Trace Minimization Algorithms

Yu Hong Yeung[1], Xin Cheng[1], Di Jin[1]

[1] *Department of Computer Science*

Problem
Given the large sparse symmetric eigenvalue problem $Ax = \lambda Bx$, where A and B are symmetric matrices of order $n = 10^6$, and B is symmetric positive definite. Use the Trace Minimization algorithms: TraceMIN and TraceMIN-Davidson for obtaining the smallest $p \ll n$ eigenpairs, and use TraceMIN for computing **all** the eigenpairs belonging to an interior interval $[a,b]$.

Contents

Introduction

The generalized eigenvalue problem

$$Ax = \lambda Bx, \tag{1}$$

where A and B are $n \times n$ real symmetric matrices with B being positive definite, arises in many applications[1]. Usually A and B are large and sparse and we need only the smallest few or the largest few, or the eigenvalues within an interval. In this project, we implement the trace minimization algorithms that can be used to solve these eigenvalue problems to obtain the smallest p eigenvalues and their corresponding eigenvectors, or all the eigenpairs within a given interval $[a,b]$.

This report is organized as follows. Section 1 presents a brief deivation of the basic trace minimization algortihm and its composite procedures, and the variations of the trace minimization algorithm, namely trace minimization algorithm with deflation and the Davidson-type trace minimization algorithm. Section 2 describes the implementation details of the algorithm: the eigen decomposition methods for dense matrices for solving the eigenvalue problem within a subspace; the iterative solvers used in solving the reduced system, particularly the variation of conjugate gradient method used in the algorithm and its stopping criterion; and also the multisectioning technique to subdivide an interval into subintervals for parallel computations. Section 3 shows our experimental results and discusses about the findings.

1. Algorithms

The trace minimization algorithm is motivated by the following theorem.

Theorem 1 (Sameh and Wisniewski[2]) *Let A and B be as given in problem 1; and let $X*$ be the set of all $n \times p$ matrices X for which $X^T BX = I_p, 1 \le p \le n$. Then*

$$\min_{X \in X*} tr(X^T AX) = \sum_{i=1}^{p} \lambda_i, \tag{2}$$

wher $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are the eigenvalues of problem 1. The equality holds if and only if the columns of the matrix X, which achieves the minimum, span the eigenspace corresponding to the smallest p eigenvalues.

Hence, if we could find such X that minimizes the trace, we would obtain the smallest p eigenpairs. In the trace minimization algorithm, we start with a random X in the subspace and iteratively refine it by computing a correction term Δ until the trace is minimized, i.e. in each iteration,

$$tr\left((Y-\Delta)^T A(Y-\Delta)\right) < tr\left(Y^T AY\right). \tag{3}$$

To achieve the convergence, we define the function

$$\begin{aligned}\phi(\Delta) &= tr\left((Y-\Delta)^T A(Y-\Delta)\right) - tr\left(Y^T AY\right) \\ &= 2tr\left(Y^T A\Delta\right) - tr(\Delta^T A\Delta),\end{aligned} \tag{4}$$

and seek to maximize ϕ subject to $Y^T B\Delta = 0$. This can be done by taking the derivates of the function

$$\psi(\Delta, \mathscr{L}) = 2tr\left(Y^T A\Delta\right) - tr(\Delta^T A\Delta) - tr(\mathscr{L} \cdot Y^T B\Delta), \tag{5}$$

Algorithm 1: Basic Trace Minimization

Input: Subspace dimension $s = 2p$,
$V_1 \in \mathbb{R}^{n\times s}$ with rank s,
$A = A^T$; B is symmetric positive definite

Output: The smallest p eigenvalues (Θ_k) and their corresponding eigenvectors (Y_k)

1 **for** $k = 1 \rightarrow$ mat_iter **do**
2 B-orthonormalize $V_k \rightarrow Z_k$ **:**
3 Compute $\hat{B}_k = BV_k$;
4 Compute all eigenpairs of $V_k^T BV_k$, $V_k^T \hat{B}_k = \Upsilon_k \Sigma_k \Upsilon_k^T$;
5 Compute $Z_k = V_k \Upsilon_k \Sigma_k^{-1/2}$;
6 Compute $\dot{B}_k = \hat{B}_k \Upsilon_k \Sigma_k^{-1/2}$;
7 Perform the Rayleigh-Ritz procedure to obtain the Ritz eigenpairs ($AY_k \approx BY_k\Theta_k$) **:**
8 Compute $\hat{A}_k = AZ_k$;
9 Compute all eigenpairs of $Z_k^T AZ_k$, $Z_k^T \hat{A}_k = \Pi_k \Theta_k \Pi_k^T$;
10 Sort the eigenpairs (Θ_k, Π_k) in assending order of Θ_k;
11 Compute the Ritz vectors $Y_k = Z_k\Pi_k$;
12 Compute $\ddot{B}_k = \dot{B}_k\Pi_k$;
13 Compute $\dot{A}_k = \hat{A}_k\Pi_k$;
14 Compute the first p columns of the residual vectors $\Phi_k = \dot{A} - \ddot{B}\Theta_k$;
15 Test for convergence **:**
16 **if** $\|\Phi_k e_j\|_2/\theta_j \leq \tau; \forall j \leq p$ **then break**;
17 Solve the reduced system $P_k A\Delta_k = P_k AY_k$ where $P_k = I - BY_k \left(Y_k^T B^2 Y_k\right)^{-1} Y_k^T B$ **:**
18 Compute the QR decomposition of BY_k, $\ddot{B}_k = [Q_{k1}\ Q_{k2}] \left[R_k^T\ 0\right]^T$;
19 Compute $N_k = P_k \dot{A}_k Y_k$;
20 Use iterative solver to solve $P_k A\Delta_k = N_k$ with initial guess $\Delta_k = 0$;
21 Compute $V_{k+1} = Y_k - \Delta_k$;

where $\mathscr{L}$ is a Langrange multiplier, with respect to Δ and $\mathscr{L}$ and set them to 0. This would yield the following two equations.

$$\frac{\partial \psi}{\partial \Delta} = 0 \quad \Rightarrow \quad 2AY - 2A\Delta - BY\mathscr{L}^T = 0, \text{ and} \tag{6}$$

$$\frac{\partial \psi}{\partial \mathscr{L}} = 0 \quad \Rightarrow \quad Y^T B\Delta = 0, \tag{7}$$

which can be expressed as a saddle-point equation,

$$\begin{bmatrix} A & BY \\ Y^T B & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ L \end{bmatrix} = \begin{bmatrix} AY \\ 0 \end{bmatrix}. \tag{8}$$

This saddle-point equation is further reduced to the following system in [2].

$$PA\Delta = PAY, \tag{9}$$

where $P = I - BY(Y^T B^2 Y)^{-1} Y^T B$, on which our project implementation is based.

1.1 Basic Trace Minimization Algorithm

The basic trace minimization algorithm is outlined in Algorithm 1. Note that since both A and B are large, we reduce the number of multiplications involving A and B to one each (Lines 3 and 8) except within the iterative solver. This is achieved by postmultiplying the products BV_k and AZ_k by other matrices (Lines 6, 12 and 13). In addition, we only compute the first p columns of the residual matrix Φ (Line 14) since we only need the first p columns to converge.

1.2 Trace Minimization Algorithm with Deflation

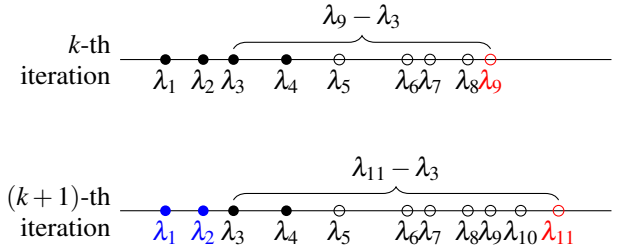

Figure 1. Eigenvalue separation before and after column replacement after converged

By observing that each column of the residual matrix *Phi* is tested for convergence independently, we can modify the trace minimization algorithm to seperate the already converged eigenpairs from the yet-to-converged ones. This can accelerate the algorithm in two ways. Firstly, we can preserve the converged eigenpairs and focus on the yet-to-converged ones, and thus no computation is wasted. Secondly, by moving the converged eigenpairs away, we get a better convergence rate. To illustrate this, suppose the distribution of the eigenvalues of a problem is given in Figure 1 and we are finding the smallest 4 eigenpairs. Let us assume that at the k-th iteration, the two smallest eigenpairs converged. We replace those two columns by some random columns. At the $(k+1)$-th iteration, the separation of the yet-to-converged eigenvalues form λ_{p+1} becomes larger. Since we know that the convergence rate of conjugate gradient method in the algorithm is proportional to the ratio of eigenvalues,

$$\frac{\lambda_i}{\lambda_{p+1}}, \tag{10}$$

the convergence rate at the $(k+1)$-th iteration is then faster than that of the k-th iteration.

Since we replace the converged columns by new random columns, we need to make sure that they do not converge to the same eigenvectors as the converged ones. To do this, we project V_k by the projection matrix $U = I - BC\left(C^T B^2 C\right)^{-1} C^T B$ such that V_k is B-orthogonal to C. The algorithm is outlined in Algorithm 2. Details are omitted if they are the same as the basic algorithm and the changes are highlighted in blue.

Algorithm 2: Trace Minimization Algorithm with Deflation

Input: Subspace dimension $s = 2p$,
$V_1 \in \mathbb{R}^{n\times s}$ with rank s,
$A = A^T$; B is symmetric positive definite
Output: The smallest p eigenvalues (Θ_k) and their corresponding eigenvectors (Y_k)

1 $C = \emptyset$;
2 **for** $k = 1 \rightarrow$ mat_iter **do**
3 **if** $C \neq \emptyset$ **then**
4 Project V_k by $U = I - BC\left(C^T B^2 C\right)^{-1} C^T B$
5 B-orthonormalize $V_k \rightarrow Z_k$;
6 Perform the Rayleigh-Ritz procedure to obtain the Ritz eigenpairs ($AY_k \approx BY_k\Theta_k$);
7 Compute the first p columns of the residual vectors $\Phi_k = \dot{A} - \ddot{B}\Theta_k$;
8 Test for convergence;
9 Move converged columns to C and replace them by some random vectors;
10 Solve the reduced system $P_k A\Delta_k = P_k AY_k$ where $P_k = I - BG_k\left(G_k^T B^2 G_k\right)^{-1} G_k^T B$ and $G_k = [Y_k\ C]$;
11 Compute $V_{k+1} = Y_k - \Delta_k$;

1.3 TraceMin-Davidson Algorithm

The trace minimization algorithm can also be acceleration by a Davidson-type technique, which is expanding the search subspace instead of modifying the column vectors. By doing so, there is a potential to find the eigenpairs in fewer iterations than the basic trace minimization algorithm. However, since the subspace is expanded in each iteration, it results in more computations in subsequent iterations. The Davidson-type trace minimization algorithm is outlined in Algorithm 3 with changes highlighted in blue. There are some computations that can be minimized[1]. The B-orthonormalization of V_k in Line 3 can be simplified as follows:

1. Project the vectors of BV_{k-1} out of Δ_{k-1}:
$\Delta_{k-1} \leftarrow \left[I - BV_{k-1}\left(V_{k-1}^T B^2 V_{k-1}\right)^{-1} V_{k-1}^T B\right]\Delta_{k-1}$
2. B-orthonormalize Δ_{k-1}.
3. Add Δ_{k-1} to the subspace:
$V_k = [V_{k-1}\ \Delta_{k-1}]$

Additionally, we only need to compute the s new vectors of AV_k in Line 5 and the corresponding columns of $Z_k^T AZ_k$ in Line 6.

2. Implementation Details

2.1 Eigen Decomposition for Dense Matrices

The B-orthonormalization and the Rayleigh-Ritz procedures both involve finding the eigen decomposition of a dense matrix. In our project, we implemented the 1-sided and 2-sided

Algorithm 3: TraceMin-Davidson Algorithm

Input: Block size $s \geq p$,
Maximum subspace dimension $d > 2s$,
$V_1 \in \mathbb{R}^{n\times s}$ with rank s,
$A = A^T$; B is symmetric positive definite
Output: The smallest p eigenvalues (Θ_k) and their corresponding eigenvectors (Y_k)

1 Initialize current subspace dimension $c = s$;
2 **for** $k = 1 \rightarrow$ mat_iter **do**
3 B-orthonormalize $V_k \rightarrow Z_k$;
4 Perform the Rayleigh-Ritz procedure to obtain the Ritz eigenpairs ($AY_k \approx BY_k\Theta_k$) **:**
5 Compute $\hat{A}_k = AZ_k$;
6 Compute all eigenpairs of $Z_k^T AZ_k$, $Z_k^T \hat{A}_k = \Pi_k\Theta_k\Pi_k^T$;
7 Sort the eigenpairs (Θ_k, Π_k) in assending order of Θ_k;
8 Compute the first s Ritz vectors $Y_k = Z_k\Pi_k$;
9 Compute $\ddot{B}_k = \dot{B}_k\Pi_k$;
10 Compute $\dot{A}_k = \hat{A}_k\Pi_k$;
11 Compute the first p columns of the residual vectors $\Phi_k = \dot{A} - \ddot{B}\Theta_k$;
12 Test for convergence;
13 **if** $c + s > d$ **then** Restart with $V_k = Y_k$ and $c = s$;
14 Solve the reduced system $P_k A\Delta_k = P_k AY_k$ where $P_k = I - BY_k\left(Y_k^T B^2 Y_k\right)^{-1} Y_k^T B$;
15 Add Δ_k to the subspace $V_{k+1} = [V_k\ \Delta_k]$;

Jacobi methods as the eigen decomposition routines. They can both be parallelized in a similar manner since the plane rotations are applied on two columns (or rows) and are independent from other columns (or rows). In the textbook[3], it provides a parallelism scheme to simultaneously compute $\lfloor n/2 \rfloor$ plane rotations. This scheme applies to both 1-sided and 2-sided Jacobi methods. In this project, we apply a simpilfied algorithm to compute the order of annihilations as described in Algorithm 4. Since this order of annihilation depends only on the matrix dimension, it does not change throughout the outer loop of the trace minimization algorithms except that for the Rayleigh-Ritz procedure in TraceMin-Davidson algorithm. Hence, the order can be computed before the outer loop and reused throughout the iterations. For Tracemin-Davidson algorithm, new order needs to be recomputed since the dimension increases over iterations.

The two Jacobi methods differ from each other in some ways. For the 1-sided Jacobi method, it finds an orthogonal matrix U such that $AU = Q$ is a matrix of orthogonal columns and Q can be written as $V\Sigma$ with $V^T V = I$; whereas for the 2-sided Jacobi method, it finds an orthogonal matrix U such that $UAU^T = D$ and D is diagonal. Plane rotations only apply on columns for 1-sided Jacobi method but apply on both columns and rows for 2-sided Jacobi method. The angle of

Algorithm 4: Order of Annihilations

Input: Matrix dimension n
Output: The order of annihilations $\mathscr{P}_k$ for each iteration k of Jacobi mehods

$m = \left\lfloor \frac{n}{2} \right\rfloor$;
for $k = 1 \rightarrow$ n **do**
  $\mathscr{P}_k = \emptyset$;
  for $j = \frac{n-k}{2} + 1 \rightarrow \frac{n-k}{2} + m$ **do**
    $i = n - k - j$;
    if $i + n \neq j$ **then**
      if $i < 0$ **then** $\mathscr{P}_k = \mathscr{P}_k \cup \{(j, i+n)\}$;
      else $\mathscr{P}_k = \mathscr{P}_k \cup \{(i, j)\}$;

plane rotation depends on the 2-norms of the columns and their inner product for 1-sided Jacobi method, and the values of the 2×2 principal submatrix for 2-sided Jacobi method. Finally, for 1-sided Jacobi method, the absolute values of the eigenvalues are the 2-norms of the columns of Q and the eigenvectors are found by scaling the columns by the inverse of the eigenvalues. For 2-sided Jacobi method, the resulting diagonal matrix D contains the eigenvalues and the columns of matrix U^T which is the product of all the plane rotations are the eigenvectors. Since the 1-sided Jacobi method naturally computes the absolute values of the eigenvalues, it is more suitable to be used on the B-orthonormalization procedure since B is symmetric positive definite and the eigenvalues of $V^T BV$ are always positive. To find negative eigenvalues by the 1-sided Jacobi method, one needs to calculate the Rayleigh quotient $v^T Av/\|v\|_2$. During the experiments, we found that the 2-sided Jacobi method is more numerically stable when calculating negative eigenvalues. Therefore, we apply the 2-sided Jacobi method for the Rayleigh-Ritz procedure althogh the 1-sided Jacobi method is computationally less expensive.

2.2 Modified Conjugate Gradient Method
2.3 Minimum Residual Method
2.4 Multisectioning

3. Results and Discussion

4. Work Distribution

The work distribution is summarized in Table 1.

Table 1. Work Distribution

Name	Work
Yu Hong Yeung	1. Basic Trace Minimization Algorithm Framework 2. 1-sided and 2-sided Jacobi Methods 3. Trace Minimization Algorithm with Deflation 4. TraceMin-Davidson Algorithm
Xin Cheng	1. Conjugate Gradient Method 2. Convergence Test for CG 3. Testing
Di Jin	1. Minimum Residual Method 2. Multisectioning 3. Testing

Acknowledgments

We would like to thank both Prof. Ahmed Sameh and our TA Zhengyi Zhang for helping us throughout this project.

References

[1] Alicia M. Klinvex. *Parallel Symmetric Eigenvalue Problem Solvers*. PhD thesis, Purdue University, May 2015.

[2] Ahmed H. Sameh and John A. Wisniewski. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal on Numerical Analysis*, 19(6):1243–1259, 1982.

[3] Efstratios Gallopoulos, Bernard Philippe, and Ahmed H. Sameh. *Parallelism in Matrix Computations*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[4] Ahmed H. Sameh and Zhanye Tong. The trace minimization method for the symmetric generalized eigenvalue problem. *Journal of Computational and Applied Mathematics*, 123(1–2):155–175, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.