STRATEGY SEARCHING IN STRATEGIC INTERACTION BETWEEN PARTIES

TECHNICAL FIELD

[0001] This specification relates to strategy searching in strategic interaction between two or more parties.

BACKGROUND

[0002] Strategic interaction between two or more parties can be modeled by a game that involves two or more parties (also referred to as players). In an Imperfect Information Game (IIG) that involves two or more players, a player only has partial access to the knowledge of her opponents before making a decision. This is similar to real-world scenarios, such as trading, traffic routing, and public auction. Many real life scenarios can be represented as IIGs, such as commercial competition between different companies, bidding relationships in auction scenarios, game relationships between a fraud party and an anti-fraud party.

[0003] Methods for solving an IIG is of great economic and societal benefits. Due to the hidden information, a player has to reason under the uncertainty about her opponents' information, and she also needs to act so as to take advantage of her opponents' uncertainty about her own information.

SUMMARY

[0004] Implementations of this specification include computer-implemented methods for strategy searching in strategic interaction between parties. More specifically, this specification describes examples of sampling schemes for performing a counterfactual regret minimization (CFR) algorithm in solving an imperfect information game (IIG), which can reduce the computational complexity and variance while improving the convergence speed of the CFR algorithm. This specification also describes techniques for performing counterfactual regret minimization (CFR) with neural networks, which can save memory space and provide faster convergence due to the generalization ability of the neural networks.

[0005] The subject matter described in this specification as implemented in particular embodiments realizes one or more of the following technical effects and advantages. In some embodiments, the described sampling techniques can help find better strategies of real-world scenarios such as resource allocation, product/service recommendation, cyber-attack prediction and/or prevention, traffic routing, fraud management, etc. that can be modeled or represented by strategic interaction between parties, such as, an IIG that involves two or more parties in a more efficient manner. In some embodiments, the described techniques can improve computational efficiency and reduce the computational load of counterfactual regret minimization (CFR) algorithm

in finding the best strategies of the real-world scenarios modeled by the IIG. In some embodiments, the described sampling techniques can provide a lower variance than outcome sampling, while being more memory efficient than external sampling. In some embodiments, the described techniques can improve the convergence speed of the CFR algorithm in finding Nash equilibrium for solving a game that represents one or more real-world scenarios. In some embodiments, the described techniques provide more balanced and comprehensive information of a game tree that represents the IIG so that the CFR algorithm can have a smaller variance and a faster convergence speed. In some embodiments, the described techniques save memory space and provide faster convergence by using neural networks in connection with the CFR algorithm. In some embodiments, the described techniques may only need a small amount of memory space for each iteration of the CFR algorithm.

[0006] This specification also provides one or more non-transitory computer-readable storage media coupled to one or more processors and having instructions stored thereon which, when executed by the one or more processors, cause the one or more processors to perform operations in accordance with embodiments of the methods provided herein.

[0007] This specification further provides a system for implementing the methods provided herein. The system includes one or more processors, and a computer-readable storage medium coupled to the one or more processors having instructions stored thereon which, when executed by the one or more processors, cause the one or more processors to perform operations in accordance with embodiments of the methods provided herein.

[0008] It is appreciated that methods in accordance with this specification may include any combination of the aspects and features described herein. That is, methods in accordance with this specification are not limited to the combinations of aspects and features specifically described herein, but also include any combination of the aspects and features provided.

[0009] The details of one or more embodiments of this specification are set forth in the accompanying drawings and the description below. Other features and advantages of this specification will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a diagram illustrating examples of partial game trees in one-card poker in accordance with embodiments of this specification.

[0011] FIG. 2 is a diagram illustrating examples of different sampling schemes in accordance with embodiments of this specification.

[0012] FIG. 3 is a pseudocode of an example of robust sampling Monte Carlo CFR (MCCFR) in accordance with embodiments of this specification.

[0013] FIG. 4 is a diagram illustrating an example of a double neural CFR algorithm applied to a game tree in accordance with embodiments of this specification.

[0014] FIG. 5 is a pseudocode of an example of a double neural CFR algorithm in accordance with embodiments of this specification.

[0015] FIG. 6 is a pseudocode of an example of an algorithm for optimizing a neural network in connection with a double neural CFR algorithm in accordance with embodiments of this specification.

[0016] FIG. 7 is a pseudocode of an example of a mini-batch MCCFR algorithm in accordance with embodiments of this specification.

[0017] FIG. 8 is a flow chart illustrating an example of a sampling process for performing MCCFR in accordance with embodiments of this specification.

[0018] FIG. 9 is a flow chart illustrating an example of a double neural CFR algorithm in accordance with embodiments of this specification.

[0019] FIG. 10 depicts a block diagram illustrating an example of a computer-implemented system used to provide computational functionalities associated with described algorithms, methods, functions, processes, flows, and procedures in accordance with embodiments of this specification.

[0020] FIG. 11 depicts examples of modules of an apparatus in accordance with embodiments of this specification.

[0021] FIG. 12 depicts examples of modules of another apparatus in accordance with embodiments of this specification.

[0022] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0023] Implementations of this specification include computer-implemented methods for strategy searching in strategic interaction between parties, for example, by solving an imperfect information game (IIG). The IIG can represent one or more real-world scenarios such as resource allocation, product/service recommendation, cyber-attack prediction and/or prevention, traffic routing, fraud management, etc. that involves two or more parties (also referred to as players), where each party may have incomplete or imperfect information about the other party's decisions. More specifically, this specification describes examples of sampling schemes for performing a counterfactual regret minimization (CFR) algorithm in solving an IIG, which can reduce the computational complexity and variance while improving the convergence speed of the CFR algorithm. This specification also describes techniques for performing counterfactual regret minimization (CFR) with neural networks,

which can save memory space and provide faster convergence due to the generalization ability of the neural networks.

[0024] Nash equilibrium is a typical solution for an IIG that involves two or more players. Counterfactual Regret Minimization (CFR) is an algorithm designed to approximately find Nash equilibrium for large games. CFR tries to minimize overall counterfactual regret. It is proven that the average of the strategies in all iterations would converge to a Nash equilibrium. When solving a game, CFR in its original form (also referred to as original CFR, standard CFR, vanilla CFR, or simply, CFR) traverses the entire game tree in each iteration. Thus, the original CFR requires large memory for large, zero-sum extensive games such as heads-up no-limit Texas Hold'em. In some instances, the original CFR may not handle large games with limited memory.

[0025] A Monte Carlo CFR (MCCFR) was introduced to minimize counterfactual regret. The MCCFR can compute an unbiased estimation of counterfactual value and avoid traversing the entire game tree. Since only subsets of all information sets are visited in each iteration, MCCFR requires less memory than the original CFR.

[0026] MCCFR can be performed with an outcome sampling algorithm or an external sampling algorithm. The outcome sampling algorithm in MCCFR has a large variance, and it is difficult to converge to an approximate Nash equilibrium solution in fewer iteration steps. The external sampling algorithm in MCCFR has a smaller variance than the outcome sampling algorithm, but this method is similar to the disadvantage of CFR. When the game tree is large, it requires a very large memory space and cannot be extended to a complex large-scale IIG.

[0027] This specification discloses a robust sampling scheme. In the robust sampling scheme, each player uses a uniform sampling method to sample at a current decision point, and the other party samples according to a corresponding strategy. The reach probability corresponding to different iterations can be fixed. It can be proved that the robust sampling scheme has a smaller variance than the outcome sampling scheme in MCCFR, while being more memory efficient than the external sampling. In some embodiments, the robust sampling scheme can make the MCCFR solve Nash equilibrium with faster convergence.

[0028] This specification discloses a depth-dependent sampling scheme. The depth-dependent sampling scheme can allocate a higher sampling probability to a state closer to a terminal state than another state further from the terminal state (or closer to an initial or beginning state). In some embodiments, the depth-dependent sampling scheme can allow more states closer to a terminal state to be sampled, providing more well-around information of the IIG and thus improve the convergence rate of the MCCFR compared to existing sampling schemes.

[0029] This specification further discloses a double neural CFR algorithm. Existing CFR methods such as CFR and MCCFR use two large tabular-based memories to record the cumulative regret and average strategy for all information sets. Such tabular representation makes these methods difficult to apply to large extensive-form games with limited time and space.

[0030] By contrast, the double neural CFR algorithm uses two neural networks to compute approximate Nash equilibrium of IIG. For example, one of the neural networks can be used to learn the cumulative regret and the other one can be used to learn the cumulative numerator of the average strategy profile. With the help of these two networks, the double neural CFR algorithm does not need to use two large tabular-based memories. Based on the generalization ability of the compact neural network, the cumulative regret and the average strategy can be learned and produced. The disclosed double neural CFR algorithm can keep the benefit of MCCFR in requiring less computational load yet without the need for two large tabular memories. The disclosed double neural CFR algorithm can be used in large games even with memory constraints. In some embodiments, the double neural method can achieve a lower exploitability with fewer iterations than existing techniques. In addition, in some embodiments, the double neural CFR can also continually improve after initialization from a poor tabular strategy.

[0031] In some embodiments, the described techniques can be used, for example, in AI poker, recommendation platforms, and many other AI and machine learning applications. The described techniques use the Monte Carlo method and does not require variables for the entire game tree.

[0032] In some embodiments, an extensive-form game with a finite set $N = \{0,1,...,n-1\}$ of players can be represented as follows. Define h^v_i as a hidden variable of player i in an IIG. For example, in a poker game, h^v_i can refer to the private cards of player i. H refers to a finite set of histories. Each member $h = (h_i^v)_{i=0,1,...,n-1}(a_l)_{l=0,...,L-1} = h_0^v h_1^v \dots h_{n-1}^v a_0 a_1 \dots a_{L-1}$ of H denotes a possible history (or state), which includes each player's hidden variable and L actions taken by players including chance. For player i, h also can be denoted as $h_i^v h_{-i}^v a_0 a_1 \dots a_{L-1}$, where h_{-i}^v refers to the opponent's hidden variables. The empty sequence $\emptyset$ is a member of H. The expression $h_j \sqsubseteq h$ denotes that h_j is a prefix of h, where $h_j = (h_i^v)_{i=0,1,...,n-1}(a_l)_{l=1,...L'-1}$ and $0 < L' < L$. $Z \subseteq H$ denotes the terminal histories and any member $z \in Z$ is not a prefix of any other sequences. $A(h) = \{a : ha \in H\}$ is the set of available actions after non-terminal history $h \in H \setminus Z$. A player function P assigns a member of $N \cup \{c\}$ to each non-terminal history, where c denotes the chance player identifier (ID), which typically can be, for example, -1. $P(h)$ is the player who takes an action after history h.

[0033] I_i of a history $\{h \in H : P(h) = i\}$ is an information partition of player i. A set $I_i \in I_i$ is an information set of player i. $I_i(h)$ refers to information set I_i at state h. In some embodiments, I_i could

only remember the information observed by player i including player i's hidden variable and public actions. Therefore I_i indicates a sequence in the IIG, *i.e.*, $h^v{}_i a_0 a_1 ... a_{L-1}$. In some embodiments, for $I_i \in \mathrm{I}_i$ and for any $h \in I_i$, the set $A(h)$ can be denoted by $A(I_i)$ and the player $P(h)$ is denoted by $P(I_i)$. For each player $i \in N$, a utility function $u_i(z)$ defines a payoff of a terminal state z. A more detailed explanation of these notations and definitions will be discussed below including an example shown in FIG. 1.

[0034] FIG. 1 is a diagram 100 illustrating examples of partial game trees 102 and 104 in One-Card Poker in accordance with embodiments of this specification. One-Card Poker is a two-players IIG of poker. One-Card Poker is an example of an extensive-form game. The game rules are defined as follows. Each player is dealt one card from a deck of X cards. The first player can pass or bet. If the first player bet, the second player can call or fold. If the first player pass, the second player can pass or bet. If second player bet, the first player can fold or call. The game ends with two passes, a call, or a fold. The fold player will lose 1 chips. If the game ended with two passes, the player with higher card win 1 chips. If the game end with a call, the player with higher card win 2 chips.

[0035] A game tree is a directed graph. The nodes of the game tree represent positions (or states of a player) in a game and edge of the game tree represent can represent moves or actions of a player of the game. In FIG. 1, z_i denotes a terminal node, representing a terminal state, and h_i denotes a non-terminal node. Each of the partial game trees 102 and 104 has a root node h_0 representing a chance. There are 19 distinct nodes in the first partial tree 102, corresponding to 9 non-terminal nodes h_i including chance h_0 and 10 terminal nodes z_i in the left tree.

[0036] In the first partial tree 102, two players (player 0 and player 1) are dealt (queen, jack) as shown as "0:Q 1:J" in the left subtree and (queen, king) as shown as "0:Q 1:K" in the right subtree.

[0037] The trajectory from the root node to each node is a history of actions. Actions are presented by letters (e.g., F, C, P, and B) or representations (e.g., "0:Q 1:J") next to edges (denoted by arrows) of the game tree. The letters F, C, P, B refer to fold, call, pass, and bet, respectively.

[0038] In an extensive-form game, h_i refers to the history of actions. For example, as illustrated in the first partial tree 102, h_3 includes actions 0:Q, 1:J and P. h_7 includes actions 0:Q, 1:J, P and B. h_8 includes actions 0:Q, 1:K, P and B. In the first partial tree 102, $h_3 \sqsubseteq h_7$, that is, h_3 is a prefix of h_7. $A(h_3) = \{\mathrm{P,B}\}$ indicating that the set of available actions after non-terminal history h_3 are P and B. $P(h_3) = 1$ indicating that the player who takes an action after history h_3 is player 1.

[0039] In the IIG, the private card of player 1 is invisible to player 0, therefore h_7 and h_8 are actually the same for player 0. An information set can be used to denote the set of these

undistinguished states. Similarly, h_1 and h_2 are in the same information set. For the right partial tree 104, h'_3 and h'_5 are in the same information set; h'_4 and h'_6 are in the same information set.

[0040] Typically, any $I_i \in \mathrm{I}_i$ could only remember the information observed by player i including player i's hidden variables and public actions. For example, as illustrated in the first partial tree 102, the information set of h_7 and h_8 indicates a sequence of 0:Q, P, and B. Because h_7 and h_8 are undistinguished by player 0 in the IIG, if I_0 is the information set of h_7 and h_8, $I_0 = I_0(h_7) = I_0(h_8)$.

[0041] A strategy profile $\sigma = \{\sigma_i | \sigma_i \in \Sigma_i, i \in N\}$ is a collection of strategies for all players, where Σ_i is the set of all possible strategies for player i. σ_{-i} refers to strategy of all players other than player i. For player $i \in N$, the strategy $\sigma_i(I_i)$ is a function, which assigns an action distribution over $A(I_i)$ to information set I_i. $\sigma_i(a|h)$ denotes the probability of action a taken by player $i \in N \cup \{c\}$ at state h. In an IIG, if two or more states have the same information set, the two or more states have a same strategy. That is, $\forall h_1, h_2 \in I_i$, $I_i = I_i(h_1) = I_i(h_2)$, $\sigma_i(I_i) = \sigma_i(h_1) = \sigma_i(h_2)$, $\sigma_i(a|I_i) = \sigma_i(a|h_1) = \sigma_i(a|h_2)$. For example, I_0 is the information set of h_7 and h_8, $I_0 = I_0(h_7) = I_0(h_8)$, $\sigma_0(I_0) = \sigma_0(h_7) = \sigma_0(h_8)$, $\sigma_0(a|I_0) = \sigma_0(a|h_7) = \sigma_0(a|h_8)$. In FIG. 1, the same color other than gray for each state in the same information set.

[0042] For player i, the expected game utility of the strategy profile σ is denoted as $u_i^{\sigma} = \sum_{z \in Z} \pi^{\sigma}(z) u_i(z)$, which is the expected payoff of all possible terminal nodes. Given a fixed strategy profile σ_{-i}, any strategy $\sigma_i^* = \arg\max_{\sigma'_i \in \sum_i} u_i^{(\sigma'_i, \sigma - i)}$ of player i that achieves maximize payoff against π_{-i}^{σ} is a best response. For two players' extensive-form games, a Nash equilibrium is a strategy profile $\sigma^* = (\sigma_0^*, \sigma_1^*)$ such that each player's strategy is a best response to the opponent. An ϵ-Nash equilibrium is an approximation of a Nash equilibrium, whose strategy profile σ^* satisfies: $\forall_i \in N, u_i^{\sigma_i} + \epsilon \geq max_{\sigma'_i \in \sum_i} u_i^{(\sigma'_i, \sigma - i)}$.

[0043] Exploitability of a strategy σ_i can be defined as $\epsilon_i(\sigma_i) = u_i^{\sigma^*} - u_i^{(\sigma_i, \sigma_{-i}^*)}$. A strategy is unexploitable if $\epsilon_i(\sigma_i) = 0$. In large two player zero-sum games such as poker, $u_i^{\sigma^*}$ can be intractable to compute. However, if the players alternate their positions, the value of a pair of games is zero, *i.e.,* $u_0^{\sigma^*} + u_1^{\sigma^*} = 0$. The exploitability of strategy profile σ can be defined as $\epsilon(\sigma) = (u_1^{(\sigma_0, \sigma_1^*)} + u_0^{(\sigma_0^*, \sigma_1)})/2$.

[0044] For iterative methods such as CFR, σ^t can refer to the strategy profile at the t-th iteration. The state reach probability of history h can be denoted by $\pi^{\sigma}(h)$ if players take actions according to σ. For an empty sequence $\pi^{\sigma}(\emptyset) = 1$. The reach probability can be decomposed into $\pi^{\sigma}(h) = \prod_{i \in N \cup \{c\}} \pi_i^{\sigma}(h) = \pi_i^{\sigma}(h) \pi_{-i}^{\sigma}(h)$ according to each player's contribution, where

and the weighted utility $u_i^{rs(k)}(z)$ can be a constant number in each iteration given the sampling profile $\sigma^{rs(k)}$, which has a low variance. In addition, because the weighted utility no longer requires explicit knowledge of the opponent's strategy, robust sampling can be used for online regret minimization.

[0065] To simplify notations, let $k = max$ refer to $k = max_{I_i \in \mathrm{I}}|A(I_i)|$. If $k = max$ and player i randomly selects k actions according to a discrete uniform distribution $unif(0,|A(I_i)|)$ at information set I_i, $\forall_i \in N, \forall I_i \in \mathrm{I}_i, \forall_a \in A(I_i), \sigma_i^{rs(k)}(a|I_i) \sim unif(0, |A(I_i)|)$, then robust sampling can be similar to external sampling when $k = max_{I_i \in \mathrm{I}}|A(I_i)|$.

[0066] If $k = 1$ and $\sigma_i^{rs(k)} = \sigma_i$, only one history z is sampled in this case, then $u_i^{rs(1)}(z) = \frac{u_i(z)}{\pi^{\sigma^{rs(k)}}(z)}$, $\exists h \in I_i$, for $a \in A^{rs(k)}(I_i)$,

$$\tilde{r}_i^{\sigma}\left((a|I_i)\middle|Q_j\right) = \tilde{r}_i^{\sigma}\left((a|h)\middle|Q_j\right) = \sum_{z \in Q_j, ha \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(ha, z) u_i^{rs}(z) - \sum_{z \in Q_j, h \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(h, z) u_i^{rs}(z)$$

$$= \frac{(1 - \sigma_i(a|h)) u_i(z)}{\pi_i^{\sigma}(ha)}$$

[0067] If action a is not sampled at state h, i.e., $a \notin A^{rs(k)}(I_i)$, the regret is $\tilde{r}_i^{\sigma}\left((a|h)\middle|j\right) = 0 - \tilde{v}_i^{\sigma}(h|j)$. In this case, robust sampling is similar to the outcome sampling when $k = 1$ and $\sigma_i^{rs(k)} = \sigma_i$.

[0068] If $k = 1$, and player i randomly selects one action according to discrete uniform distribution $unif(0,|A(I_i)|)$ at information set I_i, . Then robust sampling can be similar to outcome sampling. For example, if $k = 1$, and player i randomly selects one action according to a discrete uniform distribution $unif(0,|A(I_i)|)$ at information set I_i, then $u_i^{rs(1)}(z) = \frac{u_i(z)}{\pi^{\sigma^{rs(k)}}(z)}$ is a constant, $\exists h \in I_i$, for $a \in A^{rs(k)}(I_i)$,

$$\tilde{r}_i^{\sigma}\left((a|I_i)\middle|Q_j\right) = \sum_{z \in Q_j, ha \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(ha, z) u_i^{rs}(z) - \sum_{z \in Q_j, h \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(h, z) u_i^{rs}(z)$$

$$= (1 - \sigma_i(a|h)) \pi_i^{\sigma}(ha, z) u_i^{rs(1)}(z).$$

[0069] If action a is not sampled at state h, i.e., $a \notin A^{rs(k)}(I_i)$, the regret is $\tilde{r}_i^{\sigma}\left((a|h)\middle|j\right) = 0 - \tilde{v}_i^{\sigma}(h|j)$. Compared to outcome sampling, the robust sampling in this case has a lower variance because of the constant $u_i^{rs(1)}(z)$.

[0070] FIG. 3 is a pseudocode 300 of an example of robust sampling MCCFR in accordance with embodiments of this specification. As shown in lines 1-5 of the pseudocode 300, the overall robust sampling MCCFR is an iterative algorithm with an input of the total number of iterations, t. Within

each iteration t, a function robust sampling MCCFR (RS-MCCFR) is called for player 0 and player 1 (as shown in lines 3 and 4) to update the cumulative regret $R^t(\cdot|I_i)$ and average strategy numerator $S^t(a|I_i)$. The function RS-MCCFR can be defined as shown in lines 6-30 of the pseudocode 300. The function RS-MCCFR return the counterfactual value of each information set as the output. In some embodiments, the counterfactual value can be used to calculate the counterfactual regret. As such, the cumulative regret and time-average strategy can be obtained accordingly.

[0071] Specifically, the function RS-MCCFR samples the actions according to the robust sampling scheme as described above in connection with FIG. 2. As shown in line 16 of the pseudocode 300, k different actions can be sampled and collected as $A^{rs(k)}(I_i)$ according to the robust sampling strategy $\sigma_i^{rs(k)}$.

[0072] In some embodiments, a depth-dependent sampling scheme can be used to provide more balanced or well-around information of a game represented by a game tree. For example, the sampling strategy $\sigma_i^{rs}(a|h)$ can be a function of a depth of the state h in a game tree. For example, the sampling strategy $\sigma_i^{rs}(a|h)$ can be designed such that a state closer to the terminal state has a higher probability of being sampled than a state closer to the initial state (e.g., represented by the root of the game tree). As an example, the depth-dependent sampling scheme can be implemented by applying different weights to sampling probabilities of different states with different depths. Such a depth-dependent sampling can help provide more information on states closer to the terminal states, which can be advantageous because typically there are more states closer to the terminal states than those closer to the initial state (e.g., due to the branching structure of the game tree) and these nodes have a lower chance of being sampled than nodes closer to the root node in sampled trajectories under existing sampling schemes.

[0073] In some embodiments, the depth-dependent sampling scheme can be used in combination with the robust sampling, outcome sampling, external sampling, or any other suitable sampling algorithms. For example, the depth-dependent sampling scheme can further improve the variance and convergence speed of either of the robust sampling, outcome sampling, and external sampling as the latter three sampling schemes focus more on horizontal sampling among different actions of a state of a player (e.g., represented by the different branches of a node of a game tree).

[0074] FIG. 4 is a diagram illustrating an example 400 of a double neural CFR algorithm applied to a game tree 410 in accordance with embodiments of this specification. The double neural CFR algorithm 400 use two neural networks 420 and 430 to compute approximate Nash equilibrium of an IIG such as represented by the game tree 410. As illustrated in FIG. 4, one neural network 420 is used for obtaining a cumulative regret and is referred to as RegretSumNetwork (RSN). The other

neural networks 430 is used for obtaining an average strategy and is referred to as AveStrategyNetwork (ASN).

[0075] In some embodiments, the iterative updates of the CFR algorithm maintain two strategies: the current strategy $\sigma_i^t(a|I_i)$, and the average strategy $\bar{\sigma}_i^t(a|I_i)$ for $\forall i \in N, \forall I_i \in \mathrm{I}_i, \forall a \in A(I_i), \forall t \in \{1,...,T\}$. Accordingly, the two neural networks 420 and 430 can be designed to record these two strategies, respectively, in an iterative fashion. In some embodiments, the example 400 of the double neural CFR algorithm can be referred to a double incremental CFR algorithm as the neural networks are trained or optimized based on new, additional samples in each iteration.

[0076] According to Eq. (3), a current strategy $\sigma^{t+1}(a|I_i)$ can be computed by the cumulative regret $R^t(a|I_i)$. In some embodiments, only the numerator in Eq. (3) is tracked since the normalization in the denominator can easily be computed when the strategy is used. Given information set I_i and action a, the neural network RSN 420, denoted as $\mathcal{R}(a, I_i|\theta_R^t)$, can be used to learn $R^t(a|I_i)$, where θ_R^t is the parameter in the RSN 420 at t-th iteration.

[0077] As shown FIG. 4, a memory M_R^t 404 can be defined as $M_R^t = \{(I_i, \tilde{r}_i^{\sigma^t}((a|I_i)|Q_j))|\forall i \in N, \forall a \in A(I_i), h \in I_i, h \vee z, z \in Q_j\}$. Each member of M_R^t can include visited information set I_i and the corresponding regret $\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j)$, where Q_j is the sampled block in t-th iteration. According to Eq. (2), $\mathcal{R}(a, I_i|\theta_R^{t+1})$ can be estimated using the following optimization:

$$\theta_R^{t+1} \leftarrow \underset{\theta_R^{t+1}}{\operatorname{argmin}} \sum_{(I_i, \tilde{r}_i^{\sigma^t}((a|I_i)|Q_j)) \in M_R^t} (\mathcal{R}(a, I_i|\theta_R^t) + \tilde{r}_i^{\sigma^t}\left((a|I_i)\middle|Q_j\right) - \mathcal{R}(a, I_i|\theta_R^{t+1}))^2, \quad (7)$$

According to Eq. (4), the approximate Nash equilibrium is the weighted average of all previous strategies over T iterations. Similar to the cumulative regret, another neural network ASN 430, denoted as $S(a, I_i|\theta_S^t)$, can be used to learn the numerator of the average strategy. Define another memory M_S^t 406 as

$$M_S^t = \left\{\left(I_i, s^t(a|I_i)\right)\middle|\forall_i \in N, \forall_a \in A(I_i), h \in I_i, h \sqsubseteq z, z \in Q_j\right\} =$$
$$\left\{\left(I_i, \pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i)\right)\middle|\forall_i \in N, \forall_a \in A(I_i), h \in I_i, h \sqsubseteq z, z \in Q_j\right\}.$$

Each member of M_S^t can include the visited information set I_i and the value of $\pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i)$, where Q_j is the sampled block in t-th iteration. Then the parameter θ_S^{t+1} of the ASN 430 can be estimated by the following formula:

$$\theta_S^{t+1} \leftarrow \underset{\theta_S^{t+1}}{\operatorname{argmin}} \sum_{(I_i, s_i^t(a|I_i)) \in M_S^t} (S(a, I_i|\theta_S^t) + s_i^t(a|I_i) - S(a, I_i|\theta_S^{t+1}))^2, \quad (8)$$

[0078] In some embodiments, in each iteration, both $\mathcal{R}(a, I_i|\theta_R^t)$ and $S(a, I_i|\theta_S^t)$ can be optimized by optimizing the objectives in Eq. (7) and Eq. (8) according to a gradient descent method, such as a mini-batch stochastic gradient descent method described with respect to FIG. 7 below.

[0079] In some embodiments, the average strategy does not need to update in each iteration if a large memory is available to aggregate and save $s_i^t(a|I_i)$ within multiple iterations. If the memory M_S^t is full, the incremental value can be learned by optimizing Eq. (8).

[0080] In some embodiments, in each iteration, only a small subset of information sets are sampled, which may lead to that the neural network RSN 420 and ASN 430 forget values of those unobserved or un-sampled information sets. To address this issue, the neural network parameters from a previous iteration can be used as the initialization of a current iteration, which gives an online learning/adaptation flavor to the updates. Furthermore, due to the generalization ability of the neural networks, even samples from a small number of information sets can be used to update the new neural networks, and the newly updated neural networks can produce good values for the cumulative regret and the average strategy.

[0081] In some embodiments, as the number of iterations t increases, the value of $R_i^t(a|I_i)$ may become increasingly large, potentially making it difficult for the neural network to learn. To address this issue, the cumulative regret can be normalized by a factor of $\sqrt{t}$ to make its range more stable. This can be understood from the regret bound of online learning. More specifically, let $\Delta = \max_{Ii,a,t} |R^t(a|I_i) - R^{t-1}(a|I_i)|, \forall I_i \in \mathrm{I}_i, a \in A(I_i), t \in \{1,...,T\}$. $R_i^t(a|I_i) \leq \Delta\sqrt{|A|t}$, where $|A| = \max_{Ii\in\mathrm{I}} |A(I_i)|$. In some embodiments, the neural network RSN 420 can be used to track $\hat{R}_i^t(a|I_i) = R_i^t(a|I_i)/\sqrt{t}$, and update it by

$$\hat{R}_i^t(a|I_i) = \frac{\sqrt{t-1}\hat{R}_i^{t-1}(a|I_i)}{\sqrt{t}} + \frac{r_i^{\sigma^t}(a|I_i)}{\sqrt{t}}, \quad (9)$$

where $\hat{R}_i^0(a|I_i) = 0$.

[0082] In some embodiments, in the double incremental CFR algorithm, the memories M_R^t and M_S^t may be cleared after each iteration, for example, due to the limited size of the memories M_R^t and M_S^t. In some embodiments, for example in a large game, even with the double incremental CFR algorithm that uses the neural networks to learn the cumulative regrets and the average strategy, the size of the memories M_R^t and M_S^t may still need to be very large to record the cumulative regret and average strategies for each iteration.

[0083] In some embodiments, to continually improve the average strategy with limited memory but infinity iterations and/or to further relieve the requirement on the memory size, a double reservoir CFR algorithm can be used that use two reservoirs M_R and M_S to save the sampled cumulative regrets and the average strategies across different iterations and to dynamically learn the cumulative regret and average strategy.

[0084] In some embodiments, an average cumulative regret after T iterations can be obtained according to Eq. (10) by rewriting Eq. (2):

$$\bar{R}_i^T(a|I_i) = R_i^T(a|I_i)/T \quad (10)$$

[0085] Similarly, the average strategy can be the normalization of cumulative strategy as shown in Eq. (4), which is weighted average strategy $\sigma_i^t(a|I_i)$ by its reach probability $\pi_i^{\sigma^t}(I_i)$.

[0086] In some embodiments, two uniform reservoirs M_R and M_S can be used to save the sampled $\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j)$ and $s^t(a|I_i)$, respectively. Specifically, M_R can be the reservoir to save samples in M_R^t and M_S can be the reservoir to save samples in M_S^t. New samples can be inserted into the reservoir by a reservoir sampling algorithm. Reservoir sampling includes a family of randomized algorithms for randomly choosing k items from a list containing n items. For example, if the reservoir is not full, new samples can be added to the reservoir directly. If a reservoir is full, new samples can replace old samples according to, for example, a first in first out (FIFO) principle or according to a uniform random distribution or another distribution.

[0087] Note that both the double incremental CFR algorithm and the double reservoir CFR algorithm employ the ideas in online learning and use two neural networks to learn the updating regret and average strategy, respectively. In some embodiments, the ASN does not need to be updated in each iteration while the RSN may need to be optimized after Monte Carlo sampling so as to produce a new behavior strategy. For example, when a new behavior strategy is used to traverse the game tree, the RSN may need to be updated each iteration. On the other hand, the ASN can be used as the final approximated Nash equilibrium which is the weighted average of behavior strategy. The ASN can serves as the output of the double neural CFR algorithm. If there is a large enough data store to save all the samples, it is only needed to optimize the average strategy at the last step. In practice, for a large game, the large data store can be very expensive. As such, the average strategy can be incrementally optimized if the data store (e.g., the reservoir M_S^t) is full. As such, the double neural CFR algorithm can include two variations. In the double incremental algorithm, the neural network (e.g., one or both RSN and ASN) is optimized by the incremental samples only, while the double reservoir algorithms, the neural network (e.g., one or both RSN and ASN) can be optimized by all the samples in the reservoirs.

[0088] The double neural CFR algorithm and the double reservoir CFR algorithm have different sample collections. For the double incremental CFR, the neural network is optimized based on the newly added samples. For double reservoir CFR, the neural network is optimized based on the samples in the fixed-size reservoirs. In addition, in the double reservoir method, the average strategy can be optimized by maximum log-likelihood rather than minimum squared error.

[0104] The function Mini-Batch-MCCFR-NN shown in the pseudocode 700 presents a mini-batch sampling method, where b blocks will be sampled in parallel. This mini-batch method can help the MCCFR to achieve a more accurate estimation of CFV. The parallel sampling makes this method efficient in practice.

[0105] As shown in lines 1-6 of the pseudocode 700, the Mini-Batch-MCCFR-NN is an iterative algorithm with an input of the total number of iterations, t. Within each iteration, an MCCFR-NN function is called for player 0 and player 1 (as shown in lines 4 and 5) and the counterfactual regret and numerator of average strategy for sampled information sets in this iteration are returned and save in memories M_R^t and M_S^t, respectively.

[0106] The function MCCFR-NN can be defined as shown in lines 8-33 of the pseudocode 700. The function MCCFR-NN traverses the game tree like tabular MCCFR, which starts from the root history $h = \emptyset$. Define I_i as the information set of h. Suppose that player i will sample k actions according to the robust sampling. Then the function can be defined as follows. (1) If the history is terminal (e.g., $h \in Z$), the function returns the weighted utility. (2) If the history is the chance player (e.g., $P(I_i) = -1$,), one action $a \in A(I_i)$ can be sampled according to the strategy $\sigma_{-i}(I_i)$. Then this action will be added to the history, *i.e.*, $h \leftarrow ha$. (3) If $P(I_i) = i$, the current strategy can be updated by the cumulative regret predicted by RSN. Then sample k actions according the specified sampling strategy profile $\sigma_i^{rs(k)}$ (e.g., robust sampling with or without depth-dependent sampling). After a recursive updating, the counterfactual value and regret of each action at I_i can be obtained. For the visited node, their counterfactual regrets and numerators of the corresponding average strategy can be stored in M_R^t and M_S^t respectively. (4) If $P(I_i)$ is the opponent, only one action will be sampled according the strategy $\sigma_{-i}(I_i)$.

[0107] FIG. 8 is a flowchart of an example of a sampling process 800 for performing MCCFR in accordance with embodiments of this specification. The sampling process 800 can be an example of the depth-dependent sampling scheme described above for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more parties. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. The IIG can represent one or more real-world scenarios such as resource allocation, product/service recommendation, cyber-attack prediction and/or prevention, traffic routing, fraud management, etc. that involves two or more parties, where each party may have incomplete or imperfect information about the other party's decisions. As an example, the IIG can represent a collaborative product-service recommendation service that involves at least a first player and a second player. The first player may be, for example,

an online retailer that has customer (or user) information, product and service information, purchase history of the customers, etc. The second player can be, for example, a social network platform that has social networking data of the customers, a bank or another finical institution that has financial information of the customers, a car dealership, or any other parties that may have information of the customers on the customers' preferences, needs, financial situations, locations, etc. in predicting and recommendations of products and services to the customers. The first player and the second player may each have proprietary data that does not want to share with others. The second player may only provide partial information to the first player at different times. As such, the first player may only have limit access to information of the second player. For convenience, the process 800 will be described as being performed by a data processing apparatus such as a system of one or more computers, located in one or more locations, and programmed appropriately in accordance with this specification. For example, a computer system 1000 of FIG. 10, appropriately programmed, can perform the process 800.

[0108] At 810, a data processing apparatus identifies N1 possible actions of a first player in a first state of the first player. In some embodiments, the IIG can be represented by a game tree (e.g., the game tree 102, 104, 202, 204 or 206). The first state of the first player can be represented by a first node of the game tree (e.g., node h1 of player 0 in the game tree 102) and the N1 possible action can be the edges or branches of the first node of the game tree (e.g., the P and B edges of the node h1 of player 0 in the game tree 102). In the example of the collaborative product-service recommendation service, the first state of the first player includes a history of information provided by the second player, and the N1 possible actions of the first player includes N1 possible actions in response to the history of information provided by the second player for providing product-service recommendations to customers. The first state of the first player and the possible actions can include other features in other real-world scenarios that are modeled by the IIG.

[0109] At 820, the data processing apparatus samples a possible action out of the N1 possible actions in the first state of the first player with a first sampling probability. In some embodiments, the data processing apparatus may sample k1 possible actions out of the N1 possible actions in the first state of the first player, wherein each of the k1 possible actions is sampled with the same first sampling probability.

[0110] At 830, the data processing apparatus identifies N2 possible actions of the first player in a second state of the first player, wherein the first state of the first player is closer to a beginning state of the IIG than the second state of the first player. In the example of the game tree 102, the second state of the first player can be, for example, the h7 node, which is further from the beginning state

(e.g., the h0 node) of the game tree 102 than the first state of the first player (e.g., node h1 of player 0 in the game tree 102).

[0111] At 840, the data processing apparatus samples a possible action out of the N2 possible actions in the second state of the first player with a second sampling probability, wherein the first sampling probability is less than the second sampling probability. In some embodiments, the data processing apparatus samples k2 possible actions out of the N2 possible actions in the second state of the first player, wherein each of the k2 possible actions is sampled with the same second sampling probability.

[0112] At 850, the data processing apparatus perform CFR based on the possible actions out of the N1 possible actions in the first state of the first player and the possible action out of the N2 possible actions in the second state of the first player. In some embodiments, the CFR can be performed according to the example techniques described with respect to FIG. 3 and/or FIG. 7.

[0113] In some embodiments, a strategy of the first player resulting from solving the IIG is output. The strategy can include a series of actions of the first player in the real-world scenario modeled by the IIG. For example, in the collaborative product-service recommendation scenario, the strategy of the first player resulting from solving the IIG can include, for example, a series of actions in response to the information provided by the second player, corresponding product-service recommendations to customers based on the information of the first player and the information provided by the second player. The output strategy of the first player resulting from solving the IIG can include other information in other real-world scenarios that are modeled by the IIG.

[0114] In some embodiments, performing CFR based on the possible action out of the N1 possible actions in the first state of the first player and the possible action out of the N2 possible actions in the second state of the first player includes calculating a regret value of the possible action out of the N1 possible actions in the first state of the first player (e.g., according to Eq. (1a) and/or Eq. (2)); calculating a regret value of the possible action out of the N2 possible actions in the second state of the first player (e.g., according to Eq. (1a) and/or Eq. (2)); updating a first strategy of the first player in the first state based on the regret value of the possible action out of the N1 possible actions (e.g., according to Eq. (3)); and updating a second strategy of the first player in the second state based on the regret value of the possible action out of the N2 possible actions (e.g., according to Eq. (3)).

[0115] In some embodiments, the data processing apparatus performs CFR based on the k1 possible actions out of the N1 possible actions in the first state of the first player and the k2 possible actions out of the N2 possible actions in the second state of the first player.

[0116] In some embodiments, robust sampling can be performed in connection with the depth-dependent sampling. For example, the first sampling probability is k1/N1, and the second sampling probability is k2/N2. As such, the possible actions are sampled according to a uniform distribution.

[0117] In some embodiments, 2<= k1<= N1 and 2<= k2 <= N2 so that more than one possible action is visited for each state of the player.

[0118] In some embodiments, k1 = k2 so an equal number of samples are selected or visited in the first state and second state of the first player.

[0119] Similarly, the depth-dependent sampling can be performed in connection with a second player. For example, the data processing apparatus identifies M1 possible actions of a second player in a first state of the second player. The data processing apparatus samples a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability. The data processing apparatus identifies M2 possible actions of the second player in a second state of the second player, wherein the first state of the second player is closer to a beginning state of the IIG than the second state of the second player. The data processing apparatus samples a possible action out of the M2 possible actions in the second state of the second player with a fourth sampling probability, wherein the third sampling probability is less than the fourth sampling probability.

[0120] In some embodiments, the depth-dependent sampling can be performed in connection with both the first player and the second player. In some embodiments, the data processing apparatus identifies M1 possible actions of a second player in a first state of the second player, wherein the first state of the first player (e.g., state h1 of player 0 in the game tree 102) is closer to a beginning state (e.g., h0 state) of the IIG than the first state of the second player (e.g., state h4 of player 1 in the game tree 102). The data processing apparatus samples a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability, wherein the third sampling probability is larger than the first sampling probability.

[0121] FIG. 9 is a flowchart of an example of double neural CFR algorithm 900 for performing MCCFR in accordance with embodiments of this specification. The sampling process 900 can be an example of the double reservoir CFR algorithm described above with respect to FIGS. 4-7 for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. The IIG can represent one or more real-world scenarios such as resource allocation, product/service recommendation, cyber-attack prediction and/or prevention, traffic routing, fraud management, etc. that involves two or more parties, where each party may have incomplete or imperfect information about the other party's decisions. As an example, the IIG can represent a collaborative product-

service recommendation service that involves at least a first player and a second player. The first player may be, for example, an online retailer that has customer (or user) information, product and service information, purchase history of the customers, etc. The second player can be, for example, a social network platform that has social networking data of the customers, a bank or another finical institution that has financial information of the customers, a car dealership, or any other parties that may have information of the customers on the customers' preferences, needs, financial situations, locations, etc. in predicting and recommendations of products and services to the customers. The first player and the second player may each have proprietary data that does not want to share with others. The second player may only provide partial information to the first player at different times. As such, the first player may only have limit access to information of the second player. For convenience, the process 900 will be described as being performed by a data processing apparatus such as a system of one or more computers, located in one or more locations, and programmed appropriately in accordance with this specification. For example, a computer system 1000 of FIG. 10, appropriately programmed, can perform the process 900.

[0122] At 910, a data processing apparatus initializes parameters of a first neural network and parameters of a second neural network. The first neural network (e.g., RegretSumNetwork (RSN) 420) can be used for predicting a regret value of a possible action in a state of the player. In the example of the collaborative product-service recommendation service, the state of the player includes a history of information provided by the second player, and the possible action of the player includes a possible action in response to the history of information provided by the second player for providing product-service recommendations to customers. The second neural network (e.g., AveStrategyNetwork (ASN) 430) can be used for predicting a strategy value of a possible action in a state of the player. In some embodiments, the data processing apparatus initializes parameters according to a warm start, for example, based on parameters of the first neural network and parameters of the second neural network in a previous iteration or obtained based on an existing CFR algorithm, respectively. In some embodiments, the data processing apparatus initializes parameters of the first neural network and parameters of the second neural network randomly.

[0123] At 920, the data processing apparatus stores a number of regret samples in a first data store (e.g., the reservoir M_R), wherein each of the number of regret samples includes a state of the player and a regret value of a possible action in the state of the player. In some embodiments, the regret value of a possible action in the state of the player includes a counterfactual regret value of the possible action in the state of the player calculated based on a counterfactual value of the possible action in the state of the player. For example, each regret sample can include a regret tuple (I_i, $\tilde{r}_i^{\sigma^t}(a|I_i)$). In some embodiments, the number of regret samples are obtained in two or more

iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and at least another player. In some embodiments, the CFR algorithm includes a robust sampling CFR algorithm.

[0124] In some embodiments, in each of the two or more iterations of the CFR algorithm in solving the IIG, the data processing apparatus samples a possible action out of a number of possible actions in a second state of a player according to a sampling scheme; calculates a counterfactual value of the possible action in the second state of the player (e.g., according to Eq. (1)); calculates a regret value of the possible action in the second state of the player based on the counterfactual value of the possible action in the second state of the player (e.g., according to Eq. (1a) and/or Eq. (2)); calculate an updated strategy of the possible action in the second state of the player based the regret value of the possible action in the second state of the player according to a regret matching algorithm (e.g., according to Eq. (3)); and calculates a strategy value of the possible action in the second state of the player based on the updated strategy of the possible action in the second state of the player (e.g., according to Eq. (4) and/or Eq. (5)).

[0125] In some embodiments, the data processing apparatus can obtain a new regret sample (e.g., by performing another iteration of the MCCFR). The data processing apparatus can store the new regret sample into the first data store according to a reservoir sampling algorithm. For example, storing the new regret sample into the first data store according to a reservoir sampling algorithm includes: determining whether the first data store is full; and in response to determining that the first data store is full, replacing one of the number of regret samples in the first data store with the new regret sample.

[0126] At 930, the data processing apparatus stores a number of strategy samples in a second data store (e.g., the reservoir M_S), wherein each of the number of strategy samples includes a state of the player and a strategy value of a possible action in the state of the player. In some embodiments, the strategy value of a possible action in the state of the player includes a numerator of an average strategy. For example, each of the number of strategy samples can include a strategy tuple $\left(I_i, s^t(a|I_i)\right)$.

[0127] At 940, the data processing apparatus updates parameters of a first neural network for predicting a regret value of a possible action in a state of the player based on the number of regret samples in the first data store, for example, according to Eq. (7). In some embodiments, the parameters of the first neural network can be updated according to the NeuralAgent algorithm shown in FIG. 6, or any other algorithms for optimizing a neural network.

Further, the term “client,” “user,” or other appropriate terminology can be used interchangeably, as appropriate, without departing from the scope of the present disclosure. Moreover, the present disclosure contemplates that many users can use one Computer 1002, or that one user can use multiple computers 1002.

[0145] FIG. 11 is a diagram of on example of modules of an apparatus 1100 in accordance with embodiments of this specification. The apparatus 1100 can be an example embodiment of a data processing apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. As an example, the IIG represents a collaborative product-service recommendation service that involves at least a first player and a second player, the first player having limited access to information of the second player. The apparatus 1100 can correspond to the embodiments described above, and the apparatus 1100 includes the following: a first identifying module 1101 for identifying N1 possible actions of a first player in a first state of the first player; a first sampling module 1102 for sampling a possible action out of the N1 possible actions in the first state of the first player with a first sampling probability; a second identifying module 1103 for identifying N2 possible actions of the first player in a second state of the first player, wherein the first state of the first player is closer to a beginning state of the IIG than the second state of the first player; a second sampling module 1104 for sampling a possible action out of the N2 possible actions in the second state of the first player with a second sampling probability, wherein the first sampling probability is less than the second sampling probability; and a processing module 1105 for performing the CFR based on the possible actions out of the N1 possible actions in the first state of the first player and the possible action out of the N2 possible actions in the second state of the first player. In some embodiments, the first state of the first player includes a history of information provided by the second player, and the N1 possible actions of the first player includes N1 possible actions in response to the history of information provided by the second player for providing product-service recommendations to customers.

[0146] In an optional embodiment, the processing module includes: a first calculating module for calculating a regret value of the possible action out of the N1 possible actions in the first state of the first player; a second calculating module for calculating a regret value of the possible action out of the N2 possible actions in the second state of the first player; a first updating module for updating a first strategy of the first player in the first state based on the regret value of the possible action out of the N1 possible actions; and a second updating module for updating a second strategy of the first

player in the second state based on the regret value of the possible action out of the N2 possible actions.

[0147] In an optional embodiment, the first sampling module samples k1 possible actions out of the N1 possible actions in the first state of the first player, wherein each of the k1 possible actions is sampled with the same first sampling probability; and the second sampling module samples k2 possible actions out of the N2 possible actions in the second state of the first player, wherein each of the k2 possible actions is sampled with the same second sampling probability.

[0148] In an optional embodiment, the processing module performs the CFR based on the k1 possible actions out of the N1 possible actions in the first state of the first player and the k2 possible actions out of the N2 possible actions in the second state of the first player.

[0149] In an optional embodiment, the first sampling probability is k1/N1, and the second sampling probability is k2/N2.

[0150] In an optional embodiment, 2<= k1<= N1 and 2<= k2 <= N2.

[0151] In an optional embodiment, k1 = k2.

[0152] In an optional embodiment, the apparatus 1100 further includes the following: a third identifying module for identifying M1 possible actions of a second player in a first state of the second player; a third sampling module for sampling a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability; a fourth identifying module for identifying M2 possible actions of the second player in a second state of the second player, wherein the first state of the second player is closer to a beginning state of the IIG than the second state of the second player; and a fourth sampling module for sampling a possible action out of the M2 possible actions in the second state of the first player with a fourth sampling probability, wherein the third sampling probability is less than the fourth sampling probability.

[0153] In an optional embodiment, the apparatus 1100 further includes the following: a fifth identifying module for identifying M1 possible actions of a second player in a first state of the second player, wherein the first state of the first player is closer to a beginning state of the IIG than the first state of the second player; and a fifth sampling module for sampling a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability, wherein the third sampling probability is larger than the first sampling probability.

[0154] The system, apparatus, module, or unit illustrated in the previous embodiments can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical embodiment device is a computer, and the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal

digital assistant, a media player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0155] For an embodiment process of functions and roles of each module in the apparatus, references can be made to an embodiment process of corresponding steps in the previous method. Details are omitted here for simplicity.

[0156] Because an apparatus embodiment basically corresponds to a method embodiment, for related parts, references can be made to related descriptions in the method embodiment. The previously described apparatus embodiment is merely an example. The modules described as separate parts may or may not be physically separate, and parts displayed as modules may or may not be physical modules, may be located in one position, or may be distributed on a number of network modules. Some or all of the modules can be selected based on actual demands to achieve the objectives of the solutions of the specification. A person of ordinary skill in the art can understand and implement the embodiments of the present application without creative efforts.

[0157] Referring again to FIG. 11, it can be interpreted as illustrating an internal functional module and a structure of a data processing apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. An execution body in essence can be an electronic device, and the electronic device includes the following: one or more processors; and a memory configured to store an executable instruction of the one or more processors.

[0158] The one or more processors are configured to identify N1 possible actions of a first player in a first state of the first player; sample a possible action out of the N1 possible actions in the first state of the first player with a first sampling probability; identify N2 possible actions of the first player in a second state of the first player, wherein the first state of the first player is closer to a beginning state of the IIG than the second state of the first player; sample a possible action out of the N2 possible actions in the second state of the first player with a second sampling probability, wherein the first sampling probability is less than the second sampling probability; and perform the CFR based on the possible actions out of the N1 possible actions in the first state of the first player and the possible action out of the N2 possible actions in the second state of the first player.

[0159] Optionally, the one or more processors are configured to calculate a regret value of the possible action out of the N1 possible actions in the first state of the first player; calculate a regret value of the possible action out of the N2 possible actions in the second state of the first player; update a first strategy of the first player in the first state based on the regret value of the possible

action out of the N1 possible actions; and update a second strategy of the first player in the second state based on the regret value of the possible action out of the N2 possible actions.

[0160] Optionally, the one or more processors are configured to sample k1 possible actions out of the N1 possible actions in the first state of the first player, wherein each of the k1 possible actions is sampled with the same first sampling probability; and sample k2 possible actions out of the N2 possible actions in the second state of the first player, wherein each of the k2 possible actions is sampled with the same second sampling probability.

[0161] Optionally, the one or more processors are configured to perform CFR based on the k1 possible actions out of the N1 possible actions in the first state of the first player and the k2 possible actions out of the N2 possible actions in the second state of the first player.

[0162] Optionally, the first sampling probability is k1/N1, and the second sampling probability is k2/N2.

[0163] Optionally, 2<= k1<= N1 and 2<= k2 <= N2.

[0164] Optionally, k1 = k2.

[0165] Optionally, the one or more processors are configured to identify M1 possible actions of a second player in a first state of the second player; sample a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability; identify M2 possible actions of the second player in a second state of the second player, wherein the first state of the second player is closer to a beginning state of the IIG than the second state of the second player; and sample a possible action out of the M2 possible actions in the second state of the first player with a fourth sampling probability, wherein the third sampling probability is less than the fourth sampling probability.

[0166] Optionally, the one or more processors are configured to: identify M1 possible actions of a second player in a first state of the second player, wherein the first state of the first player is closer to a beginning state of the IIG than the first state of the second player; and sample a possible action out of the M1 possible actions in the first state of the second player with a third sampling probability, wherein the third sampling probability is larger than the first sampling probability.

[0167] FIG. 12 is a diagram of on example of modules of another apparatus 1200 in accordance with embodiments of this specification. The apparatus 1200 can be an example embodiment of a data processing apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. The apparatus 1200 can correspond to the embodiments described above, and the apparatus 1200 includes the following: a first storing module 1201 for storing a number of regret samples in a first data store, wherein each of the number of regret samples includes a state of a player and a regret value of a possible action in the state of the player,

wherein the number of regret samples are obtained in two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and at least another player; a second storing module 1202 for storing a number of strategy samples in a second data store, wherein each of the number of strategy samples includes a state of the player and a strategy value of a possible action in the state of the player; a first updating module 1203 for updating parameters of a first neural network for predicting a regret value of a possible action in a state of the player based on the number of regret samples in the first data store; and a second updating module 1204 for updating parameters of a second neural network for predicting a strategy value of a possible action in a state of the player based on the number of strategy samples in the second data store. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. As an example, the IIG represents a collaborative product-service recommendation service that involves at least the player and a second player, the player having limited access to information of the second player, wherein the state of the player includes a history of information provided by the second player, and wherein the possible action of the player includes a possible action in response to the history of information provided by the second player for providing product-service recommendations to customers.

[0168] The foregoing and other described embodiments can each, optionally, include one or more of the following features:

[0169] In an optional embodiment, the apparatus 1200 further includes the following: an identifying module for identifying a first state of the player and a first possible action in the first state of the player; a first predicting module for predicting a first regret value of the first possible action in the first state of the player using the parameters of the first neural network; and a second predicting module for predicting a first strategy value of the first possible action in the first state of the player using the parameters of the second neural network.

[0170] In an optional embodiment, wherein the first storing module is capable of obtaining a new regret sample; and storing the new regret sample into the first data store according to a reservoir sampling algorithm.

[0171] In an optional embodiment, wherein storing the new regret sample into the first data store according to a reservoir sampling algorithm includes: determining whether the first data store is full; and in response to determining that the first data store is full, replacing one of the number of regret samples in the first data store with the new regret sample.

[0172] In an optional embodiment, wherein the CFR algorithm includes a robust sampling CFR algorithm.

[0173] In an optional embodiment, wherein the strategy value of a possible action in the state of the player includes a numerator of an average strategy.

[0174] In an optional embodiment, wherein the regret value of a possible action in the state of the player includes a counterfactual regret value of the possible action in the state of the player calculated based on a counterfactual value of the possible action in the state of the player.

[0175] In an optional embodiment, the apparatus 1200 further includes the following: further including: in each of the two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and the at least another player, a sampling module for sampling a possible action out of a number of possible actions in a second state of a player according to a sampling scheme; a first calculating module for calculating a counterfactual value of the possible action in the second state of the player; a second calculating module for calculating a regret value of the possible action in the second state of the player based on the counterfactual value of the possible action in the second state of the player; a third calculating module for calculating an updated strategy of the possible action in the second state of the player based the regret value of the possible action in the second state of the player according to a regret matching algorithm; and a fourth calculating module for calculating a strategy value of the possible action in the second state of the player based on the updated strategy of the possible action in the second state of the player.

[0176] In an optional embodiment, the apparatus 1200 further includes the following: further including: a first initializing module for initializing the parameters of the first neural network based on parameters of the first neural network in a previous iteration; and a second initializing module for initializing the parameters of the second neural network based on parameters of the second neural network in a previous iteration.

[0177] The system, apparatus, module, or module illustrated in the previous embodiments can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical embodiment device is a computer, and the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0178] For an embodiment process of functions and roles of each module in the apparatus, references can be made to an embodiment process of corresponding steps in the previous method. Details are omitted here for simplicity.

[0179] Because an apparatus embodiment basically corresponds to a method embodiment, for related parts, references can be made to related descriptions in the method embodiment. The

previously described apparatus embodiment is merely an example. The modules described as separate parts may or may not be physically separate, and parts displayed as modules may or may not be physical modules, may be located in one position, or may be distributed on a number of network modules. Some or all of the modules can be selected based on actual demands to achieve the objectives of the solutions of the specification. A person of ordinary skill in the art can understand and implement the embodiments of the present application without creative efforts.

[0180] Referring again to FIG. 12, it can be interpreted as illustrating an internal functional module and a structure of a data processing apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. An execution body in essence can be an electronic device, and the electronic device includes the following: one or more processors; and a memory configured to store an executable instruction of the one or more processors.

[0181] The one or more processors are configured to store a number of regret samples in a first data store, wherein each of the number of regret samples includes a state of a player and a regret value of a possible action in the state of the player, wherein the number of regret samples are obtained in two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and at least another player; store a number of strategy samples in a second data store, wherein each of the number of strategy samples includes a state of the player and a strategy value of a possible action in the state of the player; update parameters of a first neural network for predicting a regret value of a possible action in a state of the player based on the number of regret samples in the first data store; and update parameters of a second neural network for predicting a strategy value of a possible action in a state of the player based on the number of strategy samples in the second data store. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. As an example, the IIG represents a collaborative product-service recommendation service that involves at least the player and a second player, the player having limited access to information of the second player, wherein the state of the player includes a history of information provided by the second player, and wherein the possible action of the player includes a possible action in response to the history of information provided by the second player for providing product-service recommendations to customers.

[0182] Optionally, the one or more processors are configured to: identify a first state of the player and a first possible action in the first state of the player; predict a first regret value of the first possible action in the first state of the player using the parameters of the first neural network; and

predict a first strategy value of the first possible action in the first state of the player using the parameters of the second neural network.

[0183] Optionally, the one or more processors are configured to: obtain a new regret sample; and store the new regret sample into the first data store according to a reservoir sampling algorithm.

[0184] Optionally, the one or more processors are configured to: determine whether the first data store is full; and in response to determining that the first data store is full, replace one of the number of regret samples in the first data store with the new regret sample.

[0185] Optionally, the CFR algorithm includes a robust sampling CFR algorithm.

[0186] Optionally, the strategy value of a possible action in the state of the player includes a numerator of an average strategy.

[0187] Optionally, the regret value of a possible action in the state of the player includes a counterfactual regret value of the possible action in the state of the player calculated based on a counterfactual value of the possible action in the state of the player.

[0188] Optionally, the one or more processors are configured to: in each of the two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and the at least another player, sample a possible action out of a number of possible actions in a second state of a player according to a sampling scheme; calculate a counterfactual value of the possible action in the second state of the player; calculate a regret value of the possible action in the second state of the player based on the counterfactual value of the possible action in the second state of the player; calculate an updated strategy of the possible action in the second state of the player based the regret value of the possible action in the second state of the player according to a regret matching algorithm; and calculate a strategy value of the possible action in the second state of the player based on the updated strategy of the possible action in the second state of the player.

[0189] Optionally, the one or more processors are configured to initialize the parameters of the first neural network based on parameters of the first neural network in a previous iteration; and initialize the parameters of the second neural network based on parameters of the second neural network in a previous iteration.

[0190] Described embodiments of the subject matter can include one or more features, alone or in combination. For example, in a first embodiment, a computer-implemented method for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more players. The method includes: storing a number of regret samples in a first data store, wherein each of the number of regret samples includes a state of a player and a regret value of a possible action in the state of the player, wherein the number of regret samples are obtained in two

or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and at least another player; storing a number of strategy samples in a second data store, wherein each of the number of strategy samples includes a state of the player and a strategy value of a possible action in the state of the player; updating parameters of a first neural network for predicting a regret value of a possible action in a state of the player based on the number of regret samples in the first data store; and updating parameters of a second neural network for predicting a strategy value of a possible action in a state of the player based on the number of strategy samples in the second data store.

[0191] The foregoing and other described embodiments can each, optionally, include one or more of the following features:

[0192] A first feature, combinable with any of the following features, further including: identifying a first state of the player and a first possible action in the first state of the player; predicting a first regret value of the first possible action in the first state of the player using the parameters of the first neural network; and predicting a first strategy value of the first possible action in the first state of the player using the parameters of the second neural network.

[0193] A second feature, combinable with any of the following features, further including: obtaining a new regret sample; and storing the new regret sample into the first data store according to a reservoir sampling algorithm.

[0194] A third feature, combinable with any of the following features, wherein storing the new regret sample into the first data store according to a reservoir sampling algorithm includes: determining whether the first data store is full; and in response to determining that the first data store is full, replacing one of the number of regret samples in the first data store with the new regret sample.

[0195] A fourth feature, combinable with any of the following features, wherein the CFR algorithm includes a robust sampling CFR algorithm.

[0196] A fifth feature, combinable with any of the following features, wherein the strategy value of a possible action in the state of the player includes a numerator of an average strategy.

[0197] A sixth feature, combinable with any of the following features, wherein the regret value of a possible action in the state of the player includes a counterfactual regret value of the possible action in the state of the player calculated based on a counterfactual value of the possible action in the state of the player.

[0198] A seventh feature, combinable with any of the following features, further including: in each of the two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and the at least another player, sampling

a possible action out of a number of possible actions in a second state of a player according to a sampling scheme; calculating a counterfactual value of the possible action in the second state of the player; calculating a regret value of the possible action in the second state of the player based on the counterfactual value of the possible action in the second state of the player; calculating an updated strategy of the possible action in the second state of the player based the regret value of the possible action in the second state of the player according to a regret matching algorithm; and calculating a strategy value of the possible action in the second state of the player based on the updated strategy of the possible action in the second state of the player.

[0199] An eighth feature, combinable with any of the following features, further including: initializing the parameters of the first neural network based on parameters of the first neural network in a previous iteration; and initializing the parameters of the second neural network based on parameters of the second neural network in a previous iteration.

[0200] In a second embodiment, a system, including: one or more processors; and one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon which are executable by the one or more processors to perform the method of any of the first embodiment and its optional combination of the one or more of features described above.

[0201] In a third embodiment, an apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction involves two or more players, including: a first storing module for storing a number of regret samples in a first data store, wherein each of the number of regret samples includes a state of a player and a regret value of a possible action in the state of the player, wherein the number of regret samples are obtained in two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and at least another player; a second storing module for storing a number of strategy samples in a second data store, wherein each of the number of strategy samples includes a state of the player and a strategy value of a possible action in the state of the player; a first updating module for updating parameters of a first neural network for predicting a regret value of a possible action in a state of the player based on the number of regret samples in the first data store; and a second updating module for updating parameters of a second neural network for predicting a strategy value of a possible action in a state of the player based on the number of strategy samples in the second data store. In some embodiments, strategic interaction between two or more players can be modeled by an imperfect information game (IIG) that involves two or more players. As an example, the IIG represents a collaborative product-service recommendation service that involves at least the player and a second player, the player having limited access to information of the second player, wherein the state of the player includes a history of information provided by the second player, and

wherein the possible action of the player includes a possible action in response to the history of information provided by the second player for providing product-service recommendations to customers.

[0202] The foregoing and other described embodiments can each, optionally, include one or more of the following features:

[0203] A first feature, combinable with any of the following features, further including: an identifying module for identifying a first state of the player and a first possible action in the first state of the player; a first predicting module for predicting a first regret value of the first possible action in the first state of the player using the parameters of the first neural network; and a second predicting module for predicting a first strategy value of the first possible action in the first state of the player using the parameters of the second neural network.

[0204] A second feature, combinable with any of the following features, wherein the first storing module is capable of obtaining a new regret sample; and storing the new regret sample into the first data store according to a reservoir sampling algorithm.

[0205] A third feature, combinable with any of the following features, wherein storing the new regret sample into the first data store according to a reservoir sampling algorithm includes: determining whether the first data store is full; and in response to determining that the first data store is full, replacing one of the number of regret samples in the first data store with the new regret sample.

[0206] A fourth feature, combinable with any of the following features, wherein the CFR algorithm includes a robust sampling CFR algorithm.

[0207] A fifth feature, combinable with any of the following features, wherein the strategy value of a possible action in the state of the player includes a numerator of an average strategy.

[0208] A sixth feature, combinable with any of the following features, wherein the regret value of a possible action in the state of the player includes a counterfactual regret value of the possible action in the state of the player calculated based on a counterfactual value of the possible action in the state of the player.

[0209] A seventh feature, combinable with any of the following features, further including: in each of the two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the player and the at least another player, a sampling module for sampling a possible action out of a number of possible actions in a second state of a player according to a sampling scheme; a first calculating module for calculating a counterfactual value of the possible action in the second state of the player; a second calculating module for calculating a regret value of the possible action in the second state of the player based on

CLAIMS

1. A computer-implemented method for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more parties, the method comprising:

storing a plurality of incremental regret samples in a first data store, wherein each of the plurality of incremental regret samples comprises a state of a party and a regret value of a possible action in the state of the party, wherein the plurality of incremental regret samples are new, additional regret samples obtained in a current iteration of two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the party and at least another party;

storing a plurality of strategy samples in a second data store, wherein each of the plurality of strategy samples comprises a state of the party and a strategy value of a possible action in the state of the party;

updating parameters of a first neural network for predicting a regret value of a possible action in a state of the party based on the plurality of incremental regret samples in the first data store only rather than based on regret samples obtained across different iterations; and

updating parameters of a second neural network for predicting a strategy value of a possible action in a state of the party based on the plurality of strategy samples in the second data store.

2. The method of claim 1, further comprising:

identifying a first state of the party and a first possible action in the first state of the party;

predicting a first regret value of the first possible action in the first state of the party using the parameters of the first neural network; and

predicting a first strategy value of the first possible action in the first state of the party using the parameters of the second neural network.

3. The method of claim 1, further comprising:

clearing the first data store after each iteration.

4. The method of claim 1, further comprising:

removing duplicate records from the second data store.

5. The method of claim 1, wherein the CFR algorithm comprises a robust sampling CFR algorithm.

6. The method of claim 1, wherein the strategy value of a possible action in the state of the party comprises a numerator of an average strategy.

7. The method of claim 1, wherein the regret value of a possible action in the state of the party comprises a counterfactual regret value of the possible action in the state of the party calculated based on a counterfactual value of the possible action in the state of the party.

8. The method of claim 1, further comprising:

in each of the two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the party and the at least another party,

sampling a possible action out of a plurality of possible actions in a second state of a party according to a sampling scheme;

calculating a counterfactual value of the possible action in the second state of the party;

calculating a regret value of the possible action in the second state of the party based on the counterfactual value of the possible action in the second state of the party;

calculating an updated strategy of the possible action in the second state of the party based on the regret value of the possible action in the second state of the party according to a regret matching algorithm; and

calculating a strategy value of the possible action in the second state of the party based on the updated strategy of the possible action in the second state of the party.

9. The method of claim 1, further comprising:

initializing the parameters of the first neural network based on parameters of the first neural network in a previous iteration; and

initializing the parameters of the second neural network based on parameters of the second neural network in a previous iteration.

10. The method of claim 1, wherein the party has limited access to information of the another party, wherein the state of the party comprises a history of information provided by the another party, and wherein the possible action of the party comprises a possible action in response to

the history of information provided by the another party for providing product-service recommendations to customers.

11. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform the method of any of claims 1-10.

12. A computer-implemented system, comprising:

one or more computers; and

one or more computer memory devices interoperably coupled with the one or more computers and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more computers, perform the method of any of claims 1-10.

13. An apparatus for performing counterfactual regret minimization (CFR) for strategy searching in strategic interaction between two or more parties, comprising:

a first data store for storing a plurality of incremental regret samples, wherein each of the plurality of incremental regret samples comprises a state of a party and a regret value of a possible action in the state of the party, wherein the plurality of incremental regret samples are new, additional regret samples obtained in a current iteration of two or more iterations of a counterfactual regret minimization (CFR) algorithm in strategy searching in strategic interaction between the party and at least another party;

a second data store for storing a plurality of strategy samples, wherein each of the plurality of strategy samples comprises a state of the party and a strategy value of a possible action in the state of the party;

one or more processors coupled with one or more computer memory devices and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more processors, perform one or more operations comprising:

updating parameters of a first neural network for predicting a regret value of a possible action in a state of the party based on the plurality of incremental regret samples in the first data store only rather than based on regret samples obtained across different iterations; and

updating parameters of a second neural network for predicting a strategy value of a possible action in a state of the party based on the plurality of strategy samples in the second data store.