Open Questions on the Markov Decision/Game Process

Yinyu Ye

Optimization and Algorithm Design

[1]Department of Management Science and Engineering and
Institute for Computational and Mathematical Engineering
Stanford University, Stanford

November 29, 2023

Table of Contents

Table of Contents

The Linear Programming Form of the MDP

$$\begin{array}{lll} \min_{\mathbf{x}} & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} (e_{ij} - \gamma p_{ij}) x_j & = 1, \ \forall i = 1, ..., m, \\ & x_j & \geq 0, \ \forall j, \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$, the action set at state i, and 0 otherwise; and $0 < \gamma < 1$ is the discount factor.

The Linear Programming Form of the MDP

$$
\begin{array}{lll}
\min_{\mathbf{x}} & \sum_{j=1}^{n} c_j x_j & \\
\text{s.t.} & \sum_{j=1}^{n} (e_{ij} - \gamma p_{ij}) x_j & = 1, \ \forall i = 1, ..., m, \\
& x_j & \geq 0, \ \forall j,
\end{array}
$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$, the action set at state i, and 0 otherwise; and $0 < \gamma < 1$ is the discount factor.

Primal variable x_j represents the expected jth action flux or frequency, that is, the expected present value of the number of times action j is chosen. The cost-to-go values are the "shadow Prices" of the LP problem.

The Linear Programming Form of the MDP

$$\begin{array}{lll} \min_{\mathbf{x}} & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} (e_{ij} - \gamma p_{ij}) x_j & = 1, \ \forall i = 1, ..., m, \\ & x_j & \geq 0, \ \forall j, \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$, the action set at state i, and 0 otherwise; and $0 < \gamma < 1$ is the discount factor.

Primal variable x_j represents the expected jth action flux or frequency, that is, the expected present value of the number of times action j is chosen. The cost-to-go values are the "shadow Prices" of the LP problem.

When transition probability p_{ij} is 0 or 1. then it is deterministic MDP.

The Linear Programming Form of the MDP

$$\begin{array}{lll} \min_{\mathbf{x}} & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} (e_{ij} - \gamma p_{ij}) x_j & = 1, \ \forall i = 1, ..., m, \\ & x_j & \geq 0, \ \forall j, \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$, the action set at state i, and 0 otherwise; and $0 < \gamma < 1$ is the discount factor.

Primal variable x_j represents the expected jth action flux or frequency, that is, the expected present value of the number of times action j is chosen. The cost-to-go values are the "shadow Prices" of the LP problem.

When transition probability p_{ij} is 0 or 1. then it is deterministic MDP.

When discount factor γ becomes γ_j, then the MDP has a non-uniform discount factors.

The "LP" Form of the MGP

$$
\begin{array}{rlll}
\min_{x_j \in \mathcal{A}_i, i \in I^-} \ \max_{x_j \in \mathcal{A}_i, i \in I^+} & \sum_{j=1}^{n} c_j x_j & & \\
\text{s.t.} & \sum_{j=1}^{n} (e_{ij} - \gamma p_{ij}) x_j & = 1, & \forall i = 1, ..., m, \\
& x_j & \geq 0, & \forall j.
\end{array}
$$

where the states are partitioned into two sets, I^- and I^+, one wants to minimize and the other wants to maximize the joint objective function.

The "LP" Form of the MGP

$$\min_{x_j \in \mathcal{A}_i, i \in I^-} \ \max_{x_j \in \mathcal{A}_i, i \in I^+} \quad \sum_{j=1}^n c_j x_j$$

$$\begin{array}{lrl} \text{s.t.} & \sum_{j=1}^n (e_{ij} - \gamma p_{ij}) x_j & = 1, \ \forall i = 1, ..., m, \\ & x_j & \geq 0, \ \forall j. \end{array}$$

where the states are partitioned into two sets, I^- and I^+, one wants to minimize and the other wants to maximize the joint objective function.

Again, when transition probability p_{ij} is 0 or 1. then it is deterministic turn-based zero-sum game..

Algorithmic Events of the MDP Methods

- Shapley (1953) and Bellman (1957) developed a method called the Value-Iteration (VI, first-order) method to approximate the optimal state cost-to-go values and an approximate optimal policy.

Algorithmic Events of the MDP Methods

- Shapley (1953) and Bellman (1957) developed a method called the Value-Iteration (VI, first-order) method to approximate the optimal state cost-to-go values and an approximate optimal policy.
- Another best known method is due to Howard (1960) and is known as the Policy-Iteration or multiple-pivot (PI, second-order) method, which generate an optimal policy in finite number of iterations.

Algorithmic Events of the MDP Methods

- Shapley (1953) and Bellman (1957) developed a method called the Value-Iteration (VI, first-order) method to approximate the optimal state cost-to-go values and an approximate optimal policy.
- Another best known method is due to Howard (1960) and is known as the Policy-Iteration or multiple-pivot (PI, second-order) method, which generate an optimal policy in finite number of iterations.
- de Ghellinck (1960), D'Epenoux (1960) and Manne (1960) showed that the MDP has an LP representation, so that it can be solved by the simplex method (known as the simple policy iteration) of Dantzig (1947), and in polynomial time by the Ellipsoid method and IPMs.

Complexities of the Policy Iteration and Simplex Methods

- In practice, the policy-iteration method, including the simple policy-iteration or Simplex method, has been remarkably successful and shown to be most effective and widely used, but its worst-case provable bound was something like $2^m/2$.

Complexities of the Policy Iteration and Simplex Methods

- In practice, the policy-iteration method, including the simple policy-iteration or Simplex method, has been remarkably successful and shown to be most effective and widely used, but its worst-case provable bound was something like $2^m/2$.

- In the past 60 years, many efforts have been made to resolve the worst-case complexity issue of the policy-iteration method or the Simplex method, and to answer the question: are they (strongly) polynomial-time algorithms?

Table of Contents

Complexity Theorem for MDP with Discount

- The classic simplex method (Dantzig pivoting rule) and the policy iteration method, starting from any policy, terminate in

$$\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$$

 iterations (Y 10) with $O(mn)$ operations per iteration - the first strongly-polynomial time algorithm when the discount factor is fixed.

Complexity Theorem for MDP with Discount

- The classic simplex method (Dantzig pivoting rule) and the policy iteration method, starting from any policy, terminate in

$$\frac{m(n-m)}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$$

iterations (Y 10) with $O(mn)$ operations per iteration - the first strongly-polynomial time algorithm when the discount factor is fixed.

- The policy-iteration method actually terminates

$$\frac{n}{1-\gamma} \cdot \log\left(\frac{m}{1-\gamma}\right),$$

iterations with at most $O(m^2 n)$ operations per iteration (Hansen/Miltersen/Zwick ACM 12).

High Level Ideas of the Proof

- Create a combinatorial event: a (non-optimal) action will never enter the (intermediate) policy again.

High Level Ideas of the Proof

- Create a combinatorial event: a (non-optimal) action will never enter the (intermediate) policy again.
- The event will happen in at most a certain polynomial number of iterations.

High Level Ideas of the Proof

- Create a combinatorial event: a (non-optimal) action will never enter the (intermediate) policy again.
- The event will happen in at most a certain polynomial number of iterations.
- More precisely, after $\frac{m}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ iterations, a new non-optimal action would be implicitly eliminated from appearance in any future policies generated by the simplex or policy-iteration method.

High Level Ideas of the Proof

- Create a combinatorial event: a (non-optimal) action will never enter the (intermediate) policy again.
- The event will happen in at most a certain polynomial number of iterations.
- More precisely, after $\frac{m}{1-\gamma} \cdot \log\left(\frac{m^2}{1-\gamma}\right)$ iterations, a new non-optimal action would be implicitly eliminated from appearance in any future policies generated by the simplex or policy-iteration method.
- The event then repeats for another non-optimal state-action, and there are no more than $(n-m)$ non-optimal actions to eliminate.

The Turn-Based Two-Person Zero-Sum Game

Hansen/Miltersen/Zwick ACM 12 also proved that the strategy iteration method also terminates

$$\frac{n}{1-\gamma} \cdot \log\left(\frac{m}{1-\gamma}\right)$$

iterations for MGP – the first strongly-polynomial and polynomial time algorithm when the discount factor is fixed.

The Turn-Based Two-Person Zero-Sum Game

Hansen/Miltersen/Zwick ACM 12 also proved that the strategy iteration method also terminates

$$\frac{n}{1-\gamma} \cdot \log\left(\frac{m}{1-\gamma}\right)$$

iterations for MGP – the first strongly-polynomial and polynomial time algorithm when the discount factor is fixed.

The strategy iteration method is the best-response method: the leader make a policy iteration, then the follower make the best policy given the leader's policy.

Deterministic MDP with Discounts

- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the discount factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y 16).

Deterministic MDP with Discounts

- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the discount factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y 16).

- Theorem: The simplex method for deterministic MDP with non-uniform discount factors, regardless discount factor values, terminates in $O(m^5 n^3 \log^2 m)$ iterations (Post/Y 16).

Deterministic MDP with Discounts

- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the discount factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y 16).
- Theorem: The simplex method for deterministic MDP with non-uniform discount factors, regardless discount factor values, terminates in $O(m^5 n^3 \log^2 m)$ iterations (Post/Y 16).
- Hansen/Miltersen/Zwick 15 was able to reduce a factor m from the bound.

Table of Contents

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Is the policy iteration method strongly polynomial for the deterministic MDP?

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Is the policy iteration method strongly polynomial for the deterministic MDP?
- Is there a polynomial time method for MGP (logarithmic dependence of $1/(1-\gamma)$, using IPM by the Leader)?

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Is the policy iteration method strongly polynomial for the deterministic MDP?
- Is there a polynomial time method for MGP (logarithmic dependence of $1/(1-\gamma)$, using IPM by the Leader)?
- Is there a strongly-polynomial time method for the deterministic MGP (independent of γ, extension from PY 16)?

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Is the policy iteration method strongly polynomial for the deterministic MDP?
- Is there a polynomial time method for MGP (logarithmic dependence of $1/(1-\gamma)$, using IPM by the Leader)?
- Is there a strongly-polynomial time method for the deterministic MGP (independent of γ, extension from PY 16)?
- Is there a strongly polynomial-time algorithm for MDP regardless the discount factor?

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Feinberg/Huang 2013, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Is the policy iteration method strongly polynomial for the deterministic MDP?
- Is there a polynomial time method for MGP (logarithmic dependence of $1/(1-\gamma)$, using IPM by the Leader)?
- Is there a strongly-polynomial time method for the deterministic MGP (independent of γ, extension from PY 16)?
- Is there a strongly polynomial-time algorithm for MDP regardless the discount factor?