(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7537517号
(P7537517)

(45)発行日　令和6年8月21日(2024.8.21)

(24)登録日　令和6年8月13日(2024.8.13)

(51)国際特許分類
Ｇ０６Ｎ　20/00　(2019.01)

ＦＩ
Ｇ０６Ｎ　20/00

請求項の数　9　（全16頁）

(21)出願番号　特願2022-570960(P2022-570960)
(86)(22)出願日　令和2年12月25日(2020.12.25)
(86)国際出願番号　PCT/JP2020/048791
(87)国際公開番号　WO2022/137520
(87)国際公開日　令和4年6月30日(2022.6.30)
審査請求日　令和5年6月7日(2023.6.7)

(73)特許権者　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100103090
弁理士　岩壁 冬樹
(74)代理人　100124501
弁理士　塩川 誠人
(72)発明者　江藤 力
東京都港区芝五丁目７番１号 日本電気株式会社内
審査官　渡辺 一帆

最終頁に続く

(54)【発明の名称】　学習装置、学習方法および学習プログラム

(57)【特許請求の範囲】

【請求項１】

リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力手段と、

熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定手段と、

推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する更新手段とを備えた

ことを特徴とする学習装置。

【請求項２】

更新手段は、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新する

請求項１記載の学習装置。

【請求項３】

更新手段は、パラメータ更新後のワッサースタイン距離が大きくなるように、一回前の更新時のワッサースタイン距離の勾配に対する今回の更新時のワッサースタイン距離の勾配の比の値と、一回前の更新時のステップ幅との積の値以下のステップ幅で報酬関数のパラメータを更新する

請求項１または請求項２記載の学習装置。

【請求項４】

ワッサースタイン距離が収束したか否か判定する判定手段を備え、
ワッサースタイン距離が収束していないと判定された場合、推定手段は、熟練者の軌跡の確率分布と、更新された報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定し、更新手段は、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する
請求項１から請求項３のうちのいずれか１項に記載の学習装置。
【請求項５】
関数入力手段は、線形関数になるように特徴量が設定された報酬関数の入力を受け付ける
請求項１から請求項４のうちのいずれか１項に記載の学習装置。
【請求項６】
リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付け、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定し、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する
ことを、コンピュータに実行させることを特徴とする学習方法。
【請求項７】
非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新する
請求項６記載の学習方法。
【請求項８】
コンピュータに、
リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力処理、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定処理、および、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する更新処理
を実行させるための学習プログラム。
【請求項９】
コンピュータに、
更新処理で、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新させる
請求項８記載の学習プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、逆強化学習を行う学習装置、学習方法および学習プログラムに関する。
【背景技術】
【０００２】
機械学習の手法の一つに強化学習（ＲＬ：Reinforcement Learning）が知られている。強化学習は、様々な行動を試行錯誤しながら価値を最大化するような行動を学習する手法である。強化学習では、この価値を評価するための報酬関数が設定され、この報酬関数を最大にするような行動が探索される。しかし、報酬関数の設定は、一般には困難である。
【０００３】
この報酬関数の設定を容易にする方法として、逆強化学習（ＩＲＬ：Inverse Reinforcement Learning）が知られている。逆強化学習では、熟練者の意思決定履歴データを利用して、報酬関数を用いた最適化と、報酬関数のパラメータの更新とを繰り返すことで、

熟練者の意図を反映する報酬関数を生成する。
【０００４】
非特許文献１には、逆強化学習の一つである最大エントロピー逆強化学習（ＭＥ－ＩＲＬ：Maximum Entropy-IRL）について記載されている。非特許文献１に記載された方法では、熟練者のデータＤ＝｛τ_1，τ_2，…τ_N｝（ただし、τ_i＝（（s_1，a_1），（s_2，a_2），…，（s_N，a_N））からただ１つの報酬関数Ｒ（ｓ，ａ）＝θ・ｆ（ｓ，ａ）を推定する。この推定されたθを用いることで、熟練者の意思決定を再現できる。
【０００５】
また、非特許文献２には、最大エントロピー逆強化学習を改良した逆強化学習の手法の一つであるＧＣＬ（Guided Cost Learning）について記載されている。非特許文献２に記載された手法では、重点サンプリングを用いて報酬関数の重みを更新する。
【０００６】
また、報酬関数を学習する逆強化学習と、方策を直接学習する行動模倣とを合わせて、与えられた行動履歴を再現する模倣学習も知られている（例えば、非特許文献３参照）。
【先行技術文献】
【非特許文献】
【０００７】
【文献】B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning”, In AAAI, AAAI’08, 2008.
【文献】Chelsea Finn, Sergey Levine, Pieter Abbeel, "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization", Proceedings of The 33rd International Conference on Machine Learning, PMLR 48, pp.49-58, 2016.
【文献】Jonathan Ho, Stefano Ermon, "Generative adversarial imitation learning", NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp.4572-4580, December 2016
【発明の概要】
【発明が解決しようとする課題】
【０００８】
逆強化学習や模倣学習では、再現したい熟練者の行動履歴と、最適化された実行結果との差異を小さくするように報酬関数が学習される。非特許文献１～３に記載された逆強化学習や模倣学習では、ＫＬ（Kullback-Leibler）ダイバージェンスや、ＪＳ（Jensen-Shannon）ダイバージェンスのような確率的な距離で上述する差異が定義される。
【０００９】
ここで、報酬関数のパラメータを更新する際、一般には勾配法が用いられる。しかし、組み合わせ最適化問題では確率分布の設定が難しく、実問題の多くが属している組み合わせ最適化問題に上述するような逆強化学習を適用することが困難である。
【００１０】
そこで、本発明は、組み合わせ最適化問題において、安定的に逆強化学習を実施できる学習装置、学習方法および学習プログラムを提供することを目的とする。
【課題を解決するための手段】
【００１１】
本発明による学習装置は、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力手段と、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定手段と、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する更新手段とを備えたことを特徴とする。
【００１２】
本発明による学習方法は、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付け、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定

し、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新することを、コンピュータに実行させることを特徴とする。

【００１３】

本発明による学習プログラムは、コンピュータに、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力処理、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定処理、および、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する更新処理を実行させることを特徴とする。

【発明の効果】

【００１４】

本発明によれば、組み合わせ最適化問題において、安定的に逆強化学習を実施できる。

【図面の簡単な説明】

【００１５】

【図１】本発明による学習装置の一実施形態の構成例を示すブロック図である。

【図２】ワッサースタイン距離を用いた逆強化学習の例を示す説明図である。

【図３】学習装置の動作例を示すフローチャートである。

【図４】本発明による学習装置の概要を示すブロック図である。

【図５】少なくとも１つの実施形態に係るコンピュータの構成を示す概略ブロック図である。

【発明を実施するための形態】

【００１６】

まず初めに、一般的な逆強化学習を、組み合わせ最適化問題に適用することが困難な理由を説明する。非特許文献１に記載されたＭＥ－ＩＲＬでは、熟練者の軌跡（行動履歴）を再現する報酬関数が複数存在するという不定性を解決するため、最大エントロピー原理を用いて軌跡の分布を指定し、真の分布へ近づけること（すなわち、最尤推定）により報酬関数を学習する。

【００１７】

ＭＥ－ＩＲＬでは、軌跡τは、以下に例示する式１で表わされ、軌跡の分布$p_\theta(\tau)$を表わす確率モデルは、以下に例示する式２で表わされる。式２における$c_\theta(\tau)$は、コスト関数であり、符号を逆転させる（すなわち、$-c_\theta(\tau)$）ことで報酬関数$r_\theta(\tau)$を表わす（式３参照）。また、Ｚは、全ての軌跡に対する報酬の総和を表わす（式４参照）。

【００１８】

【数１】

$$\tau = \{(s_t, a_t) | t = 0, \cdots, T\} \quad \text{（式1）}$$

$$p_\theta(\tau) := \frac{1}{Z} \exp(-c_\theta(\tau)) \quad \text{（式2）}$$

ただし、

$$-c_\theta(\tau) = r_\theta(\tau) = \sum_{t=0}^{T} \gamma^t r_\theta(s_t, a_t) \quad \text{（式3）}$$

$$Z = \sum_{\tau} \exp(-c_\theta(\tau)) \quad \text{（式4）}$$

【００１９】

そして、最尤推定による報酬関数の重みの更新則（具体的には、勾配上昇法）は、以下に例示する式５および式６で表わされる。式５におけるαはステップ幅であり、$L_{ME}(\theta)$は、ＭＥ－ＩＲＬで用いられる分布間の距離尺度である。

【００２０】

【数２】

$$\theta \leftarrow \theta + \alpha \nabla_\theta L_{ME}(\theta) \quad \text{（式5）}$$

$$L_{ME}(\theta) := \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(\tau) = \frac{1}{N}\sum_{i=1}^{N}(-c_\theta(\tau^{(i)})) - \log \sum_{\tau} \exp(-c_\theta(\tau)) \quad \text{（式6）}$$

【００２１】

上述するように、式６における第２項は、全ての軌跡に対する報酬の総和である。ＭＥ－ＩＲＬでは、この第２項の値を厳密に計算できることを前提としている。しかし、現実的には、全ての軌跡に対する報酬の総和を計算することは困難であるため、非特許文献２に記載されたＧＣＬでは、重点サンプリングで近似的にこの値を算出する。

【００２２】

しかし、組み合わせ最適化問題は、離散的な値（言い換えると、連続的ではない値）をとるため、ある値を入力したときに、その値に対応する確率を返す確率分布の設定が難しい。組み合わせ最適化問題では、目的関数における値が少しでも変化すると、結果も大きく変化する可能性があるためである。

【００２３】

例えば、典型的な組み合わせ最適化問題の例として、経路問題やスケジューリング問題、切り出し・詰め込み問題や割り当て・マッチング問題などが挙げられる。具体的には、経路問題は、例えば、運搬経路問題や巡回セールスマン問題などであり、スケジューリング問題は、例えば、ジョブショップ問題や勤務スケジュール問題などである。また、切り出し・詰め込み問題は、例えば、ナップサック問題やビンパッキング問題などであり、割り当て・マッチング問題は、最大マッチング問題や一般化割当問題などである。

【００２４】

本開示の学習装置を用いることで、これらの組み合わせ最適化問題において、安定的に逆強化学習を実施することが可能になる。以下、本発明の実施形態を図面を参照して説明する。

【００２５】

図１は、本発明による学習装置の一実施形態の構成例を示すブロック図である。本実施形態の学習装置１００は、機械学習により、対象者（熟練者）の行動から報酬関数を推定する逆強化学習を行う装置であり、熟練者の行動特性に基づく情報処理を具体的に行う装置である。学習装置１００は、記憶部１０と、入力部２０と、特徴量設定部３０と、重み初期値設定部４０と、数理最適化実行部５０と、重み更新部６０と、収束判定部７０と、出力部８０とを備えている。

【００２６】

なお、数理最適化実行部５０、重み更新部６０および収束判定部７０により、後述する逆強化学習が行われることから、数理最適化実行部５０、重み更新部６０および収束判定部７０を含む装置を、逆強化学習装置と言うことができる。

【００２７】

記憶部１０は、学習装置１００が各種処理を行うために必要な情報を記憶する。記憶部１０は、後述する入力部２０が受け付けた熟練者の意思決定履歴データ（軌跡）を記憶し

てもよい。また、記憶部１０は、後述する数理最適化実行部５０および重み更新部６０が学習に用いる報酬関数の特徴量の候補を記憶していてもよい。ただし、特徴量の候補は、必ずしも目的関数に使用される特徴量である必要はない。

【００２８】

また、記憶部１０は、後述する数理最適化実行部５０を実現するための数理最適化ソルバを記憶していてもよい。なお、数理最適化ソルバの内容は任意であり、実行する環境や装置に応じて決定されればよい。

【００２９】

入力部２０は、学習装置１００が各種処理を行うために必要な情報の入力を受け付ける。入力部２０は、例えば、上述する熟練者の意思決定履歴データ（具体的には、状態と行動のペア）の入力を受け付けてもよい。また、入力部２０は、後述する逆強化学習装置が逆強化学習を行う際に用いる初期状態の制約ｚの入力を受け付けてもよい。

【００３０】

特徴量設定部３０は、状態および行動を含むデータから、報酬関数の特徴量を設定する。具体的には、特徴量設定部３０は、後述する逆強化学習装置が分布間の距離尺度としてワッサースタイン（Ｗａｓｓｅｒｓｔｅｉｎ）距離を利用できるように、関数全体で接線の勾配が有限になるように報酬関数の特徴量を設定する。特徴量設定部３０は、例えば、リプシッツ連続条件を満たすように報酬関数の特徴量を設定してもよい。

【００３１】

例えば、f_τを軌跡τの特徴量ベクトルとする。コスト関数$c_\theta(\tau)=\theta^T f_\tau$と線形に限定した場合、写像Ｆ：$\tau \to f_\tau$がリプシッツ連続であれば、$c_\theta(\tau)$もリプシッツ連続である。そのため、特徴量設定部３０は、報酬関数が線形関数になるように特徴量を設定してもよい。

【００３２】

なお、例えば、以下に例示する式７は、a_0において勾配が無限大になってしまうため、本開示において不適切な報酬関数と言える。

【００３３】

【数３】

$$f_\tau = \begin{cases} 1 & (a_0 \geq 0) \\ 0 & (otherwise) \end{cases} \quad \text{（式7）}$$

【００３４】

特徴量設定部３０は、例えば、ユーザの指示に応じて特徴量が設定された報酬関数を決定してもよく、記憶部１０からリプシッツ連続条件を満たす報酬関数を取得してもよい。

【００３５】

重み初期値設定部４０は、報酬関数の重みを初期化する。具体的には、重み初期値設定部４０は、報酬関数に含まれる個々の特徴量の重みを設定する。なお、重みを初期化する方法は特に限定されず、ユーザ等に応じて予め定められた任意の方法に基づいて重みが初期化されればよい。

【００３６】

数理最適化実行部５０は、熟練者の軌跡（行動履歴）の確率分布と、最適化された（報酬関数の）パラメータに基づいて決定される軌跡の確率分布との間の距離を最小にする軌跡$\hat{\tau}$（$\hat{\tau}$は、τの上付き$\hat{}$）を導出する。具体的には、数理最適化実行部５０は、分布間の距離尺度として、ＫＬ／ＪＳダイバージェンスの代わりにワッサースタイン距離を利用して、そのワッサースタイン距離を最小にするよう数理最適化を実行することにより、熟練者の軌跡$\hat{\tau}$を推定する。

【００３７】

ワッサースタイン距離は、以下に例示する式８で定義される。なお、ワッサースタイン距離の制約から、コスト関数ｃθ（τ）は、リプシッツ連続条件を満たす関数である必要がある。一方、本実施形態では、特徴量設定部３０によってリプシッツ連続条件を満たすように報酬関数の特徴量が設定されているため、数理最適化実行部５０は、以下に例示するようなワッサースタイン距離を利用することが可能になる。

【００３８】

【数４】

$$W(\theta) := \frac{1}{N}\sum_{i=1}^{N}(-c_\theta(\tau^{(i)})) - \frac{1}{N}\sum_{i=1}^{N}(-c_\theta(\hat{\tau}(\theta, z^{(i)}))) \quad \text{（式8）}$$

【００３９】

上記に例示する式８で定義されるワッサースタイン距離は０以下の値をとり、この値を大きくすることが、分布同士を近づけることに対応する。また、式８の第２項において、コスト関数ｃθの引数（すなわち、τ^（θ，ｚ（ｉ））は、パラメータθで最適化したｉ番目の軌跡を表わす。なお、ｚは、軌跡パラメータである。式８の第２項は、組み合わせ最適化問題でも算出可能な項である。そのため、式８に例示するワッサースタイン距離を分布間の距離尺度として用いることで、組み合わせ最適化問題において、安定的に逆強化学習を実施することが可能になる。

【００４０】

重み更新部６０は、推定された熟練者の軌跡τ^に基づいて分布間の距離尺度を最大にするように報酬関数のパラメータθを更新する。具体的には、重み更新部６０は、上述するワッサースタイン距離を最大にするように報酬関数のパラメータを更新する。重み更新部６０は、例えば、推定された軌跡τ^を固定して、勾配上昇法によりパラメータを更新してもよい。

【００４１】

また、本実施形態では、重み更新部６０は、報酬関数のパラメータを更新する際、ワッサースタイン距離を単調増加させるために、非拡大写像による更新則（以下、非拡大写像勾配法と記すこともある。）を用いてもよい。以下、非拡大写像勾配法について詳述する。

【００４２】

ここでは、報酬関数として線形関数が用いられる場合を例示する。上記のように軌跡τの特徴量ベクトルをｆτとすると、報酬関数は、以下に例示する式９のように表される。

【００４３】

【数５】

$$-c_\theta(\tau) = r_\theta(\tau) = \theta^\top f_\tau \quad \text{（式9）}$$

【００４４】

このとき、ワッサースタイン距離の単調増加性を保証するため、任意の軌跡τａおよび軌跡τｂ、並びに、各軌跡に対する特徴量ベクトルｆτａおよび特徴量ベクトルｆτｂについて、以下の式１０に例示する関係を満たす定数Ｋが存在する必要がある。

【００４５】

【数６】

$$\|\theta^\top f_{\tau_a} - \theta^\top f_{\tau_b}\| \leq K\|\tau_a - \tau_b\| \quad \text{（式10）}$$

【００４６】
ここで、上記に示す式１０は、以下に例示する式１１のように書き換えることができる。
【００４７】
【数７】

$$\|f_{\tau_a} - f_{\tau_b}\| \leq \tilde{K}\|\tau_a - \tau_b\| \quad \text{（式11）}$$

【００４８】
ｔ回目に更新される報酬関数のパラメータをθ_t、ワッサースタイン距離を$W(\theta_t)$、および、ステップ幅をα_tとする。このとき、報酬関数のパラメータの更新則は、以下に例示する式１２のように表すことができる。
【００４９】
【数８】

$$\theta_{t+1} = \theta_t + \alpha_t \nabla W(\theta_t) \quad \text{（式12）}$$

【００５０】
重み更新部６０は、報酬関数のパラメータの更新則（すなわち、θ（ｔ）→θ（ｔ＋１））が非拡大写像になるという制約下で、ワッサースタイン距離を大きくする勾配のステップ幅を探索し、そのステップ幅で報酬関数のパラメータを更新する。具体的には、重み更新部６０は、以下の式１３および式１４に示す条件を満たすステップ幅α_tで報酬関数のパラメータを更新する。
【００５１】
【数９】

$$0 < \alpha_t \leq \alpha_{t-1} \frac{\|\nabla W(\theta_{t-1})\|}{\|\nabla W(\theta_t)\|} \quad \text{（式13）}$$

$$W(\theta_{t+1}) > W(\theta_t) \quad \text{（式14）}$$

【００５２】
式１３および式１４は、パラメータ更新後のワッサースタイン距離が大きくなる（$W(\theta_{t+1}) > W(\theta_t)$）ように、一回前の更新時ｔ－１のワッサースタイン距離$W(\theta_{t-1})$の勾配$\nabla W(\theta_{t-1})$に対する今回の更新時ｔのワッサースタイン距離$W(\theta_t)$の勾配$\nabla W(\theta_t)$の比（$\|\nabla W(\theta_{t-1})\| / \|\nabla W_t(\theta)\|$）の値と一回前の更新時ｔ－１のステップ幅$\alpha_{t-1}$との積以下になるような正のステップ幅$\alpha_t$の値を探索することを示す。
【００５３】
例えば、組み合わせ最適化問題の場合、数理最適化実行部５０による推定結果が報酬関

数の変化に対して不連続になる場合がある。具体的には、ある値の最大化と最小化とを交互に実施する更新では、多くの場合、その値が振動し、収束するまでに時間を要することがある。一方、本実施形態では、数理最適化実行部５０が、上述する非拡大写像勾配法を用いることで、ワッサースタイン距離の単調増加性を保証しながらパラメータを更新することが可能になる。

【００５４】

以降、後述する収束判定部７０によって、ワッサースタイン距離が収束されたと判定されるまで、数理最適化実行部５０による軌跡の推定処理、および、重み更新部６０によるパラメータの更新処理が繰り返される。

【００５５】

収束判定部７０は、分布間の距離尺度が収束したか否か判定する。具体的には、収束判定部７０は、ワッサースタイン距離が収束したか否か判定する。判定方法は任意であり、収束判定部７０は、例えば、分布間のワッサースタイン距離の絶対値が予め定めた閾値より小さくなったときに、分布間の距離尺度が収束したと判定してもよい。

【００５６】

収束判定部７０は、距離が収束していないと判断した場合、数理最適化実行部５０および重み更新部６０による処理を継続させる。一方、収束判定部７０は、距離が収束したと判断した場合、数理最適化実行部５０および重み更新部６０による処理を終了させる。

【００５７】

出力部８０は、学習された報酬関数を出力する。

【００５８】

図２は、ワッサースタイン距離を用いた逆強化学習の例を示す説明図である。なお、本開示で示すワッサースタイン距離を用いた逆強化学習のことを、Wasserstein IRL （ＷＩＲＬ）と記すこともある。

【００５９】

まず、初期状態の制約ｚおよび初期値が設定されたパラメータθの報酬関数に基づき、最適化ソルバを用いて、ワッサースタイン距離を最小化するよう数理最適化を行うことで軌跡$\hat{\tau}$を推定する。なお、図２に例示する最適化ソルバは、数理最適化実行部５０に対応する。

【００６０】

一方、推定された軌跡$\hat{\tau}$と入力された熟練者の軌跡τに基づいて、ワッサースタイン距離を最大化するよう数理最適化を行うことで報酬関数（コスト関数）のパラメータを更新する。この処理は、重み更新部６０の処理に対応する。

【００６１】

以降、ワッサースタイン距離が収束したと判定されるまで、図２に例示する処理が繰り返される。

【００６２】

入力部２０と、特徴量設定部３０と、重み初期値設定部４０と、数理最適化実行部５０と、重み更新部６０と、収束判定部７０と、出力部８０とは、プログラム（学習プログラム）に従って動作するコンピュータのプロセッサ（例えば、ＣＰＵ（Central Processing Unit））によって実現される。

【００６３】

例えば、プログラムは、学習装置１００が備える記憶部１０に記憶され、プロセッサは、そのプログラムを読み込み、プログラムに従って、入力部２０、特徴量設定部３０、重み初期値設定部４０、数理最適化実行部５０、重み更新部６０、収束判定部７０および出力部８０として動作してもよい。また、学習装置１００の機能がＳａａＳ（Software as a Service）形式で提供されてもよい。

【００６４】

また、入力部２０と、特徴量設定部３０と、重み初期値設定部４０と、数理最適化実行部５０と、重み更新部６０と、収束判定部７０と、出力部８０とは、それぞれが専用のハ

ードウェアで実現されていてもよい。また、各装置の各構成要素の一部又は全部は、汎用または専用の回路（circuitry ）、プロセッサ等やこれらの組合せによって実現されもよい。これらは、単一のチップによって構成されてもよいし、バスを介して接続される複数のチップによって構成されてもよい。各装置の各構成要素の一部又は全部は、上述した回路等とプログラムとの組合せによって実現されてもよい。

【００６５】

また、学習装置１００の各構成要素の一部又は全部が複数の情報処理装置や回路等により実現される場合には、複数の情報処理装置や回路等は、集中配置されてもよいし、分散配置されてもよい。例えば、情報処理装置や回路等は、クライアントサーバシステム、クラウドコンピューティングシステム等、各々が通信ネットワークを介して接続される形態として実現されてもよい。

【００６６】

次に、本実施形態の学習装置１００の動作を説明する。図３は、本実施形態の学習装置１００の動作例を示すフローチャートである。入力部２０は、エキスパートデータ（すなわち、熟練者の軌跡／意思決定履歴データ）の入力を受け付ける（ステップＳ１１）。特徴量設定部３０は、状態および行動を含むデータから、リプシッツ連続条件を満たすように報酬関数の特徴量を設定する（ステップＳ１２）。また、重み初期値設定部４０は、報酬関数の重み（パラメータ）を初期化する（ステップＳ１３）。

【００６７】

数理最適化実行部５０は、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける（ステップＳ１４）。そして、数理最適化実行部５０は、ワッサースタイン距離を最小にするように数理最適化を実行する（ステップＳ１５）。具体的には、数理最適化実行部５０は、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する。

【００６８】

重み更新部６０は、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する（ステップＳ１６）。重み更新部６０は、例えば、非拡大写像勾配法を用いて報酬関数のパラメータを更新してもよい。

【００６９】

収束判定部７０は、ワッサースタイン距離が収束したか否か判定する（ステップＳ１７）。ワッサースタイン距離が収束していないと判定された場合（ステップＳ１７におけるＮｏ）、更新された軌跡を用いてステップＳ１５以降の処理が繰り返される。一方、ワッサースタイン距離が収束したと判定された場合（ステップＳ１７におけるＹｅｓ）、出力部８０は、学習された報酬関数を出力する（ステップＳ１８）。

【００７０】

以上のように、本実施形態では、数理最適化実行部５０が、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付け、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する。そして、重み更新部６０が、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する。よって、組み合わせ最適化問題において、安定的に逆強化学習を実施できる。

【００７１】

次に、本発明の概要を説明する。図４は、本発明による学習装置の概要を示すブロック図である。本発明による学習装置９０（例えば、学習装置１００）は、リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力手段９１（例えば、数理最適化実行部５０）と、熟練者の軌跡の確率分布と、報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定手段９２（例えば、数理最適化実行部５０）と、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する更新手

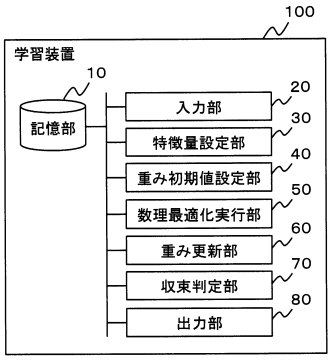
段９３（例えば、重み更新部６０）とを備えている。
【００７２】
そのような構成により、組み合わせ最適化問題において、安定的に逆強化学習を実施できる。
【００７３】
更新手段９３は、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新してもよい。
【００７４】
具体的には、更新手段９３は、パラメータ更新後のワッサースタイン距離（例えば、$W(\theta)$）が大きくなるように（すなわち、$W(\theta_{t+1}) > W(\theta_t)$）、一回前の更新時（ｔ－１回目）のワッサースタイン距離の勾配（例えば、$\nabla W(\theta_{t-1})$）に対する今回の更新時（例えば、ｔ回目）のワッサースタイン距離の勾配（例えば、$\nabla W(\theta_t)$）の比の値と、一回前の更新時のステップ幅（例えば、α_{t-1}）との積の値以下のステップ幅（例えば、α_t）で報酬関数のパラメータを更新してもよい（例えば、式１３および式１４参照）。
【００７５】
また、学習装置９０は、ワッサースタイン距離が収束したか否か判定する判定手段（例えば、収束判定部７０）を備えていてもよい。そして、ワッサースタイン距離が収束していないと判定された場合、推定手段９２は、熟練者の軌跡の確率分布と、更新された報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定し、更新手段９３は、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新してもよい。
【００７６】
また、関数入力手段９１は、線形関数になるように特徴量が設定された報酬関数の入力を受け付けてもよい。
【００７７】
図５は、少なくとも１つの実施形態に係るコンピュータの構成を示す概略ブロック図である。コンピュータ１０００は、プロセッサ１００１、主記憶装置１００２、補助記憶装置１００３、インタフェース１００４を備える。
【００７８】
上述の学習装置９０は、コンピュータ１０００に実装される。そして、上述した各処理部の動作は、プログラム（学習プログラム）の形式で補助記憶装置１００３に記憶されている。プロセッサ１００１は、プログラムを補助記憶装置１００３から読み出して主記憶装置１００２に展開し、当該プログラムに従って上記処理を実行する。
【００７９】
なお、少なくとも１つの実施形態において、補助記憶装置１００３は、一時的でない有形の媒体の一例である。一時的でない有形の媒体の他の例としては、インタフェース１００４を介して接続される磁気ディスク、光磁気ディスク、ＣＤ－ＲＯＭ（Compact Disc Read-only memory ）、ＤＶＤ－ＲＯＭ（Read-only memory）、半導体メモリ等が挙げられる。また、このプログラムが通信回線によってコンピュータ１０００に配信される場合、配信を受けたコンピュータ１０００が当該プログラムを主記憶装置１００２に展開し、上記処理を実行してもよい。
【００８０】
また、当該プログラムは、前述した機能の一部を実現するためのものであっても良い。さらに、当該プログラムは、前述した機能を補助記憶装置１００３に既に記憶されている他のプログラムとの組み合わせで実現するもの、いわゆる差分ファイル（差分プログラム）であってもよい。
【００８１】
上記の実施形態の一部又は全部は、以下の付記のようにも記載されうるが、以下には限られない。

【００８２】
（付記１）リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力手段と、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定手段と、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する更新手段とを備えた
ことを特徴とする学習装置。
【００８３】
（付記２）更新手段は、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新する
付記１記載の学習装置。
【００８４】
（付記３）更新手段は、パラメータ更新後のワッサースタイン距離が大きくなるように、一回前の更新時のワッサースタイン距離の勾配に対する今回の更新時のワッサースタイン距離の勾配の比の値と、一回前の更新時のステップ幅との積の値以下のステップ幅で報酬関数のパラメータを更新する
付記１または付記２記載の学習装置。
【００８５】
（付記４）ワッサースタイン距離が収束したか否か判定する判定手段を備え、
ワッサースタイン距離が収束していないと判定された場合、推定手段は、熟練者の軌跡の確率分布と、更新された報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定し、更新手段は、推定された軌跡に基づいてワッサースタイン距離を最大にするように報酬関数のパラメータを更新する
付記１から付記３のうちのいずれか１つに記載の学習装置。
【００８６】
（付記５）関数入力手段は、線形関数になるように特徴量が設定された報酬関数の入力を受け付ける
付記１から付記４のうちのいずれか１つに記載の学習装置。
【００８７】
（付記６）リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付け、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定し、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する
ことを特徴とする学習方法。
【００８８】
（付記７）非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新する
付記６記載の学習方法。
【００８９】
（付記８）コンピュータに、
リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力処理、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定処理、および、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数

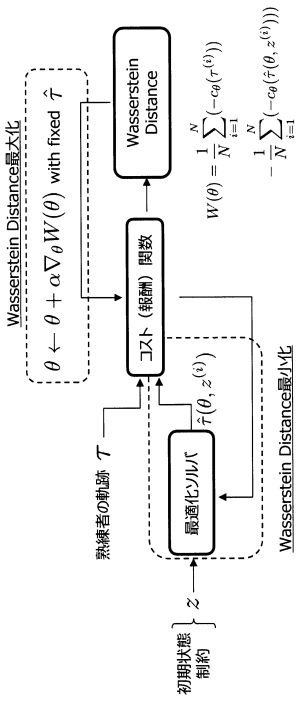
のパラメータを更新する更新処理
を実行させるための学習プログラムを記憶するプログラム記憶媒体。
【００９０】
（付記９）コンピュータに、
更新処理で、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新させる
ための学習プログラムを記憶する付記８記載のプログラム記憶媒体。
【００９１】
（付記１０）コンピュータに、
リプシッツ連続条件を満たすように特徴量が設定された報酬関数の入力を受け付ける関数入力処理、
熟練者の軌跡の確率分布と、前記報酬関数のパラメータに基づいて決定される軌跡の確率分布との距離を表わすワッサースタイン距離を最小にする軌跡を推定する推定処理、および、
推定された軌跡に基づいて前記ワッサースタイン距離を最大にするように前記報酬関数のパラメータを更新する更新処理
を実行させるための学習プログラム。
【００９２】
（付記１１）コンピュータに、
更新処理で、非拡大写像による更新則である非拡大写像勾配法を用いて報酬関数のパラメータを更新させる
付記１０記載の学習プログラム。
【符号の説明】
【００９３】
１０　記憶部
２０　入力部
３０　特徴量設定部
４０　重み初期値設定部
５０　数理最適化実行部
６０　重み更新部
７０　収束判定部
１００　学習装置

【図面】

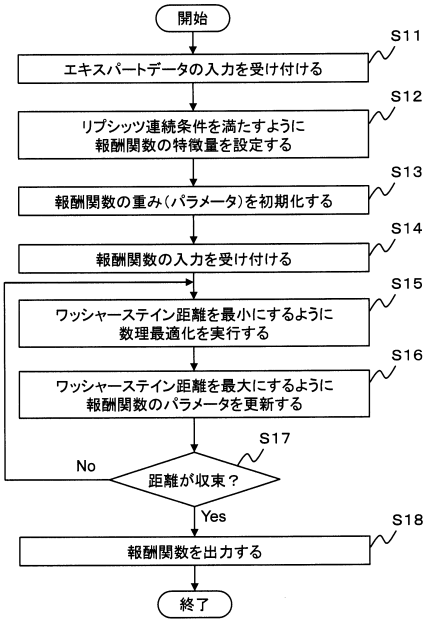
【図１】


100
学習装置
10
記憶部
20
入力部
30
特徴量設定部
40
重み初期値設定部
50
数理最適化実行部
60
重み更新部
70
収束判定部
80
出力部

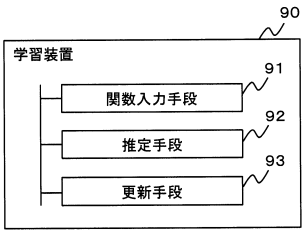
【図２】


Wasserstein Distance
Wasserstein Distance最大化
$\theta \leftarrow \theta + \alpha \nabla_\theta W(\theta)$ with fixed $\hat{\tau}$
コスト（報酬）関数
最適化ソルバ
熟練者の軌跡 τ
$\hat{\tau}(\theta, z^{(i)})$
z
初期状態制約
Wasserstein Distance最小化
$W(\theta) = \frac{1}{N}\sum_{i=1}^{N}(-c_\theta(\tau^{(i)})) - \frac{1}{N}\sum_{i=1}^{N}(-c_\theta(\hat{\tau}(\theta, z^{(i)})))$

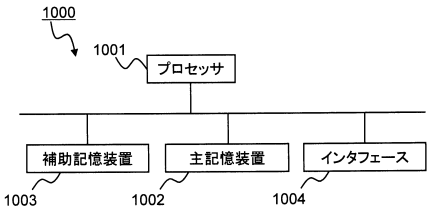
【図３】


開始
S11
エキスパートデータの入力を受け付ける
S12
リプシッツ連続条件を満たすように
報酬関数の特徴量を設定する
S13
報酬関数の重み（パラメータ）を初期化する
S14
報酬関数の入力を受け付ける
S15
ワッシャーステイン距離を最小にするように
数理最適化を実行する
S16
ワッシャーステイン距離を最大にするように
報酬関数のパラメータを更新する
S17
No
距離が収束？
Yes
S18
報酬関数を出力する
終了

【図４】


90
学習装置
91
関数入力手段
92
推定手段
93
更新手段

【図５】



フロントページの続き

(56)参考文献 国際公開第２０１９／１５５０５２（WO，Ａ１）
特開２０２０－１７７０１６（ＪＰ，Ａ）
国際公開第２０１８／１３１２１４（WO，Ａ１）
ラシュカ セバスチャン ほか，"17.3 畳み込み層とワッサースタイン距離を使って合成画像の品質を改善する"，［第3版］ Python機械学習プログラミング 達人データサイエンティストによる理論と実践，第1版，株式会社インプレス，2020年10月，pp. 548-567，ISBN 978-4-295-01007-4

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｎ ２０／００－２０／２０