



(19) **United States**

(12) **Patent Application Publication**  
**Wagner**

(10) **Pub. No.: US 2012/0259676 A1**

(43) **Pub. Date: Oct. 11, 2012**

(54) **METHODS AND APPARATUS TO MODEL CONSUMER CHOICE SOURCING**

(52) **U.S. Cl. .... 705/7.32**

(57) **ABSTRACT**

(76) **Inventor: John G. Wagner, Cincinnati, OH (US)**

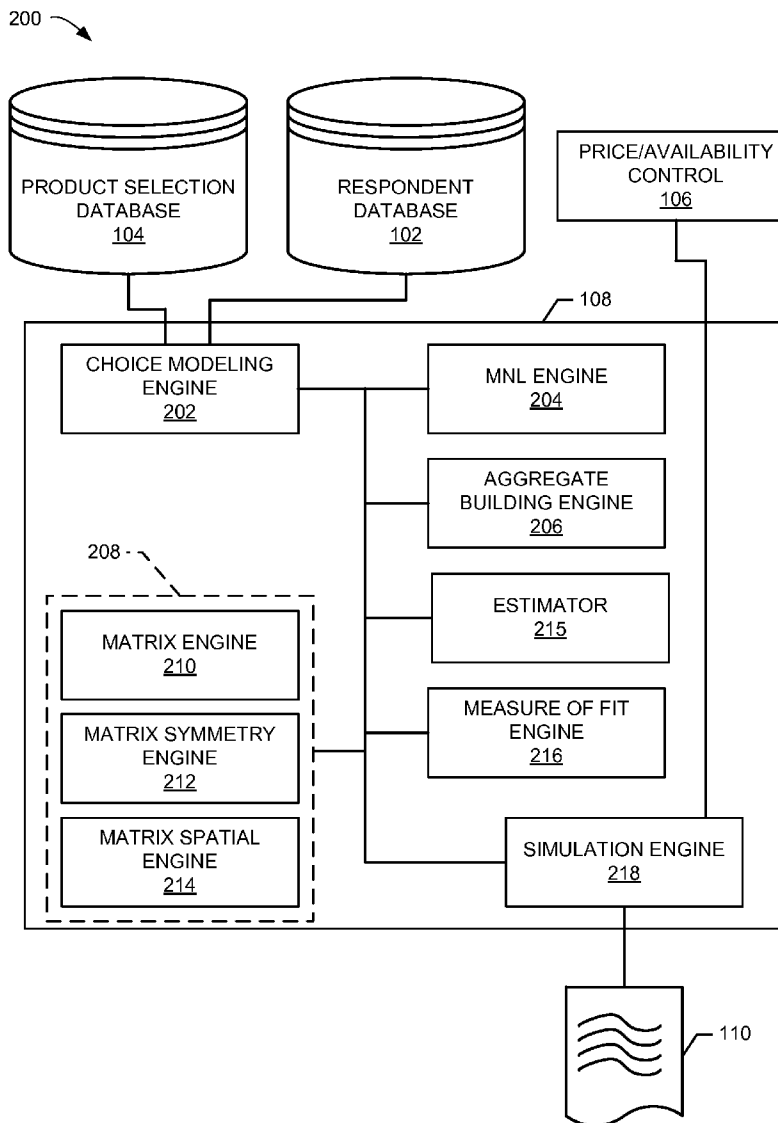
(21) **Appl. No.: 13/081,924**

(22) **Filed: Apr. 7, 2011**

Methods and apparatus are disclosed to model consumer choices. An example method includes identifying a set of products, receiving respondent choice data associated with the set of products, and adding the set of products to a base multinomial logit (MNL) model. The example method also includes generating, with a programmed processor, a number of copies of the MNL model to form an aggregate model based on a number of products in the set, each copy including an item utility parameter for each product in the set of products, and creating a matrix structure based on the number of products in the set, the matrix structure to be subtracted from each item utility parameter in the aggregate model. Further, the example method includes estimating each item utility parameter of the aggregate model and the matrix structure based on the number of copies of the MNL model and the respondent choice data, and calculating a choice probability based on each of the estimated utility parameters.

**Publication Classification**

(51) **Int. Cl. G06Q 10/00 (2006.01)**



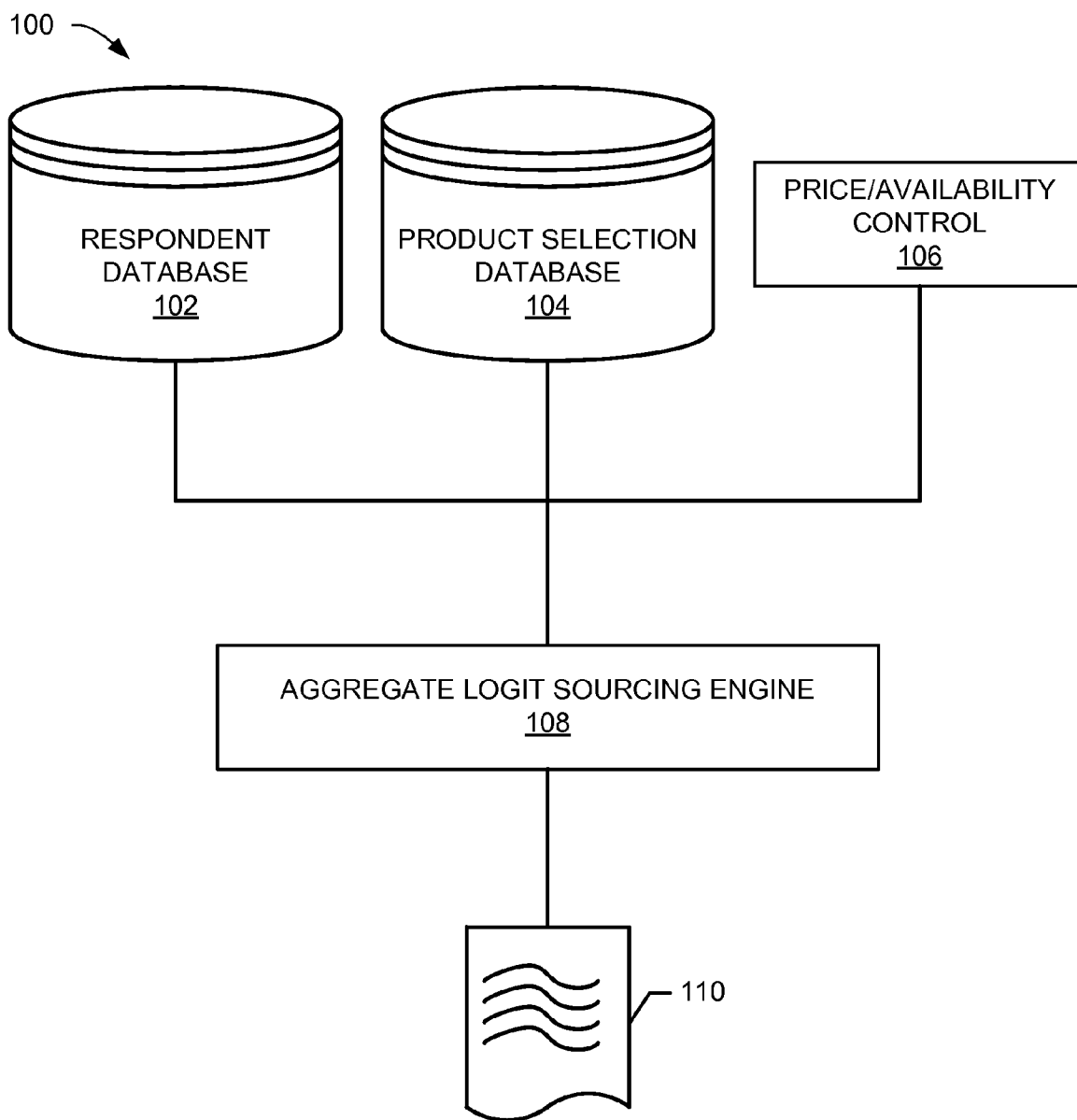


FIG. 1

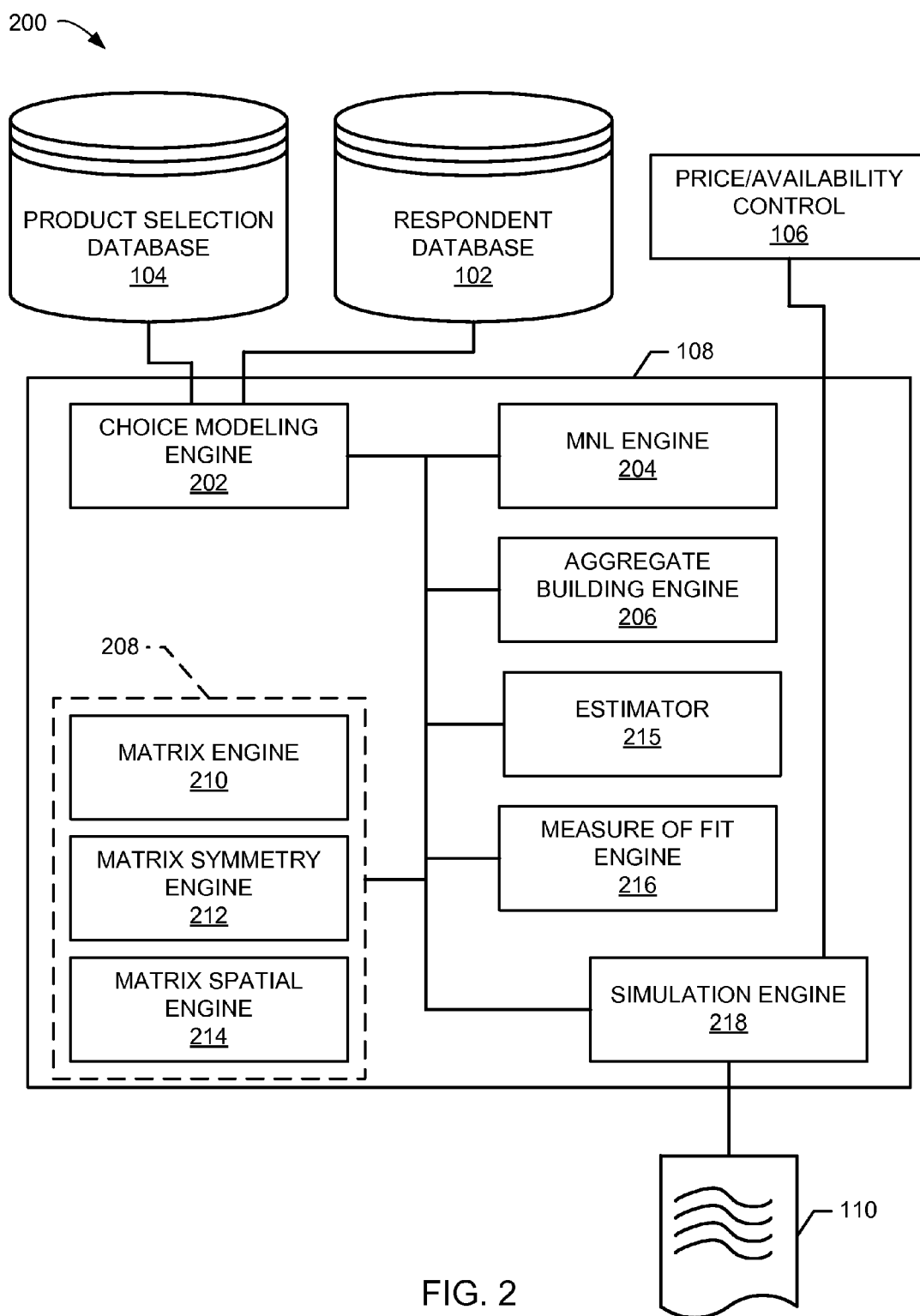


FIG. 2

300

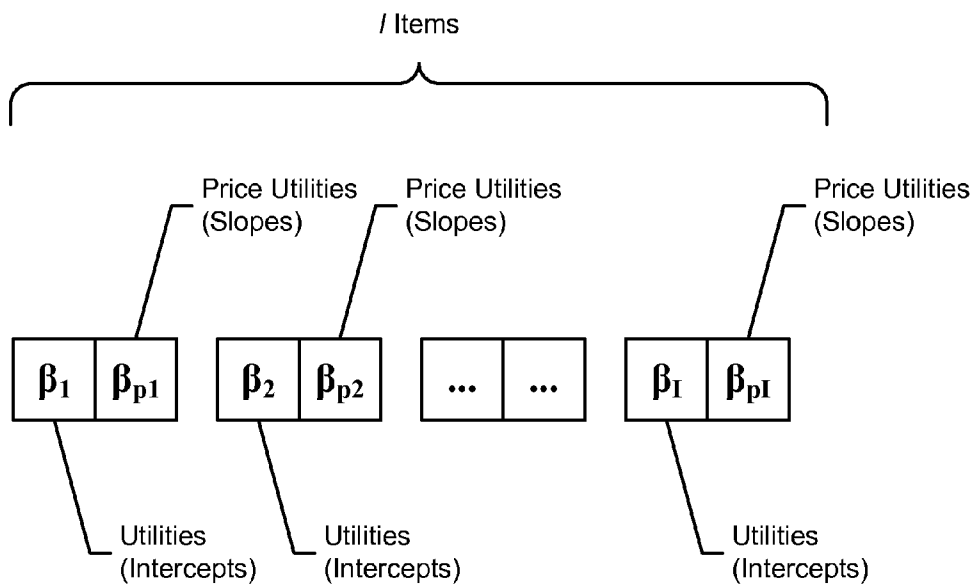


FIG. 3

400

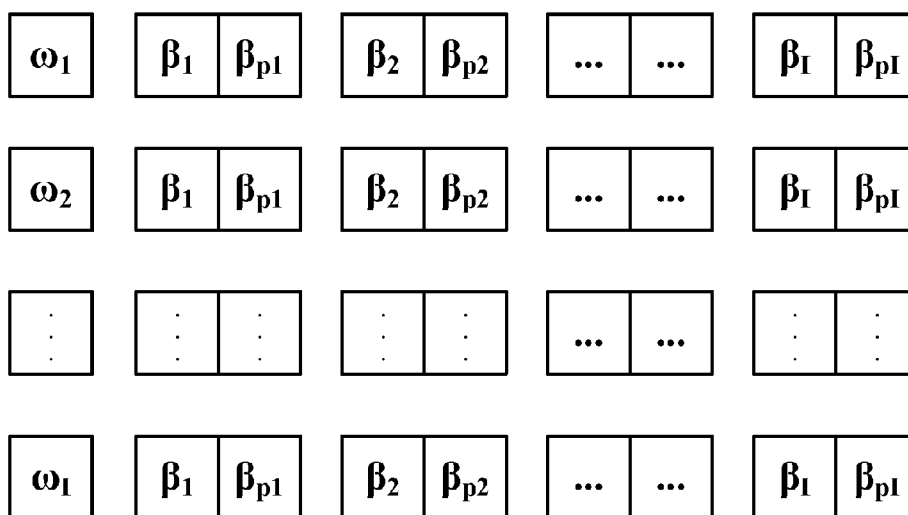


FIG. 4

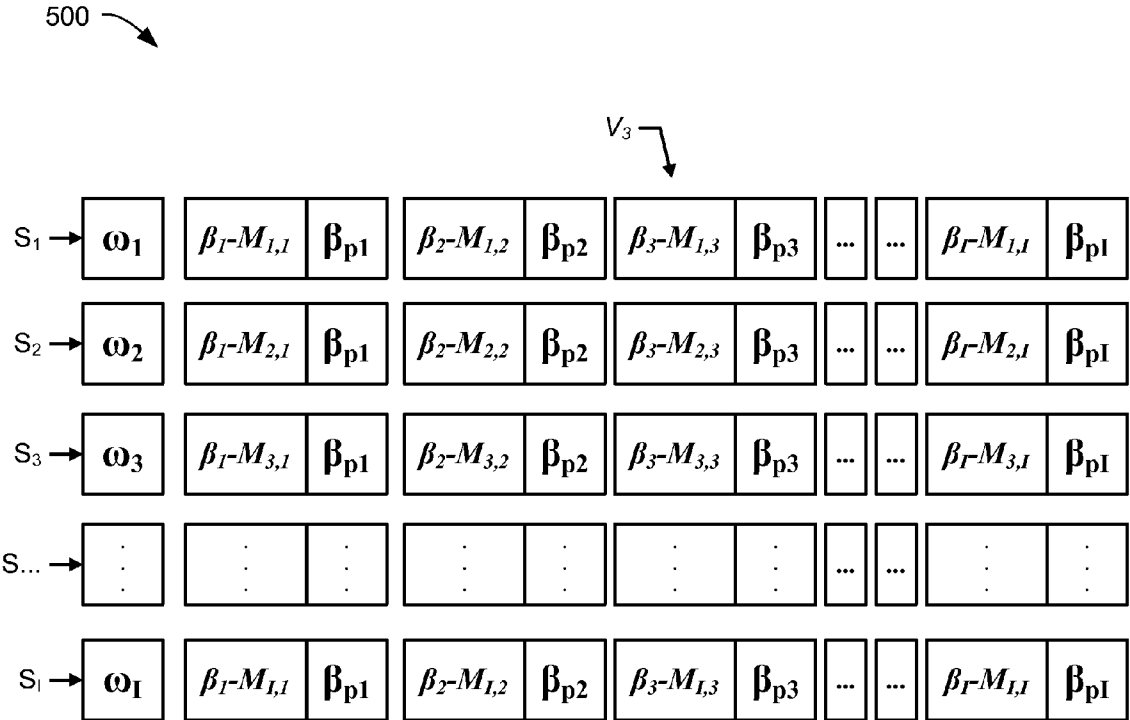


FIG. 5

600

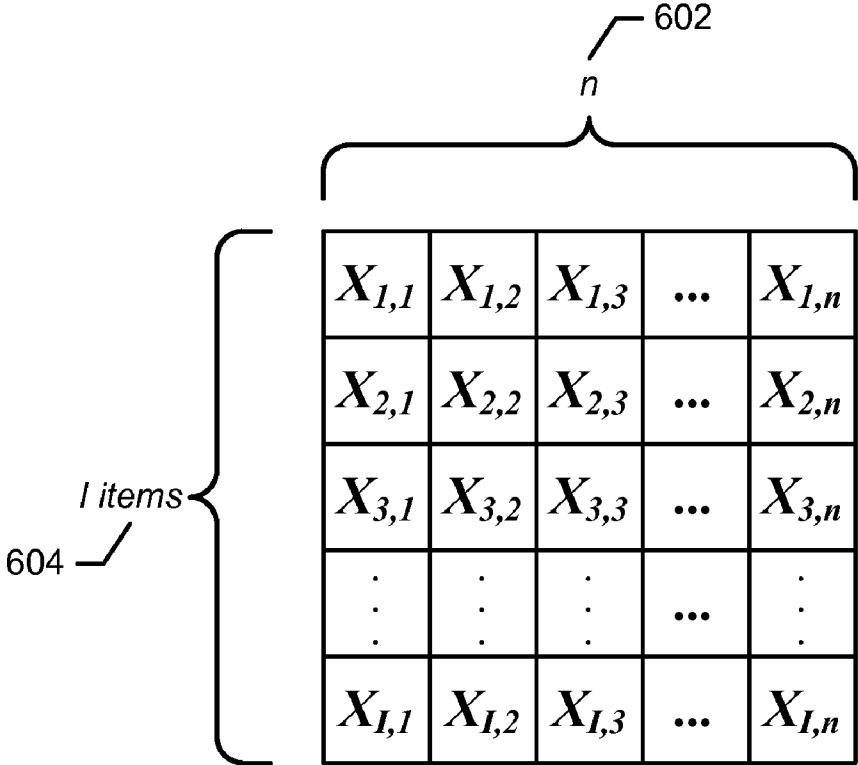


FIG. 6

700 →

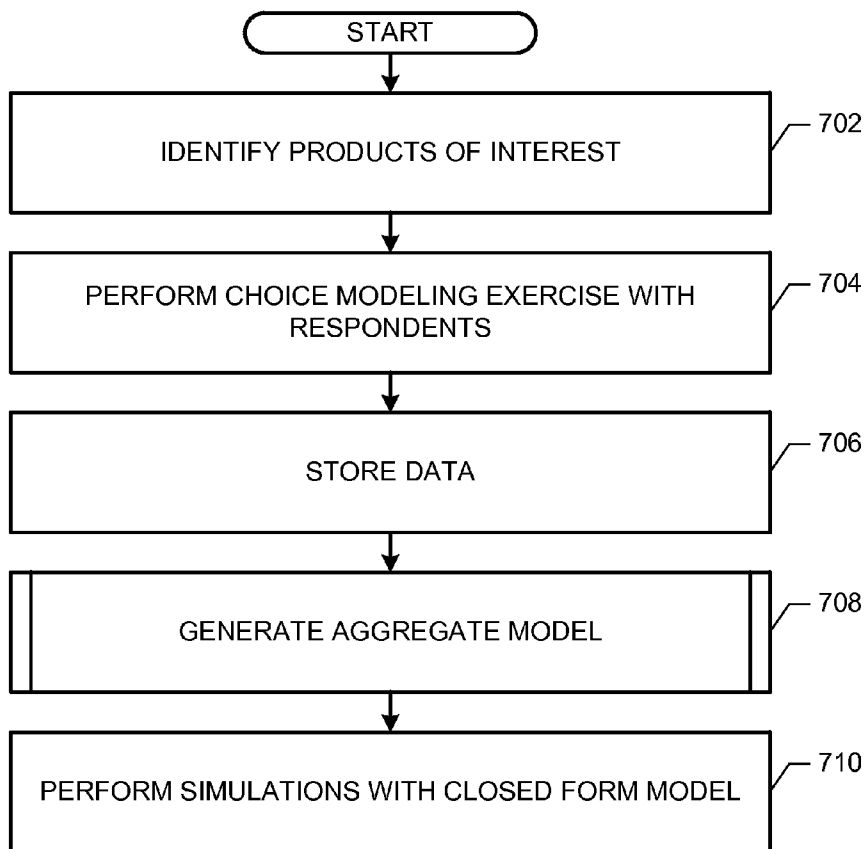


FIG. 7



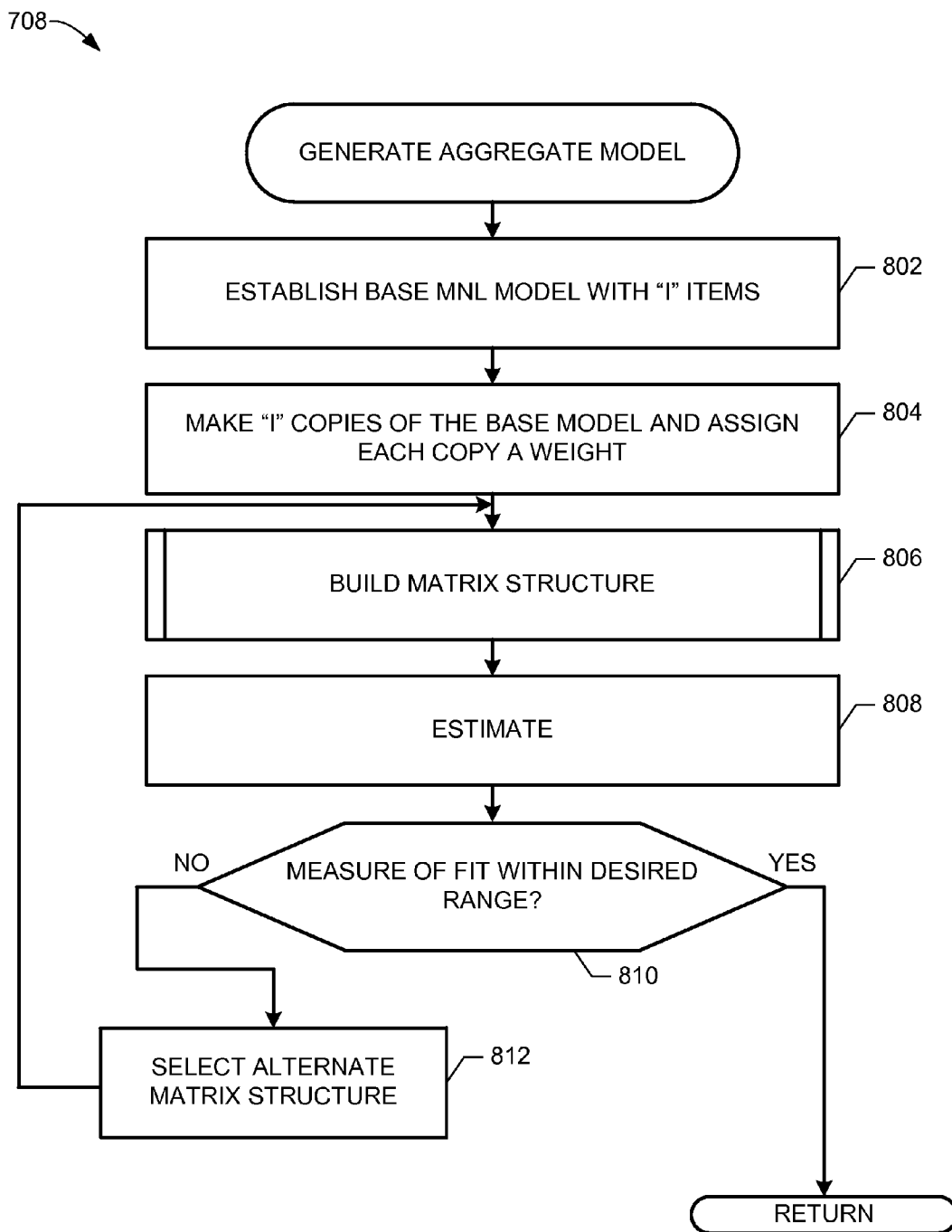


FIG. 8

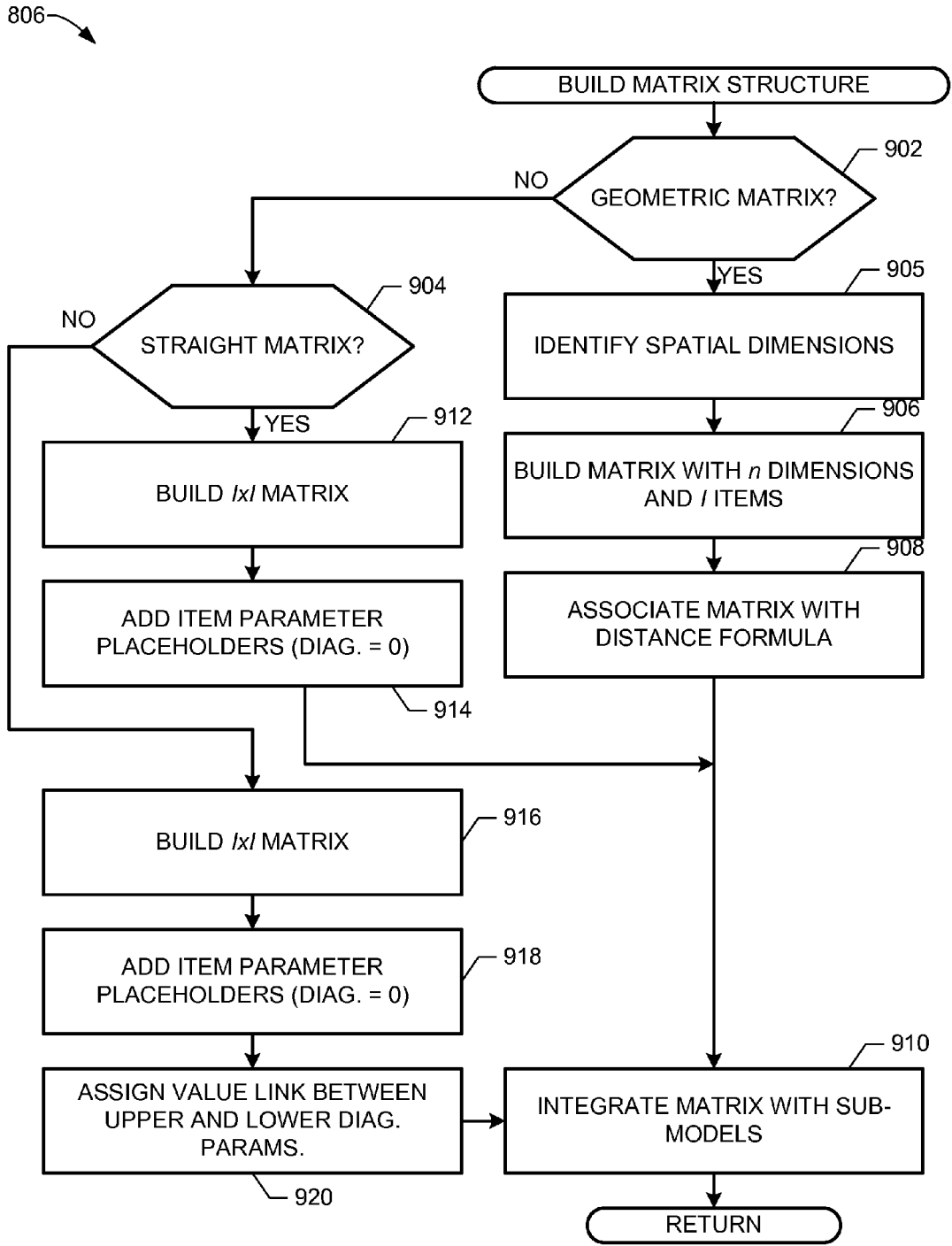


FIG. 9

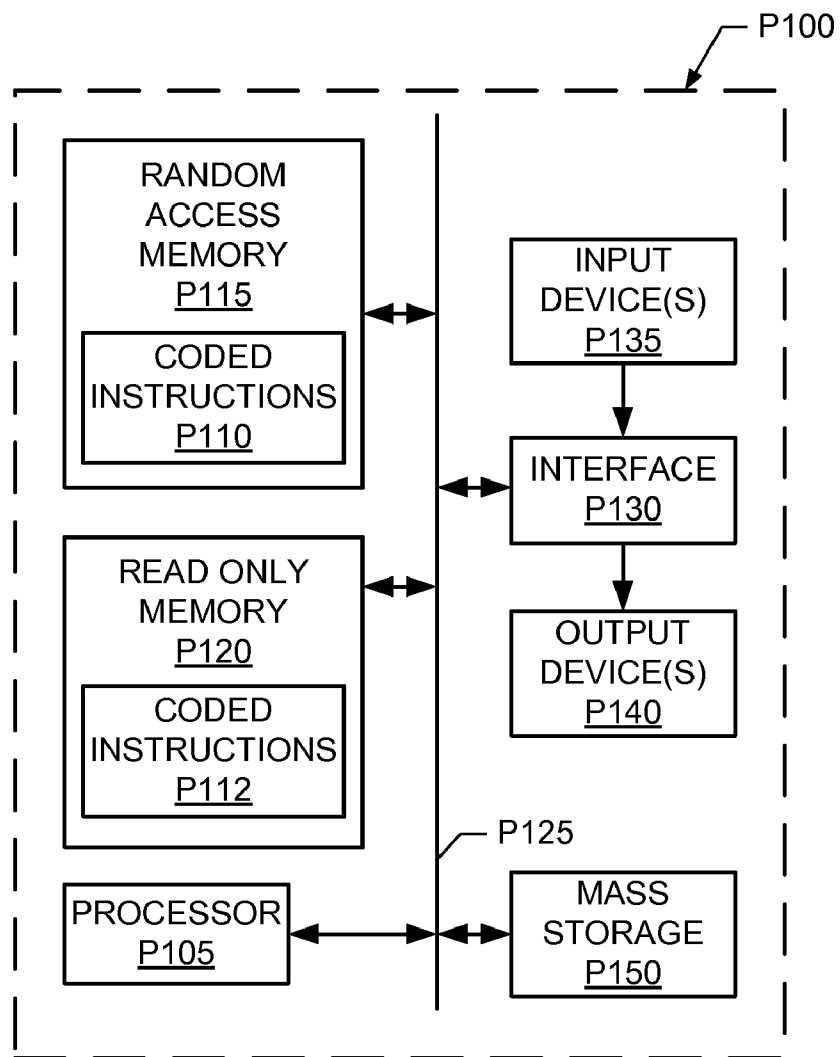


FIG. 10

**METHODS AND APPARATUS TO MODEL CONSUMER CHOICE SOURCING**

**FIELD OF THE DISCLOSURE**

**[0001]** This disclosure relates generally to market research and, more particularly, to methods and apparatus to model consumer choice sourcing.

**BACKGROUND**

**[0002]** Choice modeling techniques allow market researchers to assess consumer behavior based on one or more stimuli. Consumer preference data is collected during the one or more stimuli, such as a virtual shopping trip in which consumers are presented with any number of selectable products (e.g., presented via a kiosk, computer screen, slides, etc.). The consumer preferences associated with products may be referred to as utilities, which may be the result of one or more attributes of the product. While choice modeling allows market researchers to predict how one or more consumers will respond to the stimuli, such analysis techniques typically assume that each item in a virtual shopping trip is equally substitutable relative to all other items available to the consumer.

**BRIEF DESCRIPTION OF THE DRAWINGS**

- [0003]** FIG. 1 is a schematic illustration of an example system to model consumer choice sourcing.
- [0004]** FIG. 2 is a schematic illustration of an example aggregate logit sourcing engine of the example system of FIG. 1.
- [0005]** FIG. 3 is an example multinomial logit model generated by the example aggregate logit sourcing engine of FIGS. 1 and 2.
- [0006]** FIG. 4 is a portion of an example aggregate model structure generated by the example aggregate logit sourcing engine of FIGS. 1 and 2.
- [0007]** FIG. 5 is a portion of an example matrix infused aggregate model structure generated by the example aggregate logit sourcing engine of FIGS. 1 and 2.
- [0008]** FIG. 6 is a portion of an example geometric matrix generated by the example aggregate logit sourcing engine of FIGS. 1 and 2.
- [0009]** FIGS. 7-9 are flowcharts representative of example machine readable instructions that may be executed to implement the example system shown in FIGS. 1 and 2.
- [0010]** FIG. 10 is a schematic illustration of an example processor platform that may execute the instructions of FIGS. 7-9 to implement any or all of the example methods, systems, and apparatus described herein.

**DETAILED DESCRIPTION**

**[0011]** Methods and apparatus are disclosed to model consumer choices. An example method includes identifying a set of products, receiving respondent choice data associated with the set of products, and adding the set of products to a base multinomial logit (MNL) model. The example method also includes generating, with a programmed processor, a number of copies of the MNL model to form an aggregate model based on a number of products in the set, each copy including an item utility parameter for each product in the set of products, and creating a matrix structure based on the number of products in the set, the matrix structure to be subtracted from each item utility parameter in the aggregate model. Further,

the example method includes estimating each item utility parameter of the aggregate model and the matrix structure based on the number of copies of the MNL model and the respondent choice data, and calculating a choice probability based on each of the estimated utility parameters.

**[0012]** Market researchers, product promoters, marketing employees, agents, analysts, and/or other people and/or organizations chartered with the responsibility of product management (hereinafter collectively referred to as “analysts”) typically attempt to justify informal and/or influential marketing decisions using one or more techniques that predict sales of one or more products of interest. Accurate forecasting models are useful to facilitate these decisions. In some circumstances, a product may be evaluated by one or more research panelists/respondents, which are generally selected based upon techniques having a statistically significant confidence level that such respondents accurately reflect a given demographic of interest. Techniques to allow respondents to evaluate a product, which allows the analysts to collect valuable choice data, include focus groups and/or purchasing simulations that allow the respondents to view and evaluate product concepts (e.g., providing images of products on a monitor, asking respondents whether they would purchase the products, discrete choice exercises, etc.).

**[0013]** The methods and apparatus described herein include, in part, one or more modeling techniques to facilitate sales forecasting and allow analysts to make informed marketing decisions. The modeling techniques described herein may operate with one or more modeling techniques, consumer behavior modeling, and/or choice modeling.

**[0014]** Generally speaking, choice modeling is a method to model a decision process of an individual in a particular context. Choice models may predict how individuals will react in different situations (e.g., what happens to demand for product A when the price of product B increases/decreases?). Predictions with choice models may be made over large numbers of scenarios and are based on the concept that people choose between available alternatives in view of one or more attributes of the products (e.g., price, size, traddress, feature (s), etc.). For example, when presented with a choice to take a car or bus to get to work, each of the alternative choices may be separated into three example attributes: price, time and convenience. For each attribute, a range of possible levels may be defined, such as three levels of price (e.g., \$0.50, \$1.00 or \$1.50), two levels of time (e.g., 5 minutes or 20 minutes, corresponding to two attributes of “convenient” or “not-convenient,” respectively). In the event a transportation mode exists that is cheapest, takes the least amount of time and is most convenient, then that transportation mode is likely to be selected. However, tradeoffs exist that cause a consumer to make choices, in which some consumers place greater weight on some attributes over others. For some consumers, convenience is so important that price has little effect on the choice, while other consumers are strongly motivated by price and will endure greater inconvenience to acquire the lowest price.

**[0015]** In the context of store, retail, and/or wholesale purchases, analysts may wish to model how a consumer chooses among the products available. Alternatives may be decomposed into attributes including, but not limited to product price, product display, or a temporary price reduction (TPR), such as an in-store marketing promotion that prices the product lower than its base price. Although the methods and apparatus described herein include price, display and/or TPR, any

other attributes may be considered, without limitation. Additional or alternative attributes may include brand or variety. When making a purchase decision, consumers balance the attributes (attribute utilities), such as brand preferences balanced with the price and their attraction for displays and/or TPRs, thereby choosing the product that maximizes their overall preference.

**[0016]** Although choice modeling techniques offer analysts an opportunity to employ a multinomial logit (MNL) model to predict probabilities of different consumer purchasing behaviors, use of the MNL model requires analyst discretion when selecting candidate available products from which a customer may choose. As used herein, the term “sourcing” refers to a degree of product differentiation within a set of available products from which a consumer may choose. For example, the MNL model assumes that any choices a customer may select within a set of products are equally substitutable for each other, which is sometimes referred to as fair share sourcing. In circumstances where the list of available products includes similar products, such as a choice between Coke®, Pepsi® and RC Cola®, the degree of substitutability may be relatively high. That is, in the event the original list of available cola products removed RC Cola® as an available selection for the consumer, then the remaining available products (i.e., Coke® and Pepsi®) are likely considered realistic substitutes for each other based on, for example, comparison (s) to observed respondent behavior(s).

**[0017]** In other examples, if the analyst desires to study a group of products in which one or more of the available products is not a suitable substitute, then the MNL model exhibits output error when calculating and/or otherwise predicting probabilities of different consumer purchasing behaviors. For instance, if the analyst arranges a set of available products to include Coke®, Pepsi® and Sprite®, then the MNL model assumes that each of those available products is deemed to be equally substitutable for the other product in the event that one or more of the selection choices become unavailable. If Pepsi® were removed from the list of available choices, then the MNL model calculates the probability of remaining choice selection as though Coke® and Sprite® were equally substitutable for each other and/or otherwise preferred by the consumer. This inherent limitation of the MNL model is sometimes referred to as an independence of irrelevant alternatives (IIA) property, in which the MNL model treats all product sourcing as fair share (equal) sourcing where all sourcing (e.g., any product) is equally substitutable to any other available product(s) under consideration.

**[0018]** Efforts to minimize the negative effects of the IIA property include implementing variants to the MNL model and/or logit models in general. Example variants include a probit (multinomial probit) model and/or a nested logit model. These variants do not exhibit the negative effects of the IIA property. However, while the nested logit does not suffer the negative effects of the IIA property, such models require analyst discretion when forming one or more groups of available products under study. In other words, the MNL model and the nested logit model cannot model complex sourcing scenarios that may reflect real-world product availability combinations that consumers experience. For situations in which the analyst wishes to identify respondent behaviors for a whole category of products (e.g., beverages), a realistic product mix may not be possible when the products of a set of products cannot be considered valid substitutes for each other. Additionally, while the multinomial probit model

may handle complex sourcing scenarios, the multinomial probit model does not apply a closed-form formula to calculate choice probabilities, thereby requiring substantial numerical integration and time. For example, multinomial probit models having more than ten (10) to fifteen (15) parameters (e.g., products of interest to study) could require days or weeks of computation time.

**[0019]** The methods and apparatus disclosed herein permit an analyst to consider complex sourcing product arrangements to calculate choice probabilities using a closed-form approach. At least one benefit of the methods and apparatus described herein includes realization of a computational efficiency improvement on one or more computing resources used to calculate choice probabilities using respondent choice data.

**[0020]** FIG. 1 is a schematic illustration of a system 100 to model consumer choice sourcing. In the illustrated example of FIG. 1, the system 100 includes a respondent database 102, a product selection database 104 and price/availability control 106, each of which are inputs to an aggregate logit sourcing (ALS) engine 108. In operation, the example ALS engine 108 calculates one or more choice probabilities 110 based on choice exercise data from the respondent database 102. Additionally, the one or more of the choice probabilities 110 calculated by the ALS engine 108 may be tailored in connection with one or more simulated price points and availability variations identified with the example price/availability control 106.

**[0021]** FIG. 2 is a schematic illustration of the example ALS engine 108 of FIG. 1. In the illustrated example of FIG. 2, the ALS engine 108 includes a choice modeling engine 202, a multinomial logit (MNL) engine 204, an aggregate building engine 206 and a sourcing modifier 208. The example sourcing modifier 208 includes a matrix engine 210, a matrix symmetry engine 212 and a matrix spatial engine 214. In operation, one or more models generated by the aggregate building engine 206 employ data from one or more choice modeling exercises during a modeling estimation performed by an example estimator 215 to calculate model parameters. The calculated model parameters from the one or more generated models are compared to the data from the choice modeling engine 202 by a measure of fit engine 216 to determine how well the parameters fit the choice data. In some examples, the measure of fit engine 216 employs a likelihood ratio test, but other types of techniques may be employed to determine whether the model parameters fit with the choice data. As described in further detail below, if the example measure of fit engine 216 determines that the model parameters do not fit the choice data, then the example sourcing modifier 208 may employ one or more alternate matrix structure(s) to calculate parameter offset value(s).

**[0022]** Calculated model parameters that result in an acceptable measure of fit indicate that the one or more models developed by the aggregate engine 206 may be used for one or more market simulation(s). Market simulation(s) may be calculated by a simulation engine 218, which uses one or more product specific price points and product availability measures from the price/availability control 106 to generate the choice probabilities 110. For example, an analyst may establish a first price point for each of the products Coke®, Pepsi® and Sprite® to allow the example simulation engine 218 to calculate choice probabilities for each of those products of interest. Additionally, the analyst may change one or more

price points to a second price point (e.g., make Coke® more expensive) to observe how the choice probabilities are affected.

[0023] Unlike one or more other models and/or modeling techniques employed to calculate choice probabilities, the example ALS engine 108 generates a model having a closed form. Closed form models perform significantly faster when compared to iterative modeling approaches, such as a multinomial probit model that can require days or weeks of computation time when a relatively small number of products (e.g., ten) of interest is studied. Additionally, the systems, methods, apparatus and/or articles of manufacture disclosed herein minimize the negative effects of the IIA property when calculating choice probability values for groups of products that may not be deemed substitutable to each other, but that may be a realistic product mix that a consumer would experience when shopping. While the MNL model suffers negative effects of the IIA property, the aggregate modeling approach disclosed herein generates a number of sub-models to form an aggregate model. Each sub-model, alone, is bound by the IIA property. However, each of the sub-models is associated with a matrix structure having an offset value to represent complex and diversified sourcing possibilities so that the aggregate sum of the sub-models is unaffected by the IIA property.

[0024] In operation, the example choice modeling engine 202 receives information related to an assortment of products that is to be studied from the example product selection data (database) 104. Generally speaking, respondents that participate in one or more choice modeling exercises are presented with any number of selectable products (e.g., presented via a kiosk, computer screen, slides, etc.). A number of products are shown multiple times to each respondent, in which one or more attributes of the products may change during each instance of viewing. Each virtual shopping trip displays a virtual shelf with a range of products that are organized in a manner to reflect what the respondent would see if at a retail store, for example. The choices made by the respondents during the virtual shopping trips are stored in the respondent database 102. Unlike virtual shopping trips conducted when employing the MNL model, the example ALS engine 108 avoids the need to capture analyst subjective input regarding opinions of which products are deemed proper substitutes for each other for placement on the virtual shelf. Reliance upon analyst discretion places limitations on statistical repeatability, accuracy and legitimacy of the products and/or subcategories chosen by the analyst.

[0025] Instead, the systems, methods, apparatus and/or articles of manufacture disclosed herein allow one or more subsets of the selectable products to be presented on the virtual shelf, in which the subsets are tailored to be displayed in a manner that addresses one or more questions by the client and/or analyst. For instance, a client may be interested in the choice probabilities of RC Cola® when placed near other available cola products. On the other hand, the client may be interested in the choice probabilities of RC Cola® when placed near other soft drinks in general, and/or when placed near energy drinks. In still other examples, virtual shopping trips prompt respondents to select from a range of products from one or more categories (e.g., dental products, baby food products, hair care products, laundry products, etc.) to determine choice probability values for the products within that category.

[0026] After the example choice modeling engine 202 obtains choice data from the respondents in view of the selection of products used in the virtual shopping trip(s), the choice data is stored in the example respondent database 102. The example MNL engine 204 builds an MNL model having a structure based on the number of items used in the choice modeling exercise (virtual shopping trip). Typically, to prevent respondent fatigue during the choice modeling exercise, the number of products for their consideration is limited to eighty (80), but any other number of products may be used with the example systems, methods, apparatus and/or articles of manufacture disclosed herein.

[0027] FIG. 3 is a portion of an example MNL model structure 300 that may be generated by the example MNL engine 204. In the illustrated example of FIG. 3, the MNL model structure 300 represents a number of items equal to a corresponding number of products to be evaluated, which is indicated by the variable I. Each item (I) of the MNL model structure 300 may include any number of corresponding parameters (β), such as item utilities (β<sub>i</sub>) (e.g., intercepts) and price utilities (β<sub>p</sub>) (e.g., slopes). As used herein, a utility represents a preference magnitude in which a higher utility corresponds to a higher preference. Utility values may represent a general preference for a product, or may represent a preference for a specific product from a specific respondent. Without limitation, utilities may indicate a preference in view of one or more product attributes and/or price. For instance, a price utility may increase when the price for a product decreases. In another example, the price utility for a product may increase when the price of a competing product increases.

[0028] A closed-form of the MNL model structure 300 may be represented by example Equation 1.

$$C_i = \frac{e^{[\beta_i + \beta_{pi} \$]}}{\sum_j e^{[\beta_j + \beta_{pj} \$j]}} \tag{Equation 1}$$

[0029] In the illustrated example of Equation 1, C<sub>i</sub> represents the choice probability for the i<sup>th</sup> item (I), β<sub>i</sub> represents an item utility for the i<sup>th</sup> item (I), β<sub>pi</sub> represents a price utility for the i<sup>th</sup> item (I), \$ represents a price of an item, and j represents the set of all items for the MNL model. Expressions of price may occur and/or otherwise be represented in any manner including, but not limited to a retail price, a base price, a geographical price average, an index to a base price, a logarithm of the price index, etc. As described above, although MNL modeling facilitates the calculation of choice probabilities with a closed-loop formula, thereby simplifying calculation efforts, the MNL model typically employs a set of products of interest that are deemed substitutable for each other due to the potential negative effects of the IIA property. Such limitations inherent in the MNL model hamper efforts to study complex sourcing patterns that may be exhibited and/or experienced by consumers when shopping. In other words, a consumer is not typically exposed to a set of equally substitutable products on a store shelf when shopping, rather, the consumer is typically presented with substantially more variety when reviewing one or more retail shelves.

[0030] FIG. 4 is a portion of an example aggregate model structure 400 that may be generated by the example aggregate building engine 206. In the illustrated example of FIG. 4, the aggregate building engine 206 generates a number of copies

of the MNL model structure **300** based on the number of items (I) (i.e., products of interest) to minimize the effects of the IIA property and allow an analyst to study a set of products that may not necessarily be deemed similar to each other. Each copy of the MNL model structure **300** generated by the example aggregate building engine **206** is a sub-model. The number of sub-models is based on the number of items (I), thereby generating I rows, and each row includes I parameters ( $\beta_x$ ) (e.g., utilities). The sub-models generated by the example aggregate building engine **206** are identical to each other, except for a weighting parameter W associated with each sub-model. Each weighting parameter W may be calculated in a manner consistent with example Equation 2.

$$W_s = \frac{e^{\beta_s}}{\sum_j e^{\beta_j}} \tag{Equation 2}$$

[0031] In the illustrated example of Equation 2, S represents an indication of each row (sub-model) of the aggregate model structure **400** of FIG. 4,  $\beta_s$  represents a parameter associated with the  $S^{\text{th}}$  sub-model, and the denominator of example Equation 2 represents the sum of exponentiated parameters for the aggregate model structure **400** of FIG. 4. Choice probability values for each item (i) of the set of items (I) may be calculated in a manner consistent with example Equation 3.

$$C_i = \sum_s W_s \frac{e^{[\beta_i + \beta_{pi} S_i]}}{\sum_j e^{[\beta_j + \beta_{pj} S_j]}} \tag{Equation 3}$$

[0032] The example sourcing modifier **208** builds upon the example aggregate model structure **400** of FIG. 4 by generating a matrix structure that identifies and/or otherwise calculates a parameter offset value for one or more of the parameters ( $\beta_x$ ), which is shown in FIG. 5 as a portion of an example matrix infused aggregate model structure **500**. In the illustrated example of FIG. 5, the aggregate building engine **206** generates matrix placeholders ( $M_{i,j}$ ) and incorporates them into each sub-model (rows  $S_1, S_2, \dots, S_j$ ). Each matrix placeholder ( $M_{i,j}$ ) includes a matrix location coordinate and may be used to identify a matrix value from the example sourcing modifier **208**. In operation, the matrix placeholders ( $M_{i,j}$ ) facilitate a manner in which one or more sourcing scenarios can be incorporated into the aggregate model structure **400** and to allow the analyst to study a greater variety of products of interest (e.g., the choice share effects between products when prices and/or availability values change). In other words, the matrix placeholders ( $M_{i,j}$ ) allow the aggregate model structure **400** to consider the effects of all products of interest on a first one of those products by subtracting a representation of sourcing effects from the preference parameter of the first product. For example, if the first product of interest is Coke® and the products of interest include other soft drinks, such as Pepsi® and Sprite®, then a sourcing effect from Pepsi® has a greater relative impact on Coke® than a sourcing effect from Sprite®. In other words, changes in availability and price for Pepsi® will have a greater effect on the choice probability associated with Coke®, but changes

in availability and price for Sprite® will have a lesser effect on the choice probability associated with Coke®.

[0033] The representations of sourcing effects for each of the products of interest (i.e., the set of items I) are generated by the example matrix engine **210**. In operation, the example matrix engine **210** generates and/or otherwise forms a matrix having dimensions of (I×I), with each cell within the I×I matrix having a parameter value to represent a sourcing effect for an item (i) within the set of items (I). For example, if the set of items I includes eighty (80) products to be studied by the analyst and/or otherwise requested by a client, then the example matrix engine **210** generates a matrix that is eighty columns by eighty rows (80×80). The example I×I matrix generated by the matrix engine **210** provides a manner of aggregation of sub-model sourcing adjustments so that the IIA property does not bias resulting choice probability calculations when analysts and/or clients select diverse product sets.

[0034] An example I×I matrix may be referred to as a straight matrix in which each matrix element corresponds to an intersection of two products from the set (I). However, the matrix diagonal will always include a value of zero because the diagonal reflects a comparison between a product and itself. The intersection of each non-diagonal row and column represents two products and reflects a degree of similarity between those two products. Values for each matrix element may be constrained in a manner consistent with example Equations 4 and 5.

$$M_{s,s}=0 \tag{Equation 4}$$

$$M_{s,v} \geq 0 \text{ (} S \neq V \text{)} \tag{Equation 5}$$

[0035] In the illustrated example of Equation 4, diagonal elements are zero, which reflects matrix cells where the row and column represent the same product. In the illustrated example of Equation 5, all non-diagonal matrix elements are constrained to positive values greater than or equal to zero. Matrix placeholders are inserted into the aggregate model structure **500** after each item utility parameter. Each matrix placeholder includes a coordinate that is mapped to the aggregate model structure **500** based on matching matrix rows to structure **500** rows (e.g., S values) and matching matrix columns to structure **500** columns (e.g., V values).

[0036] Considering an example choice modeling exercise that includes Coke® soft drinks, Pepsi® soft drinks and Sprite® soft drinks, the illustrated example matrix infused aggregate model structure **500** may reveal a first item (row  $S_1$ ) with Coke®, a second item (row  $S_2$ ) with Pepsi®, and a third item (row  $S_3$ ) with Sprite®. Additionally, corresponding parameters denoted with “1,” “2,” and “3” reflect Coke®, Pepsi® and Sprite® products, respectively. In view of the instant example, parameters  $\beta_1, \beta_2$  and  $\beta_3$  refer to an indication of the preference that the respondent pool has for the corresponding brands of soft drink. From a substitution point of view, assume that the choice selections from the respondent database **102** identify that Coke® and Pepsi® are more substitutable for each other, while Sprite® is not deemed a common and/or otherwise observed substitute for the products of Coke® and Pepsi®. As such, if the price of Coke® increases, then a corresponding choice probability that Pepsi® will be purchased to a greater degree will increase. On the other hand, price and/or availability fluctuations of Sprite® have substantially less effect on the products Coke® and/or Pepsi®.

**[0037]** Matrix index value  $M_{(I,J)}$  reflects a preference of Coke® on itself, which is constrained by example Equation 4 to equal zero. Accordingly, the example matrix infused aggregate model structure **500** does not modify the sourcing behavior from  $\beta_1$  in row  $S_1$  (see row  $S_1$ , column  $V_1$  of the example matrix infused aggregate model structure **500**). On the other hand, matrix index value  $M_{(1,2)}$  reflects a relative degree of similarity between Coke® and Pepsi®, and matrix index value  $M_{(1,3)}$  reflects a relative degree of similarity between Coke® and Sprite®. The value for index value  $M_{(1,2)}$ , based on the example assumptions that Coke® and Pepsi® are deemed significantly more substitutable for each other as compared to Coke® and Sprite® and/or Pepsi® and Sprite®, will be relatively low (e.g., values closer to zero are indicative to a greater degree of similarity). That is, the sourcing effects of the preference of Pepsi® ( $\beta_2$ ) are significantly affected by price and/or availability metrics associated with Coke®, as shown by the subtraction of the sourcing modifier indicative of the Coke®/Pepsi® matrix intersection ( $M_{(0,2)}$ ) from the preference parameter associated with Pepsi® ( $\beta_2$ ) (see row  $S_1$ , column  $V_2$  of the example matrix infused aggregate model structure **500**).

**[0038]** On the other hand, matrix index value  $M_{(1,3)}$  is relatively high because it reflects a relative degree of similarity (or lack thereof) between Coke® and Sprite®. The effects of the relationship between Coke® and Sprite® are evident in the example matrix infused aggregate model structure **500** in row  $S_1$ , column  $V_3$  where the relatively high value for the sourcing modifier indicative of the Coke®/Sprite® matrix intersection ( $M_{(1,3)}$ ) is subtracted from the preference parameter associated with Sprite® ( $\beta_3$ ). In the event that a price and/or availability metric for Coke® changes, the effect on Sprite® will have a lower impact on the resulting choice probability. Unlike a traditional MNL model, in which all products under consideration are treated as equal substitutes for each other, the example methods, apparatus, systems and/or articles of manufacture disclosed herein apply a matrix placeholder ( $M_{(I,J)}$ ) having a corresponding offset value to proportionally affect choice probability calculations in a manner consistent with actual market and/or customer experiences.

**[0039]** Values for each of the matrix cells, and values for each of the parameters of the matrix infused aggregate model structure **500** (e.g.,  $\beta_1$ ,  $\beta_{P1}$ ,  $\beta_2$ ,  $\beta_{P2}$ , etc.) are calculated by iteratively estimating the matrix infused aggregate model structure **500** with the choice data stored in the example respondent database **102**. As disclosed above, the respondent database **102** stores choice selections from respondents during a choice modeling exercise, in which the respondents engage in virtual shopping trips where the set of items (I) (products) are presented via virtual shelves. Initial values for each of the matrix cells and/or parameters may be set at random, predetermined values, or set via a random number generator, in which the estimation process allows the matrix cell values and parameters to converge. After the estimation process completes, in which the matrix cell values and parameters converge, the choice probability may be calculated via a closed-form approach in a manner consistent with example Equation 6.

$$C_i = \sum_s W_s \frac{e^{[\beta_i - M_{s,i} + \beta_{pi} S_i]}}{\sum_j e^{[\beta_j - M_{s,j} + \beta_{pj} S_j]}} \tag{Equation 6}$$

**[0040]** In some examples, the parameters associated with price (price utilities/preferences) may be modified to facilitate scaling and address sub-model sensitivity. For example, each price utility (e.g.,  $\beta_{p1}$ ) may include a scaling price parameter ( $\beta_{p'1}$ ,  $\beta_{p'2}$ ,  $\beta_{p'3}$ , . . . ,  $\beta_{p's}$ ) for each row of the example matrix infused aggregate model structure **500**. The choice probability may be calculated via a closed-form approach in a manner consistent with example Equation 7.

$$C_i = \sum_s W_s \frac{e^{[\beta_i - M_{s,i} + (\beta_{pi} + \beta_{p's}) S_i]}}{\sum_j e^{[\beta_j - M_{s,j} + \beta_{pj} + \beta_{p's} S_j]}} \tag{Equation 7}$$

**[0041]** As described above, the example estimator **215** performs an iterative estimation using the model and data collected from the respondent choice exercise. Generally speaking, values of the parameters of the example model, such as the model represented by the closed-form choice probability of example Equation 7 and/or the example matrix infused aggregate model structure **500**, are estimated based on measured and/or empirical data. The example estimator **215** may employ one or more estimation methods including, but not limited to a maximum likelihood method, a Bayes estimation method and/or a minimum mean squared error. To ascertain whether any number of estimation iterations converge to acceptable parameter values, the example measure of fit engine **216** employs a fitting test (e.g., a likelihood ratio test, etc.) to determine how well the choice data fits with the converged parameters of the model (e.g., the example matrix infused aggregate model structure **500**). In the event that the model parameters converge, but do not fit the choice data to an acceptable degree, the example measure of fit engine **216** employs the example sourcing modifier **208** to apply one or more alternate matrix structures to calculate the parameter offset value(s) (e.g.,  $M_{(I,J)}$ ). In other examples, the measure of fit engine **216** employs one or more fitting tests during each iteration. In the event that successive iterations do not improve by a threshold amount, then current parameter values may be accepted as final. However, in the event that successive iterations that continue to illustrate improvement beyond a threshold value, then the one or more iterations may continue to develop parameter value(s).

**[0042]** As described above, the matrix structure is based on the number of product of interest (items) to be studied. The methods, apparatus, systems and/or articles of manufacture disclosed herein generate one or more matrix structures to reflect effects of sourcing behaviors on all products of interest under consideration. Sourcing behaviors, which are facilitated by matrix cell values (parameters), are subtracted from each product utility in a manner that is proportional to a degree of similarity between one or more other products. A straight matrix is generated by the example matrix engine **210** by assigning an equal number of matrix rows and columns to form a square  $I \times I$  matrix. For example, if eighty (80) products of interest are selected for the choice modeling exercise, then



the example matrix engine **210** generates an 80x80 square matrix and populates each matrix cell at a matrix placeholder  $M_{(i,j)}$  having a parameter and a parameter value. Each row of the straight matrix represents one of the eighty (80) items, and each column represents the same set of eighty (80) items in the same order. In other examples, a matrix is formed in connection with one or more subcategories, as described in further detail below. The diagonal of the matrix reflects intersections of the product with itself, and is set to zero. However, each off-diagonal placeholder represents an intersection indicative of a relationship between one product and another product (e.g., a degree of similarity). During the model estimation, the example estimator **215** iteratively estimates both the model parameters and the parameters of the straight matrix so that each parameter converges to a value. Initial values for all model and/or matrix parameters may be initially set at a random number, zero and/or any other value before converging during estimation in view of the choice model data stored in the respondent database **102**.

[**0043**] In some examples, the straight matrix may not be computationally efficient for the ALS engine **108**. The example straight matrix includes a relatively high degree of flexibility when compared to one or more alternate matrix structures, such as a symmetric matrix and/or a geometric matrix. As described in further detail below, while the symmetric matrix and/or the geometric matrix impose a greater degree of computational constraint when compared to the straight matrix, the symmetric matrix and/or the geometric matrix may be appropriate when model estimation overfits the choice model data stored in the example respondent database **102**. For example, although the straight matrix includes a parameter value for each and every product combination of interest, thereby having the greatest potential to fit the choice data accurately, the relatively large number of parameters may become computationally intensive and fail to produce statistically relevant convergence during each estimation iteration.

[**0044**] A symmetric matrix decreases the number of parameters by a factor of two, thereby reducing computational loads for the example estimator **215**. The example matrix symmetry engine **212** forces a symmetry structure of the straight matrix so that each parameter value on a lower half of the diagonal is the same (e.g., linked) as each corresponding parameter value on an upper half of the diagonal. A straight matrix may be converted into a symmetric matrix by linking cells above the matrix diagonal with cells below the matrix diagonal. In some examples, the analyst and/or client may begin with a matrix structure having a more rigid form when compared to the straight matrix and, depending on a measure of fit indication from the example measure of fit engine **216**, adjust the matrix structure accordingly. In some circumstances, a symmetric matrix produces statistically appropriate measure of fit values to justify using the symmetric matrix. In other circumstances, the symmetric matrix fails to cause statistically appropriate measures of fit, in which case the straight matrix may be employed.

[**0045**] A geometric matrix introduces a degree of structure greater than that of the symmetric matrix, thereby affording greater conservation of computing resources during estimation by the example estimator **215**. Matrix values within the geometric matrix have a spatial relationship to each other based on a number of matrix dimensions. For ease of discussion, the example geometric matrix will be described in connection with three (3) dimensions, but any other number of

dimensions may be employed with the example systems, methods, apparatus and/or articles of manufacture disclosed herein.

[**0046**] FIG. 6 is a portion of an example geometric matrix **600** generated by the example matrix spatial engine **214**. In the illustrated example of FIG. 6, the geometric matrix **600** includes a number of dimensions (n) (**602**) (columns) and a total number of items (I) (**604**) (rows). The number of dimensions n is constrained to be less than or equal to the number of items I. Matrix values are designated by  $\chi_{(I,n)}$ , and particular items (s) within the total number of items (I) are located in n-dimensional space in a manner consistent with example Equation 8.

$$(\chi_{s,1}, \chi_{s,2}, \dots, \chi_{s,n}) \tag{Equation 8}$$

[**0047**] For example,  $\chi_{(1,1)}$  represents a first dimensional coordinate (e.g., an x-axis spatial value) for a first product,  $\chi_{(1,2)}$  represents a second dimensional coordinate (e.g., a y-axis spatial value) for the first product, and  $\chi_{(1,3)}$  represents a third dimensional coordinate (e.g., a z-axis spatial value) for the first product. Similarly,  $\chi_{(2,1)}$  represents a first dimensional coordinate for a second product,  $\chi_{(2,2)}$  represents a second dimensional coordinate for the second product, and  $\chi_{(2,3)}$  represents a third dimensional coordinate for the second product. Conceptually, each product may be represented as a point in an n-dimensional space. Continuing with the above-described examples of Coke®, Pepsi® and Sprite®, because Coke® and Pepsi® are both cola products sold in similar markets and potentially have similar preferences, spatial coordinates for Coke® and Pepsi® are likely relatively near each other when compared to Sprite®, which is not a cola product. A relative distance between product representations in the example geometric matrix **600** may be calculated in a manner consistent with example Equation 9.

$$M_{s,v} = \sum_d (\chi_{s,d} - \chi_{v,d})^2. \tag{Equation 9}$$

[**0048**] In the illustrated example of Equation 9, M represents a distance between a first item s from the total list of items I, and another item v, and d represents a dimension from the n dimensions. As described above, the example geometric matrix **600** exhibits the least amount of flexibility because it imposes the greatest amount of structure against the provided choice data. The estimation performed by the example estimator **215** may employ the geometric matrix for circumstances in which matrices having greater degrees of freedom result in overfitting. For instance, a model (or a matrix within the model) that includes increasing numbers of parameters becomes more flexible and eventually fits the supplied data to the greatest degree, but at the expense of computational loads that may cause estimation inefficiency. Although the example geometric matrix **600** may not fit the provided choice data as well as the straight and/or the symmetric matrix, the geometric matrix **600** may be appropriate in response to statistically satisfactory measures of fit values calculated by the example measure of fit engine **216**.

[**0049**] In some examples, rigid matrix structures may not fit well with the choice data that results from the respondent choice modeling exercise. In other words, some product mixes may not follow geometric relationships, thereby causing the application of the geometric matrix to produce poor statistical measures of fit values. For example, consider a mix

of laundry detergent products that includes (a) detergent with bleach, (b) detergent with color safe bleach and (c) detergent with no bleach. Choice data results may identify a relatively large degree of substitution between (a) and (b) because consumers prefer to have a whites/colors bleach product, and the choice data results may also identify a relatively large degree of substitution between (b) and (c). Continuing with the laundry detergent example, consider that the choice data results find no substitution between (a) and (c). Generally speaking, the geometric matrix adheres to one or more logical axioms to produce conclusions. As described in the above example, if (a)=(b), and (b)=(c), then geometric principles suggest that (a) should be equal to or similar to (c) from a spatial point of view. However, the example geometric model may attempt to force relationships in a manner inconsistent with the collected choice data and force a similarity between (a) and (c) despite choice data suggesting no such relationship actually occurs. In such circumstances, a poor measure of fit results and the example measure of fit engine 216 selects an alternate matrix structure having a greater degree of flexibility.

[0050] In still further examples, subcategories may be employed to, in part, reduce matrix sizes. Generating one or more subcategories imposes further constraint on the model and may introduce a degree of analyst discretion. However, in circumstances where a matrix becomes large due to a large number of products, employing one or more subcategories reduces a computational burden of the model. For example, a matrix representing 100 items includes 100 rows and 100 columns, which corresponds to 10,000 individual matrix cells. In the event that 30 subcategories are employed, then a number of columns and rows each collapse to form a 30 by 30 matrix, thereby substantially reducing a computational burden of parameter value estimation. For instance, if Pepsi® and Coke® are deemed similar (e.g., any differences between the two are inconsequential), then a subcategory indicative of national cola brands may allow the rows and columns associated with Coke® and Pepsi® to collapse together. In other words, multiple products may share the same M values when they share the same subcategory.

[0051] Equation 10 illustrates an example manner in which each model row is calculated with a weight (w) in view of S subcategories.

$$w_s = \frac{\sum_{j \in S} e^{\beta_j}}{\sum_j e^{\beta_j}} \tag{Equation 10}$$

[0052] In the illustrated example of Equation 10, the numerator notation signifies to sum over all items in subcategory s. As such, employing subcategories allows a matrix to be formed as an S by S matrix rather than the larger I by I matrix described above. In view of the implementation of subcategories, choice probability may be calculated in a manner consistent with example Equation 11.

$$C_i = \sum_s w_s \frac{e^{[\beta_i - M_{s,vi} + (\beta_{pi} + \beta_{p's})\delta_i]}}{\sum_j e^{[\beta_j - M_{s,vj} + (\beta_{pj} + \beta_{p'j})\delta_j]}} \tag{Equation 11}$$

[0053] While an example manner of implementing an example system 100 to model consumer choice sourcing has been illustrated in FIGS. 2-6, one or more of the elements, processes and/or devices illustrated in FIGS. 1-6 may be combined, divided, re-arranged, omitted, eliminated and/or implemented in any other way. Further, the example respondent database 102, the example product selection database 104, the example price/availability control 106, the example ALS engine 108, the example choice modeling engine 202, the example MNL engine 204, the example aggregate building engine 206, the example sourcing modifier 208, the example matrix engine 210, the example matrix symmetry engine 212, the example matrix spatial engine 214, the example estimator 215, the example measure of fit engine 216 and/or the example simulation engine 218 of FIGS. 1 and 2 may be implemented by hardware, software, firmware and/or any combination of hardware, software and/or firmware. Thus, for example, any of the example respondent database 102, the example product selection database 104, the example price/availability control 106, the example ALS engine 108, the example choice modeling engine 202, the example MNL engine 204, the example aggregate building engine 206, the example sourcing modifier 208, the example matrix engine 210, the example matrix symmetry engine 212, the example matrix spatial engine 214, the example estimator 215, the example measure of fit engine 216 and/or the example simulation engine 218 of FIGS. 1 and 2 could be implemented by one or more circuit(s), programmable processor(s), application specific integrated circuit(s) (ASIC(s)), programmable logic device(s) (PLD(s)) and/or field programmable logic device(s) (FPLD(s)), etc. When any of the apparatus claims of the patent are read to cover a purely software and/or firmware implementation, at least one of the example respondent database 102, the example product selection database 104, the example price/availability control 106, the example ALS engine 108, the example choice modeling engine 202, the example MNL engine 204, the example aggregate building engine 206, the example sourcing modifier 208, the example matrix engine 210, the example matrix symmetry engine 212, the example matrix spatial engine 214, the example estimator 215, the example measure of fit engine 216 and/or the example simulation engine 218 of FIGS. 1 and 2 are hereby expressly defined to include a tangible computer readable medium such as a memory, DVD, CD, etc. storing the software and/or firmware. Further still, the example system 100 of FIG. 1 may include one or more elements, processes and/or devices in addition to, or instead of, those illustrated in FIGS. 1 and 2, and/or may include more than one of any or all of the illustrated elements, processes and devices.

[0054] Flowcharts representative of example machine readable instructions for implementing the system 100 of FIGS. 1 and 2 are shown in FIGS. 7-9. In these examples, the machine readable instructions comprise a program for execution by a processor such as the processor P105 shown in the example computer P100 discussed below in connection with FIG. 10. The program may be embodied in software stored on a tangible computer readable medium such as a CD-ROM, a floppy disk, a hard drive, a digital versatile disk (DVD), or a memory associated with the processor P105, but the entire program and/or parts thereof could alternatively be executed by a device other than the processor P105 and/or embodied in firmware or dedicated hardware. Further, although the example program is described with reference to the flowcharts illustrated in FIGS. 7-9, many other methods of imple-

menting the example system **100** may alternatively be used. For example, the order of execution of the blocks may be changed, and/or some of the blocks described may be changed, eliminated, or combined.

**[0055]** As mentioned above, the example processes of FIGS. **7-9** may be implemented using coded instructions (e.g., computer readable instructions) stored on a tangible computer readable medium such as a hard disk drive, a flash memory, a read-only memory (ROM), a compact disk (CD), a digital versatile disk (DVD), a cache, a random-access memory (RAM) and/or any other storage media in which information is stored for any duration (e.g., for extended time periods, permanently, brief instances, for temporarily buffering, and/or for caching of the information). As used herein, the term tangible computer readable medium is expressly defined to include any type of computer readable storage and to exclude propagating signals. Additionally or alternatively, the example processes of FIGS. **7-9** may be implemented using coded instructions (e.g., computer readable instructions) stored on a non-transitory computer readable medium such as a hard disk drive, a flash memory, a read-only memory, a compact disk, a digital versatile disk, a cache, a random-access memory and/or any other storage media in which information is stored for any duration (e.g., for extended time periods, permanently, brief instances, for temporarily buffering, and/or for caching of the information). As used herein, the term non-transitory computer readable medium is expressly defined to include any type of computer readable medium and to exclude propagating signals.

**[0056]** The process **700** of FIG. **7** begins at block **702** where the example choice modeling engine **202** identifies products of interest to be studied. The example choice modeling engine **202** may facilitate a user interface to allow an analyst and/or client select products of interest, which may include existing market products, existing competitive market products and/or new products that have not yet been offered for sale in the market. The example product selection database **104** may include one or more lists of available market products and/or details related to the products. In some examples, the product selection database **104** may include data from the product reference library (PRL) cultivated and managed by Nielsen®, which codes more than 700,000 items, in which each item includes an average of forty (40) descriptive characteristics, is an example source for such product information. However, the example systems, methods, apparatus and/or articles of manufacture are not limited to using the PRL as one or more alternate sources of product reference data may be employed. Product information may include, but is not limited to product name, manufacturer name, brand, packaging type, product size, flavor, lot number, serial number, nutritional information and/or a corresponding universal product code (UPC). One or more price points and/or ranges of price points may be identified and associated with the identified products of interest to be communicated to one or more respondents during a choice modeling exercise. Once a set of products of interest are identified, the example choice modeling engine performs a choice modeling exercise with respondents (block **704**) and saves choice data to the example respondent database **102** (block **706**). As described above, and in further detail below, the stored choice data is utilized during an iterative estimation process to converge parameter values of an aggregate model generated and/or otherwise built by the example ALS engine **108** (block **708**), such as the example matrix infused aggregate model structure **500** of FIG. **5**. The example aggregate

model generated by the ALS engine **108** may be applied in closed-loop format to one or more simulations to calculate choice probability values of one or more products (block **710**).

**[0057]** The program **708** of FIG. **8** illustrates further detail related to generating the aggregate model described in FIG. **7**. In the illustrated example of FIG. **8**, the program **708** invokes the MNL engine **204** to establish a base MNL model having a number of items equal to those used in the choice modeling exercise (block **802**). For example, if eighty (80) products (items) were selected to be studied, then the example MNL engine **204** generates a MNL model having eighty (80) parameters. Without limitation, the example MNL engine **204** may include product parameters, price parameters and/or any number of parameters that reflect attribute preferences associated with each product.

**[0058]** The example aggregate building engine **206** makes *I* copies of the base MNL model generated by the example MNL engine **204** (block **804**). As described above, while each MNL model, by itself, exhibits negative effects of the IIA property, the example aggregate building engine **206** builds an aggregate model to, in part, minimize and/or otherwise drown-out the negative effects of the IIA property. Additionally, to impose a sourcing effect of every product of interest on every other product of interest in a proportional manner, the example aggregate building engine **206** invokes the example sourcing modifier **208** to generate one or more matrix structures (block **806**), as described above and as described in further detail below.

**[0059]** The aggregate model generated by the aggregate building engine **206** and the matrix generated by the example sourcing modifier **208** are iteratively estimated by the example estimator **215** (block **808**). As described above, the iterative estimations allow the aggregate model parameters and the matrix parameters to converge to one or more values that fit the choice data stored in the example respondent database **102**. Resulting model parameters and matrix parameters are evaluated by the example measure of fit engine **216** to determine whether the converged parameter values are statistically appropriate in view of the supplied choice data (block **810**). If not, then the example measure of fit engine **216** invokes the example sourcing modifier **208** to generate an alternate matrix structure having a greater or lesser degree of structure (block **812**), and the example program **708** returns to block **806**. For example, in the event that a highly structured geometric model fails to converge with parameters that are statistically appropriate, then a symmetric or straight matrix may be employed. On the other hand, in the event that a straight matrix becomes computationally intensive and/or suffers from overestimation, then the example measure of fit engine may invoke the example sourcing modifier **208** to generate a matrix having additional structure, such as a symmetric matrix or a geometric matrix.

**[0060]** The program **806** of FIG. **9** illustrates further detail related to building the matrix structure described in FIG. **8**. In the illustrated example of FIG. **9**, the program **806** determines whether to generate a geometric matrix (block **902**), a straight matrix (block **904**) or a symmetric matrix (block **904**). In the event a geometric matrix is to be generated (block **902**), the example sourcing modifier **208** identifies a number of spatial dimensions to use (block **905**). Spatial dimensions can be less than or equal to the number of products (items) of interest used in the choice modeling exercise. A greater number of dimensions (*n*) result in a greater number of matrix param-

eters, a greater resolution and a greater chance for better fitting the supplied choice data. However, a greater number of dimensions (n) also results in a greater computational burden for the example estimator 215. In some examples, the example sourcing modifier 208 invokes the matrix spatial engine 214 as a first attempt under an assumption and/or empirical understanding that a geometric matrix is an appropriate and efficient choice for the type(s) of product mix of the selected products (I). The Matrix spatial engine 214 builds the geometric matrix having n dimensions as columns, and I items as rows (block 906) so that each product pair spatial distance can be calculated in a manner consistent with example Equation 9 (block 908). Each matrix element is associated with the matrix infused aggregate model structure, such as the structure 500 of FIG. 5, in a manner that allows the example estimator 215 to perform iterative estimations for parameter convergence (block 910).

[0061] In the event that a straight matrix is to be employed (block 904), the example sourcing modifier 208 employs the matrix engine 210 to build a square matrix of a size based on the number of products (items) of interest (I) (block 912). In other words, the example matrix engine 210 builds an I×I matrix structure (block 912), and then adds index parameter placeholders in each matrix cell (block 914). As described above, because the matrix row includes a representation of each product of interest (e.g., counting from a matrix row index value of 1 to I), and because the matrix column includes a representation of each product of interest in the same order as those products listed in the row (e.g., counting from a matrix column index value of 1 to I), then the diagonal cells will each be set to zero. Each matrix element is associated with the matrix infused aggregate model structure, such as the structure 500 of FIG. 5, in a manner that allows the example estimator 215 to perform iterative estimations for parameter convergence (block 910).

[0062] In the event that a symmetric matrix is to be employed (block 904), then the example sourcing modifier 208 employs the matrix symmetry engine 212 to build a square matrix of a size based on the number of products (items) of interest (I) (block 916). In other words, the example matrix symmetry engine 212 builds an I×I matrix structure (block 916), and then adds index parameter placeholders in each matrix cell (block 918). To force a symmetric structure, the example matrix spatial engine 214 identifies matching matrix cells on either side of the matrix diagonal and sets them equal to each other (block 920). In some examples, the symmetric matrix can be implemented as a half-matrix that only employs parameters on the upper or lower half of the matrix diagonal. As such, the symmetric matrix only employs half as many parameters as the straight matrix during the iterative estimation executed by the example estimator 215. Each matrix element is associated with the matrix infused aggregate model structure, such as the structure 500 of FIG. 5, in a manner that allows the example estimator 215 to perform iterative estimations for parameter convergence (block 910).

[0063] FIG. 10 is a block diagram of an example computer P100 capable of executing the instructions of FIGS. 7, 8 and/or 9 to implement the example respondent database 102, the example product selection database 104, the example price/availability control 106, the example ALS engine 108, the example choice modeling engine 202, the example MNL engine 204, the example aggregate building engine 206, the example sourcing modifier 208, the example matrix engine 210, the example matrix symmetry engine 212, the example

matrix spatial engine 214, the example estimator 215, the example measure of fit engine 216 and/or the example simulation engine 218 of FIGS. 1 and 2. The computer P100 can be, for example, a server, a personal computer, an Internet appliance, or any other type of computing device.

[0064] The system P100 of the instant example includes a processor P105. For example, the processor P105 can be implemented by one or more Intel® microprocessors from the Pentium® family, the Itanium® family or the XScale® family. Of course, other processors from other families are also appropriate.

[0065] The processor P105 is in communication with a main memory including a volatile memory P115 and a non-volatile memory P120 via a bus P125. The volatile memory P115 may be implemented by Synchronous Dynamic Random Access Memory (SDRAM), Dynamic Random Access Memory (DRAM), RAMBUS Dynamic Random Access Memory (RDRAM) and/or any other type of random access memory device. The non-volatile memory P120 may be implemented by flash memory and/or any other desired type of memory device. Access to the main memory P115, P120 is typically controlled by a memory controller (not shown).

[0066] The computer P100 also includes an interface circuit P130. The interface circuit P130 may be implemented by any type of interface standard, such as an Ethernet interface, a universal serial bus (USB), and/or a PCI express interface.

[0067] One or more input devices P135 are connected to the interface circuit P130. The input device(s) P135 permit a user to enter data and commands into the processor P105. The input device(s) can be implemented by, for example, a keyboard, a mouse, a touchscreen, a track-pad, a trackball, and/or a voice recognition system.

[0068] One or more output devices P140 are also connected to the interface circuit P130. The output devices P140 can be implemented, for example, by display devices (e.g., a liquid crystal display, a cathode ray tube display (CRT), and/or other display). The interface circuit P130, thus, typically includes a graphics driver card.

[0069] The interface circuit P130 also includes a communication device (not shown) such as a modem or network interface card to facilitate exchange of data with external computers via a network (e.g., an Ethernet connection, a digital subscriber line (DSL), a telephone line, coaxial cable, a cellular telephone system, etc.).

[0070] The computer P100 also includes one or more mass storage devices P150 for storing software and data. Examples of such mass storage devices P150 include floppy disk drives, hard drive disks, compact disk drives and digital versatile disk (DVD) drives. The mass storage device P150 may implement the example respondent database 102 and/or the example product selection database 104.

[0071] The coded instructions of FIGS. 7-9 may be stored in the mass storage device P150, in coded instructions P110 of the volatile memory P115, in coded instructions P112 of the non-volatile memory P120, and/or on a removable storage medium such as a CD or DVD.

[0072] From the foregoing, it will be appreciated that the above disclosed systems, methods, apparatus and articles of manufacture facilitate prediction of new product performance metrics within one or more geographies and/or channels of interest when no prior historical sales data is available for the new product in the corresponding geography or channel.

[0073] Therefore, although certain example methods, apparatus and articles of manufacture have been described herein, the scope of coverage of this patent is not limited thereto. On the contrary, this patent covers all methods, apparatus and articles of manufacture fairly falling within the scope of the claims either literally or under the doctrine of equivalents.

1. A method to calculate a choice probability, comprising:  
identifying a set of products;  
receiving respondent choice data associated with the set of products;  
adding the set of products to a base multinomial logit (MNL) model;  
generating, with a programmed processor, a number of copies of the MNL model to form an aggregate model based on a number of products in the set, each copy including an item utility parameter for each product in the set of products;  
creating a matrix structure based on the number of products in the set, the matrix structure to be subtracted from each item utility parameter in the aggregate model;  
estimating each item utility parameter of the aggregate model and the matrix structure based on the number of copies of the MNL model and the respondent choice data; and  
calculating a choice probability based on each of the estimated utility parameters.

2. (canceled)

3. A method as described in claim 1, wherein the base MNL model size is based on the number of products in the set.

4-8. (canceled)

9. A method as described in claim 1, further comprising estimating the matrix structure with the aggregate model to facilitate parameter convergence of each item utility parameter and a plurality of parameters in the matrix structure.

10. A method as described in claim 1, wherein the matrix structure comprises a straight matrix having a number of rows equal to the number of products and a number of columns equal to the number of products.

11. A method as described in claim 10, wherein each of the number of rows represents one of the number of products and each of the number of columns represents one of the number of products.

12. A method as described in claim 11, wherein an order of the products in the number of rows is the same as an order of the products in the number of columns.

13. A method as described in claim 12, wherein each of the rows intersects each of the columns at a matrix cell to reflect a relationship between a product of the row with a product of the column.

14. A method as described in claim 13, further comprising inserting a parameter at the matrix cell for each row and column intersection.

15. A method as described in claim 14, wherein the parameter at the matrix cell for each row and column intersection is estimated with the respondent choice data to calculate converged parameter values therein.

16. (canceled)

17. (canceled)

18. A method as described in claim 1, wherein the matrix structure comprises a geometric matrix having a number of rows equal to the number of products in the set, and a number of columns equal to a number of dimensions.

19. (canceled)

20. A method as described in claim 18, further comprising associating each cell within the geometric matrix with a spatial parameter indicative of a distance between a first and a second product from the number of products.

21. A method as described in claim 20, wherein estimating further comprises converging a value of the spatial parameter based on the aggregate model and the respondent choice data.

22. An apparatus to calculate a choice probability, comprising:

a choice modeling engine to identify a set of products and receive respondent choice data associated with the set of products;

a multinomial logit (MNL) engine to add the set of products to a base MNL model;

an aggregate building engine to generate a number of copies of the MNL model to form an aggregate model based on a number of products in the set, each copy including an item utility parameter for each product in the set of products;

a sourcing modifier to create a matrix structure based on the number of products in the set, the matrix structure to be subtracted from each item utility parameter in the aggregate model;

an estimator to estimate each item utility parameter of the aggregate model and the matrix structure based on the number of copies of the MNL model and the respondent choice data; and

a simulation engine to calculate a choice probability based on each of the estimated utility parameters.

23. (canceled)

24. An apparatus as described in claim 22, wherein the MNL engine generates the base MNL model based on the number of products in the set.

25. An apparatus as described in claim 22, further comprising the aggregate building engine inserting a price utility parameter for each product in the set of products.

26-29. (canceled)

30. An apparatus as described in claim 22, further comprising the estimator estimating the matrix structure with the aggregate model to facilitate parameter convergence of each item utility parameter and a plurality of parameters in the matrix structure.

31. An apparatus as described in claim 22, further comprising a matrix engine to generate a straight matrix having a number of rows equal to the number of products and a number of columns equal to the number of products.

32. An apparatus as described in claim 31, wherein each of the number of rows represents one of the number of products and each of the number of columns represents one of the number of products.

33. An apparatus as described in claim 32, wherein the matrix engine places the number of products in the number of rows in the same order as the number of products in the number of columns.

34. An apparatus as described in claim 33, wherein the matrix engine intersects each of the rows with each of the columns at a matrix cell to reflect a relationship between a product of the row with a product of the column.

35. (canceled)

36. (canceled)

37. An apparatus as described in claim 22, further comprising a matrix spatial engine to generate a geometric matrix

having a number of rows equal to the number of products in the set, and a number of columns equal to a number of dimensions.

38. An apparatus as described in claim 37, wherein the matrix spatial engine associates each cell within the geometric matrix with a spatial parameter indicative of a distance between a first and a second product from the number of products.

39. An apparatus as described in claim 38, wherein the estimator converges a value of the spatial parameter based on the aggregate model and the respondent choice data.

40. A tangible machine accessible medium having instructions stored thereon that, when executed, cause a machine to, at least:

- identify a set of products;
- receive respondent choice data associated with the set of products;
- add the set of products to a base multinomial logit (MNL) model;
- generate, with a programmed processor, a number of copies of the MNL model to form an aggregate model based on a number of products in the set, each copy including an item utility parameter for each product in the set of products;
- create a matrix structure based on the number of products in the set, the matrix structure to be subtracted from each item utility parameter in the aggregate model;
- estimate each item utility parameter of the aggregate model and the matrix structure based on the number of copies of the MNL model and the respondent choice data; and
- calculate a choice probability based on each of the estimated utility parameters.

41. (canceled)

42. A tangible machine accessible medium as described in claim 40 having instructions stored thereon that, when executed, cause a machine to generate the base MNL model having a size based on the number of products in the set.

43. (canceled)

44. (canceled)

45. A tangible machine accessible medium as described in claim 40 having instructions stored thereon that, when executed, cause a machine to include, in each item utility parameter, an attribute utility parameter.

46. A tangible machine accessible medium as described in claim 45 having instructions stored thereon that, when executed, cause a machine to include, in the attribute utility

parameter, at least one of a product price utility, a product size utility, a product tradedress utility, or a product feature utility.

47. (canceled)

48. A tangible machine accessible medium as described in claim 40 having instructions stored thereon that, when executed, cause a machine to estimate the matrix structure with the aggregate model to facilitate parameter convergence of each item utility parameter and a plurality of parameters in the matrix structure.

49. A tangible machine accessible medium as described in claim 40 having instructions stored thereon that, when executed, cause a machine to build a matrix structure as a straight matrix having a number of rows equal to the number of products and a number of columns equal to the number of products.

50-54. (canceled)

55. A tangible machine accessible medium as described in claim 49 having instructions stored thereon that, when executed, cause a machine to convert the straight matrix to a symmetric matrix to constrain the estimation with the respondent choice data.

56. (canceled)

57. A tangible machine accessible medium as described in claim 40 having instructions stored thereon that, when executed, cause a machine to generate the matrix structure as a geometric matrix having a number of rows equal to the number of products in the set, and a number of columns equal to a number of dimensions.

58. A tangible machine accessible medium as described in claim 57 having instructions stored thereon that, when executed, cause a machine to maintain the number of dimensions to be less than or equal to the number of products in the set.

59. A tangible machine accessible medium as described in claim 57 having instructions stored thereon that, when executed, cause a machine to associate each cell within the geometric matrix with a spatial parameter indicative of a distance between a first and a second product from the number of products.

60. A tangible machine accessible medium as described in claim 59 having instructions stored thereon that, when executed, cause a machine to converge a value of the spatial parameter based on the aggregate model and the respondent choice data.

\* \* \* \* \*