Journal of Advances in Mathematics and Computer Science

28(5): 1-9, 2018; Article no.JAMCS.43705

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Linear Least Square Method for the Computation of the Mean First Passage Times of Ergodic Markov Chains

Yaming Chen[1*]

[1] *Nanjing University Press, Nanjing, P.R. China.*

Author's contribution

The sole author designed, analyzed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/JAMCS/2018/43705

Editor(s):

(1) Dr. Jacek Dziok, Professor, Institute of Mathematics, University of Rzeszow, Poland.

Reviewers:

(1) Nasr Al Din IDE, University of Aleppo, Syria.

(2) Najmuddin Ahmad, Integral University, India.

(3) Chacha Stephen Chacha, Mkwawa University College of Education, Tanzania.

Complete Peer review History: http://www.sciencedomain.org/review-history/26158

Received: 30 June 2018

Accepted: 03 September 2018

Published: 08 September 2018

Original Research Article

Abstract

An efficient and accurate iterative scheme for the computation of the mean first passage times (MFPTs) of ergodic Markov chains has been presented. Firstly, the computation problem of MFPTs is transformed into a set of linear equations. It has been proven that each of these equations is compatible and their minimal norm solutions constitute MFPTs. A new presentation of the MFPTs is also derived. Using linear least square algorithms, some numerical examples compared with the finite algorithm of Hunter [6] and iterative algorithm of J. Xu [7] are given. These results show that the new algorithm is suitable for large sparse systems.

Keywords: Markov chain; Stochastic matrix; Mean first passage times; Moore-Penrose inverse; Linear least square.

2010 Mathematics Subject Classification: 15B51; 60J22 ; 93E24; 65F20.

**Corresponding author: E-mail: YMChern@hotmail.com, cym_shaka@126.com, cym_shaka@hotmail.com*

1 Introduction

There are two kinds of algorithms for computing the mean first passage times(MFPTs) of finite ergodic Markov chains: finite algorithm and iterative algorithm. Meyer [1] and Hunter [2] propose finite algorithms for the computation of MFPTs based on generalized inverse matrix and stationary distribution vector. In 1985, Grassman, Taksar and Heyman [3] proposed the classical GTH algorithm and EGTH algorithm to find stationary distribution vectors. In 2016, Hunter [4, 5] gave rank-one update algorithm and perturbation method. Recent reviews of these finite algorithms are presented in Hunter [6].

In 2015, J. Xu [7] constructed an iterative method with parameters for the computing of MFPTs. Compared with the classical finite algorithm, the iterative method not only guarantees the accuracy of computation, but also greatly improves the scale and stability. At the same time, J. Xu also pointed out that the determination of iterative parameter α is a difficult problem. The parameter α is decided by the transition matrix and experimental calculation. In this paper, based on the definition equation of the mean first passage times matrix, the computation of MFPTs is transformed into a set of linear equations to be solved. These equations are all compatible, but the coefficient matrix may be singular. It is proved that the minimum norm solution of these equations form the mean first passage times matrix. Then, the least norm solution of these equations is solved uniformly by the linear least square algorithm (LS). As far as the author knows, this is the first time for computing the MFPTs by LS algorithm. Numerical experiments show that this scheme is effective.

This article is organized as follows. In Section 2 the definition equation of the matrix of the mean first passage times of a n state Markov chain and the related knowledge of the generalized inverse of the matrix are introduction without proofs. In Section 3 a set of linear equations are constructed, and the equivalence of this set of equations to the original matrix equation satisfied by MFPTs is proved. Last section, a number of numerical examples are given. The LS algorithm is compared with Hunter's finite algorithm and Xu's iteration method in which the results show that the linear least squares algorithm is effective in computing the mean first passage times of Large-scale Markov chain.

2 Preliminaries

We set the scene by reintroducing the notation and theory that are stated in Hunter[6]. Let $\{X_n, n \geq 0\}$ be a finite Markov chain with state space $S = \{1, 2, \cdots, m\}$ and transition matrix $P = (p_{ij})$, where $p_{ij} = P\{X_n = j | X_{n-1} = i\}$ for all $i, j \in S$. In this article, we focus on regular Markov chain (MC). The first passage time T_{ij} is the length of time to go from a state i to a state j for the first time and we define $m_{ij} = E[T_{ij}|X_0 = i]$ as the mean first passage times from state i to state j. The mean first passage matrix, denoted by $M = (m_{ij})$, is the matrix with entries $m_{ij} = E[T_{ij}|X_0 = i]$.

Theorem 2.1. Let P be a transition matrix of a homogeneous n state Markov chain. X_d be the diagonal matrix formed from the diagonal elements of matrix X. $J = [1]$(i.e. each element of J is 1). The matrix M of mean first passage times is unique solution of the matrix equation[7, 8]

$$(I - P)X = J - PX_d. \tag{2.1}$$

Definition 2.2. [9] Let $A \in \mathbb{C}^{m\times n}$. The Moore-Penrose generalized inverse of A denoted by A^+ is the unique solution of four matrix equations: $AXA = A, XAX = X, (AX)' = AX, (XA)' = XA$, where prime $'$ stands for the conjugate transpose of a matrix.

If $A \in \mathbb{C}^{m\times m}$ and A is nonsingular, then $A^+ = A^{-1}$.

3 Construction of a Set of Equivalent Linear Equations

J. Xu[7] gives an equations with parameters α for MFPTs

$$(I-\alpha P)X_{k+1} = J + (1-\alpha)PX_k - PX_{dk}, \tag{3.1}$$

which α satisfies $0 \leq \alpha < 1$. Then, Xu constructed an iterative algorithm with parameters α for computing MFPTs and proved the convergence of this algorithm[7, Thm 3.2].
(AlgXu)Suppose that$(I-\alpha P)$admits an LU factorization.

Choose an initial approximation $X^{(0)} \in \mathbb{R}^{n\times n}$, tolerance $\varepsilon > 0$, and a matrix Frobenius norm $\|\cdot\|$, starting with $k = 1$.

(1) Compute $Y^{(k-1)} = J + (1-\alpha)PX^{(k-1)} - PX_d^{(k-1)}$;
(2) Solve $LY^{(k-1/2)} = Y^{(k-1)}$;
(3) Solve $UX^{(k)} = Y^{\frac{k-1}{2}}$;
(4) If $\|X^{(k)} - X^{(k-1)}\| < \varepsilon$, then return $M \approx X^{(k)}$ and stop; otherwise increase k by 1 and continue with step (1).

Compared with the classical finite algorithm Hunter[6] , the algorithm AlgXu is of higher accuracy and better stability. But it is still a difficult problem that choice of the best iteration parameter α. In addition, the algorithm AlgXu has to solve two trigangular equations in every iteration step by step, and the iterative format (3.1) is not a standard iteration form, so it is difficult to analyze convergence.

Now we construct a set of equivalent linear equations to (2.1).

Let e denote the column vector of n ones and $X = (x_1, x_2, \cdots, x_n)$. Then the matrix equation (2.1) is equivalent to the following n linear equations

$$x_i = Px_{id} + e, \qquad i = 1, 2, \cdots, n, \tag{3.2}$$

where $x_{id} = (x_{i1}, \cdots, x_{i,i-1}, 0, x_{i,i+1}, \cdots, x_{in})^T$. Let P_i be the matrix formed by replacing each element of column i of matrix P with 0 and $A_i = (I - P_i)$. Then (3.2) is equivalent to

$$A_i x_i = e, \qquad i = 1, 2, \cdots, n. \tag{3.3}$$

The matrix $A = I - P$ is singular, but $A_i = I - P_i$ may be singular or nonsingular. However, it is always compatible for every i to equations (3.3). Therefore, whether A_i is singular or not, its minimal norm solution is always $A_i^+ e$.

The relation between the unique solution of (2.1) and the minimal norm solution of series equations (3.3) is given below.

Theorem 3.1. Suppose the unique solution of matrix equation (2.1) is $X = (x_1, x_2, \cdots, x_n)$, then $x_i = A_i^+ e, i = 1, 2, \cdots, n$.

Proof: First, notice that for any matrix Q, QQ^+ is a projection on $R(Q)$ along $N(Q')$. If $x \in R(Q)$, then $QQ^+x = x$ or $(I - QQ^+)x = 0$.

Next, let $P_i' = P - P_i$, $P = (p_1, p_2, \cdots, p_n)$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T$, the minimal norm solution of of equations (3.3) is $x_i = A_i^+ e$. Thus

$$(I-P)x_i = (A_i - P_i')x_i = A_i x_i - P_i' x_i = A_i A_i^+ e - P_i' x_i = e - x_{ii} p_i.$$

Therefore, if $X = (x_1, x_2, \cdots, x_n)$, then $(I - P)X = AX = J - PX_d$. □

In summary, the computation problem of MFPTs can be reduced to solving the minimal norm solution of a set of equations. As is known to all, there are many effective algorithms for solving the minimal norm solutions of systems with full rank or deficient rank. Now, we use the linear least squares algorithm (LS for short) based on QR decomposition to compute $A_i^+ e$. The implementation details of the LS algorithm see [10, Chapter 5], which is no longer given here.

4 Numerical Experiment and Analysis

Two kinds of numerical experiments are done in this section. Finite algorithm and iterative method are chosen to compare with LS algorithm. Two finite algorithm are selected from the Hunter[6], the algorithm Proc2(HP2) and Proc4(HP4), which have better computational efficiency for different dimensional problems, and the computing of the inverse of matrix A in HP2 and HP4 is directly called the inv function command in program language. The iterative algorithm is selected from the J.Xu[7] algorithm.

The three test index is selected from Hunter[6]:

1. Computational time. A comparison of results running on the same machine is given.
2. Percentage of Zero errors(PZE). Let $M = \{m_{ij}\}$ be the mean first passage times, $\varepsilon_{ij} \equiv m_{ij} - \sum_{k \neq j} p_{ik} m_{kj} - 1$. PZE be the percentage of error terms ε_{ij} that are zero.
3. Overall Residual errors(ORE). ORE$\equiv \sum_{i=1}^{m} \sum_{j=1}^{m} |\varepsilon_{ij}|$.

First, we compare three indicators above with 4 examples from different literatures, [6, 7, 11]. In this part, each algorithm runs 20 times and then takes the average value. Secondly, we compare the advantages and disadvantages of finite algorithm and iterative algorithm with two large scale examples.

Example 1. In this case, we choose 4 test matrices $P_1$to P_4[6, 7, 11] for comparison.

$$P_1 = \begin{bmatrix} 0.136267 & 0.292549 & 0.266992 & 0.220856 & 0.083335 \\ 0.198798 & 0.019347 & 0.129998 & 0.321252 & 0.330605 \\ 0.246269 & 0.215116 & 0.044021 & 0.249831 & 0.244763 \\ 0.400950 & 0.149352 & 0.012546 & 0.303336 & 0.133815 \\ 0.200328 & 0.084084 & 0.351278 & 0.337325 & 0.026985 \end{bmatrix}.$$

$$P_2 = \begin{bmatrix} 0.268031 & 0.255740 & 0.201497 & 0.265012 & 0.007385 & 0.002335 \\ 0.166582 & 0.137728 & 0.032748 & 0.118446 & 0.187835 & 0.356660 \\ 0.093279 & 0.226108 & 0.081331 & 0.206803 & 0.094199 & 0.298281 \\ 0.103853 & 0.230590 & 0.261709 & 0.069110 & 0.061473 & 0.273265 \\ 0.101657 & 0.261742 & 0.128131 & 0.002138 & 0.204864 & 0.301467 \\ 0.216100 & 0.210158 & 0.154059 & 0.178624 & 0.213131 & 0.027928 \end{bmatrix}.$$

$$P_3 = \begin{bmatrix} 0.000000 & 0.701299 & 0.298701 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.437907 & 0.562093 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.632082 & 0.367918 \\ 0.471475 & 0.000000 & 0.000000 & 0.000000 & 0.528525 \\ 0.461323 & 0.538677 & 0.000000 & 0.000000 & 0.000000 \end{bmatrix}.$$

$$P_4 = \begin{bmatrix} 0.999999 & 1e-7 & 2e-7 & 3e-7 & 4e-7 \\ 0.4 & 0.3 & 0 & 0 & 0.3 \\ 5e-7 & 0 & 0.999999 & 0 & 5e-7 \\ 5e-7 & 0 & 0 & 0.999999 & 5e-7 \\ 2e-7 & 3e-7 & 1e-7 & 4e-7 & 0.999999 \end{bmatrix}.$$

The results of the 4 examples above are shown in comparison Tables 1, 2 and 3. In the first two examples, the PZE index of the LS algorithm is the best, followed by the two finite algorithm. In the last two examples, the PZE index of HP2 algorithm is the best, followed by LS and HP4A. Xu algorithm is not ideal.

Table 1. Average of computation times (seconds)

	P_1	P_2	P_3	P_4
HP2	1.0312e-04	.6912e-04	3.3034e-04	3.5310e-04
XU	0.0045	0.0068	0.0034	0.1450
LS	0.0051	0.0060	0.0046	0.0069

Table 2. Percentage of Zero Error

	P_1	P_2	P_3	P_4
HP2	0.2000	0.2222	0.6400	0.6000
XU	0.2900	0.1667	0	–
LS	0.4200	0.4722	0.5600	0.5300

Table 3. Overall Residual Error

	P_1	P_2	P_3	P_4
HP2	4.9652e-05	1.0995e-04	7.1054e-15	7.5147e-09
XU	1.1976e-09	2.1252e-09	1.3786e-09	–
LS	1.3323e-14	2.2204e-14	7.1054e-15	7.6613e-09

Example 2.

Test matrix in this example is sparse irreducible transition matrix generated randomly [6]. The scale of the test matrix is from the 10 to the 510. The Xu algorithm does not work well, so it is no longer considered here. Only HP2, HP4A algorithm and linear least square algorithm with QR decomposition (LSQR) are compared.

The generation code for the test matrix P is as follows Hunter [6] :

```
Input n;
a=0.4;
P=rand(n);
P(P>a)=0;
P=P-diag(diag(P))
c=1./sum(P');
for i=1:n
    P(i,:)=P(i,:)*c(i);
end
```

In addition, the generated matrix has been checked for irreducibility.

Example 3.

Test matrix in this example is one dimensional random walk transition matrix[11]. The scale of the test matrix is from the 100 to the 2000. The efficiencies of HP4A algorithm and Xu algorithm are not good, so only HP2 and LSQR are compared.

Test matrix[11]

$$P_8 = \begin{bmatrix} 0.75 & 0.25 & 0 & \cdots & \cdots & 0 \\ 0.25 & 0.50 & 0.25 & \ddots & \ddots & \vdots \\ 0 & 0.25 & 0.50 & 0.25 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0.25 & 0.50 & 0.25 \\ 0 & \cdots & \cdots & 0 & 0.25 & 0.75 \end{bmatrix}.$$

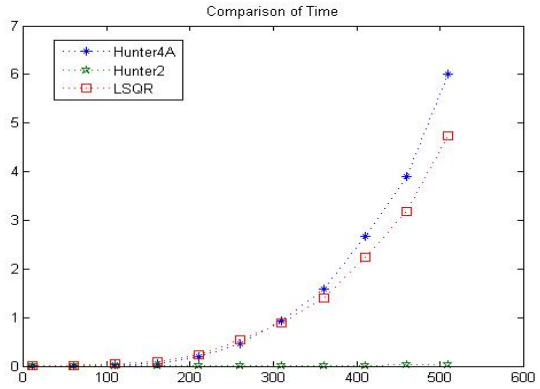

Fig. 1. Computation times for LS, HP2 and HP4A (random generating matrix)

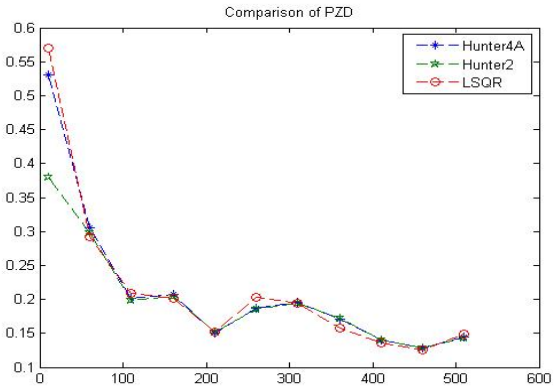

Fig. 2. PZE for LSQR, HP2 and HP4A (random generating matrix)

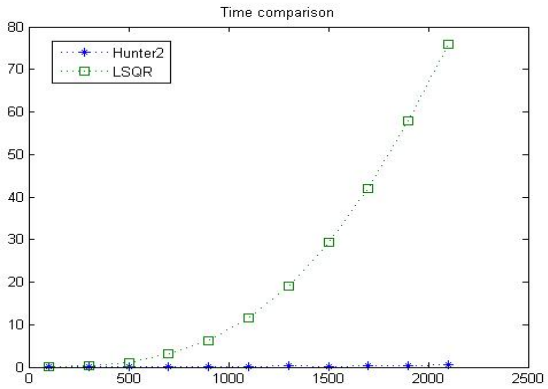

Fig. 3. Computation times for LSQR and HP2 (random walk matrix)

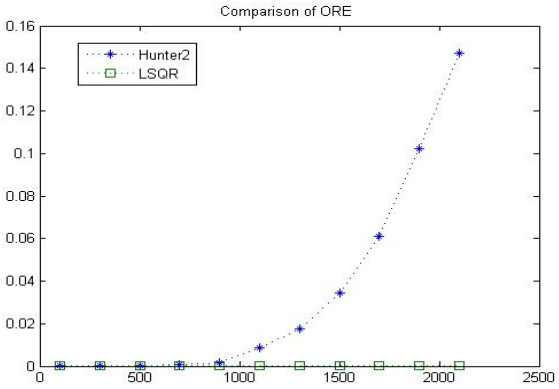

Fig. 4. ORE for LSQR and HP2 (random walk matrix)

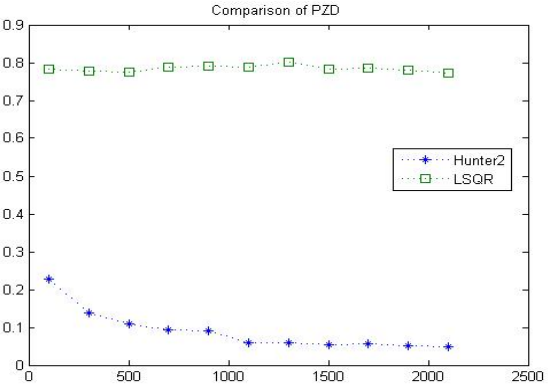

Fig. 5. PZE for LSQR and HP2 (random walk matrix)

From the results presented in Figs. 3, 4 and 5, it is obvious that with the increase of matrix order, the computation time of LS algorithm increases greatly, while that of HP2 algorithm is relatively small. The index ORE of the LS algorithm is the magnitude of $1e-4$ and the HP2 algorithm magnitude is $1e-1$. For the important index, PZE, LS is stable around 0.8, and HP2 decreases from 0.23 to 0.05. It can be seen that LS algorithm can maintain high precision convergence in addition to long computation time.

5 Conclusions

The above examples show the effectiveness of the LS algorithm. But the computation time of LS algorithm is relatively long with finite algorithm HP2. It is noteworthy that m independent linear equations are solved every time, so this kind of algorithm is very easy to implement in parallel. This may effectively reduce the computation time.

In addition, people may be able to directly computation the generalized inverse matrix A_i by parallel algorithm, so as to improve the computational efficiency.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Meyer CD. The condition of a finite Markov chain and perturbation bounds for the limiting probabilities[J]. Siam Journal of Algebraic and Discrete Methods. 1980;1(3):273-283.

[2] Hunter JJ. Stationary distributions and mean first passage times in Markov chains using generalized inverses[J]. Asia Pacific Journal of Operational Research. 1992;9(2):145-153.

[3] Grassmann WK, Taksar MI, Heyman DP. Regenerative analysis and steady state distributions for Markov chains. Informs.1985;33(5):1107-1116.

[4] Hunter JJ. The computation of key properties of Markov chains via perturbations[J]. Linear Algebra and Its Applications. 2016;511:176-202.

[5] Hunter JJ. Accurate calculations of stationary distributions and mean first passage times in Markov renewal processes and Markov chains. Special Matrices. 2016;4(1):151-175.

[6] Hunter JJ. The computation of the mean first passage times for Markov chains. Linear Algebra and Its Applications. 2018;549:100-122.

[7] Xu J. An iterative algorithm for computing mean first passage times of Markov Chains. Applied Mathematics and Computation. 2015;250:372-389.

[8] Kemeny JG, Snell JL. Finite Markov chains. Springer-Verlag New York, Inc; 1976.

[9] Ben-Israel, Greville TNE. Generalized inverses:Theory and applications (2nd ed). 2003; Springer-Verlag New York, Inc.

[10] Gene H. Golub, Charles F. Van loan. Matrix computations. Third edition. 1996; Johns Hopkins University Press, USA

[11] Ching WK, Huang X, Michael K. Ng, Siu. TK. Markov chains: Models. Algorithms and Applications. 2012; Springer-Verlag New York, Inc.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sciencedomain.org/review-history/26158