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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 December 2015 (10.12.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/184729 Al

(51) International Patent Classification:
G06F 19/00 (201 1.01)

(21) International Application Number:
PCT/CN20 14/090050

(22) International Filing Date:
31 October 2014 (31.10.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN20 14/079308 5 June 2014 (05.06.2014) CN
2014 10422475.X 25 August 2014 (25.08.2014) CN

(71) Applicant: **TSINGHUA UNIVERSITY** [CN/CN]; Qinghuayuan, Haidian District, Beijing 100084 (CN).

(72) Inventors: **YANG, Guangwen;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **JI, Yingsheng;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **CHEN, Yushu;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **ZHENG, Weijie;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **ZHANG, Wusheng;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **FU, Haohuan;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **HUANG, Xiaomeng;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **JIANG, Jinlei;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **WANG, Xiaoge;** Qinghuayuan, Haidian District, Beijing 100084 (CN). **LIU, Li;** Qinghuayuan, Haidian District, Beijing 100084 (CN).

(74) Agent: **TSINGYIHUA INTELLECTUAL PROPERTY LLC;** Room 301, Trade Building, Zhaolanyuan, Tsinghua University, Qinghuayuan, Haidian District, Beijing 100084 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on nextpage]

(54) Title: METHOD AND SYSTEM FOR HYPER-PARAMETER OPTIMIZATION AND FEATURE TUNING OF MACHINE LEARNING ALGORITHMS

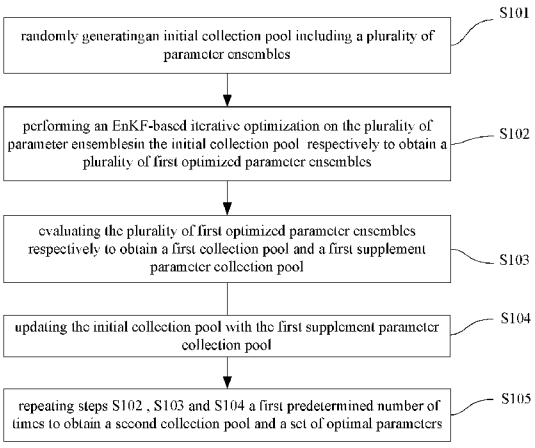

Fig. 1

(57) Abstract: The present disclosure provides a method for hyper-parameter optimization and feature tuning of machine learning algorithms. The method comprises: **A:** randomly generating an initial collection pool including a plurality of parameter ensembles; B: performing an EnKF-based iterative optimization on the plurality of parameter ensembles in the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles; C: evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool; D: updating the initial collection pool with the first supplement parameter collection pool; repeating steps B, C and D a first predetermined number of times to obtain a second collection pool and a set of optimal parameters. The method increases calculation efficiency and accuracy of the hyper-parameter optimization and feature tuning, and has strong universality. Besides, the present disclosure further provides a system for hyper-parameter optimization and feature tuning of machine learning algorithms.

WO 2015/184729 A1

Published:

— *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR HYPER-PARAMETER OPTIMIZATION AND FEATURE TUNING OF MACHINE LEARNING ALGORITHMS

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority to and benefits of Chinese Patent Application Serial No. 201410422475.X, filed with the State Intellectual Property Office of P. R. C. on Aug 25, 2014, which claims priority to the International Patent Application No. PCT/CN2014/079308, filed with the State Intellectual Property Office of P. R. C. on June 5, 2014, all of which are incorporated herein by reference.

FIELD

The present disclosure relates to machine learning technology field, and more particularly relates to a method and a system for hyper-parameter optimization and feature tuning of machine learning algorithms.

BACKGROUND

For a general machine learning algorithm, a performance of a model in the algorithm mostly depends on its parameter setting. The performances of models generated with various parameter ensembles are largely different. Hyper-parameter optimization is a stochastic optimization problem. The randomness is mainly derived from the fact that training data and testing data which are used to generate models involve finite samples based on an unknown joint distribution and that the finite samples can not reflect the overall. Then, the optimization problem is formalized: the target of hyper-parameter optimization is to find hyper-parameters Θ of a machine learning algorithm F to create a function f on the given arbitrary training samples X_T from an unknown natural distribution G. The function f is used to maximize (or minimize) performance evaluation value according to some evaluation criteria g(-) . The optimization problem is made explicit by Equation 1:

$$\theta_{opt} \approx \arg\max_{\theta} \frac{1}{n} \sum_{x \in X_V} g\left(x, F\left(\theta, X_T\right)\right) \quad (1)$$

However, it is difficult to directly compute the expectation over the distribution G. Thus, the general way is to optimize the expectation mean of function f based on the validation samples

X_V by using model selection methods such as cross validation to insure generalization performance. Having no knowledge about the space of hyper-parameters, the primary way of getting the best solution Θ_{opt} is to select a finite number of parameter ensembles from the space of hyper-parameters for training to generate models, and to evaluate each parameter ensemble and finally output the optimal solution.

For machine learning algorithms, hyper-parameter optimization is always a challenge since a plurality of parameters form a high-dimensional continuous parameter space with massive parameter ensembles. Grid search is the most widely used method for hyper-parameter optimization. The whole parameter space is expressed as a grid composed of discrete points. Each point represents a possible parameter ensemble, and for each parameter, a number of feasible values are prepared. Grid search is a method to test each parameter ensemble by exhaustive search. Grid search is simple for parallel execution and safe for global optimum on the given grid. However, conducting an exhaustive search over the grid is computationally expensive, which scales exponentially with the number of parameters and the growth of search granularity.

Over the past decades, many optimization methods have been developed to speed up the parameter optimization of machine learning, which can be divided into two categories. One category is the numerical optimization methods, such as gradient descent. The other category is the evolutionary optimization methods. The numerical optimization methods have optimized search direction and step length for the next updating step by using information obtained from numerical calculation, thus having fast convergence. Compared with other methods, the numerical optimization methods have higher efficiency in search owing to their fast convergence. However, the effect of these methods is significantly affected by the setting of the starting points since the numerical calculation method is easy to fall into local optima. Secondly, the convergence speed of the numerical optimization methods cannot be promised when handling massive parameters. Besides, most numerical methods are embedded. In other word, different algorithms need special coding for different models, which leads to poor universality. The evolutionary algorithms include genetic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO), and so on. Compared with the numerical methods, the evolutionary algorithms can effectively avoid trapping into local optima and thus can find the approximate global optimal solution. However, since the evolutionary algorithms have no explicit search directions (just neighbor search) and certain step length, low convergence speed may lead to a long training time.

For machine learning algorithms, another problem is feature tuning, which comprises feature enhancement and feature selection. The former procedure of training models is to scale the feature values of samples, so as to normalize the feature values to a uniform range, thus avoiding performance reduction of the model caused by the numerical difference of the feature values. Conversely, scale factors can be adjusted according to the importance of features to improve performance of models, which is called feature enhancement. Feature selection is the procedure of selecting a feature subset for establishing the model, where redundant or irrelevant features are removed. Feature selection can shorten the model construction time by dimension reduction, and even improve generalization performance. There are three ways for feature selection: filter, wrapper, and embedded methods. Filter methods select the most important features by evaluating the statistical properties of data. The calculation of these filter methods is simple, while is lack of model test and has lower accuracy. Wrapper methods involve an iterative procedure of selecting features, training a model and then assessing the performance. Although the wrapper methods promise higher accuracy of the feature subset, these methods consume huge computation. Embedded methods perform feature selection when training models, however, different algorithms need special coding for different models, which leads to poorer universality. Besides, feature selection can also be performed by tuning the scale factors of the features. However, feature selection usually needs the optimal parameter ensemble obtained by hyper-parameter optimization to assess the model performance. Thus, the efficient method of jointly tuning hyper-parameters and features is required.

The amount of parameters is huge by considering the scale factors of the features as a kind of parameters for feature tuning and for hyper-parameter optimization with the hyper-parameters of machine learning algorithms. Thus, there is a need to develop a quick, accurate, general and effective method for hyper-parameter optimization and feature tuning of machine learning algorithms, especially for optimization in a high-dimensional continuous parameter space.

SUMMARY

Embodiments of the present disclosure seek to solve at least one of the problems existing in the related art to at least some extent.

A first objective of the present disclosure is to provide a method for hyper-parameter optimization and feature tuning of machine learning algorithms, which increases calculation

efficiency and accuracy of the parameter optimization, and has a high generality.

A second objective of the present disclosure is to provide a system for hyper-parameter optimization and feature tuning of machine learning algorithms.

In order to achieve the above objectives, a method for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of a first aspect of the present disclosure comprises: A: randomly generating an initial collection pool including a plurality of parameter ensembles; B: performing an EnKF-based (Ensemble Kalman Filter-based) iterative optimization on the plurality of parameter ensembles in the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles; C: evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and/without a first supplement parameter collection pool; D: updating the initial collection pool with the first supplement parameter collection pool; E: repeating steps B, C and D a first predetermined number of times to obtain a second collection pool and a set of optimal parameters .

With the method for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure, the optimal parameter ensemble can be effectively searched from a high-dimensional continuous parameter space. The method can perform hyper-parameter optimization and feature tuning simultaneously, thus increasing calculation efficiency and accuracy of the hyper-parameter optimization. Moreover, the method has a high generality and is suitable for all kinds of machine learning algorithms.

In some embodiments, the method further comprises: randomly combining two parameter ensembles in the second collection pool in turn to obtain a second supplement parameter collection pool; performing an EnKF-based iterative optimization on parameter ensembles in the second supplement parameter collection pool respectively to obtain a plurality of second optimized parameter ensembles; evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool; repeating above steps a second predetermined number of times to obtain a new set of optimal parameters.

In some embodiments, step A comprises: randomly generating a parameter vector $\Theta \in ¾^{m\times 1}$, in which each parameter of the parameter vector has a predetermined value range, wherein m is a number of parameters in the parameter vector; randomly generating a set of normalized orthogonal vectors $\mathbf{j}\rho_i | \rho_i \in ¾^{m\times 1}, i = 1,...,N|$ to promise linearly independent parameter perturbations, wherein N is a number of normalized orthogonal vectors; calculating a parameter

perturbation ensemble by

$$A' = \{F_a\gamma_1 p_1, F_a\gamma_2 p_2, \ldots, F_a\gamma_N p_N) \in \Re^{m\times N}, \gamma_i \sim N(0, S_P),$$

where A' indicates the parameter perturbation ensemble, p_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_P is a configurable variance, F_α is a matrix defined as $F_\alpha = (f_1 e_1, f_2 e_2, \ldots, f_N e_N)$, e_i is a unit vector, and f_i is a configurable scaling variable used to adjust a parameter perturbation amplitude; adding N perturbation vectors s_i in the parameter perturbation ensemble A' respectively to the parameter vector Θ to obtain N sets of parameters $\theta_i = \theta + \varepsilon_i$, such that a parameter ensemble comprising N sets of parameters is obtained; repeating above steps to obtain the initial collection pool including a plurality of parameter ensembles.

In some embodiments, step B comprises: training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models; evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results; updating the plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm.

In some embodiments, training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models comprises: performing feature scaling on the predetermined training dataset to obtain a normalized training dataset; and using each set of parameters of each parameter ensemble as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble.

In some embodiments, evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results comprises: performing feature scaling on the predetermined validation dataset to obtain a normalized validation dataset; predicting each sample in the normalized validation dataset using each of the plurality of models to obtain a predicting result corresponding to each model, so as to obtain a predicting result ensemble $HA = (\mathrm{H}\theta_1, \mathrm{H}\theta_2, \ldots, \mathrm{H}\Theta_N) \in \Re^{n\chi\Lambda'}$ where HA indicates the predicting result ensemble comprising the predicting results corresponding to the plurality of models, n is the number of samples in the normalized validation dataset.

In some embodiments, step B further comprises: generating an observation ensemble and an observation perturbation ensemble; and updating the plurality of parameter ensembles respectively according to the predicting result ensemble, the observation ensemble and the observation perturbation ensemble by means of the EnKF algorithm.

In some embodiments, generating an observation ensemble and an observation perturbation comprises: randomly generating a set of vectors; randomly generating an observation perturbation ensemble according to the set of vectors; adding each set of perturbation vector in the observation perturbation ensemble to an initial observation vector, so as to obtain the observation ensemble.

In some embodiments, the parameter ensemble is updated according to the formula of

$$A^a = A^f + A'(HA')^T \left(HA'(HA')^T + \gamma\gamma^T\right)^{-1}(D - HA),$$

where A^F is the current parameter ensemble, A^a is the updated parameter ensemble, A' is the parameter perturbation ensemble, D is the observation ensemble, γ is the observation perturbation ensemble, HA is the predicting result ensemble, and HA is the ensemble perturbations of HA.

In some embodiments, step C comprises: evaluating the plurality of first optimized parameter ensembles respectively to obtain a performance value of each first optimized parameter ensemble; comparing the performance value of each first optimized parameter ensemble with a first threshold and a second threshold, in which the first threshold is larger than the second threshold; storing the first optimized parameter ensemble in the first collection pool when the performance value of the first optimized parameter ensemble is not less than the first threshold; discarding the first optimized parameter ensemble when the performance value of the first optimized parameter ensemble is less than or equal to the second threshold; determining that the first optimized parameter ensemble is a parameter ensemble with general performance when the performance value of the first optimized parameter ensemble is larger than the second threshold and less than the first threshold; and randomly combining two parameter ensembles with general performance in turn to obtain the first supplement parameter collection pool.

In some embodiments, randomly combining two parameter ensembles with general

performance in turn to obtain the first supplement parameter collection pool comprises: randomly selecting two parameter ensembles from the parameter ensembles with general performance; calculating the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles according to formulas of

$$Q_{ij} = I + \left(HA_i\right)^T \left(HA,'(HA,)^T + HA_j'(HA_j)^T\right)^{-1} (HA_j - HA,) \in \Re^{N\times N},$$
$$Q_{ji} = I + (HA_j')^T \left[HA,'(HAJ)^T + HA,'(HA,)^T\right)^{-1} (HA, - HA_j) \in \Re^{N\times N},$$

where I is an identity matrix, $A,$ and A_j are the two selected parameter ensembles, $HA,$ is the predicting result ensemble corresponding to the parameter ensemble $A,$, and $HA',$ is the ensemble perturbations of $HA,$, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and HA'_j is the ensemble perturbations of HA_j;

calculating the refined parameter ensembles A_j and A_{ji} respectively according to formulas of

$$A_{ij} = \overline{A_i} + A_i'Q_\beta$$

$$A_B = \overline{A}_j + A_j'Q_\beta;$$

decomposing Q_{ij} and Q_{ji} respectively to obtain $Q_{ij} = C_{ij}\tilde{S}_{ij}\tilde{V}_{ij}$ and $Q_{ji} = C_{ji}\tilde{S}_{ji}\tilde{V}_{ji}$, where $\tilde{U}_y$ and $\tilde{U}_j$, are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, $\tilde{V}_i$ are permutation matrices; choosing N columns with maximum absolute values of diagonal elements from $\tilde{S}_j$ and $\tilde{S}_{Ji}$ respectively, obtaining corresponding parameter vectors from A_j and A_{ji} according to the N columns to generate a combined parameter ensemble for storing in the first supplement parameter collection pool.

In some embodiments, evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool comprises: evaluating the plurality of second optimized parameter ensembles respectively to obtain a performance value of each second optimized parameter ensemble; comparing the performance value of each second optimized parameter ensemble with a third threshold; adding the second optimized parameter ensemble into

the second collection pool when the performance value of the second optimized parameter ensemble is not less than the third threshold; discarding the second optimized parameter ensemble when the performance value of the second optimized parameter ensemble is less than the third threshold.

According to a second aspect of the present disclosure, a system for hyper-parameter optimization and feature tuning of machine learning algorithms is provided, and the system comprises: a generating module used for randomly generating an initial collection pool including a plurality of parameter ensembles; a first optimization module used for performing an EnKF-based iterative optimization on the plurality of parameter ensembles respectively to obtain a plurality of optimized parameter ensembles; a first evaluating module used for evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool; an updating module used for updating the initial collection pool with the first supplement parameter collection pool; the first optimization module, the first evaluating module and the updating module are controlled to repeat working to obtain a second collection pool and a set of optimal parameters; and an output module used for outputting the optimal parameter ensemble.

With the system for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure, the optimal parameter ensemble can be effectively searched from a high-dimensional continuous parameter space. The system can perform hyper-parameter optimization and feature tuning simultaneously, thus increasing calculation efficiency and accuracy of the hyper-parameter optimization. Moreover, the system has a high generality and is suitable for all kinds of machine learning algorithms.

In some embodiments, the system further comprises: a first combining module used for randomly combining two parameter ensembles in the second collection pool in turn to obtain a second supplement parameter collection pool; a second optimization module used for performing an EnKF-based iterative optimization on parameter ensembles in the second supplement parameter collection pool respectively to obtain a plurality of second optimized parameter ensembles; a second evaluating module used for evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool; and above modules are controlled to repeat working to obtain a new set of optimal parameters.

In some embodiments, the generating module comprises: a first generating unit used for

randomly generating a parameter vector $\Theta \in \mathbb{R}^{m\times 1}$, in which each parameter of the parameter vector has a predetermined value range, wherein m is a number of parameters in the parameter vector; a second generating unit used for randomly generating a set of normalized orthogonal vectors $\{\rho_i \mid \rho_i \in \mathbb{R}^{m\times 1}, i = 1,...,N\}$ to promise linearly independent parameter perturbations, wherein N is a number of normalized orthogonal vectors; a calculating unit used for calculating a parameter perturbation ensemble by $A' = (F_\alpha \gamma_1 \rho_1, F_\alpha \gamma_2 \rho_2, \ldots, F_\alpha \gamma_N \rho_N) \in \mathbb{R}^{m\times N}, \gamma_i \sim N(0, S_p)$, where A' indicates the parameter perturbation ensemble, ρ_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_p is a configurable variance, F_α is a matrix defined as $F_\alpha = (f_1 e_1, f_2 e_2, \ldots, f_N e_N)$, e_i is a unit vector, and f_i is a configurable scaling variable used to adjust a parameter perturbation amplitude; a third generating unit used for adding N perturbation vectors ε_i in the parameter perturbation ensemble A' respectively to the parameter vector Θ to obtain N sets of parameters $\theta_i = \theta + \varepsilon_i$, such that a parameter ensemble comprising N sets of parameters is obtained.

In some embodiments, the first optimization module is further used for training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models; evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results; and updating the plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm.

In some embodiments, the first optimization module is further used for performing feature scaling on the predetermined training dataset and the predetermined validation dataset to obtain a normalized training dataset and a normalized validation dataset; using each set of parameters of each parameter ensemble as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble; predicting each sample in the normalized validation dataset using each of the plurality of models to obtain a predicting result corresponding to each model, so as to obtain a predicting result ensemble $HA = (H\theta_1, H\theta_2, \ldots, H\theta_N) \in \mathbb{R}^{n\times N}$, where HA indicates the predicting result ensemble comprising the predicting results corresponding to the plurality of models, n is the number of samples in the normalized validation dataset.

In some embodiments, the first optimization module is further used for generating an observation ensemble and an observation perturbation ensemble; and updating the plurality of

parameter ensembles respectively according to the predicting result ensemble, the observation ensemble and the observation perturbation ensemble by means of the EnKF algorithm.

In some embodiments, the first optimization module is further used for randomly generating a set of vectors; randomly generating an observation perturbation ensemble according to the set of vectors; adding each set of perturbation vector in the observation perturbation ensemble to an initial observation vector, so as to obtain the observation ensemble.

In some embodiments, the parameter ensemble is updated according to the formula of

$$A^a = A^f + A'(HA')^T \left(HA'(HA')^T + \gamma\gamma^T\right)^{-1} (D - HA),$$

where A^F is the current parameter ensemble, A^a is the updated parameter ensemble, $A' = A - \bar{A}$ is the parameter perturbation ensemble, D is the observation ensemble, γ is the observation perturbation ensemble, HA is the predicting result ensemble, and HA' is the ensemble perturbations of HA.

In some embodiments, the first evaluating module is further used for: evaluating the plurality of first optimized parameter ensembles respectively to obtain a performance value of each first optimized parameter ensemble; comparing the performance value of each first optimized parameter ensemble with a first threshold and a second threshold, in which the first threshold is larger than the second threshold; storing the first optimized parameter ensemble in the first collection pool when the performance value of the first optimized parameter ensemble is not less than the first threshold; discarding the first optimized parameter ensemble when the performance value of the first optimized parameter ensemble is less than or equal to the second threshold; determining that the first optimized parameter ensemble is a parameter ensemble with general performance when the performance value of the first optimized parameter ensemble is larger than the second threshold and less than the first threshold; and randomly combining two parameter ensembles with general performance in turn to obtain the first supplement parameter collection pool.

In some embodiments, the first evaluating module is further used for: randomly selecting two parameter ensembles from the parameter ensembles with general performance; calculating the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles according to formulas of

$$\text{¾} = I + (\boldsymbol{H}A_i')^T \left(H\boldsymbol{A}_i'(\boldsymbol{HA},')^T + \boldsymbol{H}\boldsymbol{A}_j'(\boldsymbol{HA}_j')^T\right)^{-1} (\boldsymbol{HA}_j - HA,) \in \Re^{N\times N},$$

$$Q_{ji} = I + \left(HA_j'\right)^T \left(HA_j'(HA_j')^T + HA_i'(HA_i')^T\right)^{-1} (HA_i' - HA_j') \in \Re^{N\times N},$$

where I is an identity matrix, A_j and A_j are the two selected parameter ensembles, HA_i is the predicting result ensemble corresponding to the parameter ensemble A_j, and HA_i' is the ensemble perturbations of HA_i, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and HA_j' is the ensemble perturbations of HA_j;

calculating the refined parameter ensembles A_j and A_{ji} respectively according to formulas of $A_j = \overline{A_i} + A_i'Q_j$ and $A_{ji} = \overline{A_j} + A_j'Q_{ji}$; decomposing Q_{ij} and β_{ji} respectively to obtain $Q_{ij} = U_{ij}\tilde{S}_{ij}V_{ij}$ and $Q_{ji} = U_{ji}\tilde{S}_{ji}V_{ji}$, where U_{ij} and U_{ji} are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, V_{ij} are permutation matrices; choosing N columns with maximum absolute values of diagonal elements from $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ respectively, obtaining corresponding parameter vectors from A_{ij} and A_{ji} according to the N columns to generate a combined parameter ensemble for storing in the supplement parameter collection pool.

In some embodiments, the second evaluating module is further used for: evaluating the plurality of second optimized parameter ensembles respectively to obtain a performance value of each second optimized parameter ensemble; comparing the performance value of each second optimized parameter ensemble with a third threshold; adding the second optimized parameter ensemble into the second collection pool when the performance value of the second optimized parameter ensemble is not less than the third threshold; discarding the second optimized parameter ensemble when the performance value of the second optimized parameter ensemble is less than the third threshold.

These additional aspects and advantages of the present disclosure will become apparent from the following descriptions and more readily appreciated from the embodiments of the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

These and/or additional aspects and advantages of the present disclosure will become

apparent and more readily appreciated from the following descriptions of embodiments made with reference to the drawings, in which:

Fig. 1 is a flow chart of a method for hyper-parameter optimization and feature tuning of machine learning algorithms according to an embodiment of the present disclosure;

Fig. 2 is a schematic diagram showing a principle of the method for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure;

Fig. 3 is a flow chart of an EnKF-based update phase for a plurality of parameter ensembles according to an embodiment of the present disclosure;

Fig. 4 is a flow chart of an EnKF update phase for a parameter ensemble according to an embodiment of the present disclosure;

Fig. 5 is a flow chart of a basic evolution procedure according to an embodiment of the present disclosure;

Fig. 6 is a flow chart of a fusion search procedure according to an embodiment of the present disclosure;

Fig. 7 is a schematic diagram of a system for hyper-parameter optimization and feature tuning of machine learning algorithms according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

Reference will be made in detail to embodiments of the present disclosure. The embodiments described herein with reference to drawings are explanatory, illustrative, and used to generally understand the present disclosure. The embodiments shall not be construed to limit the present disclosure. The same or similar elements and the elements having same or similar functions are denoted by like reference numerals throughout the descriptions.

In the following, the present disclosure will be described in detail with reference to drawings and embodiments.

Fig. 1 illustrates a flow chart of a method for hyper-parameter optimization and feature tuning of machine learning algorithms according to an embodiment of the present disclosure. Fig. 2 illustrates a schematic diagram showing a principle of the method for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure. Different parts of the steps carried by the method are described in detail with reference to Fig.3 to Fig.6. As shown from Fig. 1 to Fig. 6, the method comprises the following steps.

At step S101, an initial collection pool including a plurality of parameter ensembles is generated randomly.

If feature enhancement is closed, the parameter ensemble only includes hyper-parameters for machine learning algorithms; if feature tuning is activated, the parameter ensemble includes hyper-parameters for machine learning algorithms and scale factors for feature tuning. Let $A = (\theta_1, \theta_2, ...0_N) \leq \Re^{m \times N}$ denote a parameter ensemble, where the parameter ensemble member θ_i is a set of parameters (acting as a set of system sates in the EnKF algorithm), A is a mxN state matrix, N is the number of parameter ensemble members in A and m is the number of parameters in θ_i,. In some embodiments, let N_e denote the number of the plurality of parameter ensembles, and these parameter ensembles are generated by following steps.

At step 1-1, a parameter vector $6' \in ¾^{m \times 1}$ is randomly generated, where each parameter of the parameter vector has a predetermined value range.

At step 1-2, a set of normalized orthogonal vectors $\{\rho_i | \rho_i \Subset ¾^{m \times 1}, i = 1, \ldots, N\}$ are randomly generated to promise linearly independent parameter ensemble perturbations.

At step 1-3, a parameter perturbation ensemble A' is calculated by:

$$A' = (F_\alpha \gamma_{1P1}, F_\alpha \gamma_2 p_2, \ldots, F_\alpha \gamma_N p_N) \in \Re^{m \times N}, {}_{Yi} \sim N(o, S_p),$$

where A' indicates the parameter perturbation ensemble, p_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_p is a configurable variance, and F_a is a matrix defined as $F_\alpha = (f_1 e_1, f_2 e_2, ..., f_N e_N)$, e_i is an unit vector, and f_i is a configurable scaling variable used to adjust parameter perturbation amplitude.

At step 1-4, each perturbation vector $\varepsilon_i \leftarrow F_{aYi} p_i$ in the parameter perturbation ensemble A' is added to the parameter vector Θ to obtain a set of parameters: $\theta_i \leftarrow \theta + \varepsilon_i$, such that a parameter ensemble A comprising N sets of parameters is obtained.

The above steps are repeated to generate the initial collection pool including N_e parameter ensembles.

At step SI02, an EnKF-based iterative optimization is performed on the plurality of parameter ensembles in the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles. In some embodiments, this step comprises steps of training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models; evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results; and updating the

plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm. More specifically, as shown in Fig. 3, step SI02 comprises the following steps.

At step 2-1, feature scaling is performed on a predetermined training dataset recorded as X_T and a predetermined validation dataset recorded as X_V to obtain a normalized training dataset and a normalized validation dataset.

If feature tuning is closed, a feature value of each sample in X_T and X_V is normalized to a specific range, and the normalized training data and validation data are used through the whole process of the method; if feature tuning is activated but feature selection is closed, scale factors $5_i \in \theta_i$ are used to scale each feature value; if feature selection is activated, scale factors are normalized into [o,l] by the formula of $\delta_i / \max(\{\delta_i\})$ and the normalized scale factors above a predetermined threshold are used to scale the feature values. Only the scaled feature values are trained and validated, but all the scale factors take part in the EnKF algorithm. If feature tuning is closed, step 2-1 is only performed at the first time, otherwise, for each step SI02, step 2-1 should be performed.

At step 2-2, each set of parameters θ_i are used as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble. Moreover, model selection methods are generally used to ensure generalization performance.

At step 2-3, each sample is predicted by the models to obtain a predicting result of each model. More specifically, each sample in the normalized validation dataset X_v is predicted by using each of the plurality of models to obtain a predicting result corresponding to each model.

At step 2-4, models corresponding to all the parameters of a parameter ensemble are generated and evaluated. More specifically, for each set of parameters $\Theta_i \in A$, step 2-1 to step 2-3 are repeated to generate and evaluate the model corresponding to the set of parameters. Assuming that HA is an ensemble including predicting results corresponding to the models (in the EnKF algorithm, indicating the mapping from a state space to an observation space), HA can be expressed as $HA = (H\theta_1, H\Theta_2, ..., H\Theta_N) \in$ ¾"χΛ^Γ, where n is the number of samples in the normalized validation dataset X_V, $H\theta_l$ is the predicting result corresponding to the model

which is corresponding to the set of parameters θ_i, $1 \le i \le N$.

For each parameter ensemble A, step 2-1 to step 2-4 are repeated to obtain the predicting result ensemble HA corresponding to the parameter ensemble A.

In some embodiments, after generating and evaluating the models, the method further comprises generating an observation ensemble and an observation perturbation ensemble.

Let D denote the observation ensemble, D may be expressed as $D = (d_1, d_2, ..., d_N) \in \mathbb{R}^{m \times N}$, where $d_i \in \mathbb{R}^{m \times 1}$ denotes the i^{th} observation vector which holds a set of observation values. The observation vector denotes a default confidence probability. If the default confidence probability is unknown, an initial observation vector d_0 is given. The initial observation vector d_0 denotes an observation vector which holds a set of initial observation values and the observation ensemble can be generated by the following steps.

At step 3-1, a set of normalized vectors are randomly generated. If the number n of the vectors is not large, the vectors need to be orthogonalized.

At step 3-2, an observation perturbation ensemble γ is calculated according to the set of normalized vectors. Specifically, the observation perturbation ensemble is expressed as

$$\gamma = (v_1\beta_1, v_2\beta_2, ..., v_N\beta_N) \in \mathbb{R}^{m \times N}, v_i \sim N(0, S_0),$$

where $\beta_i \in \mathbb{R}^{n \times 1}$ is the vector randomly generated in step 3-1, v_i is a stochastic step length which is subject to the Gauss distribution, S_0 is a configurable variance.

At step 3-3, each perturbation vector $v_i f_i$ in the observation perturbation ensemble γ is added to the initial observation vector d_0 to obtain an observation vector $d_i = d_0 + v_i\beta_i$, such that an observation ensemble D including N observation vectors is generated.

Step 3-1 to step 3-3 can be repeated to obtain N_e observation ensembles and observation perturbation ensembles.

In one embodiment of the present disclosure, each parameter ensemble is updated by the following formula:

$$A^a = A^f + A'(HA')^T \left(HA'(HA')^T + \gamma\gamma^T\right)^{-1} (D - HA),$$

where A^F is the current parameter ensemble, A^a is the updated parameter ensemble, $A' = A - \bar{A}$ is the parameter perturbation ensemble, D is the observation ensemble, Y is the observation perturbation ensemble, HA is the predicting result ensemble, HA' is the ensemble

perturbations of HA and HA' can be calculated by the following equations.

$$\overline{HA} = HAM_N,\ HA' = HA - \overline{HA},$$

where $\overline{HA} \in \mathbb{R}^{m \times N}$ holds average values of samples in the ensemble HA, and each element in $M_N \in \mathbb{R}^{N \times N}$ is $1/N$. Since the formula of $\left(H4'(H4')^T + \gamma\gamma^{\tau}\right)^{-1}$ is expensive in computation and storage, the UR decomposition is performed for optimization by using Householder transformation. Let $X \in \mathbb{R}^{m \times \lambda_V}$ denote the array loading HA'. Let $X(i,j)$ denote one element and $X(:,j)$ denote one column, and let τ denote the boundary of the relative residual and set $1 \geq \tau > 0$. For each column index i valued from 1 to N, the following steps are running.

At step 4-1, residual norms of the rest columns are computed by:

$$RestNorm(k) = \left\| X(i:n,k) \right\|_2, k = 1, \ldots, N;$$

At step 4-2, if $\frac{\tau}{1-\tau}\sum_{l=1}^{i-1}\left|X(l,l)\right| > (N-i+1)RestNorm\wedge$, then let a column number $P = i - 1$, and step 4-7 is followed, where $\hat{k}$ is the index of the column with maximum value among the calculated residual norms; otherwise let $p \leftarrow i$, and the columns $X(:,\hat{k})$ and $X(:,i)$ are exchanged.

At step 4-3, a rotating vector $\omega_i \in \Re^n$ is initialized, where $\omega_i(k) = \begin{cases} 0, & k < i \\ X(k,i), & k \geq i \end{cases}$;

At step 4-4, $Norm = \mathrm{Re}\,stNorm(\hat{k})$ is calculated, and if $X(i,i) > 0$, then let $\omega_i \leftarrow \omega_i + Normx\, e_i$, where e_i is a unit vector whose i^{th} element is 1 and other elements are 0. Then $X(i)$ is updated, satisfying,

$$X(i,k) = \begin{cases} X(i,k), & k < i \\ -Norm, & k = i \\ 0, & k > i \end{cases};$$

otherwise, let $\omega_i = \omega_i - Normx\ e_i$ and $X(i)$ is updated, satisfying,

$$X(i,k) = \begin{cases} X(i,k), & k < i \\ +Norm, & k = i \\ 0, & k > i \end{cases}.$$

At step 4-5, $\omega_i = \omega_i / \left\| \omega_i \right\|_2$ is calculated, and for $k = i + l : N$, the following formula is

performed:

$$X(i:n,k) = X(i:\eta,k) - 2\omega_i(i:n)\left(\omega_i(i:n)^T X(i:n,k)\right).$$

At step 4-6, let $i = i+1$, if $i > N$, step 4-7 is followed, otherwise step 4-1 is followed;

At Step 4-7, a matrix $5' \Subset ¾^{m\times p}$ is constructed. Specifically, the matrix is defined as $S = X(:,\backslash:p)$;

After running the UR decomposition from step 4-1 to step 4-7, the following approximation is obtained:

$$\left(HA'(HA')^T + \gamma\gamma^T\right)^{-1} \approx U\begin{pmatrix}\left(\hat{S}\hat{S}^T\right)^{-1} & 0\\ 0 & 0\end{pmatrix}U^T,$$

where U is an orthogonal matrix, $\hat{S} \Subset ¾^{p\times"}$ denotes the upper triangular matrix holding these non-zero columns of S and $\hat{S}$ is constituted by p columns with maximum absolute value of diagonal elements. By running the UR decomposition, the matrix U is composed by p Householder transformations and is expressed as $U = H(\alpha y_1)H(\alpha y_2)...H(\alpha y_p)$, and the Householder transformation $H(\alpha y_r)$ is defined as $H(\omega_r) = I - 2\omega_i\omega_i^T$. Then, the original updating formula is converted as $A^a = A^F + A'(HA')^T U\begin{pmatrix}\left(\hat{S}\hat{S}^T\right)^{-1} & 0\\ 0 & 0\end{pmatrix}U^T(D-HA)$.

After the UR decomposition, the calculation order for the above formula is from right to left aimed for reducing the computation and storage cost. Then, the updated parameter ensemble A^a is obtained.

If the EnKF-based iterative optimization has been performed a predetermined maximum times, the optimized parameter ensembles are obtained and used for the following performance evaluation; otherwise, go to step SI02 for the next EnKF-based iterative optimization.

At step SI03, a basic evolution procedure is performed on the plurality of first optimized parameter ensembles to obtain a first collection pool and a first supplement parameter collection pool. As shown in Fig. 5, step SI03 comprises the following steps.

At step 5, the plurality of first optimized parameter ensembles are evaluated respectively to obtain a performance value of each first optimized parameter ensemble. All the performance values are recorded.

At step 6,the first optimized parameter ensembles are divided into three groups according to the performance values of the plurality of first optimized parameter ensembles. More specifically, two thresholds *threshX* and *thresh!* are used to divide the plurality of first optimized parameter ensembles. Let *score(A)* denote the performance value of a first optimized parameter ensemble, the plurality of first optimized parameter ensembles are divided as follows.

If $score(A) \geq thresh\backslash$, the performance of the first optimized parameter ensemble is considered as good and then the first optimized parameter ensemble is stored in the first collection pool.

If $score(A) \leq thresh!$, the performance of the first optimized parameter ensemble is considered as bad and thus the first optimized parameter ensemble is discarded;

If $thresh! < score(A) < thresh\backslash$, the performance of the first optimized parameter ensemble is considered as general, and parameter ensembles with general performance are randomly combined in turn to obtain the first supplement parameter collection pool.

After evaluating all the first optimized parameter ensembles, the parameter ensembles with general performance are combined using the EnKF-based combination algorithm. Specifically, the combination comprises the following steps.

At step 7-1, two parameter ensembles A_i, A_j are randomly selected from the parameter ensembles with general performance.

At step 7-2, the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles A_i, A_j are respectively calculated by using the following formulas:

$$Q_{ij} = I + (HA,')^T (HA_i'(\boldsymbol{HA},')^T + HA_j'(HA_j')^T)^{-1} (\boldsymbol{HAJ} - HA,) \in \Re^{N\times N},$$

$$Q_{ji} = I + (HA_j')^T [\boldsymbol{HA},'(\boldsymbol{HAJ}')^T + HA,'(HA,')^T)^{-1} (HA, - HA_j) \in \Re^{N\times N},$$

where I is an identity matrix, $HA,$ is the predicting result ensemble corresponding to the parameter ensemble $A,$, and $HA',$ is the ensemble perturbations of $HA,$, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and HA_j' is the ensemble perturbations of HA_j.

The above formulas are also calculated by means of an optimization method which is similar

to the optimization method described in the above steps 4-1 to 4-7.

At step 7-3, the refined parameter ensembles A_j and A_{ji} are respectively calculated by using formulas of $A_{ij} = \overline{A_i} + A_i' Q_{ij}$ and $A_{ji} = \overline{A_j} + A_j' Q_{ji}$.

At step 7-4, Q_y and Q_{ji} are respectively decomposed according to formulas of $Q_{ij} \sim \tilde{U}_{ij}\tilde{S}_{ij}\tilde{V}_{ij}$ and $Q_{ji} \sim \tilde{U}_{ji}\tilde{S}_{ji}\tilde{V}_{ji}$,

where $\tilde{U}_y$ and $\tilde{U}_\beta$ are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{Ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, $\tilde{V}_{ij}$ are permutation matrices for interchanging columns when pivot elements are chosen.

At step 7-5, N columns with maximum absolute values of diagonal elements are respectively selected from $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$, and corresponding parameter vectors are obtained from A_j and A_{ji} according to the N columns to generate a combined parameter ensemble A^m.

At step 7-6, it is determined whether there are other parameter ensembles with general performance need to be combined, if no, subsequent steps are performed, and if yes, another two parameter ensembles with general performance are randomly selected for combination, i.e. the above steps 7-1 to 7-5 are repeated.

At step SI04: the initial collection pool is updated with the first supplement parameter collection pool. If some parameter ensembles have been discarded or combined with other parameter ensembles, then go to S101 to generate new parameter ensembles randomly and the new generated parameter ensembles and the parameter ensembles in the first supplement parameter collection pool compose the initial collection pool so that the initial collection pool include N_e parameter ensembles.

At step SI05: the EnKF-based iterative optimization and the basic evolution procedure is repeated a first predetermined number of times to obtain a second collection pool and a set of optimal parameters.

If the basic evolution procedure has been performed a first predetermined number of times, a set of optimal parameters may be obtained according to recorded performance values. The parameter ensembles satisfying $score^{\wedge}A) \geq threshl$ are stored in the second collection pool for a fusion search procedure; otherwise, go to step SI02.

In some embodiments, as shown in Fig. 6, the method further comprises the following steps.

At step S201, two parameter ensembles in the second collection pool are combined in turn to obtain a second supplement parameter collection pool.

The parameter ensembles in the second collection pool are combined using the EnKF-based combination algorithm the principle of which is as described in the above steps 7-1 to 7-6.

At step S202, an EnKF-based iterative optimization is performed on parameter ensembles in the second supplement parameter collection pool respectively to obtain a plurality of second optimized parameter ensembles.

The principle of the EnKF-based iterative optimization is as described in step S102.

At step S203, a fusion search is performed on the plurality of second optimized parameter ensembles respectively to update the second collection pool. The plurality of second optimized parameter ensembles are divided as follows.

At step 8, the plurality of second optimized parameter ensembles are evaluated respectively to obtain a performance value of each second optimized parameter ensemble. All the performance values are recorded.

At step 9, the second optimized parameter ensembles are divided into two groups according to the performance values of the plurality of second optimized parameter ensembles. More specifically, a threshold *thresh* is used to divide the plurality of second optimized parameter ensembles. Let *score^A)* denote the performance value of a second optimized parameter ensemble, the plurality of second optimized parameter ensembles are divided as follows.

If $score(A) \geq thresh$, the performance of the second optimized parameter ensemble is considered as good and then the optimized parameter ensemble is added in the second collection pool.

If *score^A)* $< thresh$, the performance of the second optimized parameter ensemble is considered as bad and thus the second optimized parameter ensemble is discarded.

At step S204, steps S201-S203 are repeated a second predetermined number of times to obtain a new set of optimal parameters.

If the fusion search procedure has been performed a second predetermined number of times (may be allowed to set as 0, which means that the fusion search procedure will be skipped), then go to step 10; otherwise, go to step S201.

For each parameter ensemble in the initial collection pool the EnKF-based iterative optimization and basic evolution are performed a first predetermined number of times. For each parameter ensemble in the second supplement parameter collection pool, the EnKF-based iterative optimization and fusion search are performed a second predetermined number of times.

At step 10, the set of parameters with the best performance are outputted as the optimal solutions.

In conclusion, The EnKF-based method according to embodiments of the present disclosure is effective in hyper-parameter optimization and feature tuning of machine learning algorithms. As a kind of parameters, the feature scale factors are optimized for feature enhancement and feature selection as well as for parameter optimization with the hyper-parameters of machine learning algorithms. As shown in Fig. 2, the method mainly comprises two procedures: EnKF-based iterative optimization and ensemble evolution (basic evolution and fusion search).

More specifically, based on the EnKF algorithm which can be used to solve nonlinear problems with massive parameters, the method according to embodiments of the present disclosure adopts many optimization techniques and builds an EnKF-based framework.

Firstly, by considering the hyper-parameter optimization and feature tuning of machine learning algorithms as a problem of a nonlinear system, the parameters viewed as the state of the nonlinear system are estimated by the method with the EnKF algorithm. Secondly, the method builds an EnKF-based framework, which adopts multiple ensembles methodology to avoid trapping into local optima gradient caused by the numerical optimization technique. Then the method adopts a basic evolution technique to broaden search area and improve search efficiency by evaluating the parameter ensembles calculated by the EnKF algorithm, keeping the ensembles with good performance, discarding the ensembles with bad performance and merging the ensembles with general performance. After the basic ensemble evolution procedure, by using the fusion search technique to merge the parameter ensembles with good performance, a further search is performed, thus obtaining an approximate optimum solution in the high-dimensional parameter space. Finally, since some matrixes in the EnKF algorithm is expensive in computation and storage due to the massive data of machine learning, the UR decomposition is performed to improve efficiency of the EnKF algorithm, thus obtaining better practicability.

The EnKF-based method for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure is effective in searching the optimum parameter in the high-dimensional parameter space. The method can perform hyper-parameter optimization and feature tuning simultaneously, thus increasing calculation efficiency and accuracy. Besides, the feature scale factors used for handling feature data are viewed as a kind of parameters for feature tuning. The method is universal and suitable for all

kinds of machine learning algorithms.

A system for hyper-parameter optimization and feature tuning of machine learning algorithms is also provided according to an embodiment of the present disclosure.

Fig. 7 is a schematic diagram of the system for hyper-parameter optimization and feature tuning of machine learning algorithms according to an embodiment of the present disclosure. As shown in Fig. 7, the system 700 comprises a generating module 710, a first optimization module 720, a first evaluating module 730, an updating module 740 and an output module 750.

The generating module 710 is used for randomly generating an initial collection pool including a plurality of parameter ensembles.

If feature enhancement is closed, the parameter ensemble only includes hyper-parameters for machine learning algorithms; if feature tuning is activated, the parameter ensemble includes hyper-parameters for machine learning algorithms and scale factors for feature tuning. Let $A=(\theta_1,\theta_2,...0_N)\in\Re^{m*N}$ denote a parameter ensemble, where the parameter ensemble member θ_i is a set of parameters (acting as a set of system sates in the EnKF algorithm), A is a mxN state matrix, N is the number of parameter ensemble members in A and m is the number of parameters in θ_i. In some embodiments, let N_e denote the number of the plurality of parameter ensembles, and the generating module 710 comprises a first generating unit, a second generating unit , a calculating unit and a third generating unit.

The first generating unit is used for randomly generating a parameter vector $\Theta\in ¾^{m\times 1}$, where each parameter of the parameter vector has a predetermined value range.

The second generating unit is used for randomly generating a set of normalized orthogonal vectors $\left\{\rho_i \middle| p_i \in ¾^{m\times 1}, i=1,...,N\right|$ to promise linearly independent parameter ensemble perturbations.

The calculating unit is used for calculating a parameter perturbation ensemble A' with a formula of

$$A'=\{F_a\gamma_1 p_1,F_a\gamma_2 p_2,...,F_a\gamma_N p_N)\in\Re^{m\times N},\gamma_i\sim N(0,S_P),$$

where A' indicates the parameter perturbation ensemble, p_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_P is a configurable variance, and F_α is a matrix defined as $F_\alpha=(f_1e_1,f_2e_2,...,f_Ne_N)$, e_i is an unit vector, and f_i is a configurable scaling variable used to adjust parameter perturbation amplitude.

The third generating unit is used for adding each perturbation vector s_i in the parameter perturbation ensemble A' to the parameter vector Θ to obtain a set of parameters: $\theta_i = \theta + \varepsilon_i$, such that a parameter ensemble A comprising N sets of parameters is obtained.

The above units are controlled to repeat working to generate the initial collection pool including N_e parameter ensembles.

The first optimization module 720 is used for performing an EnKF-based iterative optimization on the plurality of parameter ensembles in the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles. In some embodiments, the first optimization module 720 is further used for training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models; evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results; and updating the plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm. More specifically, as shown in Fig. 3, the EnKF-based iterative optimization is performed by the following steps.

At step 2-1, feature scaling is performed on a predetermined training dataset recorded as X_T and a predetermined validation dataset recorded as X_V to obtain a normalized training dataset and a normalized validation dataset.

If feature tuning is closed, a feature value of each sample in X_T and X_V is normalized to a specific range, and the normalized training data and validation data are used through the whole process of the method; if feature tuning is activated but feature selection is closed, scale factors $5_i \Subset \theta_i$ are used to scale each feature value; if feature selection is activated, scale factors are normalized into [o,l] by the formula of $\delta_i / \max(\{\delta_i\})$ and the normalized scale factors above a predetermined threshold are used to scale the feature values. Only the scaled feature values are trained and validated, but all the scale factors take part in the EnKF algorithm. If feature tuning is closed, step 2-1 is only performed at the first time, otherwise, step 2-1 should be performed every time the optimization module is called.

At step 2-2, each set of parameters θ_i are used as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble. Moreover, model selection

methods are generally used to ensure generalization performance.

At step 2-3, each sample is predicted by the models to obtain a predicting result of each model. More specifically, each sample in the normalized validation dataset X_V is predicted by using each of the plurality of models to obtain a predicting result corresponding to each model.

At step 2-4, models corresponding to all the parameters of a parameter ensemble are generated and evaluated. More specifically, for each set of parameters $\theta_i \in A$, step 2-1 to step 2-3 are repeated to generate and evaluate the model corresponding to the set of parameters. Assuming that HA is an ensemble including predicting results corresponding to the models (in the EnKF algorithm, indicating the mapping from a state space to an observation space), HA can be expressed as $HA = (H\theta_1, H\theta_2, ..., H\theta_N) \in \Re^{n_x \times N}$, where n is the number of samples in the normalized validation dataset X_V.

For each parameter ensemble A, step 2-1 to step 2-4 are repeated to obtain the predicting result ensemble HA corresponding to the parameter ensemble A.

In some embodiments, the optimization module is further used for generating an observation ensemble and an observation perturbation ensemble.

Let D denote the observation ensemble, D can be expressed as $D = (d_1, d_2, ..., d_N) \in \Re^{m \times N}$, where $d_i \in \Re^{m \times 1}$ denotes the i^{th} observation vector which holds a set of observation values. The observation vector denotes a default confidence probability. If the default confidence probability is unknown, an initial observation vector d_0 is given. The initial observation vector d_0 denotes an observation vector which holds a set of initial observation values and the observation ensemble can be generated by the following steps.

At step 3-1, a set of normalized vectors are randomly generated. If the number n of the vectors is not large, the vectors need to be orthogonalized.

At step 3-2, an observation perturbation ensemble γ is calculated according to the set of normalized vectors. Specifically, the observation perturbation ensemble is expressed as

$$r = (v_1\beta_1, v_2\beta_2, ..., v_N\beta_N) \in \Re^{m \times N}, v_i \sim N(0, S_0),$$

where $\beta_i \in \Re^{m \times 1}$ is the vector randomly generated in step3-l, v_i is a stochastic step length which is subject to the Gauss distribution, S_0 is a configurable variance.

At step 3-3, each perturbation vector $v\beta_i$ in the observation perturbation ensemble γ is added to the initial observation vector d_0 to obtain an observation vector $d_i = d_0 + v_i\beta_i$, such that an observation ensemble D including N observation vectors is generated.

Step 3-1 to step 3-3 can be repeated to obtain N_e observation ensembles and observation perturbation ensembles.

In an embodiment, each parameter ensemble is updated by the optimization module 720 using the following formula:

$$A^a = A^f + A'(HA')^T \left(HA'(HA')^T + \gamma\gamma^T\right)^{-1}(D - HA),$$

where A^f is the current parameter ensemble, A^a is the updated parameter ensemble, $A' = A - \overline{A}$ is the parameter perturbation ensemble, D is the observation ensemble, γ is the observation perturbation ensemble, HA is the predicting result ensemble, HA' is the ensemble perturbations of HA and HA' can be calculated by the following equations.

$$\overline{HA} = HAM_N, \quad HA' = HA - \overline{HA},$$

where $\overline{HA} \in \mathbb{R}^{m \times N}$ holds average values of samples in the ensemble HA, and each element in $M_N \in \mathbb{R}^{N \times N}$ is $1/N$. Since the formula of $\left(HA'(HA')^T + \gamma\gamma^T\right)^{-1}$ is expensive in computation and storage, the UR decomposition is performed for optimization by using Householder transformation. Let $I \in \mathbb{R}^{m \times N}$ denote the array loading HA'. Let $X(i,j)$ denote one element and $X(:,j)$ denote one column, and let τ denote the boundary of the relative residual and $1 \geq \tau > 0$. For each column index i valued from 1 to N, the following steps are running.

At step 4-1, residual norms of the rest columns are computed by:

$$RestNorm(k) = \left\| X(i:n,k) \right\|_2, k = 1, \ldots, N;$$

At step 4-2, if $\frac{\tau}{1-\tau}\sum_{l=1}^{i-1}\left|X(l,l)\right| > (N-i+1)\,\mathrm{Re}\,stNorm\left(\hat{k}\right)$, then let a column number $P = i$, the step 4-7 is followed, where $\hat{k}$ is the index of the column with maximum value among the calculated residual norms; otherwise let $p = i$, and the columns $x\left(:,\hat{k}\right)$ and $X(:,i)$ are exchanged.

At step 4-3, a rotating vector $\omega_i \in \mathbb{R}^n$ is initialized, where $\omega_i(k) = \begin{cases} 0, & k < i \\ X(k,i), & k \geq i \end{cases}$;

At step 4-4, $Norm = RestNorm[k^{\wedge}$ is calculated, and if $X(i,i) > 0$, then let $\omega_{\iota} = \omega_{\iota}{}^{+} Norm \times e_{;}$, where $e_{;}$ is a unit vector whose i^{th} element is 1, and the other elemrnts are O.Then $X(i)$ is updated, satisfying,

$$X(i,k) = \begin{cases} X(i,k), & k < i \\ -Norm, & k = i \\ 0, & k > i \end{cases}; \text{ otherwise,}$$

let $\omega_i = \omega_i - Norm \times e_i$ and $X(i)$ is updated, satisfying,

$$X(i,k) = \begin{cases} X(i,k), & k < i \\ +Norm, & k = i \\ 0, & k > i \end{cases}.$$

At step 4-5, $\omega_i = \omega_i / \|\omega_i\|_2$ is calculated, and for $k = i + 1 : N$, the following formula is performed:

$$X\{i : n, k) = X(i : n, k) - 2\omega_i(i : n)\left(\omega_i(i : n)^T X(i : n, k)\right).$$

At step 4-6, let $i = i + 1$, if $i > N$, step 4-7 is followed, otherwise step 4-1 is followed;

At Step 4-7, a matrix $S \in \mathbb{R}^{m \times p}$ is constructed. Specifically, the matrix is defined as $S = X(:, 1 : p)$;

After running the UR decomposition from step 4-1 to step 4-7, the following approximation is obtained:

$$\left(HA'(HA')^T + \gamma\gamma^T\right)^{-1} \approx U \begin{pmatrix} \left(\hat{S}\hat{S}^T\right)^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T,$$

where U is an orthogonal matrix, $\hat{S} \in \mathbb{R}^{p \times p}$" denotes the upper triangular matrix holding these non-zero columns of S and $\hat{S}$ is constituted by p columns with top absolute value of diagonal elements. By running the UR decomposition, the matrix U is composed by p Householder transformations and is expressed as $U = H(\alpha_1)H(\alpha_2)...H(\alpha_p)$, and the Householder transformation $H(\alpha_i)$ is defined as $H(\omega_i) = I - 2\omega_i\omega_i^T$. Then, the original

updating formula is converted as $A^a = A^f + A'(HA')^T U \begin{pmatrix} (\hat{S}\hat{S}^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T (D-HA)$.

After the UR decomposition, the calculation order for the above formula is from right to left aimed for reducing the computation and storage cost. Then, the updated parameter ensemble A^a is obtained.

If the EnKF-based iterative optimization have been performed a predetermined maximum times, the optimized parameter ensembles are obtained and used for the following performance evaluation; otherwise, go to the first optimization module 720 for the next EnKF-based iterative optimization.

The first evaluating module 730 is used for performing a basic evaluation procedure on the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool, where parameter ensembles in the first collection pool have higher performance than those in the first supplement parameter collection pool. In some embodiments, the procedure comprises the following steps.

At step 5, the plurality of first optimized parameter ensembles are evaluated respectively to obtain a performance value of each first optimized parameter ensemble. All the performance values are recorded.

At step 6, the first optimized parameter ensembles are divided into three groups according to the performance values of the plurality of first optimized parameter ensembles. More specifically, two thresholds ***threshX*** and ***thresh!*** are used to divide the plurality of first optimized parameter ensembles. Let ***score^A)*** denote the performance value of a first optimized parameter ensemble, the first optimized parameter ensembles are divided as follows.

If ***score^A)*** $\geq$ ***threshX*** , the performance of the first optimized parameter ensemble is considered as good and then the first optimized parameter ensemble is stored in the first collection pool.

If $score(A) \leq thresh!$, the performance of the first optimized parameter ensemble is considered as bad and thus the first optimized parameter ensemble is discarded;

lithresh! < ***score^A)*** < ***threshX*** , the performance of the first optimized parameter ensemble is considered as general, and parameter ensembles with general performance are randomly combined

in turn to obtain the first supplement parameter collection pool.

After evaluating all the first optimized parameter ensembles, the parameter ensembles with general performance are combined using the EnKF-based combination algorithm. Specifically, the combination comprises the following steps.

At step 7-1, two parameter ensembles A_j, A_j are randomly selected from the parameter ensembles with general performance.

At step 7-2, the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles A_j, A_j are respectively calculated by using the following formulas:

$$Q_{ij} = I + \{HA, '\}^T \left(HA_i '\{HA, \}^T + HA_j '(HA_j ')^T \right)^{-1} \left(HA_j - HA, \right) \in \Re^{N \times N},$$

$$Q_{ji} = I + \left(HA_j ' \right)^T [HA, '(HA, \}^T + HA, '\{HA, '\}^T \right)^{-1} (HA, - HA_j) \in \Re^{N \times N},$$

where I is an identity matrix, $HA,$ is the predicting result ensemble corresponding to the parameter ensemble $A,$, and $HA',$ is the ensemble perturbations of $HA,$, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and $HA!_j$ is the ensemble perturbations of HA_j.

The above formulas are also calculated by means of an optimization method which is similar to the optimization method described in the above steps 4-1 to 4-7.

At step 7-3, the refined parameter ensembles A_j and A_{ji} are respectively calculated by using formulas of $A_{ij} = \overline{A,} + A, 'Q_{ij}$ and $A_{ji} = \overline{A_j} + A_j 'Q_{ji}$.

At step 7-4, Q_y and Q_{ji} are respectively decomposed according to formulas of $Q_{ij} \sim \tilde{U}_{ij} \tilde{S}_{ij} \tilde{V}_{ij}$ and $Q_{ji} \sim \tilde{U}_{ji} \tilde{S}_{ji} \tilde{V}_{ji}$,

wherein $\tilde{U}_y$ and $\tilde{U}_{ji}$ are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, $\tilde{V}_{ij}$ are permutation matrices for interchanging columns when pivot elements are chosen.

At step 7-5, N columns with maximum absolute values of diagonal elements are respectively selected from $\tilde{S}_j$ and $\tilde{S}_{ji}$, and corresponding parameter vectors are obtained from A_j and A_{ji} according to the N columns to generate a combined parameter ensemble A^m.

At step 7-6, it is determined whether there are other parameter ensembles with general performance need to be combined, if no, subsequent steps are performed, and if yes, another two parameter ensembles with general performance are randomly selected for combination, i.e. the above steps 7-1 to 7-5 are repeated.

The updating module 740 is used for updating the initial collection pool with the first supplement parameter collection pool.

If some parameter ensembles have been discarded or combined with other parameter ensembles then go to the generating module 710 to generate new parameter ensembles randomly , and the new generated parameter ensembles and the parameter ensembles in the first supplement parameter collection pool compose the initial collection pool so that the initial collection pool include N_e parameter ensembles.

The first optimization module, the first evaluating module and the updating module are controlled to repeat working to obtain a second collection pool and a set of optimal parameters.

If the basic evolution procedure has been performed a first predetermined number of times, an optimal parameter ensemble may be obtained according to recorded performance values.The parameter ensembles satisfying $score\hat{}A) \geq threshl$ are stored in the collection pool for a fusion search procedure; otherwise, go to step SI02.

In some embodiments, the system further comprises a first combining module, a second optimization module, a second evaluating module.

The first combining module is used for randomly combining two parameter ensembles in the second collection pool in turn to obtain a second supplement parameter collection pool.

The parameter ensembles in the second collection pool are combined using the EnKF-based combination algorithm the principle of which is as described in the above steps 7-1 to 7-6.

The second optimization module is used for performing an EnKF-based iterative optimization on parameter ensembles in the second supplement parameter collection pool respectively to obtain a plurality of second optimized parameter ensembles.

The operational principle of the second optimization module is as the first optimization module.

The second evaluating module is used for evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool. The plurality of second optimized parameter ensembles are divided as follows.

At step 8, the plurality of second optimized parameter ensembles are evaluated respectively to obtain a performance value of each second optimized parameter ensemble. All the performance values are recorded.

At step 9, the second optimized parameter ensembles are divided into two groups according to the performance values of the plurality of second optimized parameter ensembles. More specifically, a threshold *thresh* is used to divide the plurality of second optimized parameter ensembles. Let *score^A)* denote the performance value of a second optimized parameter ensemble, the plurality of second optimized parameter ensembles are divided as follows.

If $score(A) \geq thresh$, the performance of the second optimized parameter ensemble is considered as good and then the optimized parameter ensemble is added in the second collection pool.

If *score^A)* $< thresh$, the performance of the second optimized parameter ensemble is considered as bad and thus the second optimized parameter ensemble is discarded.

The first combining module, the second optimization module and the second evaluating module are controlled to repeat working to obtain a new set of optimal parameters.

If the fusion search procedure is performed a second predetermined number of times (may be allowed to set as 0, which means that the fusion search procedure will be skipped).

At step 10, the output module 750 is used for outputting the set of parameters with the best performance as the optimal solutions.

In conclusion, The EnKF-based system according to embodiments of the present disclosure is effective in hyper-parameter optimization and feature tuning of machine learning algorithms. As a kind of parameters, the feature scale factors are optimized for feature enhancement and feature selection as well as for parameter optimization with the hyper-parameters of machine learning algorithms. As shown in Fig. 2, the system mainly comprises two procedures: EnKF-based iterative optimization and ensemble evolution (basic evolution and fusion search).

More specifically, based on the EnKF algorithm which can be used to solve nonlinear problems with massive parameters, the system according to embodiments of the present disclosure adopts many optimization techniques and builds an EnKF-based framework.

Firstly, by considering the hyper-parameter optimization and feature tuning of machine learning algorithms as a problem of a nonlinear system, the parameters viewed as the state of the nonlinear system are estimated by the method with the EnKF algorithm. Secondly, the system

builds an EnKF-based framework, which adopts multiple ensembles methodology to avoid trapping into local optima gradient caused by the numerical optimization technique. Then the system adopts the ensemble evolution technique to broaden search area and improve search efficiency by evaluating the parameter ensembles calculated by the EnKF algorithm, keeping the ensembles with good performance, discarding the ensembles with bad performance and merging the ensembles with general performance. After the ensemble evolution procedure, by using the fusion search technique to merge the parameter ensembles with good performance , a further search is performed, thus obtaining an approximate optimum solution in the high-dimensional parameter space. Finally, since some matrixes in the EnKF algorithm is expensive in computation and storage due to the massive data of machine learning, the UR decomposition is performed to improve efficiency of the EnKF algorithm, thus obtaining better practicability.

The system for hyper-parameter optimization and feature tuning of machine learning algorithms according to embodiments of the present disclosure is effective in searching the optimum parameter in the high-dimensional parameter space. The system can perform hyper-parameter optimization and feature tuning simultaneously, thus increasing calculation efficiency and accuracy. Besides, the feature scale factors used for handling feature data are viewed as a kind of parameters for feature tuning. The system is universal and suitable for all kinds of machine learning algorithms.

In the specification, it is to be understood that terms such as "central," "longitudinal," "lateral," "length," "width," "thickness," "upper," "lower," "front," "rear," "left," "right," "vertical," "horizontal," "top," "bottom," "inner," "outer," "clockwise," and "counterclockwise" should be construed to refer to the orientation as then described or as shown in the drawings under discussion. These relative terms are for convenience of description and do not require that the present invention be constructed or operated in a particular orientation.

In addition, terms such as "first" and "second" are used herein for purposes of description and are not intended to indicate or imply relative importance or significance or to imply the number of indicated technical features. Thus, the feature defined with "first" and "second" may comprise one or more of this feature. In the description of the present invention, "a plurality of means two or more than two, unless specified otherwise.

In the present invention, unless specified or limited otherwise, the terms "mounted," "connected," "coupled," "fixed" and the like are used broadly, and may be, for example, fixed connections, detachable connections, or integral connections; may also be mechanical or electrical connections; may also be direct connections or indirect connections via intervening structures; may also be inner communications of two elements, which can be understood by those skilled in the art according to specific situations.

In the present invention, unless specified or limited otherwise, a structure in which a first feature is "on" or "below" a second feature may include an embodiment in which the first feature is in direct contact with the second feature, and may also include an embodiment in which the first feature and the second feature are not in direct contact with each other, but are contacted via an additional feature formed therebetween. Furthermore, a first feature "on," "above," or "on top of a second feature may include an embodiment in which the first feature is right or obliquely "on," "above," or "on top of the second feature, or just means that the first feature is at a height higher than that of the second feature; while a first feature "below," "under," or "on bottom of a second feature may include an embodiment in which the first feature is right or obliquely "below," "under," or "on bottom of the second feature, or just means that the first feature is at a height lower than that of the second feature.

Reference throughout this specification to "an embodiment," "some embodiments," "one embodiment", "another example," "an example," "a specific example," or "some examples," means that a particular feature, structure, material, or characteristic described in connection with the embodiment or example is included in at least one embodiment or example of the present disclosure. Thus, the appearances of the phrases such as "in some embodiments," "in one embodiment", "in an embodiment", "in another example," "in an example," "in a specific example," or "in some examples," in various places throughout this specification are not necessarily referring to the same embodiment or example of the present disclosure. Furthermore, the particular features, structures, materials, or characteristics may be combined in any suitable manner in one or more embodiments or examples.

Although explanatory embodiments have been shown and described, it would be appreciated by those skilled in the art that the above embodiments cannot be construed to limit the present disclosure, and changes, alternatives, and modifications can be made in the embodiments without departing from spirit, principles and scope of the present disclosure.

WHAT IS CLAIMED IS:

1. A method for hyper-parameter optimization and feature tuning of machine learning algorithms, comprising:

A: randomly generating an initial collection pool including a plurality of parameter ensembles;

B: performing an EnKF-based iterative optimization on the plurality of parameter ensembles in the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles;

C: evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool;

D: updating the initial collection pool with the first supplement parameter collection pool;

E: repeating steps B, C and D a first predetermined number of times to obtain a second collection pool and a set of optimal parameters.

2. The method according to claim 1, further comprising:

randomly combining two parameter ensembles in the second collection pool in turn to obtain a second supplement parameter collection pool;

performing an EnKF-based iterative optimization on parameter ensembles in the second supplement parameter collection pool respectively to obtain a plurality of second optimized parameter ensembles;

evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool;

repeating above steps a second predetermined number of times to obtain a new set of optimal parameters.

3. The method according to claim 1, wherein step A comprises:

randomly generating a parameter vector $\Theta \in ¾^{m\times 1}$, in which each parameter of the parameter vector has a predetermined value range, wherein m is a number of parameters in the parameter vector;

randomly generating a set of normalized orthogonal vectors $\{\rho_i | \rho_i \in ¾^{m\times 1}, i = 1,...,N|$ to promise linearly independent parameter perturbations, wherein N is a number of normalized orthogonal vectors;

calculating a parameter perturbation ensemble by

$$A' = \{F_{\alpha R1}\rho_1, F_{\alpha r2}p_2, \ldots, F_{\alpha r N}p_N) \in 9i^{mxN}, r_i \sim N(0, S_P),$$

where A' indicates the parameter perturbation ensemble, p_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_p is a configurable variance, F_α is a matrix defined as $F_\alpha = (f_1 e_1, f_2 e_2, ..., f_N e_N)$, e_i is a unit vector, and f_i is a configurable scaling variable used to adjust a parameter perturbation amplitude;

adding N perturbation vectors s_i in the parameter perturbation ensemble A' respectively to the parameter vector Θ to obtain N sets of parameters $\theta_i = \theta + \varepsilon_i$, such that a parameter ensemble comprising N sets of parameters is obtained;

repeating above steps to obtain the initial collection pool including a plurality of parameter ensembles.

4. The method according to claim 3, wherein step B comprises:

training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models;

evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results;

updating the plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm.

5. The method according to claim 4, wherein training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models comprises:

performing feature scaling on the predetermined training dataset to obtain a normalized training dataset; and

using each set of parameters of each parameter ensemble as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble, and

wherein evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results comprises:

performing feature scaling on the predetermined validation dataset to obtain a normalized validation dataset;

predicting each sample in the normalized validation dataset using each of the plurality of models to obtain a predicting result corresponding to each model, so as to obtain a predicting result ensemble $HA = (H\theta_1, H\Theta_2, ... H\Theta_N) \in \Re^{nxN}$,

where HA indicates the predicting result ensemble comprising the predicting results

corresponding to the plurality of models, n is the number of samples in the normalized validation dataset, $H\theta_i$ is the predicting result corresponding to each model, $1 \le i \le N$.

6. The method according to claim 5, wherein step B further comprises:

generating an observation ensemble and an observation perturbation ensemble; and

updating the plurality of parameter ensembles respectively according to the predicting result ensemble, the observation ensemble and the observation perturbation ensemble by means of the EnKF algorithm.

7. The method according to claim 6, wherein generating an observation ensemble and an observation perturbation comprises:

randomly generating a set of vectors;

randomly generating an observation perturbation ensemble according to the set of vectors;

adding each set of perturbation vector in the observation perturbation ensemble to an initial observation vector, so as to obtain the observation ensemble.

8. The method according to claim 7, wherein the parameter ensemble is updated according to the formula of

$$A^a = A^f + A'(HA'f\left(\boldsymbol{HA}'(HA')^T + \gamma\gamma^T\right)^{-1}(D - HA),$$

where A^F is the current parameter ensemble, A^a is the updated parameter ensemble, $A' - A - \bar{A}$ is the parameter perturbation ensemble, D is the observation ensemble, Y is the observation perturbation ensemble, HA is the predicting result ensemble, and HA' is the ensemble perturbations of HA.

9. The method according to claim 1, wherein step C comprises:

evaluating the plurality of first optimized parameter ensembles respectively to obtain a performance value of each first optimized parameter ensemble;

comparing the performance value of each first optimized parameter ensemble with a first threshold and a second threshold, in which the first threshold is larger than the second threshold;

storing the first optimized parameter ensemble in the first collection pool when the performance value of the first optimized parameter ensemble is not less than the first threshold;

discarding the first optimized parameter ensemble when the performance value of the first optimized parameter ensemble is less than or equal to the second threshold;

determining that the first optimized parameter ensemble is a parameter ensemble with general performance when the performance value of the first optimized parameter ensemble is larger than the second threshold and less than the first threshold; and

randomly combining two parameter ensembles with general performance in turn to obtain the first supplement parameter collection pool.

10. The method according to claim 9, wherein randomly combining two parameter ensembles with general performance in turn to obtain the first supplement parameter collection pool comprises:

randomly selecting two parameter ensembles from the parameter ensembles with general performance;

calculating the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles according to formulas of

$$Q_{ij} = I + \{HA,'\}^T \left(HA,'\{HA,\}^T + HA_j'(HA_j')^T\right)^{-1} (HA_j - HA,) \in \Re^{N\times N},$$

$$Q_{ji} = I + (HA_j')^T \left[HA,'(HAJ')^T + HA,'\{HA,'\}^T\right)^{-1} (HA,- HA_j) \in \Re^{N\times N},$$

where I is an identity matrix, $A,$ and A_j are the two selected parameter ensembles, $HA,$ is the predicting result ensemble corresponding to the parameter ensemble $A,$, HA', is the ensemble perturbations of $HA,$, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and HA'_j is the ensemble perturbations of HA_j;

calculating the refined parameter ensembles A_j and A_{ji} respectively according to formulas of

$$A_{ij} = \overline{A_i} + A_i'Q_\beta,$$

$$A_\beta = \overline{A}_j + A_j'Q_\beta;$$

decomposing Q_{ij} and Q_{ji} respectively to obtain $Q_{ij} = C_{ij}\tilde{S}_{ij}\tilde{V}_{ij}$ and $Q_{ji} = C_{ji}\tilde{S}_{ji}\tilde{V}_{ji}$, where $\tilde{U}_{ij}$ and $\tilde{U}_j$, are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, $\tilde{V}_{ij}$ are permutation matrices;

choosing N columns with maximum absolute values of diagonal elements from $\tilde{S}_{ij}$ and

$\tilde{S}_{j;}$ respectively, obtaining corresponding parameter vectors from A_{ij} and A_{ji} according to the N columns to generate a combined parameter ensemble for storing in the first supplement parameter collection pool.

11. The method according to claim 2, wherein evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool comprises:

evaluating the plurality of second optimized parameter ensembles respectively to obtain a performance value of each second optimized parameter ensemble;

comparing the performance value of each second optimized parameter ensemble with a third threshold;

adding the second optimized parameter ensemble into the second collection pool when the performance value of the second optimized parameter ensemble is not less than the third threshold;

discarding the second optimized parameter ensemble when the performance value of the second optimized parameter ensemble is less than the third threshold.

12. A system for hyper-parameter optimization and feature tuning of machine learning algorithms, comprising:

a generating module used for randomly generating an initial collection pool including a plurality of parameter ensembles;

a first optimization module used for performing an EnKF-based iterative optimization on the plurality of parameter ensembles respectively to obtain a plurality of optimized parameter ensembles;

a first evaluating module used for evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool;

an updating module used for updating the initial collection pool with the first supplement parameter collection pool;

the first optimization module, the first evaluating module and the updating module are controlled to repeat working to obtain a second collection pool and a set of optimal parameters; and

an output module used for outputting the set of optimal parameters.

13. The system according to claim 12, further comprising:

a first combining module used for randomly combining two parameter ensembles in the second collection pool in turn to obtain a second supplement parameter collection pool;

a second optimization module used for performing an EnKF-based iterative optimization on parameter ensembles in the second supplement parameter collection pool respectively to obtain a

plurality of second optimized parameter ensembles;

a second evaluating module used for evaluating the plurality of second optimized parameter ensembles respectively to update the second collection pool; and

above modules are controlled to repeat working to obtain a new set of optimal parameters.

14. The system according to claim 12, wherein the generating module comprises:

a first generating unit used for randomly generating a parameter vector $\Theta \in ¾^{m\times 1}$, in which each parameter of the parameter vector has a predetermined value range, wherein m is a number of parameters in the parameter vector;

a second generating unit used for randomly generating a set of normalized orthogonal vectors $\left\{\rho_i \middle| \rho_i \in ¾^{m\times 1}, i = 1,...,N\right|$ to promise linearly independent parameter perturbations, wherein N is a number of normalized orthogonal vectors;

a calculating unit used for calculating a parameter perturbation ensemble by

$$A' = \{F_a\gamma_1 p_1, F_a\gamma_2 p_2, \ldots, F_a\gamma_N p_N) \in \Re^{m\times N}, \gamma_i \sim N(0, S_P),$$

where A' indicates the parameter perturbation ensemble, p_i is a normalized orthogonal vector, γ_i is a stochastic step length according to a Gauss distribution, S_P is a configurable variance, F_α is a matrix defined as $F_\alpha = (f_1 e_1, f_2 e_2, \ldots, f_N e_N)$, e_i is a unit vector, and f_i is a configurable scaling variable used to adjust a parameter perturbation amplitude; and

a third generating unit used for adding N perturbation vectors s_i in the parameter perturbation ensemble A' respectively to the parameter vector Θ to obtain N sets of parameters $\theta_i = \theta + \varepsilon_i$, such that a parameter ensemble comprising N sets of parameters is obtained.

15. The system according to claim 12, wherein the first optimization module is further used for:

training each of the plurality of parameter ensembles on a predetermined training dataset by a machine learning algorithm to generate a plurality of models;

evaluating the plurality of models respectively on a predetermined validation dataset to obtain a plurality of predicting results; and

updating the plurality of parameter ensembles respectively according to the plurality of predicting results by means of the EnKF algorithm.

16. The system according to claim 15, wherein the first optimization module is further used for:

performing feature scaling on the predetermined training dataset and the predetermined validation dataset to obtain a normalized training dataset and a normalized validation dataset;

using each set of parameters of each parameter ensemble as inputs of the machine learning algorithm to perform a training on the normalized training dataset, so as to obtain the models respectively corresponding to each set of parameters of the parameter ensemble;

predicting each sample in the normalized validation dataset using each of the plurality of models to obtain a predicting result corresponding to each model, so as to obtain a predicting result ensemble $HA = (H\theta_1, H\theta_2, \ldots H\theta_N) \in \Re^{nxN}$,

where *HA* indicates the predicting result ensemble comprising the predicting results corresponding to the plurality of models, n is the number of samples in the normalized validation dataset, $H\theta_i$ is the predicting result corresponding to each model, $1 \leq i \leq N$.

17. The system according to claim 16, wherein the first optimization module is further used for:

generating an observation ensemble and an observation perturbation ensemble; and

updating the plurality of parameter ensembles respectively according to the predicting result ensemble, the observation ensemble and the observation perturbation ensemble by means of the EnKF algorithm.

18. The system according to claim 17, wherein the first optimization module is further used for:

randomly generating a set of vectors;

randomly generating an observation perturbation ensemble according to the set of vectors;

adding each set of perturbation vector in the observation perturbation ensemble to an initial observation vector, so as to obtain the observation ensemble.

19. The system according to claim 18, wherein the parameter ensemble is updated according to the formula of

$$A^a = A^f + A'(HA'f\left(\boldsymbol{HA}'(HA')^T + \gamma\gamma^T\right)^{-1}(D - HA),$$

where A^F is the current parameter ensemble, A^a is the updated parameter ensemble, $A' - A - \bar{A}$ is the parameter perturbation ensemble, D is the observation ensemble, Y is the observation perturbation ensemble, *HA* is the predicting result ensemble, and *HA*' is the ensemble perturbations of *HA*.

20. The system according to claim 12, wherein the first evaluating module is further used for:

evaluating the plurality of first optimized parameter ensembles respectively to obtain a performance value of each first optimized parameter ensemble;

comparing the performance value of each first optimized parameter ensemble with a first threshold and a second threshold, in which the first threshold is larger than the second threshold;

storing the first optimized parameter ensemble in the first collection pool when the performance value of the first optimized parameter ensemble is not less than the first threshold;

discarding the first optimized parameter ensemble when the performance value of the first optimized parameter ensemble is less than or equal to the second threshold;

determining that the first optimized parameter ensemble is a parameter ensemble with general performance when the performance value of the first optimized parameter ensemble is larger than the second threshold and less than the first threshold; and

randomly combining two parameter ensembles with general performance in turn to obtain the first supplement parameter collection pool.

21. The system according to claim 20, wherein the first evaluating module is further used for:

randomly selecting two parameter ensembles from the parameter ensembles with general performance;

calculating the ensemble updating gains Q_{ij} and Q_{ji} of the two parameter ensembles according to formulas of

$$\text{¿¾} = I + \{HA,'\}^T (HA,'\{HA,\}^T + HA_j'(HA_j')^T)^{-1} (HA_j - HA,) \in \Re^{N\times N},$$

$$Q_{ji} = I + (HA_j')^T [HA,'(HAJ')^T + HA,'\{HA,'\}^T)^{-1} (HA, - HA_j) \in \Re^{N\times N},$$

where I is an identity matrix, $A,$ and A_j are the two selected parameter ensembles, $HA,$ is the predicting result ensemble corresponding to the parameter ensemble $A,$, $HA',$ is the ensemble perturbations of $HA,$, HA_j is the predicting result ensemble corresponding to the parameter ensemble A_j, and $HA,'$ is the ensemble perturbations of $HA,$;

calculating the refined parameter ensembles A_{ij} and A_{ji} respectively according to formulas of

$$A_{ij} = \overline{A_i} + A_i'¾ \quad ,$$

$$A_{ji} = \overline{A_j} + A_j'Q_{ji};$$

decomposing Q_{ij} and Q_{ji} respectively to obtain $Q_{ij} - C_{ij}\tilde{S}_{ij}\tilde{V}_{ij}$ and $Q_{ji} - C_{ji}\tilde{S}_{ji}\tilde{V}_{ji}$, where

$\tilde{U}_y$ and $\ddot{U}_\beta$ are orthogonal matrices, $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ are upper triangular matrices with diagonal elements ordered by descending absolute values, $\tilde{V}_{ij}$ are permutation matrices;

choosing N columns with maximum absolute values of diagonal elements from $\tilde{S}_{ij}$ and $\tilde{S}_{ji}$ respectively, obtaining corresponding parameter vectors from A_{ij} and A_{ji} according to the N columns to generate a combined parameter ensemble for storing in the supplement parameter collection pool.

22. The system according to claim 13, wherein the second evaluating module is further used for:

evaluating the plurality of second optimized parameter ensembles respectively to obtain a performance value of each second optimized parameter ensemble;

comparing the performance value of each second optimized parameter ensemble with a third threshold;

adding the second optimized parameter ensemble into the second collection pool when the performance value of the second optimized parameter ensemble is not less than the third threshold;

discarding the second optimized parameter ensemble when the performance value of the second optimized parameter ensemble is less than the third threshold.

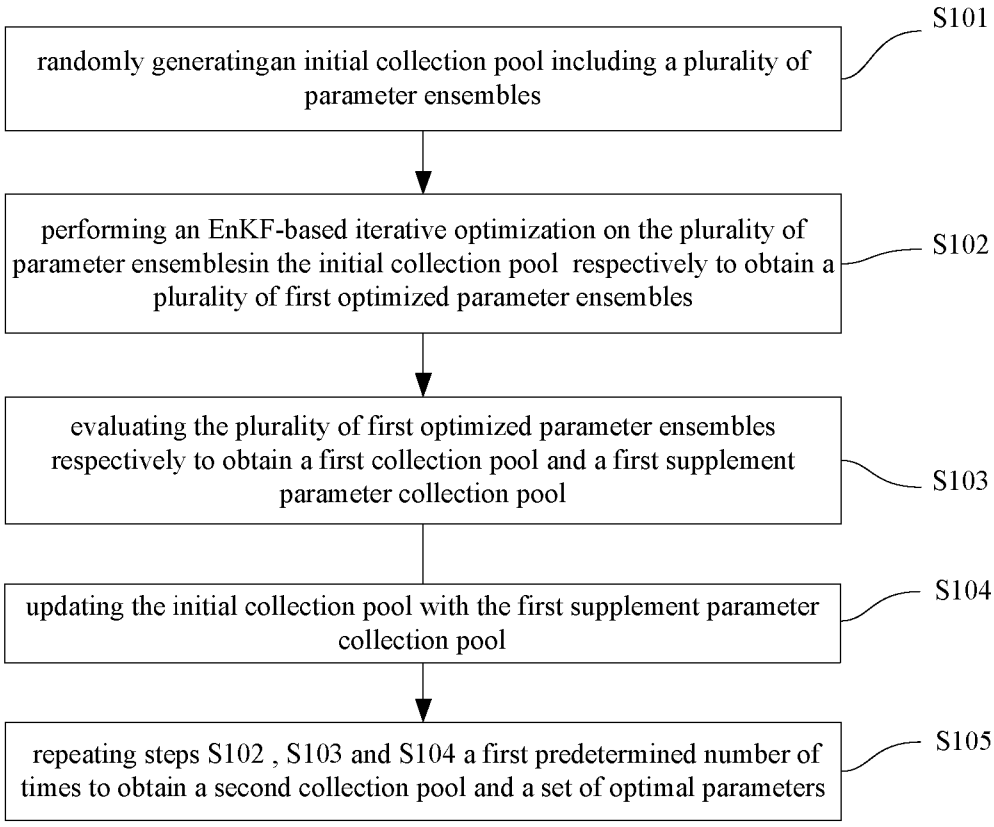
randomly generatingan initial collection pool including a plurality of parameter ensembles
S101
performing an EnKF-based iterative optimization on the plurality of parameter ensemblesin the initial collection pool respectively to obtain a plurality of first optimized parameter ensembles
S102
evaluating the plurality of first optimized parameter ensembles respectively to obtain a first collection pool and a first supplement parameter collection pool
S103
updating the initial collection pool with the first supplement parameter collection pool
S104
repeating steps S102 , S103 and S104 a first predetermined number of times to obtain a second collection pool and a set of optimal parameters
S105

Fig. 1

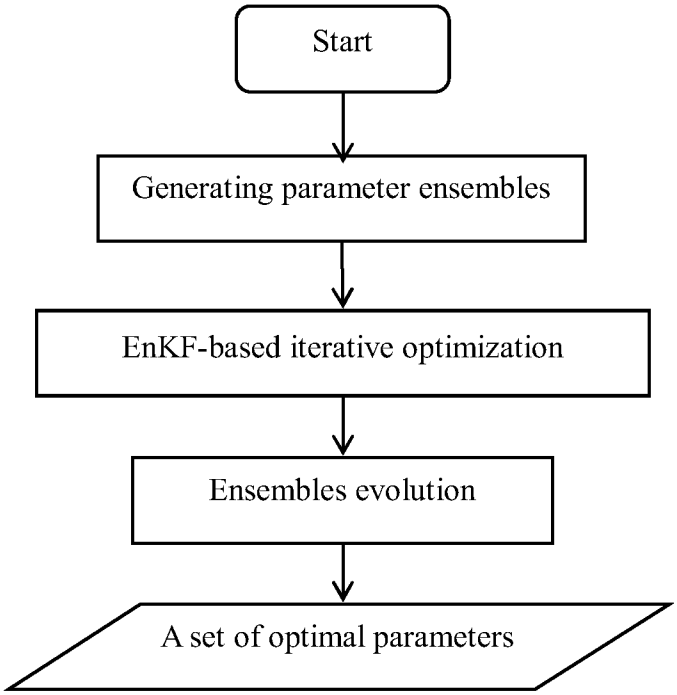
Start
Generating parameter ensembles
EnKF-based iterative optimization
Ensembles evolution
A set of optimal parameters

Fig. 2

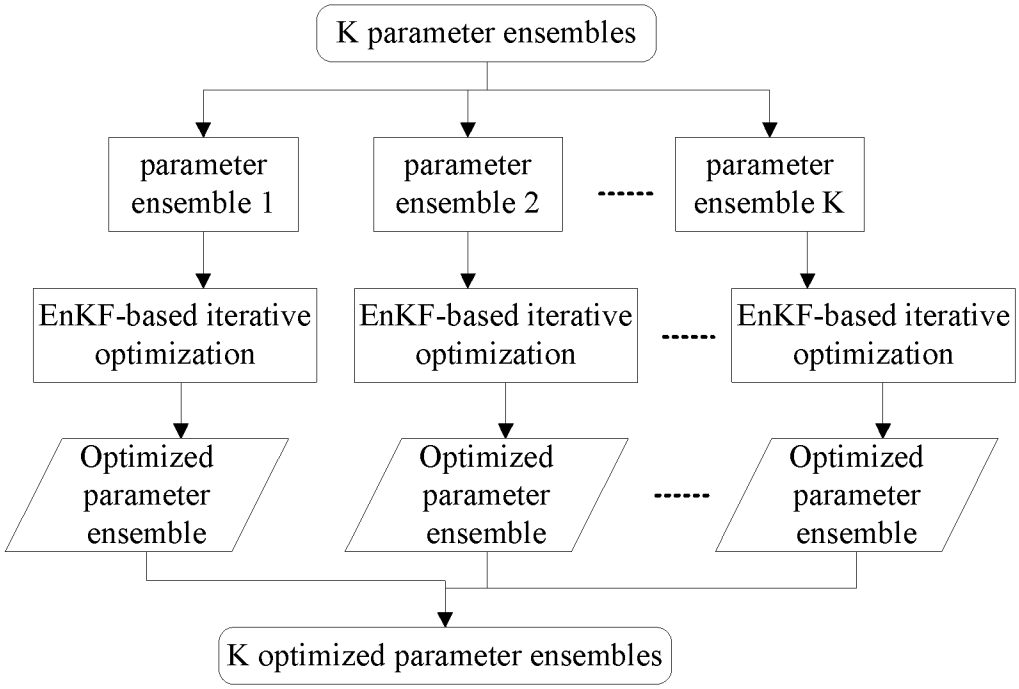
K parameter ensembles
parameter ensemble 1
parameter ensemble 2
parameter ensemble K
EnKF-based iterative optimization
EnKF-based iterative optimization
EnKF-based iterative optimization
Optimized parameter ensemble
Optimized parameter ensemble
Optimized parameter ensemble
K optimized parameter ensembles

Fig. 3

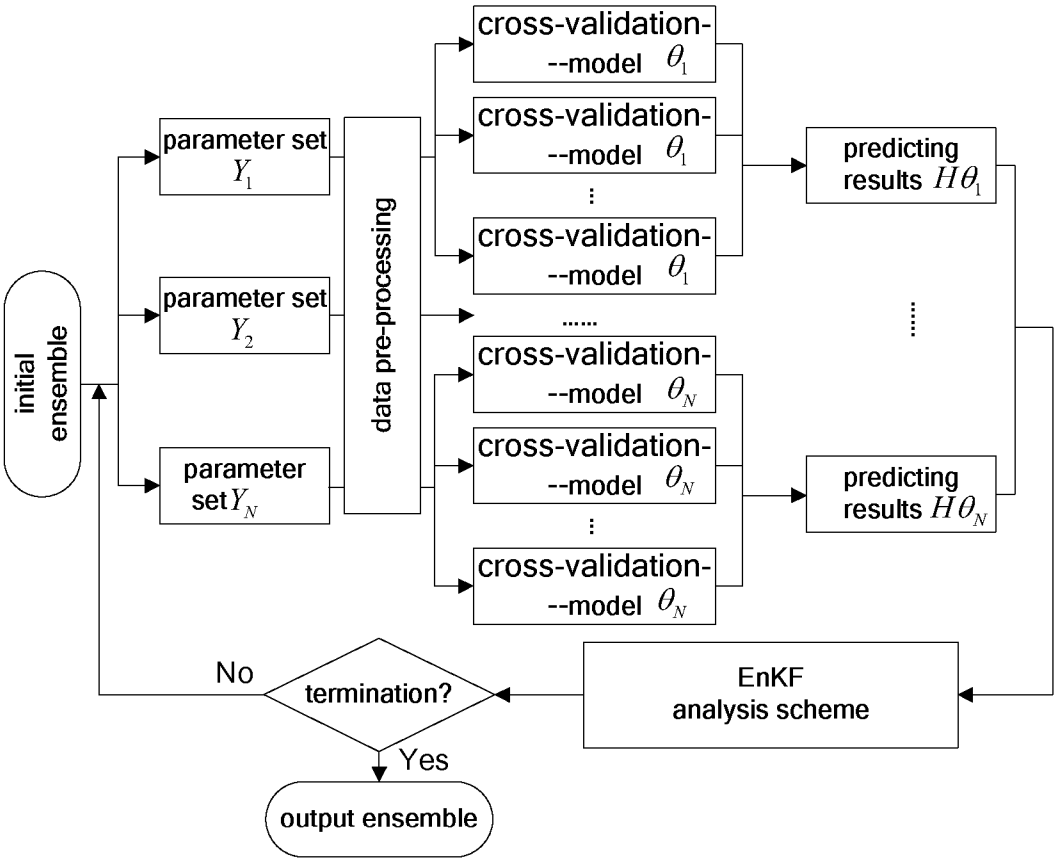
initial ensemble
parameter set Y_1
parameter set Y_2
parameter set Y_N
data pre-processing
cross-validation--model θ_1
cross-validation--model θ_1
cross-validation--model θ_1
.....
cross-validation--model θ_N
cross-validation--model θ_N
cross-validation--model θ_N
predicting results $H\theta_1$
predicting results $H\theta_N$
EnKF analysis scheme
termination?
No
Yes
output ensemble

Fig. 4

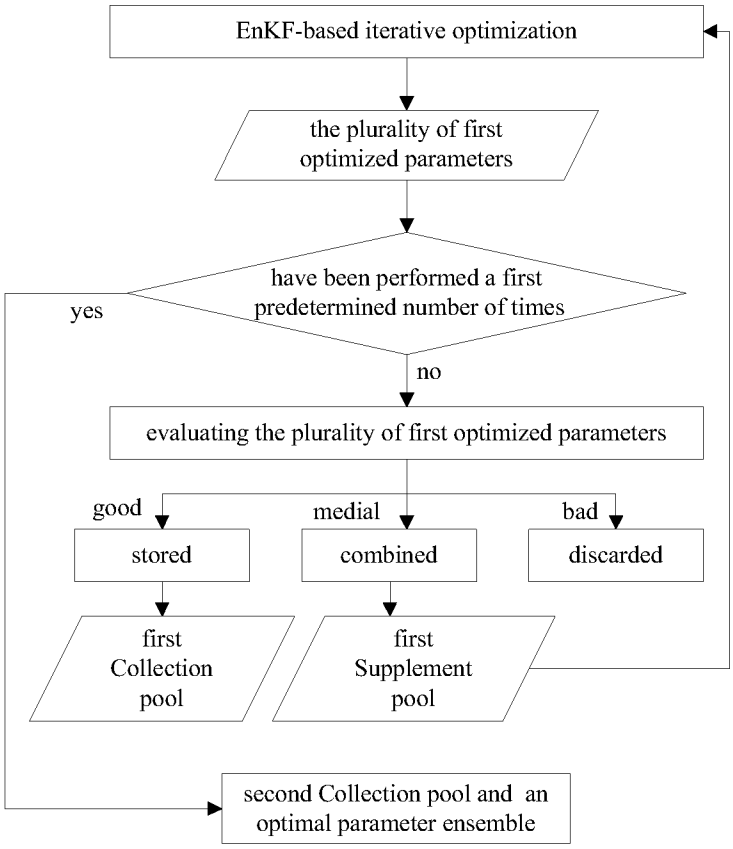
EnKF-based iterative optimization
the plurality of first optimized parameters
have been performed a first predetermined number of times
yes
no
evaluating the plurality of first optimized parameters
good
medial
bad
stored
combined
discarded
first Collection pool
first Supplement pool
second Collection pool and an optimal parameter ensemble

Fig. 5

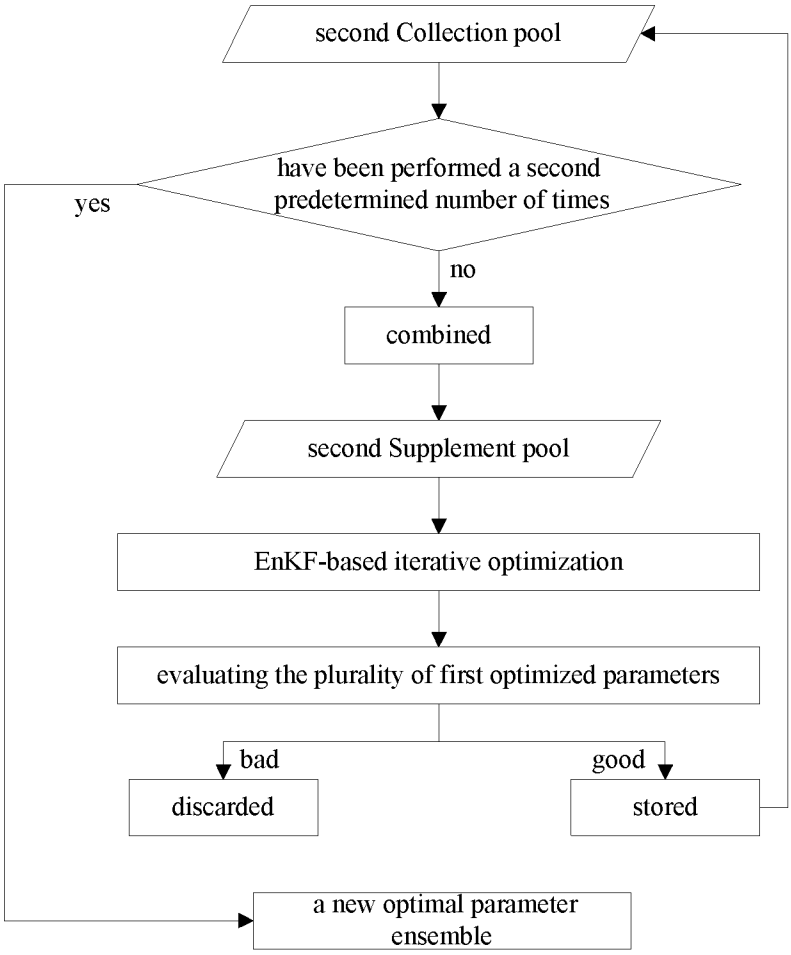
second Collection pool
have been performed a second predetermined number of times
yes
no
combined
second Supplement pool
EnKF-based iterative optimization
evaluating the plurality of first optimized parameters
bad
good
discarded
stored
a new optimal parameter ensemble

Fig. 6

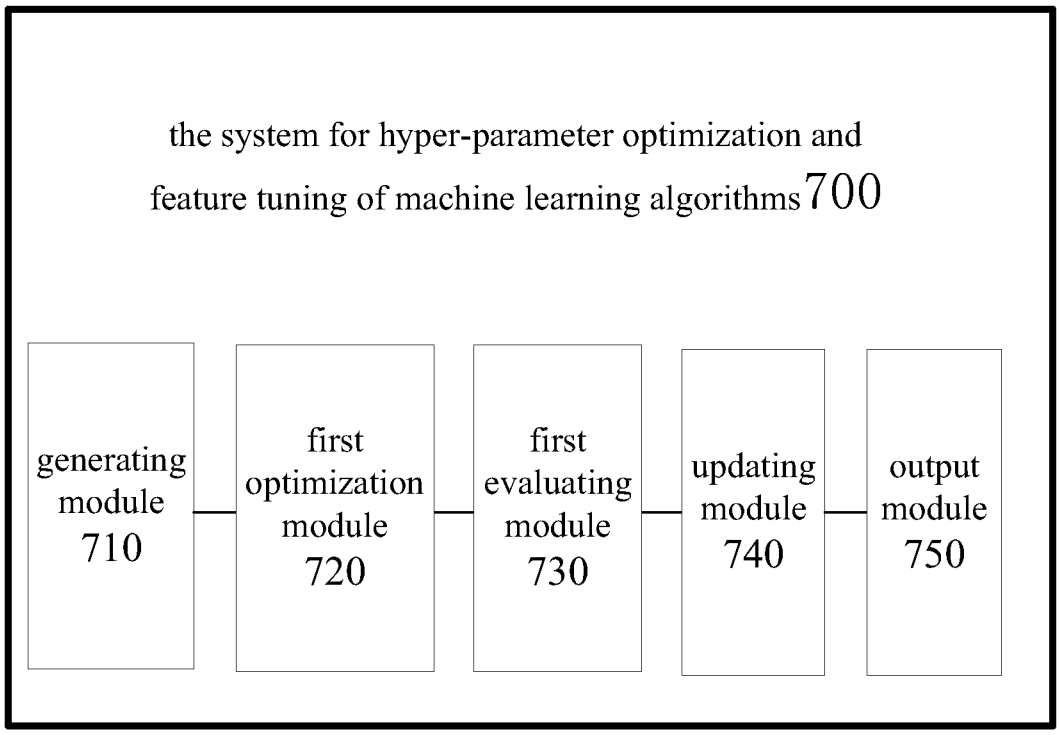
the system for hyper-parameter optimization and feature tuning of machine learning algorithms 700
generating module 710
first optimization module 720
first evaluating module 730
updating module 740
output module 750

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/090050

A. CLASSIFICATION OF SUBJECT MATTER

G06F 19/00(201 1.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

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

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

CNKI; CPRSABS; CNTXT; VEN: machine, learning, feature, parameter, optimization, iterative, evaluat+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 103744978 A (UNTV QINGHUA) 23 April 2014 (2014-04-23) see the whole document	1-22
A	CN 101782976 A (UNTV NANJING POSTS & TELECOL) 21 July 2010 (2010-07-21) see the whole document	1-22
A	US 20060224532 A1 (UNIV CASE WESTERN RESERVE) 05 October 2006 (2006-10-05) see the whole document	1-22

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 February 2015	**13 February 2015**
Name and mailing address of the ISA/CN **STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN)** **6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China** Facsimile No. **(86-10)62019451**	Authorized officer **FU,Yao** Telephone No. **(86-10)62411887**

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/090050

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 103744978 A	23 April 2014	None	
CN 101782976 A	21 May 2010	CN 101782976 B	10 April 2013
US 20060224532 A1	05 October 2006	None	