Hindawi Publishing Corporation
Advances in Numerical Analysis
Volume 2011, Article ID 826376, 9 pages
doi:10.1155/2011/826376

Research Article

Partitioning Inverse Monte Carlo Iterative Algorithm for Finding the Three Smallest Eigenpairs of Generalized Eigenvalue Problem

Behrouz Fathi Vajargah and Farshid Mehrdoust

Department of Statistics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran

Correspondence should be addressed to Behrouz Fathi Vajargah, fmehrdoust@guilan.ac.ir

Received 5 December 2010; Accepted 15 February 2011

Academic Editor: Michele Benzi

A new Monte Carlo approach for evaluating the generalized eigenpair of real symmetric matrices will be proposed. Algorithm for the three smallest eigenpairs based on the partitioning inverse Monte Carlo iterative (IMCI) method will be considered.

1. Introduction

It is well known that the problem of calculating the largest or smallest generalized eigenvalue problem is one of the most important problems in science and engineering [1, 2]. This problem arises naturally in many applications. Mathematically, it is a generalization of the symmetric eigenvalue problem, and it can be reduced to an equivalent symmetric eigenvalue problem. Let $A, B \in \Re^{n\times n}$ be real symmetric matrices and the matrix B a positive definite matrix. Consider the problem of evaluating the eigenvalues of the pencil (A, B), that is, the values for which

$$Ax = \lambda Bx. \tag{1.1}$$

A generalized eigenvalue problem (1.1) is said to be symmetric positive definite (S/PD) if A is symmetric and B is positive definite.

```
Input:
  Initial vector $x_0$
begin
  Set $\lambda_1^{(1)} = \frac{x_0^T A x_0}{x_0^T B x_0}$
  For $j = 0, 1, 2, \ldots$
  begin
    Solve linear system $Az_{j+1} = Bx_j$ for $z_{j+1}$
    Set $\lambda_1^{(j+1)} = \frac{z_{j+1}^T A z_{j+1}}{z_{j+1}^T B z_{j+1}}$
    Set $x_{j+1} = \frac{z_{j+1}}{\|z_{j+1}\|}$
    Output: $x_j, \lambda_1^{(j)}$
  end
end
```

Algorithm 1

2. Inverse Vector Iteration Method

Another procedure for eigenvalue prediction is to use the Rayleigh quotient given by [3]

$$\lambda = \mu(x) = \frac{x^T A x}{x^T B x}. \tag{2.1}$$

since B is positive definite, then (2.1) is well defined.

Theorem 2.1. *Suppose that $\lambda_{\min} = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_{n-1} < \lambda_n = \lambda_{\max}$ are n eigenvalues for pencil (A, B) and $v_1, \ldots, v_n$, corresponding eigenvectors. Then for arbitrary initial vector x one has [3]*

$$\lambda_1 < \mu(x) < \lambda_n, \tag{2.2}$$

where $\mu(x)$ is as introduced in (2.1).

Theorem 2.2. *The inverse vector iteration method for arbitrary choice vector x_0 is convergent to the smallest eigenvalue and corresponding eigenvector for pencil (A, B). Also, the rate of convergence depends on $\mathcal{O}(\lambda_2/\lambda_1)^k$, where k is number of iterations [3].*

Algorithm 1 evaluates the smallest eigenpair based on the inverse vector iteration [4].

3. Monte Carlo Method for Matrix Computations

Suppose that the matrix $A \in \Re^{n\times n}$ and two vectors $f, h \in \Re^n$ are given. Consider the following Markov chain T_i with length i:

$$T_i : k_0 \longrightarrow k_1 \cdots \longrightarrow k_i, \tag{3.1}$$

where for $j = 1,\dots,i,\ k_j \in \{1,2,\dots,n\}$. The statistical nature of constructing the chain (3.1) follows as

$$p(k_0 = \alpha) = p_\alpha, \qquad p(k_j = \beta \mid k_{j-1} = \alpha) = p_{\alpha\beta}, \tag{3.2}$$

where p_α and $p_{\alpha\beta}$ show the probability of starting chain at α and transition probability from state α to β, respectively.

In fact

$$\sum_{\alpha=1}^{n} p_\alpha = 1, \quad \sum_{\beta=1}^{n} p_{\alpha\beta} = 1, \quad \alpha = 1,2,\dots,n. \tag{3.3}$$

Define the random variable W_j using the following recursion for

$$W_0 = 1, \qquad W_j = W_{j-1}\frac{a_{k_{j-1}k_j}}{p_{k_{j-1}k_j}}, \quad j = 1,2,\dots,i. \tag{3.4}$$

Now, define the following random variable:

$$\Theta[h] = \frac{h_{k_0}}{p_{k_0}}\sum_{j=0}^{\infty} W_j f_{k_j}. \tag{3.5}$$

Theorem 3.1. *Consider the following system:*

$$Ax = b. \tag{3.6}$$

Let the nonsingular matrix $M \in \Re^n$, such that $MA = I - L$, then the system (3.6) *can be presented in the following form:*

$$x = Lx + f, \tag{3.7}$$

where $f = Mb$. Then under condition $\mathrm{Max}_i \sum_{j=1}^{n} |l_{ij}| < 1$, *one has [5]*

$$E\{\Theta[h]\} = \langle h, x\rangle. \tag{3.8}$$

Suppose that $x^{(i)}$ is the ith iterative solution of the following recursion relation with $x^{(0)} = f$. If we set the random variable

$$\Theta_i[h] = \frac{h_{k_0}}{p_{k_0}} \sum_{j=0}^{i} W_j f_{k_j}, \tag{3.9}$$

then

$$E\{\Theta_i[h]\} = \left\langle h, x^{(i+1)} \right\rangle. \tag{3.10}$$

By simulating N random paths with length i

$$T_i^{(s)} : k_0^{(s)} \longrightarrow k_1^{(s)} \longrightarrow \cdots \longrightarrow k_i^{(s)}, \quad s = 1, 2, \ldots, N, \tag{3.11}$$

we can find

$$\Theta_i^{(s)}(h) = \frac{h_{k_0}}{p_{k_0}^{(s)}} \sum_{j=0}^{i} W_j^{(s)} f_{k_j}, \quad s = 1, \ldots, N. \tag{3.12}$$

The Monte Carlo estimation can be evaluated by

$$\Theta_i = \frac{1}{N} \sum_{s=1}^{N} \Theta_i^{(s)}(h) \tag{3.13}$$

which is an approximation of $\langle h, x^{(i+1)} \rangle$.

From all possible permissible densities, we apply the following:

$$\begin{aligned} p_\alpha &= \frac{|h_\alpha|}{\sum_{\alpha=1}^{n} |h_\alpha|}, \\ p_{\alpha\beta} &= \frac{|a_{\alpha\beta}|}{\sum_{\beta=1}^{n} |a_{\alpha\beta}|}, \quad \alpha = 1, 2, \ldots, n. \end{aligned} \tag{3.14}$$

The choice of the initial density vector and the transition probability matrix leads to an almost Optimal Monte Carlo (MAO) algorithm.

Theorem 3.2. *Using the above choice $p = \{p_\alpha\}_{\alpha=1}^{n}$ and $P = \{p_{\alpha\beta}\}_{\alpha,\beta=1}^{n}$ the variance of the unbiased estimator for obtaining the inverse matrix is minimized [4].*

There is a *global algorithm* that evaluates the solution of system (3.6) for every matrix A. The complexity of algorithm is $O(n^2 l N)$, where l and N are the average length of Markov chian and the number of simulated paths, respectively [2].

```
input:
    $A \in R^{n\times n}, f_0 \in R^n$
begin
    Starting from initial vector $f_0$
    For $j = 1, 2, \ldots$
    begin
        Using global algorithm, calculate the sequence of Monte Carlo
        iterations by solving the following system
                        $Af_j = Bf_{j-1}$
        Set
                $\lambda^{(j)} = \frac{\langle Af_j, h_j\rangle}{\langle Bf_j, h_j\rangle} = \frac{\langle Bf_{j-1}, h_j\rangle}{\langle Bf_j, h_j\rangle}$
        Output:
        Smallest eigenvector $\lambda_1^{(j)}$, and corresponding eigenvector $f_j$.
    end
end
```

Algorithm 2

4. Inverse Monte Carlo Iterative Algorithm (IMCI)

Inverse Monte Carlo iterative algorithm can be applied when A is a nonsingular matrix. In this method, we calculate the following functional in each steps:

$$\frac{\langle Af_j, h_j\rangle}{\langle Bf_j, h_j\rangle} = \frac{\langle Bf_{j-1}, h_j\rangle}{\langle Bf_j, h_j\rangle}. \tag{4.1}$$

It is more efficient that we first evaluate the inverse matrix using the Monte Carlo algorithm [1, 2, 4]. The algorithm can be realized as in Algorithm 2.

5. Partitioning IMCI

Let the matrix A be partitioned into four blocks A_1, A_2, A_3, and A_4, where A_1 and A_4 are square matrices of order p and q such that $p + q = n$:

$$A = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right). \tag{5.1}$$

By assumption that all the indicated matrix inversions are realized, it is easy to verify that

$$A^{-1} = \left(\begin{array}{c|c} B & L \\ \hline M & N \end{array}\right), \tag{5.2}$$

```
Partitioning inverse $(S, n)$
begin:
      $n = \text{rank}(S);\ p = n/2$
      $A = S[1:p, 1:p];\ B = S[1:p, p+1:n]$
      $C = S[p+1:n, 1:p];\ D = S[p+1:n, p+1:n]$
      $m = \text{size}\ (A)$
   if $m \le$ threshold
      $AA$ = Monte Carlo procedure $(A)$
   else begin:
         $AA$ = Partitioning inverse $(A, m)$
         $N$ = Partitioning inverse $(D - C * AA * B)$
         $M = -N * C * AA;\ L = -AA * B * N$
         $K = AA - AA * B * M$
         $SS[1:p, 1:p] = K;\ SS[1:p, p+1:n] = L$
         $SS[p+1:n, 1:p] = M;\ SS[p+1:n, p+1:n] = N$
   end
end
```

Algorithm 3

where

$$N = \left(A_4 - A_3 A_1^{-1} A_2\right)^{-1}, \qquad M = -N A_3 A_1^{-1},$$
$$L = -A_1^{-1} A_2 N, \qquad K = A_1^{-1} - A_1^{-1} A_2 M. \tag{5.3}$$

Thus inverting a matrix of order n comes down to inverting four matrices, of which two have order p and two have order q, and several matrix multiplications. Therefore the basic Monte Carlo for solving $Af_j = Bf_{j-1}$ will be called as the dimension of matrix $A$and equals to *threshold*. This action causes the convergence acceleration. Now, we can use the following recursion algorithm to obtain the inverse of matrix A (see Algorithm 3).

6. Finding More Than One Generalized Eigenvalues

Assume that an eigenvalue λ_1 and its corresponding eigenvector v_1 have been computed using the partitioning IMCI algorithm. In the first step of the above algorithm, we deflate the matrix A to the matrix B. Then, we repeat again the first step of the algorithm to obtain the dominant eigenvalue of B which is the second dominant eigenvalue of A. Let p values of eigenvalues of pencil (A, B) be computed. Suppose that V_p is a matrix such that the columns of V_p are p vector of eigenvector of pencil (A, B), that is,

$$V_p = \left[\nu_1, \dots, \nu_p\right], \tag{6.1}$$

where ν_i is eigenvector corresponding eigevalue $\lambda_i,\ i = 1, 2, \dots, p$.

Table 1: Number of chains $N = 80$.

Dimension	Eigenvalues error			Eigenvectors error		
	$\lambda_1^{(1)}$	$\lambda_1^{(2)}$	$\lambda_1^{(3)}$	$v_1^{(1)}$	$v_1^{(2)}$	$v_1^{(3)}$
64×64	1.26×10^{-7}	4.5×10^{-7}	1.26×10^{-6}	8.36×10^{-4}	.002813	.008154
128×128	1.18×10^{-7}	4.46×10^{-7}	1.33×10^{-6}	6.33×10^{-4}	.002534	.008074
256×256	6.48×10^{-8}	4.37×10^{-7}	1.51×10^{-6}	3.16×10^{-4}	.002151	.008070
512×512	7.91×10^{-8}	2.87×10^{-7}	1.72×10^{-6}	1.83×10^{-4}	.001284	.007759
1024×1024	7.04×10^{-8}	1.99×10^{-7}	1.79×10^{-6}	1.06×10^{-4}	.000712	.00769

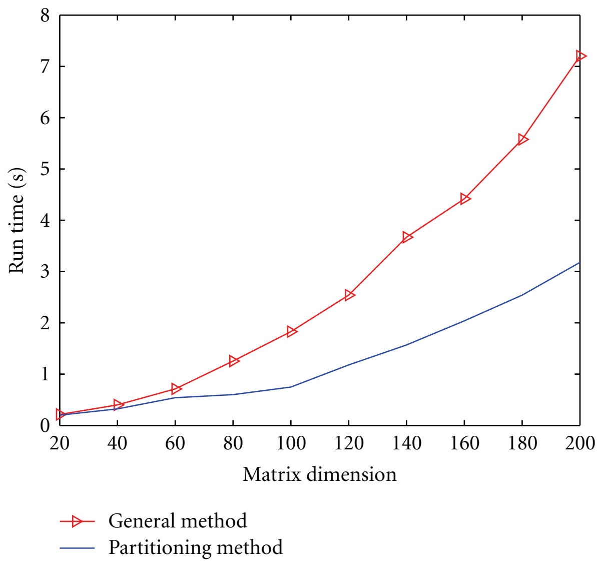

Figure 1: Computational times for general and partition methods.

Now, let

$$(A_p, B) = \left(A + (BV_p)\Lambda(BV_p)^T, B\right), \tag{6.2}$$

where

$$\Lambda = \operatorname{diag}\{\delta_i - \lambda_i\}, \quad \delta_i > \lambda_p,\ i = 1, \dots, p. \tag{6.3}$$

Hence, if we find the pth smallest eigenpair of pencil (A, B), then we can evalute λ_{p+1}, that is, $(p+1)$th smallest eigenvalue of pencil (A, B).

7. Numerical Results

In this section, the experimental results for obtaining the three smallest eigenpairs outlined in Tables 1, 2, and 3. The numerical tests are performed on Intel(R) (Core(TM)2 CPU, 1.83 GHz) personal machine.

Table 2: The solution when the number of chains increases.

Number of chains	Calculated eignvalues		
N	Exact $\lambda_1^{(1)} = .74529395$	Exact $\lambda_1^{(2)} = .85537131$	Exact $\lambda_1^{(3)} = .91032740$
20	.74737713	.85513704	.91008238
40	.74529979	.85537157	.91034304
80	.74529407	.85537175	.9102873

Table 3: Total computational time for general and partitioning methods.

Dimension	Time (Sec.)	
	General method	Partitioning method
20 × 20	.21	.20
40 × 40	.40	.32
60 × 60	.71	.54
80 × 80	1.25	.60
100 × 100	1.83	.75
120 × 120	2.54	1.18
140 × 140	3.67	1.57
160 × 160	4.42	2.04
180 × 180	5.58	2.54
200 × 200	7.20	3.18

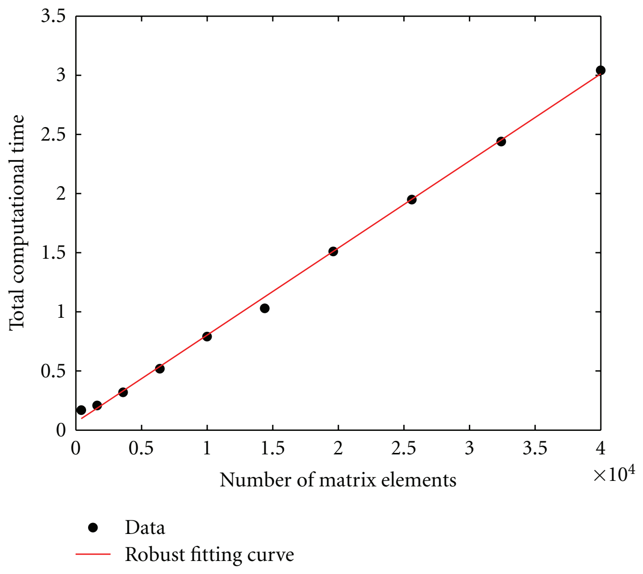

Figure 2: Regression function $y = 0.0000658008x + 0.1048607485$.

8. Conclusion and Future Study

We have seen that Monte Carlo algorithms can be used for finding more than one eigenpair of generalized eigenvalue problems. We analyze the computational complexity, speedup, and efficiency of the algorithm in the case of dealing with sparse matrices. Finally, a new method for computing eigenpairs as the partitioned method is presented. In Figure 1 the comparison of computational times between general Monte Carlo algorithm and partitioning algorithm is shown. The scatter diagram in Figure 2, shows that there is a linear relationship between matrix dimension (equivalently, the number of matrix elements) and total computational time for partitioning IMCI.

References

[1] I. Dimov and A. Karaivanova, "Iterative Monte Carlo algorithm for linear algebra problem," *Lecture Note in Computer Science*, pp. 66–77, 1996.
[2] I. Dimov, *Monte Carlo Methods for Applied Scientists*, World Scientific Publishing, 2008.
[3] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, 1991.
[4] B. Fathi, "A way to obtain Monte Carlo matrix inversion with minimal error," *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 225–233, 2007.
[5] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New York, NY, USA, 1981.