US 20160085811A1

(19) **United States**

(12) **Patent Application Publication**

Deolalikar et al.

(10) **Pub. No.: US 2016/0085811 A1**

(43) **Pub. Date: Mar. 24, 2016**

(54) **GENERATING A FEATURE SET**

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.**, Houston, TX (US)

(72) Inventors: **Vinay Deolalikar**, Cupertino, CA (US); **Hernan Laffitte**, Mountain View, CA (US)

(21) Appl. No.: **14/780,707**

(22) PCT Filed: **Mar. 28, 2013**

(86) PCT No.: **PCT/US13/34400**

§ 371 (c)(1),
(2) Date: **Sep. 28, 2015**

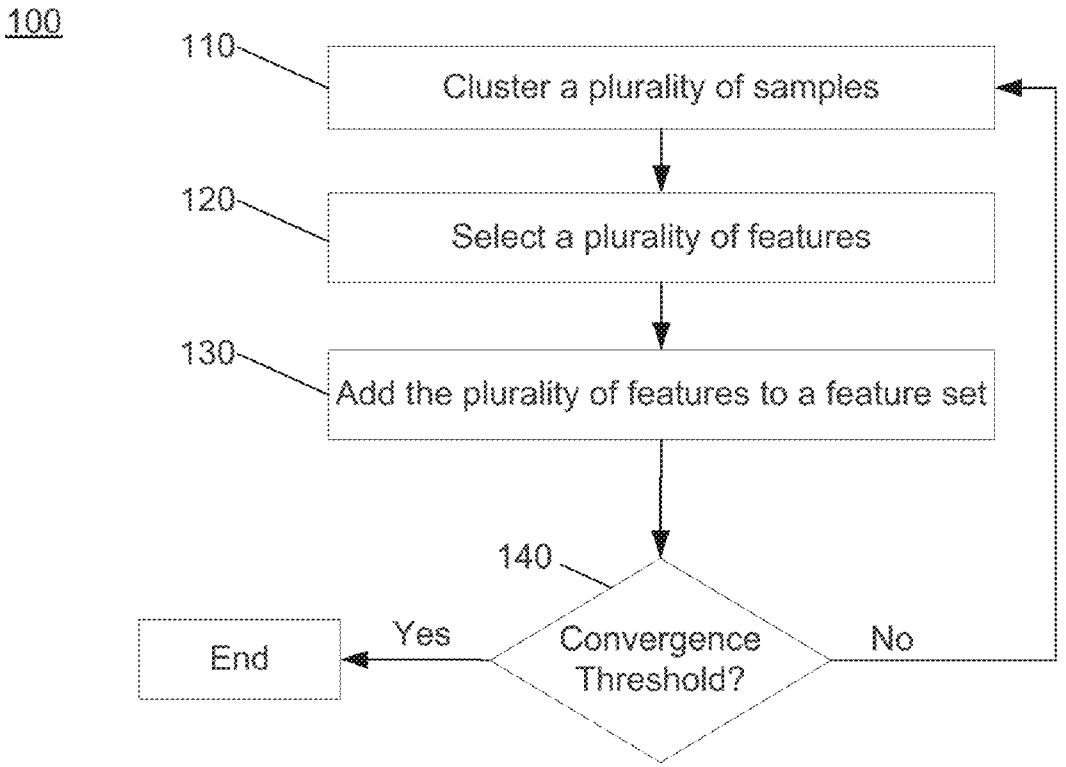
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 17/30522*** (2013.01); ***G06F 17/30539*** (2013.01)

(57) **ABSTRACT**

A technique to generate a feature set. A plurality of samples from a data set can be clustered. Features can be selected based on the clusters. The features can be added to the feature set. Additional samples can be clustered and features selected and added to the feature set until a convergence threshold is reached.



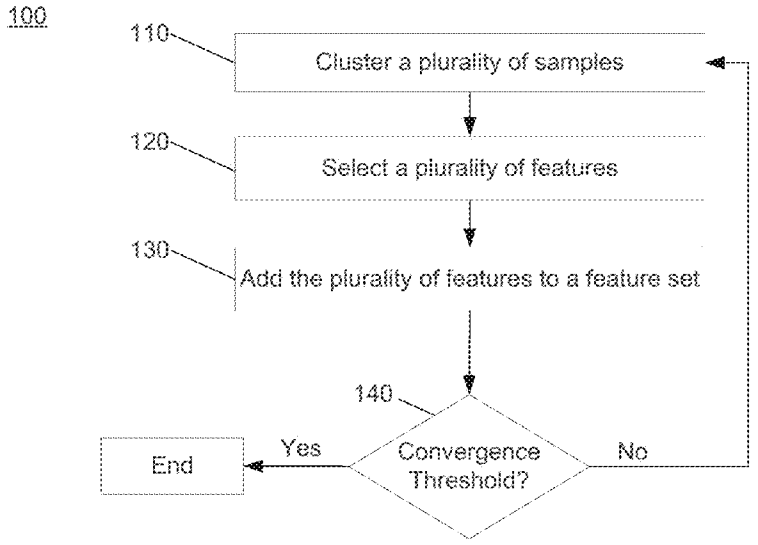

100
110
Cluster a plurality of samples
120
Select a plurality of features
130
Add the plurality of features to a feature set
140
Convergence Threshold?
Yes
End
No

FIG. 1

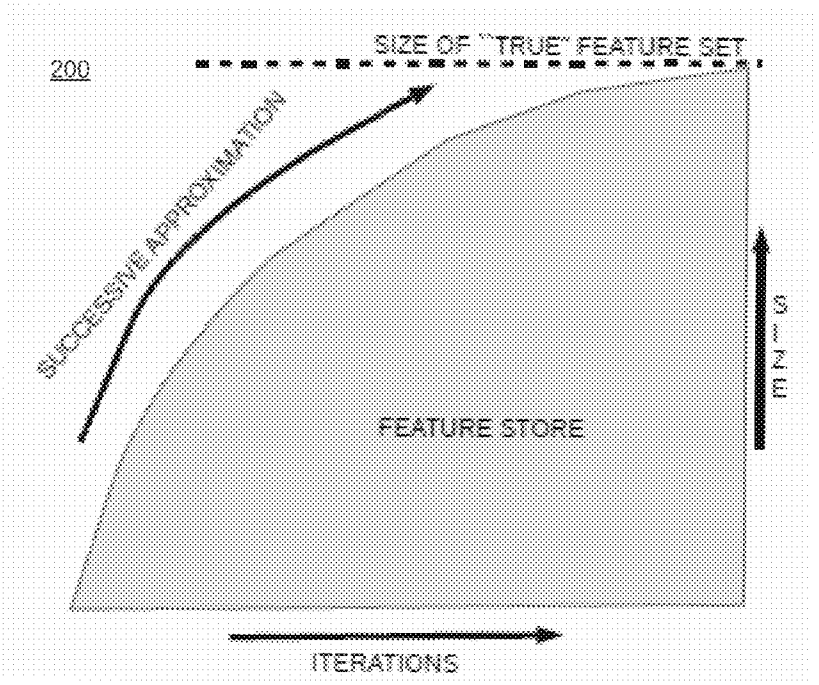

200
SIZE OF "TRUE" FEATURE SET
SUCCESSIVE APPROXIMATION
FEATURE STORE
SIZE
ITERATIONS

FIG. 2

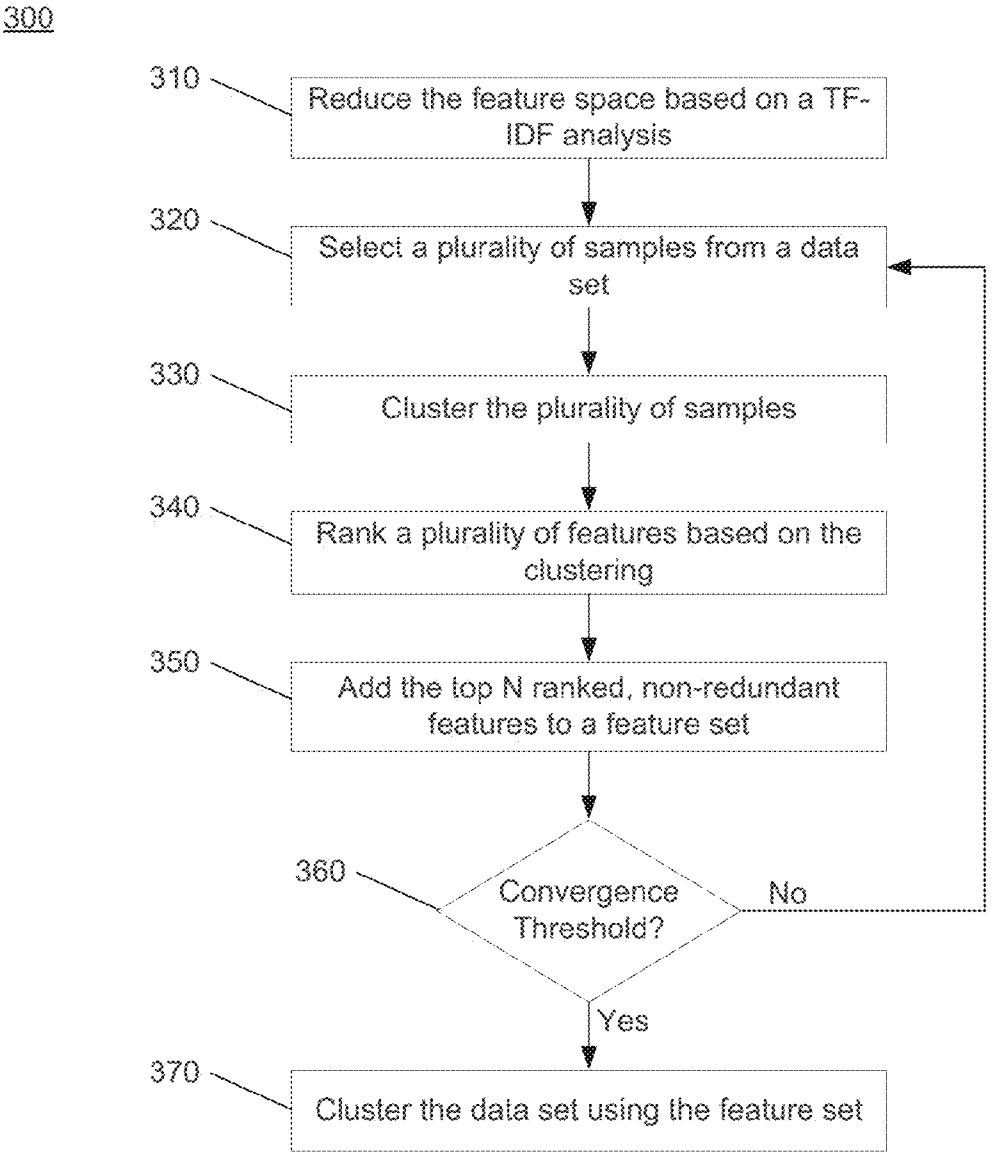

300
310
Reduce the feature space based on a TF-IDF analysis
320
Select a plurality of samples from a data set
330
Cluster the plurality of samples
340
Rank a plurality of features based on the clustering
350
Add the top N ranked, non-redundant features to a feature set
360
Convergence Threshold?
No
Yes
370
Cluster the data set using the feature set

FIG. 3

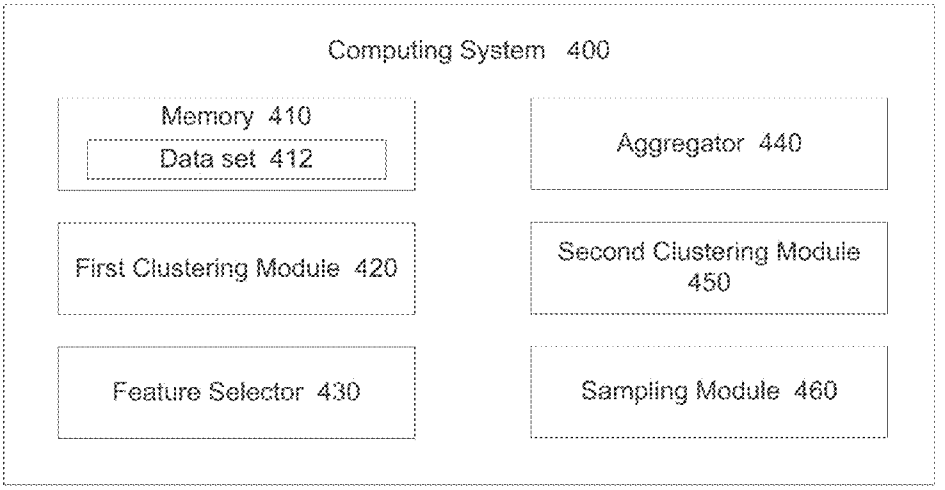

Computing System 400
Memory 410
Data set 412
Aggregator 440
First Clustering Module 420
Second Clustering Module 450
Feature Selector 430
Sampling Module 460

FIG. 4

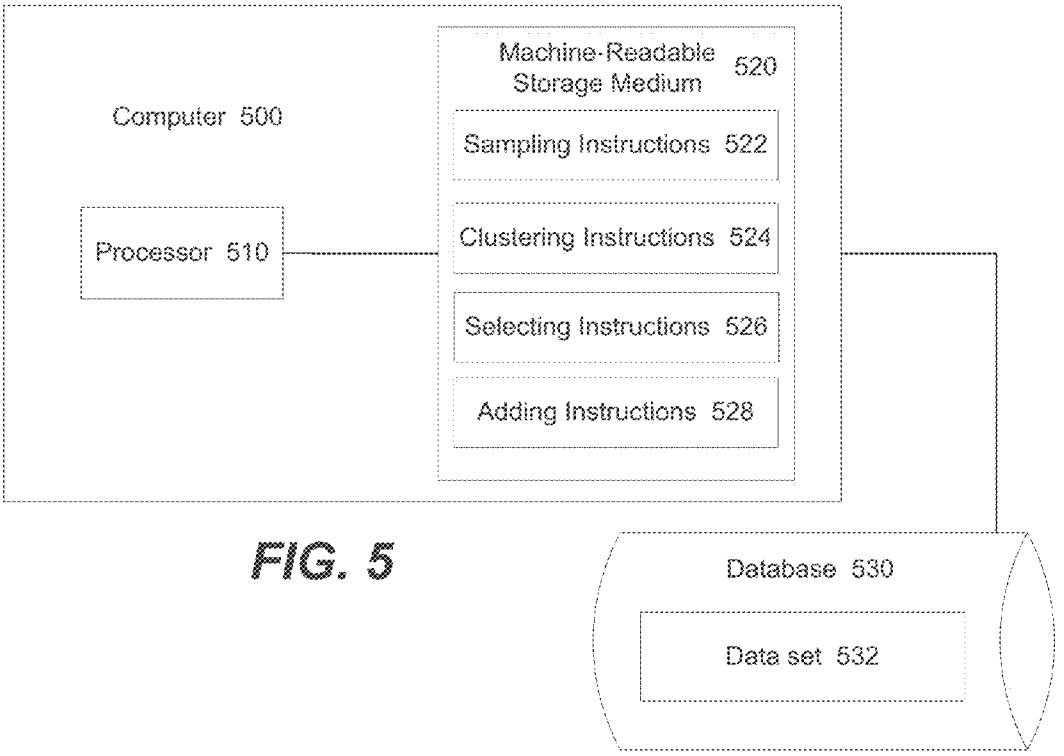

Computer 500
Machine-Readable Storage Medium 520
Sampling Instructions 522
Processor 510
Clustering Instructions 524
Selecting Instructions 526
Adding Instructions 528
Database 530
Data set 532

FIG. 5

GENERATING A FEATURE SET

BACKGROUND

[0001] In data mining, clustering can be used to group data objects based on similarities between the objects. Clustering can be useful because it can provide different perspectives on large sets of data. For example, in an enterprise setting, an enterprise may have a large corpus of documents. Clustering can be applied to the corpus to group the documents into multiple clusters. These clusters can reveal similarities between the clustered documents, enabling the enterprise to make more efficient use of its data and gain insights that may otherwise be difficult to draw.

BRIEF DESCRIPTION OF DRAWINGS

[0002] The following detailed description refers to the drawings, wherein:

[0003] FIG. **1** illustrates a method of generating a feature set, according to an example.

[0004] FIG. **2** depicts a graph illustrating how a feature set can be approximated using the disclosed techniques, according to an example.

[0005] FIG. **3** illustrates a method of generating a feature set for clustering a data set, according to an example.

[0006] FIG. **4** illustrates a system for generating a feature set, according to an example.

[0007] FIG. **5** illustrates a computer-readable medium for generating a feature set, according to an example.

DETAILED DESCRIPTION

[0008] Clustering a data set, such as a corpus of documents, can present various challenges. For example, a clustering operation may take a long time to execute if the corpus of documents is large. Typically, each document in the corpus may be represented by a feature vector. The clustering operation may cluster the documents based on the feature vectors. If the corpus is large, the total size of the feature vectors may also be large. However, due to limitations in the size of primary memory (e.g., RAM), a large set of feature vectors representing the document corpus may not all simultaneously fit into the primary memory of a computer system executing the clustering operation. As a result, the feature vectors may have to be read into primary memory from secondary memory (e.g., hard disk drive) during the clustering operation, which can cause the clustering operation to take longer to complete. According to techniques disclosed herein, a feature set may be generated for use in clustering a data set. The generated feature set may be smaller than the feature space of the data set, thus reducing the amount of memory used to perform the clustering operation.

[0009] In an example, a plurality of samples may be selected (e.g., randomly selected) from the data set and clustered using a clustering algorithm. A plurality of features may be selected based on the clustering. For instance, the features may be selected based on information gain with respect to cluster inclusion. The selected features may be added to the feature set. An additional plurality of samples may be selected from the data set and the above processing may be repeated to add additional features to the feature set. This may occur for various iterations until a convergence threshold is reached. For instance, the convergence threshold may relate to a size by which the feature set is growing, a size of the feature set, or a predetermined maximum number of iterations. The generated feature set may then be used to cluster the entire data set using the same clustering algorithm.

[0010] As a result, primary memory usage may be reduced due to the smaller number of features, enabling a clustering operation to be performed more efficiently over the entire data set. Furthermore, this smaller feature set may be obtained without first clustering the entire data set. In addition, because the feature set is generated using clusters generated by the same clustering algorithm, the feature set may be tailored specifically for that clustering algorithm, which may result in improved clustering. Additional examples, advantages, features, modifications and the like are described below with reference to the drawings.

[0011] FIG. **1** illustrates a method of generating a feature set, according to an example. Method **100** may be performed by a computing device, system, or computer, such as computing system **400** or computer **500**. Computer-readable instructions for implementing method **100** may be stored on a computer readable storage medium. These instructions as stored on the medium are referred to herein as "modules" and may be executed by a computer.

[0012] Method **100** may begin at **110**, where a plurality of samples from a data set may be clustered. The data set may include documents, images, or the like. For illustration purposes, an example will be described herein in which the data set comprises a corpus of documents. For instance, the corpus of documents may be a large corpus of documents stored in a database of an enterprise.

[0013] Each member of the data set may be represented by a feature vector. The vector space associated with the set of feature vectors representing all members of the data set is referred to herein as the "original feature space". Referring to the corpus of documents example, the feature vectors associated with the corpus may be expressed as a document-term matrix. Of course, other features of the documents may be included in the feature vectors as well, such as type of document, title, total number of words, etc.

[0014] As used herein, a "sample" is a member of the data set that has been sampled by a sampling technique. Accordingly, a plurality of samples of the document corpus would include a plurality of documents sampled using the sampling technique. Any of various sampling techniques may be used. In an example, independent, random sampling may be used to select a plurality of samples. While other sampling techniques may be used, such as user-directed sampling, independent, random sampling has the benefit of removing bias from the sampling process and potentially resulting in a more accurate representation of the distribution of the data set.

[0015] The number of samples included in the plurality of samples can be a parameter that is preset or specified by a user. The sampling size may be influenced by various factors. For example, a particular percentage (e.g., 1%, 5%) of the data set may be sampled using the sampling technique. Alternatively, a fixed number of samples may be sampled from the data set. For fast processing, the sampling size can be small enough that all of the feature vectors for the plurality of samples can fit into primary memory. Unexpected results were obtained with respect to sampling size during experimentation, as will be described later.

[0016] The plurality of sampled documents may be clustered using a clustering algorithm to yield a plurality of clusters. The number of clusters to be created can be a parameter that is preset or specified by a user. Any of various clustering algorithms may be used (e.g., hierarchical clustering, cen-

troid-based clustering, distribution-based clustering, and density-based clustering). The same clustering algorithm that will be used to cluster the entire document corpus may be used to cluster the plurality of sampled documents. Using the same clustering algorithm to generate the feature set as will be used to ultimately cluster the entire data set may be beneficial because the generated feature set will then be tailored to the chosen clustering algorithm.

[0017] At **120**, a plurality of features may be selected based on the plurality of clusters. For example, features in the original feature space may be evaluated based on the clustering generated in **110** to determine which features should be included in the feature set. The evaluation may be made based on various criteria. For instance, the features may be evaluated based on information gain with respect to cluster inclusion. This technique discerns what features are relevant to determining whether a document should be a member of a particular cluster. The features may be ranked based on the evaluation criteria and the top N features from each cluster may be selected for inclusion in the feature set. N may be a parameter that is preset or specified by a user.

[0018] At **130**, the plurality of features may be added to the feature set. Redundant features (e.g., features that have already been added to the feature set) may be ignored such that selected features are added only if they are not already present in the feature set.

[0019] At **140**, it may be determined whether a convergence threshold has been reached. If the convergence threshold has been reached ("Yes" at **140**), method **100** may be terminated. If the convergence threshold has not been reached ("No" at **140**), method **100** may continue to **110** to cluster another plurality of samples in order to add more features to the feature set. Method **100** may iterated through **110-140** multiple times until the convergence threshold is satisfied.

[0020] The convergence threshold may be any of various thresholds. The purpose of the convergence threshold is to indicate when method **100** should end, or alternatively, when the feature set has reached a satisfactory point for use. For example, the convergence threshold may be a point at which the feature set being generated exhibits a "falling profile". A falling profile as used herein indicates that the percentage by which the feature set grows after addition of the plurality of features according to, e.g., **130**, falls below a certain value, such as 2%. Other percentages may be used. In another example, the convergence threshold may be a particular number. For example, the convergence threshold may be met if the number of features added to the feature set during an iteration is less than the particular number. Alternatively, the convergence threshold may be met if the number of iterations of method **100** is greater than the particular number. In some examples, the convergence threshold may be a user tunable parameter.

[0021] Briefly turning to FIG. **2**, graph **200** illustrates a falling profile for a feature set (referred to in the drawing as "FEATURE STORE"). A "true" feature set is posited, which is an ideal feature set for the data set. Method **100** (and other methods and variations herein) can be used to successively approximate this true feature set. As illustrated by the graph **200**, as the number of iterations of the method increases, the size of the feature set increases and approaches the true feature set. The growth of the feature set, however, is not linear. Rather, the feature set grows quickly at the start of the method but slows as the number of iterations increases. This is due, for example, to the fact that redundant selected features are not added to the feature set. This trend in the growth of the feature set is what is intended by the term "falling profile".

[0022] Returning to FIG. **1**, method **100** can thus be used to quickly generate a feature set that approximates an ideal feature set for a data set. The feature space of this feature set is smaller than the original feature space of the data set, thus enabling more of the feature vectors of the data set to fit into primary memory for faster clustering of the entire data set. Indeed, the parameters noted above can be modified to ensure that a generated feature set for the data set will be sufficiently small so that all of the feature vectors of the data set may fit into primary memory of the computer system being used.

[0023] The disclosed techniques can be more effective for generation of the feature set than simply taking a single random sampling of the data set. One reason is because a single random sampling likely will miss certain groupings of members of the data set. As a result, the generated feature set may not include features relevant to these groupings. Additionally, the disclosed techniques do not require processing of the entire data set for generation of the feature set. This enables the disclosed techniques to be incorporated into data analysis tools that will be deployed in environments where the data sets are previously unknown or are constantly changing. A further benefit is that the number of features in the feature set can be automatically selected by the disclosed techniques. This saves the user from having to guess what the ideal number of features would be for a given data set, which is a complex task involving various constraints, tradeoffs, and the like, not suitable for a user.

[0024] During experimentation, the inventors obtained some unexpected results. The inventors determined that a lower sampling size can enable method **100** to reach the convergence threshold more quickly. For example, on average, a sampling size of 1% resulted in quicker convergence than a sampling size of 25%. This has the added benefit that the lower the sampling size, the more likely the feature vectors of the sampled documents will fit into primary memory. Additionally, the inventors determined that the quality of the clustering produced by method **100** when the sampling size is 5% was often better than when the sampling size was higher. In short, the inventors determined that aggregating features selected from multiple samples can improve the quality of the generated feature set and simultaneously reduce the processing time and memory space needed.

[0025] The inventors also determined that it can be better to select more features during each iteration. For example, by setting N=20 rather than N=10, the inventors determined that the feature set may be bigger at the time of convergence and the quality of the clustering of the data set using the feature set may be improved.

[0026] FIG. **3** illustrates a method **300** for generating a feature set for clustering, according to an example. Method **300** shows variations that may be used to modify method **100**. At the same time, the description of method **100** applies to method **300**. Method **300** may be performed by a computing device, system, or computer, such as computing system **500** or computer **600**. Computer-readable instructions for implementing method **300** may be stored on a computer readable storage medium. These instructions as stored on the medium are referred to herein as "modules" and may be executed by a computer.

[0027] Method **300** may begin at **310**, where the feature space may be reduced based on a term frequency-inverse document frequency (TF-IDF) analysis. The feature space

reduced by the TF-IDF analysis may be the original feature space associated with the data set and its feature vectors. The new feature space may be referred to as a reduced feature space. TF-IDF analysis is a statistical technique that can be used to reduce the dimensionality of a feature space. By applying a TF-IDF analysis to the original feature space, features that are likely to be unhelpful for clustering purposes can be removed from the feature space so as to reduce the size of the feature vectors processed by the rest of method **300**. Accordingly, the data set may be clustered based on the reduced feature space, and features selected therefrom.

[0028] At **320**, a plurality of samples may be selected from the data set. At **330**, the plurality of samples may be clustered. At **340**, a plurality of features may be ranked based on the clustering. At **350**, the top N ranked, non-redundant features may be added to the feature set. At **360**, it may be determined whether a convergence threshold is met. If the convergence threshold is not met ("No" at **360**), method **300** may continue to **320**. If the convergence threshold is met ("Yes" at **360**), method **300** may continue to **370**, where the data set may be clustered using the feature set.

[0029] FIG. **4** illustrates a system for generating a feature set, according to an example. Computing system **400** may include and/or be implemented by one or more computers. For example, the computers may be server computers, workstation computers, desktop computers, or the like. The computers may include one or more controllers and one or more machine-readable storage media.

[0030] A controller may include a processor and a memory for implementing machine readable instructions. The processor may include at least one central processing unit (CPU), at least one semiconductor-based microprocessor, at least one digital signal processor (DSP) such as a digital image processing unit, other hardware devices or processing elements suitable to retrieve and execute instructions stored in memory, or combinations thereof. The processor can include single or multiple cores on a chip, multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof. The processor may fetch, decode, and execute instructions from memory to perform various functions. As an alternative or in addition to retrieving and executing instructions, the processor may include at least one integrated circuit (IC), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing various tasks or functions.

[0031] The controller may include memory, such as a machine-readable storage medium. The machine-readable storage medium may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, the machine-readable storage medium may comprise, for example, various Random Access Memory (RAM), Read Only Memory (ROM), flash memory, and combinations thereof. For example, the machine-readable medium may include a Non-Volatile Random Access Memory (NVRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a NAND flash memory, and the like. Further, the machine-readable storage medium can be computer-readable and non-transitory. Additionally, computing system **400** may include one or more machine-readable storage media separate from the one or more controllers, such as memory **410**.

[0032] Computing system **400** may include memory **410**, first clustering module **420**, feature selector **430**, and aggregator **440**, second clustering module **450**, and sampling module **460**. Each of these components may be implemented by a single computer or multiple computers. The components may include software, one or more machine-readable media for storing the software, and one or more processors for executing the software. Software may be a computer program comprising machine-executable instructions.

[0033] In addition, users of computing system **400** may interact with computing system **400** through one or more other computers, which may or may not be considered part of computing system **400**. As an example, a user may interact with system **400** via a computer application residing on system **400** or on another computer, such as a desktop computer, workstation computer, tablet computer, or the like. The computer application can include a user interface.

[0034] Computer system **400** may perform methods **100**, **300**, and variations thereof, and components **420**-**460** may be configured to perform various portions of methods **100**, **300**, and variations thereof. Additionally, the functionality implemented by components **420**-**460** may be part of a larger software platform, system, application, or the like. For example, these components may be part of a data analysis system.

[0035] In an example, memory **410** may be configured to store a data set **412**. Sampling module **460** may be configured to generate independent, random samples of the data set for use by first clustering module **420**. First clustering module **420** may be configured to cluster various pluralities of samples generated by sampling module **460** to generate a plurality of clusters for each plurality of samples. Feature selector **430** may be configured to select one or more features based on the plurality of clusters. Aggregator **440** may be configured to aggregate features selected based on multiple clusterings of multiple pluralities of samples from the data set until a convergence threshold is reached. Aggregator **440** may work in conjunction with first clustering module **420** and feature selector **430** to aggregate selected features over multiple iterations. Second clustering module **450** may be configured to cluster the entire data set based on the aggregated features. The first clustering module **420** and the second clustering module **450** may be configured to use the same clustering algorithm.

[0036] FIG. **5** illustrates a computer-readable medium for generating a feature set, according to an example. Computer **500** may be any of a variety of computing devices or systems, such as described with respect to computing system **500**.

[0037] Computer **500** may have access to database **530**. Database **530** may include one or more computers, and may include one or more controllers and machine-readable storage mediums, as described herein. Computer **500** may be connected to database **530** via a network. The network may be any type of communications network, including, but not limited to, wire-based networks (e.g., cable), wireless networks (e.g., cellular, satellite), cellular telecommunications network(s), and IP-based telecommunications network(s) (e.g., Voice over Internet Protocol networks). The network may also include traditional landline or a public switched telephone network (PSTN), or combinations of the foregoing.

[0038] Processor **510** may be at least one central processing unit (CPU), at least one semiconductor-based microprocessor, other hardware devices or processing elements suitable to retrieve and execute instructions stored in machine-readable storage medium **520**, or combinations thereof. Processor **510** can include single or multiple cores on a chip, multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof. Processor **510** may fetch, decode,

and execute instructions **522-528** among others, to implement various processing. As an alternative or in addition to retrieving and executing instructions, processor **510** may include at least one integrated circuit (IC), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing the functionality of instructions **522-528**. Accordingly, processor **510** may be implemented across multiple processing units and instructions **522-528** may be implemented by different processing units in different areas of computer **500**.

[0039] Machine-readable storage medium **520** may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, the machine-readable storage medium may comprise, for example, various Random Access Memory (RAM), Read Only Memory (ROM), flash memory, and combinations thereof. For example, the machine-readable medium may include a Non-Volatile Random Access Memory (NVRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a NAND flash memory, and the like. Further, the machine-readable storage medium **520** can be computer-readable and non-transitory. Machine-readable storage medium **520** may be encoded with a series of executable instructions for managing processing elements.

[0040] The instructions **522**, **524** when executed by processor **510** (e.g., via one processing element or multiple processing elements of the processor) can cause processor **510** to perform processes, for example, methods **100**, **300**, and variations thereof. Furthermore, computer **500** may be similar to computing system **500** and may have similar functionality and be used in similar ways, as described above.

[0041] For example, sampling instructions **522** may cause processor **510** to select a plurality of samples from the data set **532** using a sampling technique. In an example, the sampling technique may be a random sampling algorithm. Clustering instructions **524** may cause processor **510** to cluster the plurality of samples into a plurality of clusters. Selecting instructions **526** may cause processor **510** to select a plurality of features based on the plurality of clusters. Adding instructions **528** may cause processor **510** to add the plurality of features to a feature set for clustering. Instructions **522-528** may be executed for multiple iterations until a convergence threshold is met. In an example, the convergence threshold may be met if the number of features added to the feature set for clustering in a given iteration is below a threshold. In an example, the feature set for clustering may be used to cluster the entire data set after the convergence threshold is met.

[0042] In the foregoing description, numerous details are set forth to provide an understanding of the subject matter disclosed herein. However, implementations may be practiced without some or all of these details. Other implementations may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

What is claimed is:

1. A method for generating a feature set, comprising:

(a) clustering a first plurality of samples from a data set into a first plurality of clusters using a clustering algorithm;

(b) selecting a first plurality of features based on the first plurality of clusters;

(c) adding the first plurality of features to a feature set;

(d) clustering an additional plurality of samples from the data set;

(e) selecting additional features based on the resulting clusters from (d);

(f) adding the additional features to the feature set; and

(g) iterating through (d)-(f) until a convergence threshold is reached.

2. The method of claim **1**, further comprising clustering the data set using the clustering algorithm and the feature set.

3. The method of claim **1**, further comprising:

before performing (a), generating a reduced feature space by performing a TF-IDF analysis of the data set to reduce the dimensionality of an original feature space associated with the data set, the first plurality of features and additional features being selected from the reduced feature space.

4. The method of claim **1**, wherein the convergence threshold is met if the feature set exhibits a falling profile.

5. The method of claim **1**, wherein the convergence threshold is met if the number of features added to the feature set during an iteration is less than a threshold.

6. The method of claim **1**, wherein the convergence threshold is met if the number of iterations is greater than a threshold.

7. The method of claim **1**, wherein the first and additional pluralities of samples are independent, random samples from the data set.

8. The method of claim **1**, wherein features are selected by:

ranking features in each cluster by information gain with respect to cluster inclusion; and

identifying the top N ranked features from each cluster.

9. The method of claim **1**, wherein features are added to the feature set only if they are not already present in the feature set.

10. A system, comprising:

a first clustering module to generate a plurality of clusters of a plurality of samples from a data set;

a feature selector to select one or more features based on the plurality of clusters;

an aggregator to aggregate features selected based on multiple clusterings of multiple pluralities of samples from the data set until a convergence threshold is reached; and

a second clustering module to cluster the entire data set based on the aggregated features.

11. The system of claim **10**, further comprising a sampling module to generate independent, random samples of the data set for use by the first clustering module.

12. The system of claim **10**, wherein the first clustering module and the second clustering module are configured to use the same clustering algorithm.

13. The system of claim **10**, wherein the aggregator is configured to aggregate features in conjunction with the first clustering module and feature selector.

14. A non-transitory computer readable storage medium storing instructions that, when executed by a processor, cause a computer to:

until a convergence threshold is met:

select a plurality of samples from a data set using a sampling technique;

cluster the plurality of samples into a plurality of clusters;

select a plurality of features based on the plurality of clusters; and

add the plurality of features to a feature set for clustering.

15. The storage medium of claim **14**, wherein the convergence threshold is met if the number of features added to the feature set for clustering in a given iteration is below a threshold.

16. The storage medium of claim **14**, wherein the sampling technique is a random sampling algorithm.

17. The storage medium of claim **14**, further storing instructions that cause a computer to:

use the feature set for clustering to cluster the entire data set after the convergence threshold is met.

* * * * *