US 20140032264A1

(19) **United States**

(12) **Patent Application Publication**

KALIKIVAYI et al.

(10) **Pub. No.: US 2014/0032264 A1**

(43) **Pub. Date: Jan. 30, 2014**

(54) **DATA REFINING ENGINE FOR HIGH PERFORMANCE ANALYSIS SYSTEM AND METHOD**

(71) Applicant: **INDIX CORPORATION**, BELLEVUE, WA (US)

(72) Inventors: **Satyanarayana Rao KALIKIVAYI**, Chennai (IN); **Mohammed J. ZAHOOR**, Chennai (IN); **Sanjay PARTHASARATHY**, Bellevue, WA (US)

(21) Appl. No.: **13/951,248**

(22) Filed: **Jul. 25, 2013**

Related U.S. Application Data

(60) Provisional application No. 61/675,492, filed on Jul. 25, 2012.

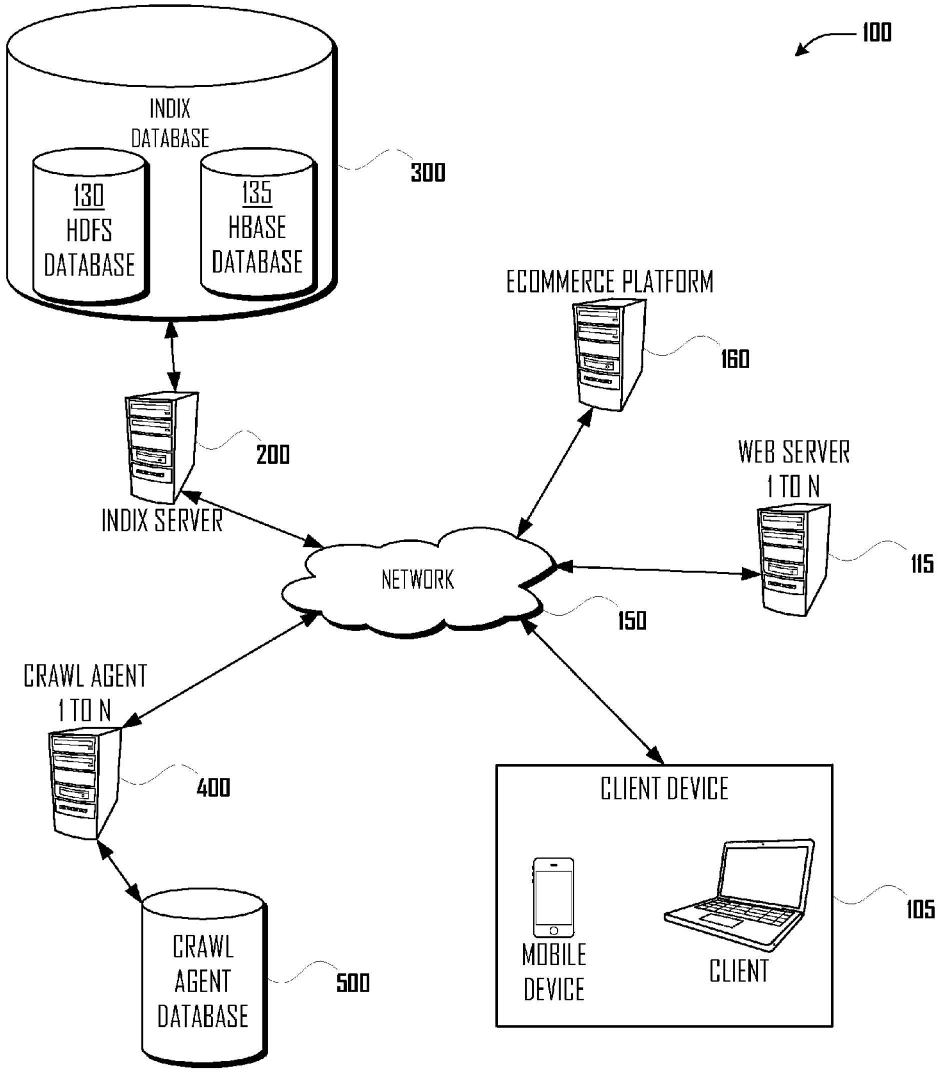
Publication Classification

(51) **Int. Cl.**
G06Q 30/02 (2006.01)

(52) **U.S. Cl.**
CPC *G06Q 30/0201* (2013.01)
USPC **705/7.29**

(57) **ABSTRACT**

Price and product attributes from webpages are analyzed over time to identify price changes specific to products on individual webpages and for products across all webpages as well as to identify longitudinal correlations between price changes and product attributes. Users may search the data and set alerts.



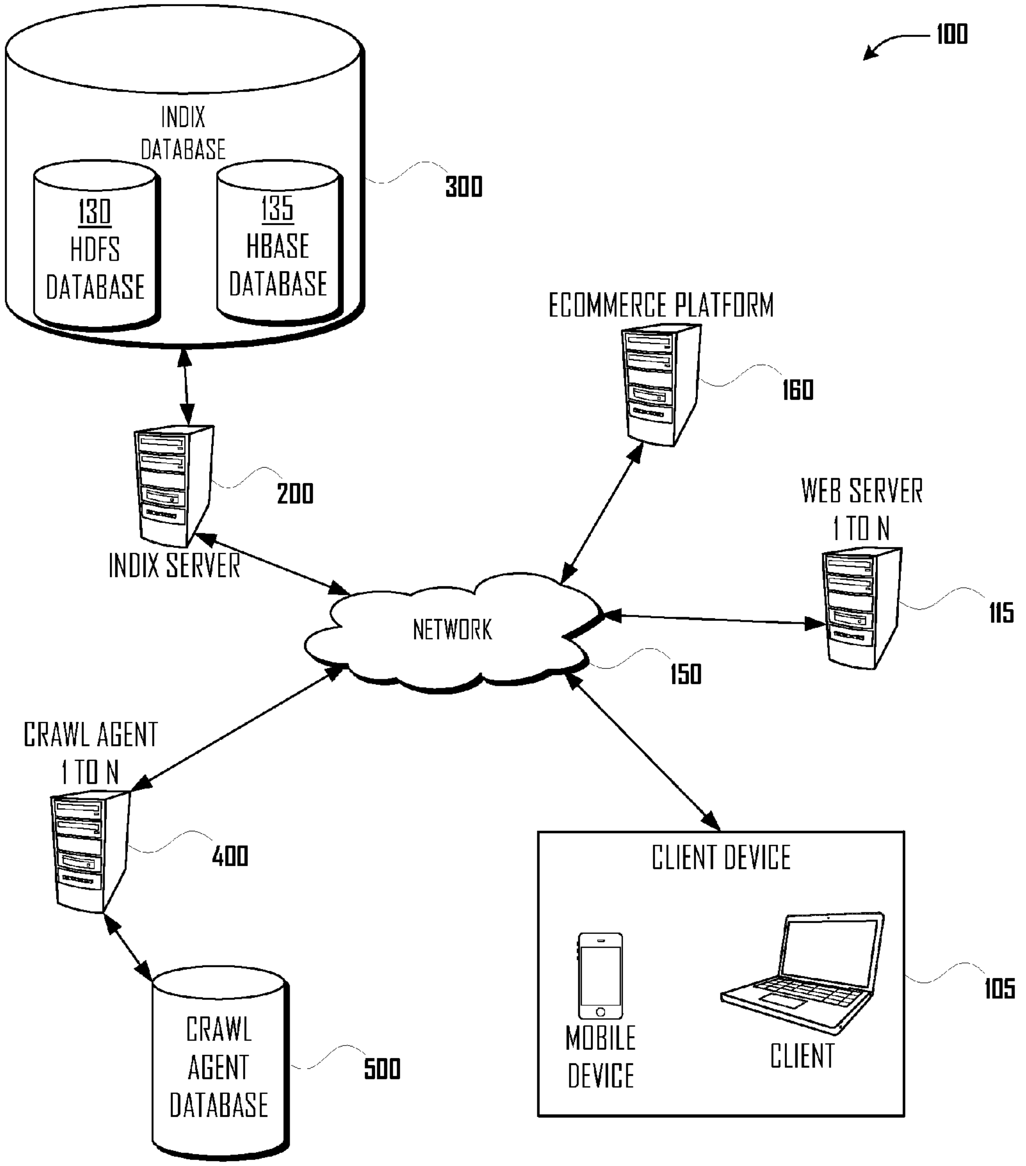

100
INDIX DATABASE
130
HDFS DATABASE
135
HBASE DATABASE
300
ECOMMERCE PLATFORM
160
200
INDIX SERVER
WEB SERVER 1 TO N
115
NETWORK
150
CRAWL AGENT 1 TO N
400
CLIENT DEVICE
MOBILE DEVICE
CLIENT
105
CRAWL AGENT DATABASE
500

Fig. 1

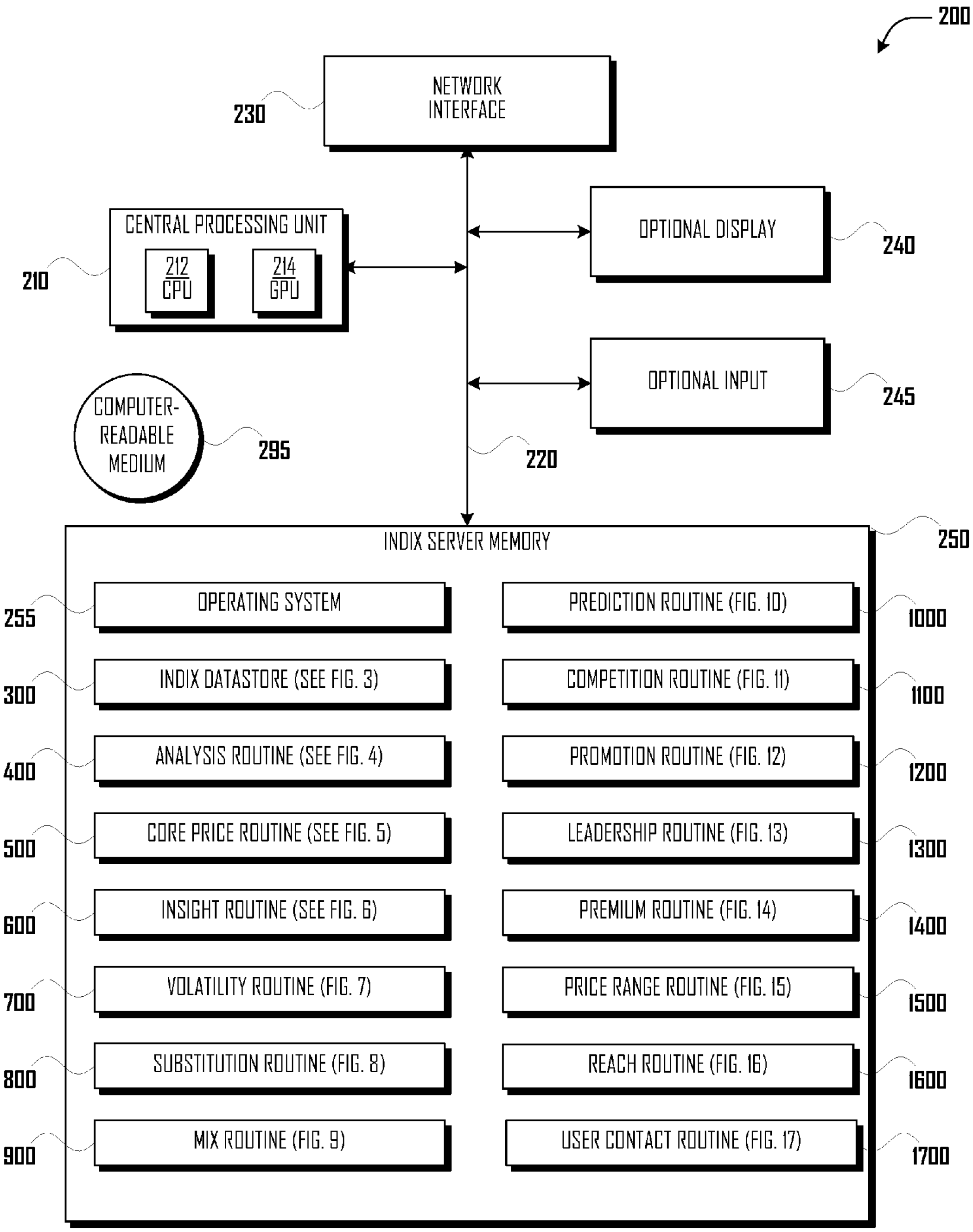

200
NETWORK INTERFACE
230
CENTRAL PROCESSING UNIT
212 CPU
214 GPU
210
OPTIONAL DISPLAY
240
OPTIONAL INPUT
245
COMPUTER-READABLE MEDIUM
295
220
INDIX SERVER MEMORY
250
255
OPERATING SYSTEM
300
INDIX DATASTORE (SEE FIG. 3)
400
ANALYSIS ROUTINE (SEE FIG. 4)
500
CORE PRICE ROUTINE (SEE FIG. 5)
600
INSIGHT ROUTINE (SEE FIG. 6)
700
VOLATILITY ROUTINE (FIG. 7)
800
SUBSTITUTION ROUTINE (FIG. 8)
900
MIX ROUTINE (FIG. 9)
PREDICTION ROUTINE (FIG. 10)
1000
COMPETITION ROUTINE (FIG. 11)
1100
PROMOTION ROUTINE (FIG. 12)
1200
LEADERSHIP ROUTINE (FIG. 13)
1300
PREMIUM ROUTINE (FIG. 14)
1400
PRICE RANGE ROUTINE (FIG. 15)
1500
REACH ROUTINE (FIG. 16)
1600
USER CONTACT ROUTINE (FIG. 17)
1700

Fig.2

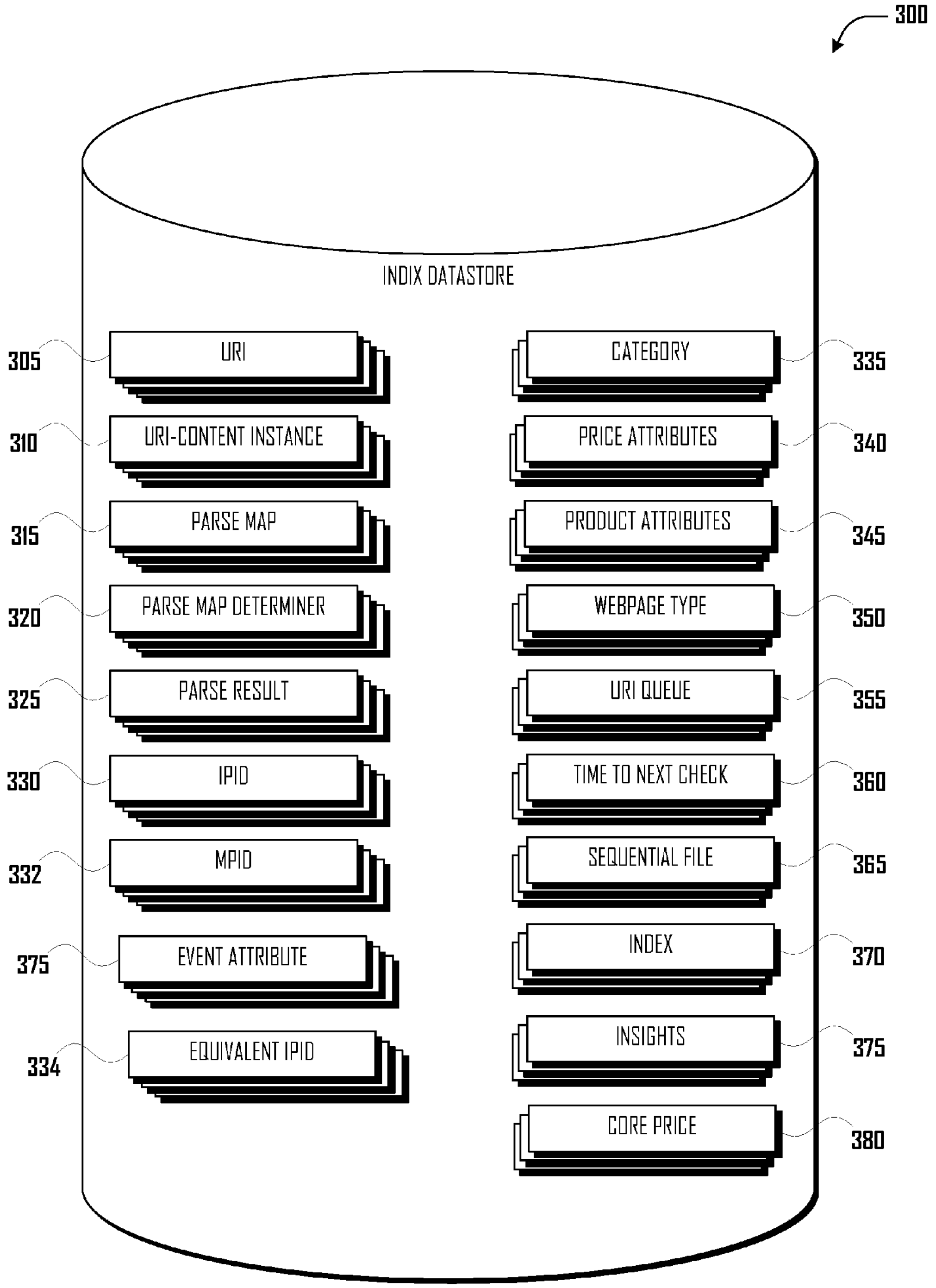

300
INDIX DATASTORE
305
URI
310
URI-CONTENT INSTANCE
315
PARSE MAP
320
PARSE MAP DETERMINER
325
PARSE RESULT
330
IPID
332
MPID
375
EVENT ATTRIBUTE
334
EQUIVALENT IPID
CATEGORY
335
PRICE ATTRIBUTES
340
PRODUCT ATTRIBUTES
345
WEBPAGE TYPE
350
URI QUEUE
355
TIME TO NEXT CHECK
360
SEQUENTIAL FILE
365
INDEX
370
INSIGHTS
375
CORE PRICE
380

Fig.3

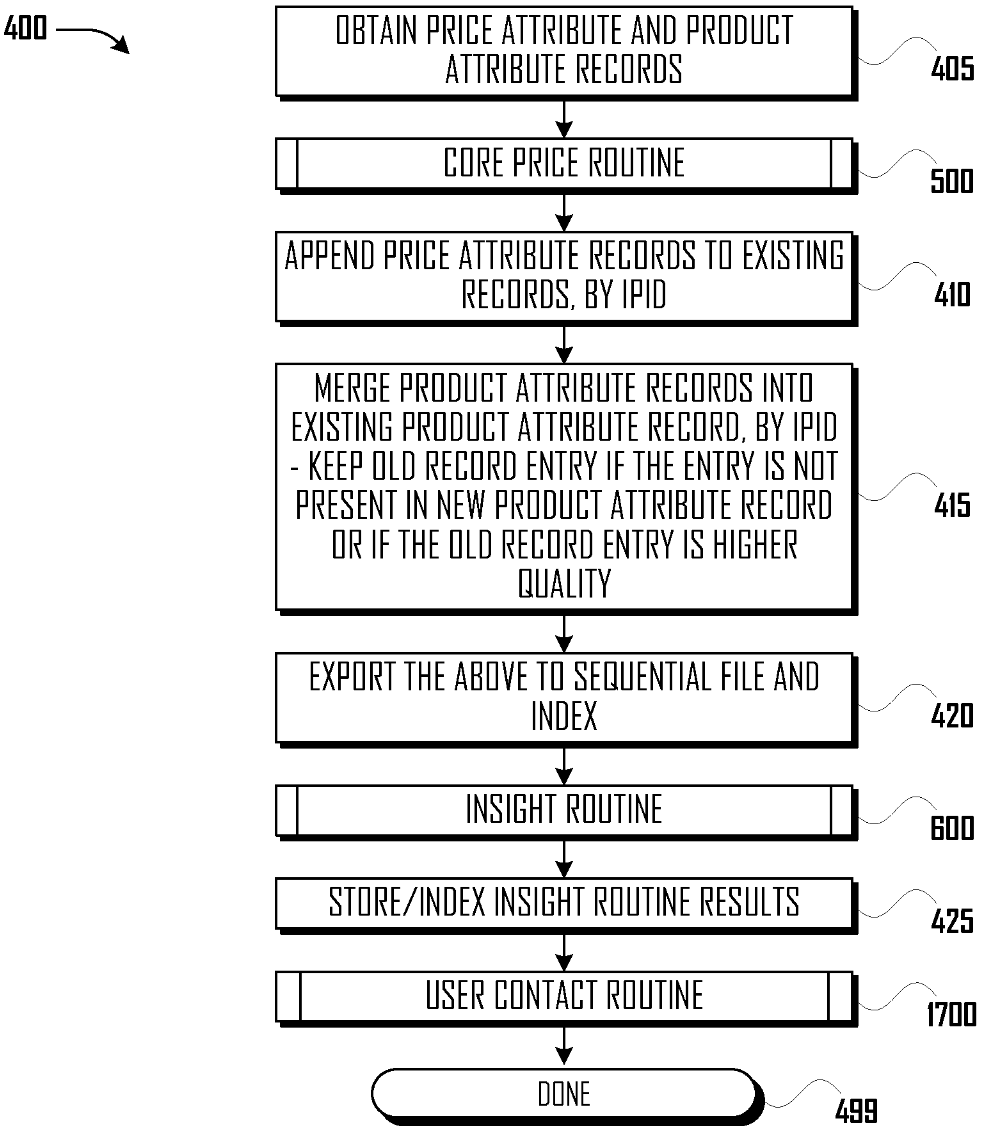

400
OBTAIN PRICE ATTRIBUTE AND PRODUCT ATTRIBUTE RECORDS
405
CORE PRICE ROUTINE
500
APPEND PRICE ATTRIBUTE RECORDS TO EXISTING RECORDS, BY IPID
410
MERGE PRODUCT ATTRIBUTE RECORDS INTO EXISTING PRODUCT ATTRIBUTE RECORD, BY IPID - KEEP OLD RECORD ENTRY IF THE ENTRY IS NOT PRESENT IN NEW PRODUCT ATTRIBUTE RECORD OR IF THE OLD RECORD ENTRY IS HIGHER QUALITY
415
EXPORT THE ABOVE TO SEQUENTIAL FILE AND INDEX
420
INSIGHT ROUTINE
600
STORE/INDEX INSIGHT ROUTINE RESULTS
425
USER CONTACT ROUTINE
1700
DONE
499

Fig. 4

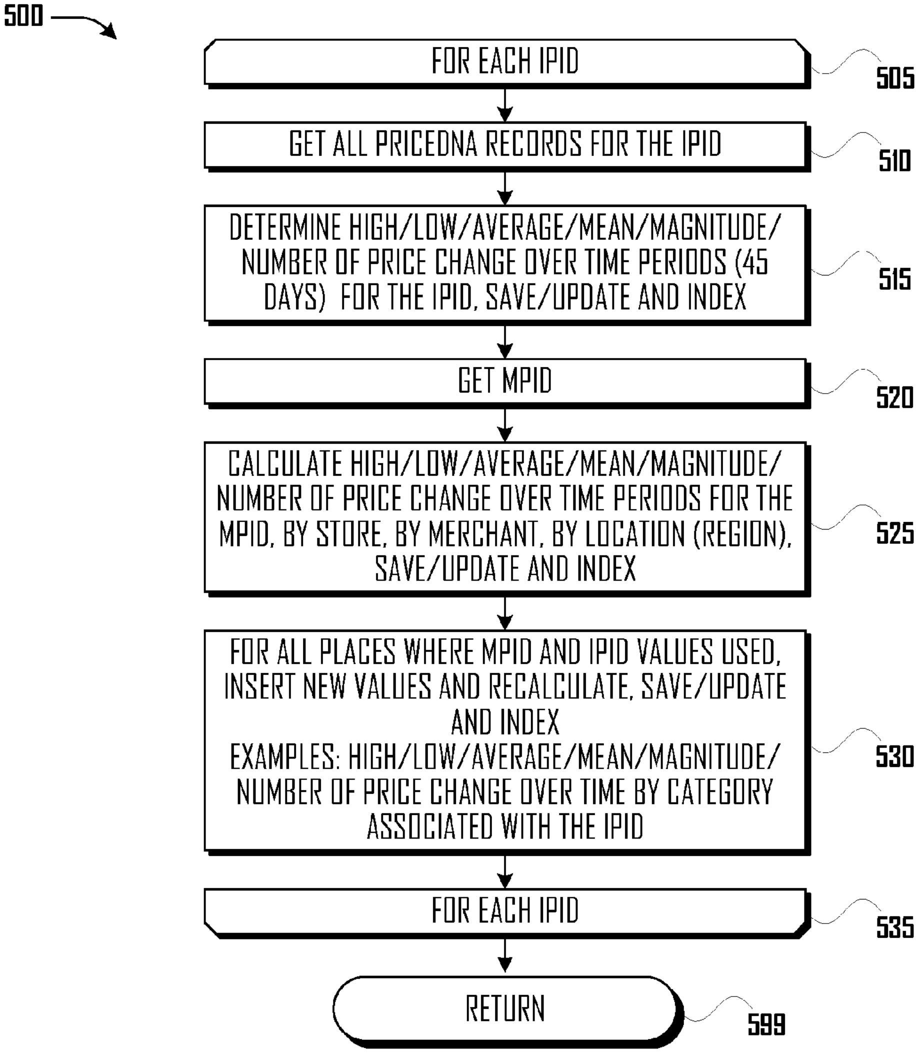

500
FOR EACH IPID
505
GET ALL PRICEDNA RECORDS FOR THE IPID
510
DETERMINE HIGH/LOW/AVERAGE/MEAN/MAGNITUDE/ NUMBER OF PRICE CHANGE OVER TIME PERIODS (45 DAYS) FOR THE IPID, SAVE/UPDATE AND INDEX
515
GET MPID
520
CALCULATE HIGH/LOW/AVERAGE/MEAN/MAGNITUDE/ NUMBER OF PRICE CHANGE OVER TIME PERIODS FOR THE MPID, BY STORE, BY MERCHANT, BY LOCATION (REGION), SAVE/UPDATE AND INDEX
525
FOR ALL PLACES WHERE MPID AND IPID VALUES USED, INSERT NEW VALUES AND RECALCULATE, SAVE/UPDATE AND INDEX
EXAMPLES: HIGH/LOW/AVERAGE/MEAN/MAGNITUDE/ NUMBER OF PRICE CHANGE OVER TIME BY CATEGORY ASSOCIATED WITH THE IPID
530
FOR EACH IPID
535
RETURN
599

Fig. 5

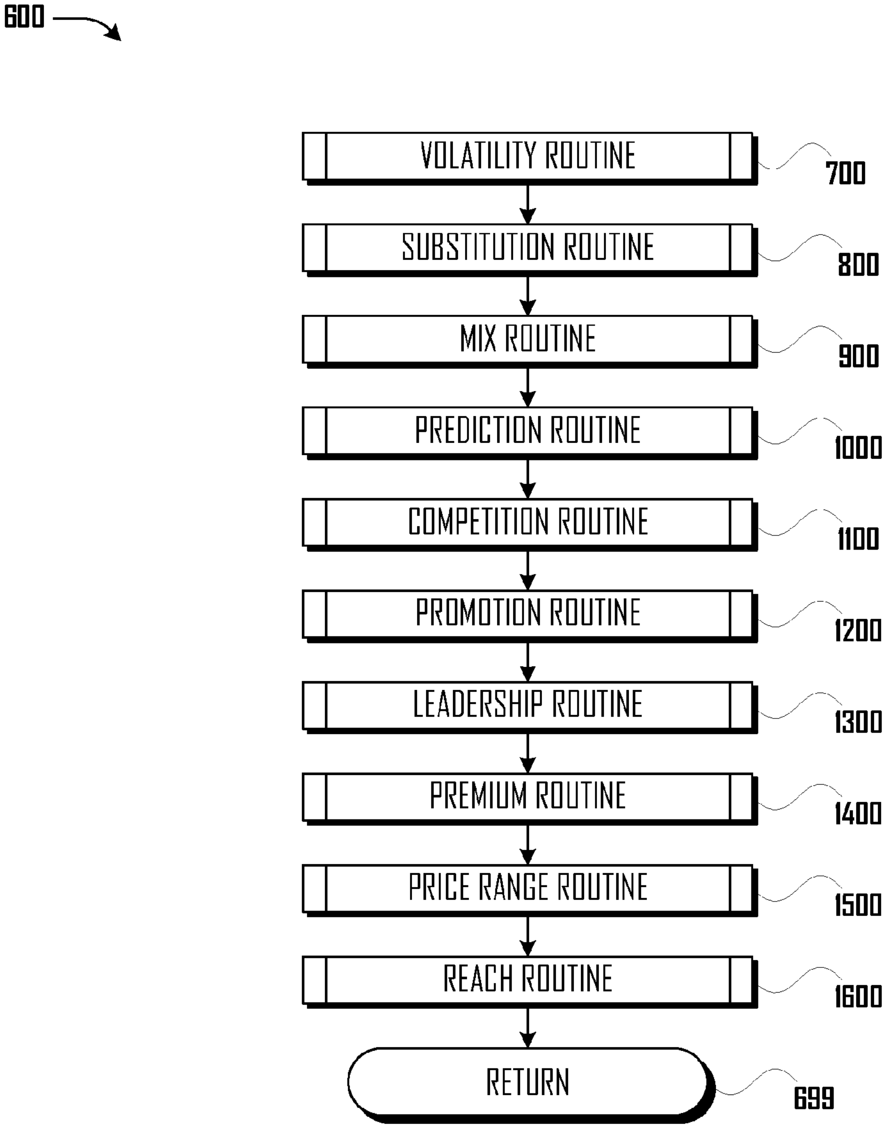

600
VOLATILITY ROUTINE
700
SUBSTITUTION ROUTINE
800
MIX ROUTINE
900
PREDICTION ROUTINE
1000
COMPETITION ROUTINE
1100
PROMOTION ROUTINE
1200
LEADERSHIP ROUTINE
1300
PREMIUM ROUTINE
1400
PRICE RANGE ROUTINE
1500
REACH ROUTINE
1600
RETURN
699

Fig. 6

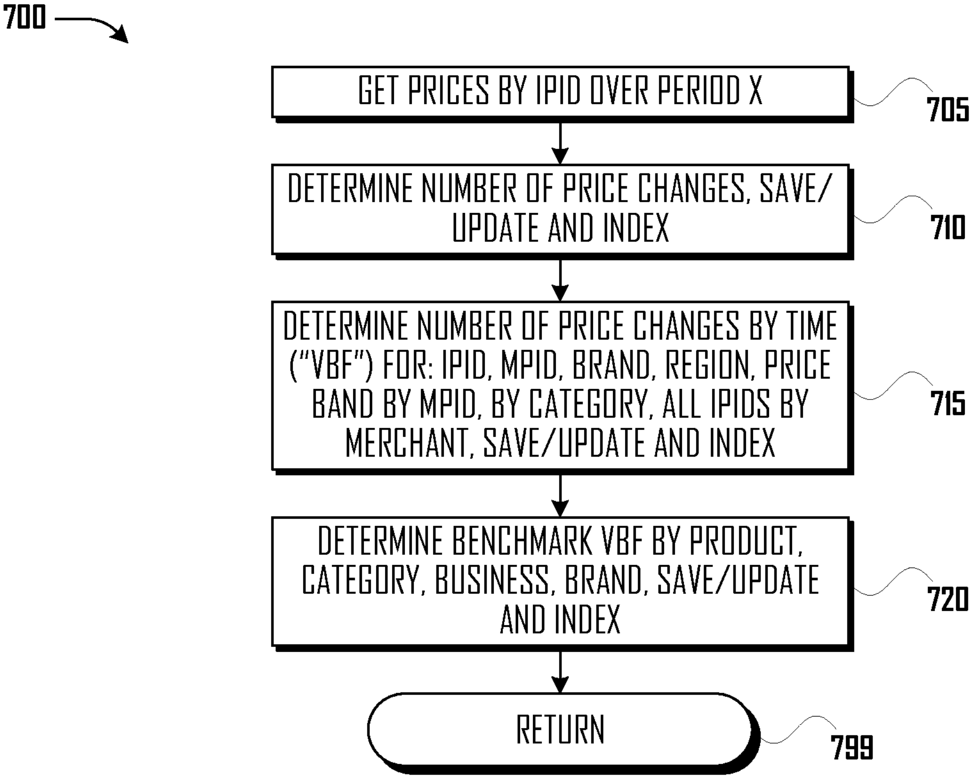

700
GET PRICES BY IPID OVER PERIOD X
705
DETERMINE NUMBER OF PRICE CHANGES, SAVE/ UPDATE AND INDEX
710
DETERMINE NUMBER OF PRICE CHANGES BY TIME ("VBF") FOR: IPID, MPID, BRAND, REGION, PRICE BAND BY MPID, BY CATEGORY, ALL IPIDS BY MERCHANT, SAVE/UPDATE AND INDEX
715
DETERMINE BENCHMARK VBF BY PRODUCT, CATEGORY, BUSINESS, BRAND, SAVE/UPDATE AND INDEX
720
RETURN
799

Fig.7

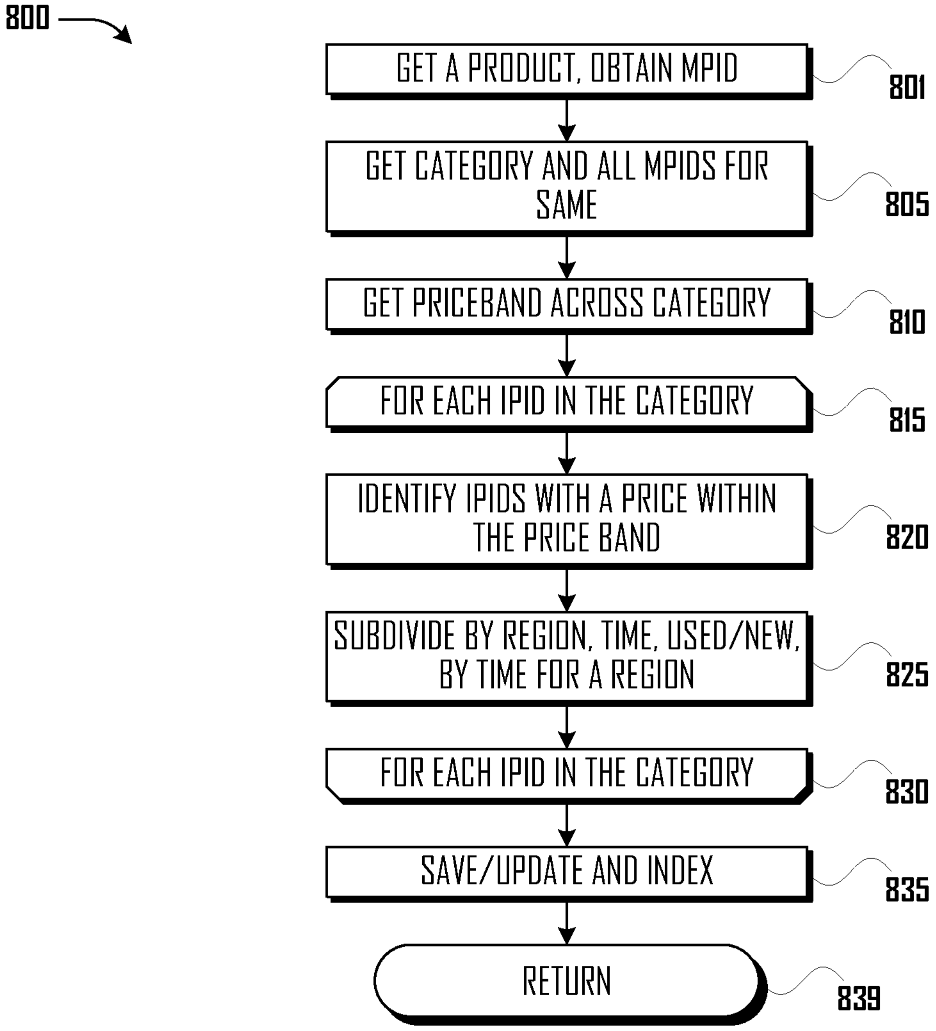

800
GET A PRODUCT, OBTAIN MPID
801
GET CATEGORY AND ALL MPIDS FOR SAME
805
GET PRICEBAND ACROSS CATEGORY
810
FOR EACH IPID IN THE CATEGORY
815
IDENTIFY IPIDS WITH A PRICE WITHIN THE PRICE BAND
820
SUBDIVIDE BY REGION, TIME, USED/NEW, BY TIME FOR A REGION
825
FOR EACH IPID IN THE CATEGORY
830
SAVE/UPDATE AND INDEX
835
RETURN
839

Fig. 8A

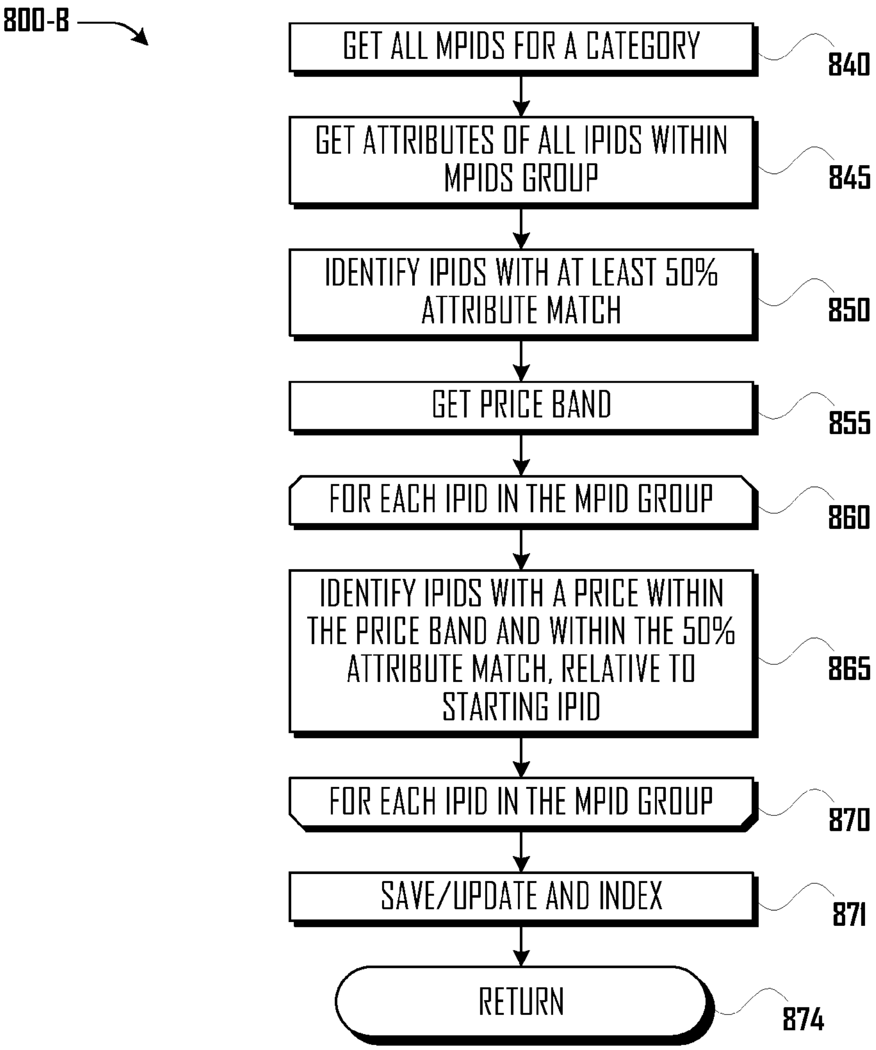

800-B
GET ALL MPIDS FOR A CATEGORY
840
GET ATTRIBUTES OF ALL IPIDS WITHIN MPIDS GROUP
845
IDENTIFY IPIDS WITH AT LEAST 50% ATTRIBUTE MATCH
850
GET PRICE BAND
855
FOR EACH IPID IN THE MPID GROUP
860
IDENTIFY IPIDS WITH A PRICE WITHIN THE PRICE BAND AND WITHIN THE 50% ATTRIBUTE MATCH, RELATIVE TO STARTING IPID
865
FOR EACH IPID IN THE MPID GROUP
870
SAVE/UPDATE AND INDEX
871
RETURN
874

Fig.8B

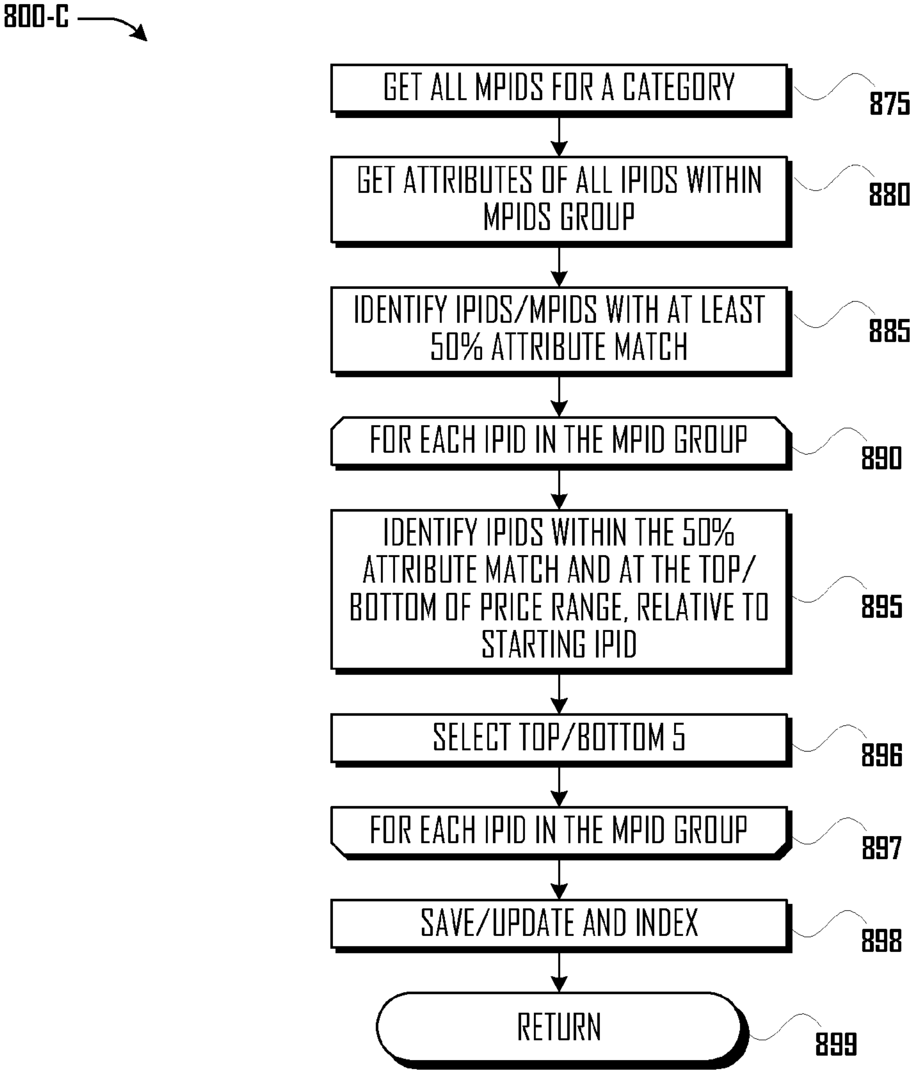

800-C
GET ALL MPIDS FOR A CATEGORY
875
GET ATTRIBUTES OF ALL IPIDS WITHIN MPIDS GROUP
880
IDENTIFY IPIDS/MPIDS WITH AT LEAST 50% ATTRIBUTE MATCH
885
FOR EACH IPID IN THE MPID GROUP
890
IDENTIFY IPIDS WITHIN THE 50% ATTRIBUTE MATCH AND AT THE TOP/ BOTTOM OF PRICE RANGE, RELATIVE TO STARTING IPID
895
SELECT TOP/BOTTOM 5
896
FOR EACH IPID IN THE MPID GROUP
897
SAVE/UPDATE AND INDEX
898
RETURN
899

Fig.8C

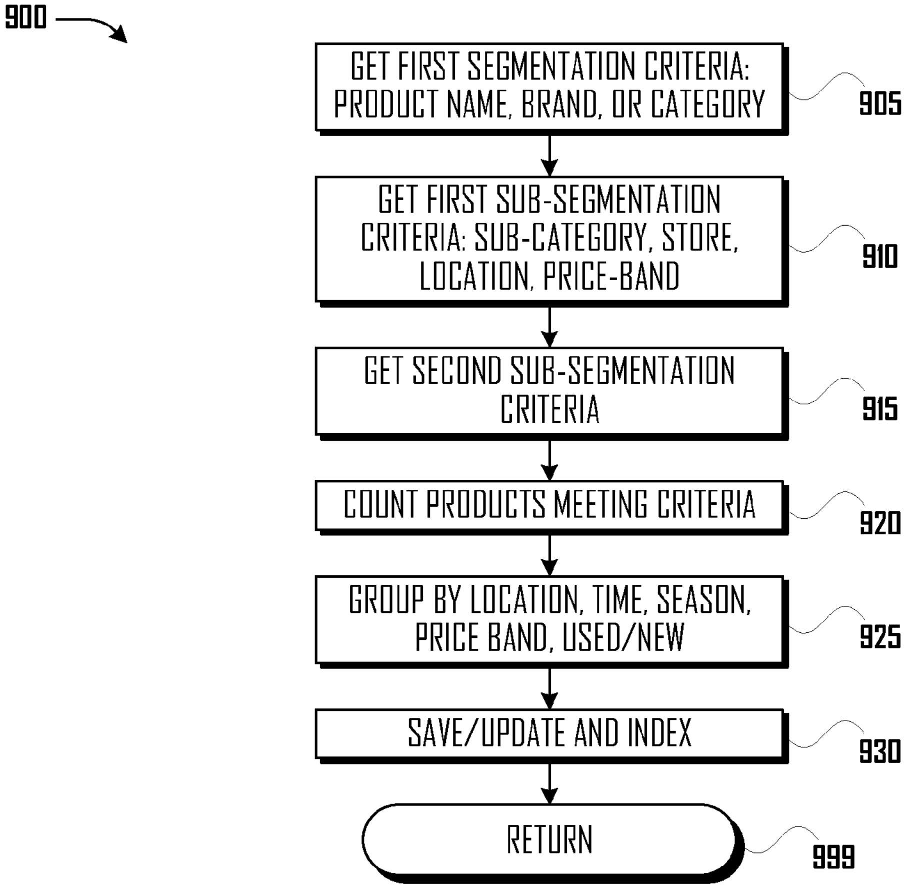

900
GET FIRST SEGMENTATION CRITERIA: PRODUCT NAME, BRAND, OR CATEGORY
905
GET FIRST SUB-SEGMENTATION CRITERIA: SUB-CATEGORY, STORE, LOCATION, PRICE-BAND
910
GET SECOND SUB-SEGMENTATION CRITERIA
915
COUNT PRODUCTS MEETING CRITERIA
920
GROUP BY LOCATION, TIME, SEASON, PRICE BAND, USED/NEW
925
SAVE/UPDATE AND INDEX
930
RETURN
999

Fig. 9

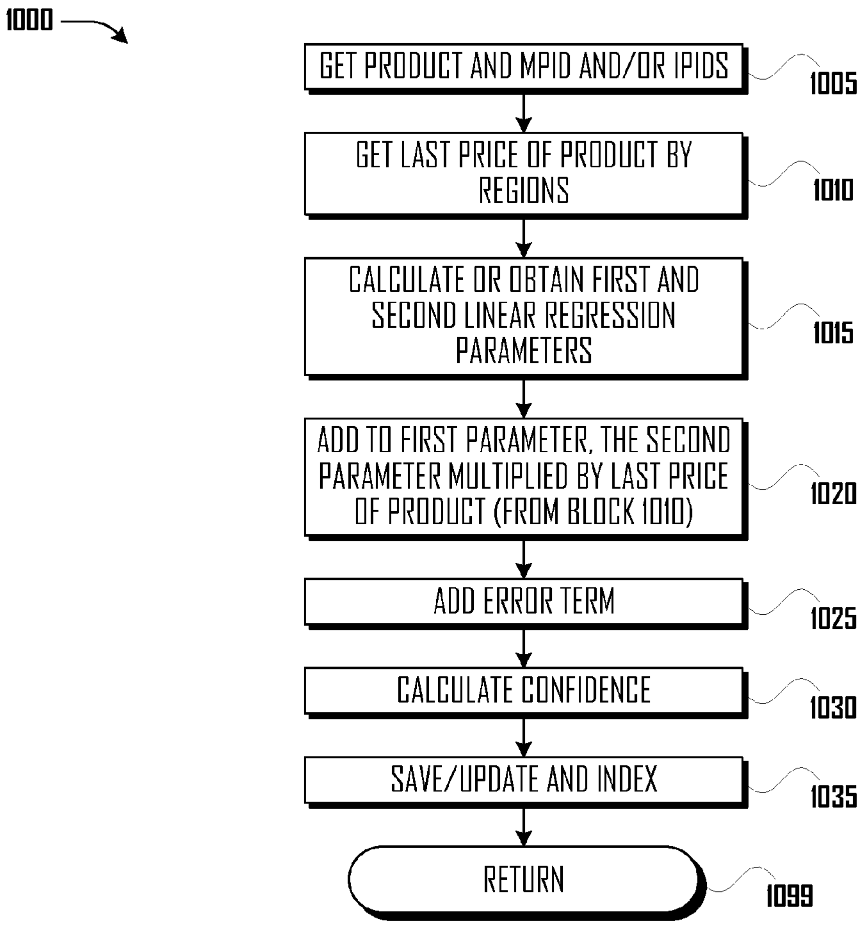

1000
GET PRODUCT AND MPID AND/OR IPIDS
1005
GET LAST PRICE OF PRODUCT BY REGIONS
1010
CALCULATE OR OBTAIN FIRST AND SECOND LINEAR REGRESSION PARAMETERS
1015
ADD TO FIRST PARAMETER, THE SECOND PARAMETER MULTIPLIED BY LAST PRICE OF PRODUCT (FROM BLOCK 1010)
1020
ADD ERROR TERM
1025
CALCULATE CONFIDENCE
1030
SAVE/UPDATE AND INDEX
1035
RETURN
1099

Fig.10

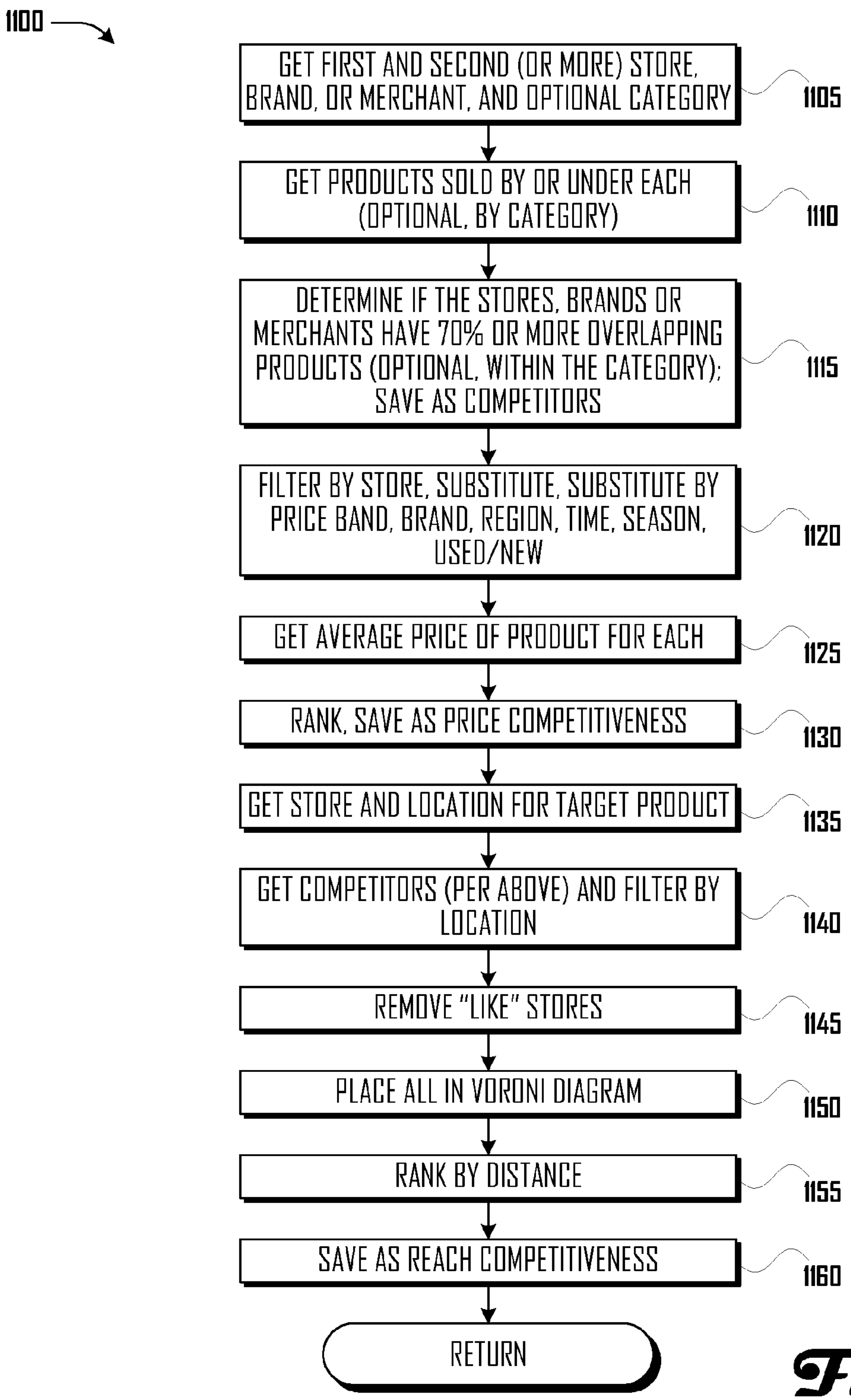

1100
GET FIRST AND SECOND (OR MORE) STORE, BRAND, OR MERCHANT, AND OPTIONAL CATEGORY
1105
GET PRODUCTS SOLD BY OR UNDER EACH (OPTIONAL, BY CATEGORY)
1110
DETERMINE IF THE STORES, BRANDS OR MERCHANTS HAVE 70% OR MORE OVERLAPPING PRODUCTS (OPTIONAL, WITHIN THE CATEGORY); SAVE AS COMPETITORS
1115
FILTER BY STORE, SUBSTITUTE, SUBSTITUTE BY PRICE BAND, BRAND, REGION, TIME, SEASON, USED/NEW
1120
GET AVERAGE PRICE OF PRODUCT FOR EACH
1125
RANK, SAVE AS PRICE COMPETITIVENESS
1130
GET STORE AND LOCATION FOR TARGET PRODUCT
1135
GET COMPETITORS (PER ABOVE) AND FILTER BY LOCATION
1140
REMOVE "LIKE" STORES
1145
PLACE ALL IN VORONI DIAGRAM
1150
RANK BY DISTANCE
1155
SAVE AS REACH COMPETITIVENESS
1160
RETURN

Fig. 11

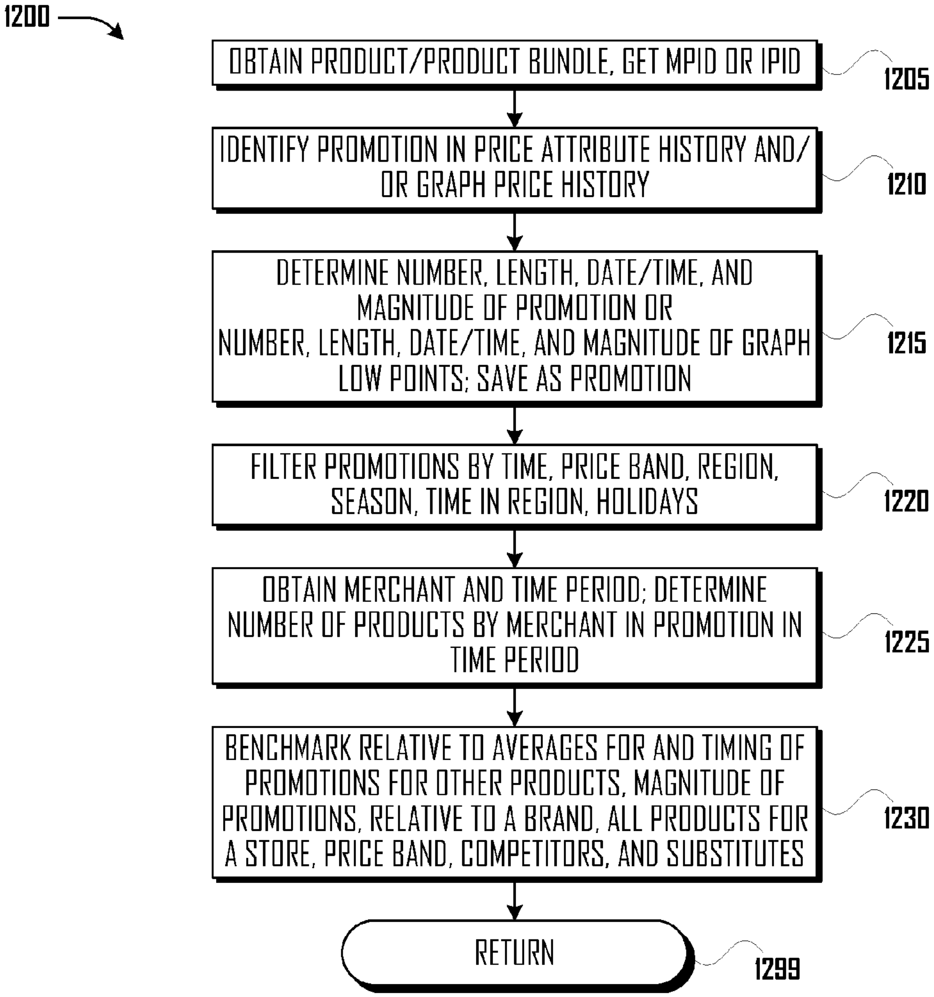

1200
OBTAIN PRODUCT/PRODUCT BUNDLE, GET MPID OR IPID
1205
IDENTIFY PROMOTION IN PRICE ATTRIBUTE HISTORY AND/OR GRAPH PRICE HISTORY
1210
DETERMINE NUMBER, LENGTH, DATE/TIME, AND MAGNITUDE OF PROMOTION OR NUMBER, LENGTH, DATE/TIME, AND MAGNITUDE OF GRAPH LOW POINTS; SAVE AS PROMOTION
1215
FILTER PROMOTIONS BY TIME, PRICE BAND, REGION, SEASON, TIME IN REGION, HOLIDAYS
1220
OBTAIN MERCHANT AND TIME PERIOD; DETERMINE NUMBER OF PRODUCTS BY MERCHANT IN PROMOTION IN TIME PERIOD
1225
BENCHMARK RELATIVE TO AVERAGES FOR AND TIMING OF PROMOTIONS FOR OTHER PRODUCTS, MAGNITUDE OF PROMOTIONS, RELATIVE TO A BRAND, ALL PRODUCTS FOR A STORE, PRICE BAND, COMPETITORS, AND SUBSTITUTES
1230
RETURN
1299

Fig. 12

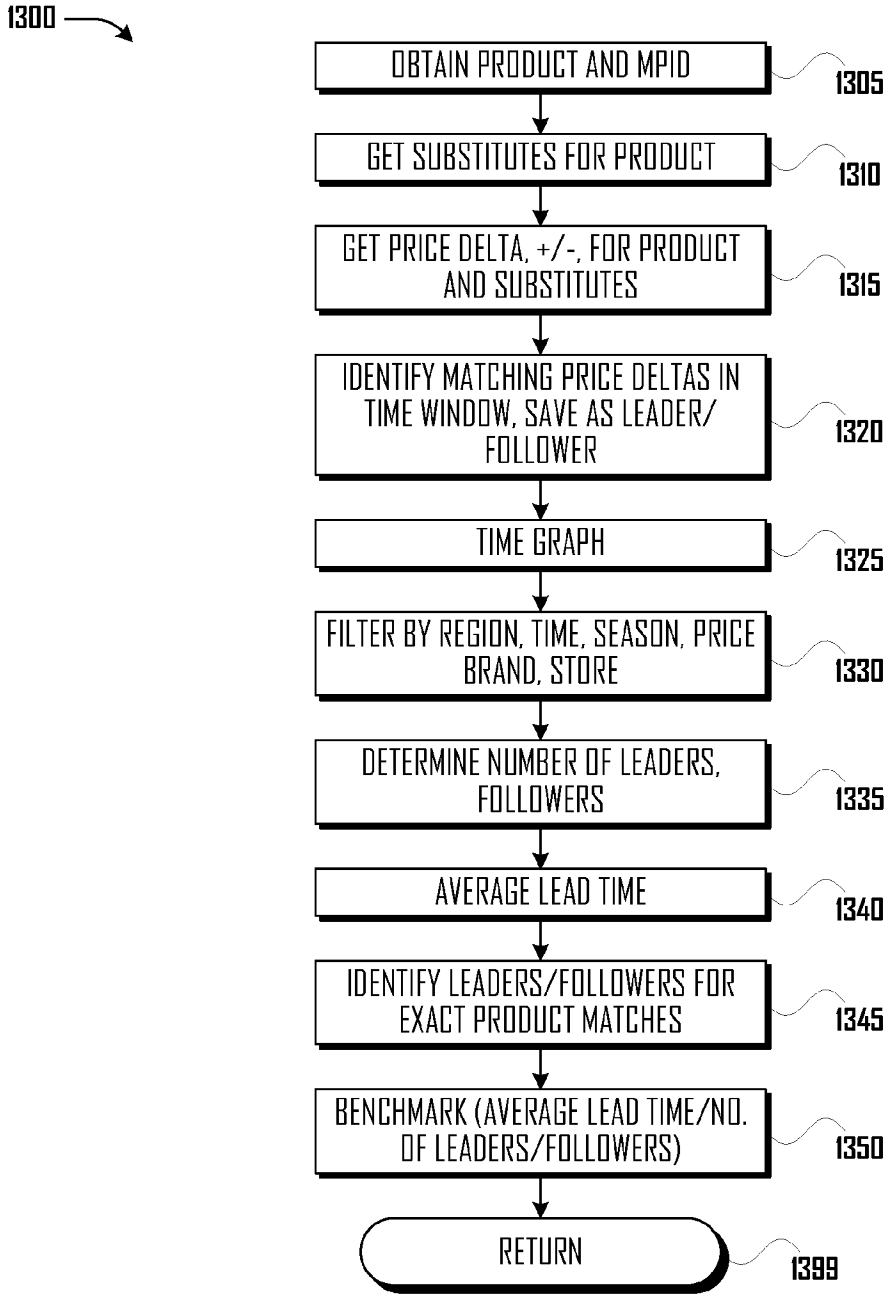

1300
OBTAIN PRODUCT AND MPID
1305
GET SUBSTITUTES FOR PRODUCT
1310
GET PRICE DELTA, +/-, FOR PRODUCT AND SUBSTITUTES
1315
IDENTIFY MATCHING PRICE DELTAS IN TIME WINDOW, SAVE AS LEADER/ FOLLOWER
1320
TIME GRAPH
1325
FILTER BY REGION, TIME, SEASON, PRICE BRAND, STORE
1330
DETERMINE NUMBER OF LEADERS, FOLLOWERS
1335
AVERAGE LEAD TIME
1340
IDENTIFY LEADERS/FOLLOWERS FOR EXACT PRODUCT MATCHES
1345
BENCHMARK (AVERAGE LEAD TIME/NO. OF LEADERS/FOLLOWERS)
1350
RETURN
1399

Fig. 13

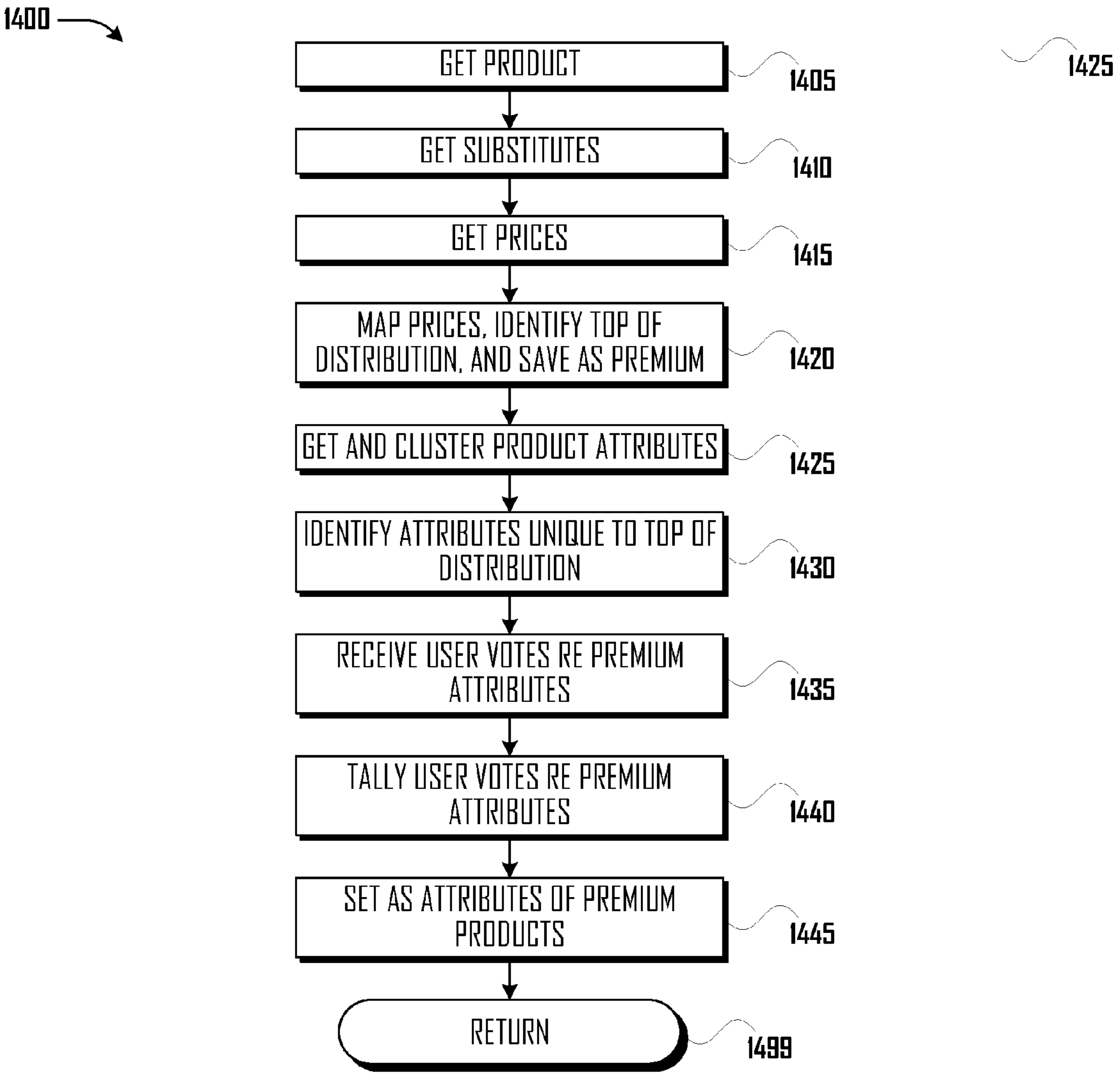

1400
GET PRODUCT
1405
1425
GET SUBSTITUTES
1410
GET PRICES
1415
MAP PRICES, IDENTIFY TOP OF DISTRIBUTION, AND SAVE AS PREMIUM
1420
GET AND CLUSTER PRODUCT ATTRIBUTES
1425
IDENTIFY ATTRIBUTES UNIQUE TO TOP OF DISTRIBUTION
1430
RECEIVE USER VOTES RE PREMIUM ATTRIBUTES
1435
TALLY USER VOTES RE PREMIUM ATTRIBUTES
1440
SET AS ATTRIBUTES OF PREMIUM PRODUCTS
1445
RETURN
1499

Fig.14

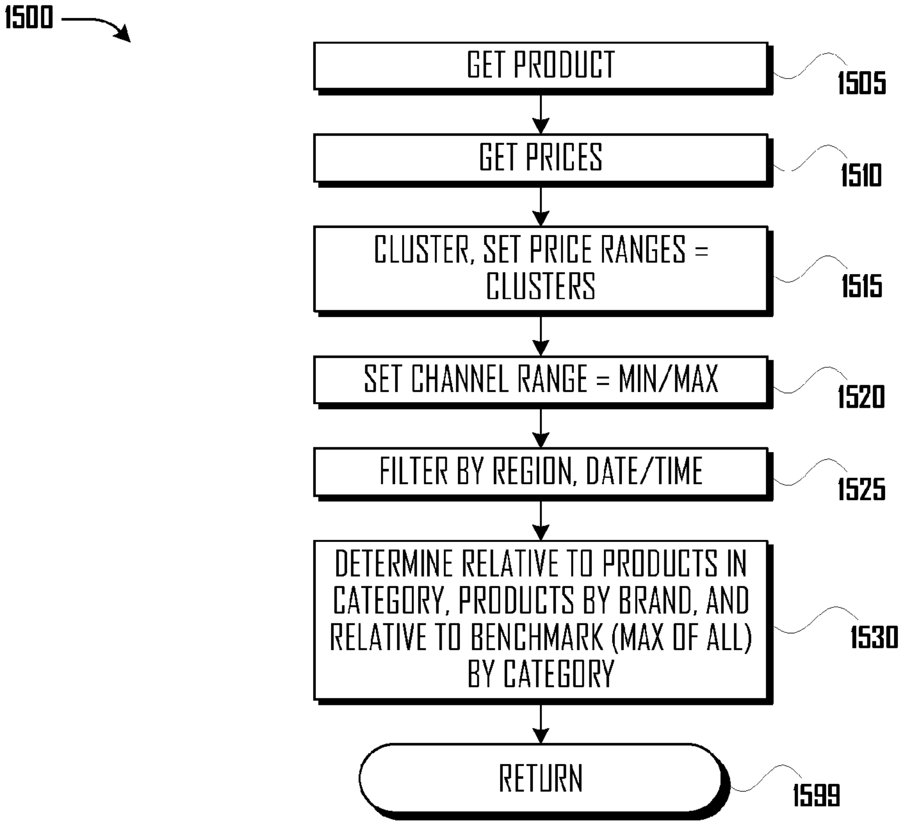

1500
GET PRODUCT
1505
GET PRICES
1510
CLUSTER, SET PRICE RANGES = CLUSTERS
1515
SET CHANNEL RANGE = MIN/MAX
1520
FILTER BY REGION, DATE/TIME
1525
DETERMINE RELATIVE TO PRODUCTS IN CATEGORY, PRODUCTS BY BRAND, AND RELATIVE TO BENCHMARK (MAX OF ALL) BY CATEGORY
1530
RETURN
1599

Fig.15

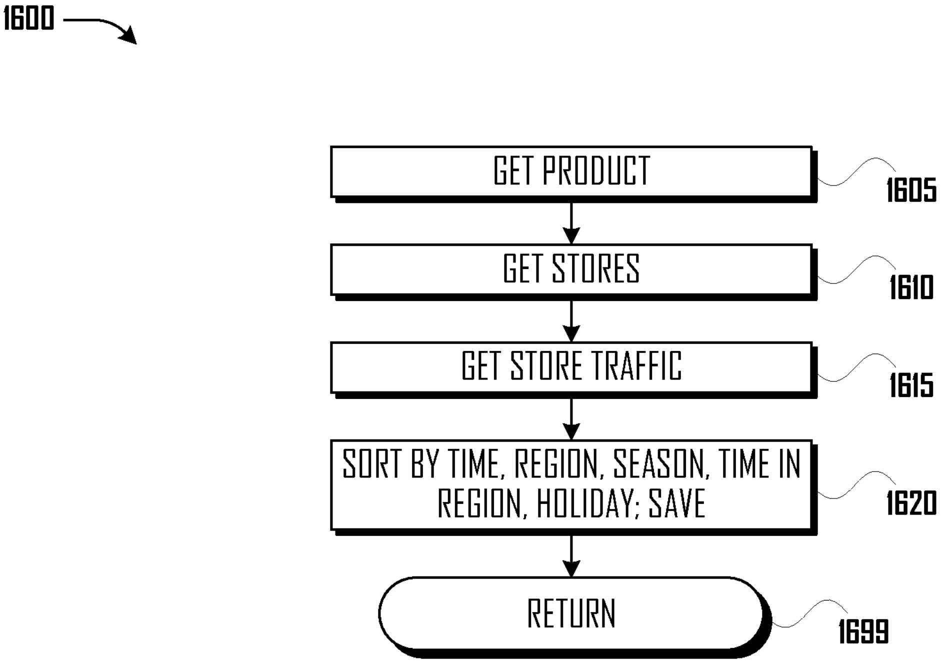

1600
GET PRODUCT
1605
GET STORES
1610
GET STORE TRAFFIC
1615
SORT BY TIME, REGION, SEASON, TIME IN REGION, HOLIDAY; SAVE
1620
RETURN
1699

Fig.16

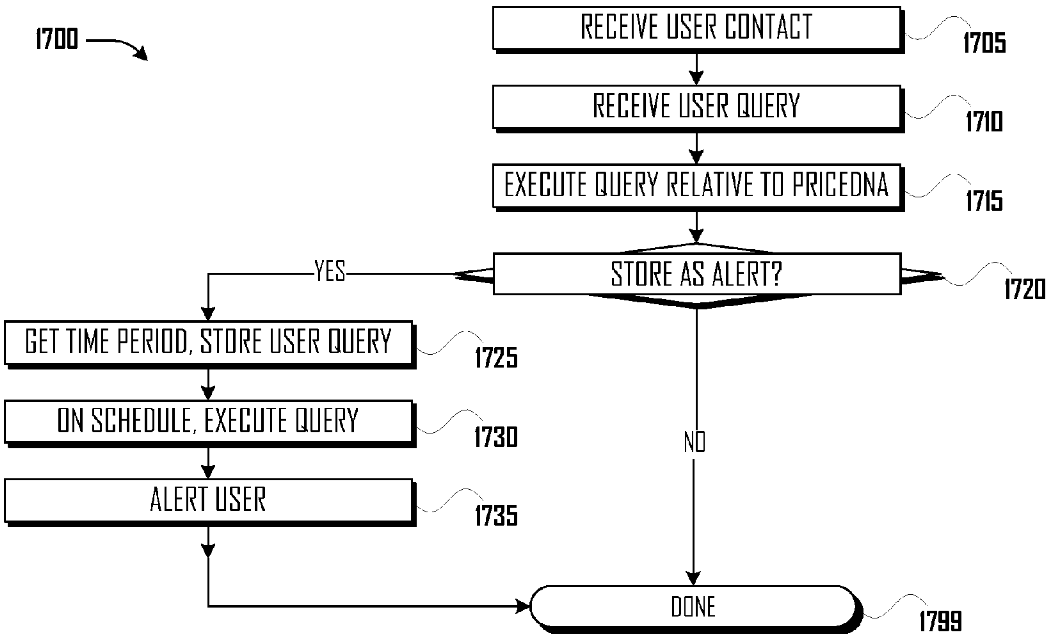

1700
RECEIVE USER CONTACT
1705
RECEIVE USER QUERY
1710
EXECUTE QUERY RELATIVE TO PRICEDNA
1715
STORE AS ALERT?
1720
YES
GET TIME PERIOD, STORE USER QUERY
1725
ON SCHEDULE, EXECUTE QUERY
1730
ALERT USER
1735
NO
DONE
1799

Fig.17

DATA REFINING ENGINE FOR HIGH PERFORMANCE ANALYSIS SYSTEM AND METHOD

CROSS-REFERENCE TO AND INCORPORATION BY REFERENCE OF RELATED APPLICATIONS

[0001] This application claims the benefit of and incorporates by reference U.S. Provisional Patent Application No. 61/675,492, filed on Jul. 25, 2012. This application also incorporates by reference co-pending U.S. patent application Ser. No. ______, filed on Jul. 25, 2013, titled, "Adaptive Gathering of Structured and Unstructured Data System and Method," which application also claims the benefit of U.S. Provisional Patent Application No. 61/675,492.

FIELD

[0002] This disclosure relates to a method and system to analyze price and product information.

BACKGROUND

[0003] The following description includes information that may be useful in understanding the present invention. It is not an admission that any of the information provided herein is prior art or relevant to the presently claimed invention, or that any publication specifically or implicitly referenced is prior art.

[0004] Search engines, such as Google, Bing, and others search and index vast quantities of information on the Internet. "Crawlers" (a.k.a. "spiders") utilize URLs obtained from a "queue" to obtain content, usually from web pages. The crawlers or other software store and index some of the content. Users can then search the indexed content, view results, and follow hyperlinks back to the original source or to the stored content (the stored content often being referred to as a "cache"). Computing resources to crawl and index, however, are not limitless. The URL queues are commonly prioritized to direct crawler resources to web page servers which can accommodate the traffic, which do not block crawlers (such as according to "robots.txt" files commonly available from webpage servers), which experience greater traffic from users, and which experience more change in content.

[0005] Conventional search engines, however, are not focused on price and product information. If a price changes on a webpage, but the rest of the webpage remains the same, traditional crawlers (or the queue manager) will not prioritize the webpage position in the queue, generally because the price is a tiny fraction of the overall content and the change is not labeled as being significant; conversely, if the webpage changes, but the price and/or product information remains the same, the change in webpage content may cause a traditional crawler to prioritize the webpage position in the queue due to the overall change in content, notwithstanding that that price and product information remained the same.

[0006] Conventional search engines, if presented with a query, will find corresponding products. For example, it is possible to search for "men's shoes" and to then be presented with a webpage comprising search results for hundreds of thousands of webpages for men's shoes. The search result may further be narrowed by category of men's shoes, brand, and store. Search engines have been incorporated into online stores, wherein a user may search for products, by keyword and/or by category and results can be ordered by price.

[0007] Price history, however, is only narrowly viewed and, when it is, never in the context of a rich attribute set which explores, in detail, which attributes are associated with changes in price. Price histories are not made available in real time, and do not allow intricate comparisons based on stores, merchants, brands, regions, time/date, and other dimensions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. **1** is a network and device diagram illustrating exemplary computing devices configured according to embodiments disclosed in this paper.

[0009] FIG. **2** is a functional block diagram of an exemplary Indix Server **200** computing device and some data structures and/or components thereof.

[0010] FIG. **3** is a functional block diagram of the Indix Datastore **300** illustrated in the computing device of FIG. **2**.

[0011] FIG. **4** is a flowchart illustrating an embodiment of an Analytics Routine **400**.

[0012] FIG. **5** is a flowchart illustrating an embodiment of a Core Price Routine **500**.

[0013] FIG. **6** is a flowchart illustrating an embodiment of an Insights Routine **600**.

[0014] FIG. **7** is a flowchart illustrating an embodiment of a Volatility Routine **700**.

[0015] FIGS. **8**A-**8**C are flowcharts illustrating embodiments of a Substitution Routine **800**.

[0016] FIG. **9** is a flowchart illustrating an embodiment of a Mix Routine **900**.

[0017] FIG. **10** is a flowchart illustrating an embodiment of a Prediction Routine **1000**.

[0018] FIG. **11** is a flowchart illustrating an embodiment of a Competition Routine **1100**.

[0019] FIG. **12** is a flowchart illustrating an embodiment of a Promotion Routine **1200**.

[0020] FIG. **13** is a flowchart illustrating an embodiment of a Leadership Routine **1300**.

[0021] FIG. **14** is a flowchart illustrating an embodiment of a Premium Routine **1400**.

[0022] FIG. **15** is a flowchart illustrating an embodiment of a Price Range Routine **1500**.

[0023] FIG. **16** is a flowchart illustrating an embodiment of a Reach Routine **1600**.

[0024] FIG. **17** is a flowchart illustrating an embodiment of a User Contact Routine **1700**.

DETAILED DESCRIPTION

[0025] The following Detailed Description provides specific details for an understanding of various examples of the technology. One skilled in the art will understand that the technology may be practiced without many of these details. In some instances, structures and functions have not been shown or described in detail or at all to avoid unnecessarily obscuring the description of the examples of the technology. It is intended that the terminology used in the description presented below be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain examples of the technology. Although certain terms may be emphasized below, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section.

[0026] Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense, as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to." As used herein, the term "connected," "coupled," or any variant thereof means any connection or coupling, either direct or indirect between two or more elements; the coupling of connection between the elements can be physical, logical, or a combination thereof. Additionally, the words, "herein," "above," "below," and words of similar import, when used in this application, shall refer to this application as a whole and not to particular portions of this application. When the context permits, words using the singular may also include the plural while words using the plural may also include the singular. The word "or," in reference to a list of two or more items, covers all of the following interpretations of the word: any of the items in the list, all of the items in the list, and any combination of one or more of the items in the list.

[0027] Certain elements appear in various of the Figures with the same capitalized element text, but a different element number. When referred to herein with the capitalized element text but with no element number, these references should be understood to be largely equivalent and to refer to any of the elements with the same capitalized element text, though potentially with differences based on the computing device within which the various embodiments of the element appears.

[0028] As used herein, a Uniform Resource Identifier ("URI") is a string of characters used to identify a resource on a computing device and/or a network, such as the Internet. Such identification enables interaction with representations of the resource using specific protocols. "Schemes" specifying a syntax and associated protocols define each URI.

[0029] The generic syntax for URI schemes is defined in Request for Comments ("RFC") memorandum 3986 published by the Internet Engineering Task Force ("IETF"). According to RFC 3986, a URI (including a URL) consists of four parts:

[0030] <scheme name>: <hierarchical part> [?<query>] [#<fragment>]

[0031] A URI begins with a scheme name that refers to a specification for assigning identifiers within that scheme. The scheme name consists of a letter followed by any combination of letters, digits, and the plus ("+"), period ("."), or hyphen ("-") characters; and is terminated by a colon (":").

[0032] The hierarchical portion of the URI is intended to hold identification information that is hierarchical in nature. Often this part is delineated with a double forward slash ("//"), followed by an optional authority part and an optional path.

[0033] The optional authority part holds an optional user information part (not shown) terminated with "@" (e.g. username:password@), a hostname (i.e., domain name or IP address, here "example.com"), and an optional port number preceded by a colon ":".

[0034] The path part is a sequence of one or more segments (conceptually similar to directories, though not necessarily representing them) separated by a forward slash ("/"). If a URI includes an authority part, then the path part may be empty.

[0035] The optional query portion is delineated with a question mark and contains additional identification information that is not necessarily hierarchical in nature. Together, the path part and the query portion identify a resource within the scope of the URI's scheme and authority.

[0036] The query string syntax is not generically defined, but is commonly organized as a sequence of zero or more <key>=<value> pairs separated by a semicolon or ampersand, for example:

[0037] key1=value1;key2=value2;key3=value3 (Semicolon), or

[0038] key1=value1&key2=value2&key3=value3 (Ampersand)

[0039] Much of the above information is taken from RFC 3986, which provides additional information related to the syntax and structure of URIs. RFC 3986 is hereby incorporated by reference, for all purposes.

[0040] As used herein, "Product" shall be understood to mean "products or services." References to "Product Attribute" herein shall be understood to mean "product or service attribute." As used herein, "Products" are associated with iPIDs.

[0041] As used herein, an "iPID" or iPID **330** is a unique identifier assigned within the Indix System to a URI for a product, such as URI **305**. The iPID **330** may be, for example, a hash of URI **305**. When multiple URIs **305** from a common base domain name lead to webpages which, when parsed for Price and Product Attributes and product the same Parse Result **325** (notwithstanding that the webpages may contain other Content which does not contribute to the Parse Result **325**) may be labeled as equivalent in, for example, the Equivalent iPID **334** record and may be treated as the same iPID **330**.

[0042] As used herein, a "Master iPID" or "MPID" or MPID **332** is an iPID **330** assigned to a group of iPIDs **330** derived from URIs **305** which lead to webpages offering the same Product for sale. An MPID is generally meant to identify a single Product, generally produced by a common manufacturer, though the Product may be distributed and sold by multiple parties.

[0043] iPIDs and MPIDSs are associated with Price Attribute **340** records and Product Attribute **345** records.

[0044] A Price Attribute **340** record may comprise one or more records comprising, for example, values which encode an iPRID which may be an identifier for a price observed at a particular time, an iPID (discussed above), a Product Name (a "Product Name" value in this record may also be referred to herein as a "Product"), a Standard Price, a Sale, a Price, a Rebate amount, a Price Instructions record (containing special instructions relating to a price, such as that the price only applies to students), a Currency Type, a Date and Time Stamp, a Tax record, a Shipping record (indicating costs relating to shipping to different locations, whether tax is calculated on shipping costs, etc.), a Price Validity Start Date, a Price Validity End Date, a Quantity, a Unit of Measure Type, a Unit of Measure Value, a Merchant Name (with the name of a merchant from whom the Product is available; a "Merchant Name" value in this record may also be referred to herein as a "Merchant"), a Store Name (a Merchant may have multiple stores; a "Store Name" value in this record may also be referred to herein as a "Store"), a User ID, a Data Channel (indicating the source of the Price Attribute **340** record, such as an online crawl, a crowdsource, a licensed supplier of price information, or from a merchant), a Source Details record (for example, indicating a URI, a newspaper advertisement), an Availability Flag, a Promotion Code, a Bundle Details record (indicating products which are part of a bundle), a Condition Type record (indicating new, used, poor, good, and similar), a Social Rank record (indicating a rank of "likes" and similar of the price), a Votes/Likes record (indicating a number of "likes" and similar which a Price or Product has received), a Price Rank record, a Visibility Indicator record (indicating whether the price is visible to the public, whether it is only visible to a Merchant, or the like), a Supply Chain Reference record (indicating whether the price was obtained from a retailer, a wholesaler, or another party in a supply chain), a Sale Location (indicating a geographic location where the product is available at the price), a Manufactured Location record (indicating where the product was produced or manufactured), a Launch Date record (indicating how long the product has been on the market), and an Age of Product record (indicating how long the product was used by the user). When capitalized herein, the foregoing terms (such as Product, Price, Merchant, Store, Source Details, etc.) are meant to refer to values in a Price Attribute **340** record.

[0045] A Product Attribute **345** record may comprise, for example, values encoding features of or describing a Product. The entire Product Attribute **345** schema may comprise thousands of columns, though only tens or hundreds of the columns may be applicable to any given Product. An example set of values in a Product Attribute **345** record for a ring is as follows: Title, "Sterling Silver Diamond & Blue Topaz Ring;" Brand, "Blue Nile;" Category (such as, for example, a Category **335** in a category schema), "rings;" Metal Name, "silver;" Stone Shape, "cushion;" Stone Name, "topaz;" Width, "3 mm;" Stone Color, "blue;" Product Type, "rings," Birthstone, "September;" and Setting Type, "prong." An example set of Product Attributes **345** for a shoe is as follows: Brand, "Asics;" Category (such as, for example, a Category **335** in a category schema or taxonomy), "Men's Sneakers & Athletic;" Shoe Size, "8;" Product Type, "wrestling shoes," Color, "black;" Shoe Style, "sneakers;" Sports, "athletic;" Upper Material, "mesh." When capitalized herein, the foregoing terms (such as Brand, Category, Metal Name, Product Type, etc.) are meant to refer to values in a Product Attribute **345** record.

[0046] As used herein, "Content" comprises text, graphics, images (including still and video images), audio, graphical arrangement, and instructions for graphical arrangement, including HTML and CSS instructions which may, for example, be interpreted by browser applications.

[0047] As used herein, "Event" is information generally in news or current events. Events may be found in Content. Listing Pages, Product Pages, and Event Pages are all examples of Webpage Types **350**.

[0048] As used herein, "PriceDNA" comprises a Product Attribute **345** record, one or more Price Attribute **340** records, the output of the Core Price Routine **500** (generally found in the Core Price **380** records), and the output of the Insights **600** routine (generally found in the Insights **375** records).

[0049] As used herein, a "Brand" is a family or group of Products sold by or under a common trademark, such as the "Nike®" Brand, which sells under this trademark a family of shoes, exercise equipment, and other apparel. Brand is a value within a Product Attribute **345** record.

[0050] As used herein, a "Store" is an online or physical sales venue. A Store is a value within a Price Attribute **340** record.

[0051] As used herein, a "Merchant" is an operator of one or more Stores. A Merchant is a value in a Price Attribute **340** record.

[0052] Generally, an Analysis Routine **400** obtains Price Attribute **340** and Product Attribute **345** records from the Indix Database **300** shortly after the records are produced following a crawl of webpages accessed via the URIs **305**. The Analysis Routine **400** merges the records, performs a Core Price Routine **500** to develop core price information, such as changes in price, and exports the records and the result to the Core Price Routine **500** to a sequential file which is indexed. The result of the Core Price Routine **500** may be searched and accessed by users in close to real-time. The Analysis Routine **400** also performs an Insight Routine **600**. The Insight Routine **600** comprises a set of sub-routines for deriving additional information from the Price Attribute **340** and Product Attribute **345** records and from the output of the Core Price Routine **500**. Generally, the Insight Routine **600** identifies what Product Attributes **345** and Price Attributes **340** across the datasets are associated with the changes in price. The output of the Insight Routine **600** is also stored in the Indix Database **300** and may be searched and accessed by users, though the accessible values may be refreshed more slowly than the data from the Core Price Routine **500**. A User Contact Routine **1700** allows users to search and obtain information and to set alerts relative to the information in the Indix Database **300**.

[0053] FIG. **1** is a network and device diagram illustrating exemplary computing devices configured according to embodiments disclosed in this paper. Illustrated in FIG. **1** are an Indix Server **200** and an Indix Database **300**. The Indix Database **300** is discussed further in relation to FIG. **3**.

[0054] Also illustrated in FIG. **1** is a Crawl Agent **400**, representing Crawl Agents **1** to N, and a Crawl Agent Database **500**. The Crawl Agent **400** and Crawl Agent Database **500** are used to crawl webpages accessed via the URIs **305**.

[0055] Also illustrated in FIG. **1** is a Client Device **105**, such as a mobile or non-mobile computer device. The Client Device **105** is an example of computing devices such as, for example, a mobile phone, a tablet, laptop, personal computer, gaming computer, or media playback computer. The Client Device **105** represents any computing device capable of rendering Content in a browser or an equivalent user-interface. Client Devices are used by "users." The Client Device **105** may interact with the User Contact Routine **1700**.

[0056] Also illustrated in FIG. **1** is a Web Server **115**, which may serve Content in the form of webpages or equivalent output in response to URIs, such as URI **305**.

[0057] Also illustrated in FIG. **1** is an Ecommerce Platform **160**, which may provide ecommerce services, such as website and/or webpage hosting via webpage templates comprising HTML and CSS elements. Customers of Ecommerce Platform **160** may complete the webpage templates with Content and serve the webpages and websites from, for example, Web Server **115**.

[0058] Interaction among devices illustrated in FIG. **1** may be accomplished, for example, through the use of credentials to authenticate and authorize a machine or user with respect to other machines.

[0059] In FIG. **1**, the computing machines may be physically separate computing devices or logically separate processes executed by a common computing device. Certain components are illustrated in FIG. **1** as connecting directly to one another (such as, for example, the Indix Database **300** to the Indix Server **200**), though the connections may be through the Network **150**. If these components are embodied in separate computers, then additional steps may be added to the disclosed invention to recite communicating between the components.

[0060] The Network **150** comprises computers, network connections among the computers, and software routines to enable communication between the computers over the network connections. Examples of the Network **150** comprise an Ethernet network, the Internet, and/or a wireless network, such as a GSM, TDMA, CDMA, EDGE, HSPA, LTE or other network provided by a wireless service provider, or a television broadcast facility. Connection to the Network **150** may be via a Wi-Fi connection. More than one network may be involved in a communication session between the illustrated devices. Connection to the Network **150** may require that the computers execute software routines which enable, for example, the seven layers of the OSI model of computer networking or equivalent in a wireless phone network.

[0061] This paper may discuss a first computer as connecting to a second computer (such as a Crawl Agent **400** connecting to the Indix Server **200**) or to a corresponding datastore (such as to Indix Database **300**); it should be understood that such connections may be to, through, or via the other of the two components (for example, a statement that a computing device connects with or sends data to the Indix Server **200** should be understood as saying that the computing device may connect with or send data to the Indix Database **300**). References herein to "database" should be understood as equivalent to "datastore." Although illustrated as components integrated in one physical unit, the computers and databases may be provided by common (or separate) physical hardware and common (or separate) logic processors and memory components. Though discussed as occurring within one computing device, the software routines and data groups used by the software routines may be stored and/or executed remotely relative to any of the computers through, for example, application virtualization.

[0062] FIG. **2** is a functional block diagram of an exemplary Indix Server **200** computing device and some data structures and/or components thereof. The Indix Server **200** in FIG. **2** comprises at least one Processing Unit **210**, Indix Server Memory **250**, a Display **240** and Input **245**, all interconnected along with the Network Interface **230** via a Bus **220**. The Processing Unit **210** may comprise one or more general-purpose Central Processing Units ("CPU") **212** as well as one or more special-purpose Graphics Processing Units ("GPU") **214**. The components of the Processing Unit **210** may be utilized by the Operating System **255** for different functions required by the routines executed by the Indix Server **200**. The Network Interface **230** may be utilized to form connections with the Network **150** or to form device-to-device connections with other computers. The Indix Server Memory **250** generally comprises a random access memory ("RAM"), a read only memory ("ROM"), and a permanent mass storage device, such as a disk drive or SDRAM (synchronous dynamic random-access memory).

[0063] The Indix Server Memory **250** stores program code for software routines, such as, for example, Analysis Routine **400**, Core Price Routine **500**, Insight Routine **600**, Volatility Routine **700**, Substitution Routine **800**, Mix Routine **900**, Prediction Routine **1000**, Competition Routine **1100**, Promotion Routine **1200**, Leadership Routine **1300**, Premium Routine **1400**, Price Range Routine **1500**, Reach Routine **1600**, and User Contact Routine **1700** as well as, for example, browser, email client and server routines, client applications, and database applications (discussed further below). Additional data groups for routines, such as for a webserver and web browser, may also be present on and executed by the Indix Server **200** and the other computers illustrated in FIG. **1**. Webserver and browser routines may provide an interface for interaction among the computing devices, for example, through webserver and web browser routines which may serve and respond to data and information in the form of webpages and html documents or files. The browsers and webservers are meant to illustrate machine- and user-interface and user-interface enabling routines generally, and may be replaced by equivalent routines for serving and rendering information to and in interfaces in a computing device (whether in a web browser or in, for example, a mobile device application).

[0064] In addition, the Indix Server Memory **250** also stores an Operating System **255**. These software components may be loaded from a non-transient Computer Readable Storage Medium **295** into Indix Server Memory **250** of the computing device using a drive mechanism (not shown) associated with a non-transient Computer Readable Storage Medium **295**, such as a floppy disc, tape, DVD/CD-ROM drive, memory card, or other like storage medium. In some embodiments, software components may also or instead be loaded via a mechanism other than a drive mechanism and Computer Readable Storage Medium **295** (e.g., via Network Interface **230**).

[0065] The computing device **200** may also comprise hardware supporting input modalities, Input **245**, such as, for example, a touchscreen, a camera, a keyboard, a mouse, a trackball, a stylus, motion detectors, and a microphone. The Input **245** may also serve as a Display **240**, as in the case of a touchscreen display which also serves as Input **245**, and which may respond to input in the form of contact by a finger or stylus with the surface of the Input **245**.

[0066] The computing device **200** may also comprise or communicate via Bus **220** with Indix Datastore **300**, illustrated further in FIG. **3**. In various embodiments, Bus **220** may comprise a storage area network ("SAN"), a high speed serial bus, and/or via other suitable communication technology. In some embodiments, the Indix Server **200** may communicate with the Indix Datastore **300** via Network Interface **230**. The Indix Server **200** may, in some embodiments, include many more components than those shown in this Figure. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment.

[0067] FIG. **3** is a functional block diagram of the Indix Datastore **300** illustrated in the computing device of FIG. **2**. The components of the Indix Datastore **300** are data groups used by routines and are discussed further herein in the discussion of other of the Figures. The data groups used by routines illustrated in FIG. **3** may be represented by a cell in a column or a value separated from other values in a defined structure in a digital document or file. Though referred to herein as individual records or entries, the records may comprise more than one database entry. The database entries may be, represent, or encode numbers, numerical operators, binary values, logical values, text, string operators, joins, conditional logic, tests, and similar.

[0068] FIG. **4** is a flowchart illustrating an embodiment of an Analytics Routine **400**. The Analytic Routine **400** may be performed by, for example, the Indix Server **200**. At box **405**, the Analytic Routine **400** obtains a new set of Price Attribute **340** records and a new set of Product Attribute **345** records, with an assigned MPID **332** and Category **335**. This may occur as frequently as URIs **305** are crawled, the webpages therefrom parsed into Parse Results **325** comprising new Price Attribute **340** and Product Attribute **345** records with an assigned MPID **332** and Category **335**.

[0069] At box **500**, the Analytic Routine **400** performs the Core Price Routine **500** (discussed further below). At box **410**, for each iPID **330** associated with a Price Attribute **340** or Product Attribute **345** record in box **405**, the Analytic Routine **400** appends the then-current Price Attribute **340** record (of box **405**) to a set of Price Attribute **340** records associated with each iPID **330** (each iPID **330** may be associated with a set of Price Attribute **340** records). At box **415**, for each iPID **330** associated with a Price Attribute **340** or Product Attribute **345** record in box **405**, the Analytic Routine **400** merges the then-current Product Attribute **345** record (of box **405**) into a Product Attribute **345** record associated with each iPID **330** (each iPID **330** may be associated with one Product Attribute **340** record). In this merger, new values overwrite old values unless the old record is longer or unless the old record otherwise is judged to be of higher quality (such as if the old record uses fewer words, but the words are less common than the words in the new record); if a new record does not have a value where an old value exists, the old value may be left.

[0070] At box **420**, the output of the Core Price Routine **500** and of boxes **410** and **415** are output to a Sequential File **365** record, which Sequential File **365** record is stored, for example, in the Indix Database **300** and which Sequential File **365** is indexed, for example, to allow the contents of the Sequential File **365** to be searched and values in it accessed as it and the index are updated. Updates may occur, for example, in close-to real-time, following crawl of a webpage and output of new Price Attribute **340** and Product Attribute **345** records.

[0071] At box **600**, the Analytic Routine **400** performs the Insight Routine **600** utilizing and expanding upon the output of the Core Price Routine **500** and the boxes, above. Generally, the Insight Routine **600** identifies what Product Attributes **345**and Price Attributes **340** across the datasets are associated with the changes in price. At box **435** the Analytic Routine **400** stores the output of the Insight Routine **600** in the Indix Database **300** as Insights **375**. At box **1700**, the Analytic Routine **400** performs the User Contact Routine **1700**. Utilizing the User Contact Routine **1700**, users may query the Indix Database **300** and set alerts.

[0072] FIG. **5** is a flowchart illustrating an embodiment of a Core Price Routine **500**. Boxes **505** through **540** may iterate for each new Price Attribute **340** record associated with an iPID **330**. At box **510** all Price Attribute **340** records associated with the iPID **330**, including the new Price Attribute **340** record of box **405** and historic records (and/or summary values derived therefrom), may be obtained. At box **515**, the high, low, average, mean, magnitude, and number of price values over several time periods for the iPID **330** may be calculated. A default time period may be 45 or 30 days, though these values may be calculated for several time periods. The output may be saved, for example, to the Core Price **380** records, and indexed.

[0073] At box **520**, an MPID **332** associated with the iPID **330** may be obtained. At box **525**, the high, low, average, mean, magnitude, and number of price values over several time periods may be calculated for the MPID **332** utilizing the new value associated with the iPID **330** from box **515**. The iPID **330** may be a hash of a URI **305** and the result of box **515** is thus limited to a particular sales channel (typically a Store or Merchant) for a particular Product (taking into account that duplicate iPIDs **330** from a base domain name may be treated as equivalent); the MPID **332** is assigned to all iPIDS **330** which represent the same Product, so the MPID version of this calculation in box **525** returns values relating to the Product across Stores, Merchants, Locations, etc. The calculation of box **525** may return values which are or may be sorted by, for example, Store, Merchant, Location (such as Region), and by time periods such as a Season. The output may be saved, for example, to the Core Price **380** records, and indexed.

[0074] At box **535**, all calculations and other routines which utilize the values for the iPID **330** and the associated MPID **332** may insert the new values calculated for the iPID **330** and the MPID **332** and may recalculate the values. For example, the high, low, average, mean, magnitude, and number of price changes over time periods by Category **335**, such as a Category **335** associated with the iPID **330**, may be calculated. The output may be saved, for example, to the Core Price **380** records, and indexed.

[0075] Calculations or other routines which utilize the values calculated in FIG. **5** may refer to data addresses. The Core Price Routine **500** may update the values stored at these data addresses, which causes the calculations or other routines to update their output, when such calculations or other routines are (re)executed, such on a schedule or on the occurrence of an event.

[0076] At box **599**, the Core Price Routine **500** may return, for example, to the Analysis Routine **400**.

[0077] FIG. **6** is a flowchart illustrating an embodiment of an Insights Routine **600**.

[0078] The Insights Routine **700** may perform one or more of a set of sub-routines. At box **700**, a Volatility Routine **700** may be performed to determine the volatility of prices relative to the many dimensions available in the PriceDNA. At box **800**, a Substitution Routine **800** determines substitutes for an iPID **330**, MPID **332**, or Category **335**. At box **900**, a Mix Routine **900** determines "how many" relative to the many dimensions available in the PriceDNA. At box **1000**, a Prediction Routine **1000** makes price predictions relative to the many dimensions available in the PriceDNA. At box **1100**, a Competition Routine **1100** determines competitors relative to a Product, Store, or Brand. At box **1200**, a Promotion Routine **1200** determines promotions relative to Products, Stores, Brand, Seasons, and other dimensions available in the PriceDNA. At box **1300**, a Leadership Routine **1300** determines which Products lead or follow others in terms of price changes. At box **1400**, a Premium Routine **1400** determines which Products in a Category **335** charge higher (premium) prices. At box **1500**, a Price Range Routine **1500** determines the number of price ranges and maximum and minimum for iPIDs, MPIDs, and categories. At box **1600**, a Reach Routine **1600** determines the reach of an iPID or MPID in terms of the number of people who visit a sales venue.

[0079] FIG. **7** is a flowchart illustrating an embodiment of a Volatility Routine **700**. At box **705**, the Prices associated with an iPID **330** over a time period, such as 30 days, may be obtained, such as from the Core Price **380** records. At box **710**, the number of price changes within the time period may be determined (if this was not already a value in the Core Price **380** records). At box **715**, the number of price changes within the time period ("VBF") may be determined relative to, for example, the iPID **330**, relative to an MPID **332** associated with the iPID **330**, relative to a Brand, relative to a Region, relative to a Price Band by MPID **332**, relative to a Category **335**, and relative to all iPIDs **330** associated with a Merchant. The values may be saved and indexed to accelerate access to and/or enable searching for the values and/or the values may be calculated on an as-needed basis. The values may be saved to the Insights **375** records.

[0080] At box **720**, the benchmark number of Price changes in the period of time may be determined. The benchmark may be, for example, the VBF relative to additional criteria, such as, for example, the VBF for a Product (or MPID), plus 1, divided by the maximum VBF of other Products in the same Category as the Product, multiplied by 100 over 101. The benchmark VBF for a Category may be determined by the VBF for the Category, plus 1, divided by the maximum VBF of the Category, multiplied by 100 over 101. The benchmark VBF for a Merchant may be the VBF of the Merchant, plus 1, divided by the maximum VBF of the Merchant, multiplied by 100 over 101. The benchmark VBF for a Brand may be the VBF of the Brand, plus 1, divided by the maximum VBF of the Brand, multiplied by 100 over 101. The values may be saved to the Insights **375** records.

[0081] FIGS. **8**A-**8**C are flowcharts illustrating embodiments of a Substitution Routine **800**. In a first example of an embodiment of a Substitution Routine **800** illustrated in FIG. **8**A, substitute Products within a Category **335** are identified. At box **801**, which, like other steps may be optional, a Product may be identified by, for example, a user or a routine, and the MPID **332** corresponding thereto may be obtained. At box **805**, a Category **335** may be obtained, whether corresponding to the Product and MPID of step **801** or via a user query or other input, and all MPIDs **332** within the Category **335** may be obtained. At box **810** a Price Band may be obtained or calculated relative to the Category **335** (such as from or according to the Price Range Routine **1500**); the Price Band may be selected by a user. Boxes **815** through **830** may iterate for each iPID **330** within the Category of box **805**.

[0082] At box **820**, the iPIDs **330** in the Category of box **805** and with a Price value within the Price Band of box **810** are identified, such as from the Core Price **380** records. At box **825**, the result of box **820** may be subdivided, grouped, or filtered by Region, Time, Used/New, and according to other dimensions available in the PriceDNA. At box **830** the Substitution Routine **800** may iterate over the remaining iPIDs **330** in the Category **335**. At box **835**, the results may be saved as Substitutes, such as to the Insights **375** records. At box **839**, the process may return.

[0083] In a second example of an embodiment of a Substitution Routine **800** illustrated in FIG. **8**B, substitute Products within a Category **335** with a percentage overlap in Attributes **340**/**345** and within a Price Band are identified. At box **840**, a Category **335** may be obtained, whether corresponding to a Product or via a user query or other input, and all MPIDs **332** within the Category **335** may be obtained. At box **845**, the Product Attributes **345** of all iPIDS **330** within the MPIDs **332** may be obtained. At box **850**, the Product Attributes **345** may be clustered to identify the iPIDs **330** with at least a 50% Product Attribute **345** match or overlap. At box **855** a Price Band may be obtained or calculated relative to the Category **335** (such as from or according to the Price Range Routine **1500**); the Price Band may be selected by a user.

[0084] Boxes **860** through **870** may iterate for each iPID **330** within the MPIDs **332** and Attribute **345** match of box **850**. At box **865**, the iPIDs **330** with a Price value within the Price Band of box **855** and with the Product Attribute **345** match or overlap of box **850** are identified. The result of box **865** may be subdivided or grouped further by sub-Price Ranges. At box **870** the Substitution Routine **800** may iterate over the remaining iPIDs **330** in the MPIDs **332** within the Category **335**. At box **871**, the results may be saved as Substitutes in the Insights **375** records. At box **874**, the process may return.

[0085] In a third example of an embodiment of a Substitution Routine **800** illustrated in FIG. **8**C, substitute Products within a Category **335** with a percentage overlap in Attributes **340**/**345** and in the top or bottom of a Price Range are identified. At box **875**, a Category **335** may be obtained, whether corresponding to a Product or via a user query or other input, and all MPIDs **332** within the Category **335** may be obtained. At box **880**, the Product Attributes **345** of all iPIDS **330** within the MPIDs **332** may be obtained. At box **885**, the Product Attributes **345** may be clustered to identify the iPIDs **330** with at least a **50**% Product Attribute **345** match or overlap.

[0086] Boxes **890** through **897** may iterate for each iPID **330** within the MPIDs **332** and Attribute **345** match of box **885**. At box **895**, the iPIDs **330** with the Product Attribute **345** match or overlap of box **885** and in the bottom of a Price Range or Price Band relative to the starting iPID **330** are identified. At box **896** the top or bottom five (or another subset) of box **895** may be selected. At box **897** this embodiment of the Substitution Routine **800** may iterate over the remaining iPIDs **330** in the MPIDs **332** within the Category **335**. At box **898**, the results may be saved as Substitutes in the Insights **375** records. At box **899**, the process may return.

[0087] FIG. **9** is a flowchart illustrating an embodiment of a Mix Routine **900**. The Mix Routine **900** determines "how many" relative to the many dimensions available in the PriceDNA. At block **905**, the Mix Routine **900** obtains a first segmentation criteria, such as, for example, a Product Name, Brand, or Category. At block **910**, a first sub-segmentation criteria may be obtained, such as, for example, a Store, Location, or Price Band. At block **915**, a second sub-segmentation criteria may be obtained, such as, for example, a Store, Location, or Price Band. At block **920**, the number of Products, such as by MPID **332**, which meet the criteria of blocks **905**, **910**, and **915** may be counted. At block **925**, the result of block **920** may be subdivided or grouped by Location, Time, Season, Price Band, Used/New or other dimensions available in the PriceDNA. At block **930**, the results of blocks **920** and/or **925** may be saved as Mix values in the Insights **375** records. At block **999**, the process may return.

[0088] FIG. **10** is a flowchart illustrating an embodiment of a Prediction Routine **1000**. The Prediction Routine **1000** makes price predictions relative to the many dimensions available in the PriceDNA. At block **1005**, the Prediction Routine **1000** obtains a Product and obtains or identifies an MPID **332** and/or iPIDs **330** associated therewith. At block **1010**, the last Price of the Product by MPID **332** and/or iPID **330** may be obtained, such as from the Core Price **380** records. At block **1015**, first and second linear regression parameters may be calculated or obtained.

[0089] At block **1020**, to the first parameter may be added the second parameter multiplied by the last price of the Product from block **1010**. At block **1025** an error term may be added to the result of block **1020**. At block **1030** a confidence interval may be calculated. At block **1035** the result may be saved as Predictions in the Insights **375** records. At block **1035** the Prediction Routine **1000** may then return.

[0090] In FIG. **10**, the predicted Price for a product may be determined according to the following equation: $p_{t=\alpha+\beta p(t-1)}+\epsilon$, where p_t is the price at time t, α and β are the parameters of the linear regression and ϵ is the error term and is assumed to be Normally distributed. Confidence, C, is a measure that represents the chance for making 0.01% error in predicting the price of the product, C=normsdist(Z)and

$$Z = \frac{.01\% * \text{Price}}{(Std.Error)}.$$

In this formula, the parameters of the model are estimated using the original least squares method as follows:

$$\hat{\beta} = \frac{\left(\Sigma p_{(t-1)} p_t + \frac{1}{n} \Sigma p_{(t-1)} \Sigma p_t\right)}{\left(\Sigma p_{(t-1)}^2 - \frac{1}{n} (\Sigma p_{(t-1)})^2\right)} \text{ and } \hat{\alpha} = \overline{p}_t - \hat{\beta} \overline{p}_{(t-1)}$$

[0091] FIG. **11** is a flowchart illustrating an embodiment of a Competition Routine **1100**. The Competition Routine **1100** determines competitors relative to a Stores, Brands, or Merchants. At box **1105**, a first and second (or more) Store, Brand, or Merchant may be obtained, along with an optional Category **335**. These may be obtained from a user or another routine. At box **1110**, all Products sold by or under each of the entities of box **1105** may be obtained, such as from the PriceDNA. The Products may optionally be filtered by the Category of box **1105**.

[0092] At box **1115**, a determination may be made regarding whether or not the entities of box **1105** have 70% or more overlapping Products, per the Products of box **1110**. The affirmative output of this box may be saved as Competitors in the Insights **375** records.

[0093] At box **1120**, the Competitors may be filtered by, for example, on or more of Store, Substitute, Substitute by Price Band, Brand, Location (including Region), Time (including Season), and whether the Products are sold as used or new. Which criteria are used in the filter may be determined by input from a user. The output of box **1120** may be saved in the Insights **375** records.

[0094] At box **1125**, the average price of Products in the Category **335** of box **1105** may be obtained relative to, for example, the Category **335**, Substitute, Substitute by Price Band, Brand, Location, Time, used/new status, and other criteria. At box **1130**, the output of box **1125** may be ranked and saved as Price Competitiveness in the Insights **375** records.

[0095] At box **1135**, a Store and Location for a target Product may be obtained, such as from a user. At box **1145**, the Competitors from box **1115** may be obtained or determined and the Competitors filtered to select only Competitors with sales in the Location of box **1135**. At box **1145**, Stores in the Location which are the same as the Store of box **1135** may be removed from the set of Competitors, leaving the remainder (those not removed).

[0096] At box **1150**, the output of box **1150** may be placed in a Voroni Diagram or similar data structure, with the location in the Vononi Diagram being based on physical location of the Stores of the Competitors. Generally, a Voroni Diagram determines the distance between objects in a geometric manner, rather than a power-law manner. At box **1155**, the distance between the target Store and each Competitor may be ranked. At box **1160**, the output of box **1160** may be saved as Reach Competitiveness in the Insights **375** records.

[0097] FIG. **12** is a flowchart illustrating an embodiment of a Promotion Routine **1200**. The Promotion Routine **1200** determines promotions relative to Products, Stores, Brand, Seasons, and other dimensions available in the PriceDNA. At box **1205**, a Product may be obtained, such as from user input, and the MPID **332** and/or an IPID **330** corresponding to the Product may be identified in the Attributes **340/345** (via, for example, the Sequential File **365**). The Product may be a single Product or a Bundle comprising multiple Products. At box **1210**, a "Promotion" value may be identified in the Attributes **340/345** associated with the MPID **332** and/or IPID **330**; the "Promotion" value may be a Sale Price and/or a Promotion Code in the Price Attribute **340** records associated with the MPID **332** and/or IPID **330**. Alternatively, at box **1210** the Price history for the MPID **332** and/or IPID **330** may be graphed.

[0098] At box **1215**, the number, length, date/time, and magnitude of the Promotions may be determined and saved as Promotions in the Insights **375** records. Alternatively, the number, length, date/time, and magnitude of the low-points in the graph of box **1210** may be determined and saved as Promotions in the Insights **375** records. At box **1220**, the output of box **1215** may be filtered by criteria such as, for example, date/time, Price Band, Location (including Region), Season, and Holidays. The criteria may be received from, for example, a user and/or a default set of criteria may be applied, with the result of each being saved in the Insights **375** records.

[0099] At box **1225** a time period and a Merchant may be obtained, such as from a user; the Merchant may be associated with the Product of box **1205**. At box **1225**, the number of Products sold by the Merchant in Promotion during the time period may be determined.

[0100] At box **1230**, the result of box **1215** may be benchmarked relative to average Promotion times, durations, and magnitude for other Products (including other Bundles of the Product), the timing of Promotions for other Products, relative to the magnitude of Promotions for other Products, relative to the Products associated with a Brand, relative to all Products sold at a Store, relative to Products in a Price Band, and relative to Competitors and Substitutes. The result may be saved in the Insights **375** records.

[0101] FIG. **13** is a flowchart illustrating an embodiment of a Leadership Routine **1300**. The Leadership Routine **1300** determines which Products lead or follow others in terms of price changes. At box **1305**, a Product may be obtained, for example, from a user or another routine, and the associated MPID **332** determined. At box **1310** Substitutes for the Product may be obtained (such as from or by the Substitutes **800** routine). At box **1315**, the change in Price, or Price delta, for the Product and the Substitutes may be determined over periods of time. The Price delta may be determined in an absolute sense (whether the change was positive or negative) and/or with a determination of the magnitude of the Price delta.

[0102] At box **1320**, the Price deltas determined at box **1315** may be matched, to determine if any of the Price deltas with the same absolute value (positive or negative) occurred within a time window of one another (deltas beyond the time window may not be considered to be correlated), with the result being saved as a Leader/Follower indication in the Insights **375** records.

[0103] At box **1325**, the matching Price deltas of box **1320** may be graphed according to time. At box **1330**, the result of box **1325** may be filtered by criteria such as Region, Rime, Date/Time, Season, Price Band, and Store.

[0104] At box **1335**, the number of Leaders and Followers may be determined relative to a time period. At box **1340**, the average lead/follow time may be determined. At box **1345**, leaders/followers with respect to exact Product matches (for different Stores selling the same Product, determined at box **1330**) may be identified. At box **1350**, the results may be benchmarked relative to the number of leaders/followers and other criteria. The result of various of the boxes in FIG. **13** may be saved in the Insights **375** records. At box **1399**, the Leadership Routine **1300** may return.

[0105] FIG. **14** is a flowchart illustrating an embodiment of a Premium Routine **1400**. The Premium Routine **1400** determines which Products (generally, by MPID) in a Category **335** charge higher Prices (premium). At box **1405**, a Product may be received, such as from input by a user or another routine. At box **1410**, the Substitutes for the Product may be determined or obtained from another routine, such as the Substitution **800** routine and/or the Insights **375** records. At box **1415**, the Prices of the Product and of the Substitutes may be obtained, such as from the Core Price **380** records. At box **1420**, the obtained Prices of box **1415** may be graphed or mapped and the top of the Price distribution identified. The top of the Price distribution may be the top five or ten percent or the top five Products or Substitutes may be identified and saved as the "Premium" Products in the Insights **375** records.

[0106] At box **1425**, the Product Attributes **345** of the Products and Substitutes of box **1410** may be obtained and clustered by similarity. At box **1430**, the Product Attributes **345** unique to or dominant in the Premium Products, determined by the clusters of box **1425**, may be identified and saved in the Insights **375** records.

[0107] At box **1435**, user votes regarding Product Attributes **345** of Premium Products may be received. At box **1440**, the user votes may be tallied and, at box **1445**, the "winning" Product Attributes **345** (with the most votes) may be set as the Product Attributes **345** associated with the Premium Products in the Insights **375** records.

[0108] FIG. **15** is a flowchart illustrating an embodiment of a Price Range Routine **1500**. The Price Range Routine **1500** determines the number of price ranges and maximum and minimum for iPIDs, MPIDs, and categories. At box **1505**, a Product may be obtained, such as from a user or another routine. At box **1510**, the Prices for the Product may be obtained, such as from the PriceDNA for the Product. At box **1515**, the Prices of box **1510** may be clustered by similarity and with a minimum cluster size, with the range in Price across each cluster being saved as Price Ranges for the Product in the Insights **375** records.

[0109] At box **1520**, the Channel Range for the Product may be set as the minimum and maximum of the Prices of box **1510** and saved in the Insights **375** records. At box **1525**, the results of boxes **1510**, **1515**, and **1520** may be filtered by, for example, Region, Date/Time, and according to other criteria and saved in the Insights **375** records. At box **1530**, the Price Ranges may be determined relative to all Products in a Category **335**, all Products by a Brand, and relative to a benchmark which may be, for example, the maximum number of Price Ranges within a Category **335**. The result thereof may be saved as Price Ranges in the Insights **375** records.

[0110] FIG. **16** is a flowchart illustrating an embodiment of a Reach Routine **1600**. The Reach Routine **1600** determines the reach of an iPID or MPID in terms of the number of people who visit a sales venue. At box **1605**, a Product may be obtained, such as from a user or another routine. At box **1610**, the Stores offering the Product for sale may be obtained. At box **1615**, the traffic at the stores may be obtained, such as from a source for online webpage/website traffic, such as Alexa or similar. At box **1620**, the result of box **1615** may be filtered by, for example, criteria such as Date/Time (including Season), Location (including Region), Holiday, and other criteria. The result thereof may be saved as Reach in the Insights **375** records. At box **1699**, the Reach Routine **1600** may return.

[0111] FIG. **17** is a flowchart illustrating an embodiment of a User Contact Routine **1700**. At box **1705**, a user contact with the User Contact Routine **1700** may be detected. The user contact may be part of a user-interface served by the User Contact Routine **1700**. At box **1710**, a user query may be received, such as for PriceDNA records and/or Insight records. At box **1715**, the user query may be executed relative to the Index **370** and the Sequential File **365**. At box **1720**, a determination may be made regarding whether the user has requested that the query be stored as an alert. If so, then at box **1725** a time period for the alert may be obtained or set (such as according to a default time period, such as once per day or week). At box **1730**, on occurrence of the time period of box **1725**, the query may be executed relative to the Index **370** and the Sequential File **365**. At box **1735**, an alert or other message may be sent to contact information associated with the user. At box **1799**, the User Contact Routine **1700** may conclude.

[0112] The above Detailed Description of embodiments is not intended to be exhaustive or to limit the disclosure to the precise form disclosed above. While specific embodiments of, and examples are described above for illustrative purposes, various equivalent modifications are possible within the scope of the system, as those skilled in the art will recognize. For example, while processes or blocks are presented in a given order, alternative embodiments may perform routines having operations, or employ systems having blocks, in a different order, and some processes or blocks may be deleted, moved, added, subdivided, combined, and/or modified. While processes or blocks are at times shown as being performed in series, these processes or blocks may instead be performed in parallel, or may be performed at different times. Further, any specific numbers noted herein are only examples; alternative implementations may employ differing values or ranges.

1. A computer implemented method of processing information from webpages, the method comprising:

receiving a first and a second set of price and product attributes for a first product, which attributes comprise:

a first identifier of a first identifier-type derived from a URI which links to a webpage offering the product for sale, a second identifier of a second identifier-type assigned to all instances of the product as offered for sale at any URI, and a first category in a category taxonomy;

performing a first URI-specific price analysis of price values in the first and second sets of price and product attributes to identify changes in price for the first product and associating the result with the first identifier of the first identifier-type and saving the result as a first URI-specific core price result;

receiving a third and fourth set of price and product attributes for a second product, which attributes comprise:

a third identifier of the first identifier-type, a fourth identifier of the second identifier-type, and a second category in the category taxonomy;

performing a second URI-specific price analysis of price values in the third and fourth sets of price and product attributes to identify changes in price for the second product and associating the result with the third identifier of the first identifier-type and saving the result as a second URI-specific core price result;

when the second identifier and the fourth identifiers are the same, performing a first non-URI-specific price analysis utilizing the first and second URI-specific core price results to identify changes in price according to the second identifier-type and saving the result as a first non-URI-specific core price result;

saving and indexing the output of the URI-specific and non-URI-specific price analyses in a first file structure and making the first file structure available to be searched substantially as the sets of product and price attributes are received;

performing a meta-analysis utilizing the URI-specific and non-URI-specific core price results to identify what product and price attributes across the datasets are associated with the changes in price; and

saving and indexing the output of the meta-analysis as a second file structure and making the second file structure available to be searched.

2. The method of claim **1**, further comprising merging new product attribute records into prior product attribute records and saving new price attribute records along with prior price attribute records.

3. The method of claim **1**, wherein the URI-specific price analysis comprises determining the high, low, average, mean, magnitude and number of price changes over at least one time period for the price and product attributes associated with the same identifier of the first identifier-type.

4. The method of claim **1**, wherein the non-URI-specific price analysis comprises determining the high, low, average, mean, magnitude and number of price changes over at least one time period for the price and product attributes associated the same identifier of the second identifier-type.

5. The method of claim **4**, wherein at least one of the first and second identifier-types are further associated with at least one of a store, a merchant, and a location and wherein the non-URI-specific price analysis produces results associated therewith.

6. The method of claim **1**, wherein the price attributes comprise at least one of a time, a product name, a price, a quantity, a unit of measurement, a merchant name, a store name, a bundle detail, and a location.

7. The method of claim **1**, wherein the product attributes comprise at least one of a title, a brand, a category in the category taxonomy, a color, a product type, and a size.

8. The method of claim **1**, further comprising receiving a user query and executing the query relative to the first and/or second file structures.

9. The method of claim **1**, further comprising receiving a user query, a schedule for executing the query, executing the query at the scheduled time on the first and/or second file structures, and alerting the user regarding the result of the query.

10. The method of claim **1**, wherein the first and second file structures may be searched by at least one of the first identifier-type, the second identifier-types, or a category in the category taxonomy.

11. The method of claim **1**, wherein the first and second categories are the same.

12. The method of claim **1**, wherein the meta-analysis determines the volatility of price changes over time for each of the first and second products.

13. The method of claim **12**, wherein the volatility is determined by counting the number of price changes in a time period according to at least one of the first identifier-type, the second identifier-type, a brand, a region, a price band, and a category in the category taxonomy.

14. The method of claim **1**, wherein the meta-analysis determines whether one of the products is a substitute for the other.

15. The method of claim **14**, wherein whether one of the products is a substitute for the other is determined by determining if the first and second products are in the same category in the category taxonomy and by determining whether the first and second products are within a price band within the category.

16. The method of claim **15**, further comprising determining if the first and second products share at least fifty-percent of the same product attributes.

17. The method of claim **1**, wherein the meta-analysis determines predictions regarding the future prices for the products.

18. The method of claim **17**, wherein the predictions are determined by obtaining the last price of at least one of the products from the URI-specific core price associated therewth, calculating or obtaining first and second linear regression parameters, multiplying the second linear regression parameter by the last price and adding this to the first linear regression parameter.

19. The method of claim **1**, wherein the price and product attributes comprise at least one of a store, merchant, or brand and the meta-analysis determines products associated therewith and competitors thereof.

20. The method of claim **1**, wherein the meta-analysis determines whether a price change for the first product leads or follows a price change for the second product.

21. The method of claim **1**, wherein the meta-analysis determines whether the first or second product is a premium product relative to the other.

22. The method of claim **1**, wherein the meta-analysis determines the price ranges in which the products are offered for sale.

23. A webpage information processing computing apparatus, the apparatus comprising a processor and a memory storing instructions that, when executed by the processor, configure the apparatus to:

receive a first and a second set of price and product attributes for a first product, which attributes comprise:

a first identifier of a first identifier-type derived from a URI which links to a webpage offering the product for sale, a second identifier of a second identifier-type assigned to all instances of the product as offered for sale at any URI, and a first category in a category taxonomy;

perform a first URI-specific price analysis of price values in the first and second sets of price and product attributes to identify changes in price for the first product and associating the result with the first identifier of the first identifier-type and save the result as a first URI-specific core price result;

receive a third and fourth set of price and product attributes for a second product, which attributes comprise:

a third identifier of the first identifier-type, a fourth identifier of the second identifier-type, and a second category in the category taxonomy;

perform a second URI-specific price analysis of price values in the third and fourth sets of price and product attributes to identify changes in price for the second product and associate the result with the third identifier of the first identifier-type and save the result as a second URI-specific core price result;

when the second identifier and the fourth identifiers are the same, perform a first non-URI-specific price analysis utilizing the first and second URI-specific core price results to identify changes in price according to the second identifier-type and save the result as a first non-URI-specific core price result;

save and index the output of the URI-specific and non-URI-specific price analyses in a first file structure and make the first file structure available to be searched substantially as the sets of product and price attributes are received;

perform a meta-analysis utilizing the URI-specific and non-URI-specific core price results to identify what price and product attributes across the datasets are associated with the changes in price; and

save and index the output of the meta-analysis as a second file structure and make the second file structure available to be searched.

24. A non-transient computer-readable storage medium having stored thereon instructions that, when executed by a processor, configure the processor to:

receive a first and a second set of price and product attributes for a first product, which attributes comprise:

a first identifier of a first identifier-type derived from a URI which links to a webpage offering the product for sale, a second identifier of a second identifier-type assigned to all instances of the product as offered for sale at any URI, and a first category in a category taxonomy;

perform a first URI-specific price analysis of price values in the first and second sets of price and product attributes to identify changes in price for the first product and associating the result with the first identifier of the first identifier-type and save the result as a first URI-specific core price result;

receive a third and fourth set of price and product attributes for a second product, which attributes comprise:

a third identifier of the first identifier-type, a fourth identifier of the second identifier-type, and a second category in the category taxonomy;

perform a second URI-specific price analysis of price values in the third and fourth sets of price and product attributes to identify changes in price for the second product and associate the result with the third identifier of the first identifier-type and save the result as a second URI-specific core price result;

when the second identifier and the fourth identifiers are the same, perform a first non-URI-specific price analysis utilizing the first and second URI-specific core price results to identify changes in price according to the second identifier-type and save the result as a first non-URI-specific core price result;

save and index the output of the URI-specific and non-URI-specific price analyses in a first file structure and make the first file structure available to be searched substantially as the sets of product and price attributes are received;

perform a meta-analysis utilizing the URI-specific and non-URI-specific core price results to identify what price and product attributes across the datasets are associated with the changes in price; and

save and index the output of the meta-analysis as a second file structure and make the second file structure available to be searched.

* * * * *