



- (51) **International Patent Classification:**
G06F 17/00 (2006.01) *G06F 17/30* (2006.01)
- (21) **International Application Number:**
PCT/US2015/020164
- (22) **International Filing Date:**
12 March 2015 (12.03.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** MAURER, Ron; Kiryat Technion, 32000 Haifa (IL). SCHEIN, Sagi; Kiryat Technion, 32000 Haifa (IL). SABO, Yaniv; Kiryat Technion, 32000 Haifa (IL). KESHET, Renato; Kiryat Technion, 32000 Haifa (IL). NACHLIELI, Hila; Kiryat Technion, 32000 Haifa (IL).
- (74) **Agent:** DAS, Manav; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** PROGRESSIVE INTERACTIVE APPROACH FOR BIG DATA ANALYTICS

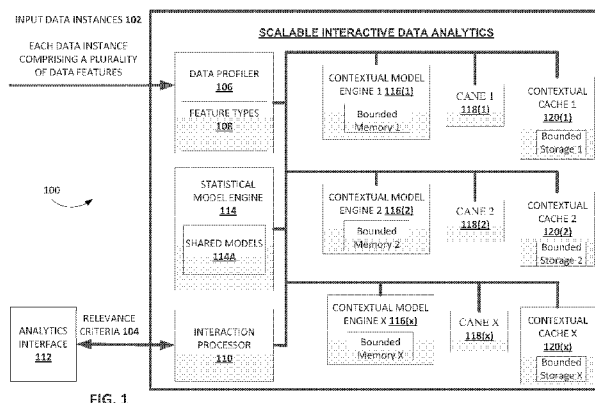


FIG. 1

(57) **Abstract:** Scalable interactive data analytics is disclosed. One example is a system including a data profiler to identify a feature type for each data feature of input data instances. An interaction processor receives relevance criteria from an analytics interface. A statistical model engine generates shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria. Contextual model engines with bounded memories generate contextual models responsive to the relevance criteria based on the shared models. Contextual analytics engines automatically detect a group of the input data instances based on a contextual model generated by a respective contextual model engine, and generate results of data analytics performed on the detected group. Contextual caches store a sub-collection of the results, where the sub-collection is indicative of high relevance to the analytics function.



SCALABLE INTERACTIVE DATA ANALYTICS

Background

[0001] Subject matter experts may look for relevant insights from analyzing data. Interactive data analytics allows subject matter experts to guide an analytics system based on relevance to the expert. For example, an analytics system may scan streaming data, perform analytics on the data, and provide results of the analytics via an interactive graphical user interface, based on feedback from subject matter experts.

Brief Description of the Drawings

[0002] Figure 1 is a functional block diagram illustrating one example of a system for scalable interactive data analytics.

[0003] Figure 2 is a block diagram illustrating one example of a processing system for implementing the system for scalable interactive data analytics.

[0004] Figure 3 is a block diagram illustrating one example of a computer readable medium for scalable interactive data analytics.

[0005] Figure 4 is a flow diagram illustrating one example of a method for scalable interactive data analytics.

Detailed Description

[0006] Subject matter experts (“SMEs”) may expect to gain relevant insights from analyzing their data. For example, a subject matter expert may search for information related to a sales budget for a quarter. Also, for example, a subject matter expert may search for information related to recall of a faulty batch of servers. However, it may require a skillful data scientist to formulate “soft” queries for input into structured analytical tasks, such as for example, an identification of clusters, a detection of anomalies, and/or an identification of salient data features. As relevance tends to vary from one user to another, and

may also change over time, an analytics interface that allows SMEs to interactively provide context to the analytics may be desirable. Ensuring that such a system can handle large, unfamiliar, datasets while retaining interactivity and simplicity, may be a complex research and engineering challenge. Also, for example, when the volume (size of input data) gets large enough, performance of a basic and/or simple analysis on the entire input data may be slow. As another example, supporting variable datasets to allow an out-of-the-box experience for SMEs (e.g., receiving insights from unfamiliar data as it flows in), may further hinder interactivity, as it may result in complex features and data models.

[0007] Instead of processing the entire input data and providing all possible outputs at once, an alternative approach may be to make real-time informed selections of the input data. As described herein, system may be designed so that the appropriate compromises may be made automatically, and adapted into an on-going stream of user feedbacks, without requiring SMEs to become data scientists.

[0008] As described herein, a virtual framework or platform for interactive, contextual analytics for big-data discovery and exploration use cases is disclosed. The platform may operate in a big data streaming environment and may construct an online shared model with bounded memory consumption. Such a collection of shared models may serve a collection of contextual analytics engines that perform specific analytics functions, such as for example, anomaly detection and cluster analysis. The contextual analytics engines supplement shared model with task-specific parameters to form usable analytics models. By carefully managing the size of each analytics model, and maintaining a small cache of relevant results, the system may decouple user feedbacks from big data processing, and allow interactivity. New insights may be provided to the user through a set of widgets that may provide context for the on-going analytics. The platform may handle datasets with a variety of attribute feature types using novel data profiling techniques.

[0009] As described in various examples herein, scalable interactive data analytics is disclosed. One example is a system including a data profiler, an

interaction processor, a statistical model engine, a plurality of contextual model engines, a plurality of contextual analytics engines, and a plurality of contextual caches. The data profiler identifies a feature type for each data feature of input data instances. The data profiler may further supply additional characteristics to some of the feature types to improve handling of the herein data feature type. The interaction processor receives relevance criteria from an analytics interface. The statistical model engine generates a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria. The plurality of contextual model engines with bounded memories generate contextual models responsive to the relevance criteria based on the shared model. The plurality of contextual analytics engines automatically detect a group of the input data instances based on a contextual model generated by a respective contextual model engine, and generate results of data analytics performed on the detected group. The plurality of contextual caches with bounded storage capacities store a progressively updated sub-collection of the generated results, where the sub-collection is indicative of high relevance to the analytics function.

[0010] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific examples in which the disclosure may be practiced. It is to be understood that other examples may be utilized, and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present disclosure is defined by the appended claims. It is to be understood that features of the various examples described herein may be combined, in part or whole, with each other, unless specifically noted otherwise.

[0011] Figure 1 is a functional block diagram illustrating one example of a system 100 for system for scalable interactive data analytics. System 100 is shown to include a data profiler 106, an interaction processor 110, a statistical model engine 114, a plurality of contextual model engines 116(1), 116(2), ..., 116(x), a plurality of contextual analytics engines ("CANEs") 118(1), 118(2), ...,

118(x), and a plurality of contextual caches 120(1), 120(2), ..., 120(x). System 100 is communicatively linked to an analytics interface 112. The data profiler 106, interaction processor 110, statistical model engine 114, plurality of contextual model engines 116(1), 116(2), ..., 116(x), plurality of contextual analytics engines 118(1), 118(2), ..., 118(x), and plurality of contextual caches 120(1), 120(2), ..., 120(x) are communicatively linked to one another via a network.

[0012] The term “system” may be used to refer to a single computing device or multiple computing devices that communicate with each other (e.g. via a network) and operate together to provide a unified service. In some examples, the components of system 100 may communicate with one another over a network. As described herein, the network may be any wired or wireless network, and may include any number of hubs, routers, switches, cell towers, and so forth. Such a network may be, for example, part of a cellular network, part of the internet, part of an intranet, and/or any other feature type of network. In some examples, the network may be a secured network.

[0013] System 100 progressively analyses the data as it streams in, making insights available to an SME in communication with the analytics interface 112 so that the analytics interface 112 may be interactively guided. The phrase “progressively analyses” as used herein, refers to a dynamic, interactive, cumulative, and/or iterative analysis of the input data instances 102. For example, system 100 may perform a first analysis of a first collection of input data instances of the input data instances 102, receive feedback from the analytics interface 112, and perform a second analysis of a second collection of input data instances of the input data instances 102. In some examples, the first collection may be the same as the second collection. In some examples, the first collection may include the second collection. In some examples, the first collection and/or the second collection may include all the input data instances 102. The data profiler 106 may identify a feature type of data feature 108 for each data feature of input data instances 102. Generally, input data instances 102 may include any type of data. For example, the input data instances 102 may include structured, mixed data types (numerical, categorical,

and ordinal). For example, input data instances 102 may include high-dimensional data sets, where each dimension represents a data feature. In some examples, the input data instances 102 may include large, unfamiliar and/or unstructured datasets. In some examples, the data profiler 106 may perform data profiling and automatic data preparation. In some examples, the data profiler 106 may analyze each data feature to detect its format (e.g. integer, real, date/time, string, free text, domain-specific: URL/IP/Email). In some examples, input data instances 102 may be data related to customer transactions, Web navigation logs (e.g. click stream), security logs, and/or DNA sequences. The input data instances 102 may be normalized in several ways. For example, a log analysis, and/or a signal analysis may be performed on the input data instances 102.

[0014] In some examples, the data profiler 106 may identify feature information type and features may be classified. In some examples, the features may be classified as categorical, ordinal, numerical, temporal, unique-index, or compound. In some examples, integer attributes may be identified, for example, as ordinal vs. categorical – as these different information types may imply completely different statistical modeling and analysis methods in each of the CANEs 118(1), 118(2), ..., 118(x). In some examples, compound attributes of known feature types may be detected. In some examples, pre-defined transformation rules may be applied to simplify the compound attributes into feature types more suitable for statistical analysis (e.g. from a URL extract domain-name as a categorical feature).

[0015] In some examples, data profiler 106 may receive a normalized input data. In some examples, data profiler 106 may perform operations to normalize the input data instances 102. In some examples, the input data instances 102 may be a stream of log messages that may be analyzed by the data profiler 106 for latent structure and transformed into a concise set of structured log message feature types and parameters. In some examples, each source of log messages may be pre-tagged. The input data instances 102 may be a corresponding stream of event feature types according to matching regular expression. Log messages that do not match may define new regular

expressions. In some examples, telemetry signals may also be analyzed by the data profiler 106 for relevant features.

[0016] The architecture of system 100 may be extensible, which means that more dedicated transformation rules may be incorporated depending on a domain. Such transformations may be related as parser or processor. For example, in the context of network security data, transformations such as, for example, “IP to categorical”, “Domain to categorical”, “Text to <num of words, num of chars, language, list of terms, string_entropy>”, and “URI splitter to <port, domain name, parameters, query>”, may be utilized.

[0017] In some examples, the data profiler 106 may determine features complexities for the input data instances 102, where the features complexities are indicative of memory consumption to be allocated to generate the contextual models. For example, as data flows through the data profiler 106, features complexities may be estimated. Generally, features complexity, or representation complexity, measures the amount of memory required to maintain an accurate representation of the distribution of values per feature as a function of the size of the data. Additionally, representation complexity may estimate trade-off curves between memory required to maintain approximated feature-values distribution and the accuracy of the target analytics. Such estimates may guide the plurality of contextual model engines 116(1), 116(2), ..., 116(x) when they determine which features to model and how to accurately model them.

[0018] In some examples, system 100 may forecast, from a small initial data-sample, what may be an appropriate feature-model size, after a much larger number of data features have been processed. In particular, a growth rate of a number of unique values may be estimated as a function of number of data instances. Using such a projected complexity estimate, system 100 may be able to strike a memory/accuracy trade-off when categorical features/combinations are modeled by lossy histograms with limited number of keys. The term “lossy histogram” may be interchangeably used with the term “adaptive histogram”. Such histograms capture a notion of aging of values during the data processing.

[0019] In some examples, the data profiler 106 may encode processed information into a parsing scheme which may be utilized to transform raw data into the structured feature representations that may be usable by the plurality of contextual model engines 116(1), 116(2), ..., 116(x), and/or the plurality of CANEs 118(1), 118(2), ..., 118(x). By automatically handling data variety in a best-effort approach (i.e. selective processing of features which the system is equipped to handle based on their data-profile), system 100 may enable an out of the box experience for the SME at the analytics interface 112.

[0020] In some examples, the parsing scheme may generate a schema. Such a schema may include a collection of functions that may be applied to each feature in the input data. In some examples, this may be a map function which ingests raw input data in a string format, and outputs a number of features values (either numerical or categorical) and their respective feature types. In some examples, such a schema may be an abstract notion that gives instructions of how to generate an appropriate transformation function that may eventually operate on the actual input data.

[0021] In some examples, the data profiler may analyze the beginning of the stream of input data instances 102, and may automatically generate an extended schema that may include both format identification (e.g., int, float, time, URL etc.), and information type identification per data feature (e.g., unique-ID, numerical, categorical, temporal and domain specific information types— e.g. port-number). In addition, the extended schema may store, for each feature a complexity factor. As an example, for categorical features, the complexity factor characterizes the expected increase in number of categories with the data size.

[0022] In some examples, the schema may be provided as a recipe to the data profiler 106 which may process only feature types that the system knows how to handle (e.g., categorical, and cyber-specific), and which may not be expected to have a sudden and/or substantial increase in space requirement according to their complexity factor. For example, data profiler 106 may identify categorical attributes in any unfamiliar data source and handle them properly, demonstrating out-of-the-box experience.

[0023] The interaction processor 110 may receive relevance criteria 104 from an analytics interface 112. The interaction processor 110 processes interactions between the components of system 100 and an external interface such as the analytics interface 112. In some examples, the analytics interface 112 may be an anomaly processor that provides an interactive visual interface to analyze anomalies in the input data instances 102. In some examples, the interaction processor 110 may receive relevance criteria 104 related to anomalies.

Generally, the anomaly processor may identify what may be “normal” (i.e. non-extreme and/or expected, and/or unremarkable) in the distribution of feature values or value combinations, and may be able to select outliers that may be representative of rare data-instances that are distinctly different from the norm.

[0024] Generally, the relevance criteria 104 may be any criteria that are relevant to a domain. A domain may be an environment associated with the input data instances 102, and the relevance criteria 104 may be semantic and/or contextual criteria relevant to aspects of the domain. For example, the input data instances 102 may be representative of Web navigation logs (e.g. click stream), and the domain may be the DNS (domain name servers) network traffic, and the relevance criteria 104 may be semantic and/or contextual criteria relevant to analysis of network traffic. Such criteria may include selection of group or conditional features (e.g. distribution of ports accessed by each IP address), increasing the relative importance of some features relative to others (e.g. boost weights for features measuring the randomness of URL strings to target data-instances with machine generated URLs). Also, for example, the input data instances 102 may be related to operational or security logs, and the domain may be a secure office space for which the security logs are being maintained and/or managed, and the relevance criteria 104 may be semantic and/or contextual criteria relevant to tracking security logs based on preferences such as location, time, frequency, error logs, warnings, and so forth. As described herein, in some examples, a weighting may be utilized to convey context (e.g., weight 0 for removal of a certain feature from consideration in one or more analytics engine).

[0025] As used herein, an SME may be an individual in possession of domain knowledge. For example, the domain may be a retail store, and the SME may be the store manager. Also, for example, the domain may be a hospital, and the SME may be a member of the hospital management staff. As another example, the domain may be a casino, and the SME may be the casino manager. Also, for example, the domain may be a secure office space, and the SME may be a member of the security staff.

[0026] The statistical model engine 114 may generate a collection of shared models 114A based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria incorporating the relevance criteria 104 related to anomalies. Generally, the analytics function is any function to be performed by system 100 that incorporates the relevance criteria 104. For example, the relevance criteria 104 may identify a collection of network request events related to a specified port number, and the analytics function may be detection of anomalies in this collection. Also, for example, the analytics function may be a function to be performed by system 100 that incorporates the relevance criteria 104 related to semantic and/or contextual criteria relevant to aspects of the domain. As another example, the analytics function may be a function that incorporates the relevance criteria 104 related to semantic and/or contextual criteria relevant to analysis of network traffic. Also, for example, the analytics function may be a function that incorporates the relevance criteria 104 related to semantic and/or contextual criteria relevant to tracking security logs.

[0027] Each model of the collection of shared models 114A may be targeted to different aspects of the input data instances 102. The statistical model engine 114 may receive structured, mixed data feature types (numerical, categorical, and ordinal) from the data profiler 106. The statistical model engine 114 may progressively model the data with joint distributions of features to generate the collection of shared models 114A. The phrase “progressively model” as used herein, refers to a dynamic, interactive, cumulative, and/or iterative modeling of the input data instances 102. For example, different shared models may be generated from a collection of data based on, for example, feedback from the

analytics interface 112. In some examples, the collection of shared models 114A may include approximated models appropriate for targeted analyses. As the targeted analyses change based on, for example, interactions with the analytics interface 112, the collection of shared models 114A may be updated accordingly. For example, a new shared model may be added to the collection of shared models 114A, and/or an existing shared model may be removed from the collection of shared models 114A.

[0028] In some examples, the collection of shared models 114A may include approximated feature histograms or pairs of feature histograms which may be appropriate for targeted analyses. For example, the statistical model engine 114 may include a progressive histogram engine that generates approximate adaptive histograms. In some examples, the statistical model engine 114 may have limited temporal memory and/or selective purging mechanisms that may limit histogram sizes and keep them fresh. In some examples, histograms may be maintained in a fast key-value store to retain fast update and response rates from multiple analytics operations.

[0029] The plurality of contextual model engines 116(1), 116(2), ..., 116(x) with bounded memories may generate contextual models responsive to the relevance criteria based on the shared models 114A. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may each may be designed to capture different statistical aspects of the input data instances 102 (e.g., frequent values, rare values, variability, coupling, etc.). In some examples, each of the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may be applicable to specific analytics functions (e.g., detect anomalies, identify clusters, etc.). For example, contextual model engine 116(1) may be applicable to the specific analytics function performed by a contextual model engine 116(1), and so forth.

[0030] In some examples, each of the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may be purposefully maintained accurate enough for a targeted use, and bounded in its memory footprint, i.e. may be associated with bounded memories. In particular, each of the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may be configured to represent an

appropriately identified combination of features, and may be configured to further determine a level of detail for such representation.

[0031] For example, one model engine may capture frequent values.

Accordingly, the collection of shared models 114A may be an adaptive histogram that supports a mechanism that purges rare values so that the memory footprint of the adaptive histogram remains manageable. As another example, another model engine may be targeted at representing infrequent values, and the adaptive histogram may support a mechanism that purges frequent values, and may age and retire infrequent values, so that the size of the adaptive histogram remains manageable even for large input data instances 102.

[0032] In order to make such decisions, each of the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may need to identify the feature type of each data feature in the input data instances 102, rather than just a format. For example, an integer-format feature may represent a categorical error-code, an ordinal grade-scale (e.g., in an interval [0-5]), and/or a quasi-continuous numerical value (e.g., size in bytes, time in epoch units, etc.).

[0033] In some examples, tradeoff optimization may be determined. Tradeoff optimization balances computational complexity against statistical accuracy of a contextual model generated by a contextual model engine of the plurality of contextual model engines 116(1), 116(2), ..., 116(x). Tradeoff optimization may require estimates of the features complexity or representation complexity (e.g. expected rate of encountering new values for categorical features/combinations, or amount of dependencies between features), as provided by the data profiler 106. In some examples, the tradeoff may be between the statistical accuracy and the computational complexity (e.g., size of memory, amount of computation).

[0034] In some examples, the bounded memory of a contextual model engine of the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may be adjusted to balance computational complexity against the statistical accuracy of a contextual model generated by the contextual model engine. For example, initially, a fixed quota of memory may be allocated for the plurality of contextual

models. By analyzing information complexity of each feature or combination of features, a fraction of the allocated quota may be further allocated to each model of the plurality of contextual models. In some examples, some models may get 0 quota. In some examples, a relative quota may be determined by the data profiler 106 at the beginning of the processing, from profiling a small sample of items, and the relative quota may be updated during data processing as more data flows into the system 100.

[0035] The plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may automatically detect a group of the input data instances 102 based on a contextual model generated by a respective contextual model engine of the plurality of contextual model engines 116(1), 116(2), ..., 116(x), and may generate results of data analytics performed on the detected group. Each of the CANEs 118(1), 118(2), ..., 118(x) may utilize a set of parameters that capture SME's intents in a compute context via the relevance criteria 104 (e.g. feature-relevance weights, selected items, transformation rules). Such a context may be modified via well-defined APIs that may be guided via the interaction processor 110. In some examples, the interaction processor 110 may include analytics widgets, i.e. small, single purpose analytics applications. In some examples, the size of the group of the input data instances may be much smaller than the size of the input data comprising input data instances 102. Accordingly, system 100 is able to maintain computational efficiency while providing fast processing and retrieval of search results responsive to SME interactions.

[0036] In some examples, each of the CANEs 118(1), 118(2), ..., 118(x) may utilize global knowledge on the input data instances 102 that the system 100 has witnessed, from the collection of shared models 114A, such as, for example, adaptive histograms. Each of the CANEs 118(1), 118(2), ..., 118(x) may be associated with an analysis algorithm that produces specific analytics results on incoming data features, using its own specific contextual models that may be derived from the collection of shared models 114A maintained by the system (e.g., adaptive histograms), and where the specific model derivation

may be influenced by user-controlled parameters (via the interaction processor 110)) that reflect relevance criteria.

[0037] As another example, the analysis algorithm of an anomaly CANE of the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may build and maintain a specific contextual model in a form of a set of mappings from feature-combination values to feature-combination anomaly scores where the mappings may be based on the collection of shared models 114A maintained by the system (e.g., feature-combination histograms). The analysis algorithm may apply the anomaly mappings to each incoming data instance (e.g., a row of data-feature values), and may compute outputs for each data instance, such as, for example, a total anomaly score, and an anomaly fingerprint (set of feature-combinations contributing the majority of the combined anomaly score). In some examples, the input data instances 102 may include rare values, and the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include an anomaly processor to detect anomalies based on the rare values.

[0038] Also, for example, a clustering CANE may produce a specific contextual model that may order different features by combined statistical consideration based on the collection of shared models 114A (e.g., common adaptive histograms), and user ranking of relevance criteria (e.g., feature relevance). An analysis algorithm associated with the clustering CANE may discover fingerprints for cluster candidates based on the common adaptive histograms and the feature-ordering, may rank the candidate fingerprints by some interestingness criterion, and may keep top-*K* interesting cluster fingerprints as the clustering model. The analysis algorithm associated with the clustering CANE may output a cluster tag for each incoming data-instance (line) that matches one of the top-*K* clusters. In some examples, the data features of the input data instances 102 may include frequent combinations of data feature values, and the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include a cluster processor to detect clusters based on the frequent combinations of the data feature values.

[0039] System 100 includes a plurality of contextual caches 120(1), 120(2), ..., 120(x) with bounded storage capacities, where each contextual cache stores a

progressively updated sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function. The phrase “progressively updated” as used herein, refers to a dynamic, interactive, cumulative, and/or iterative updating of the sub-collection of the generated results. For example, different sub-collections may be generated based on, for example, feedback from the analytics interface 112. In some examples, the sub-collection of the generated results may include generated results appropriate for targeted analyses. As the targeted analyses change based on, for example, interactions with the analytics interface 112, the sub-collection of the generated results may be updated accordingly. For example, a new result may be added to the sub-collection of the generated results, and/or an existing result may be removed from the sub-collection of the generated results.

[0040] In some examples, each of the CANEs 118(1), 118(2), ..., 118(x) may also manage, as part of its context, a cache of results (detected anomalies, discovered clusters, etc.) that scored with highest relevance at a given moment. This cache may be maintained small enough so that widget interactivity remains independent of the input data size or streaming velocity. For example, a contextual cache of a clustering CANE may manage a list of up to N (e.g., $N \sim 1000$) randomly selected data instances per each of top- K clusters, which represent a rough approximation of the cluster internal statistics and which may be available for interaction with the analytics interface 112 via the interaction processor 110. As described herein, such a list may be progressively updated to remain relevant to relevance criteria 104.

[0041] Finally, in some examples, analytics results may be stored into a persistent storage, and utilized outside the scope of the progressive framework. In some examples, the persistent storage may be an independent component of system 100. In some examples, the persistent storage may be included in the plurality of contextual caches 120(1), 120(2), ..., 120(x). In some examples, the bounded storage capacity of a contextual cache may be substantially smaller than the size of the input data comprising the input data instances 102. In some examples, the bounded storage capacities of the respective contextual caches may be independent of the size of the input data and the rate at which input

data is received. Such features facilitate fast search and retrieval of results relevant to a query received from the analytics interface 112.

[0042] In some examples, the interaction processor 110 provides the sub-collection of results to the analytics interface 112. As described herein, the interaction processor 110 is communicatively linked to the analytics interface 112 to receive relevance criteria 104. The interaction processor 110 may access various components of system 100 to provide data that may modify the visual representation provided by the analytics interface 112. For example, the interaction processor 110 may receive relevance criteria 104 such as weights for anomalies from the analytics interface 112 (e.g., an anomaly processor), and may provide the weights to the data profiler 106 and/or the plurality of contextual model engines 116(1), 116(2), ..., 116(x). Based on the relevance criteria 104, the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may progressively update results of the analytics, and the plurality of contextual caches 120(1), 120(2), ..., 120(x) may be updated accordingly. In some examples, the interaction processor 110 may access the plurality of contextual caches 120(1), 120(2), ..., 120(x) to provide updated system data to the analytics interface 112.

[0043] System 100 generally facilitates use of large scale visual and interactive analytics to reduce time from asking a business question on a dataset, to gaining relevant insights. In some examples, the backend processing may begin when a request to analyze a dataset is received from the analytics interface 112 via the interaction processor 110. In some examples, the dataset may be a table stored in a tabulated file format or a relational database, with rows corresponding to data instances, and columns corresponding to data features of the data instances.

[0044] For a new dataset, the data profiler 106 may build a data description schema along with a set of transformation rules. Such a procedure is designed to take from a few seconds and up to a minute depending on the number of features and the feature type of analysis they may require. Once such procedure is completed, it may take a few seconds for the plurality of CANEs 118(1), 118(2), ..., 118(x) to generate results of data analytics performed on a

detected group of the dataset. The plurality of contextual caches 120(1), 120(2), ..., 120(x) store these results, and make them available to the interaction processor 110. An SME may modify relevance criteria in the analytics interface 112. In some examples, the interaction processor 110 may receive the modified relevance criteria from the analytics interface 112. The interaction processor 110 may identify a contextual cache of the plurality of contextual caches 120(1), 120(2), ..., 120(x), based on the modified relevance criteria. In some examples, the identified contextual cache may be progressively updated based on the modified relevance criteria. For example, as the system 100 progresses, the plurality of contextual caches 120(1), 120(2), ..., 120(x) are automatically updated based on relevance criteria 104. The interaction processor 110 may search the identified contextual cache for additional results based on the modified relevance criteria. Finally, the interaction processor 110 may provide the additional results to the analytics interface 112.

[0045] In some examples, the SME may modify relevance criteria (e.g., reweight features) in the analytics interface 112. The interaction processor 110 may receive the modified relevance criteria from the analytics interface 112. In some examples, an anomaly score function may be recomputed on the cached items based on the new set of weights and the result may be presented via the interaction processor 110. Since the entire recompute cycle may consider data that is already in the cache, it may only take a few seconds to reflect the change so the SME may never lose visual context in the analytics interface 112. Second, the system 100 may identify the modified relevance criteria (e.g., weights) so that fresh relevant anomalies may be integrated into the cache by repeating the functions of the plurality of contextual model engines 116(1), 116(2), ..., 116(x), and the plurality of contextual model engines 116(1), 116(2), ..., 116(x). The preferable size for the cache, described herein as the bounded storage capacity, may depend on system resources and may be configured as required.

[0046] The components of system 100 may be computing resources, each including a suitable combination of a physical computing device, a virtual

computing device, a network, software, a cloud infrastructure, a hybrid cloud infrastructure that may include a first cloud infrastructure and a second cloud infrastructure that is different from the first cloud infrastructure, and so forth. The components of system 100 may be a combination of hardware and programming for performing a designated visualization function. In some instances, each component may include a processor and a memory, while programming code is stored on that memory and executable by a processor to perform a designated visualization function.

[0047] The data profiler 106 may be a combination of hardware and programming for performing a designated function. For example, the data profiler 106 may include programming to receive the input data instances 102, and perform data pre-processing on the input data instances 102, including identifying a feature type for each data feature of the input data instances 102. Also, for example, the data profiler 106 may include programming to be communicatively linked to the other components of system 100. In some instances, the data profiler 106 may include a processor and a memory, while programming code is stored on that memory and executable by a processor to perform designated functions.

[0048] The statistical model engine 114 may include hardware to physically store the collection of shared models 114A of the input data instances 102, and processors to physically process the model. Statistical model engine 114 may include software algorithms to perform statistical analyses of the input data instances 102, and algorithms to provide the collection of shared models 114A to the plurality of contextual model engines 116(1), 116(2), ..., 116(x). Statistical model engine 114 may include hardware, including physical processors and memory to house and process such software algorithms. Statistical model engine 114 may also include physical networks to be communicatively linked to the other components of system 100.

[0049] As another example, the plurality of contextual model engines 116(1), 116(2), ..., 116(x) may include hardware for their respective bounded memories. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may include hardware to physically store the contextual models responsive to the relevance

criteria 104. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may include software programming to generate the contextual models based on the collection of shared models 114A. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may include software programming to dynamically interact with the other components of system 100 to receive the collection of shared models 114A, and to be applicable to specific analytics functions. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may include software programming to adjust respective bounded memories to optimize a tradeoff between computational complexity and statistical accuracy of a respective contextual model. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may also include hardware, including physical processors and memory to house and process such software algorithms. The plurality of contextual model engines 116(1), 116(2), ..., 116(x) may also include physical networks to be communicatively linked to the other components of system 100. [0050] Likewise, the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include a combination of hardware and software programming. For example, the plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include hardware to store an automatically detected a group of the input data instances 102. The plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include software programming to automatically detect the group of the input data instances 102, and to generate results of data analytics performed on the detected group. The plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may include software programming to manage the plurality of contextual caches 120(1), 120(2), ..., 120(x). The plurality of contextual analytics engines 118(1), 118(2), ..., 118(x) may also include hardware, including physical processors and memory to house and process such software algorithms, and physical networks to be communicatively linked to the other components of system 100.

[0051] Also, for example, the plurality of contextual caches 120(1), 120(2), ..., 120(x) may include a combination of hardware and software programming. For example, the plurality of contextual caches 120(1), 120(2), ..., 120(x) may include hardware for the respective bounded storage capacities to store a

progressively updated sub-collection of the generated results. The plurality of contextual caches 120(1), 120(2), ..., 120(x) may include software programming to be responsive to the relevance criteria 104 to progressively update the sub-collection of the generated results. The plurality of contextual caches 120(1), 120(2), ..., 120(x) may also include hardware, including physical processors and memory to house and process such software algorithms, and physical networks to be communicatively linked to the other components of system 100.

[0052] The interaction processor 110 may include a combination of hardware and software programming. For example, the interaction processor 110 may include hardware to be communicatively linked to the analytics interface 112. Also, for example, the interaction processor 110 may include hardware to be communicatively linked to interactive graphical user interfaces. Also, for example, the interaction processor 110 may include a computing device to provide the interactive graphical user interfaces. The interaction processor 110 may include software programming to interact with SMEs via the analytics interface 112, and receive relevance criteria 104. The interaction processor 110 may include software programming to access various components of system 100 to provide data that may modify a visual representation provided by the analytics interface 112. The interaction processor 110 may also include hardware, including physical processors and memory to house and process such software algorithms, and physical networks to be communicatively linked to the other components of system 100.

[0053] The computing device may be, for example, a web-based server, a local area network server, a cloud-based server, a notebook computer, a desktop computer, an all-in-one system, a tablet computing device, a mobile phone, an electronic book reader, or any other electronic device suitable for provisioning a computing resource to perform a unified visualization interface. Computing device may include a processor and a computer-readable storage medium.

[0054] Although system 100 is described as linked to an analytics interface 112, system 100 may communicate with multiple analytics interfaces. For example, system 100 may communicate with a first analytics interface that is an anomaly processor, and the plurality of CANEs 118(1), 118(2), ..., 118(x) may detect first

groups of input data instances 102 that are relevant to anomaly processing. As another example, system 100 may communicate with a second analytics interface that is a feature correspondence processor, and the plurality of CANEs 118(1), 118(2), ..., 118(x) may detect second groups of input data instances 102 that are relevant to feature correspondence processing. Also, for example, system 100 may communicate with a third analytics interface that is cluster processor, and the plurality of CANEs 118(1), 118(2), ..., 118(x) may detect third groups of input data instances 102 that are relevant to cluster detection. In some examples, system 100 may communicate with multiple analytics interfaces simultaneously, and groups of input data instances 102 may be selectively identified and processed for a relevant task. In some examples, the interaction processor 110 may include multiple analytics widgets that may be communicatively linked to each of the multiple analytics interfaces to provide for an efficient and streamlined transfer of relevance criteria 104, and to provide the progressively updated sub-collection of the generated results to the relevant analytics interface.

[0032] Figure 2 is a block diagram illustrating one example of a processing system 200 for implementing the system 100 for scalable interactive data analytics. Processing system 200 includes a processor 202, a memory 204, input devices 218, and output devices 220. Processor 202, memory 204, input devices 218, and output devices 220 are coupled to each other through communication link (e.g., a bus).

[0033] Processor 202 includes a Central Processing Unit (CPU) or another suitable processor. In some examples, memory 204 stores machine readable instructions executed by processor 202 for operating processing system 200. Memory 204 includes any suitable combination of volatile and/or non-volatile memory, such as combinations of Random Access Memory (RAM), Read-Only Memory (ROM), flash memory, and/or other suitable memory.

[0055] Memory 204 also stores instructions to be executed by processor 202 including instructions for a data profiler 206, instructions for an interaction processor 208, instructions for a statistical model engine 210, instructions for contextual model engines 212, instructions for contextual analytics engines 214,

and instructions for contextual caches 216. In some examples, instructions for a data profiler 206, instructions for an interaction processor 208, instructions for a statistical model engine 210, instructions for contextual model engines 212, instructions for contextual analytics engines 214, and instructions for contextual caches 216, include instructions for the data profiler 106, the interaction processor 110, the statistical model engine 114, the plurality of contextual model engines 116(1), 116(2), ..., 116(x), the plurality of CANEs 118(1), 118(2), ..., 118(x), and the plurality of contextual caches 120(1), 120(2), ..., 120(x), respectively, as previously described and illustrated with reference to Figure 1.

[0056] Processor 202 executes instructions for a data profiler 206 to identify a feature type for each data feature of input data instances. In some examples, processor 202 executes instructions for a data profiler 206 to analyze each data feature to detect its format (e.g. integer, real, date/time, string, free text, domain-specific: URL/IP/Email). In some examples, processor 202 executes instructions for a data profiler 206 to identify feature information type, and classify features as categorical, ordinal, numerical, temporal, unique-index, or compound. In some examples, processor 202 executes instructions for a data profiler 206 to determine features complexities for the input data instances 102, where the features complexities are indicative of memory consumption to be allocated to generate the contextual models.

[0057] Processor 202 executes instructions for an interaction processor 208 to receive relevance criteria from an analytics interface. In some examples, processor 202 executes instructions for an interaction processor 208 to process interactions between the components of the system and an external interface such as the analytics interface.

[0058] Processor 202 executes instructions for a statistical model engine 210 to generate a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria. In some examples, processor 202 executes instructions for a statistical model engine 210 to receive structured, mixed data feature types (numerical, categorical, and ordinal) from the data profiler. In some examples, processor 202 executes instructions for a statistical model engine 210 to

progressively model the data with joint distributions of features to generate the shared models. In some examples, processor 202 executes instructions for a statistical model engine 210 to generate adaptive histograms which may be appropriate for targeted analytics functions.

[0059] Processor 202 executes instructions for contextual model engines 212 to generate contextual models responsive to the relevance criteria, and based on the shared models. In some examples, processor 202 executes instructions for contextual model engines 212 to capture different statistical aspects of the input data (e.g., frequent values, rare values, variability, coupling, etc.). In some examples, processor 202 executes instructions for contextual model engines 212 to adjust respective bounded memories to balance computational complexity against statistical accuracy of a contextual model generated by a contextual model engine.

[0060] Processor 202 executes instructions for contextual analytics engines 214 to automatically detect a group of the input data instances based on a contextual model generated by a respective contextual model engine of the plurality of contextual model engines, and to generate results of data analytics performed on the detected group. In some examples, processor 202 executes instructions for contextual analytics engines 214 to utilize a set of parameters that capture SME's intents in a compute context (e.g. feature-relevance weights, selected items, transformation rules). In some examples, processor 202 executes instructions for contextual analytics engines 214 to produce specific analytics results based on incoming data features.

[0061] Processor 202 executes instructions for contextual caches 216 to store a progressively updated sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function. In some examples, processor 202 executes instructions for contextual caches 216 to store a cache of results (detected anomalies, discovered clusters, etc.) that scored with highest relevance at a given moment. In some examples, processor 202 executes instructions for contextual caches 216 to maintain respective bounded storage capacities that are small enough to be independent of the input data size or streaming velocity.

[0062] In some examples, processor 202 executes instructions for the interaction processor 208 to provide the sub-collection of results to the analytics interface. In some examples, processor 202 executes instructions for the interaction processor 208 to receive modified relevance criteria from the analytics interface, identify a contextual cache based on the modified relevance criteria, search the identified contextual cache for additional results based on the modified relevance criteria, and provide the additional results to the analytics interface. In some examples, processor 202 executes instructions for contextual caches 216 to progressively update the identified contextual cache based on the modified relevance criteria.

[0063] Input devices 218 include a keyboard, mouse, data ports, and/or other suitable devices for inputting information into processing system 200. In some examples, input devices 218, such as a computing device, are used by the interaction processor to receive relevance criteria via an analytics interface. Output devices 220 include a monitor, speakers, data ports, and/or other suitable devices for outputting information from processing system 200. In some examples, output devices 220 are used to provide the sub-collection of results to the analytics interface.

[0064] Figure 3 is a block diagram illustrating one example of a computer readable medium for scalable interactive data analytics. Processing system 300 includes a processor 302, a computer readable medium 316, a data profiler 304, an interaction processor 306, a statistical model engine 308, contextual model engines 310, contextual model engines 312, and contextual caches 314. Processor 302, computer readable medium 316, data profiler 304, interaction processor 306, statistical model engine 308, contextual model engines 310, contextual model engines 312, and contextual caches 314 are coupled to each other through communication link (e.g., a bus).

[0065] Processor 302 executes instructions included in the computer readable medium 316. Computer readable medium 316 includes feature type identification instructions 318 of the data profiler 304 to identify a feature type for each data feature of input data instances.

[0066] Computer readable medium 316 includes relevance criteria receipt instructions 320 of the interaction processor 306 to receive relevance criteria from an analytics interface.

[0067] Computer readable medium 316 includes shared model generation instructions 322 of the statistical model engine 308 to generate a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria.

[0068] Computer readable medium 316 includes contextual model generation instructions 324 of the contextual model engines 310 with bounded memories to generate contextual models responsive to the relevance criteria based on the shared models.

[0069] Computer readable medium 316 includes automatic data detection instructions 326 of the contextual analytics engines 312 to automatically detect a group of the input data instances based on a contextual model generated by a respective contextual model engine.

[0070] Computer readable medium 316 includes analytics results generation instructions 328 of the contextual analytics engines 312 to generate results of data analytics performed on the detected group.

[0071] Computer readable medium 316 includes analytics results storage instructions 330 of the contextual caches 314 with bounded storage capacities, to store a sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function.

[0072] Computer readable medium 316 includes analytics results providing instructions 332 of the interaction processor 306 to provide the sub-collection of results to the analytics interface.

[0073] Computer readable medium 316 includes progressive cache updating instructions 334 of the contextual analytics engines 312 to progressively update the plurality of contextual caches based on modified relevance criteria received from the analytics interface. In some examples, computer readable medium 316 includes progressive cache updating instructions 334 of the contextual caches 314 to progressively update the plurality of contextual caches based on modified relevance criteria received from the analytics interface.

[0074] As used herein, a “computer readable medium” may be any electronic, magnetic, optical, or other physical storage apparatus to contain or store information such as executable instructions, data, and the like. For example, any computer readable storage medium described herein may be any of Random Access Memory (RAM), volatile memory, non-volatile memory, flash memory, a storage drive (e.g., a hard drive), a solid state drive, and the like, or a combination thereof. For example, the computer readable medium 316 can include one of or multiple different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; optical media such as compact disks (CDs) or digital video disks (DVDs); or other feature types of storage devices.

[0075] As described herein, various components of the processing system 300 are identified and refer to a combination of hardware and programming configured to perform a designated visualization function. As illustrated in Figure 3, the programming may be processor executable instructions stored on tangible computer readable medium 316, and the hardware may include processor 302 for executing those instructions. Thus, computer readable medium 316 may store program instructions that, when executed by processor 302, implement the various components of the processing system 300.

[0076] Such computer readable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured single component or multiple components. The storage medium or media can be located either in the machine running the machine-readable instructions, or located at a remote site from which machine-readable instructions can be downloaded over a network for execution.

[0077] Computer readable medium 316 may be any of a number of memory components capable of storing instructions that can be executed by Processor 302. Computer readable medium 316 may be non-transitory in the sense that it

does not encompass a transitory signal but instead is made up of one or more memory components configured to store the relevant instructions. Computer readable medium 316 may be implemented in a single device or distributed across devices. Likewise, processor 302 represents any number of processors capable of executing instructions stored by computer readable medium 316. Processor 302 may be integrated in a single device or distributed across devices. Further, computer readable medium 316 may be fully or partially integrated in the same device as processor 302 (as illustrated), or it may be separate but accessible to that device and processor 302. In some examples, computer readable medium 316 may be a machine-readable storage medium

[0078] Figure 4 is a flow diagram illustrating one example of a method for scalable interactive data analytics. At 400, a feature type for each data feature of input data instances may be identified. At 402, relevance criteria may be received from an analytics interface via an interaction processor. At 404, a collection of shared models may be generated via a statistical model engine, the generation based on joint distributions of data features, where each shared model may be targeted to an analytics function incorporating the relevance criteria. At 406, contextual models responsive to the relevance criteria may be generated based on the shared models, and via a plurality of contextual model engines with bounded memories. At 408, a group of the input data instances may be automatically detected via a plurality of contextual analytics engines, the detection based on a contextual model generated by a respective contextual model engine. At 410, results of data analytics performed on the detected group may be generated. At 412, a sub-collection of the generated results may be stored in a plurality of contextual caches with bounded storage capacities, the sub-collection indicative of high relevance to the analytics function. At 414, the sub-collection of results may be provided to the analytics interface via the interaction processor.

[0079] In some examples, the bounded memory of a contextual model engine is adjusted to balance computational complexity against statistical accuracy of a contextual model generated by the contextual model engine.

[0080] In some examples, the method may further include progressively updating the plurality of contextual caches based on modified relevance criteria received from the analytics interface.

[0081] In some examples, modified relevance criteria may be received from the analytics interface, a contextual cache may be identified based on the modified relevance criteria, the identified contextual cache may be searched for additional results based on the modified relevance criteria, and the additional results may be provided to the analytics interface. In some examples, the identified contextual cache may be progressively updated based on the modified relevance criteria.

[0082] In some examples, the bounded storage capacity of the respective contextual caches may be substantially smaller than the size of the input data.

[0083] In some examples, the size of the group of the input data instances may be much smaller than the size of the input data.

[0084] In some examples, the method may further include determining the bounded storage capacities of the respective contextual caches, the determining independent of the size of the input data and the rate at which input data instances are received.

[0085] In some examples, the method includes determining features complexities for the input data instances 102, the features complexities being indicative of memory consumption to be allocated to generate the contextual models

[0086] In some examples, the relevance criteria may include at least one of feature-relevance weights, selected items, and transformation rules.

[0087] In some examples, the input data instances may include rare values, and the plurality of contextual analytics engines may include an anomaly processor to detect anomalies based on the rare values.

[0088] In some examples, the data features of the input data instances may include frequent combinations of data feature values, and the plurality of contextual analytics engines may include a cluster processor to detect clusters based on the frequent combinations of the data feature values.

[0089] Examples of the disclosure provide a generalized system for scalable interactive data analytics. The generalized system provides an approach for interactive big data analytics of unknown, structured data sets. The input data may be unknown in the sense that the system may not have prior knowledge on semantics of data features or their statistical characteristics. The interactive system described herein constructs the required context, while the system processes the input data. Furthermore, the approach described herein recognizes that as the size of data sets increases, the relative importance of each item may tend to drop. Accordingly, the generalized system provides partial results as soon as such results are detected, and updates results as more data is processed.

[0090] As disclosed herein, when the size of input data is large, the generalized system defines a clear boundary between the data processing part of a system and the interactive part. Lengthy computations on large data sets are treated using a stream processing methodology. Accordingly, when a context is identified, the analytics are dynamically modified by the interactive part of the system. By decoupling big data processing from interaction processing, the system can scale up without affecting its responsiveness.

[0091] Although specific examples have been illustrated and described herein, especially as related to categorical and numerical data, the examples illustrate applications to any dataset. Accordingly, there may be a variety of alternate and/or equivalent implementations that may be substituted for the specific examples shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or variations of the specific examples discussed herein. Therefore, it is intended that this disclosure be limited only by the claims and the equivalents thereof.

CLAIMS

1. A system for scalable interactive data analytics, the system comprising:
 - a data profiler to identify a feature type for each data feature of input data instances;
 - an interaction processor to receive relevance criteria from an analytics interface;
 - a statistical model engine to generate a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria;
 - a plurality of contextual model engines with bounded memories to generate contextual models responsive to the relevance criteria based on the shared models;
 - a plurality of contextual analytics engines to automatically detect a group of the input data instances based on a contextual model generated by a respective contextual model engine, and generate results of data analytics performed on the detected group; and
 - a plurality of contextual caches with bounded storage capacities, each contextual cache to store a progressively updated sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function.
2. The system of claim 1, wherein the interaction processor is to further provide the sub-collection of results to the analytics interface.
3. The system of claim 1, wherein the interaction processor is to further:
 - receive modified relevance criteria from the analytics interface;
 - identify a contextual cache based on the modified relevance criteria;
 - search the identified contextual cache for additional results based on the modified relevance criteria; and

provide the additional results to the analytics interface.

4. The system of claim 3, wherein the identified contextual cache is progressively updated based on the modified relevance criteria.
5. The system of claim 1, wherein the bounded storage capacity of the respective contextual caches is substantially smaller than the size of the input data.
6. The system of claim 1, wherein the bounded storage capacities of the respective contextual caches are independent of the size of the input data and the rate at which input data instances are received.
7. The system of claim 1, wherein the data profiler is to further determine features complexities for the input data instances, the features complexities indicative of memory consumption to be allocated to generate the contextual models.
8. The system of claim 1, wherein the relevance criteria include at least one of feature-relevance weights, selected items, and transformation rules.
9. The system of claim 1, wherein the input data instances include rare values, and the plurality of contextual analytics engines include an anomaly processor to detect anomalies based on the rare values.
10. The system of claim 1, wherein the data features of the input data instances include frequent combinations of data feature values, and the plurality of contextual analytics engines include a cluster processor to detect clusters based on the frequent combinations of the data feature values.
11. The system of claim 1, wherein the bounded memory of a contextual model engine of the plurality of contextual model engines is adjusted to

balance computational complexity against statistical accuracy of a contextual model generated by a contextual model engine.

12. A method for scalable interactive data analytics, the method comprising:

- identifying a feature type for each data feature of input data instances;

- receiving, via an interaction processor, relevance criteria from an analytics interface;

- generating, via a statistical model engine, a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria;

- generating, via a plurality of contextual model engines with bounded memories, contextual models responsive to the relevance criteria based on the shared models;

- automatically detecting, via a plurality of contextual analytics engines, a group of the input data instances based on a contextual model generated by a respective contextual model engine;

- generating results of data analytics performed on the detected group;

- storing, in a plurality of contextual caches with bounded storage capacities, a sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function; and

- providing, via the interaction processor, the sub-collection of results to the analytics interface.

13. The method of claim 12, further comprising progressively updating the plurality of contextual caches based on modified relevance criteria received from the analytics interface.

14. The method of claim 11, further comprising determining the bounded storage capacities of the respective contextual caches, the determining independent of the size of the input data and the rate at which input data instances are received.

15. A non-transitory computer readable medium comprising executable instructions to:
- identify a feature type for each data feature of input data instances;
 - receive relevance criteria from an analytics interface;
 - generate, via a statistical model engine, a collection of shared models based on joint distributions of data features, each shared model targeted to an analytics function incorporating the relevance criteria;
 - generate, via a plurality of contextual model engines with bounded memories, contextual models responsive to the relevance criteria based on the shared models;
 - automatically detect, via a plurality of contextual analytics engines, a group of the input data instances based on a contextual model generated by a respective contextual model engine;
 - generate results of data analytics performed on the detected group;
 - store, in a plurality of contextual caches with bounded storage capacities, a sub-collection of the generated results, the sub-collection indicative of high relevance to the analytics function;
 - provide, via the interaction processor, the sub-collection of results to the analytics interface; and
 - progressively update the plurality of contextual caches based on modified relevance criteria received from the analytics interface.

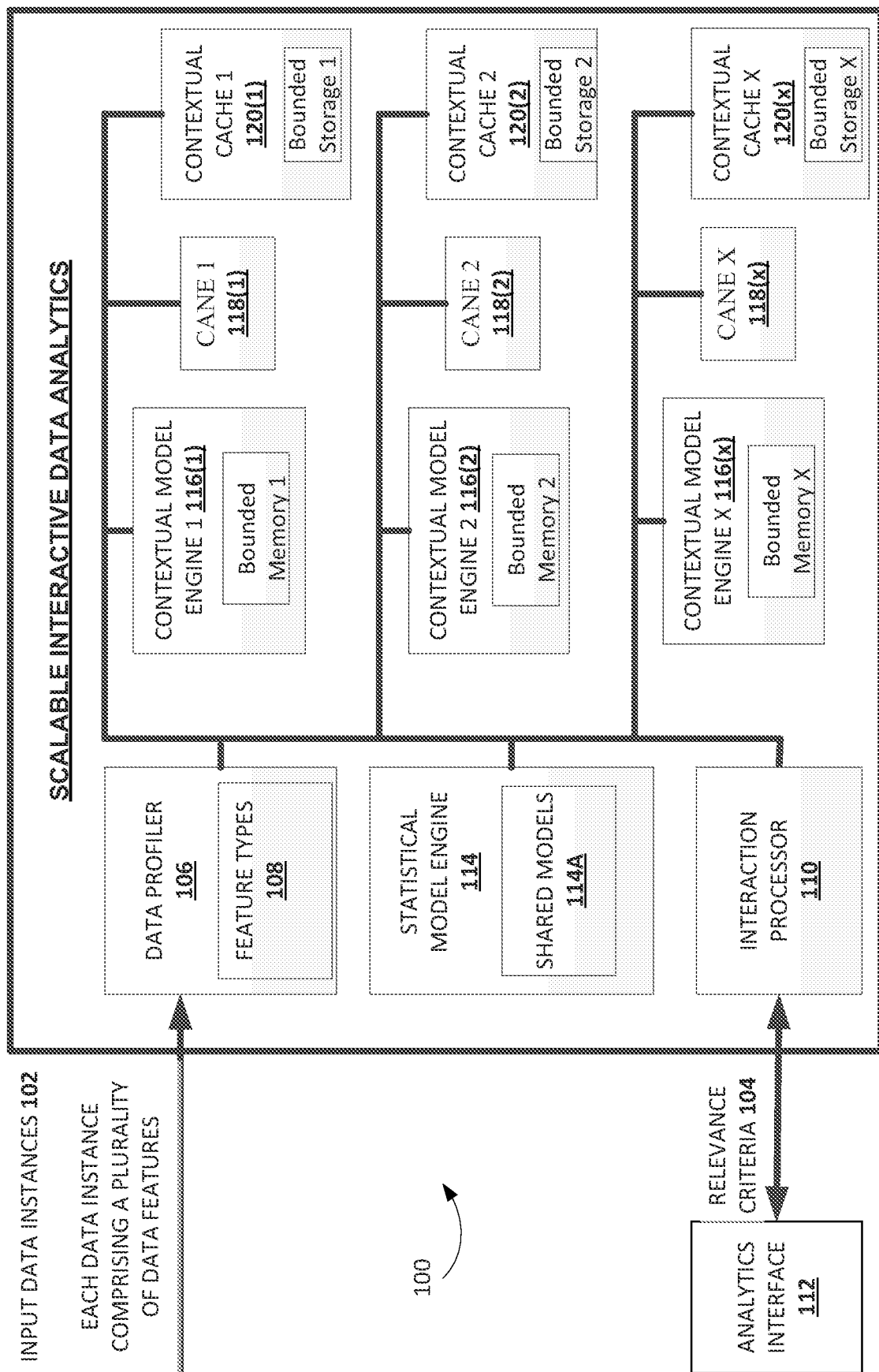
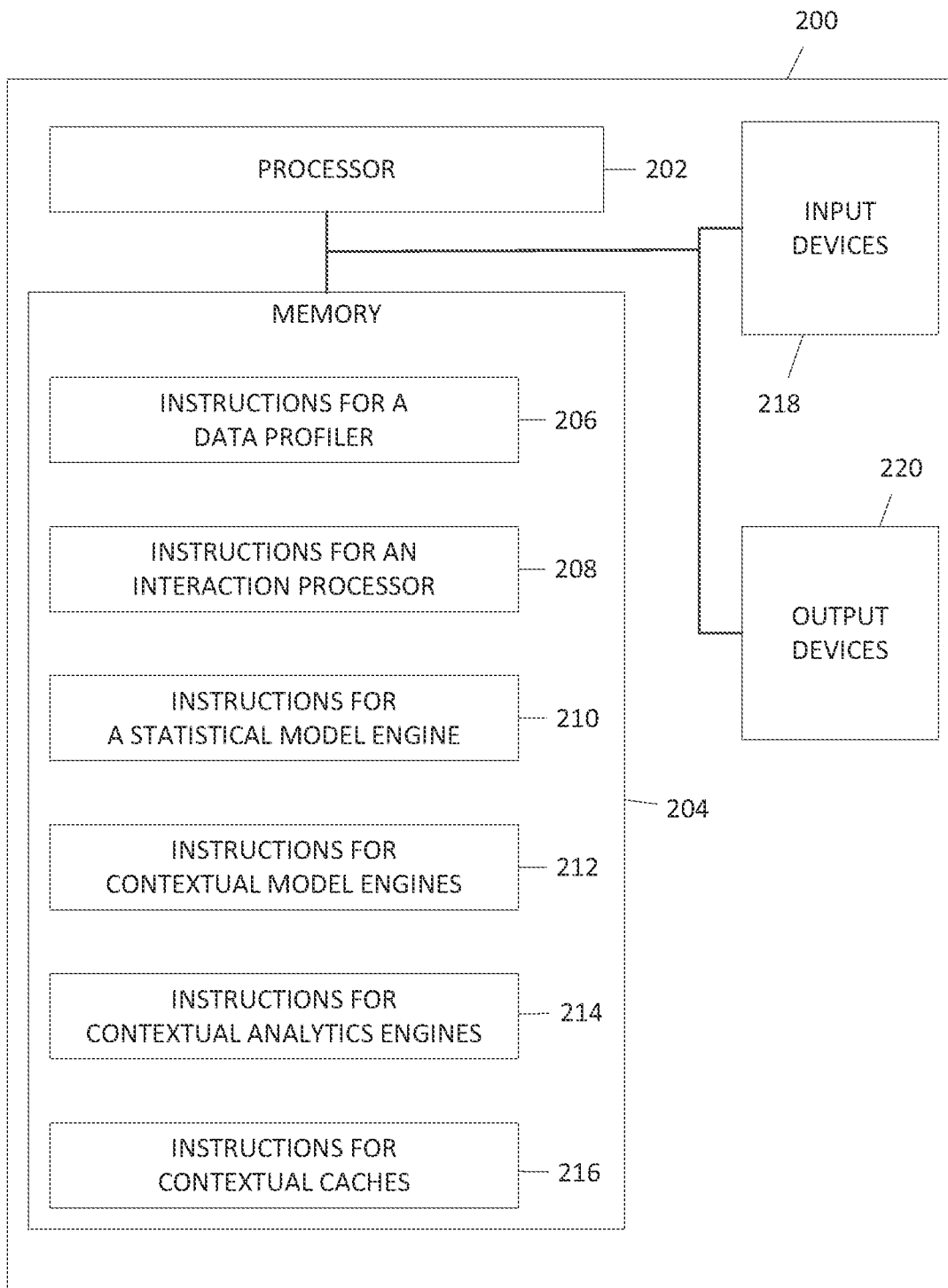
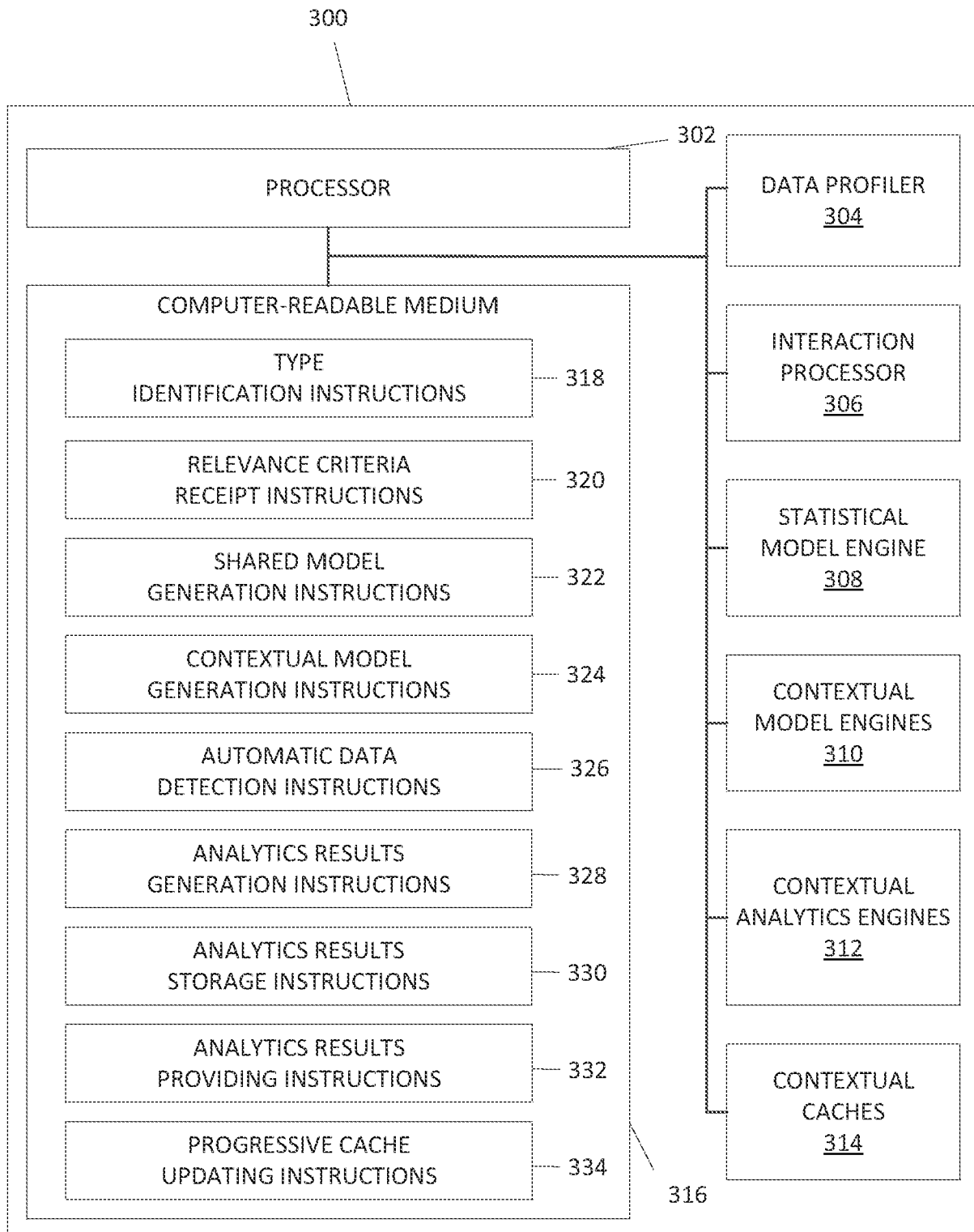
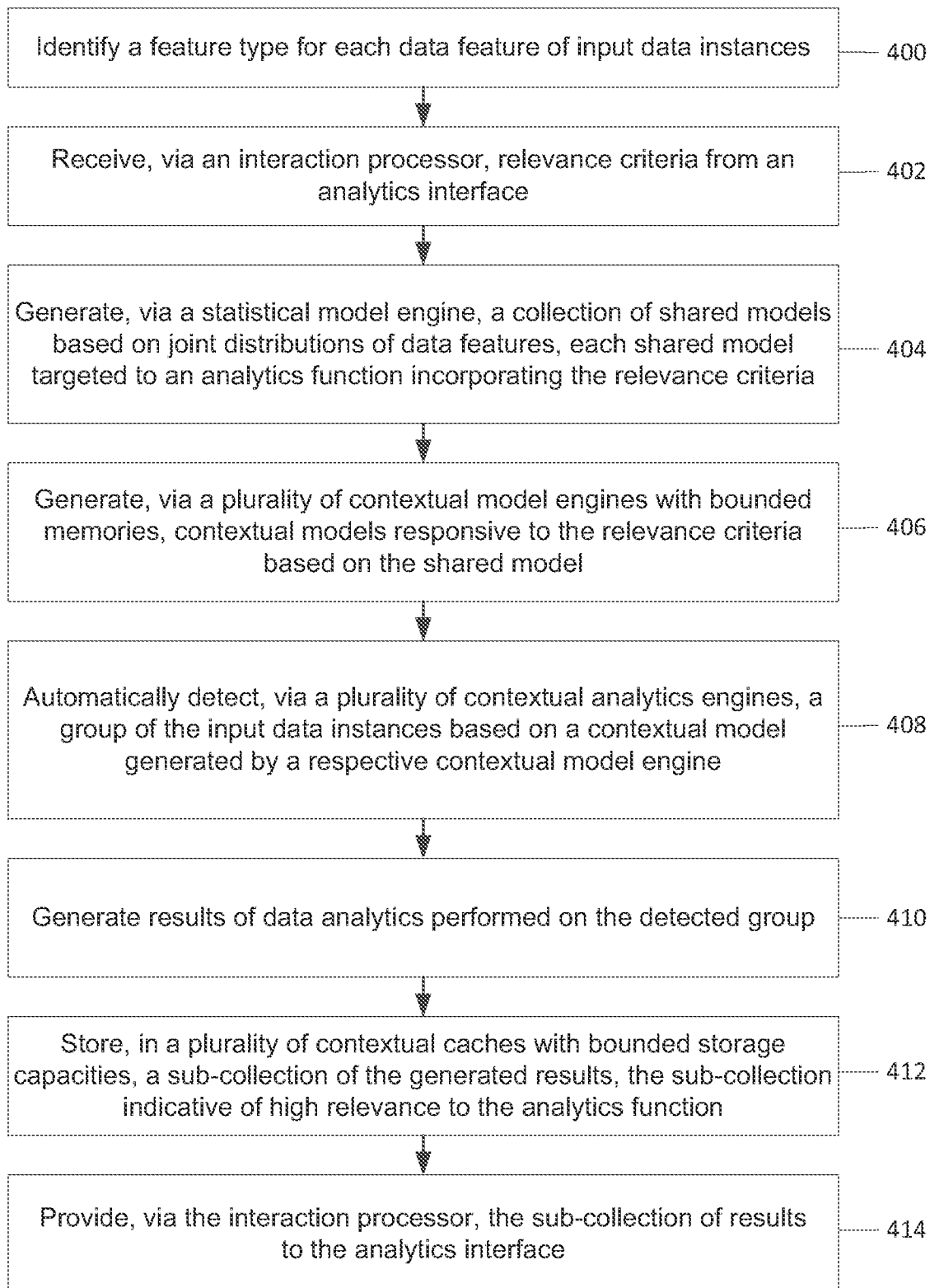


FIG. 1

**Fig. 2**

**Fig. 3**

**Fig. 4**

A. CLASSIFICATION OF SUBJECT MATTER**G06F 17/00(2006.01)i, G06F 17/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/00; G06F 3/048; G06Q 10/06; G06N 99/00; G06F 17/30; G06F 7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: relevance criteria, model, contextual, feature, data analytics, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0343997 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 20 November 2014 See paragraphs [0004] and [0025]-[0033]; claims 1, 8, and 15; and figure 1.	1-15
A	US 2014-0258197 A1 (HASAN DAVULCU et al.) 11 September 2014 See paragraphs [0030] and [0102]-[0103]; and figures 1A and 3.	1-15
A	US 2013-0268889 A1 (NIMROD BARAK et al.) 10 October 2013 See paragraphs [0074]-[0100] and figures 3A-3E.	1-15
A	EP 1622044 A1 (MICROSOFT CORPORATION) 01 February 2006 See paragraphs [0023]-[0029] and figures 1-2.	1-15
A	US 2007-0043723 A1 (ELAN BITAN et al.) 22 February 2007 See paragraphs [0181]-[0182] and figure 3.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

31 December 2015 (31.12.2015)

Date of mailing of the international search report

31 December 2015 (31.12.2015)

Name and mailing address of the ISA/KR

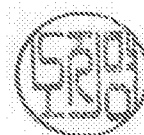
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/020164

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0343997 A1	20/11/2014	None	
US 2014-0258197 A1	11/09/2014	None	
US 2013-0268889 A1	10/10/2013	US 9170701 B2	27/10/2015
EP 1622044 A1	01/02/2006	CN 1655149 A	17/08/2005
		EP 1622044 B1	11/09/2013
		JP 2005-242998 A	08/09/2005
		JP 4139391 B2	27/08/2008
		KR 10-1130528 B1	27/03/2012
		KR 10-2005-0076764 A	27/07/2005
		US 2005-0165824 A1	28/07/2005
		US 7167866 B2	23/01/2007
US 2007-0043723 A1	22/02/2007	US 2006-0218146 A1	28/09/2006
		US 2011-0055185 A1	03/03/2011
		US 7617193 B2	10/11/2009