SAND2007-7209C

A Fast Method for Computing Principal Components Analysis on Large Data Sets

EAS 2007 Paper 420

11/14/07

Mark H. Van Benthem and Michael R. Keenan

Sandia National Laboratories, Albuquerque, NM

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Overview

- PCA and eigenanalysis methods in chemometrics
- The power method and orthogonal iteration
- Performance comparisons of various methods
- Other computational considerations
- Results and Conclusions
- Summary

Sandia National Laboratories

Motivation

- PCA is often the first step in many chemometric analysis schemes
- Data sets are growing, growing, growing!
- Chemists (and everyone else) need faster methods for analyzing their data
- Simple programming is always desirable

PCA in Chemometrics

- Given a matrix containing data, D, as a first step in many analyses we want principal components

$$D \cong TP^T$$

- Such that T and P are orthogonal basis sets, that is a reduced dimensional representation of D, with ordered maximized variance
- After computing T and P, these can be used in place of D for various other non-orthogonal factorization methods, such as MCR

Sandia National Laboratories

Computing Principal Components

- Singular value decomposition (SVD)
  - Finds left & right singular vectors & singular values
  - Best for ill-conditioned matrices
  - Slow and memory intensive
- Nonlinear Iterative Partial Least Squares (NIPALS)
  - Finds the first singular vector of the matrix
    - Performed iteratively with matrix deflation on each step
    - Find first singular vector of each successive residual matrix
  - Fast & easy to code
- Kernel method (Eigenanalysis)
  - Orthogonal matrix factorization of a square matrix
  - Find singular vectors (loadings) of D^TD (or DD^T)
  - Project data into singular vector space to obtain scores
  - Can be very fast and easy to code

Sandia National Laboratories

Solving the Symmetric Eigenvalue Problem

- Compute the cross product D^TD or DD^T
  - Rule 1: ALWAYS compute the cross product for the small side.
  - Example: For D with dimensions of 25×100, compute the 25×25 matrix, D^TD
- Compute the eigenvectors of the cross product
  - Rule 2: Compute only eigenvectors you need to use
  - Example: For data in D (above) with pseudorank 5, compute only 5 eigenvectors, not all 25
- Least squares estimate of large side eigenvectors
  - For $D \cong TP^T$, then $T\Sigma T^T \cong D^TD$ and $P^T \cong T^TD$

Sandia National Laboratories

The Power Method

- **Finds only the first eigenvector of symmetric matrix**
- **Same basic method employed by NIPALS**
- **Method used by Google's page rank algorithm**
- **Reference:**
  - **Golub and Van Loan; Matrix Computations. 3rd ed. Johns Hopkins Univ. Press, Baltimore, 1996**

$$\mathbf{t}_n = \frac{\mathbf{A}\mathbf{t}_{n-1}}{\mathbf{t}_n^{\mathrm{T}}\mathbf{t}_n}$$

Algorithm:

pick a suitable starting vector $\mathbf{t}_0$

for $n = 1,\ 2,\ 3, \ldots$

$$\mathbf{q}_n = \mathbf{A}\mathbf{t}_{n-1}$$

$$\mathbf{t}_n = \frac{\mathbf{q}_n}{\left\|\mathbf{q}_n\right\|_2}$$

end

execute until convergence

Orthogonal Iteration

- **Finds only the first r eigenvectors of symmetric matrix**
- **For r = 1, identical to power method**
- **Converges at rate proportional to the ratio of the r^{th} to $p+1^{th}$ (some $p>r$) eigenvalue to the n^{th} power**
- **Reference:**
  - **Golub and Van Loan; Matrix Computations. 3rd ed. Johns Hopkins Univ. Press, Baltimore, 1996**

Algorithm:

pick a suitable starting

orthonormal r-column matrix $\mathbf{T}_0$

for $n = 1,\ 2,\ 3, \ldots$

$$\mathbf{Q}_n = \mathbf{A}\mathbf{T}_{n-1}$$

$$\mathbf{T}_n \mathbf{\Sigma} = \mathbf{Q}_n \quad \left(orthogonalize\ \mathbf{Q}_n \right)$$

end

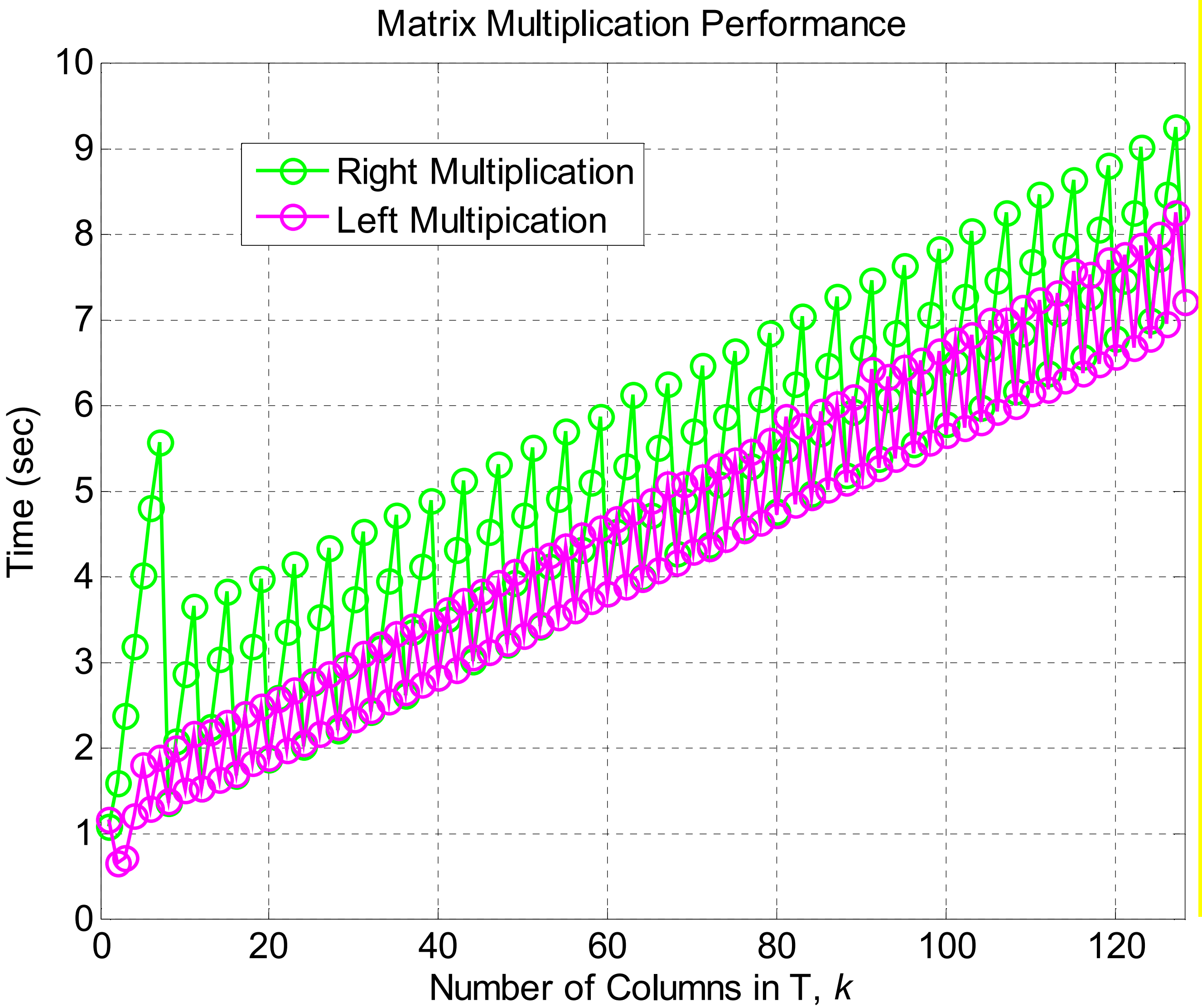
execute until convergence

-We pick radix-two factor size larger than our number of factors during iterations. 5→8, 8→16 (exception to Rule 2)

-Orthogonalize with SVD.

Sandia National Laboratories

Why Use More Factors and Radix-2?



Multiplying two matrices A (2111^2) & T ($2111\times n$), 100 iterations.

- Green AT
- Pink T^TA

Due to cache memory tiling and register tiling*

Convergence rate is proportional to $(\lambda_r/\lambda_{p+1})^k$, so having a noise eigenvalue last keeps ratio >> 1

*Yotov, et al., Proc. of the IEEE; Feb. 2005; vol.93, no.2, p.358-86

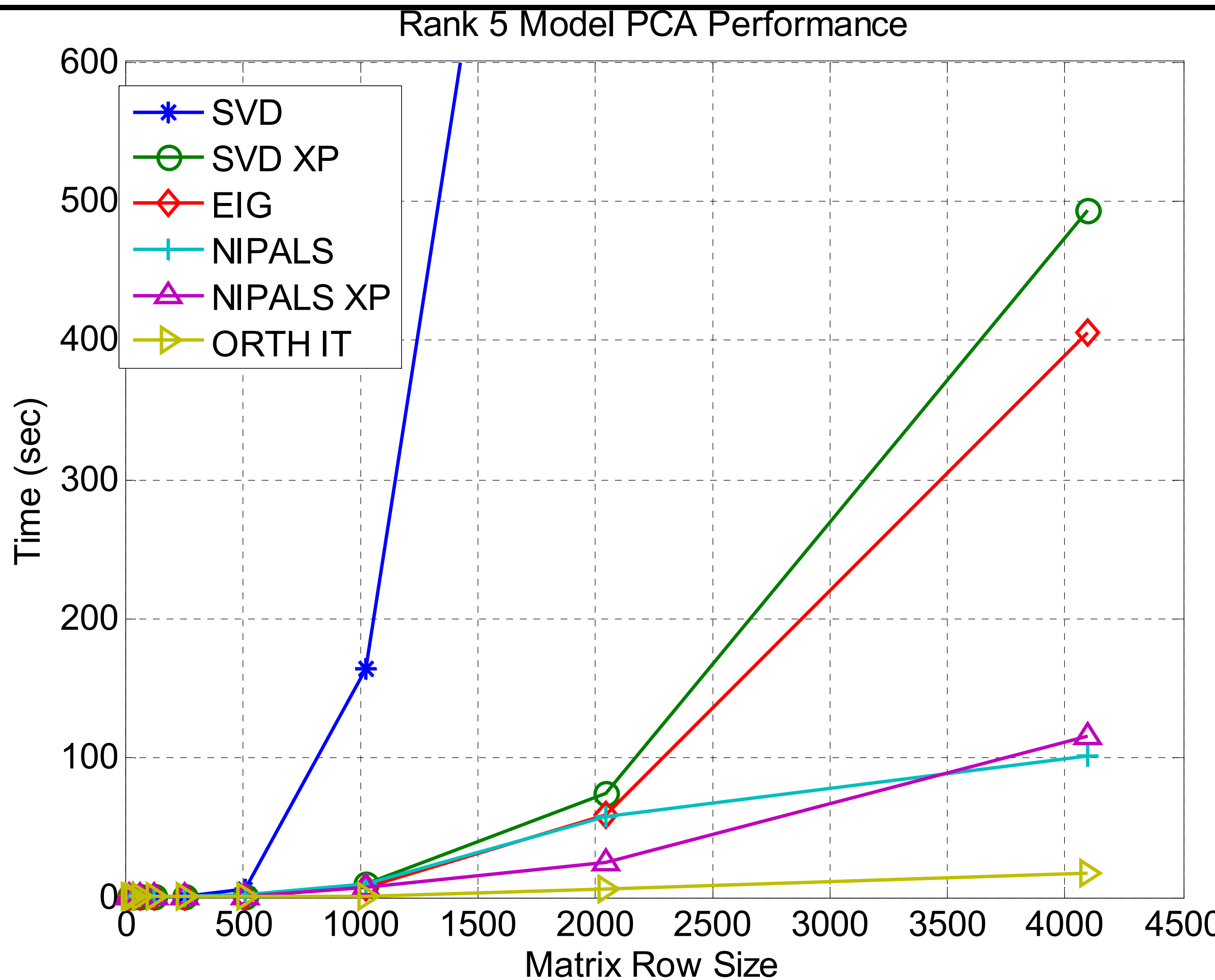
Sandia National Laboratories

Algorithm Comparisons

- Compare MATLAB® functions SVD, EIG and our versions of NIPALS with our Orthogonal Iteration
  - Data: Simulated 5, 10 and 15 component models with Gaussian noise
  - Matrix sizes: 16×32, 32×64, 64×128, 128×264, 264×512, 512×1024, 1024×2048, 2048×4096, 4096×4096
  - NIPALS and SVD use full data set
    - Also ran both with symmetric cross-product matrices
  - EIG and Orthogonal Iteration use cross-products
  - Times to compute cross-product matrices are included in results

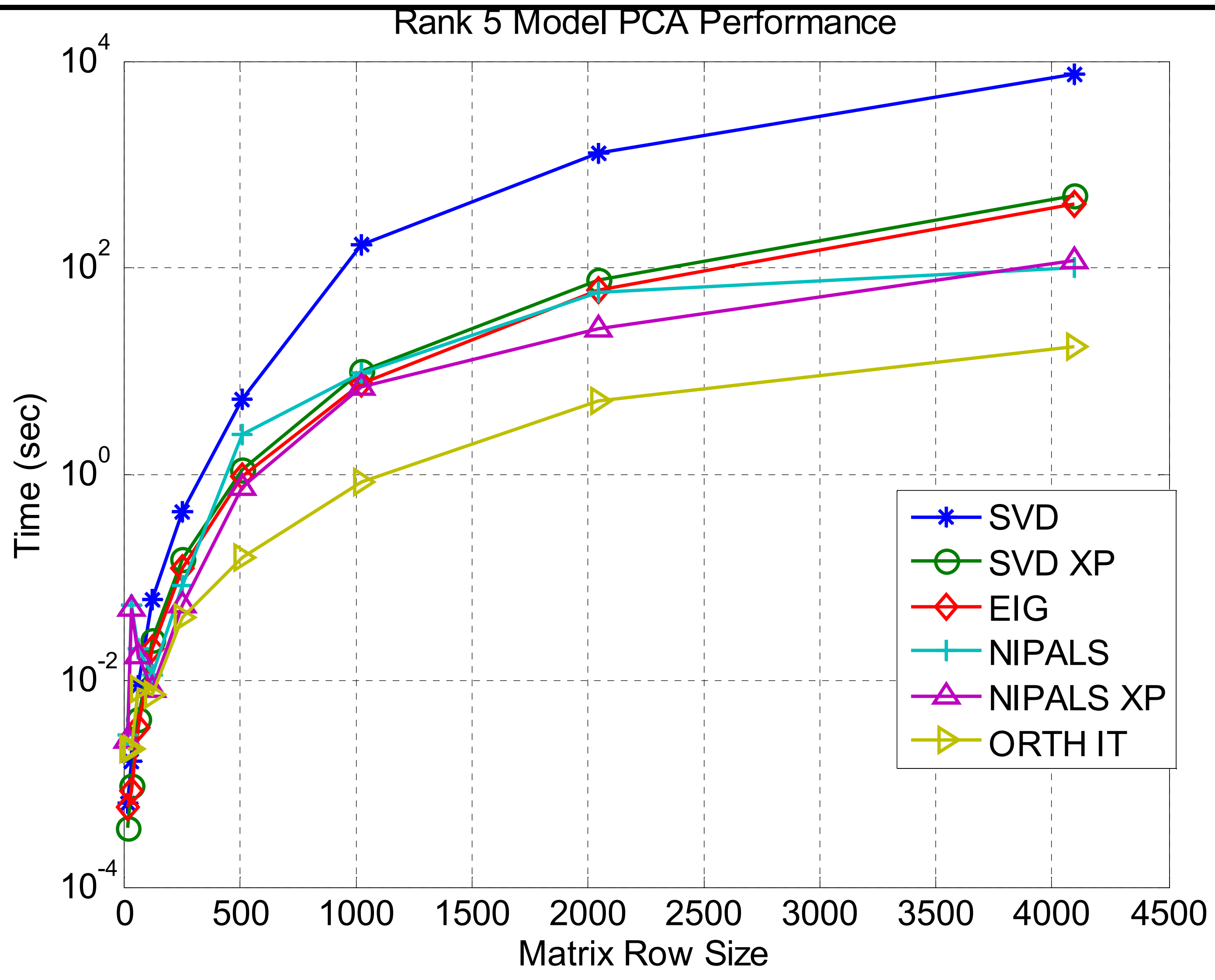
Sandia National Laboratories

Performance of Five Factor Data



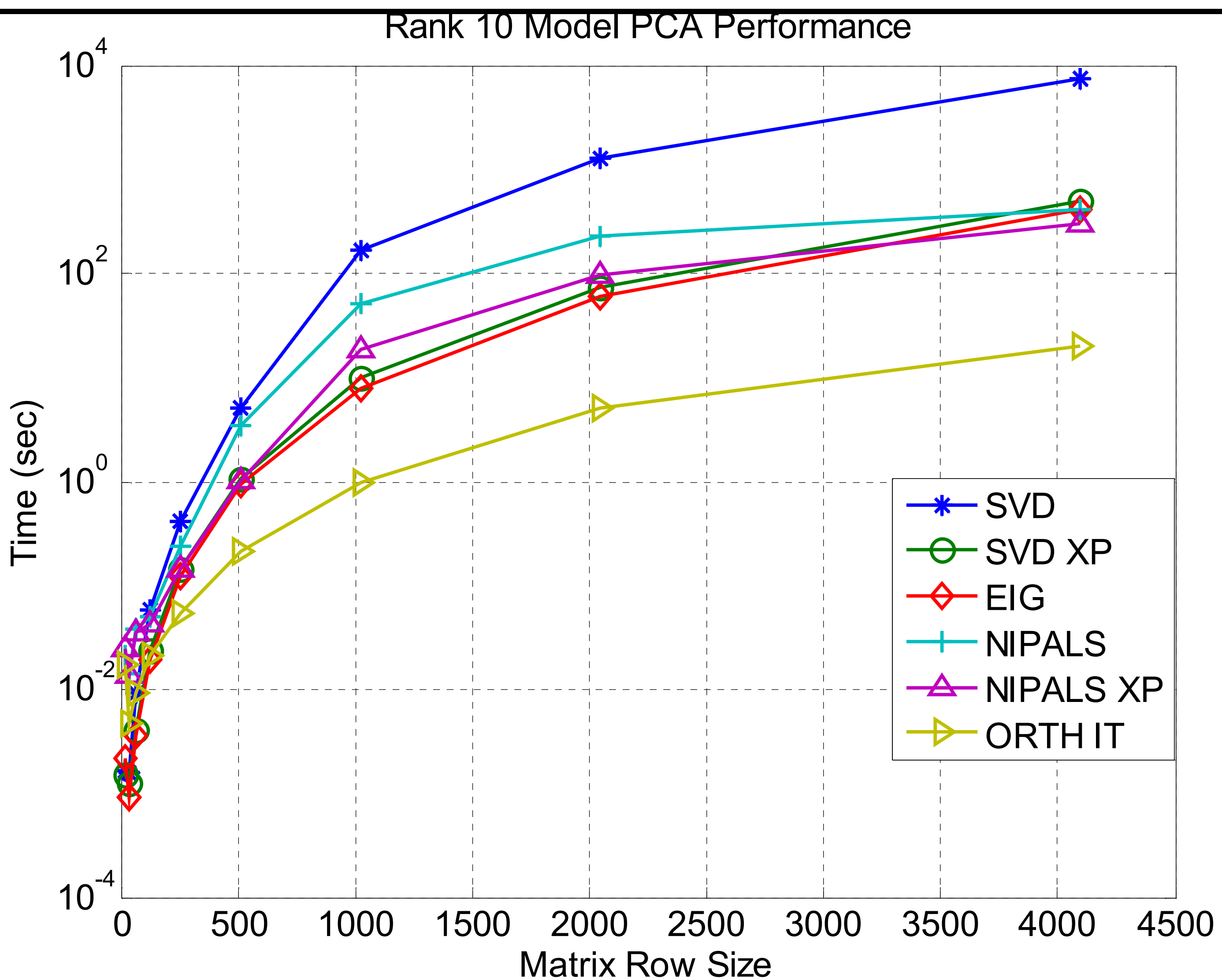
Sandia National Laboratories

Log-scale Five Factor Data Performance

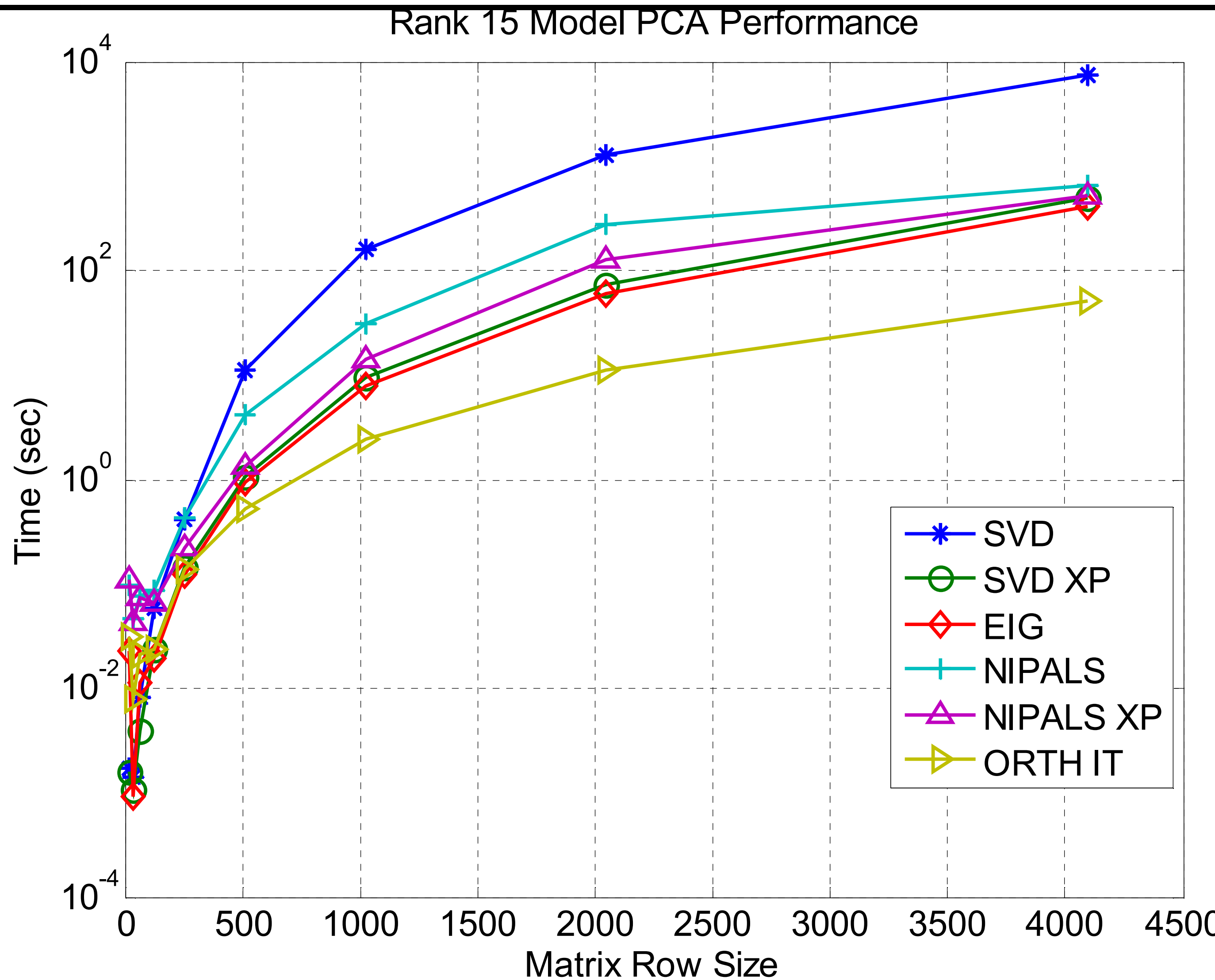


Sandia National Laboratories

Log-scale Ten Factor Data Performance



Log-scale Fifteen Factor Data Performance



Results and Conclusions

- Orthogonal iteration is a fast, well-established method for computing a limited eigenvector basis
  - Fast and accurate PCA
- It is easy to program and implement in MATLAB®
  - Very few lines of code involved in actual algorithm
- Computational considerations should always figure into algorithm implementation
  - Matrix-matrix multiplication algorithms are highly scalable
    - They work very well on multiprocessor systems
    - Better scaling properties than matrix-vector multiplication

Sandia National Laboratories

Summary

- PCA and eigenanalysis methods in chemometrics
  - Ubiquitous for initial data reduction
- The power method and orthogonal iteration
  - Simple iterative algorithms for decomposing symmetric matrices
- Other computational considerations
  - Get a rough understanding of how computations are performed before you start
- Performance comparisons of various methods
  - Orthogonal iteration is a clear winner!

Sandia National Laboratories