Journal of Mathematics and Statistics 10 (3): 331-338, 2014
ISSN: 1549-3644

doi:10.3844/jmssp.2014.331.338 Published Online 10 (3) 2014 (http://www.thescipub.com/jmss.toc)

USING COLUMN GENERATION TECHNIQUE TO ESTIMATE PROBABILITY STATISTICS IN TRANSITION MATRIX OF LARGE SCALE MARKOV CHAIN WITH LEAST ABSOLUTE DEVIATION CRITERIA

Wanida Lerspipatthananon and Peerayuth Charnsethikul

Center of Advanced Studies in Industrial Technology, Department of Industrial Engineering, Faculty of Engineering, Kasetsart University, Bangkok, 10900, Thailand

Received 2014-07-10; Revised 2014-07-24; Accepted 2014-08-02

ABSTRACT

The Least Absolute Deviation (LAD) method is the one of many methods used to estimate transition probability matrix of Markov Chain. It can be formulated as a Linear Programming Problem (LP) and solved using its regular state of the art software. However, when the Markov Chain has a large number of states and historical state probabilities data, the corresponding LP size can be very large reaching computer hardware/software limitation. The aim of this study is to apply the Column Generation (CG) technique to solve this large scale LP and to evaluate its extension beyond direct hardware/software capabilities. In this study, the sample state probabilities data were simulated statistically and two methods were used to solve the problem. The first method was using 'linprog' function in MATLAB to solve the related LP that all decision variables were considered simultaneously while the others was the CG method expected to require a much less percentage of all variables. As result effectiveness, both methods solved all test problems resulting equal LADs each. The CG method required more average time. Nevertheless, less than 30% of decision variables were considered in the CG method. The lesser percentages were found as the problem size grew. Moreover, larger size problems beyond direct use of software were solved using the proposed approach.

Keywords: Transition Matrix, Markov Chain, Linear Programming Problem, Column Generation, Least Absolute Deviation

1. INTRODUCTION

Markov chain is widely used as model in many areas i.e., economics, marketing, capital theory, industrial structure, demography and social science which shown in the review of Dent and Ballintine (1971). Later, the Markov models have also received attention from researchers in various disciplines such as Craig and Sendi (2002) used Markov chain to study the changes of chronic diseases. Jones (2005; Christodoulakis, 2006; Simister, 2007) estimated the transition matrix of Markov chain to forecast the credit risk. Thyagarajan and Mohamed (2005) used Markov chain to analyze the retail banking. Lin (2011) applied Markov chain to establish an adaptive production procurement system.

Constant and Zimmermann (2013) studied the migration of the population with Markov chain. Markov model is used to describe the chance of an event occurring in the next period when we know the chance of that event in the present period and transition probability matrix in the following manner:

$$\pi(k+1) = \mathrm{P}^{\mathrm{T}}\pi(k), k = 1,2,\ldots,S-1 \quad (1)$$

where, $\pi(k)$, $\pi(k+1)$ is the column vector order n showing the probabilities of each state in period k and

Corresponding Author: Wanida Lerspipatthananon, Center of Advanced Studies in Industrial Technology, Department of Industrial Engineering, Faculty of Engineering, Kasetsart University, Bangkok, 10900, Thailand

k+1, respectively. P = [p_{ij}] is a square matrix order n called transition probability matrix. The assumptions of P are presented in Equation 2 and 3.

$$Pe = e \tag{2}$$

$$0 \le p_{ij} \le 1 \tag{3}$$

where, e is the n-element column vector of unit, n is the number of states.

There are many researchers who proposed the methods to estimate probability statistics in transition matrices. Lee *et al*. (1968) suggested the Maximum Likelihood and Bayesian estimation. Madansky (1959; Kalbfleisch and Lawless, 1984) constructed the estimators by the Weighted Least Square method. The Least Square method without regard to assumption Equation 2 and 3 were used by Miller (1952; Goodman, 1953; Telser, 1966). Lee *et al*. (1965; Theil and Ray, 1966) offered the Restricted Least Square method that regard to Equation 2 and 3. The Minimum Absolute Deviation (MAD) was proposed by Ashar and Wallace (1963). In this study, the Restricted Least Squared (RLS) method is ignored because it is equivalent to solve the Quadratic Programming Problem (QPP). The solution of QPP is complex. As a result, it cannot solve a large scale problem. Therefore, the less complicated method than RLS will be diagnosed. The MAD method called Least Absolute Deviation (LAD) under constraints in Equation 2 and 3 is used to estimate the elements of the transition matrix when the approximation of $\pi(k)$ was known. It shows that:

$$Minimize \sum_{k=1}^{S-1} \sum_{j=1}^{n} \left| \pi_j^{(k+1)} - \sum_{i=1}^{n} p_{ij} \pi_i^{(k)} \right| \tag{4}$$

$$subject\ to \sum_{j=1}^{n} p_{ij} = 1, \forall_i \tag{5}$$

$$p_{ij} \ge 0, \forall_{i,j} \tag{6}$$

The above model (A) consisted of Equation 4-6 has n linear constraints, n^2 of non-negativity constraints and decision variables. It can be reformulated as the following LP model:

$$Minimize \sum_{k=1}^{S-1} \sum_{j=1}^{n} u_{j(k)} \tag{7}$$

$$subject\ to - u_{j(k)} - \sum_{i=1}^{n} \pi_i^{(k)} p_{ij} \le -\pi_j^{(k+1)}, \forall_{j,k} \tag{8}$$

$$-u_{j(k)} + \sum_{i=1}^{n} \pi_i^{(k)} p_{ij} \le \pi_j^{(k+1)}, \forall_{j,k} \tag{9}$$

$$\sum_{j=1}^{n} p_{ij} = 1, \forall_i \tag{10}$$

$$p_{ij} \ge 0, \forall_{i,j} \tag{11}$$

$$u_{j(k)} \ge 0, \forall_{j,k} \tag{12}$$

Equations 7-12 are the components of the corresponding LP problem referred as model (B) with n+$2n$(S-1) of linear constraints and n^2 + n(S-1) of non-negativity constraints and decision variables. The explicitly comparison of the model (A) and (B) are shown in **Table 1**. The number of variables and constraints of model (A) are influenced by the number of states n of Markov Chain while the model (B) depends on both the number of states and the number of periods of historical data S. Therefore, the number of variables grows much faster than the number of constraints. If the number of states n or the number of periods S is very large up to computer capacity, the formulated LP can be unsolvable. This study is aimed to propose the use of column generation technique to solve this obstacle. In section 2, the proposed column generation algorithm is presented with an illustrated example in section 3. The computational results with discussions from the numerical experiment are shown in section 4. The conclusions of this study are finally summarized in section 5.

Table 1. The comparison of model (A) and (B)

Model	Objective function	Number of variables	Number of constraints
(A)	Non-linear	n^2	n
(B)	Linear	n^2 + n(S-1)	n+2n(S-1)

2. COLUM GENERATION

The processes of simplex method are reviewed. It shows that, for the minimization problem, the columns of constraint matrix that is nonnegative reduced cost are only used to determine. If the negative reduced cost is found, a column corresponding to this reduced cost will be produced. The technique that generates the coefficients associated with a variable as needed is called Column Generation (CG). The expectation of CG is that it can optimize the LP by examining only a subset of vector coefficients and it can avoid unnecessary computations. Therefore CG is often used to solve the large size LP. The concept of CG algorithm can be illustrated via the model (B) called Master Problem (MP), let X is a column vector of decision variables, c is coefficient vector of objective function, b is vector of right hand side of constraint and A is constraint matrix as follows:

$$X = \begin{bmatrix} U_{(1)} \\ U_{(2)} \\ \vdots \\ U_{(S-1)} \\ P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \text{ where } U_{(k)} = \begin{bmatrix} u_{1(k)} \\ u_{2(k)} \\ \vdots \\ u_{n(k)} \end{bmatrix}; k = 1,2,\ldots,S-1$$

$$\text{and } P_i = \begin{bmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_{in} \end{bmatrix}; \; i = 1,2,\ldots,n,$$

$$c = \begin{bmatrix} [1]_{1\times n(S-1)} & [0]_{1\times n^2} \end{bmatrix},$$

$$b = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{S-1} \\ [1]_{n\times 1} \end{bmatrix} \text{ where } h_k = \begin{bmatrix} -\pi_1^{(k+1)} \\ \pi_1^{(k+1)} \\ \vdots \\ -\pi_n^{(k+1)} \\ \pi_n^{(k+1)} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{(1)} & [0]_{2n\times n} & \cdots & [0]_{2n\times n} & d^{(1)} \\ [0]_{2n\times n} & a_{(2)} & \cdots & [0]_{2n\times n} & d^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ [0]_{2n\times n} & [0]_{2n\times n} & \cdots & a_{(S-1)} & d^{(S-1)} \\ & [0]_{n\times n(S-1)} & & & I \end{bmatrix}$$

$$\text{where, } a_{(k)} = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix},$$

$$d^{(k)} = \begin{bmatrix} \pi^{(k)} & [0]_{2\times n} & \cdots & [0]_{2\times n} \\ [0]_{2\times n} & \pi^{(k)} & \cdots & [0]_{2\times n} \\ \vdots & \vdots & \ddots & \vdots \\ [0]_{2\times n} & [0]_{2\times n} & \cdots & \pi^{(k)} \end{bmatrix},$$

$$\pi^{(k)} = \begin{bmatrix} -\pi_1^{(k)} & -\pi_2^{(k)} & \cdots & -\pi_n^{(k)} \\ +\pi_1^{(k)} & +\pi_2^{(k)} & \cdots & +\pi_n^{(k)} \end{bmatrix}$$

$$\text{And } I = \begin{bmatrix} I_n | I_n | \cdots | I_n \end{bmatrix}$$

Let N is the number of decision variables or the number of column in constraint matrix A which is equal to $n(S\text{-}1)+n^2$. If $r<N$, then A can be written in form:

$$A = \begin{bmatrix} A_r | A_{N-r} \end{bmatrix}$$

where, A_r is a sub-matrix of constraint matrix that contains coefficient of basic variables from Basic Feasible Solution (BFS) and $A_{N\text{-}r}$ is sub-matrix of coefficient of the remaining variables or non-basic variables.

The model that considers only the r variables is reduced to smaller problem of manageable size which is called Restricted Master Problem (RMP). An optimal solution of the RMP is not necessarily to be the optimal for the MP. An optimal dual solution of RMP (y) is used to calculate the reduced cost of non-basic variables. For choosing a variable to enter the basis the sub-problem is solved in the following manner:

$$\text{Minimize}\left\{\tilde{c}_j = c_j - y^T A_j | \tilde{c}_j < 0\right\} \; for \; j = r+1,\ldots,N$$

Assume that the variable X_t is entering variable, a column of coefficients which according to X_t is generated and added to RMP. The above process is implemented until the objective function cannot be improved.

In this study, the summary of the CG algorithm for estimating all elements in the transition matrix of Markov Chain by LAD method reformulated into LP model is given below.

2.1. Initialization

Choose the BFS (x_B is a subvector of x). Assuming the values of $U_{j(k)}$, for all j, k and p_{ij}, for $i = j$ are 1. So the x_B consists of $U_{j(k)}$, for all j, k and p_{ij}, for $i = j$. It means that there are $n(S\text{-}1)+n$ of the number of variables in RMP.

2.2. Main Step

- Solve the RMP with 'linprog' function of MATLAB
- Use the vector of the shadow price or dual variables (y) from step 1 to calculate the reduced cost of non-basic variables in the following manner: $\tilde{c}_j = c_j - y^T A_j$ for $j = r+1,\dots,N$
- If $\tilde{c}_j \geq 0$ for every j, then the result from step 1 is optimal solution and if not, a column A_t that according to $\tilde{c}_j = \min\{\tilde{c}_j | \tilde{c}_j < 0\}$ will be generated and added to RMP and repeat to step 1

The further detail of CG technique can be studied in Bazaraa *et al*. (2010; Lasdon, 1970; Nash and Sofer, 1996). The detail of 'linprog' function can be examined at the website of MATLAB (2013).

3. EXAMPLE

In this section, the example drawn from (Theil and Rey, 1966) is considered. The market shares of three cigarette brands in the United States during the period 1939-43 are presented in **Table 2**.

The vector of X, c, b and the matrix of A are shown in **Table 3**.

Table 2. Market shares of three cigarette brands

	Camel	Lucky strike	Chesterfield
1941	0.3579	0.3653	0.2768
1942	0.3527	0.3851	0.2622
1943	0.3276	0.3875	0.2849

Table 3. The vector of X, c, b and the matrix of A from example data

	$u_{1(1)}$	$u_{2(1)}$	$u_{3(1)}$	$u_{1(2)}$	$u_{2(2)}$	$u_{3(2)}$	P_{11}	P_{21}	P_{31}	P_{12}	P_{22}	P_{32}	P_{13}	P_{23}	P_{33}	b
$c =$	1	1	1	1	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
	-1	0	0	0	0	0	-0.3579	-0.3653	-0.2768	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.3527
	-1	0	0	0	0	0	0.3579	0.3653	0.2768	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3527
	0	-1	0	0	0	0	0.0000	0.0000	0.0000	-0.3579	-0.3653	-0.2768	0.0000	0.0000	0.0000	-0.3851
	0	-1	0	0	0	0	0.0000	0.0000	0.0000	0.3579	0.3653	0.2768	00.0000	0.0000	0.0000	0.3851
	0	0	-1	0	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.3579	-0.3653	-0.2768	-0.2622
	0	0	-1	0	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3579	0.3653	0.2768	0.2622
	0	0	0	-1	0	0	-0.3527	-0.3851	-0.2622	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.3276
$A=$	0	0	0	-1	0	0	0.3527	0.3851	0.2622	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3276
	0	0	0	0	-1	0	0.0000	0.0000	0.0000	-0.3527	-0.3851	-0.2622	0.0000	0.0000	0.0000	-0.3875
	0	0	0	0	-1	0	0.0000	0.0000	0.0000	0.3527	0.3851	0.2622	0.0000	0.0000	0.0000	0.3875
	0	0	0	0	0	-1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.3527	-0.3851	-0.2622	-0.2849
	0	0	0	0	0	-1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3527	0.3851	0.2622	0.2849
	0	0	0	0	0	0	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
	0	0	0	0	0	0	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	1.0000
	0	0	0	0	0	0	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	1.0000

Table 4. The vector of X, c, b and the matrix of A for restricted master problem

	$u_{1(1)}$	$u_{2(1)}$	$u_{3(1)}$	$u_{1(2)}$	$u_{2(2)}$	$u_{3(2)}$	P_{11}	P_{22}	P_{33}	b
$c_r =$	1	1	1	1	1	1	0.0000	0.0000	0.0000	
	-1	0	0	0	0	0	-0.3579	0.0000	0.0000	-0.3527
	-1	0	0	0	0	0	0.3579	0.0000	0.0000	0.3527
	0	-1	0	0	0	0	0.0000	-0.3653	0.0000	-0.3851
	0	-1	0	0	0	0	0.0000	0.3653	0.0000	0.3851
	0	0	-1	0	0	0	0.0000	0.0000	-0.2768	-0.2622
	0	0	-1	0	0	0	0.0000	0.0000	0.2768	0.2622
	0	0	0	-1	0	0	-0.3527	0.0000	0.0000	-0.3276
$A_r =$	0	0	0	-1	0	0	0.3527	0.0000	0.0000	0.3276
	0	0	0	0	-1	0	0.0000	-0.3851	0.0000	-0.3875
	0	0	0	0	-1	0	0.0000	0.3851	0.0000	0.3875
	0	0	0	0	0	-1	0.0000	0.0000	-0.2622	-0.2849
	0	0	0	0	0	-1	0.0000	0.0000	0.2622	0.2849
	0	0	0	0	0	0	1.0000	0.0000	0.0000	1.0000
	0	0	0	0	0	0	0.0000	1.0000	0.0000	1.0000
	0	0	0	0	0	0	0.0000	0.0000	1.0000	1.0000

Table 5. The results from CG technique

Iteration	No. of available variables	LAD	$\tilde{c}_j$ p_{21}	p_{31}	p_{12}	p_{32}	p_{13}	p_{23}	The variable add to RMP
1	9	0.0898	1.50	0.52	-1.42*	-0.55	-0.71	0.73	p_{12}
2	10	0.0746	0.00	-0.03	0.00	-0.03	0.01	-0.04*	p_{23}
3	11	0.0723	0.00	-0.06*	0.00	-0.06	0.05	0.00	p_{31}
4	12	0.0188	0.05	0.00	0.00	0.02	0.00	0.00	-

* is $\min\{\tilde{c}_j \mid \tilde{c}_j < 0\}$

The CG technique is used to solve this model. For initialization step, assuming the values of $U_{j(k)}$; j = 1,2,3; k = 1,2 and p_{11}, p_{22}, p_{33} are 1. So the RMP that is used to solve in first step of main step consists of all vectors and matrix which shown in **Table 4**.

In iteration 1, there are 9 available variables in RMP. The 'linprog' function of MATLAB is used to the RMP. The value of LAD is 0.0898. The reduced cost $(\tilde{c}_j)$ of all non-basic variables are calculated in the manner that show in step 2 of main step. It found that, the reduced cost of p_{12}, p_{32} and p_{13} are negative and the minimum of them is p_{12}. It means that the 0.0890 of LAD is not minimum value. Therefore a column of constraint matrix that according to p_{12} is generated and added to RMP. The number of available variables in iteration 2 is 10. Following the main step in above section until the reduced cost of all non-basic variables are nonnegative. The results of each iteration are presented in **Table 5.**

The result of last iteration is:

$$X = \begin{bmatrix} U_{(1)} \\ U_{(2)} \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \text{ where } U_{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ U_{(2)} = \begin{bmatrix} 0.0094 \\ 0 \\ 0.0085 \end{bmatrix},$$

$$P_1 = \begin{bmatrix} 0.2121 \\ 0 \\ 1 \end{bmatrix},\ P_2 = \begin{bmatrix} 0.7879 \\ 0.2822 \\ 0 \end{bmatrix},\ P_1 = \begin{bmatrix} 0 \\ 0.7178 \\ 0 \end{bmatrix}$$

4. NUMERICAL RESULTS

In this study an Intel(R) Core(TM) i3-2120 3.30 GHz CPU 4 GB RAM is used to perform the experiments. The maximum possible array is 640 MB which the product of number of rows and number of columns of the constraint matrix must not exceed 6.4×10^8 bytes. The trials start on generating square transition probability matrix P order n and $n\times S$ matrix of data according to the Equation 1 Subsequently, prepare the matrices and vectors according to the requirements of the 'linprog' function in MATLAB. The next step is solving the LP by two methods one is directly solved by 'linprog' function that all of decision variables are available and the other is using CG method as the algorithm shown in section 2.

The numerical computation is presented in the **Table 6**. It is found that when the size of constraint matrix of model (B) is not over 10^8 bytes, the directly solved by 'linprog' function can be used to optimize the LP. It has less average time to estimation than CG method. Both methods give the same average of LAD in every scenarios. Note that in the solution when the number of periods for data collection S is 10, CG method use less than 30% of variables while the directly solved by 'linprog' function requires all variables. The number of variables that use in CG technique is increased when S is expanded. If the size of constraint matrix of model (B) is over 10^8 bytes, the directly solved by 'linprog' function cannot be used, but CG can.

The comparison between the RLS and LAD method is shown in the **Table 7 and 8**. It is found that, the LAD method has surprisingly less SSE than the RLS on average implying that solving the problem linearly can maintain a better numerical stability computationally. The average time to estimation of LAD method which directly solved by 'linprog' function will be less than RLS method when the number of periods for data collection S is 10. In the case that the number of states n is more than 300, the LAD method which solve by CG technique is only method that suitable to estimate the transition matrix.

Table 6. The numerical computation of each scenarios

		Model (B)		
S	*n*	Number of constraints	Number of decision variables	Size of the constraint matrix of model (B) (byte)
10	60	1,140	4,140	4.7196×10^6
	80	1,520	7,120	1.0822×10^6
	100	1,900	10,900	2.0710×10^7
	120	2,280	15,480	3.5294×10^7
	140	2,660	20,860	5.5488×10^7
	160	3,040	27,040	8.2202×10^7
	180	3,420	34,020	1.1635×10^8
	200	3,800	41,800	1.5884×10^8
	250	4,750	64,750	3.0763×10^8
	300	5,700	92,700	5.2839×10^8
	330	6,270	111,870	7.0142×10^8
	400	7,600	163,600	1.2433×10^9
20	200	7,800	43,800	3.4164×10^8
30	140	8,260	23,660	1.9543×10^8
100	60	11,940	9,540	1.1391×10^8

Table 6. The numerical computation of each scenarios (continue)

			Directly solved by 'linprog'		CG method		
S	*n*	Mean of LAD	Time to Estimation (sec)	Number of variables considered in the solution	Time to Estimation (sec)	Number of variables considered in the solution	Percentage of variables considered in CG method
10	60	0.0657	11.6181	4,140	64.7237	1,117	26.97
	80	0.0290	51.2985	7,120	221.8648	1,520	21.35
	100	0.0290	74.4539	10,900	275.9748	1,822	16.71
	120	0.0274	175.6527	15,480	595.0517	2,272	14.67
	140	0.0237	294.6160	20,860	761.7571	2,614	12.53
	160	0.0115	392.9262	27,040	987.3406	2,877	10.64
	180	0.0241	577.6557	34,020	1449.2400	3,250	9.55
	200	0.0205	1175.1520	41,800	1497.7800	3,435	8.22
	250	0.0226	2290.8500	64,750	4170.2000	4,301	6.64
	300	0.0101	3991.7000	92,700	8299.9000	5,207	5.62
	330	0.0148	Out of memory	111,870	10684.2300	5,596	5.00
	400	0.0139	Out of memory	163,600	50754.0000	8,382	5.12
20	200	0.0416	Out of memory	43,800	8777.3000	5,677	12.96
30	140	0.0874	Out of memory	23,660	8811.4000	5,601	23.67
100	60	0.9702	Out of memory	9,540	26,307.0000	6,802	71.30

Table 7. The sum of square error for each scenario

		SSE		% of SSE difference
S	*n*	RLS	LAD	$\left(\frac{SSE_{RLS}-SSE_{LAD}}{SSE_{RLS}}\right)\times100$
10	60	0.00004	0.000030	25.00
	80	0.00003	0.000020	33.30
	100	0.00003	0.000010	66.70
	120	0.00004	0.000002	95.00
	140	0.00005	0.000003	94.00
	160	0.00012	0.000008	93.30
	180	0.00002	0.000009	55.00
	200	0.00001	0.000002	80.00
	250	-	-	-
	300	-	-	-
	330	-	-	-
	400	-	-	-
20	60	0.00011	0.00007	36.36
	80	0.00004	0.00003	25.00
	100	0.00005	0.00002	60.00
30	60	0.00012	0.00012	0.00
	80	0.00010	0.00006	40.00
	100	0.00006	0.00003	50.00

Table 8. The average time to estimate transition matrix for each scenario

		Time to estimation (second)				
			LAD		% of time difference	% of time difference
S	n	RLS	Directly	CG	$\left(\frac{Time_{RLS} - Time_{LAD}}{Time_{RLS}}\right)\times 100$	$\left(\frac{Time_{RLS} - Time_{CG}}{Time_{RLS}}\right)\times 100$
10	60	20.2087	11.61810	64.7237	42.5	-220.3
	80	125.3724	51.29850	221.8648	59.1	-77.0
	100	505.9440	74.45390	275.9748	85.3	45.5
	120	1772.5000	175.65270	595.0517	90.1	66.4
	140	4736.6000	294.61600	761.7571	93.8	83.9
	160	10582.2000	392.92620	987.3406	96.3	90.7
	180	23091.4000	577.65570	1449.2400	97.5	93.7
	200	40677.1000	1175.15200	1497.7800	97.1	96.3
	250	-	2290.85000	4170.2000	-	-
	300	-	3991.70000	8299.9000	-	-
	330	-	-	10684.2300	-	-
	400	-	-	50754.0000	-	-
20	60	20.1708	111.7754	344.8695	-454.1	-1609.7
	80	125.4699	349.7303	674.6040	-178.7	-437.7
	100	508.5041	646.6208	1246.8880	-27.2	-145.2
30	60	19.8749	329.9915	1370.1000	-1560.3	-6793.6
	80	125.2306	966.5548	3299.2000	-671.8	-2534.5
	100	502.2330	1534.9000	4035.0000	-205.6	-703.4

The transition matrix of this example is:

$$P = \begin{bmatrix} 0.2121 & 0.7879 & 0 \\ 0 & 0.2822 & 0.7178 \\ 1 & 0 & 0 \end{bmatrix}$$

5. CONCLUSION

The probability statistics in transition matrix of large scale Markov Chain estimated by LAD method reformulated to LP model is studied. If the size of constraint matrix of LP is checked and found that it is not over the maximum possible array, the ‘linprog’ function in MATLAB should be used to optimize the LP and if not, when the number of states *n* is a large number and the number of periods for data collection *S* is a small number, the CG algorithm is an interesting method to be used to solve that problem because it uses less than 30% of the total problem variables and the number of variables used decreases when the number of states *n* is increased.

6. ACKNOWLEDGEMENT

The researchers would like to express our sincere appreciation to comments and recommendations from anonymous referees leading to the improved final version of this manuscript.

7. REFERENCES

Ashar, V.G. and T.D. Wallace, 1963. A sampling study for minimum absolute deviation estimators. Operat. Res., 11: 747-758. DOI: 10.1287/opre.11.5.747.

Bazaraa, M.S., J.J. Jarvis and H.D. Sherali, 2010. Linear Programming and Network Flows. 4th Edn., John Wiley and Sons, Inc., ISBN-10: 0470462728, pp: 353.

Christodoulakis, A.G., 2006. Markovian credit risk transition probabilities under non-negativity constraints for US portfolio 1984-2004. J. Credit Risk. DOI: 10.2139/ssrn.881534

Constant, A.F. and K.F. Zimmermann, 2013. International Handbook on the Economics of Migration. 1st Edn., Edward Elgar Publishing, Cheltenham, ISBN-10: 1782546073, pp: 584.

Craig, B.A. and P.P. Sendi, 2002. Estimation of the transition matrix of a discrete-time Markov Chain. Health Econom., 11: 33-42. DOI: 10.1002/hec.654.

Dent, W. and R. Ballintine, 1971. A review of the estimation of transition probabilities in Markov Chain. Australian J. Agric. Econom., 15: 69-81. DOI: 10.1111/j.1467-8489.1971.tb00077.x

Goodman, L.A., 1953. A further note on miller’s finite Markov processes in psychology. Psychometrika, 18: 245-248. DOI: 10.1007/BF02289062

Jones, M.T., 2005. Estimating Markov transition matrices using proportions data: An application to credit risk. IMF Working Paper.

Kalbfleisch, J.D. and J.F. Lawless, 1984. Least-squares estimation of transition probabilities from aggre gate data. Canadian J. Statist., 12: 169-182. DOI: 10.2307/3314745

Lasdon, L.S., 1970. Optimization Theory for Large Systems. 1st Edn., Courier Dover Publications, New York, ISBN-10: 0486419991, pp: 523.

Lee, T.C., G.G. Judge and A. Zellner, 1968. Maximum likelihood and Bayesian estimation of transition probabilities. J. Am. Stat. Assoc., 63: 1162-1179. DOI: 10.1080/01621459.1968.10480918

Lee, T.C., G.G. Judge and T. Takayama, 1965. On estimating the transition probabilities of a Markov process. J. Farm Econom., 47: 742-762. DOI: 10.2307/1236285

Lin, C.T., 2011. Establishing an adaptive production-procurement system with Markov Chain approach associated with 3C theory. J. Math. Stat., 7: 187-197. DOI: 10.3844/jmssp.2011.187.197

Madansky, A., 1959. Least squares estimation in finite Markov processes. Psychometrika, 24: 137-144. DOI: 10.1007/BF02289825

MATLAB, 2013. The MathWorks, Inc., MATLAB Documentation Center.

Miller, G.A., 1952. Finite Markov processes in psychology. Psychometrika, 17: 149-167. DOI: 10.1007/BF02288779

Nash, S.G. and A. Sofer, 1996. Linear and Nonlinear Programming. 1st Edn., McGraw-Hill Companies, Inc., ISBN-10: 0070460655, pp: 204.

Simister, Z., 2007. Maximum likelihood approach to the estimation of Markov transition matrices using proportions data: An application to credit risk. Bank of Jamaica.

Telser, L.G., 1966. Least Squares Estimates of Transition Probabilities: Measurement in Economics. 1st Edn., Standford University Press, ISBN-10: 0804701369, pp: 292.

Theil, H. and G. Rey. 1966. A quadratic programming approach to the estimation of transition probabilities. Manage. Sci., 12: 714-721. DOI: 10.1287/mnsc.12.9.714

Thyagarajan, V. and S.M.B. Mohamed, 2005. Retail banking loan portfolio equilibrium mix: A Markov Chain model analysis. Am. J. Applied Sci., 2: 410-419. DOI: 10.3844/ajassp.2005.410.419