Low-Rank Incremental Methods for Computing Dominant Singular Subspaces

Christopher G. Baker\textsuperscript{1,2} Kyle A. Gallivan\textsuperscript{1} Paul Van Dooren\textsuperscript{3}

\textsuperscript{1}School of Computational Science
Florida State University

\textsuperscript{2}Computer Science Research Institute
Sandia National Laboratories

\textsuperscript{3}Department of Mathematical Engineering
Université catholique de Louvain

April 2008 / Copper Mountain
Acknowledgments

Funding

- NSF Grants ACI0324944 and CCR9912415
- School of Computational Science, FSU
- Baker was funded as a student at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy; contract/grant number: DE-AC04-94AL85000.
Outline

1. **Introduction**
   - Singular Value Decomposition and its Computation

2. **Low-Rank Incremental Methods**
   - Overview of Low-Rank Methods
   - Operation of Low-Rank Methods

3. **New Analysis**
   - Link to Iterative Eigensolvers
   - Multi-pass Approaches
Outline

1 Introduction
   - Singular Value Decomposition and its Computation

2 Low-Rank Incremental Methods
   - Overview of Low-Rank Methods
   - Operation of Low-Rank Methods

3 New Analysis
   - Link to Iterative Eigensolvers
   - Multi-pass Approaches
The Singular Value Decomposition

**Definition**

The singular value decomposition of an $m \times n$ matrix $A$ is

$$ A = U \Sigma V^T = [U_1 \quad U_2] \begin{bmatrix} \Sigma^T \\ 0 \end{bmatrix} V^T = U_1 \Sigma V^T $$

with orthogonal $U$, $V$; $\Sigma$ diagonal with non-decreasing, non-negative entries.

**Terminology**

The columns of $U_1$ and $V$ are left and right singular vectors. Diagonal entries of $\Sigma$ are singular values. Largest singular values are dominant, as are their corresponding singular vectors.
Dominant SVD

Applications
Many applications require only the dominant singular triplets, e.g., PCA, KLT, POD.

Computation
Numerous approaches for computing the dominant SVD:
- compute the full SVD and truncate the unneeded part;
- transform to an eigenvalue problem, compute relevant eigenvectors via iterative eigensolver, back-transform;
- use iterative solver to compute dominant SVD:
  - Riemannian optimization gives many approaches [ABG2007]
  - Non-linear equation $\rightarrow$ JD-SVD [Hochstenbach2000]
  - Low-rank incremental methods
Outline

1. Introduction
   - Singular Value Decomposition and its Computation

2. Low-Rank Incremental Methods
   - Overview of Low-Rank Methods
   - Operation of Low-Rank Methods

3. New Analysis
   - Link to Iterative Eigensolvers
   - Multi-pass Approaches
Overview of Low-Rank Methods

Incremental SVD

Motivation
In many applications, the production of the matrix $A$ happens incrementally. This has motivated numerous methods for SVD updating. [e.g., Businger; Bunch, Nielson]

Benefits
- Latency in producing new columns of $A$ can be amortized in the SVD update
- “Online” SVD is useful/necessary in some applications

Drawbacks
- Computation, storage is expensive
- Still computes the full SVD
Low-Rank Incremental SVD

More efficient approach
The low-rank incremental SVD methods follow the example of the incremental SVD methods, but track only a low-dimensional subspace.

History
Repeatedly and independently described in the literature:
- 2001: Chahlaoui, Gallivan, Van Dooren: “Recursive SVD”
- 2002: Brand: “Incremental SVD”
- 2004: Baker, Gallivan, Van Dooren
Kernel Step

Given a matrix $A$ with factorization $A = U\Sigma V^T$, compute updated factorization of augmented matrix $[A \ A_+]$:

$$U_+\Sigma_+V_+^T = [A \ A_+] = [U\Sigma V^T \ A_+]$$

Incremental Algorithm

- Partition $A = [A_1 \ A_2 \ \ldots \ A_b]$
- Initialize $A_1 = U_1\Sigma_1 V_1^T$
- for $i = 2, \ldots, b$
  - Update factorization:
    $$U_i\Sigma_i V_i^T = [U_{i-1}\Sigma_{i-1} V_{i-1}^T \ A_i]$$
Low-rank Incremental SVD Operation

- Perform a low-rank version of the incremental SVD

**Kernel Step**

Given a factorization $U \Sigma V^T$ and columns $A_+$, compute dominant SVD $U_+ \Sigma_+ V_+^T \approx [U \Sigma V^T \ A_+]$.

**Heuristic motivation**

Approximation of an approximation is an approximation, right?

\[
U_1 \Sigma_1 V_1^T \approx A_1 \\
U_2 \Sigma_2 V_2^T \approx [U_1 \Sigma_1 V_1^T \ A_2] \approx [A_1 \ A_2] \\
\cdots \\
U_b \Sigma_b V_b^T \approx \approx A
\]
The algorithm

Given the factorization $U\Sigma V^T$ and new columns $A$:

1. Expand the factorization via Gram-Schmidt:

   $$\begin{bmatrix} U \Sigma V^T & A \end{bmatrix} = \hat{Q} \hat{R} \hat{W}^T = \begin{bmatrix} U & Q \end{bmatrix} \begin{bmatrix} \Sigma & R_2 \\ 0 & R_3 \end{bmatrix} \begin{bmatrix} V & 0 \end{bmatrix}^T$$

2. Compute transformations $G_u, G_v$ that decouple the singular subspaces in $\hat{R}$:

   $$G_u^T \hat{R} G_v = \begin{bmatrix} \hat{R}_1 & 0 \\ 0 & \hat{R}_2 \end{bmatrix}, \quad \sigma(\hat{R}_1) > \sigma(\hat{R}_2)$$

3. Insert $G_u, G_v$ into expanded factorization:

   $$\bar{Q} \hat{R} \bar{W}^T = (\hat{Q} G_u)(G_u^T \hat{R} G_v)(G_v^T \hat{W}^T) = \hat{Q} \hat{R} \hat{W}^T$$

4. **Truncate** the dominated part of the factorization.
Cost/benefit

Benefits:
- Requires only a single pass through $A$
- Exploits latency in producing/retrieving columns of $A$
- Flop count is linear: $O(mnk)$
  - Leading coefficient varies according to requirements on structure of intermediate factorizations
  - Method from [Baker2004] requires $10mnk$ flops
- Storage of $O(mk + nk)$ is minimal

Drawback:
- Factorization is inexact due to truncation
- Previous literature makes no suggestion for improving factorization
Outline

1. Introduction
   - Singular Value Decomposition and its Computation

2. Low-Rank Incremental Methods
   - Overview of Low-Rank Methods
   - Operation of Low-Rank Methods

3. New Analysis
   - Link to Iterative Eigensolvers
   - Multi-pass Approaches
New interpretation

Take an orthogonal matrix $D = [D_1 \ldots D_b]$. Consider the low-rank incremental SVD of $AD = [AD_1 \ldots AD_b]$.

A locally optimal solver

At iterate $U_i$, $\Sigma_i$, $V_i$, the algorithm inputs $AD_{i+1}$ and chooses $V_{i+1}$ which maximizes $\text{trace} \left( V^T A^T A V \right)$ over all orthonormal $V$ in $\text{span}(V_i [\begin{array}{c} D_i \\ \ldots \\ D_{i+1} \end{array}])$.

Implications

- IncSVD of $A$ (i.e., $D = I$) implicitly performs coordinate ascent, optimization-based eigensolve of $A^T A$
- Choice of $D$ gives a hook to affect the performance.
Multi-pass Approaches

**Multi-pass Method**

**Targeted initialization**
- Given approximate right singular vectors \( \hat{V} \), choose \( D \):
  \[
  D = [\hat{V} \ D_2 \ \ldots \ \ D_b]
  \]
- The iteration immediately captures the information in \( \hat{V} \) and improves thereafter.
- Use to **restart** the algorithm if \( A \) is still available.
- Can be done in a pass-efficient manner.

**Better choices for \( D \)?**
- If \( D_i \) is exact dominant right singular vectors, then incremental algorithm is exact.
- Speedup convergence by inserting **gradient** information into \( D \).
The decoupling technique makes explicit the effort necessary to implement a member of this family of methods.

The novel analysis shows the link to an iterative, optimization-based eigensolver approach.

This analysis allows the description of methods which can exploit multiple passes through $A$.

Convergence proof with rate of convergence is forthcoming.

“Killer apps” wanted.