Modelling share prices via the random walk on the lamplighter group

Rukiye Samci Karadeniz[1] and Sergey Utev[2]

[1] Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH, UK
rs408@le.ac.uk
[2] Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH, UK
su35@leicester.ac.uk

Abstract. This research is a continuation of the study in [9]. It is based on the analysis of arbitrarily selected share prices with the relatively small data size (around 250 closing prices for each group). Specifically, we model data as a geometric Markov chain with a hidden random walk on a group. The hidden random walk is constructed on the lamplighter group $\mathbb{Z}_3$ and on the tensor product of groups $\mathbb{Z}_2 \otimes \mathbb{Z}_2$. The lamplighter group has a specific structure where the hidden information is actually explicit. We assume that the positions of the lamplighters are known, but we do not know the status of the lamps. It is referred to as a hidden random walk on the lamplighter group. The biased random walk (as introduced [8]) is constructed to fit the data. The missing data algorithms and Monte Carlo simulation are used to find the best fit for smallest trace norm difference of the transition matrices for tensor product of the original transition matrices from the appropriately split data. The fit is relatively good. Moreover, for the randomly chosen data sets, the α-biased random walk on the tensor product of the lamplighter group and $\alpha-\lambda$-biased random walk provide significantly better fit to the data.

Keywords. Embedding problem, lamplighter groups, random walks on groups, tensor product, missing data, machine learning.

M.S.C. classification. 62M45, 65C05, 68T05.
J.E.L. classification. C15, C45, D83.

1 Introduction

Motivated by the nature of share prices, we discuss several procedures to model risky assets via the random walk on the lamplighter group (or its tensor prod-

Mathematical Methods in Economics and Finance – m²ef
Vol. 9/10, No. 1, 2014/2015

ucts). Random walks on the wreath products (which is a specialized product of two groups based on a semi-direct product) are known in the literature as lamplighter random walks because of the intuitive interpretation of such walks in terms of the configuration of lamps (as defined in [12]). Specifically, we model data as a geometric Markov chain with a hidden random walk on group ([5]). The hidden random walk is constructed on the lamplighter group on $\mathbb{Z}_3$ and on the tensor product of groups $\mathbb{Z}_2 \otimes \mathbb{Z}_2$. The lamplighter group has a specific structure where the hidden information is actually explicit. We assume that the positions of the lamplighters are known, but we do not know the status of the lamps. We refer to it as a hidden random walk on the lamplighter group. To analyse the sensitivity of the generators, we choose at least two different generator sets (choice of an optimal generator is an open question [3]).

We also construct the biased random walks on the tensor product of the lamplighter group models (as introduced in [8]) to fit the data. Overall, several branching walk models are considered. A Monte Carlo simulation is then applied to find the best fit. The results are then compared with analytic errors computed for the relative distance between two tensor products of random stochastic matrices.

The missing data algorithms (which are considered in Section 2) and Monte Carlo simulation are used to find the best fit in the sense of finding the random walk for which the distance between the original matrix and the corresponding 3×3 reduced transition matrix is smallest. In this research, as a measure of the fit of the class of stochastic matrices, we consider the trace norm ($\sqrt{tr(A^*A)}$) between two transition matrices. The fit is relatively good. Moreover, for the randomly chosen data sets, the α-biased random walk on the tensor product of the lamplighter group and $\alpha - \lambda$-biased random walk provide significantly better fits to the data.

This research is devoted to treat stock price data as a discrete time Markov chain perturbed by the Gaussian noise. The data used in this research consists of a share price dataset from "British Petroleum (London)", commonly known as BP. The stock price data were chosen arbitrarily from the internet for BP's day-by-day closing share prices for four different financial years, 2009-2010, 2010-2011, 2011-2012, 2012-2013 (April to April). The datasets were obtained randomly from the website *http://uk.finance.yahoo.com*.

In Part 1 of Section 2, we first introduce a model as an additive functional of Markov chains perturbed by the Gaussian noise and then estimate the transition matrices for the Markov chains via the MLE method. In Part 2 of Section 2, tensor product splitting data is explained and the missing data algorithms (EM and C4.5 algorithms) are briefly introduced to treat the missing data. In Section 3, the lamplighter groups and related random walks are introduced. Finally, case studies are combined in Section 4 and some final remarks are presented in Section 5.

2 Initial Modelling and Splitting the Data

Initial modelling . Initially, the data are modelled as an additive functional of Markov chains perturbed by the Gaussian noise

$$logS_t = \log_1 + bt + \Sigma_{i=1}^{t}(M_i + \sigma\eta_i)$$

where S_t is the daily share price process, S_1 is the initial value of the share price, t is the daily unit in a financial year, b is the slope of the share price, M_i is modelled by Markov Chain, σ is the volatility of the residual, η_i are iid Gaussian random variables, and $i = \{1, 2, \cdots, t\}$.

To construct the Markov chain, first we choose three states as "stay (no jump)", "small jump", "big jump". The three-state Markov Chain is chosen to avoid overcomplicated calculations whilst still being representative of the data's behaviour. This creates the states
(i) $Z_t < \theta_1$ $\quad$ ("no jump", $0, M_i = 0$);
(ii) $\theta_1 \leq Z_t < \theta_2$ $\quad$ ("small jump", $s, M_i = 1$);
(iii) $Z_t > \theta_2$ $\quad$ ("big jump", $b, M_i = 2$).

We choose the same θ_i for all models for the comparison reasons and in addition, θ_i are aimed to maximize the largest embeddable proportion.

Hence the value of the Markov chain is defined on the transformed data for each data $Z_t, t = \{1, \ldots, n\}$ as $M_j = Z_t, j = \{1, 2, 3\}$. By abuse of notation, the Markov chain M_i will have states: "no jumps", "small jump" and "big jump".

Notice that M_i does not represent the approximate changes in Z_i. This simplified labelling is convenient and sufficient in our research.

Tensor product splitting . The term "tensor product" refers to another way of constructing a big vector space out of two (or more) smaller vector spaces. Let $A = (a_{ij})$ and $B = (b_{km})$ be the matrices. Then their tensor product is defined by $A \otimes B = (a_{ij}B)$, and in the relabelled form $(A \otimes B)_{ik,jm} = a_{ij}b_{km}$..

The tensor product structure arises from splitting the data into "no jump", "small jump" and "big jump" groups and matching into the "no small jump-small jump" and "no big jump-big jump" groups. The tensor product splitting data is a way to significantly reduce the number of parameters. In addition, from the financial perspective, the $(2 \times 2) \otimes (2 \times 2)$ allows to apply the hedging argument for each component and to encompass the incomplete data. However, the side effect of the tensor product modelling is that the construction leads to a missing data.

To explain it why, we notice that the transformed data Z is an observable data which might be considered a hidden pair (X, Y) needed to construct the tensor product structure. The key point of this structure is Z, which is the maximum of the pair $Z = \max(X, Y)$ (such as a censored data). Z represents the "no jump", "small jump" and "big jump" group which is split into two groups. Therefore, X represents the "no small jump-small jump" group and Y represents the "no big jump-big jump" group.

Let us consider a simple example to clarify the tensor product structure. Let $Z = \{s, s, s, 0, b, s, 0, 0, b, s, b, 0, s, 0\}$ be transformed data where 0 is "no jump", s is "small jump" and b is "big jump" and, the data is split as follows:

$$X = \{s, s, s, 0, ?, s, 0, 0, ?, s, ?, 0, s, 0\}\ ,\ Y = \{\hat{0}, \hat{0}, \hat{0}, \hat{0}, b, \hat{0}, \hat{0}, \hat{0}, b, \hat{0}, b, \hat{0}, \hat{0}, \hat{0}\}$$

where 0 is "no small jump", s is "small jump" and $\hat{0}$ is "no big jump", b is "big jump". Also "?" represents the missing values. Then, in general Y ("no big jump-big jump" group) is a complete dataset, however X ("no small jump-small jump" group) has missing values.

Therefore, this requires us to deal with the missing data. In the literature, the methods of dealing with the missing data are divided into the following three categories: (i) Ignoring and Discarding Data, (ii) Parameter Estimation and (iii) Imputation ([2]). In order to treat the missing values, we apply Expectation-Maximization algorithm ([4]) as the parameter estimation method and Machine Learning algorithm (C4.5) ([11]) as the imputation method.

2.1 EM algorithm

The Expectation Maximisation (EM) algorithm is a well-known iterative algorithm for parameter estimation by maximum likelihood to deal with the dataset that has missing or incomplete random variables ([4], [10]). Each iteration of the algorithm includes two steps:

- The expectation step (E-step): replacing missing parameters by estimated parameters.
- The maximization step (M-step) using the updated data from the first step to find a maximum likelihood estimation of the parameters.

The algorithm is run until the change of the estimated parameter reaches the chosen threshold.

2.2 C4.5 algorithm

C4.5 is a decision tree-based machine learning classifier, that is the C4.5 algorithm constructs classifiers expressed as decision trees built from root to leaves, [11]. As a learning algorithm, it generates a class of tests based on training examples and improves itself.

To rank possible tests, the information gain criteria (referred to as the InfoGain) is applied, more exactly we choose the test which minimises the total entropy of the resulting classified map.

3 Random Walk on the Group

In this section, we work mainly with directed graphs and have found that branching trees and graphs are particularly useful in the stochastic modelling of the data. In Subsection 3.2 we construct a branching tree or directed graph (a variant of the Caley graph) on the lamplighter group by choosing particular semi group generators. Then, we model the jump part of the shares as a random walk on the associated branching tree of the lamplighter group, which is referred to as a random walk on the lamplighter group.

As stated above, the data jumps are modelled as a geometric Markov chain with a hidden random walk on the lamplighter group on $\mathbb{Z}_3$ and on the tensor product of groups $\mathbb{Z}_2 \otimes \mathbb{Z}_2$. Let us begin with definition of the lamplighter group:

3.1 Lamplighter Group

The lamplighter group $L(G)$ on the group G is defined as a semi direct product, $L(G) := \mathbb{G} \ltimes \Sigma_{x \in G} \mathbb{Z}_2$, with the direct sum of copies of $\mathbb{Z}_2$ indexed by G; for $m, m' \in G$ and $\eta, \eta' \in \Sigma_{x \in \mathbb{Z}} \mathbb{Z}_2$ the group operation is

$$(m, \eta)(m', \eta') := (m + m', \eta \oplus \rho^{-m} \eta')$$

where $\oplus$ is component wise addition modulo 2 and ρ is left shift ([8]). The m-move is for the lamplighter and the η-move for the status of lamps (on $\to$ off and off $\to$ on). Then m is a position of the lamplighter, η is a configuration, status of lamps, (see the Figure 1 and [7] for more details).

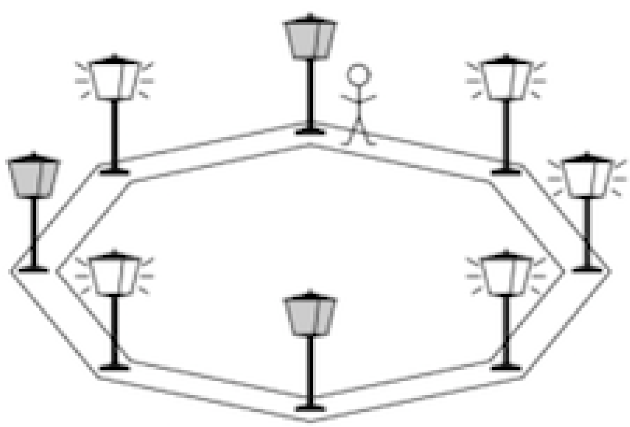

Fig. 1. Lamplighter group.

For example, take $G = Z_3$ (modulo 3), ρ is a left shift (so ρ^{-1} is a right shift).

$$\begin{aligned}(1, (1, 1, 0)) + (1, (0, 1, 0)) &= (1 + 1, (1, 1, 0) + \rho^{-1}(0, 1, 0)) \\ &= (2, (1, 1, 0) + (0, 0, 1)) = (2, (1, 1, 1)).\end{aligned}$$

The answer is interpreted as an element where position of the lamplighter is 2 and the configuration is $(1, 1, 1)$, i.e. all lamps are on.

3.2 Random Walk on the lamplighter group

To construct a random walk on the group G and apply it to fit the data, we first choose a semigroup generator set S of G, next we construct a branching tree (direct graph on G), finally we generate a random walk, referred to as a branching random walk.

In addition, using specifics of the lamplighter group, we construct a random walk on the combined states.

We begin with considering the lamplighter group on $\mathbb{Z}_3$. There are three lamplighter positions 0, 1, and 2 in the lamplighter group. They refer to differences between daily adjusted closing values of the share prices such as no jump, small jump, big jump. In addition, each lamp has two possibility "on" or "off" on the each positions which gives overall 8 states of lamps for each fixed position of the lamplighter. Therefore, the lamplighter group on $\mathbb{Z}_3$ has 24 elements which are listed in the following set:

$$
\begin{aligned}
E = \{&e1 = (0, (0, 0, 0)), e2 = (0, (0, 0, 1)), e3 = (0, (0, 1, 0)), e4 = (0, (1, 0, 0)),\\
&e5 = (0, (0, 1, 1)), e6 = (0, (1, 0, 1)), e7 = (0, (1, 1, 0)), e8 = (0, (1, 1, 1)),\\
&e9 = (1, (0, 0, 0)), e10 = (1, (0, 0, 1)), e11 = (1, (0, 1, 0)), e12 = (1, (1, 0, 0)),\\
&e13 = (1, (0, 1, 1)), e14 = (1, (1, 0, 1)), e15 = (1, (1, 1, 0)), e16 = (1, (1, 1, 1,)),\\
&e17 = (2, (0, 0, 0)), e18 = (2, (0, 0, 1)), e19 = (2, (0, 1, 0)), e20 = (2, (1, 0, 0)),\\
&e21 = (2, (0, 1, 1)), e22 = (2, (1, 0, 1)), e23 = (2, (1, 1, 0)), e24 = (2, (1, 1, 1))\}.
\end{aligned}
$$

A subset $S \subseteq G$ is a semigroup generator of G if any element $a \in G$ has a product representation of elements in S, i.e. $a = x_1 \dots x_n$ for some n, $x_i \in S$.

Let us fix a non-empty set S and an element $e \in S$. With each x we associate a set of its offsprings $\{xy : y \in S\}$. We construct the **branching tree** recursively, as a direct graph (G^*, V) by adding links and offsprings to each generation. The origin element e is referred to as 0th generation (G_0) and we take $V_0 = \emptyset$. Given generation G_k and the branching tree V_k of links, we first take $V_{k+1} = V_k \cup \{x \to xy : x \in G_k, y \in S\}$ (add new links to V_k). Then $G_{k+1} = \{xy : x \in G_k, y \in S\}$ is a set of offsprings of the previous generation. In order to avoid repetitions, we terminate links from the offspring which was born before. The overall procedure stops when there are no new elements. Notice that S is a semi group generator if and only if the branching tree will produce all the elements in group G, i.e. $G^* = G$.

The random walk on the graph (G^*, V) is then defined by

$$w_{ij} = 1/d_i \text{ if } i \text{ links to } j, w_{ij} = 0, \text{ otherwise}$$

where d_i is the number of links from element i.

In our case, $d_i \equiv 2$ and the 24×24 transition matrix W of the random walk (referred to as a simple random walk) is defined as follows:

$$w_{ij} = \begin{cases} 1/2 & \text{if} \quad i \quad \text{links to} \quad j, \\ 0 & \text{otherwise.} \end{cases}$$

The hidden Markov chain on the lamplighter group is then constructed to model the data. For the hidden part, it is assumed that we know the lamplighter positions, but we do not know the status of the lamps. So, the possible positions of lamplighter $(0, 1, 2)$ are observed as follows:

$$\begin{gathered} \mathbf{0} = \{e1 = (0,(0,0,0)), e2 = (0,(0,0,1)), e3 = (0,(0,1,0)), e4 = (0,(1,0,0)), \\ e5 = (0,(0,1,1)), e6 = (0,(1,0,1)), e7 = (0,(1,1,0)), e8 = (0,(1,1,1))\} \\ \mathbf{1} = \{e9 = (1,(0,0,0)), e10 = (1,(0,0,1)), e11 = (1,(0,1,0)), e12 = (1,(1,0,0)), \\ e13 = (1,(0,1,1)), e14 = (1,(1,0,1)), e15 = (1,(1,1,0)), e16 = (1,(1,1,1,))\} \\ \mathbf{2} = \{e17 = (2,(0,0,0)), e18 = (2,(0,0,1)), e19 = (2,(0,1,0)), e20 = (2,(1,0,0)), \\ e21 = (2,(0,1,1)), e22 = (2,(1,0,1)), e23 = (2,(1,1,0)), e24 = (2,(1,1,1))\} \end{gathered}$$

Next, we construct the branching tree by choosing one of the elements from the generator set as the origin.

Finally, we construct the branching type random walk on the branching tree treated as a direct graph.

Overall based on the original branching random walk, we construct a new random walk on combined states $0, 1, 2$. Then, for the new random walk, the simulation is run 10^5 times to find the transition matrix. The answer may be found theoretically, but it seems the random simulation is a more efficient way of finding it.

To examine the sensitivity of the generators, two different generator sets are chosen at random. First, we choose a random set of elements and verified that the set was indeed the generator (as a semi-group). If the set generates the group, the set is chosen as the generator set. Else, we choose another random set and repeat all the steps again until we find two different generator sets. Theoretically, it may appear that for two different generators the results may be qualitatively different. Choosing the "right" generator is still an open question [3].

The two randomly chosen generator sets of the lamplighter group on $\mathbb{Z}_3$ are:

$$\begin{aligned} S_1 &= \{e4 = (0,(1,0,0)), e11 = (1,(0,1,0))\}, \\ S_2 &= \{e10 = (1,(0,0,1)), e20 = (2,(1,0,0))\}. \end{aligned}$$

We compare the results for both choices in the Section 4.

Figure 2 shows the generated branching tree with the first generator set S_1, started at e_4.

In the next two parts, we construct biased random walks on the lamplighter group.

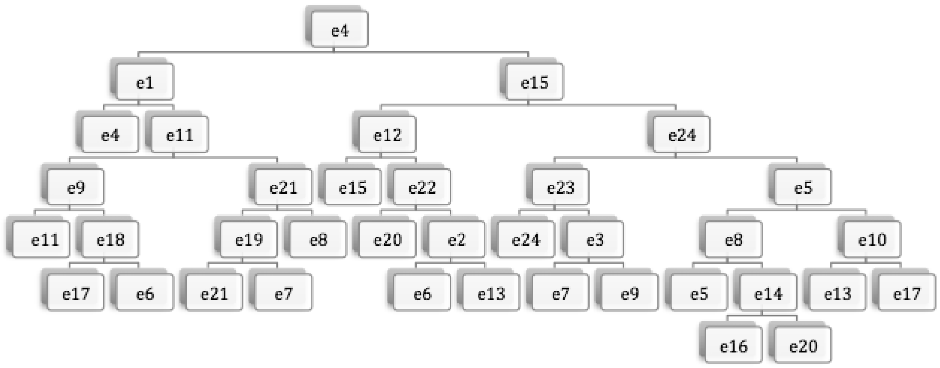

Fig. 2. The branching tree via S_1.

Biased Random Walk on the lamplighter group λ, α and $\alpha - \lambda$ biased random walks on the lamplighter group are considered in this part.

The λ- biased random walk on the lamplighter group:

Following R. Lyons, R. Pemantle, and Y. Peres (1996,[8]), for $\lambda > 0$, define the λ- biased random walk RW_λ on a connected locally finite graph with a distinguished vertex Θ as the time-homogeneous Markov chain $\{X_n; n \geq 0\}$ with the following transition probabilities. The distance from a vertex $|v|$ to Θ is the number of the edges on a shortest path joining the two vertexes. Suppose that v is a vertex of the graph. Let $v_1, ..., v_k$ ($k \geq 1$ unless $v = \Theta$) be the neighbours of v at distance $|v| - 1$ from Θ and let $u_1, u_2, ..., u_j$ $(j \geq 0)$ be the other neighbours of v. Then the transition probabilities are

$$w(v, v_i) = \frac{\lambda}{(k\lambda + j)} \quad \text{for} \quad i = 1, ..., k,$$
$$w(v, u_i) = \frac{1}{(k\lambda + j)} \quad \text{for} \quad i = 1, ..., k.$$

And,

$$w_{ij} = \begin{cases} 1/d & \text{if there are } \ d \ \text{ links where } \ d > 0, \\ 0 & \text{otherwise,} \end{cases}$$

when the λ- biased condition for the neighbours of the vertex v is satisfied ([8]).

To construct the λ- biased random walk for the data, we closely follow the construction procedure of the simple random walk. We work with the same 3×3 state Markov chain and generator sets S_1, S_2. Moreover, we only treat the case of the first generator set S_1. The results are then analysed for both generator sets (S_1, S_2) in Section 4.

Then, the λ- biased random walk is constructed on the lamplighter group via transition probabilities:

$$w(v, v_i) = \frac{\lambda}{(\lambda+1)} \ , \ w(v, u_i) = \frac{1}{(\lambda+1)} \tag{1}$$

and

$$w_{ij} = \begin{cases} 1/2 & \text{if} \quad d = 2, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

when the λ- biased condition for the neighbours is satisfied.

The transition probabilities of 24×24 Markov chain are calculated based on the Cayley table. The transition matrix P_1 consist of two part.
In the fist part all links below have transition probabilities being 1/2, i.e.
$w_{4,1} = w_{4,15} = w_{1,4} = w_{1,11} = \frac{1}{2}$, $w_{11,9} = w_{11,21} = w_{18,17} = w_{18,6} = \frac{1}{2}$;
$w_{17,18} = w_{17,4} = w_{6,2} = w_{6,15} = \frac{1}{2}$, $w_{21,19} = w_{21,8} = w_{19,21} = w_{19,7} = \frac{1}{2}$;
$w_{14,16} = w_{14,20} = w_{16,14} = w_{16,23} = \frac{1}{2}$, $w_{15,12} = w_{15,24} = w_{1,4} = w_{1,11} = \frac{1}{2}$;
$w_{12,15} = w_{12,22} = w_{20,22} = w_{20,1} = \frac{1}{2}$, $w_{2,6} = w_{2,13} = w_{13,10} = w_{13,19} = \frac{1}{2}$;
$w_{24,23} = w_{24,5} = w_{23,24} = w_{23,3} = \frac{1}{2}$, $w_{10,13} = w_{10,17} = \frac{1}{2}$.
The second part probabilities are defined by
$w_{9,11} = \frac{1}{\lambda+1}$, $w_{9,18} = \frac{\lambda}{\lambda+1}$, $w_{7,12} = \frac{1}{\lambda+1}$, $w_{7,3} = \frac{\lambda}{\lambda+1}$;
$w_{8,5} = \frac{1}{\lambda+1}$, $w_{8,14} = \frac{\lambda}{\lambda+1}$ $w_{22,2} = \frac{1}{\lambda+1}$, $w_{22,20} = \frac{\lambda}{\lambda+1}$;
$w_{3,7} = \frac{1}{\lambda+1}$, $w_{3,9} = \frac{\lambda}{\lambda+1}$, $w_{5,8} = \frac{1}{\lambda+1}$, $w_{5,10} = \frac{\lambda}{\lambda+1}$.

Then, the hidden Markov chain on the lamplighter group is constructed to model the data. The hidden part is same as before, with the known lamplighter positions but unknown states of lamps. The Monte Carlo simulation is run 10^5 times with choosing the optimal parameter λ to find the transition matrix to find the best fit for the estimated transition matrices.

The α- biased random walk on the lamplighter group:

We consider a slightly perturbed simple random walk on the lamplighter group generated as a semi group with non-symmetric set of generators. The approach is similar to the previous cases with the same set up: the same two generator sets (S_1, S_2) are chosen and the same 3×3 state Markov chain is considered as the initial matrix.

And again, we only treat the case of the first generator set S_1 with the results then being analysed for both generator sets (S_1, S_2) in Section 4.

As before, based on the Cayley table, we calculate the transition probability of Markov Chain 24×24 state. Notice that $e_1 = 0$ is not in our generators (i.e. $e_1 = 0 \notin S_i$), and so staying at the same position is not allowed in the branching-type random walk. To modify this, the α parameter is introduced such as $\alpha \in [0, 1]$ and the transition matrix is perturbed by the diagonal matrix, i.e. the 24×24 matrix for the α- biased random walk is:

$$w_{ij} = \begin{cases} \frac{1}{2}(1-\alpha) & \text{if} \quad i \quad \text{links to} \quad j, \\ \alpha & \text{if} \quad i = j, \\ 0 & \text{othewise.} \end{cases} \tag{3}$$

The other steps (e.g. hidden part, observations, ...) are similar to the previous approaches. Monte Carlo simulation is used to find the best fit with using trace norm differences for the estimated transition matrices.

The $\alpha - \lambda$- biased random walk on the lamplighter group: The process is similar to the previous cases.

The transition probabilities for the $\alpha - \lambda$- biased random walk are:

$$w(v, v_i) = (1-\alpha)\frac{\lambda}{(k\lambda + j)}) \ , \ w(v, u_i) = (1-\alpha)\frac{1}{(k\lambda + j)}) \quad \text{for} \quad i = 1, ..., k.$$

v_i and u_i are sites satisfying conditions explained in the part "Biased Random Walk on the lamplighter group".

And

$$w_{ij} = \begin{cases} (1-\alpha)\frac{1}{d} & \text{if} \quad i \quad \text{links to} \quad j, \\ \alpha & \text{if} \quad i = j, \\ 0 & \text{othewise} \end{cases}$$

when the neighbours of the vertex v are satisfied the λ- biased condition.

Specifically for the case, the 24×24 matrix W is:

$$w(v, v_i) = (1-\alpha)\frac{\lambda}{(\lambda + 1)} \ , \ w(v, u_i) = (1-\alpha)\frac{1}{(\lambda + 1)} \tag{4}$$

and

$$w_{ij} = \begin{cases} \frac{1}{2}(1-\alpha) & \text{if} \quad i \quad \text{links to} \quad j, \\ \alpha & \text{if} \quad i = j, \\ 0 & \text{othewise,} \end{cases} \tag{5}$$

when the λ- biased condition for the neighbours is satisfied.

Notice that the 24×24 transition matrix P_2 of the Markov Chain is found by

$$P_2 = (1-\alpha)P_1 + \alpha I$$

where I is the identity matrix and P_1 is the transition matrix of the λ-biased case. Finally, the Monte Carlo simulation is run 10^5 times with choosing the optimal parameters λ and $\alpha \in [0, 1]$ to compute the transition matrix to find the best fit for the original transition matrices.

3.3 Random Walk on the lamplighter group on the tensor product group

We are going to construct a random walk on the lamplighter group on the tensor product group $\mathbb{Z}_2 \otimes \mathbb{Z}_2$, for brevity referred to as a tensor product of lamplighter groups. First, let the group G be the tensor product of two groups $G = G_1 \otimes G_2$. The elements of the group G are pairs of the elements of the groups G_1 and G_2.

$$G = G_1 \otimes G_2 = (a, b), \quad a \in G_1, b \in G_2;$$
$$(a_1, b_1) \otimes (a_2, b_2) = (a_1 + a_2, b_1 + b_2).$$

Consider the lamplighter group on the group G, and particularly $G_1 = G_2 = \mathbb{Z}_2$. Notice that the elements of the lamplighter group on $\mathbb{Z}_2$:

$$E = (0,(0,0)),(0,(0,1)),(0,(1,0)),(0,(1,1)),(1,(0,0)),(1,(0,1)),(1,(1,0)),(1,(1,1))$$

Now, we introduce the elements of the lamplighter group on the tensor product group $G = \mathbb{Z}_2 \otimes \mathbb{Z}_2$. The lamplighter group has 64 elements because of the tensor product property. By relabelling there are four positions $0, 1, 2, 3$ for the lamplighter. They refer to differences between daily adjusted closing values of the share prices such as "no small jump, small jump" and "no big jump, big jump" (observed data). In addition, each lamp has two possibility "on" or "off" on each position (hidden data). Therefore, a straightforward analysis (by permutation) shows that the lamplighter group on $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ has 64 elements which are:

$$\begin{gathered}
E = \{e1 = (0,(0,0,0,0)), e2 = (0,(0,0,0,1)), e3 = (0,(0,0,1,0)), e4 = (0,(0,1,0,0)),\\
e5 = (0,(1,0,0,0)), e6 = (0,(0,0,1,1)), e7 = (0,(0,1,1,0)), e8 = (0,(1,1,0,0)),\\
e9 = (0,(0,1,0,1)), e10 = (0,(1,0,0,1)), e11 = (0,(1,0,1,0)), e12 = (0,(0,1,1,1)),\\
e13 = (0,(1,0,1,1)), e14 = (0,(1,1,0,1)), e15 = (0,(1,1,1,0)), e16 = (0,(1,1,1,1)),\\
e17 = (1,(0,0,0,0)), e18 = (1,(0,0,0,1)), e19 = (1,(0,0,1,0)), e20 = (1,(0,1,0,0)),\\
e21 = (1,(1,0,0,0)), e22 = (1,(0,0,1,1)), e23 = (1,(0,1,1,0)), e24 = (1,(1,1,0,0)),\\
e25 = (1,(0,1,0,1)), e26 = (1,(1,0,0,1)), e27 = (1,(1,0,1,0)), e28 = (1,(0,1,1,1)),\\
e29 = (1,(1,0,1,1)), e30 = (1,(1,1,0,1)), e31 = (1,(1,1,1,0)), e32 = (1,(1,1,1,1)),\\
e33 = (2,(0,0,0,0)), e34 = (2,(0,0,0,1)), e35 = (2,(0,0,1,0)), e36 = (2,(0,1,0,0)),\\
e37 = (2,(1,0,0,0)), e38 = (2,(0,0,1,1)), e39 = (2,(0,1,1,0)), e40 = (2,(1,1,0,0)),\\
e41 = (2,(0,1,0,1)), e42 = (2,(1,0,0,1)), e43 = (2,(1,0,1,0)), e44 = (2,(0,1,1,1)),\\
e45 = (2,(1,0,1,1)), e46 = (2,(1,1,0,1)), e47 = (2,(1,1,1,0)), e48 = (2,(1,1,1,1)),\\
e49 = (3,(0,0,0,0)), e50 = (3,(0,0,0,1)), e51 = (3,(0,0,1,0)), e52 = (3,(0,1,0,0)),\\
e53 = (3,(1,0,0,0)), e54 = (3,(0,0,1,1)), e55 = (3,(0,1,1,0)), e56 = (3,(1,1,0,0)),\\
e57 = (3,(0,1,0,1)), e58 = (3,(1,0,0,1)), e59 = (3,(1,0,1,0)), e60 = (3,(0,1,1,1)),\\
e61 = (3,(1,0,1,1)), e62 = (3,(1,1,0,1)), e63 = (3,(1,1,1,0)), e64 = (0,(1,1,1,1))\}.
\end{gathered}$$

And now, same as in the first model (see Section 3.2) we randomly choose two different generator sets of the tensor product of the lamplighter group $G_1 \otimes G_2$, generating it a semi group. The generator sets of the group are chosen as:

$$\begin{gathered}
S_3 = \{e18 = (1,(0,0,0,1)), e35 = (2,(0,0,1,0))\},\\
S_4 = \{e36 = (2,(0,1,0,0)), e50 = (3,(0,0,0,1))\}.
\end{gathered}$$

Then, we construct the simple random walk and the biased random walks via the tensor product of the lamplighter groups. We also consider the hidden Markov chain on the group to model the data. For the hidden part, it is assumed that we know the lamplighter positions, but we do not know the status of the lamps. So, the possible positions of lamplighter $(0, 1, 2, 3)$. Therefore, the 64×64 transition matrices is reduced to 4×4 transition matrices. Finally, we estimate the transition matrix for the new model. The overall procedure is then similar to the case of the lamplighter group on $\mathbb{Z}_3$, where necessary definitions and explanations can be found.

Let us start by constructing the simple random walk on the group. Transition matrix of the simple random walk:

$$w_{ij} = \begin{cases} 1/2 & \text{if } i \text{ links to } j, \\ 0 & \text{otherwise.} \end{cases}$$

The hidden part and observations are similar to the previous model. Figure 3 shows the generated branching tree with the first generator set (S_3).

We run the simulation 10^5 times to find the best fit to the original matrix. The data is split as in Section 2. The transition matrices are estimated by the MLE and their tensor product is used further in the tensor product modelling.

Biased Random Walk on the tensor product of the lamplighter group
In this part, we construct the biased random walk via tensor product of the lamplighter group by choosing optimal parameters. Let us start with the λ-biased random walk:

The λ- biased random walk on the tensor product of the lamplighter group: We consider a slightly perturbed λ- biased random walk on a lamplighter group ([8]).

Then, the λ- biased random walk is constructed on the lamplighter group via transition probabilities as defined in (1)-(2).

The transition probability of Markov Chain 64×64 state is calculated based on the Cayley graph. Particularly, the transition probabilities as below:

$w_{1,18} = w_{1,35} = w_{2,17} = w_{2,38} = w_{3,22} = w_{3,33} = w_{5,26} = w_{5,43} = \frac{1}{2}$,
$w_{6,19} = w_{6,34} = w_{7,28} = w_{7,36} = w_{8,30} = w_{8,47} = w_{9,20} = w_{9,44} = \frac{1}{2}$,
$w_{10,21} = w_{10,45} = w_{11,29} = w_{11,37} = w_{12,23} = w_{12,41} = w_{13,27} = w_{13,42} = \frac{1}{2}$,
$w_{14,24} = w_{14,48} = w_{15,32} = w_{15,40} = w_{16,31} = w_{16,46} = w_{17,37} = w_{17,50} = \frac{1}{2}$,
$w_{18,42} = w_{18,49} = w_{19,43} = w_{19,54} = w_{21,33} = w_{21,58} = w_{23,47} = w_{23,60} = \frac{1}{2}$,
$w_{24,36} = w_{24,62} = w_{25,46} = w_{25,52} = w_{27,35} = w_{27,61} = w_{28,48} = w_{28,55} = \frac{1}{2}$,
$w_{29,38} = w_{29,59} = w_{30,41} = w_{30,56} = w_{32,44} = w_{32,63} = w_{33,52} = w_{33,5} = \frac{1}{2}$,
$w_{34,57} = w_{34,10} = w_{35,55} = w_{35,11} = w_{36,49} = w_{36,8} = w_{37,56} = w_{37,1} = \frac{1}{2}$,
$w_{38,60} = w_{38,13} = w_{39,51} = w_{39,15} = w_{40,53} = w_{40,4} = w_{42,62} = w_{42,2} = \frac{1}{2}$,
$w_{43,63} = w_{43,3} = w_{44,54} = w_{44,16} = w_{45,64} = w_{45,6} = w_{47,59} = w_{47,7} = \frac{1}{2}$,
$w_{48,61} = w_{48,12} = w_{49,3} = w_{49,20} = w_{51,1} = w_{51,23} = w_{52,7} = w_{52,17} = \frac{1}{2}$,
$w_{53,11} = w_{53,24} = w_{54,2} = w_{54,28} = w_{55,4} = w_{55,19} = w_{56,15} = w_{56,21} = \frac{1}{2}$,
$w_{57,12} = w_{57,18} = w_{58,13} = w_{58,30} = w_{59,5} = w_{59,31} = w_{60,9} = w_{60,22} = \frac{1}{2}$,
$w_{61,10} = w_{61,32} = w_{62,16} = w_{62,26} = w_{63,8} = w_{63,27} = w_{64,14} = w_{64,29} = \frac{1}{2}$,
$w_{31,64} = \frac{1}{\lambda+1}, \quad w_{31,39} = \frac{\lambda}{\lambda+1}, \; w_{46,58} = \frac{1}{\lambda+1}, \quad w_{46,9} = \frac{\lambda}{\lambda+1}$,
$w_{26,53} = \frac{1}{\lambda+1}, \quad w_{26,24} = \frac{\lambda}{\lambda+1}, \; w_{50,25} = \frac{1}{\lambda+1}, \quad w_{50,6} = \frac{\lambda}{\lambda+1}$,
$w_{22,45} = \frac{1}{\lambda+1}, \quad w_{22,51} = \frac{\lambda}{\lambda+1}, \; w_{20,40} = \frac{1}{\lambda+1}, \quad w_{20,57} = \frac{\lambda}{\lambda+1}$,
$w_{4,25} = \frac{1}{\lambda+1}, \quad w_{4,39} = \frac{\lambda}{\lambda+1}, \; w_{41,14} = \frac{1}{\lambda+1}, \quad w_{41,50} = \frac{\lambda}{\lambda+1}$.

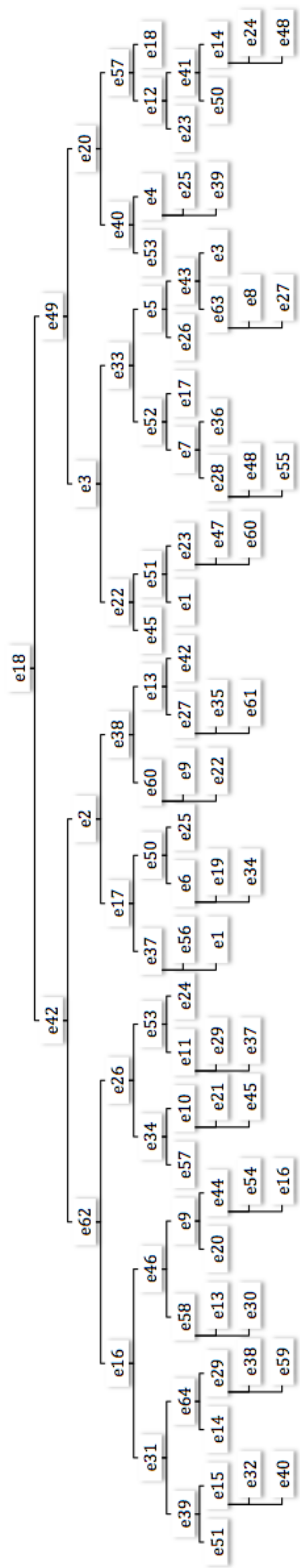

Fig. 3. The branching tree via S_3.

We use similar process to the simple random walk on the tensor product of the lamplighter group. Finally the simulation is run 10^5 times, and the optimal parameter λ is found to give the best fit to the original matrix.

The α- biased random walk on the tensor product of the lamplighter group: The 64×64 transition matrix is defined as in (3).

The other steps (e.g. hidden part, observations,...) are similar to the previous approaches. Monte Carlo simulation used to find the best fit with using norm differences for the estimated transition matrices.

The $\alpha - \lambda$- biased random walk on the tensor product of the lamplighter group: The transition probabilities for the $\alpha - \lambda$- biased random walk on the lamplighter group are defined similar to the first model, more exactly as in (4) -(5).

The Monte Carlo simulation is run 10^5 times to choose the optimal parameter λ and $\alpha \in [0, 1]$ to find the best fit for the estimated transition matrices.

4 Results and Comparisons

Modelling without the tensor product structure. Branching type random walk is constructed on the lamplighter group with two different generator sets (S_1, S_2) in Section 3.2. Also, biased random walk is considered on the lamplighter group. 3×3 transition matrices estimated by the Monte Carlo simulation. We estimate the transition matrices by constructing the model as the simple random walk and biased random walks on the lamplighter group to find the best fit for the estimated transition matrices. Table 1 shows the estimated transition matrices by Maximum likelihood. Then Tables 2-5 illustrate transition matrices for the branching type random walk on the lamplighter group.

Table 1. Three-by-three estimated transition matrices by MLE.

Cases	P
BP(2009-2010)	$\begin{pmatrix} 0.2051 & 0.7436 & 0.0513 \\ 0.1429 & 0.7551 & 0.1020 \\ 0.1667 & 0.7500 & 0.0833 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.4381 & 0.4762 & 0.0857 \\ 0.4310 & 0.4310 & 0.1379 \\ 0.2500 & 0.4444 & 0.3056 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.0667 & 0.8667 & 0.0667 \\ 0.1140 & 0.7668 & 0.1192 \\ 0.1667 & 0.6667 & 0.1667 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.2162 & 0.6757 & 0.1081 \\ 0.1327 & 0.7704 & 0.0969 \\ 0.1200 & 0.8000 & 0.0800 \end{pmatrix}$

Table 2. Branching-type random walk on the lamplighter group.

Cases	Generator S_1	Generator S_2
BP(2009-2010)	$\begin{pmatrix} 0.4845 & 0.5155 & 0 \\ 0 & 0.4908 & 0.5092 \\ 0.4716 & 0 & 0.5284 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5271 & 0.4729 \\ 0.5638 & 0 & 0.4362 \\ 0.5128 & 0.4872 & 0 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.4702 & 0.5298 & 0 \\ 0 & 0.4847 & 0.5153 \\ 0.4732 & 0 & 0.5268 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.4738 & 0.5262 \\ 0.5385 & 0 & 0.4615 \\ 0.5106 & 0.4894 & 0 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.4947 & 0.5093 & 0 \\ 0 & 0.5015 & 0.4985 \\ 0.4670 & 0 & 0.5330 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5233 & 0.4767 \\ 0.5444 & 0 & 0.4556 \\ 0.5031 & 0.4969 & 0 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.4654 & 0.4346 & 0 \\ 0 & 0.5157 & 0.4843 \\ 0.5136 & 0 & 0.4864 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5701 & 0.4299 \\ 0.4766 & 0 & 0.5234 \\ 0.5294 & 0.4706 & 0 \end{pmatrix}$

Table 3. λ- biased random walk on the lamplighter group.

Cases	Generator S_1	Generator S_2
BP(2009-2010)	$\begin{pmatrix} 0.4088 & 0.5912 & 0 \\ 0 & 0.5955 & 0.4045 \\ 0.4909 & 0 & 0.5091 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5836 & 0.4164 \\ 0.3623 & 0.1836 & 0.4541 \\ 0.4295 & 0.5705 & 0 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.4718 & 0.5282 & 0 \\ 0 & 0.6084 & 0.3916 \\ 0.4474 & 0 & 0.5526 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5604 & 0.4396 \\ 0.4144 & 0.1663 & 0.4194 \\ 0.4348 & 0.5652 & 0 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.3723 & 0.6277 & 0 \\ 0 & 0.5438 & 0.4562 \\ 0.4928 & 0 & 0.5072 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5749 & 0.4251 \\ 0.3687 & 0.2048 & 0.4265 \\ 0.4463 & 0.5537 & 0 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.4418 & 0.5582 & 0 \\ 0 & 0.5888 & 0.4112 \\ 0.5159 & 0 & 0.4841 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5019 & 0.4981 \\ 0.3131 & 0.2015 & 0.4854 \\ 0.3921 & 0.6079 & 0 \end{pmatrix}$

Table 4. α- biased random walk on the lamplighter group.

Cases	Generator S_1	Generator S_2
BP(2009-2010)	$\begin{pmatrix} 0.6860 & 0.3140 & 0 \\ 0 & 0.7927 & 0.2073 \\ 0.2247 & 0 & 0.7753 \end{pmatrix}$	$\begin{pmatrix} 0.6167 & 0.3833 & 0 \\ 0.3174 & 0.6807 & 0.0018 \\ 0 & 1.0000 & 0 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.6070 & 0.3930 & 0 \\ 0 & 0.6990 & 0.3010 \\ 0.1886 & 0 & 0.8114 \end{pmatrix}$	$\begin{pmatrix} 0.5482 & 0.2193 & 0.2326 \\ 0.2295 & 0.5184 & 0.2521 \\ 0.1561 & 0.3035 & 0.5405 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.6808 & 0.3192 & 0 \\ 0 & 0.7629 & 0.2371 \\ 0.2103 & 0 & 0.7897 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.4000 & 0.6000 \\ 0.0061 & 0.3347 & 0.6592 \\ 0.0020 & 0.6514 & 0.3466 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.2857 & 0.7143 & 0 \\ 0 & 0.8000 & 0.2000 \\ 0.0041 & 0 & 0.9959 \end{pmatrix}$	$\begin{pmatrix} 0.3438 & 0.2188 & 0.4375 \\ 0.0181 & 0.5221 & 0.4598 \\ 0.0234 & 0.4936 & 0.4830 \end{pmatrix}$

Table 5. $\alpha - \lambda$- biased random walk on the lamplighter group.

Cases	Generator S_1	Generator S_2
BP(2009-2010)	$\begin{pmatrix} 0.6478 & 0.3522 & 0 \\ 0 & 0.7650 & 0.2350 \\ 0.2222 & 0 & 0.7778 \end{pmatrix}$	$\begin{pmatrix} 0.2500 & 0.7500 & 0 \\ 0.0020 & 0.9970 & 0.0010 \\ 0 & 1.0000 & 0 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.5000 & 0.5000 & 0 \\ 0 & 0.8571 & 0.1429 \\ 0 & 0 & 1.0000 \end{pmatrix}$	$\begin{pmatrix} 0.3273 & 0.3165 & 0.3561 \\ 0.2414 & 0.4409 & 0.3177 \\ 0.2785 & 0.4399 & 0.2816 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.5584 & 0.4416 & 0 \\ 0 & 0.7258 & 0.2742 \\ 0.2544 & 0 & 0.7456 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.6667 & 0.3333 \\ 0.0010 & 0.9970 & 0.0020 \\ 0.3333 & 0.6667 & 0 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.4000 & 0.6000 & 0 \\ 0 & 0.9932 & 0.0068 \\ 0.0036 & 0 & 0.9964 \end{pmatrix}$	$\begin{pmatrix} 0.2500 & 0.5000 & 0.2500 \\ 0.0010 & 0.9950 & 0.0040 \\ 0.2000 & 0.8000 & 0 \end{pmatrix}$

Motivated by [9], to avoid a random fit and to compare the four different methods with the four different random walks on the lamplighter group and two different generator sets, we calculate the trace error norms between the simulated matrices and the one based on the data. First part of the Table 6 shows the comparison of trace norm values for all of the cases for each of the four methods with the first generator S_1. And, comparison of trace norm values for all of the methods with second generator S_2 are stated in the second part of the table. It shows that the best approximation was given by the α- biased and $\alpha - \lambda$- biased random walks. The smallest norm value is around 0.025 (BP, 2012-2013). Also, there is no significant difference between the values for the two different generator sets.

Table 6. Norm errors of the random walk on the lamplighter group.

Cases	Simple RW	λ biased RW	α biased RW	$\alpha - \lambda$ biased RW
BP(2009-2010)	0.9715	0.8422	0.3077	0.0997
BP(2010-2011)	0.9684	0.8561	0.2952	0.0530
BP(2011-2012)	0.9664	0.8330	0.3140	0.0731
BP(2012-2013)	0.9781	0.9630	0.3136	0.0742
BP(2009-2010)	0.9832	0.8996	0.2224	0.0455
BP(2010-2011)	0.9076	0.8616	0.2649	0.0417
BP(2011-2012)	0.9868	0.9003	0.3426	0.0669
BP(2012-2013)	0.9855	0.9080	0.2992	0.0253

Adding a tensor product structure. Additionally, we assume that the transformed data are such as $Z = (X, Y)$ where X represents a "no jump", "small jump"

group variable and Y - "no big jump", "big jump" groups. P_X - estimated transition matrix of X and P_Y - estimated transition matrix of Y. In order to estimate the transition matrix P_X, we deal with the missing data by applying two different methods: EM algorithm and C4.5 algorithm. Altogether, we estimate two different transition matrices for each case and then take their tensor products which are illustrated by Table 7 and Table 8. Moreover, branching type random walk is constructed on the tensor product of the lamplighter group with two different generator sets (S_3, S_4) in Section 3.3. Additionally, biased random walk is considered on the tensor product lamplighter group. Their transition matrices and the estimated transition matrices from Section 3.2 are compared with tensor product of the original transition matrices $(P_X \otimes P_Y)$. The trace norm is applied to find the best fit to the data. To give an idea of the results of this methods, Table 9 shows the transition matrices for λ- biased random walk on tensor product of the lamplighter group with the generator S_3.
Finally, Table 10 shows the comparison of the norm errors of the random walk on the tensor product of lamplighter group with two different generator sets (S_3, S_4) with the transition matrices are estimated via EM algorithm and Machine learning (C4.5 algorithm). The best approximation is again achieved by the α-biased and $\alpha - \lambda$- biased random walks. The smallest norm value is around 0.01 (Table 10, BP 2011-2012) . Also, there is no significant difference between the values for the two different generator sets and the two different missing value treatment methods.

Table 7. Transition matrices via EM algorithm.

Cases	P_X	P_Y	$P_X \otimes P_Y$
BP(2009-2010)	$\begin{pmatrix} 0.2419 & 0.7581 \\ 0.2449 & 0.7551 \end{pmatrix}$	$\begin{pmatrix} 0.9060 & 0.0940 \\ 0.9167 & 0.0833 \end{pmatrix}$	$\begin{pmatrix} 0.2192 & 0.0227 & 0.6868 & 0.0713 \\ 0.2218 & 0.0202 & 0.6949 & 0.0632 \\ 0.2219 & 0.0230 & 0.6841 & 0.0710 \\ 0.2245 & 0.0204 & 0.6922 & 0.0629 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.5319 & 0.4681 \\ 0.5652 & 0.4348 \end{pmatrix}$	$\begin{pmatrix} 0.8864 & 0.1136 \\ 0.6944 & 0.3056 \end{pmatrix}$	$\begin{pmatrix} 0.4715 & 0.0604 & 0.4149 & 0.0532 \\ 0.3694 & 0.1625 & 0.3251 & 0.1430 \\ 0.5010 & 0.0642 & 0.3854 & 0.0494 \\ 0.3925 & 0.1727 & 0.3019 & 0.1329 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.2333 & 0.7667 \\ 0.2344 & 0.7656 \end{pmatrix}$	$\begin{pmatrix} 0.7293 & 0.2707 \\ 0.6761 & 0.3239 \end{pmatrix}$	$\begin{pmatrix} 0.3298 & 0.1224 & 0.3995 & 0.1483 \\ 0.3057 & 0.1465 & 0.3704 & 0.1775 \\ 0.3354 & 0.1245 & 0.3939 & 0.1462 \\ 0.3109 & 0.1490 & 0.3652 & 0.1750 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.2787 & 0.7213 \\ 0.2296 & 0.7704 \end{pmatrix}$	$\begin{pmatrix} 0.9009 & 0.0991 \\ 0.9200 & 0.0800 \end{pmatrix}$	$\begin{pmatrix} 0.2511 & 0.0276 & 0.6498 & 0.0715 \\ 0.2564 & 0.0223 & 0.6636 & 0.0577 \\ 0.2068 & 0.0228 & 0.6940 & 0.0764 \\ 0.2112 & 0.0184 & 0.7088 & 0.0616 \end{pmatrix}$

Table 8. Transition matrices via Machine Learning.

Cases	P_X	P_Y	$P_X \otimes P_Y$
BP(2009-2010)	$\begin{pmatrix} 0.2072 & 0.7928 \\ 0.3729 & 0.6271 \end{pmatrix}$	$\begin{pmatrix} 0.9060 & 0.0940 \\ 0.9167 & 0.0833 \end{pmatrix}$	$\begin{pmatrix} 0.1877 & 0.0195 & 0.7183 & 0.745 \\ 0.1899 & 0.0173 & 0.7268 & 0.0660 \\ 0.3378 & 0.351 & 0.5682 & 0.0589 \\ 0.3418 & 0.0311 & 0.5749 & 0.0522 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0.3489 & 0.6511 \\ 0.6504 & 0.3496 \end{pmatrix}$	$\begin{pmatrix} 0.8864 & 0.1136 \\ 0.6944 & 0.3056 \end{pmatrix}$	$\begin{pmatrix} 0.3093 & 0.0396 & 0.5771 & 0.0740 \\ 0.2423 & 0.1066 & 0.4521 & 0.1990 \\ 0.5765 & 0.0739 & 0.3099 & 0.0397 \\ 0.4516 & 0.1988 & 0.2428 & 0.1068 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0.0463 & 0.9537 \\ 0.3221 & 0.6779 \end{pmatrix}$	$\begin{pmatrix} 0.7293 & 0.2707 \\ 0.6761 & 0.3239 \end{pmatrix}$	$\begin{pmatrix} 0.0338 & 0.0125 & 0.6955 & 0.2582 \\ 0.0313 & 0.0150 & 0.6448 & 0.3089 \\ 0.2349 & 0.0872 & 0.4944 & 0.1835 \\ 0.2178 & 0.1043 & 0.4583 & 0.2196 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0.1332 & 0.8668 \\ 0.3146 & 0.6854 \end{pmatrix}$	$\begin{pmatrix} 0.9009 & 0.0991 \\ 0.9200 & 0.0800 \end{pmatrix}$	$\begin{pmatrix} 0.1200 & 0.0132 & 0.7809 & 0.0859 \\ 0.1225 & 0.0107 & 0.7975 & 0.0693 \\ 0.2834 & 0.0312 & 0.6175 & 0.0679 \\ 0.2894 & 0.0252 & 0.6306 & 0.0548 \end{pmatrix}$

Table 9. λ biased random walk on the tensor product of the lamplighter group with S_3.

Cases	$\hat{P}$
BP(2009-2010)	$\begin{pmatrix} 0 & 0.4323 & 0.5677 & 0 \\ 0 & 0 & 0.5973 & 0.4027 \\ 0.5018 & 0 & 0 & 0.4982 \\ 0.5391 & 0.4609 & 0 & 0 \end{pmatrix}$
BP(2010-2011)	$\begin{pmatrix} 0 & 0.4568 & 0.5432 & 0 \\ 0 & 0 & 0.5261 & 0.4739 \\ 0.5919 & 0 & 0 & 0.4081 \\ 0.5273 & 0.4727 & 0 & 0 \end{pmatrix}$
BP(2011-2012)	$\begin{pmatrix} 0 & 0.3827 & 0.6173 & 0 \\ 0 & 0 & 0.5577 & 0.4423 \\ 0.5261 & 0 & 0 & 0.4739 \\ 0.5526 & 0.4474 & 0 & 0 \end{pmatrix}$
BP(2012-2013)	$\begin{pmatrix} 0 & 0.4167 & 0.5833 & 0 \\ 0 & 0 & 0.5804 & 0.4196 \\ 0.5246 & 0 & 0 & 0.4754 \\ 0.5000 & 0.5000 & 0 & 0 \end{pmatrix}$

5 Some Final Remarks

The fit is relatively good. For the randomly chosen data sets, the α-biased random walk on the lamplighter group and $\alpha - \lambda$- biased random walk provide significantly better fit to the data. The smallest trace norm values is around 0.01. Also, the α-biased random walk on the tensor product of the lamplighter group and $\alpha - \lambda$- biased random walk provide significantly better fit to the data comparing with other models.

Table 10. Norm errors of the random walk on the tensor product of the lamplighter group

Cases	Simple RW	λ biased RW	α biased RW	$\alpha - \lambda$ biased RW
BP(2009-2010)	1.2659	0.9677	0.3767	0.1797
BP(2010-2011)	1.1207	0.8234	0.2334	0.0290
BP(2011-2012)	1.1675	0.8690	0.2934	0.0123
BP(2012-2013)	1.2215	0.9214	0.3400	0.1089
BP(2009-2010)	1.2201	0.9123	0.3001	0.1569
BP(2010-2011)	1.1814	0.8692	0.2771	0.0783
BP(2011-2012)	1.2062	0.8966	0.3166	0.0995
BP(2012-2013)	1.2662	0.9528	0.3531	0.1534
BP(2009-2010)	1.2400	0.9460	0.3495	0.1261
BP(2010-2011)	1.1738	0.8759	0.2082	0.1334
BP(2011-2012)	1.1611	0.8612	0.2050	0.1399
BP(2012-2013)	1.2504	0.9528	0.3347	0.1142
BP(2009-2010)	1.2575	0.9457	0.3138	0.1141
BP(2010-2011)	1.1882	0.8790	0.2684	0.1016
BP(2011-2012)	1.1674	0.8626	0.2424	0.1283
BP(2012-2013)	1.2553	0.9407	0.3201	0.1176

The random walk on the tensor product of the lamplighter group gives better approximation than the random walk on the lamplighter group. Two different generators are chosen arbitrarily for the each case and the results for different generators are similar which shows a mild sensitivity on the generator. Two different methods (EM and Machine Learning) are applied to deal with the missing data and again, the results look similar showing the robustness of the overall method.

Note that our missing data comes from the choice of the tensor product model, which can be seen as a side-effect. However, by adding the tensor product structure, we greatly simplify the number of parameters that need to be estimated. For example, the transition matrix in the lamplighter group on $\mathbb{Z}_4$ is identified by 64×64 parameters, but on $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ it requires at most 4×4 parameters.

References

1. Baaquie, B. E.: The theoretical foundations of quantum mechanics. Springer Science and Business Media, (2013)
2. Batista, P.A., Gustavo, E.A. and Monard, M.C.: An analysis four missing data treatment methods for supervised learning. Applied Artificial Intelligence, **17** (5-6): 519-533, (2003)
3. Chung, F.K., Graham, R.L. : Random walks on generating sets for finite groups. Electronic Journals of Combinatorics, **4**, 14-28, (1997)
4. Dempster, A.P., Laird, N.M. and Rubin, D.B.: Maximum Likelihood for Incomplete Data via the EM algorithm. Journal of the Royal Statistical Society,Series B, **39**, 1-38, (1977)
5. Guedon, Y.: Estimating hidden semi-Markov chains from discrete sequences. Journal of computational and graphical statistics, **12**, 604-639, (2003)
6. Israel, R., Rosenthal, J.S. and Wei, J.Z.: Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings. Mathematical Finance, **11**, 245-65, (1999)
7. Levin, D.A., Peres, Y. and Wilmer E.L.: Markov chains and mixing times. books.google.com (2009)
8. Lyons, R., Pemantle, R. and Peres, Y.: Random walks on the lamplighter group. The annals of probability, **24**, 1993-2006, (1996)
9. Ma, X., Utev, S.: Modelling the share prices as a hidden random walk on the lamplighter group. Mathematical and Statistical Methods for Actuarial, Springer, (2012)
10. Orchard, T., Woodbury, M. A.: A missing information principle: theory and applications. Sixth Berkeley Symp. on Math. Statist. and Prob., Univ. of California Press, **1**, 697-715, (1972)
11. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann CA.,(1988)
12. Varopoulos, N.: Random walks on solvable groups. Bull. Sci. math., **107**, 337-44, (1983)