ * & * & * \end{pmatrix}} \cdot {}^{\ell}\overset{b}{\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}}.$$

Then a current block update requires cache storage of

$$s\ell + \ell b + sb \leq M.$$

Thus we will constrain

$$\ell \leq \frac{M - sb}{s + b}.$$

The total runtime T is

$$\begin{aligned} T &= 2pmn \cdot t_f + \left(\frac{p}{s}\right)\left(\frac{m}{\ell}\right)\left(\frac{n}{b}\right)(s\ell + \ell b + sb) \cdot t_m \\ &= 2pmn \cdot t_f + pmn\left(\frac{s+b}{sb} + \frac{1}{\ell}\right) \cdot t_m \\ &\geq 2pmn \cdot t_f + pmn\left(\frac{s+b}{sb} + \frac{s+b}{M - sb}\right) \cdot t_m \\ &= 2pmn \cdot t_f + pmnM\left(\frac{s+b}{sb\,(M - sb)}\right) \cdot t_m \\ &=: 2pmn \cdot t_f + pmnML\,(s,b,M) \cdot t_m. \end{aligned}$$

Given $\mathbf{\Omega}$ and $\mathbf{A}$, changing the block sizes has no effect on the flop count. Optimizing $L\,(s,b,M)$ over s yields

$$s^2 + 2sb = M.$$

By symmetry

$$b^2 + 2sb = M.$$

Note, nevertheless, that $s \leq p$ by definition. Hence

$$s^* = \min\left(\sqrt{\frac{M}{3}}, p\right),$$

and

$$b^* = \max\left(\sqrt{\frac{M}{3}}, \sqrt{p^2 + M} - p\right).$$

These values assume

$$\ell^* = \frac{M - sb}{s + b} = \max\left(\sqrt{\frac{M}{3}}, \sqrt{p^2 + M} - p\right) = b^*.$$

This analysis applies to matrix-matrix multiplication where the matrices are fixed and the leading matrix is short and fat or the trailing matrix is tall and skinny. As noted above, nevertheless, the oversampling parameter p is a constant amount larger than the block size used during the LU factorization. The total initialization time is

$$\begin{aligned} T^{\text{init}} &= 2pmn \cdot t_f + pmnM\left(\frac{s+b}{sb\,(M - sb)}\right) \cdot t_m \\ &= 2pmn \cdot t_f + mn \cdot \min\left(3\sqrt{3}\frac{p}{\sqrt{M}}, \frac{M}{\left(\sqrt{p^2 + M} - p\right)^2}\right) \cdot t_m. \end{aligned}$$

We next choose the parameter b used for blocking the LU factorization, where $p = b + O(1)$. The cumulative matrix multiplication (DGEMM) runtime is

$$\begin{aligned} T^{\text{DGEMM}} &= \sum_{j=b:b:k-b} \left[2jb(m-j) + 2jb(n-j-b)\right] \cdot t_f + 2\left[j(m-j) + j(n-j-b)\right] \frac{M}{\left(\sqrt{b^2 + M} - b\right)^2} \cdot t_m \\ &= \left[(m+n-k)\left(k^2 - kb\right) - \frac{4}{3}k^3 + 2bk^2 - \frac{2}{3}b^2k\right] \cdot t_f + \\ &\quad + \left[(m+n-k)\left(\frac{k^2}{b} - k\right) - \frac{4}{3}\frac{k^3}{b} + 2k^2 - \frac{2}{3}bk\right] \frac{M}{\left(\sqrt{b^2 + M} - b\right)^2} \cdot t_m \\ &=: N_f^{\texttt{DGEMM}} \cdot t_f + N_m^{\texttt{DGEMM}} \cdot t_m. \end{aligned}$$

The methodology for choosing a block size is compared to other choices of block size in Figure 1. Note that LAPACK generally chooses a block size of 64 for these matrices, which is suboptimal in all cases, and can be up to twice as slow. In all of the cases tested, the calculated block size is close to or exactly the optimal block size.

6. Additional Notes and Experiments

6.1. Efficiency of SRLU

Not only is the TRLUCP component efficient compared with other low-rank approximation algorithms, but also it becomes arbitrarily faster than the standard right-looking LU decomposition as the data size increases. Because the LU decomposition is known to be efficient compared to algorithms such as the SVD (Demmel, 1997), comparing TRLUCP to right-looking LU exemplifies its efficiency, even though right-looking LU is not a low-rank approximation algorithm.

In Figure 2, TRLUCP is benchmarked against truncated right-looking LU (called using a truncated version of the LAPACK library DGETRF). Experiments are run on random matrices, with the x-axis reflecting the approximate number of floating point operations. Also plotted is the theoretical peak performance, which illustrates that TRLUCP is a highly efficient algorithm.

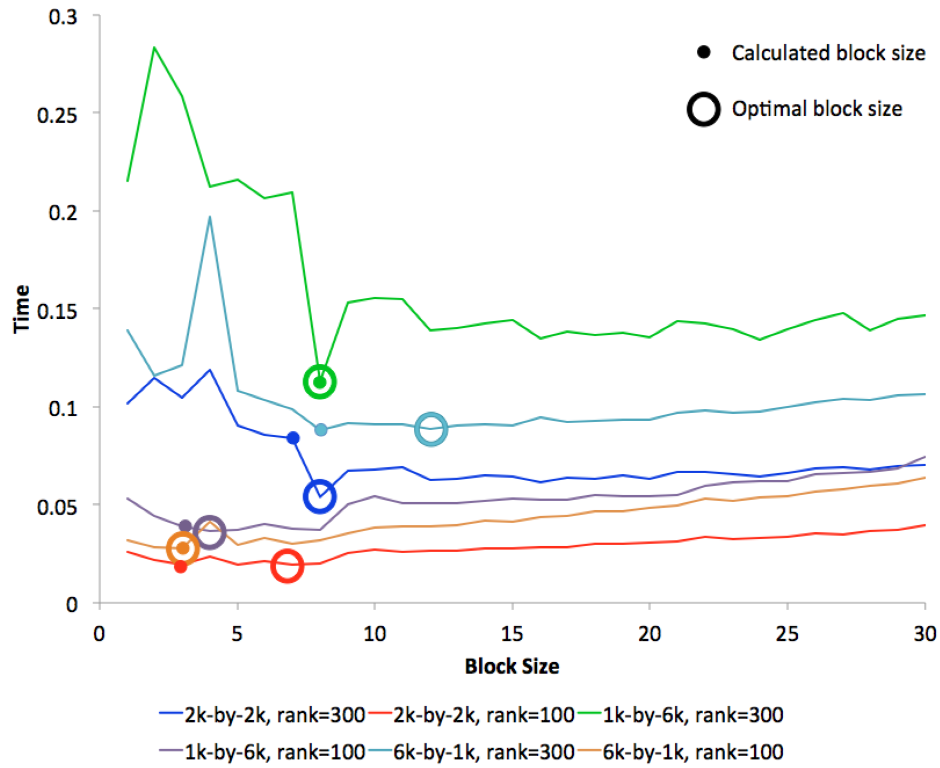

Figure 1. Benchmarking TRLUCP with various block sizes on random matrices of different sizes and truncation ranks.

6.2. Sparsity-Preservation

Table 1 contains additional sparsity-preservation experiments on matrices from (David & Hu, 2011).

Table 1. Sparsity preservation experiments of various sparse, non-symmetric data matrices. The SRLU factorization is computed to 20% of full-rank. The Full SRLU factorization is the SRLU factorization with the Schur complement. LU and SVD are the standard LU and SVD decompositions. The SRLU relative error is the Frobenius-norm relative error of the SRLU factorization, which has a target rank that is 20 percent of the matrix rank.

Matrix Description			Nonzeros (rounded) In:				
Name	Application	Nonzeros	SRLU	Full SRLU	LU	SVD	SRLU Rel. Error
`oscil_dcop`	Circuits	1,544	1,570	4.7K	9.7K	369K	1.03e-3
`g7jac020`	Economics	42,568	62.7K	379K	1.7M	68M	1.09e-6
`tols1090`	Fluid dynamics	3,546	2.2K	4.7K	4.6K	2.2M	1.18e-4
`mhd1280a`	Electromagnetics	47,906	184K	831K	129K	3.3M	4.98e-6

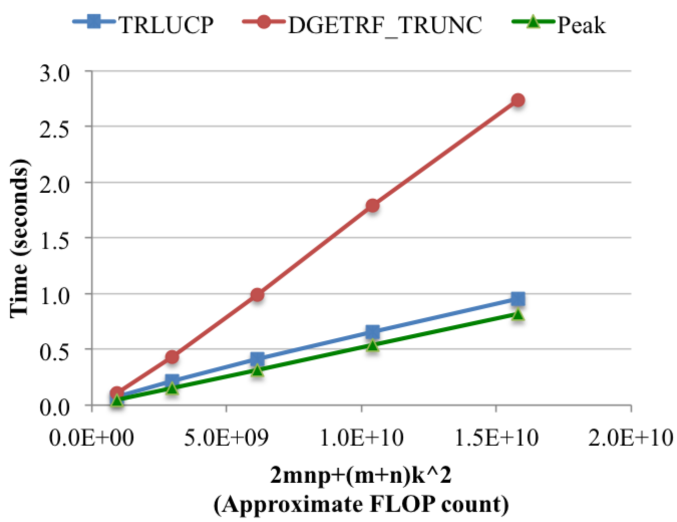

Figure 2. Computation time of TRLUCP versus the efficiency LU decomposition.

6.3. Online Data Processing

In many applications, reduced weight is given to old data. In this context, multiplying the matrices $\mathbf{U}_{11}$, $\mathbf{U}_{12}$ and $\mathbf{S}$ by some scaling factor less than 1 before applying spectrum-revealing pivoting will reflect the reduced importance of the old data.

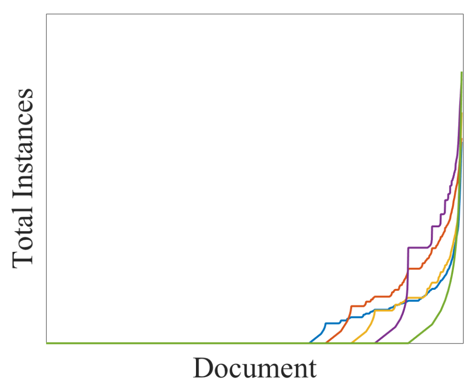

Figure 3. The cumulative uses of the top five most commonly used words in the Enron email corpus after reordering.

The cumulative usages of the top 5 words in the Enron email corpus (after reordering) is plotted in Figure 3. For the online updating experiment with the Enron email corpus, the covariance matrix of the top five most frequent words is

$$\begin{array}{l} \\ \text{power} \\ \text{company} \\ \text{energy} \\ \text{market} \\ \text{california} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{power} & \text{company} & \text{energy} & \text{market} & \text{california} \end{array} \\ \begin{pmatrix} 1 & 0.40 & 0.81 & 0.51 & 0.78 \\ 0.40 & 1 & 0.42 & 0.57 & 0.28 \\ 0.81 & 0.42 & 1 & 0.51 & 0.78 \\ 0.51 & 0.57 & 0.51 & 1 & 0.48 \\ 0.78 & 0.23 & 0.78 & 0.48 & 1 \end{pmatrix} \end{array}.$$

References

David, T. A. and Hu, Y. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38: 1–25, 2011. URL http://www.cise.ufl.edu/research/sparse/matrices.

Demmel, J. *Applied Numerical Linear Algebra*. SIAM, 1997.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Golub, G. H. and van Loan, C. F. *Matrix Computations*. JHU Press, 4th edition, 2013.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Melgaard, C. and Gu, M. Gaussian elimination with randomized complete pivoting. *CoRR*, abs/1511.08528, 2015.

Miranian, L. and Gu, M. Stong rank revealing lu factorizations. *Linear Algebra and its Applications*, 367:1–16, 2003.

Pan, C.-T. On the existence and computation of rank-revealing lu factorizations. *Linear Algebra and its Applications*, 316 (1):199–222, 2000.