OD Estimation from Traffic Count using Gibbs Sampler and Parallel Computing

Saumya Bhatnagar

16338296

Abstract: The Origin-Destination Estimation (abbreviated ODE) is a challenging endeavor in transportation forecasting model. The ODE model reflects the total travel volume flowing through routes at any given time. For traffic planning, the ODE modeling is used in organizations such as traffic planning, urban & regional planning and tourism. In this report, OD matrix is assumed and created using a Gibbs sampler and Kalman Filter. To avoid the extensive sampling, parallel computing is implemented increasing the efficiency of the process.

I. Introduction

Traveling is one of the unquestionable elements of life, and it advances with globalization, increase in population, technology and many more. For any trip planning, understanding of its both ends, is important. Origin and destination - the locations where a trip starts and ends respectively, constitute an O-D matrix. The matrix provides a detailed picture of the trip patterns and travel choices. These matrices are categorized as static O-D matrices and time-dependent O-D matrices. While static O-D matrices are worthy in case of stagnant data, in this world of machine learning, artificial intelligence and parallel computing, owing quick access to huge data, a time dependent OD matrix can be computed easily, resulting in real time data.

This report considers a time dependent origin-destination (O-D) estimation involving the state space model, Gibbs sampler, and Kalman Filter.

Gibbs sampler is used for simultaneous sampling of state vector and transition matrix, along with Kalman filter. The whole network is divided into multiple sub-networks and computed in parallel.

The next sections of the report include, starting with background study in section II, followed by Section III giving the outline of approach used, and Section IV implementing the parallel algorithm approach using parallel chain convergence, to the results and references in Section V and VI respectively.

II. Background

Since our O-D matrix is Euclidean in nature, analogy can be drawn between the state variables and vectors.

The O-D matrix can be pictured as a State Space Model in Euclidean space. State space model lets the inputs and outputs of a model in a system of difference equations. The values of the state variable in the state space system evolve over time. A worth noticing feature of state variables x(t) is that they are reconstructed from the measured input-output data, instead of getting measured during an experiment. The state-space model structure is a good choice for quick estimation because it requires only one input, the model order, n. The model order is an integer equal to the dimension of x(t) and relates to, but is not necessarily equal to, the number of delayed inputs and outputs used in the corresponding linear difference equation.

In one approach, data was required for any analysis. This prior data came from extensive survey. Though survey data is true data, but it is time consuming and doesn't give real-time results. Hence, it is not suitable for traffic operation problems, and other real-time applications.

Yet another approach involves trip distribution models, such as gravity model, Tobit model, dynamic traffic assignment(DTA). The gravity model fails due to its aggregate nature and sequent assumption that the interaction between two locations declines with increasing (distance, time, and cost) between them. The Tobit model fails as the model requires dependent variable to be regressed, skewed to one direction, which is unlikely to fetch realistic results when assumed in case of traffic problem formulations. The DTA model is static in nature for traffic problems.

The third approach uses non-assignment models. It assumes that O-D parameters remain constant during consecutive intervals of estimations. As there is huge amount of link flow, they are computationally intensive for large scale network. So, they are abstract unless a parallel network is applied in these models.

Introduction of state space model in time dependent O-D estimation using space vector as unknown O-D flows, was first introduced by Okutani (1987). The usage of state space vectors as OD links is used by Ashok and Ben-Akiva (1993). In 2002, Jou used the same approach without considering any prior information of state vectors and transition matrices. In this report, the same is used with parallel computation.

Kalman Filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. A direct analogy can be seen between Kalman Filter and the link flow counts in traffic conditions. Gibbs sampler, a type of Monte Carlo Markov Chain sampling method, is used to generate posterior samples, by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values. For sampling of transition matrix from state vector Gibbs sampler is used. The Gibbs sampling scheme on a random transition matrix and state vector forms the center part of the algorithm. In the process of generating state vectors, Kalman filtering mechanism is added.

The least square method is disregarded as it is used to solve overdetermined systems, which is most likely to fail in traffic system.

III. Solution Approach

1. <u>Designing of state space model</u> (without considering travel time): composed of two equations – transition equations and observation equations

Transition equation: path flows at time t can be related to the path flows at time t–1 by the following autoregressive form

$x_t = Fx_{t-1} + u_t$; (t = 1, 2, 3, … n) … (i)

Observation equation: observed vector y with respect to state vector x is given by

$y_t = Hx_t + v_t$; (t = 1, 2, 3, … n) … (ii)

2. Introduction of travel time: When considering the travel time, the observation equation is modified

Transition Equation:

$x_t = Fx_{t-1} + u_t$, (t = 1, 2, 3, … n) … same as (i)

Observation equation: with introduced time lag

$y_t = Hx_t^{lag} + v_t$, (t = 1, 2, 3, … n) … (iii)

where,

$x_t^{lag} = \{ x_t^T x_{t-1}^T x_{t-2}^T \cdots\cdots\cdots x_{t-Maxlag}^T \}$ … (iv)

Here,
x_t = a p×1 network path flows on time t
y_t = a q×1 link traffic observation vector on time t
F = a p×p path flow transition matrix
u_t, v_t = iid Gaussian noise terms
H = a q×p zero-one matrix which denotes the path-observation incidence matrix
x_t^{lag} = p (MaxLag+1) × 1 matrix composed of series of state vectors (takes the travel time into account)
MaxLag denotes the maximum time lag intervals that can be observed on observation sites at time t

3. Kalman Filter:
The structure of the filter can be derived in a Bayesian framework as follows:

a. At the first stage (i.e. t=1), there's no observation, thus the state vector x_0 must be generated by a prior distribution such that the distribution for the state vector in the first stage will be normal with parameters,
$E[x_t | y_{t-1}] = \mu_{t-1} = F \mu_{t-1}$
$Var[x_t | y_{t-1}] = V_{t|t-1} = F V_{t-1} F^T + \Sigma$
$x_0 \sim N(\mu_0, V_0)$, … (v)
where μ_0 = mean (expected value of state vector),
V_0 = covariance matrix of state vector

b. Since prior is normal, the posterior will be normal. The forecast observation would be normal.

c. The next parameters will be updated according to Bayes theorem:
Bayes theorem:

$p(x_t | y_t) \propto p(y_t | x_t)\, p(x_t | y_t)$ … (vi)
Updated equation using Bayes rule:
$V_{t|t} = V_{t|t-1} - V_{t|t-1} H^t M_T^{-1} HV_{t|t-1}$ … (vii)

d. These variables are set for the next update

4. Gibbs Sampler:
Assumptions:

- The initial $x_0 \sim N(\mu_0, V_0)$,
- The covariance matrix Σ and Γ are known,
- Given F, the distribution x_t is Gaussian

The structure of the sampler can be derived in a Bayesian framework as follows:

a. Stage one, state vector is derived from transition matrix as:
$x_0 | F, x_{t-1} \sim N(Fx_{t-1}, \Sigma)$
b. Transition matrix is derived from state vector as:
The state equation:
$x_t = Fx_{t-1} + u_t$;(t = 1, 2, 3, … n)…(same as i)

can written in column vector form as:
$X_n = F^T X_{n-1} + U_n$; (t = 1, 2, 3, … n) … (viii)

where, variables are column matrix, such that

$Xn = \{ x_1^T \ldots u_i^T \ldots \ldots u_n^T \}$ … (ix)

$Un = \{ u_1^T \ldots u_i^T \ldots \ldots x_n^T \}$ … (ix)

$F = \{ F_1^T \ldots Fi^T \ldots \ldots F_p^T \}$ … (x)

Then, the covariance matrix for variance-covariance matrix of F is given by:

$S(F^T) = \{ Sij\, (Fi^T Fj^T) \}$ … (xi)

where, Sij denotes the (i, j) element of the matrix

Consequently, the posterior distribution of F is given by:

$p(F^T | X) \propto |S(F^T)|^{-n/2}$ ... (xii)

The solution framework of state space model is shown in Figure 1.

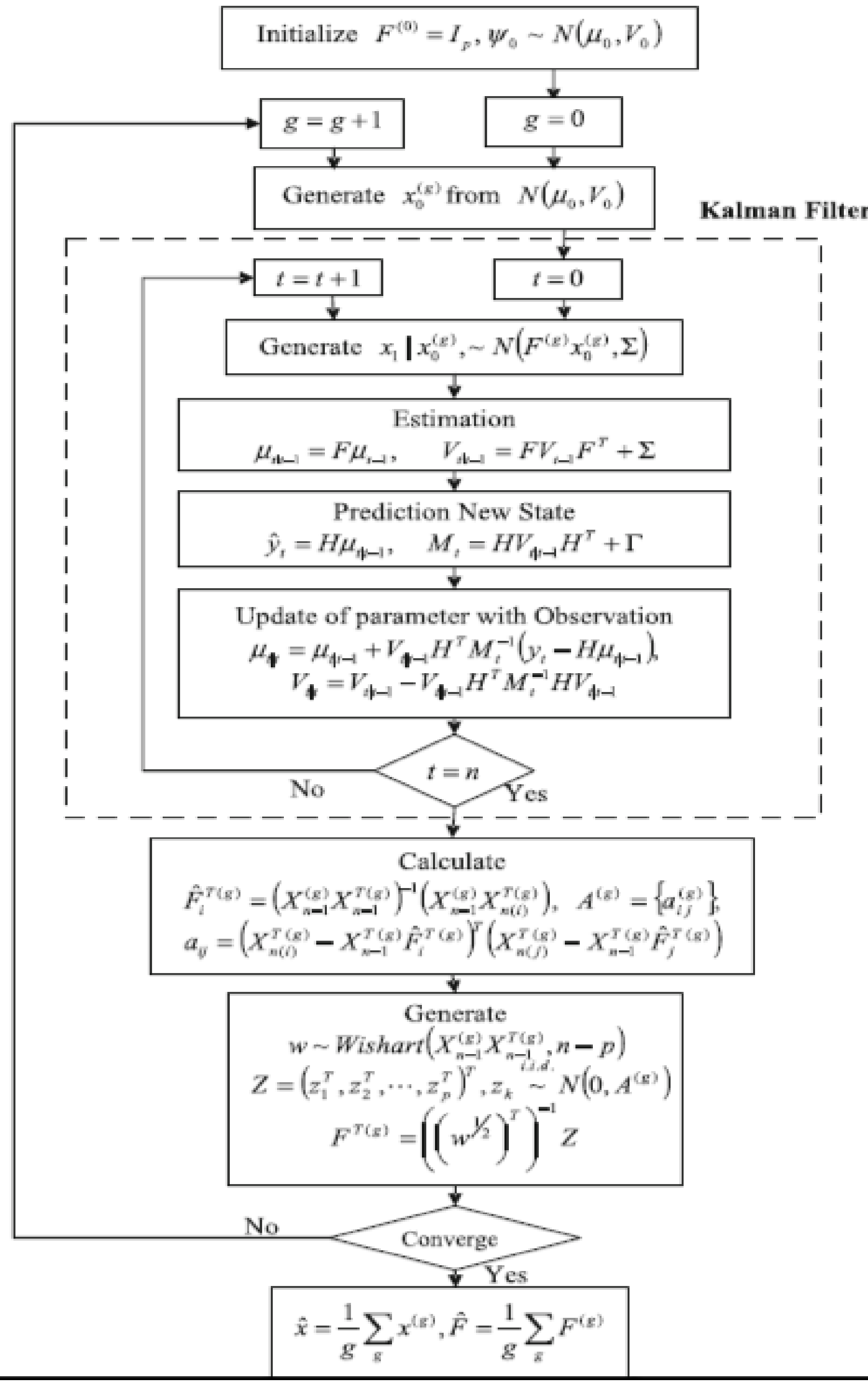

Figure 1: solution framework of state space model

IV. Parallel Implementation

Convergence:

The potential scale reduction can be estimated by:

$$\sqrt{R} = \sqrt{\frac{V}{W}\frac{df}{df-2}}$$
$$= \sqrt{\left(\frac{n+1}{n} + \frac{M+1}{Mn}\frac{B}{W}\right)\frac{df}{df-1}}$$

where,

$$df = \frac{\left(\sum_{m=1}^{M} S_m{}^2\right)^2 (n-1)}{\sum_{m=1}^{M} S_m{}^4}$$

and,

V = current variance estimate,
W = within-chain variance,
R is the ratio of,V to W,
n = the count of iterations,
S_m = the standard deviation of each chain,
B = between-chain variance, and
M = number of parallel chains.

Parallel Chain Convergence:

Gibbs sampling requires tremendous iterations during computation. To achieve real-time information requirement, parallel computing technique is introduced to increase the performance. Tens of thousands of chains are created with different random seeds. The chain in each computing part is then gathered to check the Gibbs convergence. At this step, communication between computing nodes is minimum, and computing power can be easily increased without communication bandwidth limitation

Steps:

1. Initialize the parameters used and the necessary input data.
2. The computational procedure for the parallel process consists of:
   a. Load input data and parameters. Initialize MPI environment.
   b. Count the computing nodes exists in the cluster environment. Send data to each computing nodes.
   c. Each computing nodes generate its own X and F' sequences by given input data. These results are sent to the server for convergence check.
   d. The server checks the convergence(R).
   e. If R > 1.1, the computing nodes will continue step 3. otherwise, the server estimates the global X and F by sequences generated by computing nodes.
   f. Stop MPI environment. Output data

V. Results

The technique has been implemented on a part of a real network of the Taipei Mass Rapid Transit (MRT)

The parallel environment of PC-cluster used consists (Figure 2)

- 16 computing nodes; each contains two processors equivalent to Intel XEON 3.2 GHz and 1 GB memory.
- Nodes are connected with a Gigabits Ethernet switch for MPI protocol and a 100 Mbits PCI fast Ethernet switch for Network File System (NFS) and Network Information System (NIS).
- For assigning jobs, the input data are sent to computing nodes in the cluster through TCP/IP base intranet with Message Passing Interface (MPI) Library through gigabit switch.

The results are shown in Figure 3. The proposed model is a good estimator of time-dependent O-D flows. The parallel scheme has efficiency of 72.5% in a 32 processors environment.

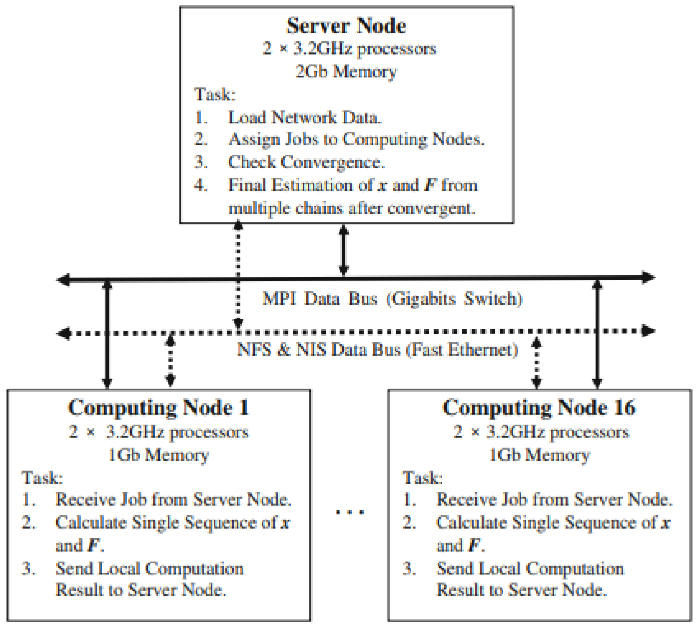

Figure 2. Parallel Implementation in Taipei MRT

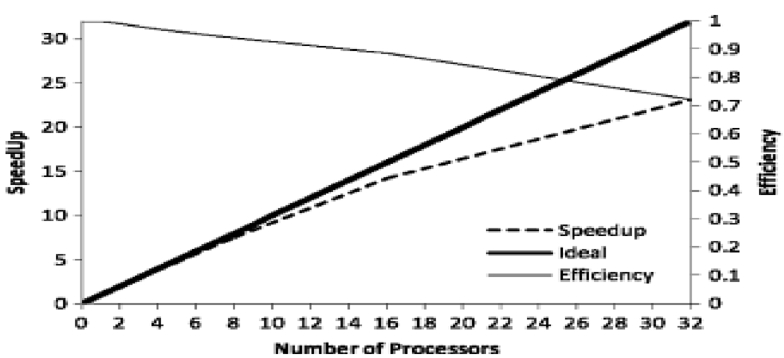

Figure 3. Speed up and efficiency vs Number of processors for the Taipei MRT experiment

VI. References

"Estimation of an Origin/Destination matrix: Application to a ferry transport data", Adrien Ickowicz, Ross Sparks

"A Kalman filter approach to dynamic OD flow estimation for urban road networks using multi-sensor data", Zhenbo Lu, Wenming Rao, Yao-Jan Wu, Li Guo1, Jingxin Xia(2014)

"Estimation of origin-destination matrix from traffic counts: the state of the art", Sharminda Bera, K. V. Krishna Rao, (2011)

"Implementing A Dynamic O-D Estimation Algorithm within the Microscopic Traffic Simulator Paramics", California PATH Working Paper, UCB-ITS-PWP-2002-4 California Partners for Advanced Partners for Advanced Transit and Highways, Reinaldo C. Garcia University of California, Irvine (2002)

"The Origin–Destination Matrix Estimation Problem — Analysis and Computations", Book by Anders Peterson (2007)

"Time Dependent Origin-destination Estimation from Traffic Count without Prior Information", Hsun-Jung Cho, Yow-Jen Jou, Chien-Lun Lan (2008)

"The Kalman filter approach in some transportation and traffic problems", Okutani I (1987)

"Control System Design: An Introduction to State-Space Methods", Friedland, Bernard (2005).