Inverse Optimization for Utility Measurement

Yi-Hau Huang and Yu-Ching Lee†
Department of Industrial Engineering and Engineering Management
National Tsing Hua University, Hsinchu, Taiwan
Email: borrorboy@gmail.com, yclee@ie.nthu.edu.tw

Abstract. Utility function has been prevalent expressing one consumer's preference representing consumer's demand. We assume that one product is composed of many characteristics. Based on this concept, the consumer will rank different products by a unique score toward each characteristic. According to the scores, each customer will get his own utilities corresponding to each product. As a result, the customer is willing to buy the most satisfying product, i.e., the product with the highest utility within one specific market. We establish the pure characteristic demand model for consumer's utility function. We then formulate a mathematical program with quadratic objective function and complementarity constraints as the inverse problem that minimizes the error of the utility measured function. By deriving the weights for the program, we can calculate the consumer's utility with the weights. We use the real vehicle data to prove the validity of the program. Then, we adopt the big data storage and analysis framework to handle the real vehicle data. Finally, our research indicates that the program with complementary constraints will help us find a set of more accurate parameters. The vehicle company may refer to this utility function to estimate customers' willingness to purchase the designed vehicle type.

Keywords: Optimization Techniques

1. INTRODUCTION

An inverse optimization program is about inferring the parameters for a forward optimization program. The easiest inverse optimization model is that for a linear forward optimization program $\min_{\mathbf{x}}\{\ \mathbf{c}^{\mathbf{T}}\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\ \}$ (Ahuja and Orlin 2001). Given a desired target $\mathbf{x}^*$, the inverse problem aims to solve for a vector $\mathbf{c}$ such that the optimal solution of the forward program coincides with $\mathbf{x}^*$. This is equivalent to finding the pair **(p,c)** satisfying strong duality and dual feasibility, namely,

$$\mathbf{p}^{\mathrm{T}}\mathbf{b} = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{*}, \mathbf{p}^{\mathrm{T}}\mathbf{A} = \mathbf{c}^{\mathrm{T}}, \mathbf{p} \geq \mathbf{0} \tag{1}$$

The inverse problem can be formulated as $\min_{\mathbf{c}} ||\mathbf{c} - \hat{\mathbf{c}}||_n$ subject to (1), where n = 1, 2, ∞ are commonly chosen, and $\hat{\mathbf{c}}$ is often set at an incumbent coefficient of the forward program.

In this work we consider on the forward problem---consumer's utility maximization problem:

$$\max_{\pi} \{\ \textstyle\sum_j \pi_j\, (\mathbf{x_j}'\boldsymbol{\beta} - \alpha P_j + \xi_j) \ \mid\ \sum_j \pi_j = 1\ \} \tag{2}$$

The coefficient $(\mathbf{x_j}'\boldsymbol{\beta} - \alpha P_j + \xi_j) \equiv u_j$ is the utility for the consumer buying product j. π_j is the probability of buying product j and is the decision variables in the forward problem. In the basic setting of inverse optimization, we observe π_j^* and look for a value of u_j (which is closed to an incumbent $\hat{u}_j$) so that π_j^* is optimal.

We face at least two challenges to formulate the inverse optimization of this problem that is different from the basic form.

(i) We do not observe π_j^* for every individual in the market. Instead, we observed the aggregated result, the market share of product j, $S_j^* = (1/N)\ \Sigma_i \pi_{ij}$. N is the size of the collection of consumers in a market, and π_{ij} is individual i's probability to buy product j.

(ii) The coefficient u_j we aim to solve for in the inverse optimization is believed to have an intrinsic structure depending on the observed vector of product characteristics $\mathbf{x_j}$ and the price P_j. The objective function $\min_u\ \Sigma_j(u_j - \hat{u}_j)^2$ can be replaced by the objective function $\min\ \Sigma_j (\xi_j)^2$ subject to a sufficient condition of (2). The coefficients we solve for will be $\boldsymbol{\beta}$, α, and ξ_j.

Furthermore, to accommodate the aggregated observation, the coefficients $\boldsymbol{\beta}$ and α are rewritten as the consumer-dependent coefficients $\boldsymbol{\beta_i}$ and α_i.

1.1 Advances of the Inverse Optimization

The research on inverse optimization can be traced back to Berton and Toint (1992) on the inverse shortest path problems. Later on, the inverse optimization model and method for the forward linear program has been extended to integer programming (a comprehensive survey in Heuberger 2004, Shafer 2009), mixed integer programming (Wang 2009), convex programming (Iyengar and Kang 2005, Zhang and Xu 2010), and multiobjective linear optimization (Chan et al 2014).

The formulation of an inverse optimization, however, can be more general. In a recent work (Chan et al 2014), the desired target $\mathbf{x}^*$ is not necessarily an optimal solution to the forward problem. This issue arises naturally when the observed $\mathbf{x}^*$ renders the inverse problem infeasible. A generalization for a linear forward problem is achieved by replacing the strong duality $\mathbf{p}^T\mathbf{b} = \mathbf{c}^T\mathbf{x}^*$ with one of the following two alternatives: (i) $\mathbf{c}^T\mathbf{x}^* = \varepsilon_r\mathbf{p}^T\mathbf{b}$ and (ii) $\mathbf{c}^T\mathbf{x}^* = \mathbf{p}^T\mathbf{b}+\varepsilon_a$. The variables ε_r and ε_a, which should be minimized in the inverse optimization, are interpreted as the relative duality gap and the absolute duality gap respectively.

Another important class of the inverse optimization is for the convex nonlinear forward problem. Three typical applications, as enumerated in Zhang and Xu (2010) are (i) quality control in production systems, (ii) portfolio optimization, and (iii) production capacity planning. Assuming a concave demand function $\mathbf{F}(\mathbf{x})$ with respect to the quality level $\mathbf{x}$ and a convex working-hours function $\mathbf{g}(\mathbf{x})$ with respect to $\mathbf{x}$, the forward profit maximization with an upper bound in the total working hours for maintaining the quality level is generally formulated as

$$\max_{\mathbf{x}}\left\{ \mathbf{c}^T\mathbf{F}(\mathbf{x}) \mid \mathbf{b}^T\mathbf{g}(\mathbf{x}) \le D,\ \ell \le \mathbf{x} \le u \right\}.$$

Given an optimal quality level that minimizes the total working hours and meets the required total profit, the inverse optimization problem is about pricing the unit cost b; in portfolio optimization, the proportions x to be allocated in each asset is obtained by solving the Markowitz's quadratic constrained program

$$\max_{\mathbf{x}}\left\{ \boldsymbol{\mu}^T\mathbf{x} \mid \mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x} \le \sigma^2,\ \mathbf{A}\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0} \right\},$$

where $\boldsymbol{\mu}$ is the vector of expected return, $\boldsymbol{\Sigma}$ is the covariance matrix of return, and σ^2 is an acceptable level of risk. Let the optimal solution be $\mathbf{x}^*$ (an efficient portfolio), the inverse optimization seeks a value of $\boldsymbol{\mu}$ that minimizes the weighted norm $||\boldsymbol{\mu}-\bar{\boldsymbol{\mu}}||^2_W = (\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T\mathbf{W}(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})$ with a positive definite matrix $\mathbf{W}$ and an expected return vector $\boldsymbol{\mu}$ of the latest information such that the portfolio $\mathbf{x}^*$ remains efficient; in capacity planning, the capacity of each workstation is often re-distributed to optimize the performance of a particular metric. For n workstations, a cost minimization forward problem is of the form

$$\min_{x_i}\left\{ \sum_{i=1}^{n} c_i f_i(x_i) \mid \sum_{i=1}^{n} x_i = b,\ x_i \ge u_i,\ \forall i \right\},$$

where c_i is the average value of work-in-process with each job at station i, $f_i(x_i)$ is the average numbers of jobs with respect to the capacity x_i at station i, b is the total capacity, and u_i is the rate of jobs arriving at station i. The inverse problem updates the total capacity b dynamically according to the previous distribution of capacity $\hat{x}_i$. Solving alternatively for the forward and inverse optimization is essentially the philosophy of just-in-time scheduling.

1.2 MPEC

For the convex forward program, the Karush-Kuhn-Tucker (sufficient) optimal condition becomes critical in analyzing the properties and designing the solution techniques. The Karush-Kuhn-Tucker condition consists of primal feasibility, dual feasibility, and complementarity slackness. Hence, some inverse optimization problems can be properly reformulated as the linear complementarity problems (without outer objective) and the mathematical program with complementarity constraints (with an outer objective). The general form of a mathematical program with complementarity constraints (MPCC, or MPEC, Luo et al. 1996) is as follows:

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{y}) \\ \text{subject to} \quad & g_{\mathrm{E}i}(\mathbf{y}) = 0, \quad \forall i \in \mathrm{I_E}, \\ & g_{\mathrm{I}i}(\mathbf{y}) \ge 0, \quad \forall i \in \mathrm{I_I}, \\ \text{and} \quad & 0 \le y_j \perp h_j(\mathbf{y}) \ge 0, \quad \forall j \in \mathrm{I_C}, \end{aligned} \tag{3}$$

where $y_j \perp h_j$ denotes the complementarity $y_j h_j = 0$ and $\mathrm{I_E}, \mathrm{I_I}, \mathrm{I_C}$ denotes the index sets. If $g_{\mathrm{E}}, g_{\mathrm{I}}$, and h are linear functions, the constraint set is named as the linear complementarity. The satisfiability problem $0 \le \mathbf{y} \perp h(\mathbf{y}) \ge 0$ itself is called the linear complementarity problem (LCP, Cottle et al. 2009). If $\mathbf{y}$ solves $0 \le \mathbf{y} \perp h(\mathbf{y}) \ge 0$, $\mathbf{y} \ge 0$ also solves the affine variational inequality (Facchinei and Pang 2003)

$$(\boldsymbol{\Phi} - \mathbf{y})^T h(\mathbf{y}) \ge 0, \forall \boldsymbol{\Phi} \ge 0,$$

and the optimization problem

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{y}^T h(\mathbf{y}) \\ \text{subject to} \quad & \mathbf{y} \ge 0, h(\mathbf{y}) \ge 0. \end{aligned} \tag{4}$$

If the complementarities in (3) are replaced by (4), the resulting formulation becomes a nonlinear bi-level program (Dempe 2002), a class of the hierarchical programming. In the case where $h(\mathbf{y})$ is the gradient of another function $q(\mathbf{y})$, i.e., $h(\mathbf{y}) = q(\mathbf{y})$, problem (4) is equivalent to $\min_{\mathbf{y}\ge 0} q(\mathbf{y})$.

1.3 Prospective Applications of Inverse Optimization

The inverse optimization has been studied in the parameter estimation for the earthquake data (Tarantola 2005), demand (Carr and Lovejoy 2000), auctions (Beil and Wein

2003), finance (Bertsimas et al. 2012), consumers' choice and firms' pricing (Pang et al. 2015) and cancer therapy (Chan et al. 2014).

The scale of the input data of inverse optimization in the literature remains small and experimental. In this paper we construct the inverse optimization methods approach on the big-data storage, database, and query technology stack. For further explanation, an overview of a 4-layer Real-Time Big Data Analytics (RTBDA) technology stack proposed by David Smith is displayed in Figure 1

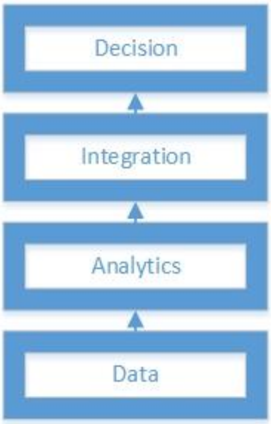

Figure 1: David Smith's 4-layer of RTBDA technology stack

The core technology in the data layer is about the storage techniques and the query processing. For examples, RDBMS, Hbase, and Impala are systems for structured data storage; Hadoop MapReduce is a software framework for unstructured vast amounts of data; and Spark is a cluster big-data computing framework that supports SQL, streaming processing, machine learning, graph, and R. The data can be obtained from data warehouse appliances, or it can be the streaming data from the sensors, real-time data center, and websites. The analytics layer involves data mart which updates from the data layer constantly and the development environment for model constructing. The integration layer is like a broker, a business rule engine, and sometimes an Application Programming Interface (API). The decision layer is the end-users applications.

This research is aims to further the analytical technology that belongs to the analytics and the integration layers as a basis for the efficient use of data in terms of robust forecasting and optimal decision making. I propose a Data-to-Value technology stack in the Figure 2.

The inverse optimization layer is the main method we will design for the parameter estimation; the algorithm layer contains the development of the parallel programming and distributed algorithm to accommodate both the models of inverse and forward optimization; and the forward layer computes the decision variable and the forecast quantities.

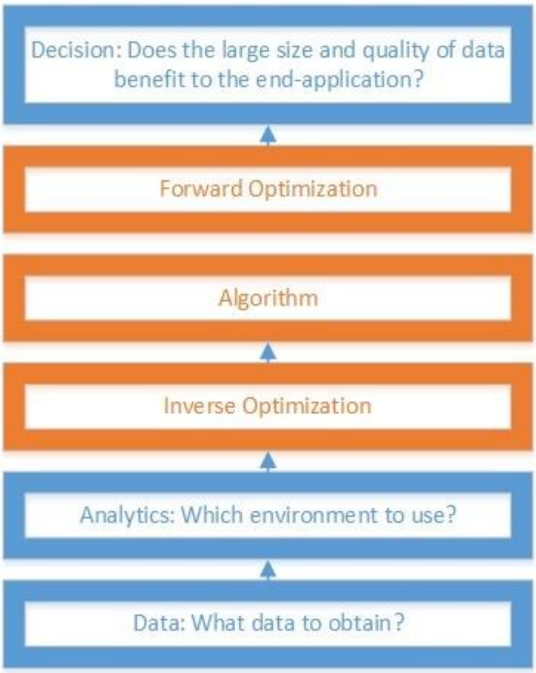

Figure 2: Data-to-Value technology stack.

The data, analytics, and decision layer stays at the same position as in the Figure 1, and additional issues need to be addressed sequentially: specific columns of data to be obtained in the base layer, compatible developer's environments where the optimization techniques perform, and the feedback to scale and quality of the data according to the inverse/forward optimization study.

2. Utility Measurement with Inverse Optimization Modeling

Following the description in Section 1, the inverse problem is to find the parameters α_i and β_i such that consumer i' decision π_{ij} is reflected in an aggregated observation. Denote J the total numbers of products. The MPEC formulation for inverse optimization is as follows:

$$
\begin{aligned}
&\min_{\alpha,\beta,\xi,\pi,\gamma} && \xi'\xi \\
&\textit{subject to} && \frac{1}{N}\sum_{i=1}^{N}\pi_{ij} = S_j, \quad \forall j, \\
&\left.\begin{matrix}\textit{for all} & i = 1,\cdots,N \\ \textit{and} & j = 1,\ldots,J\end{matrix}\right\} && 0 \le \pi_{ij} \perp \gamma_i - \left(\mathbf{x}_j'\boldsymbol{\beta}_i - \alpha_i p_j + \xi_j\right) \ge 0 \\
&\forall i: && 0 \le \gamma_i \perp 1 - \sum_{j=1}^{J}\pi_{ij} \ge 0
\end{aligned}
\qquad (5)
$$

Since the observations on the choices of every individual consumer are not directly made, we need the first constraint to relate the estimated π_{ij} with the observed market share, S_j. ξ_j within the utility expression is interpreted as the only unobserved characteristic of product j. In the inverse optimization framework, we may view it as the product-wise ξ-tolerance to fit the structural utility with the reality. That is,

$$u_{ij} - \hat{u}_{ij} = (\mathbf{x}_j'\hat{\boldsymbol{\beta}}_i - \hat{\alpha}_i P_j + \xi_j) - (\mathbf{x}_j'\hat{\boldsymbol{\beta}}_i - \hat{\alpha}_i P_j) = \xi_j$$

3. Measuring Consumer Utility in the Car Market

The validation of the inverse optimization model (5) is done for a car market. We employ the open nonlinear programming (NLP) package for R and solve the small instance of the model (5) on one single machine.

3.1 Car market description

The sales data in the car market is from the UK government for transport statistics. The cars in the data registered for the first time by its generic model. The data is recorded annually from 2001 to 2015-Sep-30. Part of the data is shown in Figure 3. To obtain a better understanding of the car market, it is necessary to sum the sales for each brand in Figure 4.

From the Figure 4, the top 10 best sellers with the sales for this car market are Ford (4,948,838), Vauxhall (4,148,409), Volkswagen (2,754,599), Peugeot (1,993,282), Renault (1,712,299), BMW (1,686,298), Toyota (1,578,508), AUDI (1,489,349), Nissan (1,443,920), Citroen (1,345,245).

3.2 NLP package for R

The R Rsolnp package includes several documents (functions) such as benchmark, benchmarkids, gosolnp, solnp, startpars. The Rsolnp package is the solver we used for this study. The solnp function is based on the method developed by Ye (1987) which solves the general nonlinear programming problem. The solver belongs to the class of indirect solvers and implements the augmented Lagrange multiplier method with an SQP interior algorithm. The main reason to select this function as our method is the high efficiency feature of the SQP algorithm for solving the nonlinear programming problems.

3.3 Result on Single Machine

The numbers of rows and columns for the data are 212 and 23 respectively. Supposedly, there are 5 people in the car market. Therefore, the value for i is 1,…,5 and the value for j is 1,…212. The dimension of $\boldsymbol{\beta}_i$ is 23, which is the number of the vehicle characteristics. The results for ξ_j, π_{ij}, $\boldsymbol{\beta}_i$, α_i, and γ_i shows in Figure 5, Figure 6, Table 1, Table 2, and Table 3 respectively.

4. ON-GOING WORK

An algorithm that solves the inverse optimization for a larger scale of problems on a cluster of machines is under construction.

ACKNOWLEDGMENTS

The authors thank the financial support from MOST 104-2218-E-007-025-, Republic of China.

REFERENCES

Ahuja, R. K., Orlin, J. B. (2001). Inverse optimization. *Operations Research*. **49**(5):771-783.

Aswani, A, Shen, Z-J. M., Siddiq, A. (2015). Inverse Optimization with noisy data. manuscript.

Beil, D. R., Wein, L. M. (2003). An inverse-optimization-based mechanism to support a multiattribute RFQ process. *Management Science*. **49**(11):1529-1545.

Bertsimas, D., Gupta, V., Paschalidis, I. C. (2012). Inverse optimization: A new perspective on the Black-Litterman model. *Operations Research*. **60**(6):1389-1403.

Burton, D., Toint, P. L. (1992). On an instance of the inverse shortest paths problem. *Mathematical Programming*. **53**(1):45-61.

Carr, S., Lovejoy, W. (2000). The inverse newsvendor problem: Choosing an optimal demand portfolio for apacitated resources. *Management Science*. **46**(7):921-927.

Cevher, V., Becker, S., Schmidt, M. (2014). Convex optimization for big data. manuscript.

Chan, T. C., Craig, T., Lee, T., Sharpe, M. B. (2014). Generalized Inverse Multiobjective Optimization with Application to Cancer Therapy. *Operations Research*. **62**(3):680-695.

Cottle, R. W., Pang, J-S., and Stone, R. E. (2009). *The Linear Complementarity Problem. 2nd Edition.* Society for Industrial and Applied Mathematics, Philadephia.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Kluwer.

Facchinei, F. and Pang, J-S. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume I and II.* Springer-Verlag New York, Inc.

Heuberger, C. (2004). Inverse combinatorial optimization: A on problems, methods, and results. *Journal of Combinatorial Optimization*. **8**(3):329-361.

Iyengar, G., Kang, W. (2005). Inverse conic programming with applications. *Operations Research Letters.* **33**(3):319-330.

Luo, Z-Q, Pang, J-S, Ralph, D. (1996). *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge.

Pang, J-S., Su, C-L., Lee, Y-C. (2015). A constructive approach to estimating pure characteristics demand models with pricing. *Operations Research*. **63**(3):639-659.

Shafer, A. J. (2009). Inverse integer programming. *Optimization Letters*. **3**(4):483-489.

Tarantola, A. (2005). *Inverse Problem theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia.

Wang, L. (2009). Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations Research Letters*. **37**(2):114-116.

Yinyu Ye (1987). Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming. Ph.D. Thesis, Department of EES, Stanford University.

Zhang, J., Xu, C. (2010). Inverse optimization for linearly constrained conex separable programming problems. *EURO Journal of Operations Research*. **200**(3):671-679.

Generic Model	2015 to 30 Sep	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005	2004	2003	2002	2001
ABARTH 500	731	851	613	909	1079	1249	1089								
ABARTH 595	1197	628	583	206											
ABARTH 695	14	16	2												
ABARTH Model Missing	2	2	7	2	16	2									
ABARTH PUNTO		62	84	97	149	128	234	100							
AC Model Missing	10	6	8	10	11	14	9	10	11	10	8	13	17	10	4
ACCESS Model Missing		2					2								
ACURA CL											1	1	1		
ACURA Model Missing	14	15	16	13	19	21	20	11	9	20	16	22	6	17	
ACURA RL												1			
ACURA TL	1	1										1	1		
ADLY ATV 320							1								
AEON COBRA												3			
AEON OVERLAND												3			
AIE QUADZILLA										1					
AIXAM 400															1
AIXAM 500											61	83	52	62	56
AIXAM A751						3	16	39	81	163	107				
AIXAM COUPE	4	8	10												
AIXAM CROSSLINE	18	15	34	50	99	67	101	84	95						
AIXAM CROSSOVER	44	51	44	30											
AIXAM MAC 500		1									1		1		
AIXAM MEGA				1	2	6	14	63	84			1			
AIXAM Model Missing					2	1			1		2	65	133	59	82
AIXAM SCOUTY					1	1		4	3						

Figure 3: The vehicle sales data

Table 1: The values of $\boldsymbol{\beta}_i$

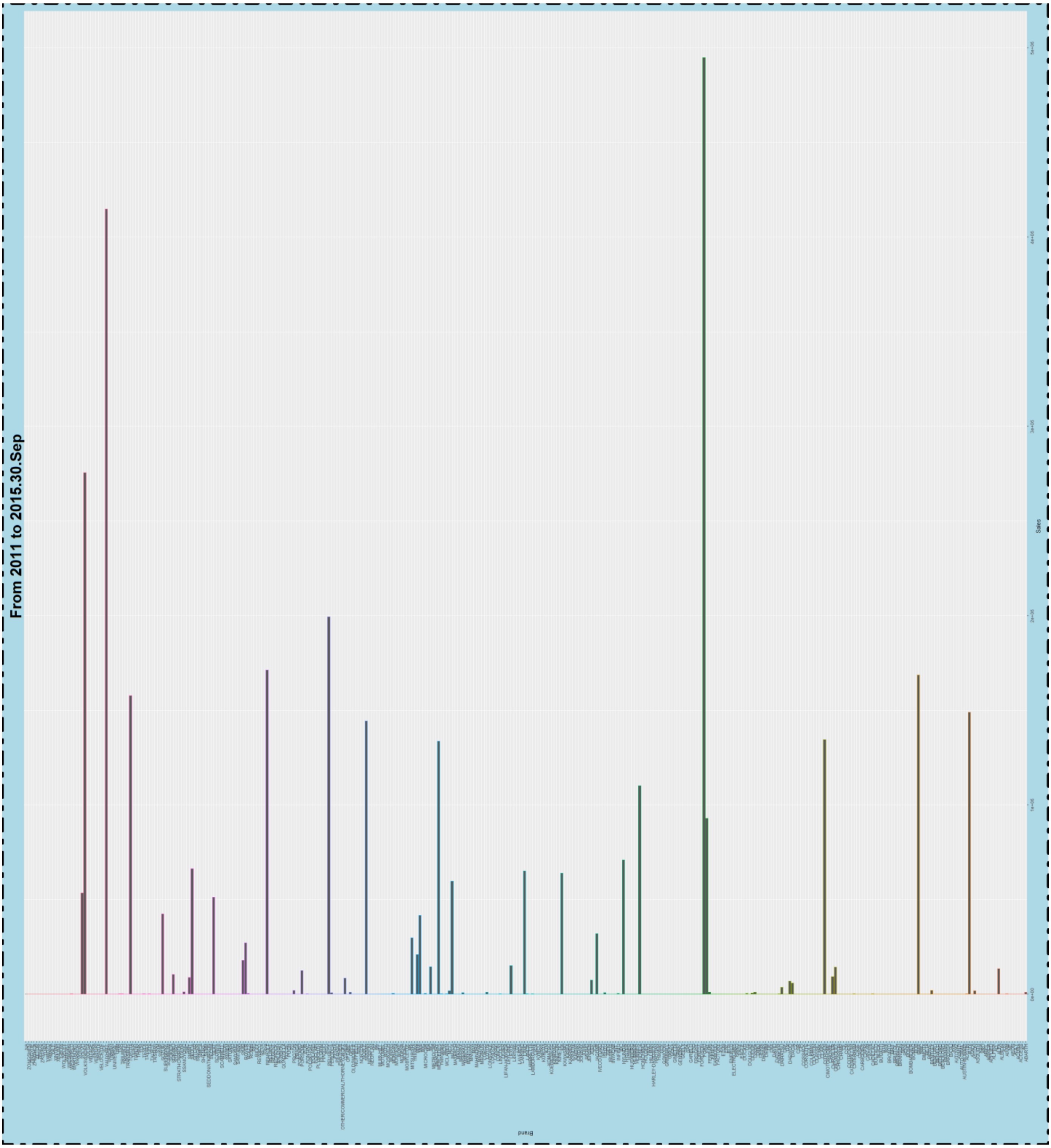

Figure 4: Total sales for each brand

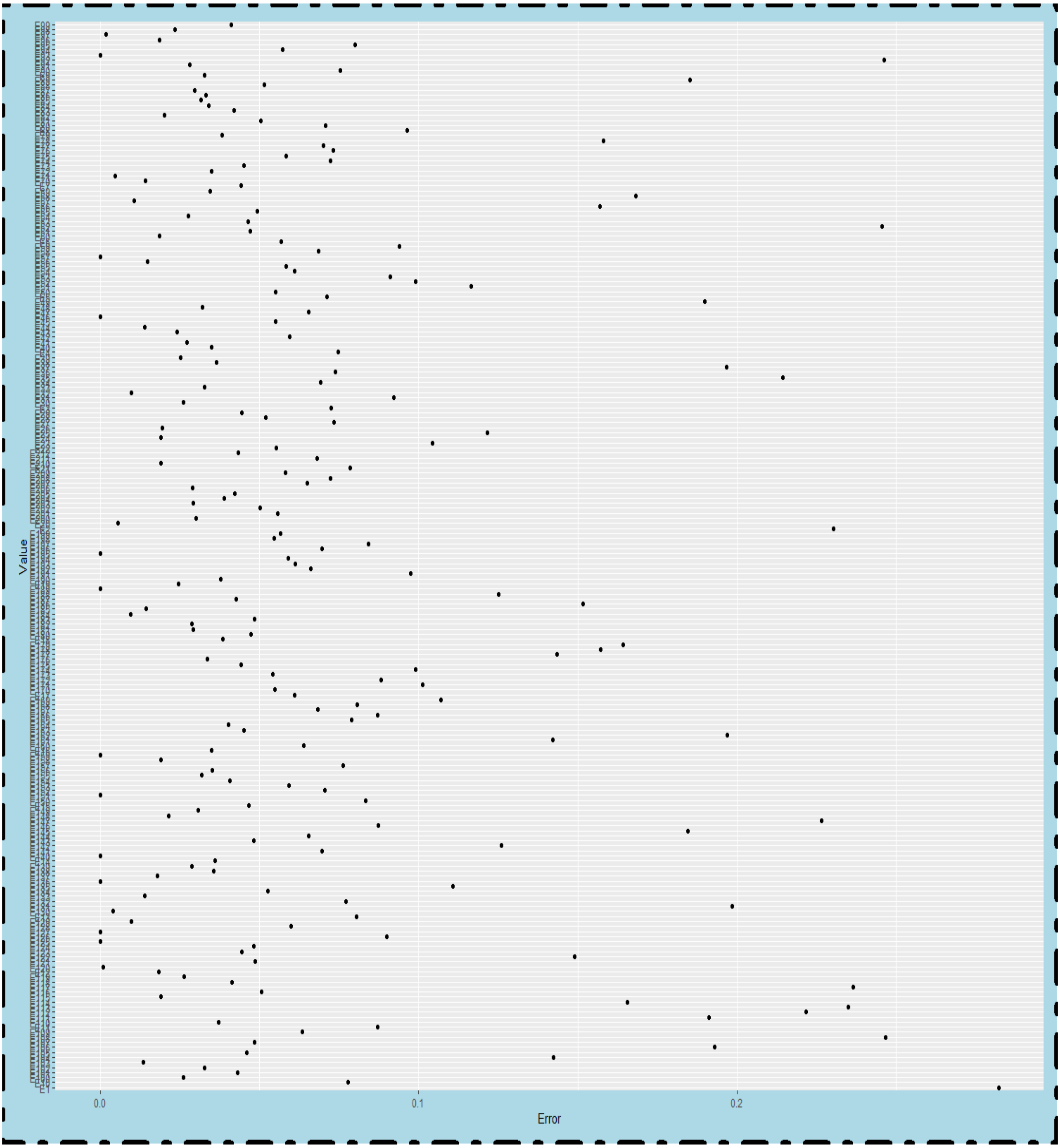

Figure 5: The values of ξ_j

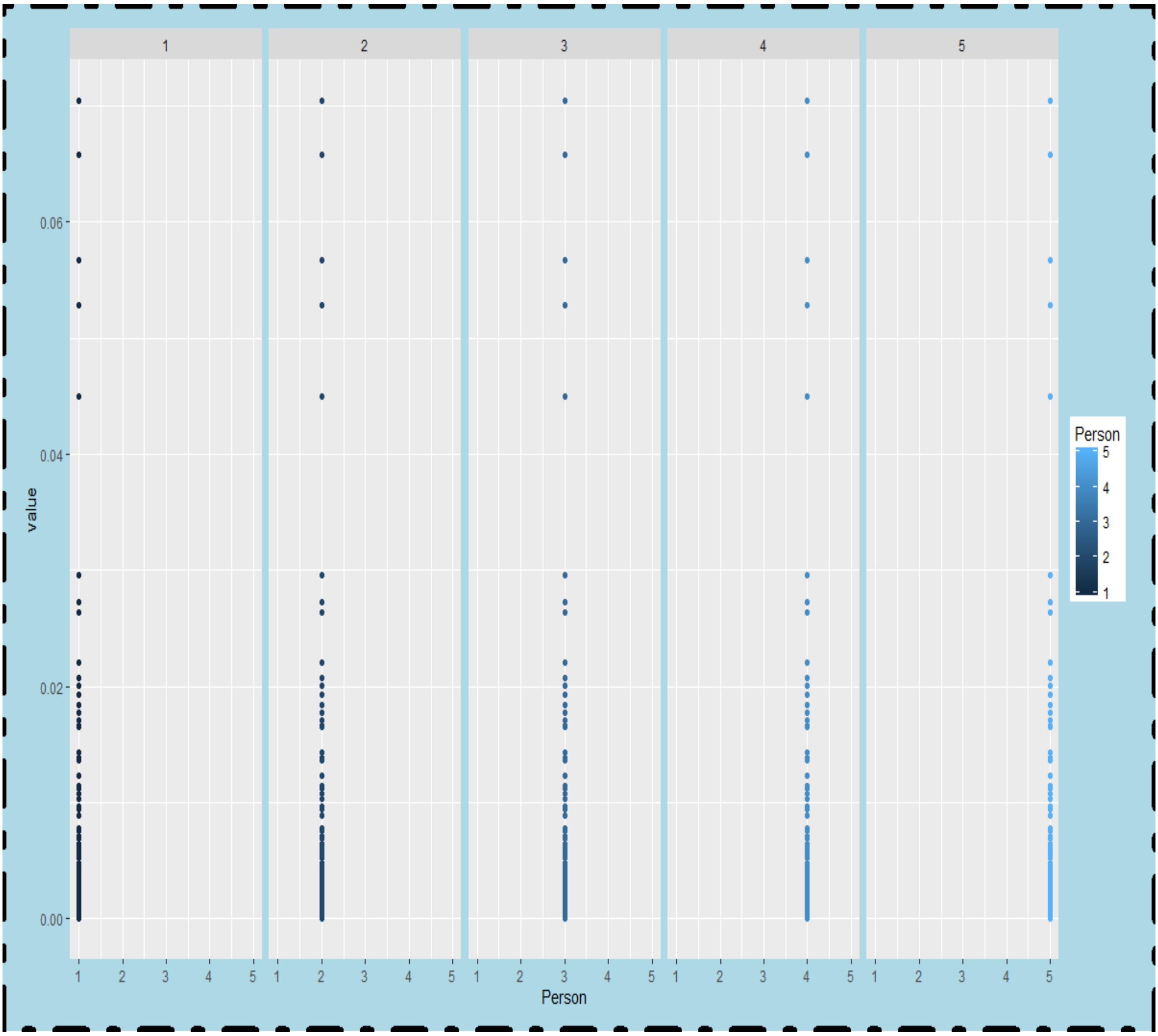

Figure 6: The values of π_{ij}

Table 2: The value of α_i

	Person 1	Person 2	Person 3	Person 4	Person 5
1	0.02026456	0.02029453	0.02031857	0.02029184	0.02032344

Table 3: The value of γ_i

	Person 1	Person 2	Person 3	Person 4	Person 5
1	0.9989019	0.9988701	1.001434	0.9996846	1.001108