Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

An Integrated Intelligent System for Estimating and Updating a Large-size Matrix

Ting Yu
Centre of Integrated Sustainability Analysis, Physics Building A28,
University of Sydney, NSW 2006, Australia
t.yu@physics.usyd.edu.au

***Abstract*—this paper presents an integrated intelligent system being capable of automatically estimating and updating a large-size matrix. In the theoretical economics, the input-output model of economics 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. The system in this paper aims to estimate the large-size input-output model and consists of a series of components with the purposes of data retrieval, data integration, data analysis, and quality checking. This unique system is able to interpret and follow users' XML-based query scripts, retrieve data from various sources and integrate them for the following data mining components. The data mining component is based on a unique modelling algorithm which constructs the matrix from the historical data and the spatial data simultaneously. This unique data analysis algorithm runs over the parallel computer to enable the system to estimate a matrix of the size up to 3700-by-3700. The result demonstrates the acceptable accuracy by comparing a part of the multipliers with the corresponding multipliers calculated by the matrix constructed by the surveys.**

***Keywords*—integrated intelligent system, matrix estimation, parallel computing**

I. Introduction

In the theoretical economics, the input-output model of economics 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 [1]. Because the economic constantly evolves, the input-output model needs to be updated at least annually to reflect the new circumstance. Unfortunately, in most countries such as Australia, the input-output model is only constructed every 3-4 years, because the large amount of monetary and human cost is involved. The Centre for Integrated Sustainability Analysis (ISA), University of Sydney, is developing an integrated intelligent system to estimate and update the input-output model at different level on a regular basis.

The input-output model often consists of a time series of matrices which may have temporal stability or temporal patterns. At the same time, within a given time period, extra information regarding certain parts of the matrix is often available from various government departments or other public or private organizations. However, most of this information is often incomplete and only gives a snapshot of a part of the underlying model. Apart from the massive data, hundreds of years of research has accumulated substantial amount of general knowledge of the national economic. Any researcher could utilize this public knowledge to facilitate their discovery. On the contrast, other knowledge discovery activities often do not have such rich resource.

A time series of input-output models represents the evolution of industry structure within and between regions, where the region is defined as a geographic concept. It is a spatio-temporal knowledge discovery process with the help of rich domain knowledge. Including time introduces additional complexity to the geographic knowledge discovery [2]. This paper presents a novel algorithm which estimates and updates the economic matrix for the general equilibrium theory.

II. System Design

The whole system consists of a series of functional components: data retrieval and integration, query, data mining and model presentation (See Figure 1).

As the first step, the data retrieval component acts as interfaces to various types of datasets including macro and micro economic data that are stored in various formats such as Excel files, databases etc. The data integration component unifies these heterogeneous datasets to a single format, integrates and restructures the data retrieved by the previous component. At the same time, users' specification of the concept hierarchy is interpreted and translated into hierarchies defining the structure of the matrix. This concept hierarchy is very similar to the data warehousing [3]. This hierarchy allows users to roll up and drill down the data very easily and also introduce the dynamic to the matrix. The data mining component employs a unique algorithm designed to estimate the matrix. In order to process extremely large amount of data, this component sits on a parallel optimization algorithm to quickly converge to the optimal estimation of the unknown matrix. As the final step, the model checking and presentation components check the quality of the estimated model and present it to users in a structured format.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

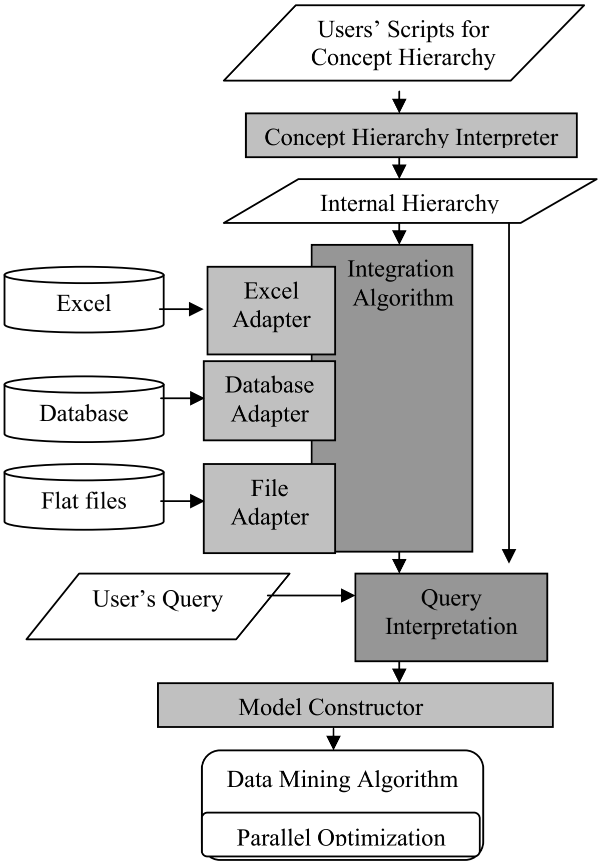

Figure 1. System Architecture

At the following sections, some key components are discussed in detail.

A. Model Constructor

The model constructor component communicates with other two components: the concept hierarchy interpreter and query interpreter. The concept hierarchy interpreter constructs the tree-like hierarchy that we will discuss in detail later, and the query interpreter translates the users' query written in a special meta language. The model constructor then 1) require the data integration component to retrieval data from various sources and integrate them, and 2) restructure and assign the meaning to the data according to the previous concept hierarchy and users' query in order to populate the data mining model.

On the process of building a model, the first step is to construct the concept hierarchy. The hierarchy is pre-required for restructuring data from various sources. The hierarchy structure is introduced by a multi-tree structure. For example, the hierarchy representing Australia national economic can be like fig 2.

The hierarchy brings two major benefits. First, it provides the different levels of abstraction. The flexibility of the concept hierarchy makes the users to have snapshots of the matrix at various levels of abstraction. Users can easily drill down or roll up the matrix without redefining the matrix.

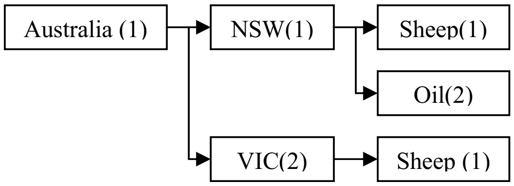

Figure 2. An Example of Concept Hierarchy

Secondly, it allows the dynamic structure of the resultant matrix. Regarding the difference between applications, a dynamical hierarchy provides the flexibility to expand this system to different application with different structure of matrices. It is very common to mix two different concept hierarchies for a matrix. For example, in Table 1, a matrix is organized by one three-level hierarchy and one two-level hierarchy. The coordinate of one entry, say X_1, can be defined as by [1,1,1] at the row side and [1,1] at the column side. That means the entry, X_1, defined by a three-level concept hierarchy and a two-level concept hierarchy at the column side.

TABLE I. AN EXAMPLE OF THE MATRIX DEFINED BY THE 3-LEVEL TREE AND THE 2-LEVEL TREE

			China (1)	
			Shoe (1)	Retail (2)
Australia (1)	NSW (1)	Sheep (1)	X_1	X_2
		Oil (2)	X_3	X_4
	VIC (2)	Sheep (1)	X_5	$X_6 = 0.23$
		Oil (2)	X_7	X_8

Considering the complexity caused by introducing the concept hierarchy, a query language is introduced to provide users' an easy but powerful way to retrieval and organize their data. The query language must be compact and accurate to make the description to be readable and expressive. It is unrealistic to write hundred thousands of code to describe a single model at a daily base. The query language we create is based on the coordinate of the valuable in the matrix whose structure is defined by the concept hierarchies. For example, the export X_6 is written as [1, 2, 1 -> 1, 2]. The value of coefficient for X_6 is indicated as (0.23) [1, 2, 1 -> 1, 2], which means this coefficient for X_6 is 0.23. Some other notations are also included to improve the flexibility and efficiency of the query language. In the system, users' specification is a set of XML-based files including some scripts written in the query language. The concept hierarchy is crucial to assign the meaning to the data retrieved from various sources, since the coordinates of X are completely determined by the hierarchies.

B. Spatio-Temporal Mining with Conflicts

In the data mining component, a unique data mining algorithm is designed to estimate the matrix. This mining algorithm utilizes two types of information: the historical information which contains the temporal patterns between matrices of previous years, and the spatial information within the current year. For example, this spatial information can be the total commodity output of the given industry within the current year, or the total greenhouse emission of the given industry. The simplified version of the mining algorithm can be written in the format of an optimization model as below:

$$Min[\frac{dis(X-\overline{X})}{\varepsilon_1}+\sum\frac{e_i^2}{\varepsilon_{i+1}}] \quad (1)$$
$$\text{subject to: } GX+E=C$$

where: X is the target matrix to be estimated , $\overline{X}$ is the matrix of the previous year, E is a vector of the error components $[e_1,...,e_i]^T$, dis is a distance metric which quantifies the difference between two matrices. As the dis metric has many variety, the one used in this experiment is $\sum(x_i-\overline{x}_i)^2$. G is the coefficient matrix for the local constraints, C is the right-hand side value for the local constraints.

The idea here is to minimize the difference between the target matrix and the matrix of the previous year, while the target matrix satisfies with 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 , then $GX+E=C$ can be $[1,1]\begin{bmatrix}X_1\\X_2\end{bmatrix}+e_1=c_1$. The element e_i in E represents the difference between the real value and estimate value, for example, $e_1=c_1-[1,1]\begin{bmatrix}X_1\\X_2\end{bmatrix}$.

This mining algorithm assumes the temporal stability, which assumes the industry structure of a certain region keeps constant or has very few changes within the given time period. This assumption often required to be verified for long time period. Within the short time period, the dramatic change of the industry structure is relatively rare.

Data objects in typical data mining algorithm often can be reduced to points in some multidimensional space without information loss. But the spatio-temporal modelling algorithm can analyze data containing two types of information simultaneously; thereby maximally utilize the available information. In our case, $dis(X-\overline{X})$ models the temporal information of the input-output models between years, and $GX+E=C$ models the spatial or other type of information of the input-output model within the current year.

The reason why the spatio-temporal modelling algorithm is suitable to this system is due to the unique characteristics of the datasets that the system aims to process. The datasets often contain the temporal patterns between years, such as the trend of the carbon emission of certain industry sections, and also much spatial information regarding the total emission within a certain region such as national total emission and state total emission. Even more, the datasets also contains the interrelationship between the industries within a given region or between regions. On the other hand, it is very common that either of datasets is not comprehensive and imperfect and even the conflicts between the datasets exist. Thereby, the modelling algorithm is required to consolidate the conflicted datasets to uncover underlying models, and at the same time, the modelling algorithm is required to incorporate the spatial information and keep the spatial relationship (such as dependency and heterogeneity [4]) within datasets.

C. Parallel Optimization

In real world practice, the previous modeling algorithm often processes matrix with dimensions over 1000-by-1000. In the foreseeable future, the size of estimated matrix will increase over 100,000-by-100,000. This requires the algorithm to have extremely outstanding capacity of processing large datasets. In order to address this problem, one parallel optimization algorithm is designed as the solver. The key idea is to divide the constraints into a few subsets of constraints, and then to do optimization against the subset of constraints respectively instead of the whole set of constraints. The simplest case is that the original optimization problem is rewritten as a set of sub-problems

Sub-problem 1 (soft constraints):

$$Min[\frac{(X-\overline{X})^2}{\varepsilon_1}+\sum\frac{e_i^2}{\varepsilon_{i+1}}]$$

$$\text{subject to } G_1X+E=C_1$$

Sub-problem 2 (hard constraints):

$$Min[\frac{(X-\overline{X})^2}{\varepsilon_1}]$$

$$\text{subject to } G_2X=C_2$$

Sub-problem 3 (nonnegative constraints):

$$Min[\frac{(X-\overline{X})^2}{\varepsilon_1}]$$

$$\text{subject to } X\geq 0$$

The results from the sub-problems are combined as a weighted sum which consequently acts as a start point for the next iteration. Suppose the result from the ith sub-problems is $P_i(X_n)$, the weighed sum is written as $X_{n+1}=X_n+L*[\sum w_iP_i(X_n)-X_n]$, where L is the relaxation parameter. This method is a special case of the parallel projection method (PPM) [5]. Because the objective function of this particular problem is quadratic, thereby convex and the constraints are linear thereby convex as well, the optimization process is simpler than general projection methods. This parallel optimization algorithm is implemented over the Message Passing Interface (MPI). For the purpose of demonstration, the performance is compared with a commercial optimization package, the CPLEX by using the same test dataset concluding 12-by-12 entries.

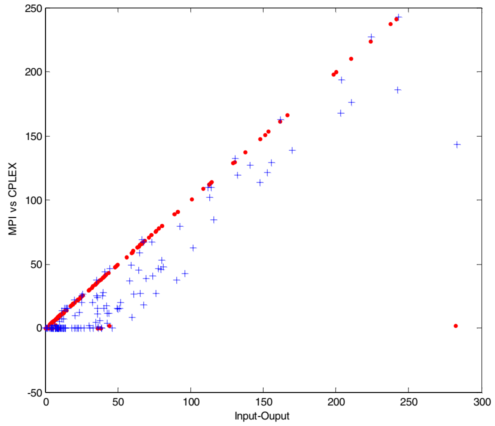

Figure 3. Results from the CPLEX (Blue Cross Points) vs. results from the Parallel Optimization (Red Dot Points) by comparing with the real matrix data

According to the experiments (see Figure 3), the parallel optimization estimates the underlying the matrix better as the linear relationship between its estimated result and the real matrix are very clear. The drawback is that the parallel optimization does not prevent the data from becoming the negative number.

III. Experimental Results

Here we present two methods of checking the quality of the estimated matrix: direct and indirect checking. The reason why we introduce the indirect method is that the direct evaluation of a large-size matrix is a rather difficult task. A thousand-by-thousand matrix contains up to ten million of numbers. The simple measurements such as the sum do not make too much sense, as the important deviation is submerged by the total deviation which normally is far larger than the individual ones. The key criterion here is the distribution or the interrelationship between the entries of the matrix: whether the matrix reflects the true underlying industry structure, not necessary the exactly right value, at least the right ratios.

First, we create some artificial data, a 12-by-12 matrix in order to see the performance of this approach. During the experiment, the coefficient $1/\varepsilon_1$ in the equation (1) is tuned to fit the data properly. We change $1/\varepsilon_1$ ranging from 10, 1, 0.1 to 0.01, and the result is below:

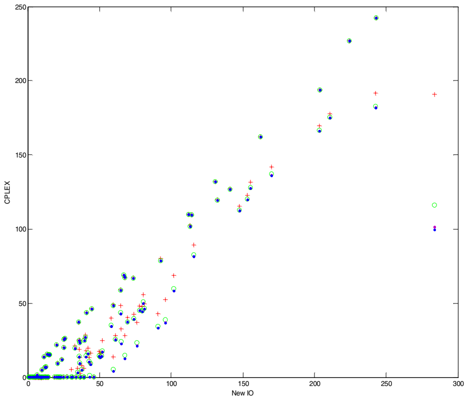

Figure 4. Results with the deviation from 10 to 0.01. The red cross is the result from the derivation 10, the green circle is the result from the derivation 1, the red dot is the results from the deviation 0.1, and the black dot is the result from the deviation 0.01.

Clearly, with the decrease of value of $1/\varepsilon_1$, the resulted matrix is moving away from the matrix of previous year. While $1/\varepsilon_1$ is set smaller, the mining algorithm pushes the model toward the second part of the equation (1).

The multipliers in the input-output framework reflect the impacts of the final demand changes on the upstream industries [1]. The information contained by the multipliers is very similar to the sensitivity analysis in the general statistics. The general formula of constructing the multipliers is:

$$M = D(I - A)^{-1}$$

where M is the multiplier, I is the identity matrix, A is the technique coefficients matrix each element of which $a_{ij} = \dfrac{x_{ij}}{x_{1,j} + x_{2,j} + \ldots + x_{n,j}}$, and D is the change of the final demand.

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, a matrix aims to calculate the total water usage of the different industries in Australia.

Industry	Direct Intensity	Total Multiplier
Sheep and lambs	0.175306192	0.229353737
Wheat	0.179059807	0.27047284
Barley	0.178962619	0.235765397
Beef cattle	0.175528219	0.265691956
Untreated milk & Dairy cattle	1.233958	1.46699422
Pigs	0.177604301	0.273531332
Poultry &	0.288054	0.47927187

Eggs		
Sugar cane	3.664307556	3.720540595
Vegetables & Fruit	0.262444	0.3157365
Ginned cotton	0.000836372	0.297221617
Softwoods	0.175060473	0.236982027
Hardwoods	0.175416808	0.239130312
Forestry	0.175046587	0.229861225
Black coal	0.008572854	0.10262849
Crude oil	0.001043964	0.016968976
Natural gas	0.006772327	0.03029614
LPG, LNG	0	0.035575921

Here a part of the multipliers are used to measure the quality of the resulting matrix. This matrix aims to calculate 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 [6].

At the table above, the direct intensity indicates the direct usage of the water by the industry, and total multipliers indicate the all upstream water usage of the industry. The difference between the direct intensity and total multiplier indicates the upstream consumption. For example, the pig industry has total multiplier 0.273531332. That means each dollar of pork will cost 0.27 litre of water, but the direct usage of water is only 0.177 litre per dollar. The 0.1 litre water is consumed by upstream industries such as some agriculture sections which supply the food for pigs.

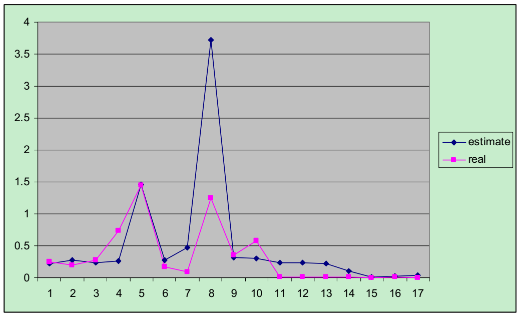

Figure 5. Comparison between two series of multipliers

From the above plot comparing the two series of the multipliers, two series basically follow the same trend, which indicates the industry structure is estimated properly. However the estimated multipliers are more frustrated than the true underlying multipliers, which indicates the estimated multipliers amplifies the impact on the upstream.

IV. Conclusion

This system is an integrated data analysis system for updating a large-scale matrix. The unique characteristics of the data determine the data analysis system must be capable of dealing the temporal and spatial data simultaneously. At the same time, the large size of the estimated matrix requires the system to process a large amount of data efficiently. This paper presents a completed data analysis system starting from data collection to data analysis and quality checking. 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 model. Before this system, this kind of collection and updating work costs months of work, but now it takes only a few days with the consistent quality.

Acknowledgment

Author would like to give thanks to A/Prof. Regina Burachik and Dr. Yalcin Kaya from the School of Mathematics and Statistic, University of South Australia, Australia, and my colleague Prof. Manfred Lenzen, from the Centre for Integrated Sustainability Analysis, University of Sydney, Australia.

References

[1] Miller, R.E. and P.D. Blair, *Input-output Analysis, Foundations and Extensions*. 1985, Englewood Cliffs, New Jersey: Prentice-Hall Inc.

[2] Miller, H.J. and J. Han, *Geographic Data Mining and Knowledge Discovery* 2001, CRC.

[3] Hobbs, L., et al., *Oracle Database 10g Data Warehousing*. 2005: Elsevier Digital Press.

[4] Miller, H.J., *Geographic Data Mining and Knowledge Discovery*, in *The Handbook of Geographic Information Science*, J. Wilson and A.S. Fotheringham, Editors. 2007, Wiley-Blackwell.

[5] Combettes, P.L., *A Block-iterative Surrogate Constraint Splitting Method for Quadratic Signal Recovery*. IEEE Transactions on Signal Processing, 2003. **51**(7): p. 1771- 1782.

[6] *4610.0 - Water Account, Australia*. 2004-05, The Australian Bureau of Statistics: Canberra.