Far East Journal of Applied Mathematics
Volume 52, Number 2, 2011, Pages 81-91
Published Online: May 22, 2011
This paper is available online at http://pphmj.com/journals/fjam.htm

PERFORMANCE OF OPTIMIZED GENERATED MARKOV CHAIN BASED ON LEAST SQUARES METHOD FOR SOLVING LINEAR SYSTEMS

MAHMOOD KHOSHHAL

Department of Mathematics
Islamic Azad University
Khalkhal Branch, Iran
e-mail: khoshhal_aukh@yahoo.com

Abstract

Solving linear systems $Hx = f$ with minimizing the error vector $v(x) = Hx - f$ by an optimized stochastic method is considered. This solution can be significantly obtained even when the coefficient matrix H is singular.

0. Introduction

One of the most suitable methods to obtain the solution of linear systems is the least squares method, especially when the coefficient matrix is not invertible. This method considers an algebraic equation system $Hx = f$, and gives the $\min v(x) = v(x^*)$, where $x^* = H^L f$ and $H^L = (H^T H)^{-1} H^T$. The desired results can be estimated using Monte Carlo methods. The Monte Carlo method gives statistical estimates of the elements of inverse of the matrix or for components of the solution vector of a linear system by performing random sampling of a certain random variable whose mathematical expectation is the desired solution.

2010 Mathematics Subject Classification: 34A30.

Keywords and phrases: error vector, least squares, minimization, singular system.

Received January 27, 2011

In this paper, we first obtain the elements of $[C]_{ij} = [(H^T H)^{-1}]_{ij}$ by Monte Carlo methods. Then for finding the minimization solution x^*, we employ $CH^T f$.

We consider solving the given linear systems by using the principle of least squares. Assume that

$$Hx = f, \tag{1}$$

where $H \in R^{m\times n}$, $x \in R^{n\times 1}$ and $f \in R^{m\times 1}$. Define

$$v(x) = Hx - f, \tag{2}$$

where $x \in R^{n\times 1}$. The vector $r(x)$ is called the *residual vector*. Thus, $v(x)$ is the error when we have *x* as an approximate solution of the system (1). The least squares solution *x* will have the property of minimizing of $\| v(x) \|^2$. Since

$$\begin{aligned} \| v(x) \|_2^2 &= (Hx - f, Hx - f) \\ &= (Hx - f)^T (Hx - f) \\ &= x^T H^T Hx - 2x^T H^T f + f^T f \end{aligned}$$

is a scalar, we have

$$\lambda(x) = x^T H^T Hx - 2x^T H^T f + f^T f.$$

Let $h \in R^{n\times 1}$ and $h \neq 0$. Then

$$\begin{aligned} \lambda(x + h) - \lambda(x) &= h^T H^T Hh + h^T H^T Hx + x^T H^T Hh - 2h^T H^T f \\ &= h^t H^T Hh + 2h^T H^T Hx - 2h^T H^T f. \end{aligned}$$

Since $x^T H^T Hh$ is a scalar quantity and

$$x^T H^T Hh = (x^T H^T Hh)^T = h^T H^T Hx,$$

we have

$$\lambda(x + h) - \lambda(x) = \| Hh \|_2^2 + 2h^T (H^T Hx - H^T f). \tag{3}$$

From the definition of a relative minimum (maximum), it follows that if $\lambda(x) = \| v(x) \|_2^2$ has a minimum (maximum), then $\lambda(x + h) - \lambda(x)$ must be of the same sign for all small values of the vector *h*. However, in (3), the sign of $\lambda(x + h) - \lambda(x)$ depends on the term $h^T (H^T Hx - H^T f) \neq 0$. Hence, for $\lambda(x)$ to have an extremum, it is necessary that $H^T Hx - H^T f = 0$. Moreover, if this condition is satisfied for $x = x^*$ (say), then $\lambda(x)$ has minimum at $x = x^*$, because

$$\lambda(x^* + h) - \lambda(x^*) = \| Hh \|_2^2 \geq 0$$

for small value the system values of *h*. This establishes that the least squares solution of (1) is given by the solution of normal equations $H^T Hx = H^T f$.

1. Stochastic Computation and Parameters

It is well known that classical methods can solve (1) by least square procedures. But, we are looking for extending this computation by Monte Carlo methods, since Monte Carlo methods have more quality for large systems [3, 4]. In this part, we explain the Monte Carlo can be employed in least square computations, also. This way, we will open a new application of Monte Carlo methods.

The Monte Carlo method, with generation of random numbers, randomly selects the elements of matrix $T = H^T H$, for computation of inverse and $C = T^{-1}$. Then we obtain x^* as we said about in the last Section. It is trivial that the results of minimization error will be different, if we employ different Monte Carlo method, such as with absorption or with absorption and different probabilistic structure, also. In this situation, we can compare the results, and select the best one. It is well known that any different Monte Carlo method can have only one best solution which makes $\| Hx - f \|$ the least.

As in [3], we consider the linear system

$$Hx = f, \tag{4}$$

where *H* is a given nonsingular matrix, $f = (f_1, f_2, ..., f_n)^T$ is a known vector and $x = (x_1, x_2, ..., x_n)^t$ is the solution vector that we are looking for finding it. If we

consider matrix $M^{n\times n}$ such that $MH = I - T$, $Mf = c$, then the linear system (4) is converted to

$$x^{(k+1)} = Tx^{(k)} + c, \tag{5}$$

where $T^{n\times n}$ is a given nonsingular matrix. It is well known that provided,

$$\rho(T) \le \| T \| < 1, \tag{6}$$

where $\rho(T)$ is the spectral radius of matrix T and $\| T \|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} | T_{ij} |$, the $x^{(k)}$ tends to the exact solution $x = (I - T)^{-1}c$ [1]. The inner product of x and $t = (t_1, t_2, ..., t_n) \in R^n$ is defined by $\langle t, x\rangle = t_1x_1 + t_2x_2 + \cdots + t_nx_n$. This inner product when $t = (\underbrace{0, 0, ..., 0, 1}_{i}, 0, ..., 0)$ is produced x_j, i.e., $\langle t, x\rangle = x_j$ which is the ith element of the solution vector. With Markov chain $i_0 \to i_1 \to \cdots \to i_k$ of the sample state $S = \{1, 2, ..., n\}$ which it will select the rows of columns indices of the matrix A, with initial distribution $p_{i_0} = P$ (Makov chain starts at $i_0 \in S$), and for $m \in \{1, 2, ..., n-1\}$, one step transition probability function is $p_{ij} = P(i_{m+1} = j \,|\, i_m = i)$ [2]. For solving the linear system (4) by Monte Carlo method, we consider transition probability matrix $P = [p_{ij}]_{i,j=1}^{n}$. Under the following conditions:

$$\begin{cases} \sum_{i=1}^{n} p_i = 1, & \\ \sum_{j=1}^{n} p_{ij} = 1, & i = 1, 2, ..., n, \\ p_i \ge 0,\ p_{ij} \ge 0, & i, j = 1, 2, ..., n, \\ t_i \ne 0 \to p_i > 0, & i = 1, 2, ..., n, \\ T_{ij} \ne 0 \to p_{ij} > 0, & i, j = 1, 2, ..., n. \end{cases}$$

We define $W_m = \dfrac{T_{i_0 i_1} T_{i_1 i_2} \cdots T_{i_{m-1} i_m}}{p_{i_0 i_1} p_{i_1 i_1} \cdots p_{i_{m-1} i_m}}$ and $W_m = W_{m-1} \dfrac{T_{i_{m-1} i_m}}{p_{i_{m-1} i_m}}$, $W_0 = 1$, $\eta_k(t) = \dfrac{h_{i_0}}{p_{i_0}} \sum_{m=0}^{k} W_m f_{i_m}$. It has been proved that $E[\eta_k(t)] = \left\langle h, \sum_{m=0}^{k} T^m c \right\rangle = \langle h, x^{(k+1)} \rangle$ [3]. Since $\eta_k(t)$ is an unbiased estimator of $\langle h, x^{(k+1)} \rangle$, we can introduce the Monte Carlo estimator based on

$$i_0^{(s)} \to i_1^{(s)} \to i_2^{(s)} \to \cdots \to i_k^{(s)}, \quad s = 1, 2, ..., N.$$

We have $\Theta_k(h) = \dfrac{1}{N} \sum_{s=1}^{N} \eta_k^{(s)}(h) \approx \langle h, x^{(k+1)} \rangle$, where for $s = 1, 2, ..., N$, $\eta_k^{(s)}(h) = \dfrac{t_{i_0^{(s)}}}{p_{i_0^{(s)}}} \sum_{m=0}^{k} W_m^{(s)} c_{i_m}$, with $W_m^{(s)} = \dfrac{T_{i_0^{(s)} i_1^{(s)}} T_{i_1^{(s)} i_2^{(s)}} \cdots T_{i_{m-1}^{(s)} i_m^{(s)}}}{p_{i_0^{(s)} i_1^{(s)}} p_{i_1^{(s)} i_2^{(s)}} \cdots p_{i_{m-1}^{(s)} i_m^{(s)}}}$, $W_0^{(s)} = 1$ and $\Theta_k(h) = \dfrac{1}{N} \sum_{s=1}^{N} \eta_k^{(s)}(h)$ [3]. $E[\eta_\infty(h)] = \lim_{k \to \infty} E[\eta_k(h)] = \lim_{k \to \infty} \langle h, x^{(k+1)} \rangle = \langle h, x \rangle$. $\Theta_k(e_j) = \dfrac{1}{N} \sum_{s=1}^{N} \eta_k^{(s)}(e_j) \approx x_j^{(k+1)} \approx x_j$.

For obtaining the element x_j of solution vector *x*, we simulate *N* random paths

$$j \to i_1^{(s)} \to i_2^{(s)} \to \cdots \to i_k^{(s)} \quad s = 1, 2, ..., N.$$

Then we have $\eta_k^{(s)}(e_j) = \sum_{m=0}^{k} W_m^{(s)} f_{i_m}^{(s)}$ with $W_m^{(s)} = \dfrac{T_{j i_1^{(s)}} T_{i_1^{(s)} i_2^{(s)}} \cdots T_{i_{m-1}^{(s)} i_m^{(s)}}}{p_{j i_1^{(s)}} p_{i_1^{(s)} i_2^{(s)}} \cdots p_{i_{m-1}^{(s)} i_m^{(s)}}}$, $W_0^{(s)} = 1$. The above Monte Carlo method is called *standard* (or *basic*) *Monte Carlo method*.

2. Optimized Monte Carlo Method

It has been discussed in [3] that how we can reduce the *nN* random paths to *N* paths.

We just have a quick review this concept.

Definition 1. For a Markov chain with sample state $S = \{1, 2, ..., n\}$, the random path $i_0^{(s)} \to i_1^{(s)} \to \cdots \to i_R^{(s)}$ is called *covering* if each state $j \in S$, can be visited at least once.

Definition 2. Let $j \in S$ be an arbitrary state of a Markov chain. Then the first time that Markov chain visit *j*, i.e., $R_j = \min_t \{t : i_t = j\}$, is called the *hitting time* to state *j*. In this way, for $j = 1, 2, ..., n$, we can reach to the all unbiased estimators $\eta_k(e_j)$. We first consider the covering path: $i_0 \to i_1 \to \cdots \to i_{R_j} \to \cdots \to i_R \to \cdots \to i_{R+k}$, where $R = \underset{1\le j\le n}{Max}\{R_j\} = \underset{1\le j\le n}{Max}\, Min\{t : i_t = j\}$ and k is an integer number. For every $j = 1, 2, ..., n$, we consider a sub path with length k as $i_{R_j} \to i_{R_j+1} \to \cdots \to i_{R_j+k}$, now we set $\tilde{\eta}_k(e_j) = \sum_{m=R_j}^{R_j+k} W_m b_{i_m}$, where

$$W_m = \frac{T_{i_{R_j} i_{R_j+1}} T_{i_{R_j+1} i_{R_j+2}} \cdots T_{i_{m-1} i_m}}{p_{i_{R_j} i_{R_j+1}} p_{i_{R_j+1} i_{R_j+2}} \cdots p_{i_{m-1} i_m}}, \; W_{R_j} = 1.$$

Theorem 1. *Under the above conditions,* $E[\tilde{\eta}_k(e_j)] = x_j^{(k+1)}$ [3].

Theorem 2. $Var[\Theta_k(e_j)] = Var[\tilde{\Theta}_k(e_j)]$, *where* $\Theta_k(e_j) = \frac{1}{N}\sum_{s=1}^{N} \eta_k^{(s)}(e_j)$ *and* $\tilde{\Theta}_k(e_j) = \frac{1}{N}\sum_{s=1}^{N} \tilde{\eta}_k^{(s)}(e_j)$ *for* $j = 1, 2, ..., n$ [3].

3. Increasing the Efficiency of the Method

In this section, we compare the efficiency of two methods where they discussed here based on [3, 4]. For simplicity of this comparison and bringing them to our analysis here, the method of the Monte Carlo without covering property (usual Monte Carlo method) is considered as method (1) and the Monte Carlo method based on covering property is considered as method (2). We remember that method (1) has $\Theta_k(e_j) = \frac{1}{N}\sum_{s=1}^{N} \eta_k^{(s)}(e_j)$, as the Monte Carlo estimator and method (2) has

$\widetilde{\Theta}_k(e_j) = \frac{1}{N}\sum_{s=1}^{N} E[(\widetilde{\eta}_k^{(s)}(e_j))^2(e_j)$ as the Monte Carlo estimator for x_j of the solution vector *x*.

Definition 3. Let $\Theta_k(e_j)$ and $\widetilde{\Theta}_k(e_j)$ be two Monte Carlo estimators for the parameter θ. Then the efficiency of $\widetilde{\Theta}_k(e_j)$ respect to $\Theta_k(e_j)$ is defined by $\varepsilon = \frac{t\, MSE[\Theta_k(e_j)]}{\widetilde{t}\; MSE[\widetilde{\Theta}_k(e_j)]}$, where t, $\widetilde{t}$ are the necessary lengths of Markov chains to reach the estimation for x_j. We recall that for any estimator *Y* of θ, we have $MSE[Y] = E[Y-\theta]^2 = Var[Y] + [E[Y]-\theta]^2$.

Definition 4. For two unbiased estimators $\Theta_k(e_j)$ and $\widetilde{\Theta}_k(e_j)$ of x_j, the efficiency of $\widetilde{\Theta}_k(e_j)$ with respect to $\Theta_k(e_j)$ is defined by $\varepsilon = \frac{t\ \mathrm{var}(\Theta_k(e_j))}{\widetilde{t}\ \mathrm{var}(\widetilde{\Theta}_k(e_j))}$, where *t* and $\widetilde{t}$ are the necessary lengths of Markov chains to reach the Monte Carlo estimation for x_j using methods (1) and (2), respectively.

Without loss of generality, we consider $N = 1$. Then by Theorem 1, $Var(\Theta_k(e_j)) = Var(\widetilde{\Theta}_k(e_j))$, and therefore $\varepsilon = \frac{t}{\widetilde{t}}$. In this case, in method (1), we use *n* paths with length *k*. Then the total lengths used in these paths are equal to *nk*. But, in method (2), we use only one path with length $(\max_{1\le j\le n}\{R_j\}) + k$ with average $E[\max_{1\le j\le n}\{R_j\}] + k$. For $n > 1$, we prove that the inequality $nk > E[\max_{1\le j\le n}\{R_j\}] + k$ is valid. To prove this inequality, we obtain $nk - k > E[\max_{1\le j\le n}\{R_j\}]$ or $k > E[\max_{1\le j\le n}\{R_j\}]/n - 1$. It proves that whenever *k* is larger than $E[\max_{1\le j\le n}\{R_j\}]/n - 1$, method (2) is more efficient than method (1).

Selecting a suitable starting point of the Markov chain using method (2), we can reduce the cost of computation. In needs, just we select the starting point of chain $i_0 = l$ such that $R = E[\underset{1\le j\le n}{Max}\{R_j\} | i_0 = l] = \underset{1\le i_0\le n}{Min}\ E[Max\{R_j\} | i_0 = l]$. For this, we set $p_{i_0} = \begin{cases} 1, & i_0 = l, \\ 0, & i_0 \ne l. \end{cases}$ Thus we prove the following theorem:

Theorem 3. *In a covering Markov chain* (*as discussed above*), *if* $k > E[\max_{1\le j\le n}\{R_j\}]/n - 1$, *then the covering method is more efficient than the standard Monte Carlo method*.

4. Parameter Estimation and Method

The estimator Θ^* for the solution vector is computed. The sum for Θ^* must be dropped when $|W_i| < \delta$, where δ is any given small number. Note that

$$|W_i| = \left| \frac{T_{\alpha_0\alpha_1}, ..., T_{\alpha_{i-1}\alpha_i}}{P_{\alpha_0\alpha_1}P_{\alpha_1\alpha_2}, ..., P_{\alpha_{i-1}\alpha_i}} \right| < \delta$$

and the length of Markov chain can be $L = i \le \dfrac{\log(\delta/\| f \|)}{\log\| T \|}$. Moreover, according to the central limit theorem for the given error ε, $N \ge \dfrac{0.6745^2}{\varepsilon^2}\dfrac{\| f\|^2}{(1-\| T \|)^2}$ [3]. Uniform transition probability with stopping rule $|W_i f_i| < \delta$ has been implemented.

Algorithm: Finding Monte Carlo H^{-1}:

1. **Input initial data:** Input matrix ***H***, the parameters γ and ε.
2. **Pre-processing:**

   2.1 Split $\boldsymbol{H} = \boldsymbol{D} - (\boldsymbol{D} - \boldsymbol{H})$, where ***D*** is a diagonally dominant matrix.

   2.2 Set $D = B - B_1$ where B is a diagonal matrix $b_{ii} = d_{ii}$, $i = 1, 2, ..., n$.

   2.3 Compute the matrix $T = B^{-1}B_1$.

   2.4 Compute $\| T \|$, the Number of Markov chain $N = \left(\dfrac{0.6745}{\gamma}\cdot\dfrac{1}{1-\| T \|}\right)^2$.
3. **For** $j = 1$ to n;

   3.1 For $i = 1$ to N;

   Markov Chain Monte Carlo computation:

   3.1.1 Set $\boldsymbol{t_k} = 0$ (stopping rule); $\boldsymbol{W}_0 = 1$,

   $$\boldsymbol{SUM}[i] = 0 \text{ and } \boldsymbol{point} = \boldsymbol{i};$$

3.1.2 Generate a uniformly distributed random number nextpoint.

3.1.3 If $T[\text{point}][\text{nextpoint}] != 0$.

LOOP

3.1.3.1 Compute

$$W_n = W_{n-1} \frac{T[\text{point}][\text{nextpoint}]}{p[\text{point}][\text{Nextpoint}]}.$$

3.1.3.2 Set Point=next point and $SUM[i] = SUM[i] + W_j$.

3.1.3.3 If $| W_j$

3.1.3.4 $| \geq \gamma$, $t_k = t_k + 1$.

3.1.3.5 If $t_k \geq n$, end of LOOP.

3.1.4 End If.

3.1.5 Else go to step 3.1.2.

3.2. End **of Loop *j***.

3.3. Compute the average of results.

4. End of Loop *i*.

5. Obtain the matrix $V = (I - T)^{-1}$.

6. Therefore $D^{-1} = VB^{-1}$.

7. Compute the MC inversion $D^{-1} = B(I - T)^{-1}$.

8. Set $D_0 = D^{-1}$ (approximate inversion) and $R_0 = I - DD_0$.

9. Use filter procedure $D_i = D_{i-1}(I + R_{i-1})$, $i = 1, 2, ..., m$, where $m \leq k$.

10. Consider accurate inversion of D by step 9 given by $D_0 = D_k$.

11. Compute $S = D - H$ where S can be any matrix with all non zero elements in diagonal and all of its off-diagonal elements are zero.

12. Main function for obtaining the inversion of $\boldsymbol{H}$ based on $\boldsymbol{D}^{-1}$ step 9:

12.1 Compute the matrices S_i, $i = 1, 2, ...,$

$\boldsymbol{k}$, where each $\boldsymbol{S_i}$ has just one element

of matrix $\boldsymbol{S}$.

12.2 Set $\boldsymbol{H_0} = \boldsymbol{D_0}$ and $\boldsymbol{H_k} = \boldsymbol{H} + \boldsymbol{S}$.

12.3 For $i = k - 1, k - 2, ..., 1, 0,$ apply

$$H_i^{-1} = H_{i+1}^{-1} + \frac{H_{i+1}^{-1} S_{i+1} H_{i+1}^{-1}}{1 - trace(H_{i+1}^{-1} S_{i+1})}.$$

13. Print the inversion of matrix $\boldsymbol{H}$.

14. End of algorithm.

5. Experimental Results

Here, we run and compare the algorithm based on usual Monte Carlo method and optimized Monte Carlo method.

Table 1. Least squares solution based on Monte Carlo methods

n	Usual Monte Carlo		Optimized Monte Carlo	
\	Time	Error	Time	Error
100	5.12	0.0005	1.29	0.0005
300	68.23	0.0005	13.91	0.0005
500	431.54	0.0005	91.57	0.0005
1000	24522.64	0.0005	691.22	0.0005
2000	24544.96	0.0005	3237.0592	0.0005
3000	81604.54	0.0005	10481.96	0.0005
4000	126429.41	0.0005	13653.54	0.0005

6. Conclusion

Usual and optimized Monte Carlo methods can be employed for least square problems. The performances of presented algorithms show that the optimized Monte Carlo method is up to 10 times faster than usual Monte Carlo method, with a predefined error 0.0005.

References

[1] R. L. Burden, J. D. Faires and A. C. Reynold, Numerical Analysis, Prindle, Weber and Schmidt, Boston, Massachusetts, USA, 1981.

[2] E. Cinlar, Introduction to Stochastic Processes, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[3] B. Fathi Vajargah and M. Saraji, Significant reduction of the number of Markov chains in Monte Carlo computation, Far East J. Appl. Math. 26(1) (2007), 73-83.

[4] I. M. Sobol, Monte Carlo Numerical Methods, Nauka, Moscow, 1973 (Russian).