US 20200293951A1

(19) **United States**
(12) **Patent Application Publication**
Johnston et al.

(10) **Pub. No.: US 2020/0293951 A1**
(43) **Pub. Date: Sep. 17, 2020**

(54) **DYNAMICALLY OPTIMIZING A DATA SET DISTRIBUTION**

(71) Applicant: **Groupon, Inc.**, Chicago, IL (US)

(72) Inventors: **David Alan Johnston**, Portola Valley, CA (US); **Jonathan Esterhazy**, San Francisco, CA (US); **Gaston L'Huillier**, San Francisco, CA (US); **Hernan Enrique Arroyo Garcia**, Mountain View, CA (US)

(21) Appl. No.: **16/832,696**

(22) Filed: **Mar. 27, 2020**

Related U.S. Application Data

(63) Continuation of application No. 14/817,005, filed on Aug. 3, 2015, now Pat. No. 10,650,326.

(60) Provisional application No. 62/055,958, filed on Sep. 26, 2014, provisional application No. 62/039,314, filed on Aug. 19, 2014.

Publication Classification

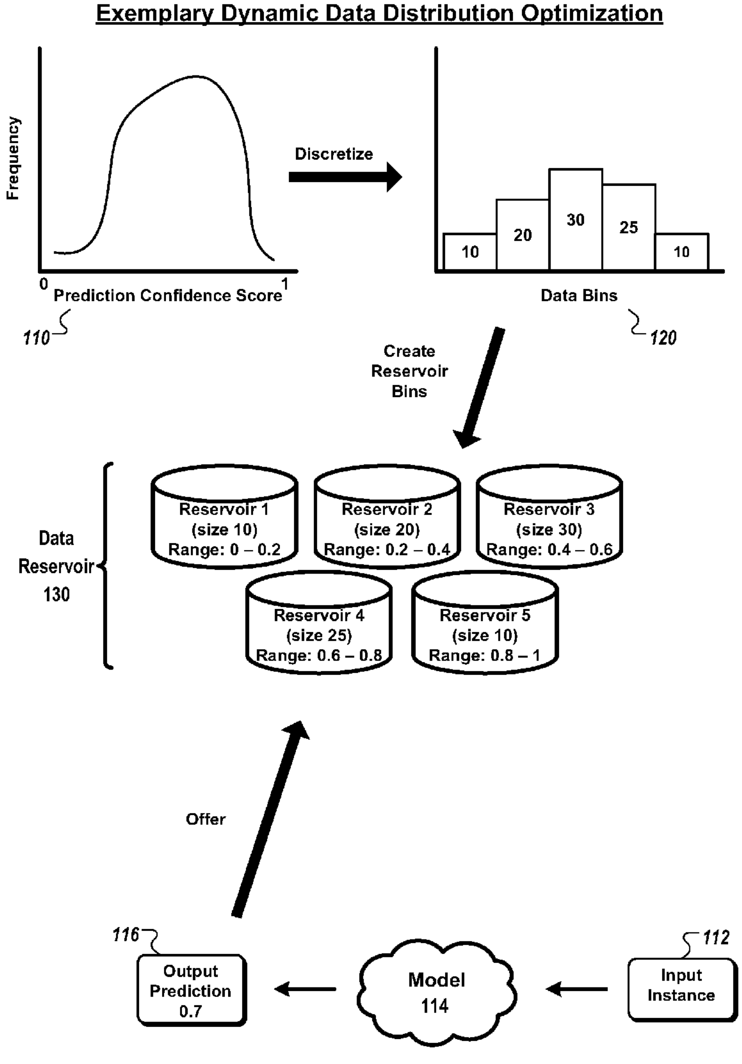
(51) **Int. Cl.**

G06N 20/00	(2006.01)
G06N 5/02	(2006.01)
G06F 16/23	(2006.01)
G06F 16/28	(2006.01)

(52) **U.S. Cl.**
CPC ***G06N 20/00*** (2019.01); ***G06F 16/285*** (2019.01); ***G06F 16/23*** (2019.01); ***G06N 5/02*** (2013.01)

(57) **ABSTRACT**

In general, embodiments of the present invention provide systems, methods and computer readable media configured to receive configuration data describing a desired data set distribution, and, in response to receiving new data instances, use the configuration data and the new data instances to dynamically optimize the distribution of data already stored in a data reservoir that has been discretized into bins representing the desired data distribution.


Exemplary Dynamic Data Distribution Optimization
Frequency
0
Prediction Confidence Score
1
110
Discretize
10
20
30
25
10
Data Bins
120
Create Reservoir Bins
Data Reservoir 130
Reservoir 1 (size 10) Range: 0 – 0.2
Reservoir 2 (size 20) Range: 0.2 – 0.4
Reservoir 3 (size 30) Range: 0.4 – 0.6
Reservoir 4 (size 25) Range: 0.6 – 0.8
Reservoir 5 (size 10) Range: 0.8 – 1
Offer
116
Output Prediction 0.7
Model 114
112
Input Instance

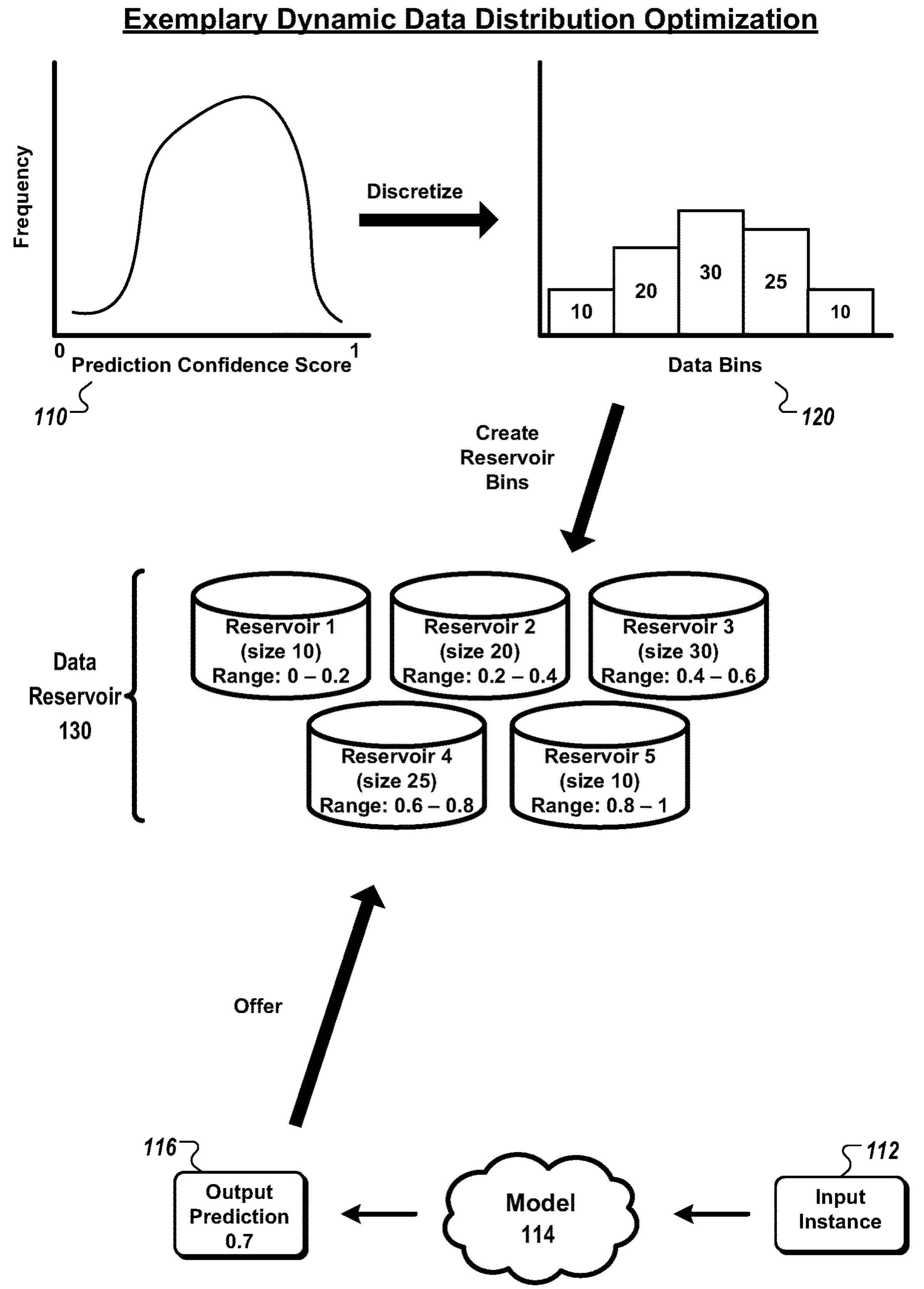

Exemplary Dynamic Data Distribution Optimization
Frequency
0
1
Prediction Confidence Score
110
Discretize
10
20
30
25
10
Data Bins
120
Create Reservoir Bins
Data Reservoir 130
Reservoir 1 (size 10) Range: 0 – 0.2
Reservoir 2 (size 20) Range: 0.2 – 0.4
Reservoir 3 (size 30) Range: 0.4 – 0.6
Reservoir 4 (size 25) Range: 0.6 – 0.8
Reservoir 5 (size 10) Range: 0.8 – 1
Offer
116
Output Prediction 0.7
Model 114
112
Input Instance

FIG. 1

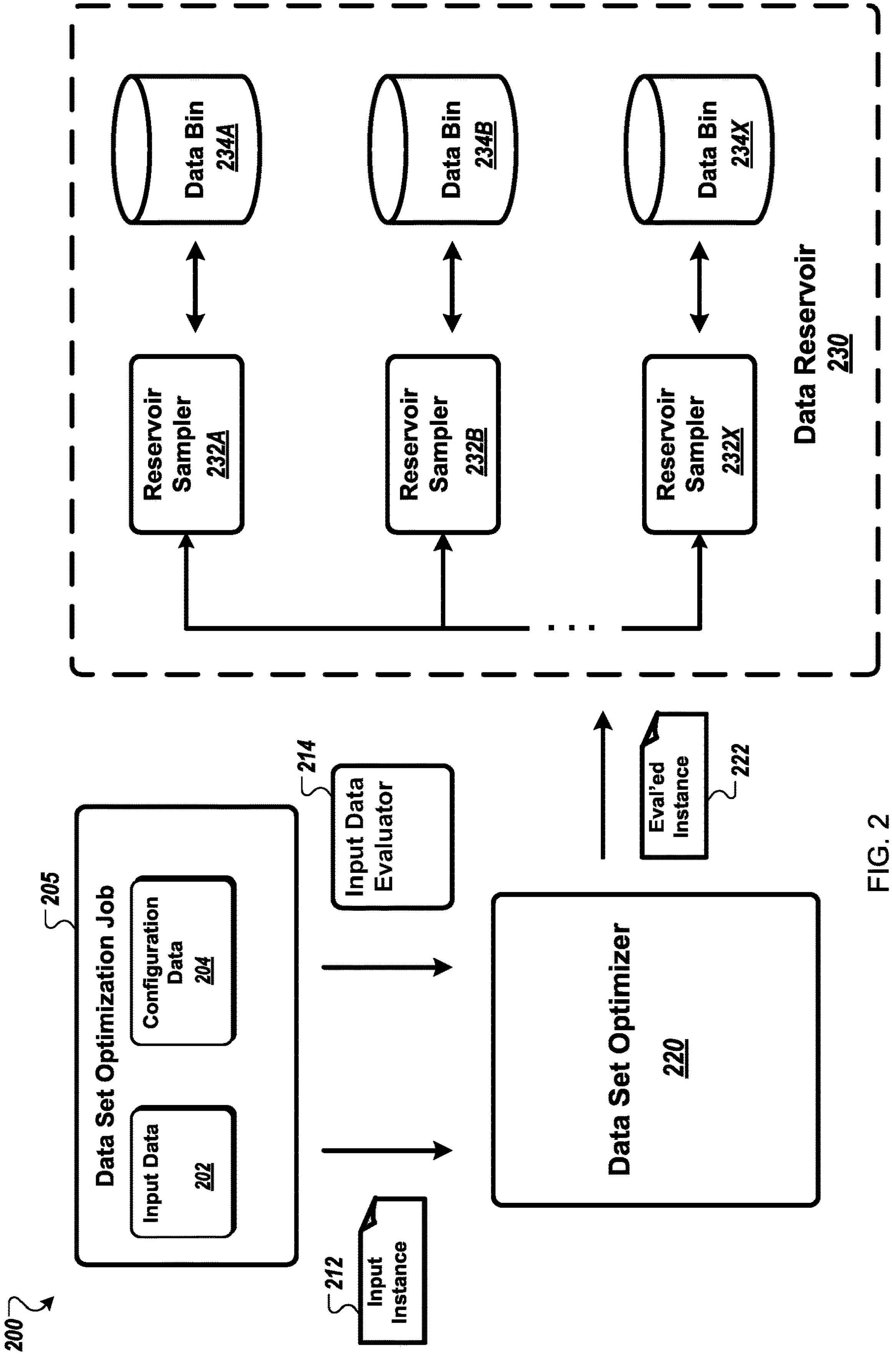

200
205
Data Set Optimization Job
Input Data
202
Configuration Data
204
212
Input Instance
214
Input Data Evaluator
Data Set Optimizer
220
Eval'ed Instance
222
Reservoir Sampler
232A
Reservoir Sampler
232B
Reservoir Sampler
232X
Data Bin
234A
Data Bin
234B
Data Bin
234X
...
Data Reservoir
230

FIG. 2

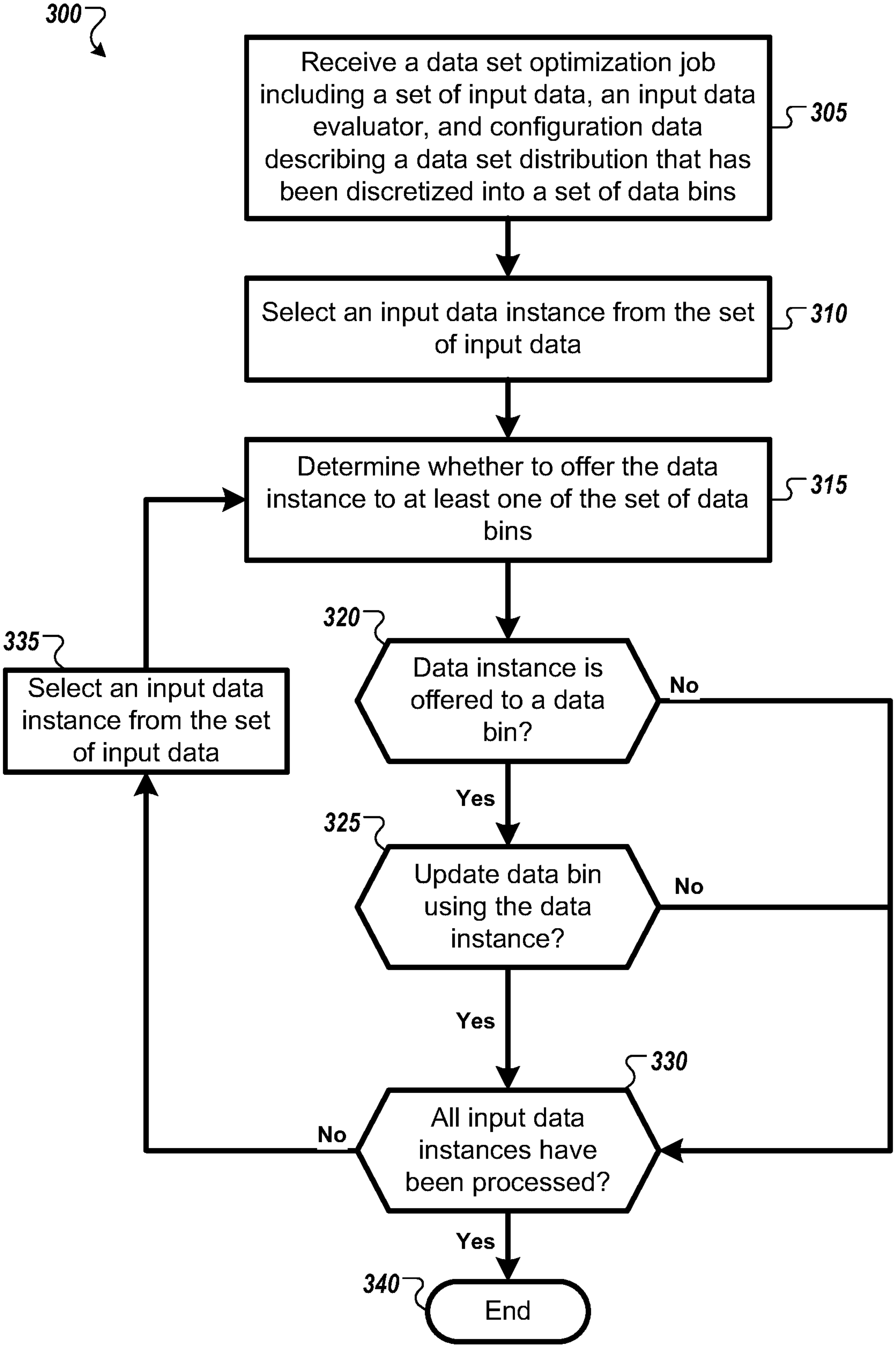

300
Receive a data set optimization job including a set of input data, an input data evaluator, and configuration data describing a data set distribution that has been discretized into a set of data bins
305
Select an input data instance from the set of input data
310
Determine whether to offer the data instance to at least one of the set of data bins
315
320
Data instance is offered to a data bin?
No
Yes
325
Update data bin using the data instance?
No
Yes
330
All input data instances have been processed?
No
Yes
335
Select an input data instance from the set of input data
340
End

FIG. 3

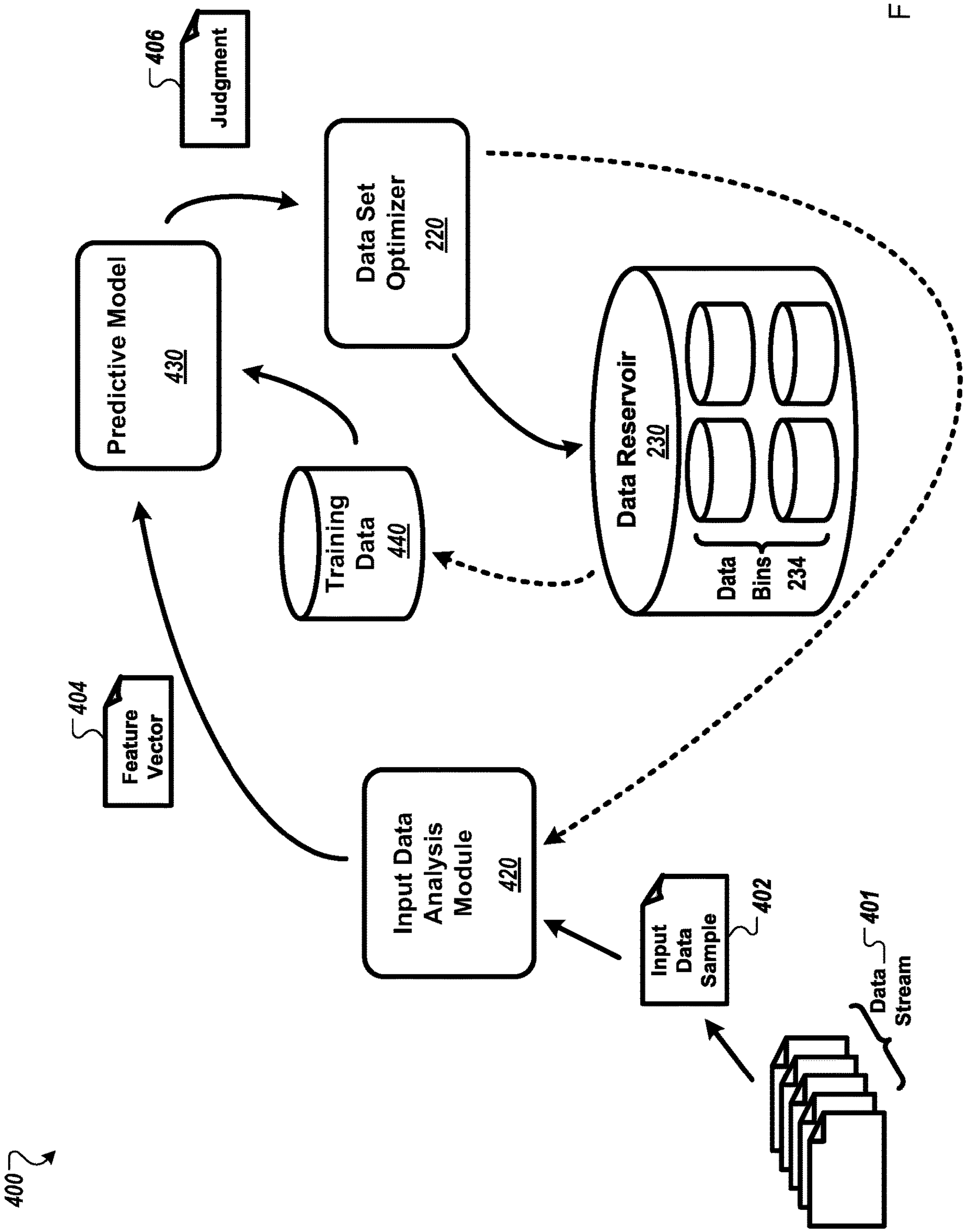

400
Judgment
406
Predictive Model
430
Data Set Optimizer
220
Training Data
440
Data Reservoir
230
Data Bins
234
Feature Vector
404
Input Data Analysis Module
420
Input Data Sample
402
Data Stream
401

FIG. 4

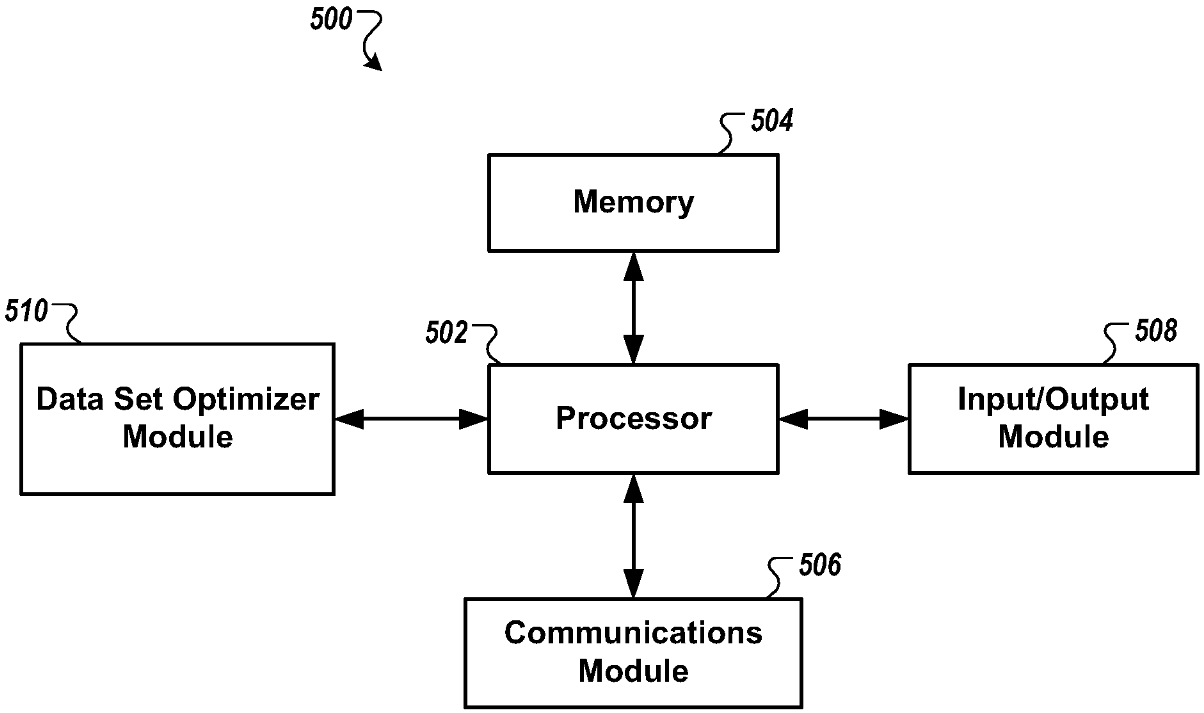

500
504
Memory
510
Data Set Optimizer Module
502
Processor
508
Input/Output Module
506
Communications Module

FIG. 5

DYNAMICALLY OPTIMIZING A DATA SET DISTRIBUTION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of and claims priority to U.S. patent application Ser. No. 14/817,005, titled "DYNAMICALLY OPTIMIZING A DATA SET DISTRIBUTION," and filed Aug. 3, 2015, which claims the benefit of U.S. Provisional Application No. 62/055,958, entitled "DYNAMICALLY OPTIMIZING A DATA SET DISTRIBUTION," and filed Sep. 26, 2014, and of U.S. Provisional Application No. 62/039,314, entitled "DYNAMICALLY OPTIMIZING A DATA SET DISTRIBUTION," and filed Aug. 19, 2014, the contents of which are hereby incorporated herein by reference in their entirety.

FIELD

[0002] Embodiments of the invention relate, generally, to dynamic optimization of a data set distribution.

BACKGROUND

[0003] A classifier is an example of an algorithm that may be derived using supervised machine learning. In order to make accurate predictions, supervised machine learning classifiers are derived through training using a set of labeled data examples. In modeling a classification problem in which the classifier must make a categorical prediction, the training data set should contain many labeled examples of each possible category to ensure that the classifier will make accurate predictions for new input examples that might fall into one of the categories.

[0004] A common way to improve a classifier's prediction performance is to sample a labeling data set from the general population, obtain true labels for the labeling set, and add these labels to the training set used to derive the classifier. For some classification problems in which instances of one or more of the classification categories are relatively rare (e.g., predicting gene mutations, earthquakes, or ad click throughs), the distribution in the general population of the true labels is skewed in favor of the most commonly occurring instances. If naïve (random) sampling from a general population were used to generate a labeling set, it is likely that the labeling set, after being labeled, also will have a skewed distribution of labels. A skewed distribution of labels may not add much support to the model to make predictions for rare events.

[0005] Current methods for dynamic optimization of a data set distribution exhibit a plurality of problems that make current systems insufficient, ineffective and/or the like. Through applied effort, ingenuity, and innovation, solutions to improve such methods have been realized and are described in connection with embodiments of the present invention.

SUMMARY

[0006] In general, embodiments of the present invention provide herein systems, methods and computer readable media configured to receive configuration data describing a desired data set distribution, and, in response to receiving new data instances, use the configuration data and the new data instances to dynamically optimize the distribution of data already stored in a data reservoir that has been discretized into bins representing the desired data distribution.

[0007] The details of one or more embodiments of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S)

[0008] Having thus described the invention in general terms, reference will now be made to the accompanying drawings, which are not necessarily drawn to scale, and wherein:

[0009] FIG. **1** illustrates an exemplary dynamic data distribution optimization scenario according to various embodiments of the invention in accordance with some embodiments discussed herein;

[0010] FIG. **2** illustrates an example system that can be configured to implement dynamic optimization of a data set distribution in accordance with some embodiments discussed herein;

[0011] FIG. **3** is a flow diagram of an example method for optimizing a data set distribution in accordance with some embodiments discussed herein;

[0012] FIG. **4** illustrates an example adaptive data analysis system that is configured to include dynamic data set distribution optimization in accordance with some embodiments discussed herein; and

[0013] FIG. **5** illustrates a schematic block diagram of circuitry that can be included in a computing device, such as a dynamic data set distribution optimization system, in accordance with some embodiments discussed herein.

DETAILED DESCRIPTION

[0014] The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which some, but not all embodiments of the invention are shown. Indeed, this invention may be embodied in many different forms and should not be construed as being limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will satisfy applicable legal requirements. Like numbers refer to like elements throughout.

[0015] As described herein, system components can be communicatively coupled to one or more of each other. Though the components are described as being separate or distinct, two or more of the components may be combined into a single process or routine. The component functional descriptions provided herein including separation of responsibility for distinct functions is by way of example. Other groupings or other divisions of functional responsibilities can be made as necessary or in accordance with design preferences.

[0016] As used herein, the terms "data," "content," "information" and similar terms may be used interchangeably to refer to data capable of being captured, transmitted, received, displayed and/or stored in accordance with various example embodiments. Thus, use of any such terms should not be taken to limit the spirit and scope of the disclosure. Further, where a computing device is described herein to receive data from another computing device, the data may be received directly from the another computing device or may be received indirectly via one or more intermediary computing devices, such as, for example, one or more servers, relays, routers, network access points, base stations, and/or the like. Similarly, where a computing device is described herein to send data to another computing device, the data may be sent directly to the another computing device or may be sent indirectly via one or more intermediary computing devices, such as, for example, one or more servers, relays, routers, network access points, base stations, and/or the like.

[0017] Entity resolution is an example of a categorical classification problem for which the accuracy of the predictions made by a classifier derived using supervised machine learning requires a training data set having a balanced class distribution. In entity resolution, a classifier is trained to return “match” or “no match” in response to receiving a pair of input examples. In entity resolution, a reference is a description of an entity, which is a real-world object. A reference may be a listing in a database or a list (which may contain some duplicate entries). Multiple references may describe the same entity. A labeling set example is a pair of references, and a label is either “match” or “no match.”

[0018] Given a list of references of size n, potentially comparing every reference to every other reference would result in making n^2 comparisons. Of those comparisons, the expectation is to see O(n^2) non-matches and O(n) matches (assuming the list has some duplicate references, but is not all duplicate references). Therefore, the set of possible labeling examples from which to draw a training data set for an entity resolution classifier is heavily skewed towards non-matches. As previously described, if a labeling set were generated from a general population using naïve (random) sampling, it is likely that the labeling set, after being labeled, would have a distribution of labels skewed towards “no match.” A labeling set containing such a skewed distribution of labels may not add much support to the entity resolution model for making accurate predictions for the rare “match” events.

[0019] In embodiments, a declarative system for dynamic data distribution optimization (i.e., the system receives a configuration for collecting data samples, the configuration specifying a particular data sample distribution to be satisfied over time) enables selection of a labeling set that, added to a training data set, will optimally improve the performance of a machine learning algorithm derived from that training data set. In embodiments, dynamic data distribution optimization may improve classifier performance for the specific entity resolution case where non-matches are much more frequent than matches, as well as for any categorical classification model based on a general population with rare target variables.

[0020] FIG. **1** illustrates an exemplary dynamic data distribution optimization scenario according to various embodiments of the invention. The scenario is described for clarification and without limitation.

[0021] In embodiments, the output of a classifier predictive model in response to a new input example may include the prediction (e.g., “match” or “no match” for an entity resolution classification problem) as well as a prediction confidence value. In some embodiments, the prediction confidence value is a probability score between 0 and 1 that represents the distance between the value on which the prediction is based and the distance between that value and the decision boundary in decision space. The closer a prediction value is to the decision boundary, the lower the prediction confidence score.

[0022] In some embodiments, an active learning component may facilitate adaptation of a predictive model. Active learning, as described, for example, in Settles, Burr (2009), “Active Learning Literature Survey”, Computer Sciences Technical Report 1648. University of Wisconsin—Madison, is semi-supervised learning, in which the distribution of samples composing a training data set can be adjusted to optimally represent a machine learning problem by interactively querying a source of truth (e.g., an oracle) to assign labels to new data samples that are to be added to the training data set.

[0023] In reference to the example, a desired prediction accuracy of a classifier, represented by a distribution **110** of prediction confidence scores (in this case, a normal distribution), is discretized into a set of data bins **120**. Each of the data bins is a reservoir for storing input data instances that each generates a prediction associated with a confidence score that falls within a respective range of confidence scores. Each bin also has a reservoir size capacity, represented on the bar graph as a maximum number of input instances to be stored within the bin. In embodiments, the choice of the number of data bins and their relative capacities may be expressed as a configuration parameter representing a training data set distribution that is desired for a particular classification problem. For example, if the machine learning model is a binary classifier, a configuration parameter may specify a distribution of 50% predicted matches, 50% predicted non-matches. If the machine learning model is a probabilistic estimator, a configuration parameter may specify a Gaussian distribution of predicted probabilities that are centered around the confidence score of 0.5.

[0024] In embodiments, the set of reservoir bins may be used as a data reservoir **130** for maintaining a random sample of a population representing a particular classification problem. In some embodiments, the data reservoir **130** may be a “hopper” of interesting data points. In embodiments, the reservoir data sample may be selected from a data stream. In some embodiments, the data stream may be received from an online data processing system. In some alternative embodiments, the data stream may be generated from a very large data store.

[0025] In embodiments in which the sample in the data reservoir is selected from a data stream, the sample may be a temporally unbiased sample, in which the first data point that flows through the stream is equally likely to end up in the sample as the last (or any points in between). In some embodiments, a temporally unbiased sample may be selected from a data stream using a well-known technique called reservoir sampling. In embodiments, the sample in the data reservoir is selected such that the distribution of the target variable as predicted by the existing machine learning algorithm satisfies the desired distribution for the classification problem. Referring to the exemplary scenario, a model **114** generates an output prediction **116** with a confidence score of 0.7 for a particular input instance **112**. The input instance **112** and its output **116** are offered to reservoir bin **4**, which stores data instances for which the model **114** has generated predictions with confidence scores within the range of 0.6 to 0.8. In some embodiments, each data reservoir bin may be associated with a reservoir sampler that determines whether to update the reservoir bin using an offered input instance **112**. Reservoir samplers will be described in more detail with reference to FIG. **2**.

[0026] As such, and according to some example embodiments, the systems and methods described herein are therefore configured to receive configuration data describing a desired data set distribution, and, in response to receiving new data instances, use the configuration data and the new data instances to dynamically optimize the distribution of data already stored in a data reservoir that has been discretized into bins representing the desired data distribution.

[0027] FIG. **2** illustrates an example system **200** that can be configured to implement dynamic optimization of a data set distribution. In embodiments, system **200** may include a data reservoir **230** that has been discretized into multiple data bins (**234**A, **234**B, . . . , **234**X) based on a desired overall statistical distribution of data in the reservoir **230**; and a data set optimizer **220** that automatically maintains a fresh, up-to-date data reservoir **230** with the desired distribution by receiving newly collected data and then determining whether to update the data reservoir **230** using the newly collected data.

[0028] In embodiments, the system **200** receives a data set optimization job **205** that includes input data **202** and configuration data **204**. In some embodiments, the input data set **202** may be a data stream, as previously described. In some embodiments, the configuration data **204** may include a description of the discretized data reservoir **230** (e.g., the configuration of the set of bins and, additionally and/or alternatively, a desired distribution of data across the set of bins). In some embodiments, the data set optimization job **205** also may include an input data evaluator **214** while, in some alternative embodiments, the input data evaluator **214** may be a component of system **200**. In embodiments, the input data evaluator **214** may be a supervised machine learning algorithm (e.g., a classifier). In some embodiments, evaluating an input data instance may include assigning the instance an evaluation value (e.g., a classification prediction confidence value as previously described with reference to FIG. **1**).

[0029] In some embodiments, each of the input data instances **212** from the input data set **202** is processed by the data set optimizer **220** using the input data evaluator **214**, and then the system determines whether the evaluated data instance **222** is to be offered to any of the data bins **234** in the data reservoir **230**. In some embodiments, the evaluated data instance **222** includes a prediction and/or prediction confidence value, and the determination is based at least in part on matching the prediction and/or prediction confidence value to attributes of the data that are respectively stored within each data bin **234**.

[0030] In some embodiments, each of the data bins **234** is respectively associated with a reservoir sampler **232** that maintains summary statistics of the distribution of data within the bin and determines whether to update the data bin **234** based in part on those summary statistics. For example, in some embodiments, the summary statistics may include a size capacity for the data bin **234** (i.e., the maximum number of data instances that can be stored in the data bin) since the set of data bins is selected to represent a discretized overall distribution of the data reservoir **230**. Additionally, each of the data bins **234** may be associated with a particular range of evaluation values. Thus, in some embodiments, a reservoir sampler **232** may determine that an evaluated data instance **222** is to be added to a data bin **234** if the evaluation value associated with the data instance is within the range of evaluation values associated with the bin and if the current bin size is below the bin size capacity. Additionally and/or alternatively, a reservoir sampler **232** may determine that adding an evaluated data instance **222** to a data bin **232** will replace a data instance that currently is stored in the data bin **232**.

[0031] FIG. **3** is a flow diagram of an example method **300** for optimizing a data set distribution. Specifically, the method **300** is described with respect to processing of a data set optimization job **205** by system **200**.

[0032] In embodiments, the system receives **305** a data set optimization job that may include a set of input data, configuration data describing a data set distribution that has been discretized into a set of data bins, and an input data evaluator. In some embodiments, the set of input data may be a data stream that is received from an online data processing system. In some alternative embodiments, the set of input data may be a data stream that is generated from a very large data store that contains more data than can be analyzed at once. In some embodiments, the input data evaluator may be a supervised machine learning algorithm (e.g., a classifier).

[0033] In embodiments, the system selects **310** an input data instance from the set of input data, and then determines **315** whether to offer the data instance to at least one of the set of data bins. In some embodiments, as described with reference to FIG. **2**, the system uses the received input data evaluator to evaluate the input data instance. In some embodiments, the input data evaluator may be a supervised machine learning algorithm (e.g., a classifier). In some embodiments, the evaluated data instance includes a prediction and/or prediction confidence value, and the determination **315** whether to offer the data instance to at least one of the set of data bins is based at least in part on matching the prediction and/or prediction confidence value to attributes of the data that are respectively stored within each data bin.

[0034] In an instance **320** in which the data instance is offered to a data bin, the system determines **325** whether to update the data bin using the data instance. In some embodiments, as described with reference to FIG. **2**, determining whether to update a data bin is implemented by a reservoir sampler respectively associated with that data bin. In some embodiments, the reservoir sampler maintains summary statistics of the distribution of data within the data bin and determines whether to update the data bin based in part on those summary statistics. In an instance in which the data bin is updated **325**, in some embodiments, the system may add the evaluated data instance to the data bin and/or replace a data instance currently stored in the data bin with the evaluated data instance.

[0035] The system processes each input data instance received in the data set optimization job. The process ends **340** in an instance in which the system determines **330** that all the input data instances have been processed. In an instance **330** in which all the input data instances have not been processed, the system selects **335** another input data instance to process from the set of input data.

[0036] In some embodiments in which the system is an entity resolution sampling system, the received set of input data are entity references (some of which may refer to the same entity, as previously described), and the input data evaluator is a machine learning algorithm that evaluates each possible pair of references as either being a match or not being a match. Thus, prior to evaluation, a received set of input data explodes into a large data space of possible reference pairs for the system to process (e.g., an input list of N=100,000 references explodes to N^2=10 billion possible pairs).

[0037] Searching through 10 billion pairs for less than 100,000 matching references may be computationally expensive. In some embodiments, selection of reference pairs to input to a predictive model for entity resolution may be optimized by first converting a list of references into a data stream of pairs of references. Thus, entity resolution sampling is converted from list processing to a stream sampling problem. There are a variety of different approaches to schedule the addition of pairs of references to a data stream. Table 1 illustrates, for clarity and without limitation, a pseudocode implementation of one example approach in which pairs of references are added to the data stream according to a random schedule.

TABLE 1

Exemplary random selection of input data instances for entity resolution sampling

```
while(true):
    reference1 = choose_randomly_from_input_list(input_list)
    reference2 = choose_randomly_from_input_list(input_list)
    if reference1 != reference2:
        emit( new Pair(reference1, reference2) )
```

[0038] In some embodiments, selecting a set of input data instances may be further optimized to increase search efficiency. For example, in some embodiments, selecting a set of input data instances may include partitioning the input references by a grouping function to create pre-defined groupings of references that are more likely to match, and then calculating the proportion of references that exist in each group (in entity resolution, this approach is called "blocking."). In an example embodiment, geospatial data may be grouped into latitude/longitude "squares." Inside each square, the geospatial data will be in close geographical proximity, and thus may have a higher likelihood of matching together. Other exemplary grouping fields for geographical data may include city and/or postal code.

[0039] An exemplary pseudocode implementation of optimized selection of input data for entity resolution sampling by using a grouping function is described in Table 2.

TABLE 2

Exemplary data stream preprocessing using a grouping function

```
while(true):
    group = choose_a_group_randomly_according_to_group_proportions(group_proportions)
    reference1 = choose_randomly_from_group(group)
    reference2 = choose_randomly_from_group(group)
    if reference1 != reference2:
        emit( new Pair(reference1, reference2) )
```

[0040] In some embodiments, an adaptive data analysis system that includes a predictive model may be configured to further include an active learning component to facilitate adaptation of the predictive model. In embodiments, a dynamic data set distribution system (e.g., system **200**) may complement an active learning component by facilitating selection of a labeling set of samples that will optimally improve the performance of the predictive model.

[0041] FIG. **4** illustrates an example adaptive online data analysis system **400** that is configured to include dynamic data set distribution optimization according to various embodiments of the invention. In embodiments, system **400** may comprise an input data analysis module **420** for creating an optimal feature representation (e.g., a feature vector **404**) of a received input data sample **402** selected from a data stream **401**; a predictive model **430** that has been generated using machine learning based on a set of training data **440**, and that is configured to generate a judgment **406** about the input data sample **402** in response to receiving a feature vector **404** representing the input data sample **402**; a data set optimizer **220** for evaluating the input data sample **402** and its associated judgment **406**; and a data reservoir **230** that includes a set of data bins **234** maintained by data set optimizer **220**. The data reservoir **230** thus is ensured to store fresh, up-to-date data that, in embodiments, may be selected from at least one of the bins **234** to update the training data **440**, thus enabling the model to be improved incrementally by being re-trained with a currently optimal set of examples.

[0042] In embodiments, the configuration of the reservoir data bins **234** may be used to ensure that the data reservoir stores up-to-date samples in a distribution that, if samples were selected from the bins and used to update the training data **440**, those samples potentially would create training data that would improve the performance of the predictive model **430**. In a first example, a set of bins **234** may be used to generate labeling sets that don't match the distribution of the general population. Each of the bins may be used to store data representing one of the possible labels, and a labeling set with equal frequencies of samples of each label may be generated even though at least one of the labels may be rare in the general population distribution. In a second example, each of the bins may represent one of the sources that have contributed to the data stream, and training data may be selected from the bins to match a distribution that represents a particular machine learning problem. Thus, if each of the data sources is a particular location (e.g., the US, Europe, and Asia), each of the bins stores data samples selected from one of the sources, and the desired training data **440** distribution should represent 10% US sources, 10% of a labeling sample may be selected from the data bin storing data selected from US sources.

[0043] In some embodiments, dynamic data set distribution optimization may be used as an anomaly detection system to support the quality assurance of data flowing through a real time data processing system. In these embodiments, for example, system **400** may be configured to include an anomaly scorer instead of a predictive model **430**, and the data bins **234** would be configured to represent a distribution of anomaly scores.

[0044] In some embodiments, dynamic data set distribution optimization may be used to assess the predictive model **430** calibration. In a perfectly calibrated model, the model predictions exactly match reality. Thus, for example, if the model is a probabilistic estimator, the model should predict 50% yes and 50% no for a probability 0.5; the model should predict 30% yes and 70% no for a probability 0.7; and the like. The empirical distribution within each data bin may be used to test the extent of the model **430** calibration. A small sample of data may be pulled out of a bin for analysis, and the distribution of predictions in the sample may be used for the test. For example, if the data in the bin represent a probability of 0.1, the data distribution may be tested to determine if 10% of the predictions in the sample match that probability.

[0045] In some embodiments, dynamic data set distribution optimization may be used to optimize feature modeling. The accuracy of the decisions within a bin may be tested (e.g., in some embodiments, a sample of the decisions within a bin may be sent to a crowd for verification), and the results may be used to adjust the feature modeling performed by the input data analysis module **420**.

[0046] FIG. **5** shows a schematic block diagram of circuitry **500**, some or all of which may be included in, for example, dynamic data set distribution optimization system **200**. As illustrated in FIG. **5**, in accordance with some example embodiments, circuitry **500** can include various means, such as processor **502**, memory **504**, communications module **506**, and/or input/output module **508**. As referred to herein, "module" includes hardware, software and/or firmware configured to perform one or more particular functions. In this regard, the means of circuitry **500** as described herein may be embodied as, for example, circuitry, hardware elements (e.g., a suitably programmed processor, combinational logic circuit, and/or the like), a computer program product comprising computer-readable program instructions stored on a non-transitory computer-readable medium (e.g., memory **504**) that is executable by a suitably configured processing device (e.g., processor **502**), or some combination thereof.

[0047] Processor **502** may, for example, be embodied as various means including one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more multi-core processors, one or more controllers, processing circuitry, one or more computers, various other processing elements including integrated circuits such as, for example, an ASIC (application specific integrated circuit) or FPGA (field programmable gate array), or some combination thereof. Accordingly, although illustrated in FIG. **5** as a single processor, in some embodiments processor **502** comprises a plurality of processors. The plurality of processors may be embodied on a single computing device or may be distributed across a plurality of computing devices collectively configured to function as circuitry **500**. The plurality of processors may be in operative communication with each other and may be collectively configured to perform one or more functionalities of circuitry **500** as described herein. In an example embodiment, processor **502** is configured to execute instructions stored in memory **504** or otherwise accessible to processor **502**. These instructions, when executed by processor **502**, may cause circuitry **500** to perform one or more of the functionalities of circuitry **500** as described herein.

[0048] Whether configured by hardware, firmware/software methods, or by a combination thereof, processor **502** may comprise an entity capable of performing operations according to embodiments of the present invention while configured accordingly. Thus, for example, when processor **502** is embodied as an ASIC, FPGA or the like, processor **502** may comprise specifically configured hardware for conducting one or more operations described herein. Alternatively, as another example, when processor **502** is embodied as an executor of instructions, such as may be stored in memory **504**, the instructions may specifically configure processor **502** to perform one or more algorithms and operations described herein, such as those discussed in connection with FIGS. **2-4**.

[0049] Memory **504** may comprise, for example, volatile memory, non-volatile memory, or some combination thereof. Although illustrated in FIG. **5** as a single memory, memory **504** may comprise a plurality of memory components. The plurality of memory components may be embodied on a single computing device or distributed across a plurality of computing devices. In various embodiments, memory **504** may comprise, for example, a hard disk, random access memory, cache memory, flash memory, a compact disc read only memory (CD-ROM), digital versatile disc read only memory (DVD-ROM), an optical disc, circuitry configured to store information, or some combination thereof. Memory **504** may be configured to store information, data (including analytics data), applications, instructions, or the like for enabling circuitry **500** to carry out various functions in accordance with example embodiments of the present invention. For example, in at least some embodiments, memory **504** is configured to buffer input data for processing by processor **502**. Additionally or alternatively, in at least some embodiments, memory **504** is configured to store program instructions for execution by processor **502**. Memory **504** may store information in the form of static and/or dynamic information. This stored information may be stored and/or used by circuitry **500** during the course of performing its functionalities.

[0050] Communications module **506** may be embodied as any device or means embodied in circuitry, hardware, a computer program product comprising computer readable program instructions stored on a computer readable medium (e.g., memory **504**) and executed by a processing device (e.g., processor **502**), or a combination thereof that is configured to receive and/or transmit data from/to another device, such as, for example, a second circuitry **500** and/or the like. In some embodiments, communications module **506** (like other components discussed herein) can be at least partially embodied as or otherwise controlled by processor **502**. In this regard, communications module **506** may be in communication with processor **502**, such as via a bus. Communications module **506** may include, for example, an antenna, a transmitter, a receiver, a transceiver, network interface card and/or supporting hardware and/or firmware/software for enabling communications with another computing device. Communications module **506** may be configured to receive and/or transmit any data that may be stored by memory **504** using any protocol that may be used for communications between computing devices. Communications module **506** may additionally or alternatively be in communication with the memory **504**, input/output module **508** and/or any other component of circuitry **500**, such as via a bus.

[0051] Input/output module **508** may be in communication with processor **502** to receive an indication of a user input and/or to provide an audible, visual, mechanical, or other output to a user. Some example visual outputs that may be provided to a user by circuitry **500** are discussed in connection with FIGS. **2-3**. As such, input/output module **508** may include support, for example, for a keyboard, a mouse, a joystick, a display, a touch screen display, a microphone, a speaker, a RFID reader, barcode reader, biometric scanner, and/or other input/output mechanisms. In embodiments wherein circuitry **500** is embodied as a server or database, aspects of input/output module **508** may be reduced as compared to embodiments where circuitry **500** is implemented as an end-user machine or other type of device designed for complex user interactions. In some embodiments (like other components discussed herein), input/output module **508** may even be eliminated from circuitry **500**. Alternatively, such as in embodiments wherein circuitry **500** is embodied as a server or database, at least some aspects of input/output module **508** may be embodied on an apparatus used by a user that is in communication with circuitry **500**, such as for example, pharmacy terminal **108**. Input/output module **508** may be in communication with the memory **504**, communications module **506**, and/or any other component(s), such as via a bus. Although more than one input/output module and/or other component can be included in circuitry **500**, only one is shown in FIG. **5** to avoid overcomplicating the drawing (like the other components discussed herein).

[0052] Data set optimizer module **510** may also or instead be included and configured to perform the functionality discussed herein related to the dynamic data quality assessment discussed above. In some embodiments, some or all of the functionality of dynamic data quality assessment may be performed by processor **502**. In this regard, the example processes and algorithms discussed herein can be performed by at least one processor **502** and/or data set optimizer module **510**. For example, non-transitory computer readable media can be configured to store firmware, one or more application programs, and/or other software, which include instructions and other computer-readable program code portions that can be executed to control each processor (e.g., processor **502** and/or data set optimizer module **510**) of the components of system **400** to implement various operations, including the examples shown above. As such, a series of computer-readable program code portions are embodied in one or more computer program products and can be used, with a computing device, server, and/or other programmable apparatus, to produce machine-implemented processes.

[0053] Any such computer program instructions and/or other type of code may be loaded onto a computer, processor or other programmable apparatus's circuitry to produce a machine, such that the computer, processor other programmable circuitry that execute the code on the machine create the means for implementing various functions, including those described herein.

[0054] It is also noted that all or some of the information presented by the example displays discussed herein can be based on data that is received, generated and/or maintained by one or more components of dynamic data quality assessment system **100**. In some embodiments, one or more external systems (such as a remote cloud computing and/or data storage system) may also be leveraged to provide at least some of the functionality discussed herein.

[0055] As described above in this disclosure, aspects of embodiments of the present invention may be configured as methods, mobile devices, backend network devices, and the like. Accordingly, embodiments may comprise various means including entirely of hardware or any combination of software and hardware. Furthermore, embodiments may take the form of a computer program product on at least one non-transitory computer-readable storage medium having computer-readable program instructions (e.g., computer software) embodied in the storage medium. Any suitable computer-readable storage medium may be utilized including non-transitory hard disks, CD-ROMs, flash memory, optical storage devices, or magnetic storage devices.

[0056] Embodiments of the present invention have been described above with reference to block diagrams and flowchart illustrations of methods, apparatuses, systems and computer program products. It will be understood that each block of the circuit diagrams and process flow diagrams, and combinations of blocks in the circuit diagrams and process flowcharts, respectively, can be implemented by various means including computer program instructions. These computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus, such as processor **502** and/or data set optimizer module **510** discussed above with reference to FIG. **5**, to produce a machine, such that the computer program product includes the instructions which execute on the computer or other programmable data processing apparatus create a means for implementing the functions specified in the flowchart block or blocks.

[0057] These computer program instructions may also be stored in a computer-readable storage device (e.g., memory **504**) that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable storage device produce an article of manufacture including computer-readable instructions for implementing the function discussed herein. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide steps for implementing the functions discussed herein.

[0058] Accordingly, blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program instruction means for performing the specified functions. It will also be understood that each block of the circuit diagrams and process flowcharts, and combinations of blocks in the circuit diagrams and process flowcharts, can be implemented by special purpose hardware-based computer systems that perform the specified functions or steps, or combinations of special purpose hardware and computer instructions

[0059] Many modifications and other embodiments of the inventions set forth herein will come to mind to one skilled in the art to which these inventions pertain having the benefit of the teachings presented in the foregoing descriptions and the associated drawings. Therefore, it is to be understood that the inventions are not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

1-42. (canceled)

43. A system, comprising one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one or more computers to:

receive a data set optimization job that comprises a set of input data associated with an input range and distribution configuration data that describes a discretization of a data set distribution of the set of input data into a plurality of data bins, wherein the discretization defines a range of evaluation values for a data bin from the plurality of data bins, and wherein the discretization further defines a label for the data bin;

generate an evaluation determination that indicates whether to provide a portion of the input data to the data bin;

in response to a determination that the evaluation determination satisfies a defined criterion, generate an updated determination for the data bin based on the range of evaluation values associated with the data bin and an instance evaluation of the portion of the input data, wherein the updated determination provides an indication to associate the portion of the input data instance with the data bin; and

update the label for the data bin based on the updated determination for the data bin.

44. The system of claim **43**, wherein the discretization further defines a prediction confidence score for the data bin to store the portion of the input data.

45. The system of claim **43**, wherein the discretization further defines a size capacity for the data bin.

46. The system of claim **43**, wherein the label for the bin identifies another portion of the input data that is associated with the data bin.

47. The system of claim **43**, wherein the one or more storage devices store instructions that are operable, when executed by the one or more computers, to further cause the one or more computers to:

determine whether an evaluation value of the portion of the input data is within the range of evaluation values associated with the data bin.

48. The system of claim **43**, wherein the evaluation determination provides an indication as to whether a bin size capacity for the data bin is satisfied.

49. The system of claim **43**, wherein the data set optimization job is associated with an input data evaluator.

50. The system of claim **43**, wherein the data set optimization job is associated with a predictive model, and wherein the one or more storage devices store instructions that are operable, when executed by the one or more computers, to further cause the one or more computers to:

generate a model prediction for the portion of the input data based on the predictive model.

51. The system of claim **50**, wherein the one or more storage devices store instructions that are operable, when executed by the one or more computers, to further cause the one or more computers to:

determine whether the model prediction for the portion of the input data is within the range of evaluation values associated with the data bin.

52. The system of claim **43**, wherein the instance evaluation for the first input data instance is an anomaly score generated based on an anomaly scorer.

53. The system of claim **43**, wherein the set of input data is received from a data stream.

54. The system of claim **43**, wherein the set of input data is received from a data stream generated from online data.

55. A computer-implemented method, comprising:

receiving a data set optimization job that comprises a set of input data associated with an input range and distribution configuration data that describes a discretization of a data set distribution of the set of input data into a plurality of data bins, wherein the discretization defines a range of evaluation values for a data bin from the plurality of data bins, and wherein the discretization further defines a label for the data bin;

generating an evaluation determination that indicates whether to provide a portion of the input data to the data bin;

in response determining that the evaluation determination satisfies a defined criterion, generating an updated determination for the data bin based on the range of evaluation values associated with the data bin and an instance evaluation of the portion of the input data, wherein the updated determination provides an indication to associate the portion of the input data instance with the data bin; and

updating the label for the data bin based on the updated determination for the data bin.

56. The computer-implemented method of claim **55**, further comprising:

defining a prediction confidence score for the data bin to store the portion of the input data.

57. The computer-implemented method of claim **55**, further comprising:

determining whether an evaluation value of the portion of the input data is within the range of evaluation values associated with the data bin.

58. The computer-implemented method of claim **55**, further comprising:

generating a model prediction for the portion of the input data based on a predictive model associated with the data set optimization job.

59. The computer-implemented method of claim **58**, further comprising:

determining whether the model prediction for the portion of the input data is within the range of evaluation values associated with the data bin.

60. A computer program product, stored on a computer readable medium, comprising instructions that when executed by one or more computers cause the one or more computers to:

receive a data set optimization job that comprises a set of input data associated with an input range and distribution configuration data that describes a discretization of a data set distribution of the set of input data into a plurality of data bins, wherein the discretization defines a range of evaluation values for a data bin from the plurality of data bins, and wherein the discretization further defines a label for the data bin;

generate an evaluation determination that indicates whether to provide a portion of the input data to the data bin;

in response to a determination that the evaluation determination satisfies a defined criterion, generate an updated determination for the data bin based on the range of evaluation values associated with the data bin and an instance evaluation of the portion of the input data, wherein the updated determination provides an indication to associate the portion of the input data instance with the data bin; and update the label for the data bin based on the updated determination for the data bin.

61. The computer program product of claim **60**, wherein the discretization further defines a prediction confidence score for the data bin to store the portion of the input data.

62. The computer program product of claim **60**, wherein the discretization further defines a size capacity for the data bin.

* * * * *