(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 August 2016 (04.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/122561 A1

(51) International Patent Classification:
G06F 17/00 (2006.01)

(21) International Application Number:
PCT/US2015/013678

(22) International Filing Date:
30 January 2015 (30.01.2015)

(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: **MARWAH, Manish**; 1501 Page Mill Rd., Palo Alto, California 94304 (US). **CHAKRABARTI, Aniket**; 1554 Worthington Street, Columbus, Ohio 43201 (US). **ARLITT, Martin**; 39 Arbour Vista Close NW, Calgary, Alberta T3G 5P4 (CA).

(74) Agents: **KIRCHEV, Ivan T.** et al.; 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: SYNTHESIZING A GRAPH

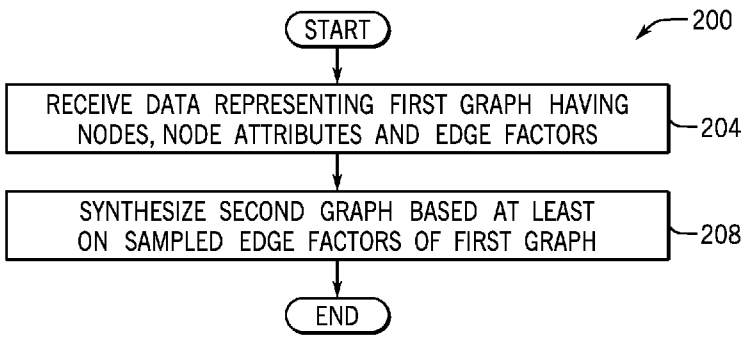


FIG. 2

(57) Abstract: A technique including receiving data representing a first graph having nodes, node attributes and edge factors. The technique includes synthesizing a second graph based at least in part on sampled edge factors of the first graph.

WO 2016/122561 A1

SYNTHESIZING A GRAPH

Background

[0001] For purposes of analyzing relatively large data sets (often called "big data"), it may be beneficial to represent the data in the form of a graph. For example, the decreasing cost of sensors has led to their deployment in large numbers. The data collected from sensors may be processed and analyzed for such purposes as monitoring and managing infrastructure and resources.

Brief Description of the Drawings

[0002] Fig 1 is a schematic diagram of a sensor-based system according to an example implementation.

[0003] Figs. 2 and 3 are flow diagrams depicting techniques to synthesize a graph according to example implementations.

[0004] Fig. 4 is a flow diagram depicting a technique to determine node attributes for a synthesized graph according to an example implementation.

[0005] Fig. 5 is a schematic diagram of a physical machine according to an example implementation.

Detailed Description

[0006] Relations between objects may be modeled using a graph. In this manner, a graph has vertices, or nodes, and lines, or edges, which interconnect the nodes. For example, sensors of a sensor network may be represented by corresponding nodes, where edges that extend between nodes are described by edge factors. In general, an "edge factor" represents a relationship among a pair of nodes (such as sensors). A given model for objects, such as sensors, may be developed at least in part from historical data (acquired sensor data, for example), which allows the construction of relatively small graphs (graphs having hundreds to thousands of sensors, for example). Relatively larger graphs (a graph describing a sensor network having hundreds to tens of millions of sensors, for example), may be significantly more challenging to develop.

[0007] Techniques and systems are described herein for purposes of synthetically generating a graph from a relatively smaller seed graph. In accordance with example implementations, the synthetically-generated graph may be a large, such as a graph that has hundreds tens of millions of nodes (as an example, and the seed graph may be small, such as a graph that has fewer than one hundred nodes (as an example). The synthetically-generated graph may be a small graph, in accordance with further example implementations, in that the techniques and systems, which are disclosed herein, may be used, in general, to synthetically generate a graph of any size while preserving the properties of the seed graph. The synthesized graph may be beneficial for such purposes as testing the scalability of software that is used to collect, manage and/or analyze data from a large sensor network; testing or evaluating the scaling of a large sensor network; and so forth.

[0008] In accordance with example implementations that are described herein, a relatively large graph is generated from a relatively small seed graph by determining nodes and edges of the larger graph, while preserving the properties of the original seed graph. More specifically, as described herein, in accordance with example implementations, a synthetic graph generator generates a synthetic Markov Random Field (MRF) graph, which has similar joint probability distributions as a smaller seed graph. In accordance with example systems and techniques that are described

herein, a synthetic graph may be generated having any arbitrary size while retaining properties, such as degree distributions, edge factor and the global structural relationships that exist in the original seed graph.

[0009] As a more specific example, Fig. 1 depicts a system 100, which includes a synthetic graph generation engine 130. In general, the synthetic graph generation engine 130 accesses data that describes a seed graph 110. In this regard, the seed graph 110 contains nodes 120, node attributes 122 and edge factors 124, describing the interconnection of the nodes 120. The synthetic graph generation engine 130 uses the seed graph 110 and observed/extracted properties of the seed graph 110 to generate a relatively larger, synthesized graph 150 while preserving properties of the seed graph 110. In this manner, the synthesized graph 150 is "larger," in that the synthesized graph has a larger number of nodes 160, corresponding node attributes 162, and edge factors 164. In accordance with example implementations, the synthetic graph generation engine 130 preserves global structural relationships of the seed graph 110.

[0010] In accordance with example implementations, the seed graph 110 is a spatial dependency graph that is constructed by a dependency graph generation engine 134 using historical data 102. In this context, a "dependency graph" describes how the nodes of the seed graph 110 depend on each other. As an example, the dependency graph may describe pairwise dependencies of nodes of the seed graph 110. In accordance with further example implementations, the seed graph 110 may already contain the dependencies, which may be, for example, based on an input from an expert.

[0011] In general, the historical data 102 contains data, which represents interrelationships between objects, which form the corresponding nodes of the graph 110. As an example, the objects may be sensors of a sensor network; and the seed graph 110 may be a graph that models the sensor network. The dependency graph generation engine 134, in general, learns the spatial dependency structure of the graph 110 from the historical data 102. This may be an offline process, in accordance with example implementations. In this manner, the dependency graph generation engine 134 may, in accordance with example implementations, construct

a pairwise node dependency graph from the historical data 102 and from this dependency graph, the engine 134 creates a pairwise Markov Random Field (MRF)-based seed graph 110.

[0012] More specifically, in accordance with example implementations, the dependency graph generation engine 134 discretizes the historical data 102. For example, in accordance with example implementations, the dependency graph generation engine 134 may apply a fixed width binning to discretize the domain of the historical data 102 into a finite number of intervals. As an example, the dependency graph generation engine 134 may create sixteen possible states for the node attributes. Because each node may assume one of the sixteen states, a pair of nodes may report 16x16=256 jointly occurring states. In other words, for this specific example, the dependency scored vector has a length of "256."

[0013] The engine 134 may initially recognize the identity of nodes 120 (corresponding to sensors at certain locations, for example) of the seed graph 110 but may not know many of the node attributes and edge factors 124 of the graph 110. For each pair of nodes 120, the data dependency graph generation engine 134 determines the frequency of co-occurring observations for the pair using the historical data 102; and the engine 134 normalizes the frequencies to a 0-1 scale. The normalized frequency data may be labeled a "dependency score."

[0014] For each pair of nodes of the seed graph 110, a corresponding dependency score vector of such scores is produced, and the dependency score vector is compared to a metric for purposes of determining whether an edge (and corresponding edge factor) exists between the nodes. For example, the metric may be a threshold value: if the maximum value in the dependency score vector exceeds the threshold value, then the dependency graph generation engine 134 creates an edge between the pair of nodes. The distribution of the dependency score vectors forms a "dependency score vector distribution." As an example, for a graph representing a sensor network having spatially distributed sensors, the corresponding dependency score vector distribution represents a spatial distribution of the dependency score vectors. In general, for each pair of nodes of the seed graph 110, a representation of a joint probability distribution for the pair (such as the

dependency score vector) is determined; and the pairs are then filtered to remove pairs from edge assignment based at least one metric of corresponding joint probability distribution representations. In accordance with some implementations, a frequency threshold may be established as the metric.

[0015] In accordance with example implementations, the dependency graph generation engine 134 may represent the seed graph 110 using a pairwise or bipartite MRF topology. Regardless of the particular structure of the graph 110, however, the synthetic graph generation engine 130 uses a learned dependency graph structure 138, which is generated by the dependency graph generation engine 134, for purposes of constructing the synthesized graph 150.

[0016] More specifically, in accordance with example implementations, the synthetic graph generation engine extracts a degree distribution from the dependency graph 138. In this regard, the degree of a graph is determined at nodes of the graph and refers to the number of edges that are connected to each node. If the graph represents a spatially varying network (a sensor network, for example), then the degree of the graph spatially varies.

[0017] In addition to the degree distribution indicated by the dependency graph 138, the synthetic graph generation engine 130 further uses a dependency score vector distribution 139 that is provided by the dependency graph generation engine 134. In particular, the synthetic graph generation engine 130 constructs the synthesized graph 150 so that the degree distribution of the synthesized graph 150 is representative of the degree distribution of the seed graph 110. In this manner, the construction of the synthesized graph 150 is based on a sampling of the degree distribution of the seed graph 110, such that the sampled degree distribution is represented by the synthesized graph 150. Moreover, in accordance with example implementations, the synthetic graph generation engine 130 samples the dependency score vector distribution for purposes of ensuring that the dependency score vector distribution of the seed graph 110 is represented in the synthesized graph 150. In other words, the synthetic graph generation engine 130 samples the edge factors of the seed graph 110 for purposes of creating a synthetic graph 150 that has the same distribution as the edge factors of the seed graph 110.

[0018] Thus, using the above-described sampling of the degree and edge factor distributions, the synthetic graph generation engine 130 constructs an initial topology for the synthesized graph 150 for purposes of creating the nodes and edge factors. The remaining step, the assignment of attributes of the nodes of the synthesized graph 150 may be performed as follows, in accordance with example implementations. The synthetic graph generation engine 130 selects a subset of the nodes 160 and assigns attributes to this subset of nodes 160. As a more specific example, in accordance with some implementations, the synthetic graph generation engine 130 randomly or pseudo randomly selects a small percentage (a percentage less than ten percent, for example) of the nodes 160 and randomly or pseudo randomly assigns attributes to the subset of nodes 160. It is noted that because the percentage of the subset of nodes 160 is relatively small, the assigned attributes are fairly independent of each other. Hence, randomly assigning the attributes does not conflict with the dependency structure of the synthesized graph 150.

[0019] Next, the synthetic graph generation engine 130 runs a message passing algorithm (a belief propagation algorithm or a loopy belief propagation algorithm, as examples) to predict, or estimate, the attributes for the remaining nodes 160. The message passing algorithm is one example of a graphical model inference algorithm, which is an algorithm that may be used to estimate one or multiple attributes of one or multiple nodes of a graph based on one or multiple known node attributes. Graphical model inference algorithms, other than a message passage algorithm (a variable elimination algorithm, a Markov chain Monte Carlo (MCMC) algorithm, a variational method, and so forth) may be used to predict, or estimate the attributes for the remaining nodes 160, in accordance with further example implementations.

[0020] Thus, referring to Fig. 2, in accordance with example implementations, the system 100 of Fig. 1 performs a technique 200 of Fig. 2. Pursuant to the technique 200, the system 100 receives (block 204) data describing a first graph having nodes, node attributes and edge factors and synthesizes (block 208) a relatively larger graph based at least in part on sampled edge factors of the first graph.

[0021] In this context, receiving the data refers to the system 100 accessing the data, such as accessing files of historical, or recorded data, acquired by sensors; the data

being communicated to the system 100 using a communication interface of the system 100; the data being entered into the system by a user; and so forth. Moreover, "synthesizing" a graph refers to creating a graph that is new in at least some aspect relative to the seed graph, such as creating a graph that has one or multiple nodes that are not present in the seed graph. As described herein, in accordance with example implementations, the dependency graph generation engine 134 may receive historical data 102 for the system 100; and the synthetic graph generation engine 130 may generate data for the system 100, which represents the synthesized graph 150 based on the dependency score vector distribution 139 and dependency graph 138 produced by the engine 134.

[0022] More specifically, in accordance with example implementations, the system 100 performs a technique 300 of Fig. 3. Referring to Fig. 3, the technique 300 includes receiving (block 304) data describing a first graph having nodes, node attributes and edge factors and determining (block 308) a degree distribution of the first graph. Pursuant to the technique 300, a relatively larger, second graph is constructed (block 312) based at least in part on the degree distribution of the first graph. In this regard, block 312 involves creating nodes and assigning edges and internode connections for the edges based on the degree distribution of the first graph such that the degree distribution of the second graph is approximately the same. Next, pursuant to the technique 300, edge factors are sampled (block 316) and assigned to the second graph based on the sampled edge factors. In this regard, block 316 is used to assign the edge factors and to ensure that the edge factor distribution of the second graph is approximately the same as the edge factor distribution of the first graph. In this context, the "edge factor distribution" refers to a collection of values derived from the edge factors, such as the probability distribution of the edge factors. For example, if the seed graph is a pairwise MRF that has n edges, then there is a factor associated with each edge, with a total of n edge factors; and the edge factor distribution is the probability distribution of these n factors. In accordance with example implementations, the edge factor distribution may be viewed as the histogram of the edge factors (that is, if there are k unique factors in the n factors, the histogram shows these k factors and their frequencies). The technique 300 further includes sampling (block 320) node attributes of the first

graph and assigning to seed nodes of the second graph. As discussed above, the assignment may be an assignment to a pseudo randomly or randomly selected small number of seed nodes in the second graph. Finally, pursuant to the technique 300, the attributes for the remaining nodes of the second graph may be determined using a message passing algorithm, pursuant to block 324.

[0023] For example implementations, in which belief propagation is used, the belief propagation algorithm may be performed pursuant to a technique 400 that is depicted in Fig. 4. Referring to Fig. 4, the technique 400 includes for each node, reading messages from neighboring nodes, updating a marginal belief and sending updated messages to neighboring nodes, pursuant to block 404. This process is repeated until a determination is made (decision block 408) that convergence has occurred.

[0024] Referring to Fig. 5, in accordance with example implementations, the system 100 may include a physical machine 500. In this manner, the physical machine 500 is an actual machine that is made up of actual hardware 510 and actual machine executable instructions 560, or "software."

[0025] The hardware 510 may include, for example, one or multiple central processing units (CPUs) 514 and a memory 516. In general, the memory 516 is a non-transitory storage medium that may store data, program instructions, data structures, and so forth, depending on the particular implementation. The memory 516 may be formed from semiconductor storage devices, phase change memory devices, magnetic storage devices, optical storage devices, memristors, and so forth, as well as one or more of these types. In accordance with example implementations, the memory 516 may store program instructions that when executed by the CPU(s) 514 cause the CPU(s) to form the synthetic graph generation engine 130 and the dependency graph generation engine 134

[0026] Therefore, in accordance with example implementations, the synthetic graph generation engine 130 and the dependency graph generation engine 134 may be software components, i.e., components formed by at least one processor executing machine executing instructions, or software. In further example implementations, the synthetic graph generation engine 130 and/or the dependency graph generation

engine 134 may be considered a hardware component that is formed from dedicated hardware (one or more integrated circuits that contain logic configured to perform outlier detection, as described herein, for example). Thus, the engines 130 and 134 may take on one of many different forms and may be based on software and/or hardware, depending on the particular implementation.

[0027] The memory 516 may store other data, in accordance with example implementations, such as data 561 describing the seed (or first) graph; data 562 describing the synthetically generated graph; data 570 describing a dependency score distribution; data 572 describing a dependency graph; and so forth. It is noted that the engines 130 and 134 are depicted as machine executable instructions 560 of the physical machine 500, in accordance with example implementation. The physical machine 500 may include other hardware 510 and machine executable instructions 560. In this regard, the physical machine 500 may include such additional hardware 510 as a network interface 520, input/output (I/O) devices, displays, and so forth. Moreover, the machine executable instructions 560 may also include an operating system 564, applications, device drivers 568, and so forth. Thus, many implementations are contemplated, which are within the scope of the appended claims.

[0028] While the present techniques have been described with respect to a number of embodiments, it will be appreciated that numerous modifications and variations may be applicable therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the scope of the present techniques.

What is claimed is:

1. A method comprising:

receiving data representing a first graph having nodes, node attributes and edge factors; and

synthesizing a second graph based at least in part on sampled edge factors of the first graph.

2. The method of claim 1, further comprising further basing the synthesizing on a degree distribution of the first graph.

3. The method of claim 1, further comprising sampling an edge distribution of the first graph and further basing the synthesizing on the sampled edge distribution.

4. The method of claim 3, further comprising determining the edge factor distribution, comprising determining a representation of a joint probability distribution for pairs of the nodes.

5. The method of claim 4, wherein determining the representation of the joint probability distribution comprises a determining a frequency of co-occurring observations for each pair of nodes based on the data.

6. The method of claim 1, wherein the synthesizing comprises:

selecting a subset of nodes of the larger graph;

assigning attributes to the selected nodes of the larger graph based on a sampling of the node attributes of the first graph; and

using a graphical model inference algorithm to determine attributes for other nodes of the larger graph.

7. An apparatus comprising:

a dependency graph generation engine comprising a processor to receive data representing nodes and to construct a dependency graph representing pairwise dependencies among the nodes; and

a synthetic graph generation engine comprising a processor to:

extract a degree distribution of the dependency graph;

extract a dependency score vector distribution for the dependency graph; and

synthesize a second graph based at least in part on the extracted degree distribution and the extracted dependency score vector distribution.

8. The apparatus of claim 7, wherein the dependency graph generation engine discretizes the data into a finite number of discrete values and determines the pairwise dependencies based at least in part on the discretized data.

9. The apparatus of claim 7, wherein the synthetic graph generation engine randomly or pseudo randomly selects subsets of the nodes of the second graph, assigns attributes to the selected nodes based on sampled attributes of the dependency graph, and applies a graphical model inference algorithm to determine attributes for remaining nodes of the second graph.

10. The apparatus of claim 9, wherein the graphical model inference algorithm comprises a belief propagation algorithm.

11. The apparatus of claim 7, wherein the second graph comprises a Markov Random Field (MRF)-based graph.

12. An article comprising a computer readable non-transitory storage medium to store instructions that when executed by a computer cause the computer to:

receive data describing a first graph having properties defined by an associated set of nodes, node attributes and edge factors; and

determine a second graph having more associated nodes than the first graph while preserving at least one or more of the properties of the first graph, the determination being based at least in part on sampled edge factors of the first graph.

13. The article of claim 12, the storage medium storing instructions that when executed by the computer cause the computer to sample an edge distribution of the first graph and further base the synthesization of the second graph on the sampled edge distribution.

14. The article of claim 12, the storage medium storing instructions that when executed by the computer cause the computer to:

select a subset of nodes of the second graph;

assign attributes to the selected nodes of the second graph based at least part on a sampling of the node attributes of the first graph; and

use a graphical model inference algorithm to determine attributes for other nodes of the second graph.

15. The article of claim 14, wherein the graphical model inference algorithm comprises a loopy belief propagation algorithm.

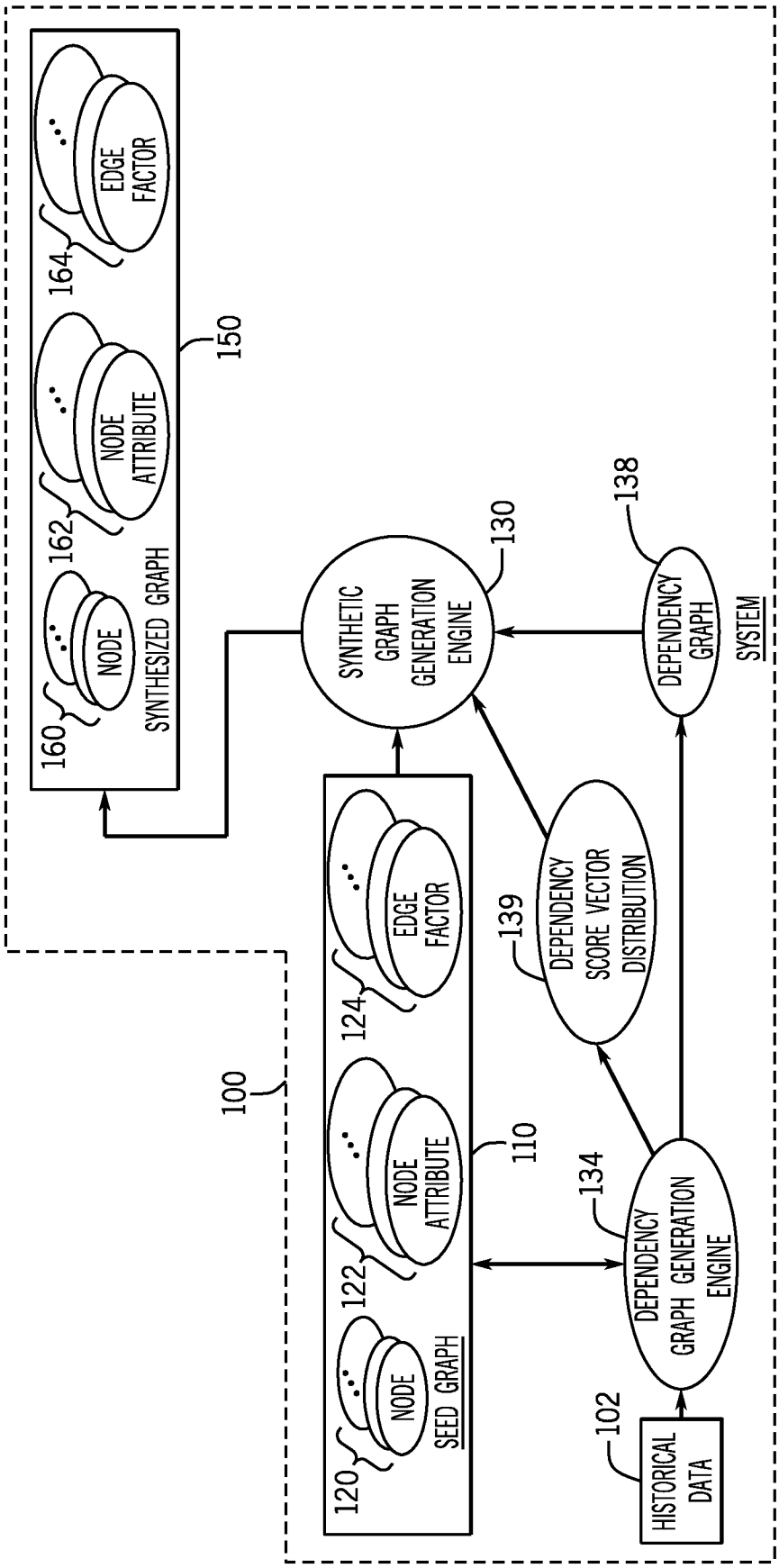

100
SYSTEM
SEED GRAPH
NODE
NODE ATTRIBUTE
EDGE FACTOR
120
122
124
110
HISTORICAL DATA
102
DEPENDENCY GRAPH GENERATION ENGINE
134
DEPENDENCY SCORE VECTOR DISTRIBUTION
139
SYNTHETIC GRAPH GENERATION ENGINE
130
DEPENDENCY GRAPH
138
SYNTHESIZED GRAPH
NODE
NODE ATTRIBUTE
EDGE FACTOR
160
162
164
150

FIG. 1

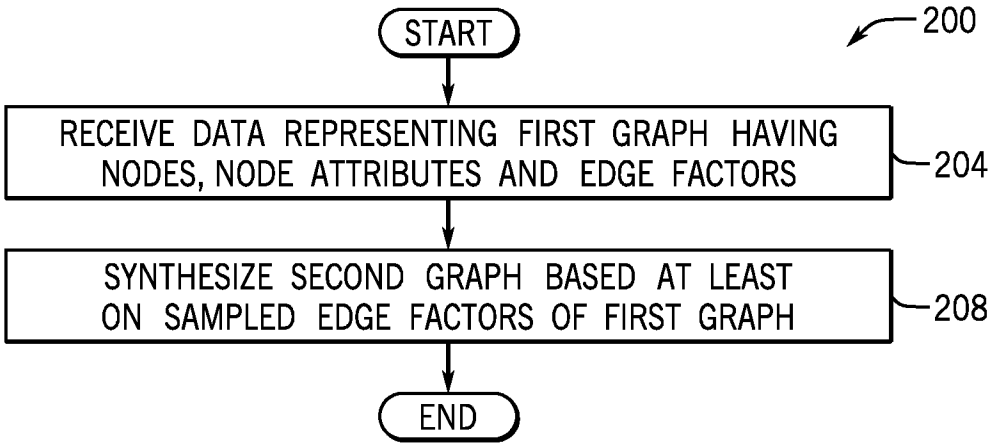

START
200
RECEIVE DATA REPRESENTING FIRST GRAPH HAVING NODES, NODE ATTRIBUTES AND EDGE FACTORS
204
SYNTHESIZE SECOND GRAPH BASED AT LEAST ON SAMPLED EDGE FACTORS OF FIRST GRAPH
208
END

FIG. 2

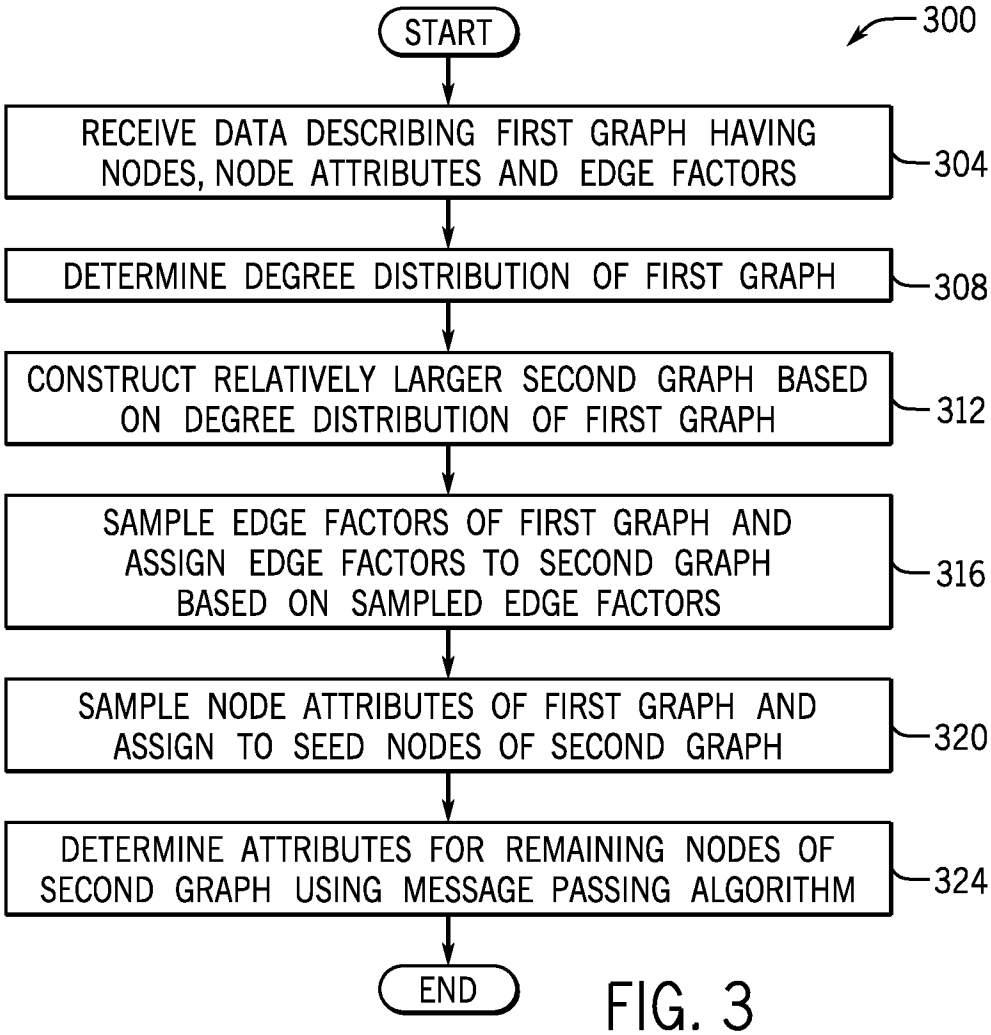

300
START
RECEIVE DATA DESCRIBING FIRST GRAPH HAVING NODES, NODE ATTRIBUTES AND EDGE FACTORS
304
DETERMINE DEGREE DISTRIBUTION OF FIRST GRAPH
308
CONSTRUCT RELATIVELY LARGER SECOND GRAPH BASED ON DEGREE DISTRIBUTION OF FIRST GRAPH
312
SAMPLE EDGE FACTORS OF FIRST GRAPH AND ASSIGN EDGE FACTORS TO SECOND GRAPH BASED ON SAMPLED EDGE FACTORS
316
SAMPLE NODE ATTRIBUTES OF FIRST GRAPH AND ASSIGN TO SEED NODES OF SECOND GRAPH
320
DETERMINE ATTRIBUTES FOR REMAINING NODES OF SECOND GRAPH USING MESSAGE PASSING ALGORITHM
324
END

FIG. 3

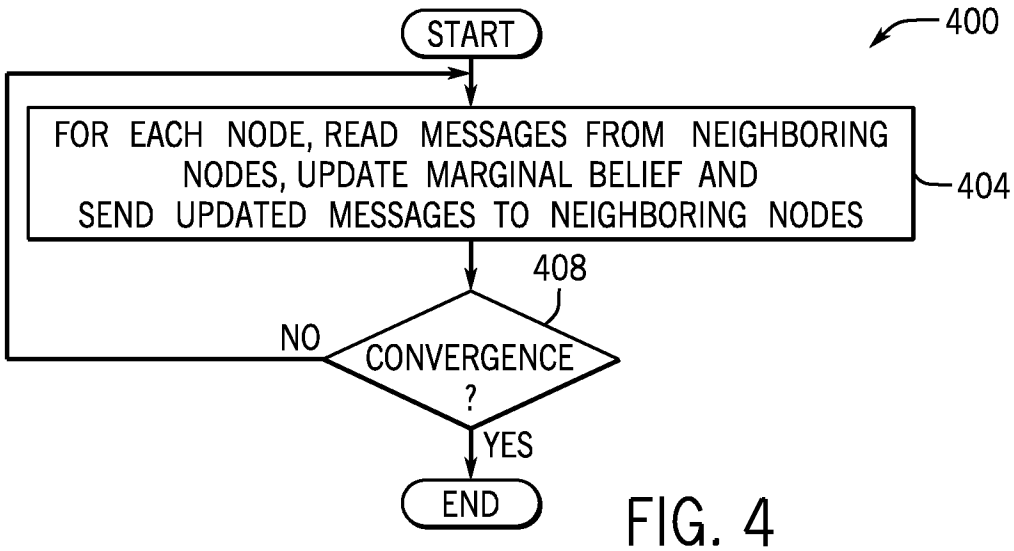

400
START
FOR EACH NODE, READ MESSAGES FROM NEIGHBORING NODES, UPDATE MARGINAL BELIEF AND SEND UPDATED MESSAGES TO NEIGHBORING NODES
404
408
CONVERGENCE ?
NO
YES
END

FIG. 4

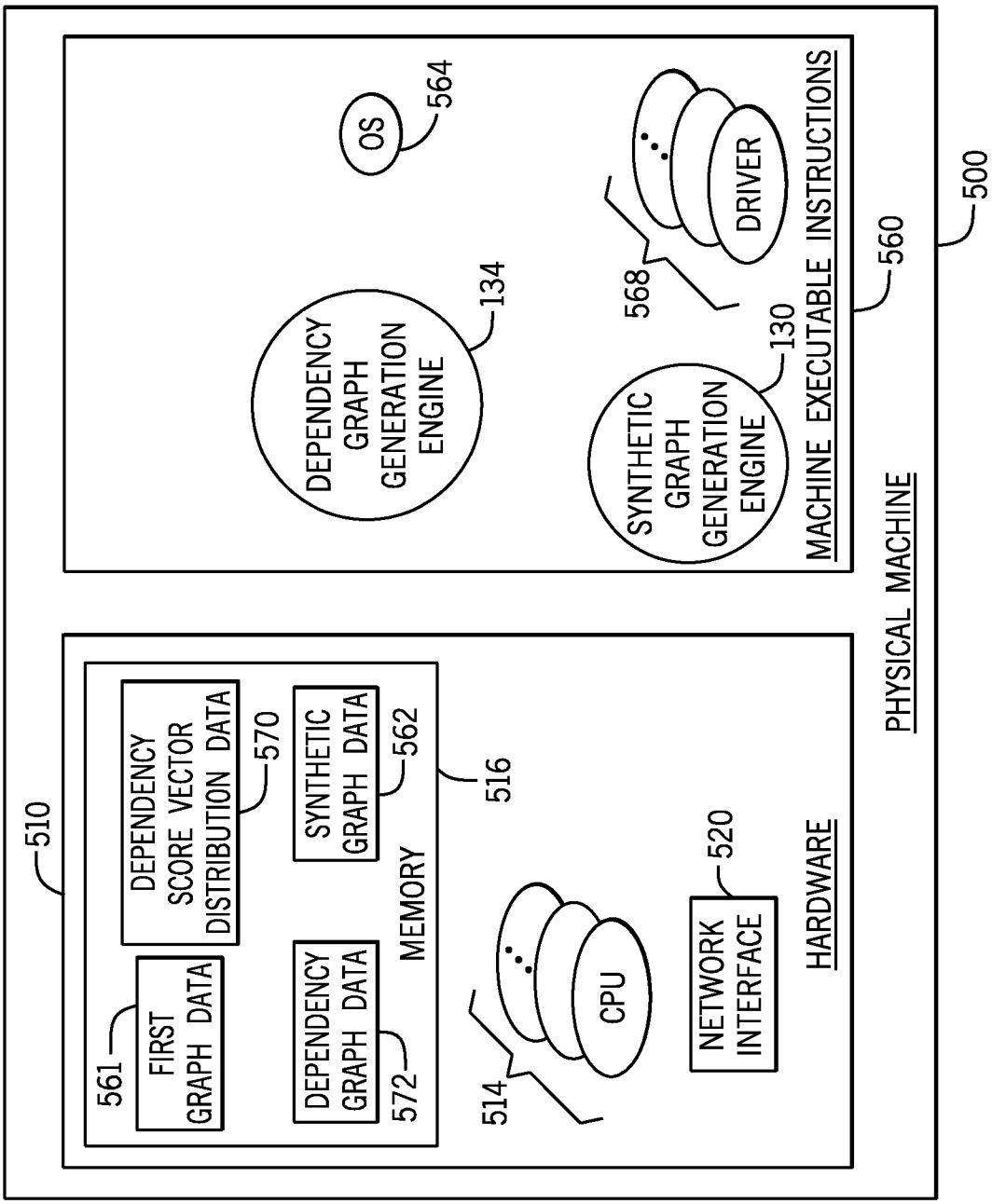

PHYSICAL MACHINE
500
MACHINE EXECUTABLE INSTRUCTIONS
560
OS
564
DEPENDENCY GRAPH GENERATION ENGINE
134
SYNTHETIC GRAPH GENERATION ENGINE
130
DRIVER
568
HARDWARE
510
CPU
514
NETWORK INTERFACE
520
MEMORY
516
FIRST GRAPH DATA
561
DEPENDENCY GRAPH DATA
572
SYNTHETIC GRAPH DATA
562
DEPENDENCY SCORE VECTOR DISTRIBUTION DATA
570

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013678

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(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; G06T 11/20; G06F 9/46

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: synthesizing, graph, degree distribution, node, sampled edge, 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.
X	US 2011-0074786 A1 (CHARU C. AGGARWAL) 31 March 2011 See paragraphs [0039], [0054]-[0056] and [0075]-[0076]; claims 1 and 5; and figure 5.	1
Y		2-5,12-13
A		6-11,14-15
Y	DAVIDE PROSERPIO et al., "A Workflow for Differentially-Private Graph Synthesis," In: Proceedings of the ACM SIGCOMM 2012 Workshop on Online Social Networks (WOSN'12), pp. 13-18, 17 August 2012 See pages 13 and 15-16.	2-5,13
Y	JURIJ LESKOVEC et al., "Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication," Knowledge Discovery in Databases: PKDD 2005 - 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, LNCS 3721, pp. 133-145, 03 October 2005 See pages 136-138 and figure 1.	12-13
A	US 2013-0247052 A1 (SHOUKAT ALI et al.) 19 September 2013 See paragraphs [0058]-[0064] and figure 10.	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	Date of mailing of the international search report
30 October 2015 (30.10.2015)	**30 October 2015 (30.10.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-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

International application No.
PCT/US2015/013678

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PRIYA MAHADEVAN et al., "Systematic Topology Analysis and Generation Using Degree Correlations," In: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM`06), pp. 135-146, 11-15 September 2006 See pages 135-136 and figure 1.	1-15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013678

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Group Ⅰ: Claims 1-6 and 12-15 are directed to synthesizing a second graph based at least in part on sampled edge factors of a first graph.

Group Ⅱ: Claims 7-11 are directed to synthesizing a second graph based at least in part on a degree distribution and a dependency score vector distribution.

1. ☐ As all required addtional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☒ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of any additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/013678

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0074786 A1	31/03/2011	KR 10-2011-0035899 A	06/04/2011
		US 2014-122540 A1	01/05/2014
		US 8659604 B2	25/02/2014
US 2013-0247052 A1	19/09/2013	US 9009007 B2	14/04/2015