Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3083008 A1 2019/06/06

(21) 3 083 008

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2018/11/30
(87) Date publication PCT/PCT Publication Date: 2019/06/06
(85) Entrée phase nationale/National Entry: 2020/05/20
(86) N° demande PCT/PCT Application No.: CA 2018/051534
(87) N° publication PCT/PCT Publication No.: 2019/104443
(30) Priorités/Priorities: 2017/12/01 (US62/593,563); 2018/08/08 (US62/716,041)

(51) Cl.Int./Int.Cl. *G06F 17/00* (2019.01)

(71) Demandeur/Applicant:
1QB INFORMATION TECHNOLOGIES INC., CA

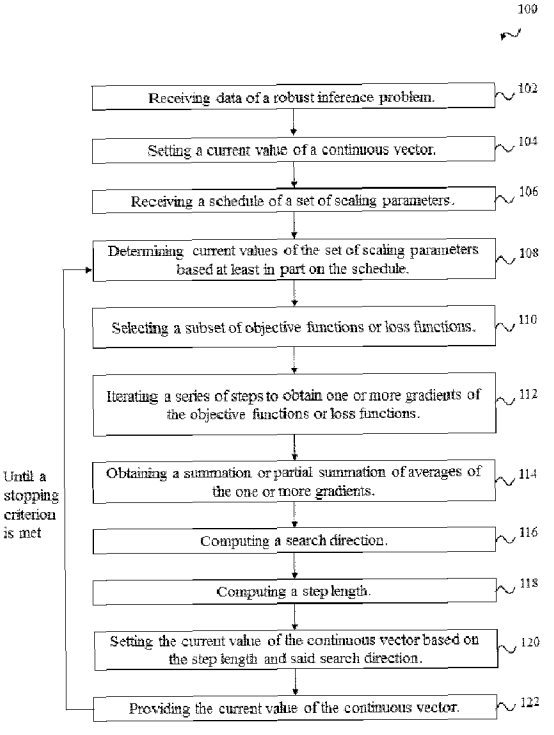
(72) Inventeurs/Inventors:
FRIEDLANDER, MICHAEL PAUL, CA;
RONAGH, POOYA, CA;
SEPEHRY, BEHROOZ, CA

(74) Agent: GOWLING WLG (CANADA) LLP

(54) Titre : SYSTEMES ET PROCEDES D'OPTIMISATION STOCHASTIQUE D'UN PROBLEME D'INFERENCE ROBUSTE

(54) Title: SYSTEMS AND METHODS FOR STOCHASTIC OPTIMIZATION OF A ROBUST INFERENCE PROBLEM

Fig. 1



(57) Abrégé/Abstract:
The present disclosure provides methods and systems for stochastic optimization of a robust inference problem using a sampling device. Specifically, the methods and systems of the present disclosure enable smoothing of objective functions, thereby making

(21) 3 083 008

(13) A1

(57) Abrégé(suite)/Abstract(continued):
such functions amenable to computation via stochastic-gradient methods using sampling in place of solving the inference problem exactly. Such methods and systems advantageously connect the gradient of the smoothed function approximation to a Boltzmann distribution, which can be sampled by a sampling device, such as a Gibbs sampler, using a simulated process and/or quantum process, in particular quantum-annealing process, thermal or adiabatic relaxation of a classical computer, semi-classical computer, or a quantum processor/device, and/or other physical process.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
06 June 2019 (06.06.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/104443 A1

(51) International Patent Classification:
G06F 17/00 (2019.01)

(21) International Application Number:
PCT/CA2018/051534

(22) International Filing Date:
30 November 2018 (30.11.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/593,563 01 December 2017 (01.12.2017) US
62/716,041 08 August 2018 (08.08.2018) US

(71) Applicant: **1QB INFORMATION TECHNOLOGIES INC.** [CA/CA]; 458-550 Burrard Street, Vancouver, British Columbia V6C 2B5 (CA).

(72) Inventors: **FRIEDLANDER, Michael Paul**; 458-550 Burrard Street, Vancouver, British Columbia V6C 2B5 (CA). **RONAGH, Pooya**; 458-550 Burrard Street, Vancouver, British Columbia V6C 2B5 (CA). **SEPEHRY, Behrooz**; 458-550 Burrard Street, Vancouver, British Columbia V6C 2B5 (CA).

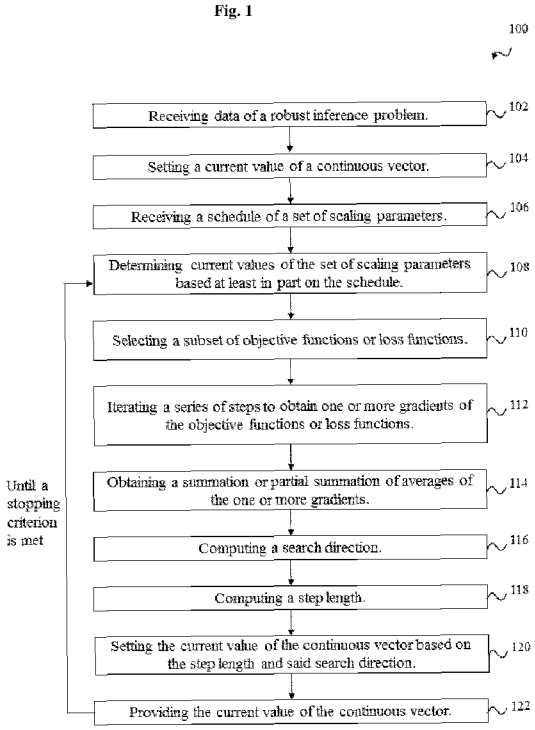
(74) Agent: **SCHROEDER, Hans** et al.; GOWLING WLG (CANADA) LLP, Suite 2600, 160 Elgin Street, Ottawa, Ontario K1P 1C3 (CA).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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).

(54) Title: SYSTEMS AND METHODS FOR STOCHASTIC OPTIMIZATION OF A ROBUST INFERENCE PROBLEM

Fig. 1



(57) Abstract: The present disclosure provides methods and systems for stochastic optimization of a robust inference problem using a sampling device. Specifically, the methods and systems of the present disclosure enable smoothing of objective functions, thereby making such functions amenable to computation via stochastic-gradient methods using sampling in place of solving the inference problem exactly. Such methods and systems advantageously connect the gradient of the smoothed function approximation to a Boltzmann distribution, which can be sampled by a sampling device, such as a Gibbs sampler, using a simulated process and/or quantum process, in particular quantum-annealing process, thermal or adiabatic relaxation of a classical computer, semi-classical computer, or a quantum processor/device, and/or other physical process.

WO 2019/104443 A1

[Continued on next page]

WO 2019/104443 A1

Declarations under Rule 4.17:

— *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))*

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

SYSTEMS AND METHODS FOR STOCHASTIC OPTIMIZATION OF A ROBUST INFERENCE PROBLEM

CROSS-REFERENCE

[001] This application claims priority to U.S. Provisional Patent Application No. 62/593,563, filed December 01, 2017, and U.S. Provisional Patent Application No. 62/716,041, filed August 08, 2018, each of which is entirely incorporated herein by reference.

BACKGROUND

[002] In various engineering fields, a robust inference problem often times can be too complex for solving it numerically directly even though the direct mathematical modeling may exist. Stochastic optimization is an approach for minimizing or maximizing a function that uses randomness to partially evaluate constituent functions and may thus be applicable to optimize very complex models.

SUMMARY

[003] Methods and systems of the present disclosure advantageously enable smoothing of various objective functions in robust inference problems, thereby making such functions amenable to computation via stochastic-gradient methods using sampling in place of solving the inference problem exactly. Such methods and systems advantageously connect the gradient of the smoothed function approximation to a Boltzmann distribution, which can be sampled by a sampling device using a simulated process and/or quantum process, in particular quantum-annealing process, thermal or adiabatic relaxation of a classical computer, semi-classical computer, or a quantum processor/device, and/or other physical process.

[004] The present disclosure provides systems for stochastic optimization of a robust inference problem, which may be used to learn or estimate the parameter(s) of a model expressed via a mathematical or a statistical function with a maximum margin principle and/or maximum likelihood principle, where the learned model parameter(s) determine an instance of the mathematical or statistical function. In particular, model parameter(s) may determine the weight(s) of a probabilistic graphical model for prediction in the case of the statistical function being the energy function of the graphical model, for example a transverse field Ising model or other classical or quantum model of a many-body system. This approach may provide a general framework for many machine learning

algorithms and tasks. Non-limiting examples of machine learning algorithms include structured support vector machines (SSVMs).

[005] Systems and methods of the present disclosure may advantageously improve the technical field of data science so that complex inference problems can be solved in various applications in data science, such as clustering of documents, group detection in a crowd, recommender systems, semi-supervised learning, and active learning. The systems and methods disclosed herein can also have various applications in natural language processing, such as noun phrase coreference resolution, and computer vision and image processing applications, such as image segmentation.

[006] In an aspect, the present disclosure provides a computer-implemented method for stochastic optimization of a robust inference problem using a sampling device, comprising receiving, by a digital computer, data of the robust inference problem, wherein the data comprises: a set of loss functions grouped into non-overlapping subsets, wherein each loss function in the set of loss functions accepts a first and second arguments, wherein the first and second arguments are independent, and wherein the first argument employs a continuous vector as its value, and the second argument employs a discrete vector as its value; a set of permissible discrete vectors for each loss function in the set of the loss functions; and an initial continuous vector for the first argument of each loss function in the set of loss functions; setting, by the digital computer, a current value of the continuous vector as the initial continuous vector; receiving, by the digital computer, a schedule of a set of scaling parameters; setting, by the digital computer, initial values of the set of scaling parameters based at least in part on the schedule; and until a stopping criterion is met, the stopping criterion comprising a set of rules for determining accuracy of a solution to the robust inference problem: determining current values of the set of scaling parameters, wherein the current values are based at least in part on the schedule of the set of scaling parameters; selecting a subset of the loss functions from the non-overlapping subsets, wherein the selection is non-repetitive or repetitive; iterating the following steps for each loss function of the selected subset of the loss functions: generating, by the sampling device, one or more samples of discrete vectors, each sample of the one or more samples being generated from the set of permissible discrete vectors associated with the loss function, wherein each sample of the one or more samples is generated based on a probability distribution determined at least in part by the set of scaling parameters and the loss function, wherein the first argument of the loss function takes the current value of the continuous vector; obtaining, by the digital computer, one or more gradients, wherein each of the one or more gradients is of the loss

function taken with respect to the first argument; wherein the first argument of the loss function takes the current value of the continuous vector, and the second argument takes value of a selected sample from the one or more samples, wherein the selected sample is non-repetitively selected; and obtaining, by the digital computer, an average of the one or more gradients; obtaining, by the digital computer, a summation and/or a partial summation of the averages of the one or more gradients, wherein the summation is for all loss functions in the selected subset of the loss functions, and wherein the partial summation is for more than one loss functions in the selected subset of the loss functions; computing, by the digital computer, a search direction based at least in part on: v1) the summation or the partial summation of the averages of the one or more gradients, v2) the current values of the set of scaling parameters, v3) at least part of a history of the summation or partial summation of the averages of the one or more gradients, and/or v4) at least part of a history of the values of the set of scaling parameters; computing, by the digital computer, a step length based at least in part on: vi1) the current values of the set of scaling parameters, vi2) the set of loss functions, vi3) at least part of a history of values of the set of scaling parameters, and/or vi4) at least part of a history of the set of loss functions; computing, by the digital computer, an updated current continuous vector using the step length and the search direction; and setting, by the digital computer, the current value of the continuous vector to be the updated current continuous vector.

[007] The present disclosure advantageously utilizes a sampling device for solving the complex robust inference problem. The sampling device can comprise a quantum processor and a quantum device control system for obtaining the schedule of the set of scaling parameters and the data of the robust inference problem. The quantum processor may be coupled to the digital computer and to the quantum device control system. The quantum processor may comprise a plurality of qubits and a plurality of couplers, each coupler of the plurality of couplers for providing a communicative coupling at a crossing of two qubits of the plurality of qubits. The one or more samples of discrete vectors may follow a Boltzmann distribution.

[008] The sampling device can be a network of optical parametric oscillators, the network can comprise: an optical device, the optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of the plurality of optical parametric oscillators. The sampling device may comprise a central processing unit, e.g., a digital computer or a mobile device, and a memory unit coupled to the central processing unit. The memory

unit may comprise an application for obtaining the schedule of the scaling parameter and the data of the robust inference problem. Such application can be web application or mobile application.

[009] The sampling device can comprise a reconfigurable digital hardware, a central processing unit and a memory unit, the central processing unit and the memory unit coupled to the reconfigurable digital hardware. The reconfigurable digital hardware may be adapted for obtaining the schedule of the scaling parameter and the data of the robust inference problem, and wherein the reconfigurable digital hardware is adapted to perform a Markov Chain Monte Carlo algorithm. The Markov Chain Monte Carlo algorithm may be Simulated Quantum Annealing. The Markov Chain Monte Carlo algorithm may be Simulated Annealing. The Markov Chain Monte Carlo algorithm may be Gibbs Sampling.

[010] The set of loss functions may comprise one or more loss functions.

[011] The stochastic optimization of the robust inference problem may be associated with training a structured support vector machine. Each subset of the non-overlapping subsets of loss functions may comprise only two loss functions. The stochastic optimization of the robust inference problem may be associated with image segmentation.

[012] The stochastic optimization of the robust inference problem may be associated with a dual of the basis pursuit problem from compressed sensing.

[013] The stochastic optimization of the robust inference problem may be associated with semi-supervised learning. The data of the robust inference problem can be associated with one or more image segmentation problems. The data of the robust inference problem can be associated with a dual of the basis pursuit problem from one or more compressed sensing problems. The data of the robust inference problem may be associated with semi-supervised learning. The data of the robust inference problem can be obtained from a noun phrase co-reference resolution problem. The data of the robust inference problem may be associated with active learning. The data of the robust inference problem may be associated with one or more image tagging problems. The data of the robust inference problem may be associated with a recommender system.

[014] The schedule of the set of scaling parameters can be determined manually by a user or automatically by an algorithm or a computer program. The schedule of the set of scaling parameters can be determined using a machine learning algorithm based on history of the set of scaling parameters.

[015] The digital computer can be remotely located with respect to the sampling device.

[016] The stopping criterion may be based at least in part on a magnitude of a distance between the current and the updated current continuous vectors.

[017] The loss functions can comprise of composite functions of the first and second set of arguments.

[018] In the operation of obtaining one or more gradients, each of the one or more gradients may be of the loss function taken with respect to the first argument comprises of iterative applications of chain rule. The iterative application of chain rule may be performed using auto-differentiation.

[019] In some cases, in the argument functions of the composite functions are differentiable feature extractors. In some cases, in the differentiable feature extractors are deep neural networks.

[020] Computing a search direction may utilize one or more of: stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), and stochastic dual coordinate ascent (SDCA).

[021] Computing a step length uses one of the adaptive gradient descent methods may include but may not be limited to Adam, reduced mean square (RMS), RMSProp, and AdaGrad.

[022] In an aspect, the present disclosure provides a system for stochastic optimization of a robust inference problem using a sampling device, comprising a digital computer configured to: receive data of the robust inference problem, wherein the data comprises: a set of loss functions grouped into non-overlapping subsets, wherein each loss function in the set of loss functions accepts a first and second arguments, wherein the first and second arguments are independent, and wherein the first argument employs a continuous vector as its value, and the second argument employs a discrete vector as its value; a set of permissible discrete vectors for each loss function in the set of the loss functions; and an initial continuous vector for the first argument of each loss function in the set of loss functions; set a current value of the continuous vector as the initial continuous vector; receive a schedule of a set of scaling parameters; set initial values of the set of scaling parameters based at least in part on the schedule; and until a stopping criterion is met, the stopping criterion comprising a set of rules for determining accuracy of a solution to the robust inference problem: determine current values of the set of scaling parameters, wherein the current values are based at least in part on the schedule of the set of scaling parameters; select a subset of the loss functions from the non-overlapping subsets, wherein the selection is non-repetitive or repetitive; iterate the following steps for each loss function of the selected subset of the loss functions: generating, by the sampling device, one or more samples of discrete vectors, each sample of the one or more samples being generated

from the set of permissible discrete vectors associated with the loss function, wherein each sample of the one or more samples is generated based on a probability distribution determined at least in part by the set of scaling parameters and the loss function, wherein the first argument of the loss function takes the current value of the continuous vector; obtaining one or more gradients, wherein each of the one or more gradients is of the loss function taken with respect to the first argument; wherein the first argument of the loss function takes the current value of the continuous vector, and the second argument takes value of a selected sample from the one or more samples, wherein the selected sample is non-repetitively selected; and obtaining an average of the one or more gradients; obtain a summation and/or a partial summation of the averages of the one or more gradients, wherein the summation is for all loss functions in the selected subset of the loss functions, and wherein the partial summation is for more than one loss functions in the selected subset of the loss functions; compute a search direction based at least in part on: v1) the summation or the partial summation of the averages of the one or more gradients, v2) the current values of the set of scaling parameters, v3) at least part of a history of the summation or partial summation of the averages of the one or more gradients, and/or v4) at least part of a history of the values of the set of scaling parameters; compute a step length based at least in part on: vi1) the current values of the set of scaling parameters, vi2) the set of loss functions, vi3) at least part of a history of values of the set of scaling parameters, and/or vi4) at least part of a history of the set of loss functions; compute an updated current continuous vector using the step length and the search direction; and set the current value of the continuous vector to be the updated current continuous vector.

[023] The present disclosure advantageously utilizes a sampling device for solving the complex robust inference problem. The sampling device can comprise a quantum processor and a quantum device control system for obtaining the schedule of the set of scaling parameters and the data of the robust inference problem. The quantum processor may be coupled to the digital computer and to the quantum device control system. The quantum processor may comprise a plurality of qubits and a plurality of couplers, each coupler of the plurality of couplers for providing a communicative coupling at a crossing of two qubits of the plurality of qubits. The one or more samples of discrete vectors may follow a Boltzmann distribution.

[024] The sampling device can be a network of optical parametric oscillators, the network can comprise: an optical device, the optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and a plurality of coupling devices,

each of which controllably couples an optical parametric oscillator of the plurality of optical parametric oscillators. The sampling device may comprise a central processing unit, e.g., a digital computer or a mobile device, and a memory unit coupled to the central processing unit. The memory unit may comprise an application for obtaining the schedule of the scaling parameter and the data of the robust inference problem. Such application can be web application or mobile application.

[025] The sampling device can comprise a reconfigurable digital hardware, a central processing unit and a memory unit, the central processing unit and the memory unit coupled to the reconfigurable digital hardware. The reconfigurable digital hardware may be adapted for obtaining the schedule of the scaling parameter and the data of the robust inference problem, and wherein the reconfigurable digital hardware is adapted to perform a Markov Chain Monte Carlo algorithm. The Markov Chain Monte Carlo algorithm may be Simulated Quantum Annealing. The Markov Chain Monte Carlo algorithm may be Simulated Annealing. The Markov Chain Monte Carlo algorithm may be Gibbs Sampling.

[026] The set of loss functions may comprise one or more loss functions.

[027] The stochastic optimization of the robust inference problem may be associated with training a structured support vector machine. Each subset of the non-overlapping subsets of loss functions may comprise only two loss functions. The stochastic optimization of the robust inference problem may be associated with image segmentation.

[028] The stochastic optimization of the robust inference problem may be associated with a dual of the basis pursuit problem from compressed sensing.

[029] The stochastic optimization of the robust inference problem may be associated with semi-supervised learning. The data of the robust inference problem can be associated with one or more image segmentation problems. The data of the robust inference problem can be associated with a dual of the basis pursuit problem from one or more compressed sensing problems. The data of the robust inference problem may be associated with semi-supervised learning. The data of the robust inference problem can be obtained from a noun phrase co-reference resolution problem. The data of the robust inference problem may be associated with active learning. The data of the robust inference problem may be associated with one or more image tagging problems. The data of the robust inference problem may be associated with a recommender system.

[030] The schedule of the set of scaling parameters can be determined manually by a user or automatically by an algorithm or a computer program. The schedule of the set of scaling parameters

can be determined using a machine learning algorithm based on history of the set of scaling parameters.

[031] The digital computer can be remotely located with respect to the sampling device.

[032] The stopping criterion may be based at least in part on a magnitude of a distance between the current and the updated current continuous vectors.

[033] The loss functions can comprise of composite functions of the first and second set of arguments.

[034] In the operation of obtaining one or more gradients, each of the one or more gradients may be of the loss function taken with respect to the first argument comprises of iterative applications of chain rule. The iterative application of chain rule may be performed using auto-differentiation.

[035] In some cases, in the argument functions of the composite functions are differentiable feature extractors. In some cases, in the differentiable feature extractors are deep neural networks.

[036] Computing a search direction may utilize one or more of: stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), and stochastic dual coordinate ascent (SDCA).

[037] Computing a step length uses one of the adaptive gradient descent methods may include but may not be limited to Adam, reduced mean square (RMS), RMSProp, and AdaGrad.

[038] In another aspect, a computer-implemented method for stochastic optimization of a robust inference problem using a sampling device may comprise: (a) receiving, by a digital computer, data of said robust inference problem, wherein said data comprises: (i) a set of objective functions or loss functions grouped into non-overlapping subsets, wherein each objective function or loss function in said set of loss functions accepts first and second arguments; and (ii) a set of permissible vectors for each objective function or loss function in said set of said objective functions or loss functions; (b) setting, by said digital computer, a current value of a vector; (c) receiving, by said digital computer, a schedule of a set of scaling parameters; and (d) until a stopping criterion is met: (i) determining current values of said set of scaling parameters based at least in part on said schedule; (ii) selecting a subset of said objective functions or loss functions from said non-overlapping subsets; (iii) iterating the following steps for each objective function or loss function of said selected subset of said objective functions or loss functions: (1) generating, by said sampling device, one or more samples of vectors from said set of permissible vectors associated with said objective function or loss function; (2) obtaining, by said digital computer, one or more gradients, wherein each of said one or

more gradients is of said objective function or loss function taken with respect to said first argument; and (3) obtaining, by said digital computer, an average of said one or more gradients; (iv) obtaining, by said digital computer, a summation or a partial summation of said averages of said one or more gradients, wherein said summation is for all objective functions or loss functions in said selected subset of said objective functions or loss functions, and wherein said partial summation is for more than one objective function or loss function in said selected subset of said loss functions; (v) computing, by said digital computer, a search direction based at least in part on one or more of: v1) said summation or said partial summation of said averages of said one or more gradients; v2) said current values of said set of scaling parameters; v3) at least part of a history of said summation or partial summation of said averages of said one or more gradients; and v4) at least part of a history of said values of said set of scaling parameters; (vi) computing, by said digital computer, a step length based at least in part on one or more of: vi1) said current values of said set of scaling parameters; vi2) said selected subset of said loss functions; vi3) at least part of a history of values of said set of scaling parameters; and vi4) at least part of a history of said selected subset of said objective functions or loss functions; and (vii) setting, by said digital computer, said current value of said vector based on said step length and said search direction. Said objective functions or loss functions may comprise one or more composite functions of said first and second arguments. Obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument may comprise iterative applications of a chain rule. Said chain rule may be performed using auto-differentiation. One or more argument functions of said composite functions may comprise differentiable feature extractors. Said differentiable feature extractors may comprise deep neural networks. Computing, by said digital computer, a search direction may comprise using one or more of stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), or stochastic dual coordinate ascent (SDCA). Computing, by said digital computer, a step length may comprise using one or more of said adaptive gradient descent methods, and wherein said adaptive gradient descent methods comprises Adaptive Moment Estimation (Adam), reduced mean square (RMS), Root Mean Square Propagation (RMSProp), and/or adaptive gradient algorithm. (AdaGrad). Said sampling device may comprise a quantum processor and a quantum device control system for obtaining said schedule of said set of scaling parameters and said data of said robust inference problem. Said quantum processor may be coupled to said digital computer and to said

quantum device control system. Said quantum processor may comprise a plurality of qubits and a plurality of couplers, each coupler of said plurality of couplers for providing a communicative coupling at a crossing of two qubits of said plurality of qubits. Said one or more samples of discrete vectors may follow a Boltzmann distribution. Said sampling device may comprise a network of optical parametric oscillators, said network comprising: (a) an optical device, said optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and (b) a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of said plurality of optical parametric oscillators. Said sampling device may comprise a central processing unit and a memory unit coupled to said central processing unit. Said memory unit may comprise an application for obtaining said schedule of said scaling parameter and said data of said robust inference problem, wherein said application is configured to implement a Markov Chain Monte Carlo algorithm. Said sampling device may comprise a reconfigurable digital hardware, a central processing unit and a memory unit, said central processing unit and said memory unit coupled to said reconfigurable digital hardware. Said reconfigurable digital hardware may be configured to obtain said schedule of said scaling parameter and said data of said robust inference problem, and said reconfigurable digital hardware may be configured to implement a Markov Chain Monte Carlo algorithm. Said Markov Chain Monte Carlo algorithm may comprise simulated quantum annealing. Said Markov Chain Monte Carlo algorithm may comprise simulated annealing. Said Markov Chain Monte Carlo algorithm may comprise Gibbs sampling. Said set of objective functions or loss functions may comprise one or more objective functions or loss functions. Said stochastic optimization of said robust inference problem may be associated with training a structured support vector machine. Each subset of said non-overlapping subsets of objective functions or loss functions may comprise only two objective functions or loss functions. Said data of said robust inference problem may be associated with an image segmentation problem. Said data of said robust inference problem may be associated with a dual of said basis pursuit problem from a compressed sensing problem. Said data of said robust inference problem may be associated with semi-supervised learning. Said data of said robust inference problem may be obtained from a noun phrase co-reference resolution problem. Said data of said robust inference problem may be associated with active learning. Said data of said robust inference problem may be associated with an image tagging problem. Said data of said robust inference problem may be associated with a recommender system. Said schedule of said set of scaling parameters may be determined by a user or automatically by an

algorithm, Said digital computer may be remotely located with respect to said sampling device. Said stopping criterion may be based at least in part on a magnitude of a distance between said current and said updated current vectors. Said first and second arguments may be independent, and said first argument may employ a continuous vector as its value, said second argument may employ a discrete vector as its value, and said set of permissible vectors may comprise a set of permissible discrete vectors. (1) may comprise generating, by said sampling device, one or more samples of discrete vectors, each sample of said one or more samples being generated from said set of permissible discrete vectors associated with said objective function or loss function, wherein each sample of said one or more samples is generated based on a probability distribution determined at least in part by said set of scaling parameters and said objective function or loss function, wherein said first argument of said objective function or loss function takes said current value of said continuous vector. (2) may comprise obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said loss function taken with respect to said first argument, wherein said first argument of said loss function takes said current value of said continuous vector, and said second argument takes value of a selected sample from said one or more samples, wherein said selected sample is non-repetitively selected. Said stopping criterion may comprise a set of rules for determining accuracy of a solution to said robust inference problem. Said selection of said subset of objective functions or loss functions may be non-repetitive or repetitive.

[039] In another aspect, a system for stochastic optimization of a robust inference problem using a sampling device may comprise a digital computer configured to: (a) receive data of said robust inference problem, wherein said data comprises: (i) a set of objective functions or loss functions grouped into non-overlapping subsets, wherein each objective function or loss function in said set of loss functions accepts first and second arguments; and (ii) a set of permissible vectors for each objective function or loss function in said set of said objective functions or loss functions; (b) set a current value of a vector; (c) receive a schedule of a set of scaling parameters; and (d) until a stopping criterion is met: (i) determine current values of said set of scaling parameters based at least in part on said schedule; (ii) select a subset of said objective functions or loss functions from said non-overlapping subsets; (iii) iterate the following steps for each objective function or loss function of said selected subset of said objective functions or loss functions: (1) generating, by said sampling device, one or more samples of vectors from said set of permissible vectors associated with said objective function or loss function; (2) obtaining, by said digital computer, one or more gradients,

wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument; and (3) obtaining an average of said one or more gradients; (iv) obtain a summation or a partial summation of said averages of said one or more gradients, wherein said summation is for all objective functions or loss functions in said selected subset of said objective functions or loss functions, and wherein said partial summation is for more than one objective function or loss function in said selected subset of said loss functions; (v) compute a search direction based at least in part on one or more of: v1) said summation or said partial summation of said averages of said one or more gradients; v2) said current values of said set of scaling parameters; v3) at least part of a history of said summation or partial summation of said averages of said one or more gradients; and v4) at least part of a history of said values of said set of scaling parameters; (vi) compute a step length based at least in part on one or more of: vi1) said current values of said set of scaling parameters; vi2) said selected subset of said loss functions; vi3) at least part of a history of values of said set of scaling parameters; and vi4) at least part of a history of said selected subset of said objective functions or loss functions; and (vii) set said current value of said vector based on said step length and said search direction. Said objective functions or loss functions may comprise one or more composite functions of said first and second arguments. Obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument may comprise iterative applications of a chain rule. Said chain rule may be performed using auto-differentiation. One or more argument functions of said composite functions may comprise differentiable feature extractors. Said differentiable feature extractors may comprise deep neural networks. Computing, by said digital computer, a search direction may comprise using one or more of stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), or stochastic dual coordinate ascent (SDCA). Computing, by said digital computer, a step length may comprise using one or more of said adaptive gradient descent methods, and wherein said adaptive gradient descent methods comprises Adaptive Moment Estimation (Adam), reduced mean square (RMS), Root Mean Square Propagation (RMSProp), and/or adaptive gradient algorithm. (AdaGrad). Said sampling device may comprise a quantum processor and a quantum device control system for obtaining said schedule of said set of scaling parameters and said data of said robust inference problem. Said quantum processor may be coupled to said digital computer and to said quantum device control system. Said quantum processor may comprise a plurality of qubits and a plurality of

couplers, each coupler of said plurality of couplers for providing a communicative coupling at a crossing of two qubits of said plurality of qubits. Said one or more samples of discrete vectors may follow a Boltzmann distribution. Said sampling device may comprise a network of optical parametric oscillators, said network comprising: (a) an optical device, said optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and (b) a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of said plurality of optical parametric oscillators. Said sampling device may comprise a central processing unit and a memory unit coupled to said central processing unit. Said memory unit may comprise an application for obtaining said schedule of said scaling parameter and said data of said robust inference problem, wherein said application is configured to implement a Markov Chain Monte Carlo algorithm. Said sampling device may comprise a reconfigurable digital hardware, a central processing unit and a memory unit, said central processing unit and said memory unit coupled to said reconfigurable digital hardware. Said reconfigurable digital hardware may be configured to obtain said schedule of said scaling parameter and said data of said robust inference problem, and said reconfigurable digital hardware may be configured to implement a Markov Chain Monte Carlo algorithm. Said Markov Chain Monte Carlo algorithm may comprise simulated quantum annealing. Said Markov Chain Monte Carlo algorithm may comprise simulated annealing. Said Markov Chain Monte Carlo algorithm may comprise Gibbs sampling. Said set of objective functions or loss functions may comprise one or more objective functions or loss functions. Said stochastic optimization of said robust inference problem may be associated with training a structured support vector machine. Each subset of said non-overlapping subsets of objective functions or loss functions may comprise only two objective functions or loss functions. Said data of said robust inference problem may be associated with an image segmentation problem. Said data of said robust inference problem may be associated with a dual of said basis pursuit problem from a compressed sensing problem. Said data of said robust inference problem may be associated with semi-supervised learning. Said data of said robust inference problem may be obtained from a noun phrase co-reference resolution problem. Said data of said robust inference problem may be associated with active learning. Said data of said robust inference problem may be associated with an image tagging problem. Said data of said robust inference problem may be associated with a recommender system. Said schedule of said set of scaling parameters may be determined by a user or automatically by an algorithm, Said digital computer may be remotely located with respect to said sampling device. Said

stopping criterion may be based at least in part on a magnitude of a distance between said current and said updated current vectors. Said first and second arguments may be independent, and said first argument may employ a continuous vector as its value, said second argument may employ a discrete vector as its value, and said set of permissible vectors may comprise a set of permissible discrete vectors. (1) may comprise generating, by said sampling device, one or more samples of discrete vectors, each sample of said one or more samples being generated from said set of permissible discrete vectors associated with said objective function or loss function, wherein each sample of said one or more samples is generated based on a probability distribution determined at least in part by said set of scaling parameters and said objective function or loss function, wherein said first argument of said objective function or loss function takes said current value of said continuous vector. (2) may comprise obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said loss function taken with respect to said first argument, wherein said first argument of said loss function takes said current value of said continuous vector, and said second argument takes value of a selected sample from said one or more samples, wherein said selected sample is non-repetitively selected. Said stopping criterion may comprise a set of rules for determining accuracy of a solution to said robust inference problem. Said selection of said subset of objective functions or loss functions may be non-repetitive or repetitive.

[040] Another aspect of the present disclosure provides a non-transitory computer readable medium comprising machine executable code that, upon execution by one or more computer processors, implements any of the methods above or elsewhere herein.

[041] Another aspect of the present disclosure provides a system comprising one or more computer processors and a non-transitory computer readable medium (e.g., computer memory) coupled thereto. The non-transitory computer readable medium comprises machine executable code that, upon execution by the one or more computer processors, implements any of the methods above or elsewhere herein.

[042] Additional aspects and advantages of the present disclosure will become readily apparent to those skilled in this art from the following detailed description, wherein only illustrative embodiments of the present disclosure are shown and described. As will be realized, the present disclosure is capable of other and different embodiments, and its several details are capable of modifications in various obvious respects, all without departing from the disclosure. Accordingly, the drawings and description are to be regarded as illustrative in nature, and not as restrictive.

INCORPORATION BY REFERENCE

[043] All publications, patents, and patent applications mentioned in this specification are herein incorporated by reference to the same extent as if each individual publication, patent, or patent application was specifically and individually indicated to be incorporated by reference. To the extent publications and patents or patent applications incorporated by reference contradict the disclosure contained in the specification, the specification is intended to supersede and/or take precedence over any such contradictory material.

BRIEF DESCRIPTION OF THE DRAWINGS

[044] A better understanding of the features and advantages of the present subject matter will be obtained by reference to the following detailed description that sets forth illustrative embodiments and the accompanying drawings of which:

[045] **Fig. 1** shows a flowchart for a non-limiting example of a method for stochastic optimization of a robust inference problem using a sampling device.

[046] **Fig. 2** shows a non-limiting example of a system for stochastic optimization of a robust inference problem.

DETAILED DESCRIPTION

[047] While various embodiments of the invention have been shown and described herein, it will be obvious to those skilled in the art that such embodiments are provided by way of example only. Numerous variations, changes, and substitutions may occur to those skilled in the art without departing from the invention. It should be understood that various alternatives to the embodiments of the invention described herein may be employed.

[048] As used herein, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise. Any reference to “or” herein is intended to encompass “and/or” unless otherwise stated.

[049] As disclosed herein, “variable” is equivalent to “argument” herein.

[050] As disclosed herein, setting a value of a vector, e.g., a continuous vector or discrete vector can be setting values for every element of the vector. In other cases, setting a value of a vector can be setting values for one or more element of the vector.

Robust inference problems

[051] In an aspect, the present disclosure provides methods and systems that utilize a sampling

device within a stochastic optimization method for solving robust inference problems. The methods and systems may provide a framework that enables efficient and robust optimization techniques in machine learning methods. Nonlimiting examples of machine-learning methods include: structural support vector machines (SSVMs), semi-supervised learning, and active learning. These methods may be useful in applications such as natural language processing (e.g., noun phrase coreference resolution), computer vision and image processing (e.g., image segmentation, image tagging), and data science (e.g. clustering of documents, group detection in a crowd, recommender systems, semi-supervised, and active learning, etc).

[052] The robust inference problems herein may be associated with the robustness and/or accuracy of inference or assumption(s) under which a solution may be found for a computational task. In other words, how much deviation(s) from the inference or assumption may occur under which a solution may be obtained. The robustness of an inference method may pertain to its resistance to outliers, such as by training models that increase the probability of or guarantee good performance even for non-ideal or least confident predictions. The robust inference problems may be expressed as in equation (1):

$$\min_{x\in C}(f(x)+\sum_{i_1}^{m_1}\sum_{i_2}^{m_2}\omega_{i_1i_2}\max_{y\in Y_{i_1i_2}}g_{i_1i_2}(x,y)) \quad (1)$$

where $C \subseteq \mathbb{R}^n$ can be a set that defines admissible vectors in an n-dimensional real vector space; $f\colon \mathbb{R}^n \to \mathbb{R}$ can be a function mapping vectors in an n-dimensional real vector space to a real value; every $g_{i_1,i_2}\colon \mathbb{R}^n \times \mathrm{Y}_{\mathrm{i}_1\mathrm{i}_2} \to \mathbb{R}$ can be a real-valued function; ω_{i_1,i_2} can be real numbers, $\mathrm{Y}_{\mathrm{i}_1\mathrm{i}_2}$ can be sets of vectors (such as finite set of vectors) from which argument y can take values, i_1 and i_2 can be independent indexes in the range from 1 to m_1 and 1 to m_2, respectively, and x and y can be two arguments (such as two independent arguments) which may be any vector from C and $\mathrm{Y}_{\mathrm{i}_1\mathrm{i}_2}$, respectively. Optimization of the robust inference problems described herein may refer to solving a minimization problem as described in equation (1) or any other problem that may be equivalent to or can be described by equation (1). In some cases, the set C is a convex set, and the function $f(x)$ and all functions g_{i_1,i_2} are convex, and all real numbers ω_{i_1,i_2} are positive. In such a case, the optimization problem described by equation (1) may become a convex optimization problem, allowing convex optimization methods to be employed in order to create convenient approximations of the global optimal solution. Such convex optimization methods may be efficiently implemented using procedures that scale with polynomial time. However, in other cases, the optimization problem

may be non-convex. For example, when the real numbers ω_{i_1,i_2} may be negative, as in a latent SSVM optimization problem, the optimization problem can be non-convex. As another example, when the functions g_{i_1,i_2} are non-convex, as in the case that the functions g_{i_1,i_2} correspond to neural networks, the optimization problem in (1) can be non-convex.

[053] Referring to one of the functions (e.g., objective functions or loss functions) $g_{i_1,i_2}(x,y)$ (also referred to as $g(x,y)$ herein), the function $g(x) \coloneqq \max_{y \in Y} g(x,y)$ may be only differentiable with respect to x where the max is uniquely attained. In the case when the max (e.g., $\max_{y \in Y} g(x, y)$) is not unique, $g(x,y)$ may be only subdifferentiable, and the subdifferential may be given by equation (2) as follows:

$$\partial g(x) = co\{\partial_x g(x,y) | \text{ for all } y \text{ such that } g(x,y) = g(x)\} \quad (2)$$

where co may be the notation for convex hull and ∂_x may be the partial derivative with respect to x. Computing an element of this set (i.e., computing $\partial g(x)$) then may amount to solving the inner maximization problem in equation (1) (i.e., $\max_{y \in Y_{i_1 i_2}} g_{i_1 i_2}(x,y)$) and differentiating $g(x,y)$ with respect to x at points y that achieve a threshold or maximum value.

[054] In some cases, the objective function or loss function $g(x,y)$ can be any real valued function. In some cases, the objective function or loss function $g(x,y)$ can be any real valued function with the second argument y being a discrete and the first argument x being a discrete vector. In some cases, the objective function or loss function $g(x,y)$ can be any real valued function with the second argument y being a continuous vector and the first argument x being a discrete vector. In some cases, the objective function or loss function $g(x,y)$ can be any real valued function with the second argument y being a continuous vector and the first argument $\underline{x}$ being a continuous vector. In some cases, the objective function or loss function $g(x,y)$ can be any real valued function with the second argument y being a discrete vector and the first argument x being a continuous vector.

[055] In some cases, the objective function or loss function $g(x,y)$ may be linear in its second argument y. In some cases, the objective function or loss function $g(x,y)$ may be quadratic in its second argument y. In the case that the objective function or loss function $g(x,y)$ is quadratic in its second argument y, the inner maximization in equation (1) can be rephrased as a quadratic optimization problem (such as a quadratic optimization problem over binary variables) as in equation (3):

$$\max\{g(x,y) | y \in Y\} = \max\{\langle Q(x)z, z\rangle + \langle c(x), z\rangle \mid z \in \{0,1\}^m\} \quad (3)$$

for some m and symmetric matrix $Q(x)$ and vector $c(x)$ that may depend on x, where the solution y on the left-hand-side expression of equation (3) can be constructed from the solution z on the right-hand side of equation (3). Here the variable z may correspond to an encoding (such as a binary encoding) of variable y. In some embodiments, an encoding as such can be given by a computable mapping $z \rightarrow y$. Thus, a subgradient for g can be obtained by solving a quadratic optimization problem (such as a binary quadratic optimization problem).

[056] Equation (1) can include a function $f(x)$, which may be a function of a continuous vector x wherein the continuous vector may be real-valued. The function $f(x)$ may regularize the optimization of equation (1). The function $f(x)$ may be an optional function and may take a value of zero for one or more possible values that variable x may take. For example, in structured support vector machines (SSVMs), $f(x)$ can be a regularizer, which may help to reduce overfitting.

[057] In equation (1), x may be a first variable or argument that may take its value from the constraint set C, which may include real-values. The constraint set C may include a number of continuous vectors, the number may be any number that is no less than one. Each continuous vector may contain real-values. Optimization of the robust inference problems may include thresholding or minimization only with respect to x that is determined by y, while the vector(s) y can be "internal".

Data of the robust inference problem

[058] The methods for the stochastic optimization of robust inference problems described herein may include obtaining data of the robust inference problem. Such data may be pre-generated manually or automatically from raw data. Such data may be obtained by a digital computer. Such data may be utilized at least in part by the methods described herein for the stochastic optimization of robust inference problems, such as the robust inference problems described herein.

[059] Such data of the robust inference problem may include initial values for one or more parameter(s) and/or argument(s) in equation (1) so that the iterative optimization process can start with the initial values.

[060] Such data of the robust inference problem may include a set of objective functions or loss functions, wherein each objective function or loss function can be expressed as $g_{i_1,i_2}(x,y)$, wherein i_1 and i_2 are independent indices with index $i_1 \in \{1,\ldots,m_1\}$ and $i_2 \in \{1,\ldots,m_2\}$, x can be a continuous vector, and y $\in Y_{i_1 i_2}$ can be a discrete vector. The index i_1 may be a fixed number for one

or more entire iterations of an optimization process selecting a non-overlapping subset of objective functions or loss functions, and it may be selected for other iterations based on predetermined selection procedure. The index i_2 may be an index over the number of objective functions or loss functions in each non-overlapping subset corresponding to index i_1.

[061] The set of objective functions or loss functions may include one or more objective functions or loss functions. The set of objective functions or loss functions may include a plurality of objective functions or loss functions. In the case that only one objective function or loss function is contained in the set, the subset of objective functions or loss functions may be the set of objective functions or loss functions. In cases when the set of objective functions or loss functions includes two or more objective functions or loss functions, the set of objective functions or loss functions may be grouped into non-overlapping subsets. Each of the non-overlapping subsets of objective functions or loss functions may comprise only two objective functions or loss functions. Each subset may contain 1, 2, 3, 4, 5, 6, 10, 20, 30, 50, 60, 70, 80, 90 100, or more objective functions or loss functions. Each subset may contain at least about 1, 2, 3, 4, 5, 6, 10, 20, 30, 40, 50, 60, 70, 80, 90 100, or more objective functions or loss functions. In other cases, each subset may contain at most about 1, 2, 3, 4, 5, 6, 10, 20, 30, 40, 50, 60, 70, 80, 90 100, or less objective functions or loss functions. Among the set of objective functions or loss functions, each objective function or loss function may accept a first argument x and a second argument y. The first and second arguments may be independent arguments. The first and second arguments may be dependent arguments. The first argument may take a continuous vector as its value, and the second argument may take a discrete vector as its value.

[062] Such data of the robust inference problem may include a linear-combination weight for each objective function or loss function: each loss function in equation (1) may be weighted by a weight, which can be a scalar ω_{i_1,i_2} that influences its contribution to the overall sum.

[063] Such data of the robust inference problem may include a set of permissible discrete vectors, $Y_{i_1 i_2}$, for each loss function from which variable y may take values.

[064] Such data of the robust inference problem may include an initial continuous vector for the first argument of all loss functions in the first iteration. Such data of the robust inference problem may include an initial continuous vector for the first argument of one or more loss functions in the first iteration.

[065] The methods of the robust inference problem may set the current values of a set of scaling

parameters in the first iteration of the iterative optimization process to be initial values of the set of scaling parameters. The initial values may be based at least in part on a schedule described herein. In later iterations, the current values of the set of scaling parameters may be updated based at least in part on the schedule, as described herein.

[066] The methods for solving the robust inference problem may receive or generate a schedule of the set of scaling parameters, from which the current values of the set of scaling parameters may be obtained from. The schedule may be determined *a priori* by the user or adjusted automatically by a selected algorithm. Initial values of the set of scaling parameters may be included in the schedule or based at least in part on the schedule. The schedule may be generated based on theoretical or empirical knowledge. The schedule may be generated using one or more algorithms or procedures selected from: a statistical algorithm or procedure, a pattern recognition algorithm or procedure, a machine learning algorithm or procedure, a deep learning algorithm or procedure, an artificial intelligence algorithm or procedure, a neural network, or the like. The schedule may be generated using historical values of the set of scaling parameters. The set of scaling parameters may take values of any real numbers. The set of scaling parameters may take values of any non-zero real numbers. The set of scaling parameters may take values of any positive and/or negative numbers.

[067] The set of scaling parameters may be used in a "softmax function" for solving the robust inference problems herein. The "Softmax function" can approximate the "max" function in equation (1), e.g., $\max_{y \in Y_{i_1,i_2}} g_{i_1,i_2}(x,y)$, with a smooth function as in equation (4):

$$\max_{y \in Y_{i_1 i_2}} g_{i_1,i_2}(x,y) \approx \left(1/\beta\right) \log \sum_{y \in Y_{i_1 i_2}} exp\left(\beta g_{i_1,i_2}(x,y)\right) \quad (4)$$

where x can be a continuous vector, $y \in Y_{i_1 i_2}$ can be a discrete vector such that the discrete vector is selected from a discrete set of permissible states, $Y_{i_1 i_2}$, from which variable y may take values, $g_{i_1,i_2}(x,y)$ can be an objective function or loss function wherein i_1 and i_2 may be dependent or independent indexes with index $i_1 \in \{1,\ldots,m_1\}$ and $i_2 \in \{1,\ldots,m_2\}$, and β may be an element in the set of scaling parameters that obtains value from a schedule of a set of scaling parameters. In equation (4), β may be the only element in the set of scaling parameters.

[068] Those β with higher values may enable better approximation of the "max" function on the left-hand side in equation (4) than lower values. However, higher values can slow down the generation of the samples as a tradeoff. If the optimization starts with a relatively low β and increases it gradually, the optimization problem may be solved in fewer iterative steps than in the

case of starting with a set of scaling parameters with higher values.

[069] "Smooth-max" approximation, equivalent herein to a softmax approximation, as in equation (4), has the following beneficiary properties.

[070] A max function $h(z) = \max\{z_1, \dots, z_n\}$ can be approximated by the softmax function of equation (5):

$$h_\mu(z) = \mu \log \sum_{i=1}^{n} \exp(\frac{z_i}{\mu})$$

(5)

where μ may be a positive parameter equivalent to $1/\beta$.

[071] The max function $h(z) = \max\{z_1, \dots, z_n\}$ may be composed with a set of functions (such as a set of nonlinear functions), and the max function and its smooth approximation may be considered together as in equations (4.1a) and (4.1b):

$$h(x) = \max_{i=1:m} g_i(x), \tag{4.1a}$$

$$h_\mu(x) = \mu \log \sum_{i=1}^{m} \exp\left(\frac{g_i(x)}{\mu}\right). \tag{4.1b}$$

[072] The properties of this smooth approximation as in equation (4.1b) may depend on the smoothness properties of the component functions g_i. In the context of equation (1), we may index each element in Y by $i = 1{:}\,m$, where $m = |Y|$, and so $g_i(x) \coloneqq g(x, y_i)$, where $y_i \in Y$.

[073] Referring to the objective functions or loss functions $g_{i_1,i_2}(x, y)$ indexed by $i_1 \in \{1, \dots, m_1\}$ and $i_2 \in \{1, \dots, m_2\}$, which are to be maximized, the softmax described herein can be applied to each of the objective functions or loss functions $g_{i_1,i_2}(x, y)$ separately with $y \in Y_{i_1 i_2}$.

[074] The functions, $g_i: \mathbb{R}^n \to X$, with $X \subseteq \mathbb{R}$ compact, may be convex and may have Lipschitz continuous gradients with constant L_i, and bounded norm $M_i \coloneqq \max\{\|\nabla g_i(x)\|^2 | x \in X\}$, with $i = 1{:}\,m$. Let $L \coloneqq \max_i L_i$, and $M \coloneqq \max_i M_i$. Then the functions described in equations (4.1a) and (4.1b) may satisfy one or more of the following three statements:

1. $h(x) + \mu \log(|\arg\max\{g_i(x)\}|) \leq h_\mu(x) \leq h(x) + \mu \log(m)$;
2. $h(x) - c_1\mu \leq h_\mu(x) \leq h(x) + c_2\mu \; \forall \mu > 0$, where c_i may be constants such that $c_1 + c_2 = \log(m)$; and
3. $\|\nabla h_\mu(x) - \nabla h_\mu(y)\| \leq [\frac{M}{\mu} + L]\|x - y\|$.

[075] The norms that appear in these statements may be 2-norm, even though the Lipschitz

constant of the uncomposed smooth approximation may be stated in the infinity norm. The gradient of the smooth approximation given in equation (4.1b) may be given by:

$$\nabla h_\mu(x) = \sum_{i=1}^{m} p_i^\mu(x) \nabla g_i(x) \text{ with } p_i^\mu(x) = \frac{\exp\left(\frac{g_i(x)}{\mu}\right)}{\sum_{i=1}^{m} \exp\left(\frac{g_i(x)}{\mu}\right)}, \quad (4.2)$$

where $\sum_{i=1}^{m} p_i^\mu(x) = 1$, and $p_i^\mu(x) \geq 0$. The gradient of the smooth approximation h_μ may be obtained as an average or weighted average of the gradients. Thus, the gradient of the approximation h_μ may be obtained as an expected value where i may be a random variable. The gradient of the approximation h_μ may be obtained as an expected value where i may be a random variable that follows a Boltzmann distribution given by equation (5):

$$i = \frac{\exp\left(\frac{g_i(x)}{\mu}\right)}{\sum_{i=1}^{m} \exp\left(\frac{g_i(x)}{\mu}\right)} \quad (5)$$

where $\frac{1}{\mu}$ may be the only element in a set of scaling parameters.

Iterative optimization process

[076] The methods for stochastic optimization of robust inference problems may include iteratively performing one or more steps in each iteration of the iterative optimization process until at least one stopping criterion is met. Such stopping criterion may comprise a set of rules containing one or more rules for determining one or more of an accuracy, sensitivity, or specificity of a solution to the robust inference problem. The stopping criterion may be based at least in part on a magnitude of a distance between the current continuous vector in one iteration of the optimization process and the updated current continuous vectors in the same iteration or a different iteration, e.g., a previous or subsequent iteration.

[077] In each iteration, the one or more steps in the iterative optimization process may include determining current values of the set of scaling parameters. The current values may be based at least in part on the schedule of the set of scaling parameters.

[078] In each iteration, the one or more steps in the iterative optimization process may include selecting a subset of the objective functions or loss functions from the non-overlapping subsets, either non-repetitively or repetitively.

[079] In each iteration, one or more sub-steps may be performed for each objective function or loss function of the selected subset of the objective functions or loss functions. The one or more sub-steps may include generating one or more samples of discrete vectors for variable or argument y in

equation (1). Each sample of the one or more samples may be selected from the set of permissible discrete vectors associated with the specific objective function or loss function. Each sample of the one or more samples may be generated based on a probability distribution. In some cases, the probability distribution may be determined at least in part by the set of scaling parameters and the specific objective function or loss function. The first argument of the loss function may take the current value of the continuous vector in the iteration. For instances, each sample may be generated according the probability distribution in equation (6):

$$p_x^{(i_1,i_2)}(y) = \frac{\exp\left(\beta g_{i_1,i_2}(x,y)\right)}{\sum_{y \in Y_{i_1,i_2}} \exp\left(\beta g_{i_1,i_2}(x,y)\right)} \quad (6)$$

where x may be held as a fixed continuous vector for all the samples in each iteration and β may be a scaling parameter. Each of the one or more samples may be generated using the sampling device disclosed herein. For example, a number of k samples may be generated and each sample may be selected from the set of permissible states so that k samples $(\mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{k}) \in Y_{i_1 i_2}$, and wherein a choice of $i_1 \in \{1, \ldots, m_1\}$ may represent the selected subset of the objective functions or loss functions, and $i_2 \in \{1, \ldots, m_2\}$ may represent the function in the selected subset. The probability distributions of the samples may be any single probability distribution or any combination of different probability distributions.

[080] The sampling device herein may include a random or pseudo-random generator that produces samples distributed according to a Boltzmann model. Such a sampling device may include hardware (e.g., a specialized computing device, a quantum processor, a non-classical computer, a quantum computing system, a digital computer, a digital processing device, or the like) and/or software that is configured to perform "Boltzmann sampling." The approximated gradient then can be used to solve the robust inference problem with a pre-selected level of accuracy. The utilization of the sampling device and the connection of the sampling device may advantageously connect the gradient of the smoothed function approximation to a Boltzmann distribution, so that a complex robust inference problem can be solved. The sampling device may exhibit one or more properties determined by the mathematical definition of a Boltzmann distribution given in equation (6). The sampling device may include any hardware, software, or combination of hardware and software that may be configured to exhibit one or more properties determined by the mathematical definition of a Boltzmann distribution given in equation (6). In some cases, the normalized frequencies of observing different configurations fall within a selected distance from the mathematically defined probabilities given in

equation (6) of the respective configurations. The one or more samples may be of the discrete vectors and/or may follow a Boltzmann distribution.

[081] The systems for solving a robust inference problem may include a sampling device for generating a number of samples. The sampling device may comprise a quantum processor and a quantum device control system for obtaining the schedule of the set of scaling parameters, the data of the robust inference problem, or their combination. The quantum processor may be coupled to a digital computer and to the quantum device control system. The quantum processor can comprise a plurality of qubits and a plurality of couplers, each coupler of the plurality of couplers for providing a communicative coupling at a crossing of two qubits of the plurality of qubits. The digital computer may be remotely located with respect to the sampling device.

[082] The quantum processor or quantum computer may comprise one or more adiabatic quantum computers, quantum gate arrays, one-way quantum computers, topological quantum computers, quantum Turing machines, superconductor-based quantum computers, trapped ion quantum computers, trapped atom quantum computers, optical lattices, quantum dot computers, spin-based quantum computers, spatial-based quantum computers, Loss-DiVincenzo quantum computers, nuclear magnetic resonance (NMR) based quantum computers, solution-state NMR quantum computers, solid-state NMR quantum computers, solid-state NMR Kane quantum computers, electrons-on-helium quantum computers, cavity-quantum-electrodynamics based quantum computers, molecular magnet quantum computers, fullerene-based quantum computers, linear optical quantum computers, diamond-based quantum computers, nitrogen vacancy (NV) diamond-based quantum computers, Bose–Einstein condensate-based quantum computers, transistor-based quantum computers, and rare-earth-metal-ion-doped inorganic crystal based quantum computers. The quantum processor or quantum computer may comprise one or more of: quantum annealers, Ising solvers, optical parametric oscillators (OPO), and gate models of quantum computing.

[083] The quantum processor or quantum computer may comprise one or more qubits. The one or more qubits may comprise superconducting qubits, trapped ion qubits, trapped atom qubits, photon qubits, quantum dot qubits, electron spin-based qubits, nuclear spin-based qubits, molecular magnet qubits, fullerene-based qubits, diamond-based qubits, nitrogen vacancy (NV) diamond-based qubits, Bose-Einstein condensate-based qubits, transistor-based qubits, or rare-earth-metal-ion-doped inorganic crystal based qubits.

[084] The sampling device may comprise a network of optical parametric oscillators, in which the network includes an optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of the plurality of optical parametric oscillators. The sampling device may include a network of optical parametric oscillators simulating two-body, three-body, or many-body interactions via interference of the optical pulses relevant to a reference phase. The sampling device may include one or more physical system with tunable and/or controllable many-body interactions that can stay close to its thermal equilibrium or approach its steady states.

[085] The systems for solving a robust inference problem may include a digital computer, or use of the same. The sampling device may include a digital computer, a central processing unit and a memory unit coupled to the central processing unit. The sampling device may include an application, a software module, a computer program, a user console, or use of the same, for obtaining the schedule of the scaling parameter, the data of the robust inference problem, or a combination thereof. The application, software module, or use of the same may be adapted for performing a Monte Carlo based algorithm. The Monte Carlo based algorithm may include Simulated Annealing, Simulated Quantum Annealing, Gibbs Sampling, or any combination thereof.

[086] The sampling device may include a reconfigurable digital hardware, a central processing unit and a memory unit with the central processing unit and the memory unit coupled to the reconfigurable digital hardware. The reconfigurable digital hardware may be adapted for obtaining the schedule of the scaling parameter, the data of the robust inference problem or a combination thereof. The reconfigurable digital hardware may be adapted to perform a Monte Carlo based algorithm. The Monte Carlo based algorithm may include Simulated Annealing, Simulated Quantum Annealing, Gibbs Sampling, or any combination thereof.

[087] Devices and systems for generating approximations of the Boltzmann distribution at one or more given or user-specified scaling parameter(s) can be used as a sampling device herein. The sampling device herein may be devices or systems that can utilize Simulated Annealing, Monte Carlo, and/or quantum Monte Carlo methods. The sampling device may include implemented algorithm(s) on a processor, a digital processing device, a digital computer, a CPU or any other customized hardware such as an field-programmable gate array (FPGA), a graphics processing unit (GPU), an application-specific integrated circuit (ASIC), or a combination thereof. The sampling

device may include a quantum computing system based on quantum circuits, a computing device that carries physical and/or approximate realizations of quantum annealing or quantum adiabatic computation, or their combination.

[088] The stochastic optimization of the robust inference problem herein may be associated with training a structured support vector machine (SSVM). The stochastic optimization of the robust inference problem may be associated with image segmentation, image tagging and/or recommendation system. The stochastic optimization of the robust inference problem may be associated with a dual of the basis pursuit problem from compressed sensing. The stochastic optimization of the robust inference problem may be associated with unsupervised learning, semi-supervised learning, supervised learning, and/or active learning.

[089] The one or more sub-steps may include obtaining a gradient of the objective function or loss function taken with respect to the first argument x, wherein the first argument of the objective function or loss function may take the current values of the continuous vector, and the second argument y of the objective function or loss function can take value of a selected sample. The k samples $(y_1, \ldots, y_k) \in y$ may be generated using a sampling device according to the probabilities using equation (6), where x may be held fixed. For each sample y_j, the index j can be in the range from 1 to k, and the gradient of the function $g_{i_1,i_2}(x, y_j)$ may be evaluated with respect to the continuous variables x evaluated at their current value. For that sample, y_j may be generated using equation (7):

$$grad_j^{(i_1,i_2)} := \nabla_x g_{i_1,i_2}(x, y_j), \forall j \in \{1, \ldots, k\} \quad (7)$$

[090] For example, if there may be a total number of k samples (k may be any integer number) for a selected objective function or loss function, k gradients can be generated with each gradient for one of the k samples. Each gradient may be obtained with the first argument x taking the same current continuous vector and the second argument y of the objective function or loss function taking values of a selected sample. The gradient may be obtained using a digital computer using the samples generated by a sampling device. The sampling device may comprise a digital computer, a quantum computer, or any other digital processing device and/or devices. The other digital processing devices may include but are not limited to: a hybrid computer including at least a digital computer and a quantum computer.

[091] The one or more sub-steps may include obtaining an average of the one or more gradients obtained in equation (7) using equation (8):

$$grad^{(i_1,i_2)} := \frac{1}{k}\sum_{j=1}^{k} grad_j^{(i_1,i_2)} \quad (8)$$

[092] For example, if there are k gradients obtained for k samples, an average of the k gradients may be obtained. k may be an integer greater than one. If k equals one, the average of the one or more gradients may be equal to the single gradient.

[093] In each iteration, the one or more steps in the iterative optimization process may include obtaining a summation and/or a partial summation of the averages of the one or more gradients, wherein the summation may be for all objective functions or loss functions in the selected subset of the objective functions or loss functions, and the partial summation may be for more than one objective functions or loss functions in the selected subset of the objective functions or loss functions. The summation and/or partial summation may be a linear combination of the gradient averages as in equation (9):

$$d^{(i_1)} := \sum_{i_2=1}^{m_2} \omega_{i_1,i_2} grad_j^{(i_1 i_2)} \quad (9)$$

[094] For example, a selected subset of objective functions or loss functions may contain four objective functions or loss functions; and for each objective function or loss function, an average of gradients may be obtained. The summation herein may include adding up 4 different averages of gradients multiplied by its associated weight, while the partial summation herein may include adding up any 2 or 3 different averages of gradients multiplied by its associated weight. If there is only one objective function or loss function in the selected subset, the sum may be the average of gradients multiplied by its associated weight for the one objective function or loss function. A selected subset of objective functions or loss functions may contain at least about 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, or more objective functions or loss functions. A selected subset of objective functions or loss functions may contain at most about 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 9, 8, 7, 6, 5, 4, 3, 2, or 1 objective functions or loss functions. A selected subset of objective functions or loss functions may contain a number of objective functions or loss functions that is within a range defined by any two of the preceding values.

[095] In each iteration, the one or more steps in the iterative optimization process may further include computing a search direction based at least in part on any one or more of: v1) the summation or partial summation of the averages of the one or more gradients (e.g., weighted summation or

partial summation), v2) the current values of the set of scaling parameters, v3) at least part of a history of the summation or partial summation of the averages of the one or more gradients, and/or v4) at least part of a history of the values of the set of scaling parameters. The history of a specific element, e.g., the summation of the averages of the gradients herein can include current values of the specific element (for currently selected subset of the objective functions or loss functions) and/or previous values of the specific element (for previously selected subsets of the objective functions or loss functions). This search direction, or equivalently, descent direction $-d$, may depend on current $d^{(i_1)}$and may additionally depend on previous calculations of $d^{(i_1)}$ for previous choices of i_1.

[096] The one or more steps in the iterative optimization process may further include computing a step length α that ensures that the direction $-d$ makes progress towards optimizing equation (1) or that tends to increase the probability that the direction $-d$ makes progress towards optimizing equation (1). The step length may be computed based at least in part on one or more of: vi1) the current values of the set of scaling parameters, vi2) the selected subset of loss functions, vi3) at least part of a history of values of the set of scaling parameters, and/or vi4) at least part of a history of the selected subset of loss functions. The history of the selected subset of objective function or loss function herein can include current selected subset of objective functions or loss functions and/or one or more previously selected subsets of the objective functions or loss functions. The one or more steps in the iterative optimization process may further include computing an updated current continuous vector using the step length and the search direction, setting the current value of the continuous vector to be the updated current continuous vector. The update may be given by equation (10):

$$x \leftarrow \Pi_C(x - \alpha d) \tag{10}$$

where Π_C denotes projection on the constraint set C. The sequence of updates may converge to an approximate solution of equation (1).

[097] **Fig. 1** shows a flowchart for a non-limiting example of a method 100 for stochastic optimization of a robust inference problem using a sampling device. In a first operation 102, the method 100 may comprise receiving (for instance, by a digital computer data of the robust inference problem. The data may comprise a set of objective functions or loss functions grouped into non-overlapping subsets. Each objective function or loss function in the set of loss functions may accept first and second arguments. The data may further comprise: a set of permissible vectors for each objective function or loss function in the set of objective functions or loss

functions.

[098] In a second operation 104, the method 100 may comprise setting (for instance, by the digital computer), a current value of a continuous vector.

[099] In a third operation 106, the method 100 may comprise receiving (for instance, by the digital computer), a schedule of a set of scaling parameters.

[0100] In a fourth operation 108, the method 100 may comprise determining current values of the set of scaling parameters based at least in part on the schedule.

[0101] In a fifth operation 110, the method 100 may comprise selecting a subset of the objective functions or loss functions from the non-overlapping subsets.

[0102] In a sixth operation 112, the method 100 may comprise iterating a series of steps to obtain one or more gradients for each objective function or loss function of the objective functions or loss functions. The series of steps may comprise generating, by the sampling device, one or more samples of vectors from a set of permissible vectors associated with the objective function or loss function. The series of steps may comprise obtaining (for instance, by the digital computer) one or more gradients. Each of the one or more gradients may be of the objective function or loss function taken with respect to the first argument. The series of steps may comprise obtaining (for instance, by the digital computer) an average of the one or more gradients.

[0103] In a seventh operation 114, the method 100 may comprise obtaining (for instance, by the digital computer) a summation or partial summation of the averages of the one or more gradients. The summation may be for all objective functions or loss functions in the selected subset of objective functions or loss functions. The partial summation may be for more than one objective function or loss function in the selected subset of objective functions or loss functions.

[0104] In an eighth operation 116, the method 100 may comprise computing (for instance, by the digital computer) a search direction. The search direction may be based at least in part on one or more of: v1) the summation or the partial summation of the averages of the one or more gradients; v2) the current values of the set of scaling parameters; v3) at least part of a history of the summation or partial summation of the averages of the one or more gradients; and v4) at least part of a history of the values of the set of scaling parameters.

[0105] In a ninth operation 118, the method 100 may comprise computing (for instance, by the digital computer) a step length. The step length may be based at least in part on one or more of: vi1) the current values of the set of scaling parameters; vi2) the selected subset of the objective

functions or loss functions; vi3) at least part of a history of values of the set of scaling parameters; and vi4) at least part of a history of the selected subset of the objective functions or loss functions.

[0106] In a tenth operation 120, the method 100 may comprise setting (for instance, by the digital computer) the current value of the continuous vector based on the step length and the search direction.

[0107] In an eleventh operation 122, the method 100 may comprise providing the current value of the continuous vector.

[0108] Any 1, 2, 3, 4, 5, 6, 7, or 8 of the fourth operation 108, fifth operation 110, sixth operation 112, seventh operation 114, eighth operation 116, ninth operation 118, tenth operation 120, and eleventh operation 122 may be repeated until a stopping criterion is met. The stopping criterion may be any stopping criterion described herein.

[0109] The objective functions or loss functions may comprise one or more composite functions of the first and second arguments. Obtaining (for instance by the digital computer) one or more gradients, wherein each of the one or more gradients may be of the objective function or loss function taken with respect to the first argument may comprise iterative applications of a chain rule. The iterative applications of the chain rule may be performed using auto-differentiation. One or more argument functions of the composite functions may comprise differentiable feature extractors. The differentiable feature extractors may comprise deep neural networks.

[0110] Computing (for instance, by the digital computer) a search direction may comprise using one or more of stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), and/or stochastic dual coordinate ascent (SDCA).

[0111] Computing (for instance, by the digital computer) the step length may comprise using one or more adaptive gradient descent methods. The adaptive gradient descent methods may comprise Adaptive Moment Estimation (Adam), reduced mean square (RMS), Root Mean Square Propagation (RMSProp), and/or adaptive gradient algorithm (AdaGrad).

[0112] The sampling device may comprise a quantum processor and a quantum device control system for obtaining the schedule of the set of scaling parameters and the data of the robust inference problem. The quantum processor may be coupled to the digital computer and to the quantum device control system. The quantum processor may comprise any quantum processor or quantum computer described herein. The quantum processor may comprise a plurality of

qubits and a plurality of couplers. Each coupler of the plurality of couplers may be for providing a communicative coupling at a crossing of two qubits of the plurality of qubits. The one or more samples of vectors may follow a Boltzmann distribution. The sample device may be a network of optical parametric oscillators. The network may comprise an optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators and a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of the plurality of optical parametric oscillators. The sampling device may comprise a central processing unit and a memory unit coupled to the central processing unit. The memory unit may comprise an application for obtaining the schedule of the scaling parameter and the data of the robust inference problem, and the application may be configured to implement a Markov Chain Monte Carlo algorithm. The sampling device may comprise a reconfigurable digital hardware, a central processing unit and a memory unit. The central processing unit and the memory unit may be coupled to the reconfigurable digital hardware. The reconfigurable digital hardware may be configured to obtain the schedule of the scaling parameter and the data of the robust inference problem, and the reconfigurable digital hardware may be configured to implement a Markov Chain Monte Carlo algorithm. The Markov Chain Monte Carlo algorithm may comprise simulated quantum annealing. The Markov Chain Monte Carlo algorithm may comprise simulated annealing. The Markov Chain Monte Carlo algorithm may comprise Gibbs sampling.

[0113] The set of objective functions or loss functions may comprise one or more objective functions or loss functions. The subset of the non-overlapping subsets of objective functions or loss functions may comprise only two objective functions or loss functions.

[0114] The stochastic optimization of the robust inference problem may be associated with training a structured support vector machine. The data of the robust inference problem may be associated with an image segmentation problem. The data of the robust inference problem may be associated with a dual of a basis pursuit problem from a compressed sensing problem. The data of the robust inference problem may be associated with semi-supervised learning. The data of the robust inference problem may be obtained from a noun phrase co-reference resolution problem. The data of the robust inference problem may be associated with active learning. The data of the robust inference problem may be associated with an image tagging problem. The data of the robust inference problem may be associated with a recommender system.

[0115] The schedule of the set of scaling parameters may be determined by a user or automatically by an algorithm.

[0116] The stopping criterion may be based at least in part on a magnitude of a distance between the current and the updated current vectors. The first and second arguments may be independent, and the first argument may employ a continuous vector as its value, the second argument may employ a discrete vector as its value, and the set of permissible vectors may comprise a set of permissible discrete vectors.

[0117] Many variations, alterations, and adaptations based on method 100 provided herein are possible. For example, the order of the operations of the method 100 may be changed, some of the operations removed, some of the operations duplicated, and additional operations added as appropriate. Some of the operations may be performed in succession. Some of the operations may be performed in parallel. Some of the operations may be performed once. Some of the operations may be performed more than once. Some of the operations may comprise sub-operations. Some of the operations may be automated and some of the operations may be manual.

[0118] **Fig. 2** schematically illustrates a non-limiting example of a system 200 for stochastic optimization of a robust inference problem using a sample device. The system may comprise a digital computer interacting with a quantum computing system. The system 200 can comprise a digital computer 202 and non-classical computing system, which may be a quantum computing system 204. The system 200 may implement the method 100 of **Fig. 1**. The system 200 may be, for example, as described in U.S. Patent Publication Nos. 2017/0357539 and 2018/0091440, each of which is entirely incorporated herein by reference. The quantum computing system 204 may include one or more superconducting qubits. The quantum computing system may comprise any quantum computer or quantum processor described herein. The quantum computing system may comprise any quantum computing qubits described herein. The digital computer 202 may communicate (e.g., via direct communication or over a network) with the quantum computing system 204 by transmitting and/or receiving data therefrom. The digital computer and the qubits may be remotely located from each other. The digital computer and the qubits may be remotely locally to one another. In some embodiments, the digital computer 202 may be any type. The digital computer 202 may be a desktop computer, laptop computer, tablet personal computer, server, or smartphone. The digital computer 202 may comprise a central processing unit (CPU) 302, also referred to as a microprocessor, a display device 304, input devices 306, communication

ports 308, a data bus 310, a memory unit 312 and a network interface card (NIC) 322. The CPU 302 may be a single core or multi-core processor. The digital computer 202 may include a plurality of processors for parallel processing.

[0119] The display device 304 can include a user interface (UI). Examples of UI's include, without limitation, a graphical user interface (GUI) and web-based user interface.

[0120] The CPU 302 may be used for processing computer instructions. Various embodiments of the CPU 302 may be provided. The central processing unit 302 may be a CPU Core i7-3820 running at 3.6 GHz and manufactured by Intel(TM), for example.

[0121] The display device 304 can be used for displaying data to a user. The skilled addressee will appreciate that various types of display device 304 may be used. The display device 304 may be a liquid-crystal display (LCD) monitor. The display device 304 may have a touchscreen, such as, for example, a capacitive or resistive touchscreen.

[0122] The communication ports 308 may be used for sharing data with the digital computer 202. The communication ports 308 may comprise, for instance, a universal serial bus (USB) port for connecting a keyboard and a mouse to the digital computer 202. The communication ports 308 may further comprise a data network communication port such as an IEEE 802.3 port for enabling a connection of the digital computer 202 with another computer via a data network. The skilled addressee will appreciate that various alternative embodiments of the communication ports 308 may be provided. The communication ports 308 may comprise an Ethernet port and a mouse port.

[0123] The memory unit 312 may be used for storing computer-executable instructions. The memory unit 312 may comprise an operating system module 314. The operating system module 314 may be of various types. In some embodiments, the operating system module 314 may be OS X Yosemite manufactured by Apple(TM).

[0124] The memory unit 312 can further comprise one or more applications. One or more of the central processing unit 302, the display device 304, the input devices 306, the communication ports 308 and the memory unit 312 may be interconnected via the data bus 310.

[0125] The system 202 may further comprise a network interface card (NIC) 322. The application 320 can send the appropriate signals along the data bus 310 into NIC 322. NIC 322, in turn, may send such information to quantum device control system 324.

[0126] The quantum computing system 204 may comprise a plurality of quantum bits and a

plurality of coupling devices. Further description of the quantum computing system 204 is disclosed in, for example, U.S. Patent Publication No. 2006/0225165, which is entirely incorporated herein by reference.

[0127] The quantum computing system 204 of the quantum computing device can further comprise a quantum device control system 324 and a quantum processor or a quantum computer 330. The control system 324 may comprise coupling controller for each coupling in the plurality 328 of couplings of the quantum computing system 204 capable of tuning the coupling strengths of a corresponding coupling, and local field bias controller for each qubit in the plurality 326 of qubits of the quantum computing system 204 capable of setting a local field bias on each qubit.

[0128] Methods described herein can be implemented by way of machine (e.g., computer processor) executable code stored on an electronic storage location of the computer system 200, such as, for example, on the memory unit 312 or an electronic storage unit. The machine executable or machine readable code can be provided in the form of software. During use, the code can be executed by the CPU 302. In some cases, the code can be retrieved from the electronic storage unit and stored on the memory unit 312 for ready access by the CPU 302. In some situations, the electronic storage unit can be precluded, and machine-executable instructions are stored on memory unit 312.

[0129] The code can be pre-compiled and configured for use with a machine having a processer adapted to execute the code, or can be compiled during runtime. The code can be supplied in a programming language that can be selected to enable the code to execute in a pre-compiled or as-compiled fashion.

[0130] Aspects of the systems and methods provided herein, such as the computer system 1101, can be embodied in programming. Various aspects of the technology may be thought of as "products" or "articles of manufacture" typically in the form of machine (or processor) executable code and/or associated data that is carried on or embodied in a type of machine readable medium. Machine-executable code can be stored on an electronic storage unit, such as memory (e.g., read-only memory, random-access memory, flash memory) or a hard disk. "Storage" type media can include any or all of the tangible memory of the computers, processors or the like, or associated modules thereof, such as various semiconductor memories, tape drives, disk drives and the like, which may provide non-transitory storage at any time for the software programming. All or portions of the software may at times be communicated through the Internet

or various other telecommunication networks. Such communications, for example, may enable loading of the software from one computer or processor into another, for example, from a management server or host computer into the computer platform of an application server. Thus, another type of media that may bear the software elements includes optical, electrical and electromagnetic waves, such as used across physical interfaces between local devices, through wired electrical and/on optical landline networks and/or over various air-links. The physical elements that carry such waves, such as wired or wireless links, electrical links, optical links, or the like, also may be considered as media bearing the software. As used herein, unless restricted to non-transitory, tangible "storage" media, terms such as computer or machine "readable medium" refer to any medium that participates in providing instructions to a processor for execution.

[0131] Hence, a machine readable medium, such as computer-executable code, may take many forms, including but not limited to, a tangible storage medium, a carrier wave medium or physical transmission medium. Non-volatile storage media include, for example, optical or magnetic disks, such as any of the storage devices in any computer(s) or the like, such as may be used to implement the databases, etc. shown in the drawings. Volatile storage media include dynamic memory, such as main memory of such a computer platform. Tangible transmission media include coaxial cables; copper wire and fiber optics, including the wires that comprise a bus within a computer system. Carrier-wave transmission media may take the form of electric or electromagnetic signals, or acoustic or light waves such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of computer-readable media therefore include for example: a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD or DVD-ROM, any other optical medium, punch cards paper tape, any other physical storage medium with patterns of holes, a RAM, a ROM, a PROM and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave transporting data or instructions, cables or links transporting such a carrier wave, or any other medium from which a computer may read programming code and/or data. Many of these forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to a processor for execution.

[0132] As used in this specification and the claims, unless otherwise stated, the term "about," "substantially," and "approximately" refers to variations of less than or equal to +/- 1%, +/- 2%, +/-

3%, +/- 4%, +/- 5%, +/- 6%, +/- 7%, +/- 8%, +/- 9%, +/- 10%, +/- 11%, +/- 12%, +/- 14%, +/- 15%, or +/- 20% of the numerical value depending on the embodiment. As a non-limiting example, about 100 meters represents a range of 95 meters to 105 meters (which is +/- 5% of 100 meters), 90 meters to 110 meters (which is +/- 10% of 100 meters), or 85 meters to 115 meters (which is +/- 15% of 100 meters) depending on the embodiments.

[0133] Methods and systems of the present disclosure may be combined with or modified by other methods and systems, such as those described in U.S. Patent Publication Nos. 2017/0357539 and 2018/0091440, each of which is entirely incorporated herein by reference.

[0134] While preferred embodiments of the present invention have been shown and described herein, it will be obvious to those skilled in the art that such embodiments are provided by way of example only. It is not intended that the invention be limited by the specific examples provided within the specification. While the invention has been described with reference to the aforementioned specification, the descriptions and illustrations of the embodiments herein are not meant to be construed in a limiting sense. Numerous variations, changes, and substitutions will now occur to those skilled in the art without departing from the invention. Furthermore, it shall be understood that all aspects of the invention are not limited to the specific depictions, configurations or relative proportions set forth herein which depend upon a variety of conditions and variables. It should be understood that various alternatives to the embodiments of the invention described herein may be employed in practicing the invention. It is therefore contemplated that the invention shall also cover any such alternatives, modifications, variations or equivalents. It is intended that the following claims define the scope of the invention and that methods and structures within the scope of these claims and their equivalents be covered thereby.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method for stochastic optimization of a robust inference problem using a sampling device, comprising:

a) receiving, by a digital computer, data of said robust inference problem, wherein said data comprises:

i) a set of objective functions or loss functions grouped into non-overlapping subsets, wherein each objective function or loss function in said set of objective functions or loss functions accepts first and second arguments;

ii) a set of permissible discrete vectors for each objective function or loss function in said set of said objective functions or loss functions;

b) setting, by said digital computer, a current value of a continuous vector;

c) receiving, by said digital computer, a schedule of a set of scaling parameters; and

d) until a stopping criterion is met:

i) determining current values of said set of scaling parameters based at least in part on said schedule;

ii) selecting a subset of said objective functions or loss functions from said non-overlapping subsets;

iii) iterating the following steps for each objective function or loss function of said selected subset of said objective functions or loss functions:

1) generating, by said sampling device, one or more samples of discrete vectors from said set of permissible discrete vectors associated with said objective function or loss function;

2) obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument; and

3) obtaining, by said digital computer, an average of said one or more gradients;

iv) obtaining, by said digital computer, a summation or a partial summation of said averages of said one or more gradients, wherein said summation is for all objective functions or loss functions in said selected subset of said objective functions or loss functions, and wherein said partial summation is for more than one objective function or loss function in said selected subset of said objective functions or loss functions;

v) computing, by said digital computer, a search direction based at least in part on one or more of: v1) said summation or said partial summation of said averages of said one or more gradients; v2) said current values of said set of scaling parameters; v3) at least part of a history of said summation or partial summation of said averages of said one or more gradients; and v4) at least part of a history of said values of said set of scaling parameters;

vi) computing, by said digital computer, a step length based at least in part on one or more of: vi1) said current values of said set of scaling parameters; vi2) said selected subset of said objective functions or loss functions; vi3) at least part of a history of values of said set of scaling parameters; and vi4) at least part of a history of said selected subset of said objective functions or loss functions;

vii) setting, by said digital computer, said current value of said continuous vector based on said step length and said search direction;

viii) and providing said current value of said continuous vector.

2. The method of claim 1, wherein said objective functions or loss functions comprise one or more composite functions of said first and second arguments.

3. The method of claim 2, wherein obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument comprises iterative applications of a chain rule.

4. The method of claim 3, wherein said iterative applications of said chain rule is performed using auto-differentiation.

5. The method of claim 2, wherein one or more argument functions of said composite functions comprises differentiable feature extractors.

6. The method of claim 5, wherein said differentiable feature extractors comprise deep neural networks.

7. The method of claim 1, wherein computing, by said digital computer, a search direction comprises using one or more of stochastic gradient descent (SGD), stochastic average gradient methods (SAG and SAGA), stochastic variance-reduced gradient (SVRG), or stochastic dual coordinate ascent (SDCA).

8. The method of claim 1, wherein computing, by said digital computer, a step length comprises using one or more of said adaptive gradient descent methods, and wherein said adaptive gradient descent methods comprises Adaptive Moment Estimation (Adam), reduced mean square (RMS), Root Mean Square Propagation (RMSProp), and/or adaptive gradient algorithm (AdaGrad).

9. The method of claim 1, wherein said sampling device comprises a quantum processor and a quantum device control system for obtaining said schedule of said set of scaling parameters and said data of said robust inference problem.

10. The method of claim 9, wherein said quantum processor is coupled to said digital computer and to said quantum device control system.

11. The method of claim 10, wherein said quantum processor comprises a plurality of qubits and a plurality of couplers, each coupler of said plurality of couplers for providing a communicative coupling at a crossing of two qubits of said plurality of qubits.

12. The method of claim 11, wherein said one or more samples of discrete vectors follow a Boltzmann distribution.

13. The method of claim 12, wherein said sampling device is a network of optical parametric oscillators, said network comprising:

a) an optical device, said optical device configured to receive energy from an optical energy source and generate a plurality of optical parametric oscillators; and

b) a plurality of coupling devices, each of which controllably couples an optical parametric oscillator of said plurality of optical parametric oscillators.

14. The method of claim 1, wherein said sampling device comprises a central processing unit and a memory unit coupled to said central processing unit.

15. The method of claim 14, wherein said memory unit comprises an application for obtaining said schedule of said scaling parameter and said data of said robust inference problem, and wherein said application is configured to implement a Markov Chain Monte Carlo algorithm.

16. The method of claim 1, wherein said sampling device comprises a reconfigurable digital hardware, a central processing unit and a memory unit, said central processing unit and said memory unit coupled to said reconfigurable digital hardware.

17. The method of claim 16, wherein said reconfigurable digital hardware is configured to obtain said schedule of said scaling parameter and said data of said robust inference problem, and wherein said reconfigurable digital hardware is configured to implement a Markov Chain Monte Carlo algorithm.

18. The method of claim 15 or 17, wherein said Markov Chain Monte Carlo algorithm comprises simulated quantum annealing.

19. The method of claim 15 or 17, wherein said Markov Chain Monte Carlo algorithm comprises simulated annealing.

20. The method of claim 15 or 17, wherein said Markov Chain Monte Carlo algorithm comprises Gibbs sampling.

21. The method of claim 1, wherein said set of objective functions or loss functions comprises one or more objective functions or loss functions.

22. The method of claim 1, wherein said stochastic optimization of said robust inference problem is associated with training a structured support vector machine.

23. The method of claim 1, wherein each subset of said non-overlapping subsets of objective functions or loss functions comprises only two objective functions or loss functions.

24. The method of claim 1, wherein said data of said robust inference problem is associated with an image segmentation problem.

25. The method of claim 1, wherein said data of said robust inference problem is associated with a dual of said basis pursuit problem from a compressed sensing problem.

26. The method of claim 1, wherein said data of said robust inference problem is associated with semi-supervised learning.

27. The method of claim 1, wherein said data of said robust inference problem is obtained from a noun phrase co-reference resolution problem.

28. The method of claim 1, wherein said data of said robust inference problem is associated with active learning.

29. The method of claim 1, wherein said data of said robust inference problem is associated with an image tagging problem.

30. The method of claim 1, wherein said data of said robust inference problem is associated with a recommender system.

31. The method of claim 1, wherein said schedule of said set of scaling parameters is determined by a user or automatically by an algorithm.

32. The method of claim 1, wherein said digital computer is remotely located with respect to said sampling device.

33. The method of claim 1, wherein said stopping criterion is based at least in part on a value of said current vector and past values of said current vector.

34. The method of claim 1, wherein (1) comprises generating, by said sampling device, one or more samples of discrete vectors, each sample of said one or more samples being generated from said set of permissible discrete vectors associated with said objective function or loss function, wherein each sample of said one or more samples is generated based on a probability distribution determined at least in part by said set of scaling parameters and said objective function or loss function, wherein said first argument of said objective function or loss function takes said current value of said continuous vector.

35. The method of claim 1 or 34, wherein (2) comprises obtaining, by said digital computer, one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument, wherein said first argument of said objective function or loss function takes said current value of said continuous vector, and said second argument takes value of a selected sample from said one or more samples, wherein said selected sample is non-repetitively selected.

36. The method of claim 1, wherein said stopping criterion comprising a set of rules for determining accuracy of a solution to said robust inference problem.

37. The method of claim 1, wherein said selection of said subset of objective functions or loss functions is non-repetitive or repetitive.

38. The method of claim 1, wherein said sampling device comprises a non-classical computer.

39. The method of claim 38, wherein said non-classical computer is a quantum computer.

40. A system for stochastic optimization of a robust inference problem using a sampling device, comprising a digital computer configured to:

a) receive data of said robust inference problem, wherein said data comprises:

i) a set of objective functions or loss functions grouped into non-overlapping subsets, wherein each objective function or loss function in said set of objective functions or loss functions accepts first and second arguments;

i) a set of permissible discrete vectors for each objective function or loss function in said set of said objective functions or loss functions;

b) set a current value of a continuous vector as said initial vector;

c) receive a schedule of a set of scaling parameters;

d) set initial values of said set of scaling parameters based at least in part on said schedule; and

e) until a stopping criterion is met:

i) determine current values of said set of scaling parameters based at least in part on said schedule;

ii) select a subset of said objective functions or loss functions from said non-overlapping subsets;

iii) iterate said following steps for each objective function or loss function of said selected subset of said objective functions or loss functions:

1) control said sampling device to generate one or more samples of discrete vectors from said set of permissible discrete vectors associated with said objective function or loss function;

2) obtain one or more gradients, wherein each of said one or more gradients is of said objective function or loss function taken with respect to said first argument; and

3) obtain an average of said one or more gradients;

iv) obtain a summation and a partial summation of said averages of said one or more gradients, wherein said summation is for all objective functions or loss functions in said selected subset of said objective functions or loss functions, and wherein said partial summation is for more than one objective function or loss function in said selected subset of said objective functions or loss functions;

v) compute a search direction based at least in part on one or more of: v1) said summation or said partial summation of said averages of said one or

more gradients; v2) said current values of said set of scaling parameters; v3) at least part of a history of said summation or partial summation of said averages of said one or more gradients; and v4) at least part of a history of said values of said set of scaling parameters;

vi) compute a step length based at least in part on one or more of: vi1) said current values of said set of scaling parameters; vi2) said selected subset of said objective functions or loss functions; vi3) at least part of a history of values of said set of scaling parameters; and vi4) at least part of a history of said selected subset of said objective functions or loss functions;

vii) set said current value of said continuous vector based on said step length and said search direction; and

viii) provide said current value of said continuous vector.

41. The system of claim 40, wherein said sampling device comprises a non-classical computer.

42. The system of claim 41, wherein said non-classical computer is a quantum computer.

Fig. 1

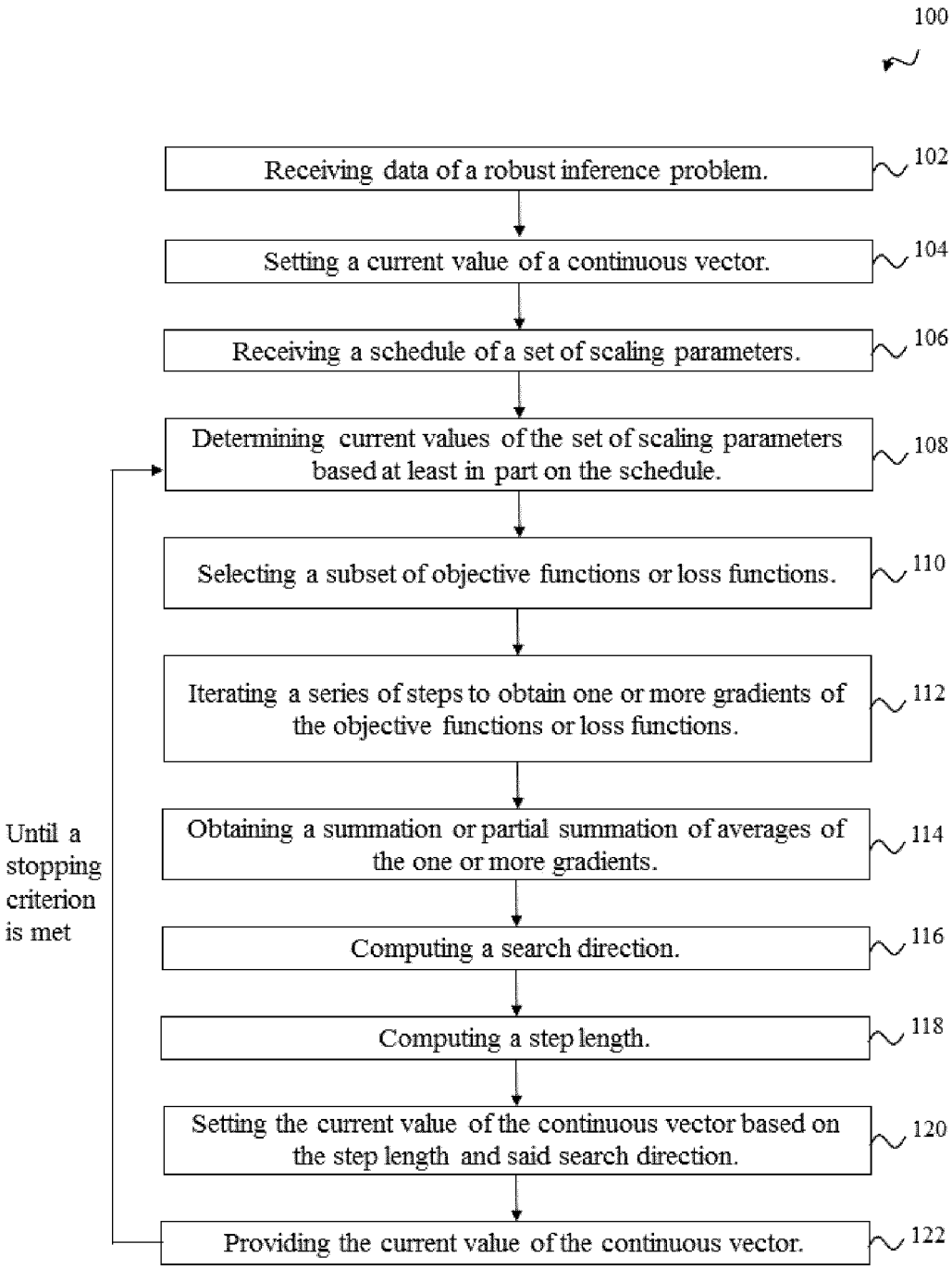

100
Receiving data of a robust inference problem.
102
Setting a current value of a continuous vector.
104
Receiving a schedule of a set of scaling parameters.
106
Determining current values of the set of scaling parameters based at least in part on the schedule.
108
Selecting a subset of objective functions or loss functions.
110
Iterating a series of steps to obtain one or more gradients of the objective functions or loss functions.
112
Obtaining a summation or partial summation of averages of the one or more gradients.
114
Computing a search direction.
116
Computing a step length.
118
Setting the current value of the continuous vector based on the step length and said search direction.
120
Providing the current value of the continuous vector.
122
Until a stopping criterion is met

Fig. 2

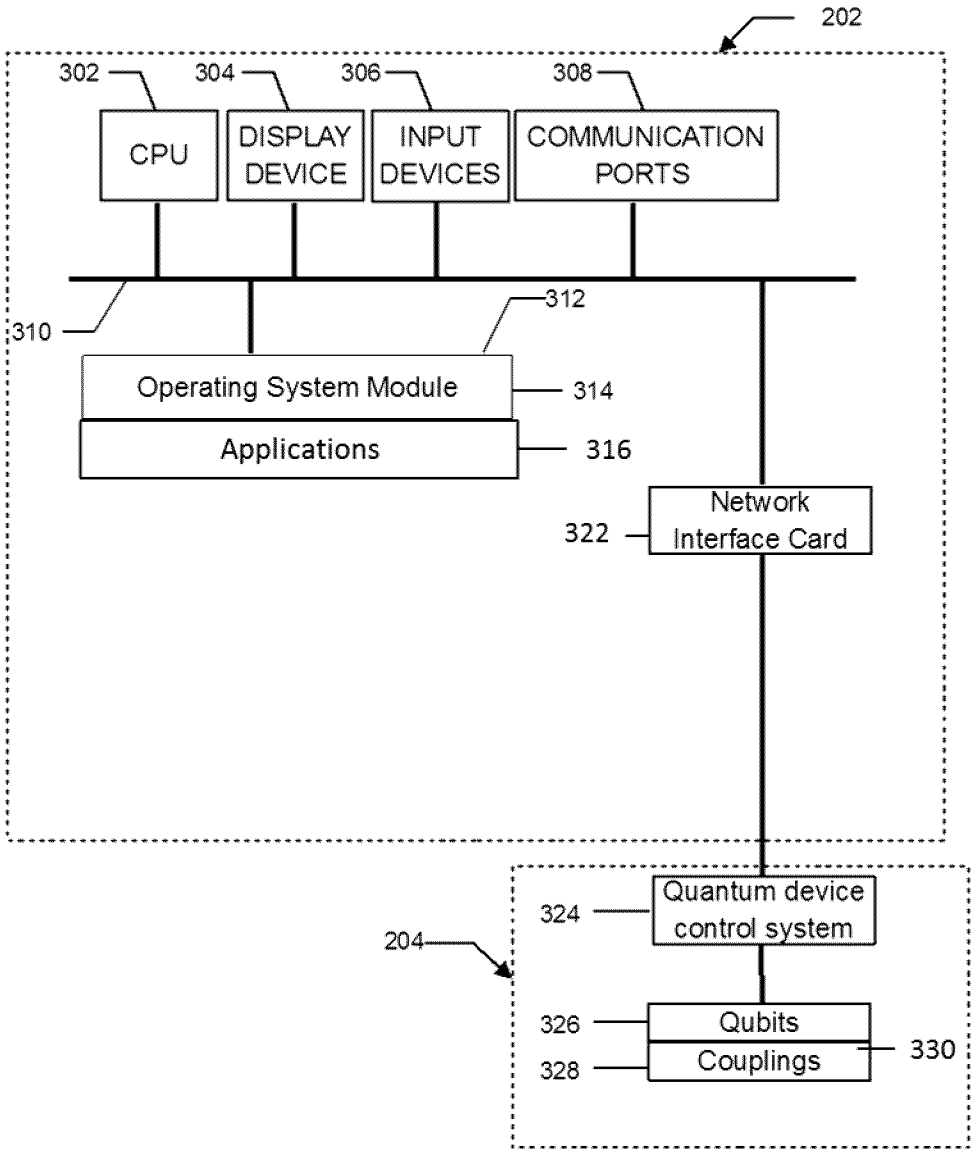
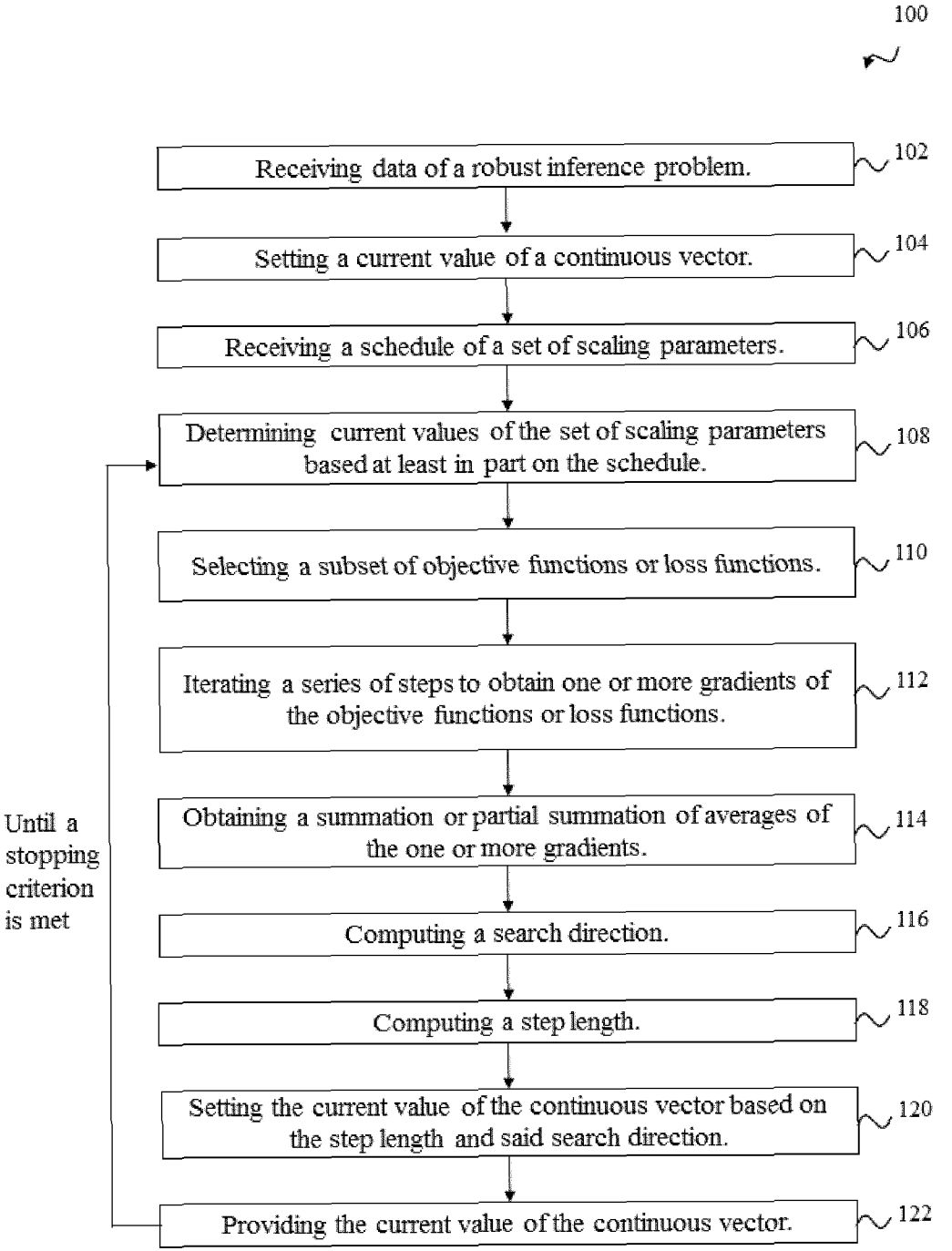

202
302
304
306
308
CPU
DISPLAY DEVICE
INPUT DEVICES
COMMUNICATION PORTS
310
312
Operating System Module
314
Applications
316
322
Network Interface Card
204
324
Quantum device control system
326
Qubits
328
Couplings
330

Fig. 1


100
Receiving data of a robust inference problem.
102
Setting a current value of a continuous vector.
104
Receiving a schedule of a set of scaling parameters.
106
Determining current values of the set of scaling parameters based at least in part on the schedule.
108
Selecting a subset of objective functions or loss functions.
110
Iterating a series of steps to obtain one or more gradients of the objective functions or loss functions.
112
Obtaining a summation or partial summation of averages of the one or more gradients.
114
Computing a search direction.
116
Computing a step length.
118
Setting the current value of the continuous vector based on the step length and said search direction.
120
Providing the current value of the continuous vector.
122
Until a stopping criterion is met