A modified parallel optimization system for updating large-size time-evolving flow matrix

Ting Yu *, Julien Ugon, Wei Yu

Integrated Sustainability Analysis, School of Physics A28, University of Sydney, NSW, Australia
School of Information Technology & Mathematical Sciences, University of Ballarat, VIC, Australia
Vancelino Technologies Inc. Shanghai, PR China.

**Abstract**

Flow matrices are widely used in many disciplines, but few methods can estimate them. This paper presents a knowledge-based system as capable of estimating and updating large-size time-evolving flow matrix. The system in this paper consists of two major components with the purposes of matrix estimation and parallel optimization. The matrix estimation algorithm interprets and follows users’ query scripts, retrieves data from various sources and integrates them for the matrix estimation. The parallel optimization component is built upon a supercomputing facility to utilize its computational power to efficiently process a large amount of data and estimate a large-size complex matrix. The experimental results demonstrate its outstanding performance and the acceptable accuracy by directly and indirectly comparing the estimation matrix with the actual matrix constructed by surveys.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

A flow matrix is a matrix representing the volumes of entities transferred between an origin and a destination. Conventionally, the row of the matrix represents the origin of the flow and the column represents the destination of the flow. The flow is defined by the application of the matrix, including commodity flow, capital flow, traffic flow and data flow, etc. Flow matrices appear in many different research areas across a wide range of disciplines. For example, commodity flow matrices and capital flow matrices are widely studied by the econometric community as input–output tables. Traffic flow matrices are studied by the traffic and urban planning community, and data flow matrices are studied by the internetworking and telecommunication community. These flow matrices share many similar characteristics, one of which is spatial interaction, which estimates the flow of people, material or information between locations in geographic space [16]. Another common character is the balancing or isolation [17], which means that the sum of each row is equivalent to the sum of the corresponding column.

The nature of the problem addressed here is that we only observe a few entries of the targeted flow matrix \(X \in \mathbb{R}^{n \times n}\) at the given time \(t\), the completed flow matrix \(X_{t-1}\) at the previous time \(t-1\), and also aggregated data for the target matrix. Then is it possible to accurately guess the missing entries of the flow matrix \(X_t\) at the time \(t\)? This paper uses the input–output table (or model) aggregation problem as a case study to demonstrate a knowledge-based estimation system for processing complex information.

In theoretical economics, an input–output table uses a matrix representation of a nation’s (or a region’s) economy to predict the effect of changes in one industry on others and by consumers, government, and foreign suppliers on the economy.
Because both national and global economy constantly evolve, the input–output table needs to be updated regularly to reflect any new circumstance. In most developed countries such as Australia, the input–output table is constructed and updated every 3–4 years, because of the large amount of monetary and human cost involved. Fig. 1 is an input–output table representing the Australian Economy. This table covers 8 states of Australia, and each state has 344 industry sectors and 7 final demands or 6 value added sectors, thereby around 7,840,000 variables have to be surveyed to estimate the entries and populate this table. The system in this paper utilizes various types of data to populate this table automatically with minimum human interaction. The system consists of two major parts: a matrix estimation algorithm and a parallel optimization engine. The matrix estimation algorithm retrieves row data from various data sources, and then restructures and integrates

<table>
<thead>
<tr>
<th>Out of domestic production</th>
<th>Into domestic production</th>
<th>Domestic inter-industry transactions</th>
<th>Total output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Domestic final uses</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value added</th>
<th>Domestic primary inputs to domestic production</th>
<th>International imports</th>
<th>Total input</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>International imports</td>
<td>Gross State Input into industrial sectors</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>X&lt;sub&gt;(i)&lt;/sub&gt;</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domestic final uses</th>
<th>Domestic final uses</th>
<th>International exports</th>
<th>Intern. state imports</th>
</tr>
</thead>
<tbody>
<tr>
<td>f&lt;sub&gt;i&lt;/sub&gt;(&lt;sup&gt;r&lt;/sup&gt;) = final demand of region (&lt;sup&gt;r&lt;/sup&gt;) for the output of sector (&lt;sup&gt;i&lt;/sup&gt;) in region (&lt;sup&gt;r&lt;/sup&gt;).</td>
<td>f&lt;sub&gt;0&lt;/sub&gt;(&lt;sup&gt;r&lt;/sup&gt;) = international exports out of sector (&lt;sup&gt;i&lt;/sup&gt;) in region (&lt;sup&gt;r&lt;/sup&gt;).</td>
<td>e&lt;sub&gt;0&lt;/sub&gt;(&lt;sup&gt;r&lt;/sup&gt;) = international exports into final demand in region (&lt;sup&gt;r&lt;/sup&gt;).</td>
<td>M&lt;sub&gt;r&lt;/sub&gt; = international re-exports in region (&lt;sup&gt;r&lt;/sup&gt;).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total output</th>
<th>International imports</th>
<th>International imports</th>
<th>Intern. state exports</th>
</tr>
</thead>
<tbody>
<tr>
<td>G &lt;sup&gt;r&lt;/sup&gt;</td>
<td>X&lt;sub&gt;(i)&lt;/sub&gt;</td>
<td>F&lt;sub&gt;r&lt;/sub&gt;</td>
<td>E&lt;sub&gt;r&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

Fig. 1. An input–output table representing the Australian Economy [8]. Its structure is defined by two three-level hierarchies including 8 states and n industry sectors.
them into a matrix estimation model. Then the matrix estimation model is solved by the optimization engine. The result from the optimization engine is an estimation of the target matrix, i.e. input–output table. (see Fig. 2).

2. Background and related works

The available information for estimating the target flow matrix comes in three formats. First, the observed entries about $X_t$ is a sample set of entries $X_{t,ij}$, where $\Omega$ is a subset of the complete set of entries, and the sampling operator $P_\Omega(X)_{ij} = \begin{cases} X_{t,ij} & (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$. Secondly, the linear functional of the entries of $X_t$, for example $U_1 = \sum X_{t,ij}$. The sources of these linear functionals come from different levels of geometry regions. Thirdly, temporal stability is assumed, so that the previous flow matrix $X_{t-1}$ provides a starting point for the updating.

On Table 1, each entry $x_{i,j}$ represents the commodity flow between the industry sections of different regions. For example, the entry $x_{3,5}$ represents the commodity flow from the sheep industry in Victoria (VIC), a state of Australia, to the shoe manufacturing industry in China. In this example, goods worth of 0.23 million dollar are sold by the Victorian sheep industry. Often the value of $x_{i,j}$ is missing as very detailed information is difficult and expensive to survey. However government agents frequently publish aggregated information, such as Vs and Us which represent the total input and production of a given industry in Australia (see Table 1). Some of the aggregated information is available for a rather long period, for example agriculture information from 1861 to 2007 in the database [2]. The aggregated information is not limited to the sums of rows or columns as Vs and Us. The main purpose of this knowledge-based estimation system is to utilize available aggregated information and the input–output tables from previous years to populate and update current or future $x_{i,j}$s to populate a series of input–output tables for current year or coming years.

A time series of input–output tables represents the evolution of industry structure within and between regions, where a region is defined as a geographic concept. The input–output tables often consist of a time series of matrices which may have temporal stability or temporal patterns. On the other hand, within a given time period, extra spatial information regarding certain parts of the matrix is often available from various government departments or other public or private organizations. The spatial relations, both metric (such as distance) and non-metric (such as topology, direction etc.) and temporal relations (such as time trend) may be explicit or implicit in the databases [21]. Spatiality and temporality are two unique dimensions, and conventional machine learning does not recognize the uniqueness of spatial and temporal dimensions [19].

![Figure 2](image.png)

Moreover, most spatio-temporal information is often incomplete and only gives snapshots of parts of the underlying model, and may even contain conflicting bits of information. Although data conflicts occur in non-spatial machine learning [7] as well, the conflicting reconciliation strategies proposed in non-spatial machine learning are not readily applicable for a spatio-temporal datasets. Apart from the massive data, dozens of years of research has accumulated substantial amount of general knowledge of the national economy. This public knowledge can be used to facilitate the discovery. On the contrary, many other computational modeling activities often do not have such rich resources [18,22].

In information science community, major research has been done to estimate matrices for transportation planning and network design, and discover and preserve the structure of the matrix. There are three main approaches: Linear Programming approach, Bayesian Inference techniques, and Expectation Maximization (EM) [14]. These methods are widely employed to estimate the traffic flow matrix in the transportation planning [13]. These results have significant influence on the algorithm discussed in this paper, because the nature of traffic flow is very similar to the nature of the industry flow. However the data availability and quality causes differences between the two problems. There is a large amount of heterogeneous data related to the national economy, and the data often contains a large amount of noises. By contrast, traffic flow data is relatively simply structured and normally collected very accurately.

**Matrix completion or low-rank matrix recovery** assumes that the only available information is a subset of matrix entries that is $P_\Omega(X)$ [3,4,10,11]. The formula of the matrix completion is:

\[
\text{Min} \| X \| \quad \text{subject to } \quad P_\Omega(X) = P_\Omega(M),
\]

where the nuclear norm $\| X \| := \sum_{k=1}^r \sigma_k$ is the sum of the singular values, and $P_\Omega(M)$ contains the subset of observed entries of the unknown matrix $M$. Clearly, this method is closely related to the Singular Value Decomposition (SVD) of a matrix of rank $r$.
r. \( X_t = \sum_k \sigma_k u_k v_k^T \), where \( \sigma_k \) is the singular values, \( u_k \) and \( v_k \) are the singular vectors. This method is based on an assumption: the unknown matrix is known to have low rank or approximately low rank. In the case of a high-rank matrix, it is impossible to estimate the other entries only based on the available subset. More importantly, this method is unable to utilize the linear functionals and previous matrix which often carry related information.

In econometric research, the Gravity Model is employed to preserve and reveal spatial relations by considering the transport distance [8]. Apart from that, the RAS type methods provide a procedure of estimating and updating the matrix [12,17]. Still by using the previous problem, the matrix of the current year can be updated as follows: first, construct the technical coefficient matrix \( A_{t-1} \) of the previous year by dividing each element of the matrix \( X_{t-1} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{pmatrix} \in \mathbb{R}^{n \times n} \) by the sum of its corresponding row:

\[
A_{t-1} = \begin{pmatrix} x_{1,1}/U_1 & \cdots & x_{1,n}/U_1 \\ \vdots & \ddots & \vdots \\ x_{n,1}/U_n & \cdots & x_{n,n}/U_n \end{pmatrix} \in \mathbb{R}^{n \times n},
\]

where the sum of the first row \( U_1 = \sum x_{1,j} \). Second, a ratio matrix \( R_t \in \mathbb{R}^{n \times n} \) is constructed by dividing the sums of rows of the current year \( U \in \mathbb{R}^n \) by the sums of rows of previous year \( U^0 \), where

\[
R_t = \begin{pmatrix} U_t^0/U_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & U_t^0/U_n \end{pmatrix}.
\]

The second ratio matrix \( S_t \in \mathbb{R}^{n \times n} \) is constructed by dividing the sums of columns of current year \( V \in \mathbb{R}^{m \times n} \) by the sums of columns of previous year \( V_{t-1} \), where

\[
S_t = \begin{pmatrix} V_{t-1,1}/V_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_{t-1,n}/V_n \end{pmatrix}.
\]

The technical coefficient matrix \( A_t \) of the current year can be approximated as \( A_t \approx R_t A_{t-1} S_t \) until the satisfied result is reached. The problem of the RAS method is: (1) it assumes that the available information, such as \( V_t \), \( U_t \) and \( X_{t-1} \), are perfect without errors; (2) when extra information other than \( V_t \), \( U_t \) and \( X_{t-1} \) is available, the RAS method is unable to utilize it; (3) users do not have the flexibility to control the impacts from various information.

The modified parallel optimization algorithm discussed in this paper extends the linear programming method by considering the conflicting information and implementing it over parallel computing facility. A similar parallel optimization algorithm can be found in [24]. In order to frequently update the flow matrix, the software package is designed to integrate and simplify the data collection and data fusion process and facilitate users to conduct a spatio-temporal knowledge discovery process [5]. Including temporal information introduces additional complexity to the geographic knowledge discovery. The modified parallel optimization aims to balance the impacts from various sources to reach the best approximation to the unknown matrix. Since this method combines information from heterogeneous sources, it can be classified as a type of data fusion from multi-dimensional data [9,20].

3. System overview

The whole system consists of two major parts: matrix estimation algorithm and parallel optimization engine. The matrix estimation algorithm retrieves row data from various data sources, restructures and integrates them into a matrix estimation model. Then the matrix estimation model is solved by the optimization engine. The result from the optimization engine is the estimate of the target matrix. This optimization engine is built upon a parallel supercomputing facility, and automated to reduce human intervention. The parallel computing requires the modeling methods to be specially designed to fully utilize the computational resource to process massive amount of data efficiently. As a consequence, a parallel optimization method is proposed here and this parallel spatio-temporal modeling method is one of a few parallel version of this type of spatio-temporal knowledge discovery algorithm.

The matrix estimation algorithm employs various interfaces to all types of datasets that are stored in various formats such as Excel files, databases etc. the data sources accessed in this paper includes public information published by the Australian Bureau of Statistics, for example, the Australian National Accounts: State Accounts, Environment and Energy, and Economy, Industry, Value of Agricultural Commodities Produced. Other government agents, Australian business register and Reserve Bank of Australia regularly publish data about private business. Aside from government, private companies, e.g. Sydney Water, release data for a particular industry. The matrix estimation algorithm unifies these heterogeneous datasets to a single format, integrates and restructures the data retrieved by the previous component and presents the result as a matrix estimation model. The parallel optimization engine then solves the matrix estimation model to produce the flow matrix.

4. Matrix estimation algorithm

The matrix of the input–output table is “vectorized” row-by-row to form a row vector. Suppose the input–output table is in format of a \( n \times v \) matrix, and the “vectorizing” process horizontally concatenates the rows of the matrix to form a row vector, \( X \in \mathbb{R}^k \), where \( k = n \times v \). The matrix estimation algorithm integrates all necessary data together and from a matrix estimation model as Eq. (1):
where $X_t$ is the target matrix to be estimated, $X_{t-1}$ is the matrix of previous year, $E$ is a vector of the error components $[e_1, \ldots, e_1]^T$, $dis$ is a distance metric which quantifies the difference between two matrices, e.g. squared L2 norm $\sum (X_{t-1} - X_{t-1})^2$ in this case. $G$ is the coefficient matrix for the local constraints, and $C$ is the right-hand side value for the local constraints. Linear functional $G_1 X_t + E = C_1$ and $G_2 X_t = C_2$ contains the aggregated information (e.g. local regional information etc.) about the unknown entries. $P_0(X_t) = P_0(M)$ forces the optimal solution to be equal to the observed entries.

The idea here is to minimize the difference between the target matrix and the matrix of the previous year, while the target matrix corresponds to the local regional information to some degree. For example, if the total export of the sheep industry from Australia to China is known as $c_1$, that is $U_1$ in Table 1, then $G X + E = C$ can be $[1, 1]^T X_t + e_t = c_1$. The element $e_t$ in $E$ represents the difference between the actual value and estimate value, that is, $e_t = c_t - [1, 1]^T X_t$. The reason why it is introduced is to resolve the conflicting information.

First, the estimation algorithm categorizes data into two types: the historical information which contains the temporal patterns between matrices of previous years, such as the trend of productivity of industry sections, and the spatial information within the current year, for example, the total production within a certain region such as national total emission and state total emission within the current year. This estimation algorithm assumes temporal stability, meaning that the industry structure of a certain region remains constant or has very few changes within the given time period. By this assumption, this algorithm considers the matrix for the previous year as a reasonable proxy to the current year. Within two successive years, dramatic change of the industry structure is relatively rare, and this assumption has a good ground. At the same time, the estimation algorithm is required to incorporate the spatial information and keep the spatial relationship (such as dependency and heterogeneity [15]) within datasets.

Secondly, as data is collected from different sources, it is common that the dataset is not comprehensive and imperfect and even conflicts within the dataset exist. Therefore, the estimation algorithm is required to consolidate the conflicting data to uncover underlying models. Here $e_t$ is introduced to balance the influence and reach a tradeoff between the conflicting information. By introducing $e_t$, constraints with $e_t$ is called “soft” constraints, $G_1 X + E = C_1$ in Eq. (1). This subset of constraints addresses the conflicting information and the rest of constraints are called “hard” constraints which the optimal solution has to satisfy.

Furthermore, this system is designed for various purposes and aims to process heterogeneous data in unforeseeable formats. It requires certain flexibility to fit into new applications, and process data in new format. The matrix estimation model for real applications often contains million of data, and therefore it is infeasible to manually fill the data. The matrix estimation algorithm automates this process by introducing functional components including dynamic tree structure, data source interfaces and data fusion.

The first step of this matrix estimation algorithm is to construct the dynamic tree structure to represent the dependency and heterogeneity inherited from the spatial data. The tree structure is pre-required for restructuring data collected from various sources. The tree structure represents the geographic concept hierarchies within the spatial information. It is a natural representation of hierarchy of national or global economy. An example of tree structure is a three-level tree representing the Australian Economy (See Table 1), one branch of which represents the sheep industry section within the New South Wales, a state of Australia. If the numerical indices are employed instead of their names, the sheep industry section within the New South Wales, a state of Australia can be written in [1, 1, 1] which means the first leaf in the first branch of the first tree.

The row and column of the matrix is defined by this multi-level tree structures, thereby the matrix is defined by the tree structures. A matrix (see Table 1) can be organized by one three-level tree at the row side and one two-level tree at the column side. The coordinate of one entry, say $x_{11}$, can be defined as by [1, 1, 1] at the row side and [1, 1] at the column side. That means the entry, $x_{11}$, defined by a three-level tree structure and a two-level tree structure at the column side. The tree structure is crucial to assign the meaning to the data retrieved from various sources and reorganize them later, since the coordinates of entries are completely determined by it.

Considering the difference between applications, a dynamical structure of resultant matrix provides the flexibility to expand this software system to different application. On the other hand, the flexibility of the structure also makes the system to be applicable to various level of implementation. For example, there is huge difference between the structures of resultant matrix at the national and at the corporate level, as the operations within a corporate are much simpler than those of a nation in the most cases.

Considering the complexity of defining a model, a meta language is introduced to provide users’ an easy way to describe their intention to fuse heterogeneous data via selection and restructuring. The Meta language must be compact and accurate to make the description readable and useful. It is unrealistic to write hundred thousands of code to describe a single model.
on a daily basis. The meta language presented in this paper is based on the coordinate of the valuable in the resultant matrix, which is defined by the dynamic tree structure. For example, the coordinates of one entry is written as $[1, 2, 1 \rightarrow 1, 1]$. The value of this entry $x_{1,1}$ is indicated as $(0.23) [1, 2, 1 \rightarrow 1, 1]$ (see Table 1). The system will fill the value 0.23 in the cell with the coordinate $[1, 2, 1]$ at the row side and $[1, 1]$ at the column side. Consequently, this script indicates that 0.23 dollar worth of sheep products are transferred to the shoe industry in China.

5. Modified parallel projection optimization

In real world practice, the previous estimation algorithm often processes a matrix whose dimension is over thousands-by-thousands. In the foreseeable future, the size of estimated matrix will increase over 100,000-by-100,000. This requires the algorithm to have the capacity to process large datasets. In order to address this problem, one parallel optimization algorithm is implemented over the Message Passing Interface (MPI).

5.1. Performance

A small size problem is used to test the convergence of the proposal algorithm with 4 CPUs (or nodes), and the small problem estimates 5 variables with 7 linear constraints. Here the problem is presented as Eq. (2), in which

\[
\begin{align*}
  X_{t-1} = [1, 1, 1, 1, 1], \quad \sigma_e = [0.1, 0.1, 0.1, 0.1, 0.1], \quad G = \begin{bmatrix}
  1 & 0 & 1 & 0 & 0 \\
  0 & 1 & 0 & 1 & 0 \\
  1 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 1 \\
  0 & 0 & 1 & 0 & 0 
\end{bmatrix}, \quad \text{and} \quad C = [1, 3, 1, 1, 1, 1, 4].
\end{align*}
\]

“hard” constraints which must be satisfied, and the rest of constraints are “soft” constraints. The 1st constraint is conflicting
with the 7th constraint, while $X_t$ is nonnegative. After 926 iterations, the optimal solution is $[0.0118072, 1.32451, 4.05129, 1.0118, 2]$. From Fig. 3, the distance from the solution to the constraint set declines and becomes stable after about 200 iterations. The same phenomenon occurs with the values of the objective function.

A medium size optimization problem consists of 25,070 variables, 219 constraints. The optimization runs 5,000 iterations over 16 CPUs (or nodes). The whole process takes 01:15 min and uses 918 MB memory totally. A large size optimization problem consists of 3,340,800 variables, over 3,100 constraints. The optimization runs over 2,000 iterations over 16 CPUs (or nodes). The whole process takes 37:29 min and uses 2,280 MB memory totally.

6. Experimental results

In the real world, few government agents have published two consecutive input–output tables. In the meantime, researchers are keen to control noise levels (or conflicting information level) of the data, and also keen to measure the impact of different noise level on the resulted matrix. In order to examine the accuracy of this estimation approach and its sensitivity to noise, a set of artificial data is generated by following the main characteristics which researchers are facing in the real practice. In this data set, the constraints include: (1) linear functional: the sum of each row is equal to the sum of the corresponding columns; (2) known subsets of entries of the matrix; (3) known sums of some subsets of entries of the matrix; and (4) known ratios between entries of the matrix.

The procedure of generating the artificial data set is: (1) generate the input–output table presenting the previous year, (2) generate another input–output table representing the difference between previous year and current year, and then (3) add up these two input–output tables to generate the input–output table for the current year. In the following experiment, the maximum value of entry of the 120-by-120 input–output table for previous year is set as 10,000, and the maximum value of entry of the 120-by-120 table of the difference is set as 2,000.

The direct evaluation of the accuracy of estimating a large-size matrix is a rather difficult task. A thousand-by-thousand matrix contains up to ten million numbers. A simple measurement, such as sum, does not make too much sense, as any important deviation is submerged by the total deviation which normally is far bigger than individual ones. The key criterion here is to correctly estimate the distribution or the interrelationship between the entries of the matrix: the matrix reflects the true underlying structure, not necessary the exact value, but at least the right ratios. As a direct evaluation of the difference between two matrices, matrix distance can be measured in many different ways. Here some prominent methods are listed [8]:

The relative arithmetic mean of absolute differences:

$$AMAD = \frac{\sum_j |X_{act,i} - X_{sup,i}|}{\sum_i X_{sup,i}}.$$  

The relative geometric mean of absolute differences:

$$GMAD = \sqrt[\frac{1}{2}] {\sum_j |X_{act,i} - X_{sup,i}|^2 \sum_i X_{sup,i}}.$$  

The Isard/Romanoff Similarity Index: $DSIM = \frac{\sum_i (|X_{act,i} - X_{sup,i}|)}{\sum_i X_{sup,i}}$.

The Chi-square distribution of absolute difference: $CHI = \sum_i \frac{(X_{act,i} - X_{sup,i})^2}{X_{sup,i}}$.

The arithmetic mean of relative differences: $AMRD = \frac{1}{N} \sum_i |X_{act,i} - X_{sup,i}| \sum_i X_{sup,i}$.

The correlation coefficient: $CORR = \frac{\sum_i (X_{act,i} - X_{sup,i}) \sum_i X_{sup,i}}{\sqrt{\sum_i (X_{act,i} - X_{sup,i})^2 \sum_i X_{sup,i}}}$.
This experiment compares three matrix estimation methods: RAS, Matrix completion, and proposed Paralleled Matrix Estimation by using a set of artificial data. RAS only uses the linear functional and matrix of previous year, and Matrix completion method only uses the observed entries of the target matrix. The code of matrix completion method is coded by Emmanuel Candes and Stephen Becker [4], and is downloaded from the website: <http://www.convexoptimization.com/wikimization/index.php/Matrix_Completion.m>. The distance between the input–output table of previous year and the table of current year is measured to demonstrate the efficiency of three methods.

On Table 2 the first row, Table of previous year, demonstrates the dispersion between two artificial tables (actual table and the table for previous year) created for this experiment. The second row demonstrates the dispersion between the actual table and the table estimated by the Paralleled Matrix Estimation method presented in this paper. The third row, RAS, demonstrates the dispersion between the actual table and the table estimated by the RAS method, and the fourth row, matrix completion, demonstrates the dispersion between the actual table and the table estimated by the matrix completion method. According to the results, the values of $AMAD$, $GMAD$, $DSIM$ and $CORR$ indicate the table estimated by our method has the shortest distance from the actual table, and values of $CHI$ and $AMRD$ indicate that input–output table for previous year has the shortest distance from the actual table. The RAS also demonstrates rather good estimation, because the $V$s and $U$s cover the entire actual table with certain degree of noise at this experiment. The added noise causes the accuracy of the RAS to be inferior to our method. The matrix completion method provides the worst result, mainly due to the low ratio of observed entries to the entire table in this experiment. Figs. 4–6 demonstrate the distribution of the dispersion between the actual table and the table estimated by the Paralleled Matrix Estimation Method. On Fig. 4, the estimated entries and real data are distributed along the diagonal line of entries running from the top-right corner to the bottom-left corner, but the dispersion is bigger for entries with higher values. The same phenomenon occurs on Fig. 5. However, when the difference is normalized by the real value of each entry, the largest dispersion occurs at the lower end instead of high end. This indicates the estimation method tries to distribute the unsolved portion of gap to each entry by portion to its value. It means that in the estimation process, the most of residual $X_{est}^j - X_{act}^j$ of each entry is portion to $X_{est}^j$, apart from a few extreme observations. As we stated previously, the largest difference between entries of two tables is set as 2,000, and the highest value of entries of the table is set as 100,000 in this experiment. On Fig. 5, the largest difference between the real table and estimated table is between 200 and 250. Clearly this method manages to control the majority of its estimation error below 10% of the real difference between two tables.

The multipliers in the input–output framework reflect the aggregated impacts of the final demand changes on the upstream industries [17]. The information contained by the multipliers is very similar to the sensitivity analysis in general.

Table 2

<table>
<thead>
<tr>
<th></th>
<th>AMAD</th>
<th>GMAD</th>
<th>DSIM</th>
<th>CHI</th>
<th>AMRD</th>
<th>CORR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of previous year</td>
<td>0.1544</td>
<td>0.1635</td>
<td>0.1273</td>
<td>2.9504e+004</td>
<td>0.2017</td>
<td>0.9904</td>
</tr>
<tr>
<td>Paralleled Matrix Estimation</td>
<td>0.1185</td>
<td>0.1278</td>
<td>0.1250</td>
<td>3.7159e+004</td>
<td>0.4877</td>
<td>0.9909</td>
</tr>
<tr>
<td>RAS</td>
<td>0.1655</td>
<td>0.1744</td>
<td>0.1265</td>
<td>3.8101e+004</td>
<td>0.4944</td>
<td>0.9843</td>
</tr>
<tr>
<td>Matrix completion</td>
<td>0.2154</td>
<td>0.2055</td>
<td>0.1433</td>
<td>4.0111e+004</td>
<td>0.4701</td>
<td>0.8801</td>
</tr>
</tbody>
</table>

Fig. 4. Comparison between the actual table and estimated table estimated by the Paralleled Matrix Estimation Method.
statistics. The general equation of constructing a multiplier is: \( M = D(I - A)^{-1} \) where \( M \) is the multiplier, \( I \) is the identity matrix, \( D \) is the change in the final demand, and \( A \) is the matrix, each entry of which is \( x_i / \sum_{i=1}^{n} x_i \). Here, \( x_i \) is a value from the matrix estimated by the Eq. (1).

This sensitivity multiplier counts the impact of any change of outputs on the whole upstream inputs, and not only the direct inputs. Any deviation occurring in the upstream inputs from the underlying true structure will be amplified and reflected on the multipliers. Thereby, the multipliers send an indirect warning signal to imply the structural deviation occurring on the upstream inputs.

As a case study, we construct a matrix showing the total water usage of the different industries in Australia. A part of the data is collected from the Water Account reports produced by the Australian Bureau of Statistics [1]. The full Australian Economy consists of 8 states with 344 industry sections plus 7 final demands and 6 value added sections. In total, the Australian input–output table is a 2808-by-2800 matrix, containing 7,862,400 entries. In order to estimate the matrix, more than 260,000 constraints are included. This experience setting is the same as the case study in the previous study [23].
Based on comparison the estimated series and the actual series of the multipliers, two series basically follow the same trend, which indicates the industry structure is estimated properly. However, the estimated multiplier series is more volatile than the true underlying multiplier series. This result is consistent with the previous research [23]. This phenomenon indicates the estimated multipliers amplify the errors introduced to the upstream industries. Another possible explanation of this difference is an underlying structural change within a given industry. One of the big gaps between two series indeed indicates from 1999 to 2004, the Australian rice industry dramatically reduces its rice production due to the continuous draught, but imports more and more rice from other nations. As its price has been inflated but its water usage drops, the ratio of the water usage by price drops dramatically.

7. Conclusion

This paper presents an integrated knowledge-based system for estimating and updating a large-size flow matrix. The unique characteristics of flow matrices require the system to be capable of dealing with a large amount of temporal and spatial data efficiently. The complexity and heterogeneous nature of data leads to a powerful meta-language describing the data sources efficiently, and highly noisy and inconsistent data forces the matrix estimation method to address conflicting information. This completed system starts from data collection, and progresses data fusion to matrix estimation. According to the result of the experiments, the system successfully produces the matrix, and makes it a rather easy task without a huge amount of work to collect and update both data and matrix. Before this system, this kind of collection and updating work cost months of workload, but now it takes only a few days and provides result with consistent quality. Furthermore, this system and these methods can be easily applied for other flow matrix estimation.

Table 3 compares the various matrix updating methods in terms of the data sources they can utilize and their capacity. The modified parallel optimization method being discussed in this paper covers the broadest range of data sources and owns the extra capacity to deal with conflicting information and extremely large-size datasets.

In general, the coverage of available data to the target matrix is one of the key criteria influencing how accurate the estimated matrix is. By including more data sources, new methods can be less sensitive to the quality of an individual data source. In contrast, the ratio of observed entries to the unknown entries of the matrix is very important to the matrix completion method. As temporal stability is a major assumption, further developments will emphasis how to deal with the major structural changes of the matrix.

Acknowledgement

Authors would like to give thanks to Prof Manfred Lenzen, Dr. Chris Dey, and Dr. Joy Murray at the University of Sydney.

References


