Distributionally Robust Stochastic Optimization and Learning

Models/Algorithms for Data-Driven Optimization and Learning

Yinyu Ye

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

US & Mexico Workshop on Optimization and its Applications in Honor of Don Goldfarb

January 8-12, 2018

Outline

- Computation and Sample Complexity of Solving Markov Decision/Game Processes
- Distributionally Robust Optimization under Moment, Likelihood and Wasserstein Bounds, and its Applications

Outline

- Computation and Sample Complexity of Solving Markov Decision/Game Processes
- Distributionally Robust Optimization under Moment, Likelihood and Wasserstein Bounds, and its Applications

Analyze and develop tractable and provable models and algorithms for optimization with uncertain and sampling data.

Table of Contents

The Markov Decision/Game Process

- Markov decision processes (MDPs) provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.

The Markov Decision/Game Process

- Markov decision processes (MDPs) provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.
- Markov game processes (MGPs) provide a mathematical framework for modeling sequential decision-making of two-person turn-based zero-sum game.

The Markov Decision/Game Process

- Markov decision processes (MDPs) provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.
- Markov game processes (MGPs) provide a mathematical framework for modeling sequential decision-making of two-person turn-based zero-sum game.
- MDGPs are useful for studying a wide range of optimization/game problems solved via dynamic programming, where it was known at least as early as the 1950s (cf. Shapley 1953, Bellman 1957).

The Markov Decision/Game Process

- Markov decision processes (MDPs) provide a mathematical framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision maker.
- Markov game processes (MGPs) provide a mathematical framework for modeling sequential decision-making of two-person turn-based zero-sum game.
- MDGPs are useful for studying a wide range of optimization/game problems solved via dynamic programming, where it was known at least as early as the 1950s (cf. Shapley 1953, Bellman 1957).
- Modern applications include dynamic planning under uncertainty, reinforcement learning, social networking, and almost all other stochastic dynamic/sequential decision/game problems in Mathematical, Physical, Management and Social Sciences.

The Markov Decision Process/Game continued

- At each time step, the process is in some state $i = 1, \ldots, m$, and the decision maker chooses an action $j \in \mathcal{A}_i$ that is available in state i, and giving the decision maker an immediate corresponding cost c_j.

The Markov Decision Process/Game continued

- At each time step, the process is in some state $i = 1, ..., m$, and the decision maker chooses an action $j \in \mathcal{A}_i$ that is available in state i, and giving the decision maker an immediate corresponding cost c_j.
- The process responds at the next time step by randomly moving into a new state i'. The probability that the process enters i' is influenced by the chosen action in state i. Specifically, it is given by the state transition distribution probability $\mathbf{p}_j \in \mathbf{R}^m$.

The Markov Decision Process/Game continued

- At each time step, the process is in some state $i = 1, \ldots, m$, and the decision maker chooses an action $j \in \mathcal{A}_i$ that is available in state i, and giving the decision maker an immediate corresponding cost c_j.
- The process responds at the next time step by randomly moving into a new state i'. The probability that the process enters i' is influenced by the chosen action in state i. Specifically, it is given by the state transition distribution probability $\mathbf{p}_j \in \mathbf{R}^m$.
- But given state/action j, the distribution is conditionally independent of all previous states and actions; in other words, the state transitions of an MDP possess the Markov property.

MDP Stationary Policy and Cost-to-Go Value

- A stationary policy for the decision maker is a function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies an action in each state, $\pi_i \in \mathcal{A}_i$, that the decision maker will always choose; which also lead to a cost-to-go value for each state

MDP Stationary Policy and Cost-to-Go Value

- A stationary policy for the decision maker is a function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies an action in each state, $\pi_i \in \mathcal{A}_i$, that the decision maker will always choose; which also lead to a cost-to-go value for each state
- The MDP is to find a stationary policy to minimize/maximize the expected discounted sum over the infinite horizon with a discount factor $0 \leq \gamma < 1$.

MDP Stationary Policy and Cost-to-Go Value

- A stationary policy for the decision maker is a function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies an action in each state, $\pi_i \in \mathcal{A}_i$, that the decision maker will always choose; which also lead to a cost-to-go value for each state
- The MDP is to find a stationary policy to minimize/maximize the expected discounted sum over the infinite horizon with a discount factor $0 \leq \gamma < 1$.
- If the states are partitioned into two sets, one is to minimize and the other is to maximize the discounted sum, then the process becomes a two-person turn-based zero-sum stochastic game.

MDP Stationary Policy and Cost-to-Go Value

- A stationary policy for the decision maker is a function $\pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ that specifies an action in each state, $\pi_i \in \mathcal{A}_i$, that the decision maker will always choose; which also lead to a cost-to-go value for each state
- The MDP is to find a stationary policy to minimize/maximize the expected discounted sum over the infinite horizon with a discount factor $0 \leq \gamma < 1$.
- If the states are partitioned into two sets, one is to minimize and the other is to maximize the discounted sum, then the process becomes a two-person turn-based zero-sum stochastic game.
- Typically, discount factor $\gamma = \frac{1}{1+\rho}$ where ρ is the interest rate, where we assume it is uniform among all actions.

The Optimal Cost-to-Go Value Vector

Let $\mathbf{y} \in \mathbf{R}^m$ represent the cost-to-go values of the m states, one entry for each state i, of a given policy.

The Optimal Cost-to-Go Value Vector

Let $\mathbf{y} \in \mathbf{R}^m$ represent the cost-to-go values of the m states, one entry for each state i, of a given policy.

The MDP problem entails choosing the optimal value vector $\mathbf{y}^*$ such that it is the fixed point:

$$y_i^* = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i,$$

with optimal policy

$$\pi_i^* = \arg\min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i.$$

The Optimal Cost-to-Go Value Vector

Let $\mathbf{y} \in \mathbf{R}^m$ represent the cost-to-go values of the m states, one entry for each state i, of a given policy.

The MDP problem entails choosing the optimal value vector $\mathbf{y}^*$ such that it is the fixed point:

$$y_i^* = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i,$$

with optimal policy

$$\pi_i^* = \arg\min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i.$$

In the Game setting, the fixed point becomes:

$$y_i^* = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i \in I^-,$$

and

$$y_i^* = \max\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*, \ \forall j \in \mathcal{A}_i\}, \ \forall i \in I^+.$$

The Linear Programming Form of the MDP

The fixed-point vector can be formulated as

$$
\begin{aligned}
\text{maximize}_{\mathbf{y}} \quad & \textstyle\sum_{i=1}^{m} y_i \\
\text{subject to} \quad & y_1 \leq c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_1 \\
& \cdots \quad \cdots \quad \cdots \\
& y_i \leq c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_i \\
& \cdots \quad \cdots \quad \cdots \\
& y_m \leq c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_m,
\end{aligned}
$$

where $\mathcal{A}_i$ represents all actions available in state i, and $\mathbf{p}_j$ is the state transition probabilities to all states when action j is taken.

The Linear Programming Form of the MDP

The fixed-point vector can be formulated as

$$
\begin{array}{lrcl}
\text{maximize}_{\mathbf{y}} & \sum_{i=1}^{m} y_i & & \\
\text{subject to} & y_1 & \leq & c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_1 \\
& \cdots & \cdots & \cdots \\
& y_i & \leq & c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_i \\
& \cdots & \cdots & \cdots \\
& y_m & \leq & c_j + \gamma \mathbf{p}_j^T \mathbf{y}, \ \forall j \in \mathcal{A}_m,
\end{array}
$$

where $\mathcal{A}_i$ represents all actions available in state i, and $\mathbf{p}_j$ is the state transition probabilities to all states when action j is taken.

This is the Standard Dual LP form.

The Primal LP Form of the MDP

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}} & \sum_{j=1}^{n} x_j & \\ \text{subject to} & \sum_{j=1}^{n}(e_{ij} - \gamma p_{ij})x_j & = 1, \ \forall i, \\ & x_j & \geq 0, \ \forall j. \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$ and 0 otherwise.

The Primal LP Form of the MDP

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}} & \sum_{j=1}^{n} x_j & \\ \text{subject to} & \sum_{j=1}^{n}(e_{ij} - \gamma p_{ij})x_j & = 1, \ \forall i, \\ & x_j & \geq 0, \ \forall j. \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$ and 0 otherwise.

Primal variable x_j represents the expected jth action flow 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 Primal LP Form of the MDP

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}} & \sum_{j=1}^{n} x_j & \\ \text{subject to} & \sum_{j=1}^{n}(e_{ij} - \gamma p_{ij})x_j & = 1, \ \forall i, \\ & x_j & \geq 0, \ \forall j. \end{array}$$

where $e_{ij} = 1$ when $j \in \mathcal{A}_i$ and 0 otherwise.

Primal variable x_j represents the expected jth action flow 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 discount factor γ becomes γ_j, then the MDP has a non-uniform discount factors.

Algorithmic Events of the MDP Methods

- Shapley (1953) and Bellman (1957) developed a method called the Value-Iteration (VI) 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) 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 (PI) method, which generate an optimal policy in finite number of iterations in a distributed and decentralized way, where two key procedures are the policy evaluation and the policy improvement.

Algorithmic Events of the MDP Methods

- Shapley (1953) and Bellman (1957) developed a method called the Value-Iteration (VI) 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 (PI) method, which generate an optimal policy in finite number of iterations in a distributed and decentralized way, where two key procedures are the policy evaluation and the policy improvement.
- 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 of Dantzig (1947) in finite number of steps, and the Ellipsoid method of Kachiyan (1979) in polynomial time.

Open Question on the Complexity of the Policy Iteration Method

- 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.

Open Question on the Complexity of the Policy Iteration Method

- 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.

- In the past 50 years, many efforts have been made to resolve the worst-case complexity issue of the policy-iteration method, and to answer the question: are they also efficient in Theory?

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 MOR10).

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 MOR10).
- 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 ACM12).

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

- Again, the states are partitioned into two sets where one set is to maximize and the other is to minimize.

The Turn-Based Two-Person Zero-Sum Game

- Again, the states are partitioned into two sets where one set is to maximize and the other is to minimize.
- It does not admit a convex programming formulation, and it is unknown if it can be solved in polynomial time in general.

The Turn-Based Two-Person Zero-Sum Game

- Again, the states are partitioned into two sets where one set is to maximize and the other is to minimize.
- It does not admit a convex programming formulation, and it is unknown if it can be solved in polynomial time in general.
- Strategy-Iteration Method: One player continues policy iterations from the policy where the other player chooses the best-response action in every one of his or her state set.

The Turn-Based Two-Person Zero-Sum Game

- Again, the states are partitioned into two sets where one set is to maximize and the other is to minimize.
- It does not admit a convex programming formulation, and it is unknown if it can be solved in polynomial time in general.
- Strategy-Iteration Method: One player continues policy iterations from the policy where the other player chooses the best-response action in every one of his or her state set.
- Hansen/Miltersen/Zwick ACM12 proved that the strategy iteration method also terminates

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

iterations – the first strongly polynomial time algorithm when the discount factor is fixed.

Deterministic MDP with Discount

- Every probability distribution contains exactly one 1 and 0 everywhere else, where the primal LP problem resembles the generalized cycle flow problem.

Deterministic MDP with Discount

- Every probability distribution contains exactly one 1 and 0 everywhere else, where the primal LP problem resembles the generalized cycle flow problem.
- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y MOR2016).

Deterministic MDP with Discount

- Every probability distribution contains exactly one 1 and 0 everywhere else, where the primal LP problem resembles the generalized cycle flow problem.
- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y MOR2016).
- Theorem: The simplex method for deterministic MDP with non-uniform discount factors, regardless factor values, terminates in $O(m^5 n^3 \log^2 m)$ iterations (Post/Y MOR2016).

Deterministic MDP with Discount

- Every probability distribution contains exactly one 1 and 0 everywhere else, where the primal LP problem resembles the generalized cycle flow problem.
- Theorem: The simplex method for deterministic MDP with a uniform discount factor, regardless the factor value, terminates in $O(m^3 n^2 \log^2 m)$ iterations (Post/Y MOR2016).
- Theorem: The simplex method for deterministic MDP with non-uniform discount factors, regardless factor values, terminates in $O(m^5 n^3 \log^2 m)$ iterations (Post/Y MOR2016).
- Hansen/Miltersen/Zwick 15 were able to reduce a factor m from the bound.

The Value-Iteration Method (VI)

Let $\mathbf{y}^0 \in \mathbf{R}^m$ represent the initial cost-to-go values of the m states.

The Value-Iteration Method (VI)

Let $\mathbf{y}^0 \in \mathbf{R}^m$ represent the initial cost-to-go values of the m states.

The VI for MDP:

$$y_i^{k+1} = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k,\ \forall j \in \mathcal{A}_i\},\ \forall i.$$

The Value-Iteration Method (VI)

Let $\mathbf{y}^0 \in \mathbf{R}^m$ represent the initial cost-to-go values of the m states.

The VI for MDP:

$$y_i^{k+1} = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k,\ \forall j \in \mathcal{A}_i\},\ \forall i.$$

The VI for MGP

$$y_i^{k+1} = \min\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k,\ \forall j \in \mathcal{A}_i\},\ \forall i \in I^-,$$

and

$$y_i^{k+1} = \max\{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k,\ \forall j \in \mathcal{A}_i\},\ \forall i \in I^+.$$

The values inside the parenthesis are the so-called Q-values.

Sample Value-Iteration

- Rather than compute each quantity $\mathbf{p}_j^T \mathbf{y}^k$ exactly, we approximate it by sampling, that is, we construct a sparser sample distribution $\hat{\mathbf{p}}_j$ for the evaluation. (Thus, the method does not need to know $\mathbf{p}_j$ exactly).

Sample Value-Iteration

- Rather than compute each quantity $\mathbf{p}_j^T \mathbf{y}^k$ exactly, we approximate it by sampling, that is, we construct a sparser sample distribution $\hat{\mathbf{p}}_j$ for the evaluation. (Thus, the method does not need to know $\mathbf{p}_j$ exactly).
- Even we know $\mathbf{p}_j$ exactly, it may be too dense so that the computation of $\mathbf{p}_j^T \mathbf{y}^k$ takes $O(m)$ up to operations.

Sample Value-Iteration

- Rather than compute each quantity $\mathbf{p}_j^T \mathbf{y}^k$ exactly, we approximate it by sampling, that is, we construct a sparser sample distribution $\hat{\mathbf{p}}_j$ for the evaluation. (Thus, the method does not need to know $\mathbf{p}_j$ exactly).
- Even we know $\mathbf{p}_j$ exactly, it may be too dense so that the computation of $\mathbf{p}_j^T \mathbf{y}^k$ takes $O(m)$ up to operations.
- We analyze this performance using Hoeffdings inequality and classic results on contraction properties of value iteration. Moreover, we improve the final result using Variance Reduction and Monotone Iteration.

Sample Value-Iteration

- Rather than compute each quantity $\mathbf{p}_j^T \mathbf{y}^k$ exactly, we approximate it by sampling, that is, we construct a sparser sample distribution $\hat{\mathbf{p}}_j$ for the evaluation. (Thus, the method does not need to know $\mathbf{p}_j$ exactly).
- Even we know $\mathbf{p}_j$ exactly, it may be too dense so that the computation of $\mathbf{p}_j^T \mathbf{y}^k$ takes $O(m)$ up to operations.
- We analyze this performance using Hoeffdings inequality and classic results on contraction properties of value iteration. Moreover, we improve the final result using Variance Reduction and Monotone Iteration.
- Variance Reduction enables us to update the Q-values so that the needed number of samples is decreased from iteration to iteration.

Sample Value-Iteration Results

Two results are developed (Sidford, Wang, Wu and Y [2017]):

- Knowing $\mathbf{p}_j$:

$$O\left((mn + \frac{n}{(1-\gamma)^3})\log(\frac{1}{\epsilon})\log(\frac{1}{\delta})\right)$$

to compute an ϵ-optimal policy with probability at least $1 - \delta$.

Sample Value-Iteration Results

Two results are developed (Sidford, Wang, Wu and Y [2017]):

- Knowing $\mathbf{p}_j$:

$$O\left((mn + \frac{n}{(1-\gamma)^3})\log(\frac{1}{\epsilon})\log(\frac{1}{\delta})\right)$$

  to compute an ϵ-optimal policy with probability at least $1-\delta$.

- Pure Sampling:

$$O\left(\frac{n}{(1-\gamma)^4\epsilon^2}\log(\frac{1}{\delta})\right)$$

  to compute an ϵ-optimal policy with probability at least $1-\delta$.

Sample Value-Iteration Results

Two results are developed (Sidford, Wang, Wu and Y [2017]):

- Knowing $\mathbf{p}_j$:

$$O\left((mn + \frac{n}{(1-\gamma)^3})\log(\frac{1}{\epsilon})\log(\frac{1}{\delta})\right)$$

  to compute an ϵ-optimal policy with probability at least $1-\delta$.

- Pure Sampling:

$$O\left(\frac{n}{(1-\gamma)^4\epsilon^2}\log(\frac{1}{\delta})\right)$$

  to compute an ϵ-optimal policy with probability at least $1-\delta$.

- Sample lower bound: $O\left(\frac{n}{(1-\gamma)^3\epsilon^2}\right)$.

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Kitahara and Mizuno 2012, Feinberg/Huang 213, 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., Kitahara and Mizuno 2012, Feinberg/Huang 213, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Lin, Sidford, Wang, Wu and Y 2018 on approximate PI method to achieve the optimal sample complexity.

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Kitahara and Mizuno 2012, Feinberg/Huang 213, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Lin, Sidford, Wang, Wu and Y 2018 on approximate PI method to achieve the optimal sample complexity.
- Lin, Sidford, Wang, Wu and Y 2018 on approximate PI method for solving Ergodic MDP where the dependence on γ is removed.

More Results and Extensions

- Renewed exciting research work on the simplex method, e.g., Kitahara and Mizuno 2012, Feinberg/Huang 213, Lee/Epelman/Romeijn/Smith 2013, Scherrer 2014, Fearnley/Savani 2014, Adler/Papadimitriou/Rubinstein 2014, etc.
- Lin, Sidford, Wang, Wu and Y 2018 on approximate PI method to achieve the optimal sample complexity.
- Lin, Sidford, Wang, Wu and Y 2018 on approximate PI method for solving Ergodic MDP where the dependence on γ is removed.
- All results are extended to the discounted Markov Game Process.

Remarks and Open Problems

- Dynamic sampling over actions in each iteration to deal with a large number of actions in each state?

Remarks and Open Problems

- Dynamic sampling over actions in each iteration to deal with a large number of actions in each state?
- Dimension reduction to reduce the number of states?
- Is there a simplex-type method that is (strongly) polynomial for the deterministic MGP (independent of γ)?

Remarks and Open Problems

- Dynamic sampling over actions in each iteration to deal with a large number of actions in each state?
- Dimension reduction to reduce the number of states?
- Is there a simplex-type method that is (strongly) polynomial for the deterministic MGP (independent of γ)?
- Is there an algorithm whose running time is PTAS for the general MGP?

Remarks and Open Problems

- Dynamic sampling over actions in each iteration to deal with a large number of actions in each state?
- Dimension reduction to reduce the number of states?
- Is there a simplex-type method that is (strongly) polynomial for the deterministic MGP (independent of γ)?
- Is there an algorithm whose running time is PTAS for the general MGP?
- Is there a strongly polynomial-time algorithm for MDP regardless the discount factor?

Remarks and Open Problems

- Dynamic sampling over actions in each iteration to deal with a large number of actions in each state?
- Dimension reduction to reduce the number of states?
- Is there a simplex-type method that is (strongly) polynomial for the deterministic MGP (independent of γ)?
- Is there an algorithm whose running time is PTAS for the general MGP?
- Is there a strongly polynomial-time algorithm for MDP regardless the discount factor?
- Is there a strongly polynomial-time algorithm for LP?

Table of Contents

Introduction to DRO

We start from considering a stochastic optimization problem as follows:

$$\text{maximize}_{\mathbf{x}\in X} \quad \mathbb{E}_{F_\xi}[h(\mathbf{x},\xi)] \tag{1}$$

where $\mathbf{x}$ is the decision variable with feasible region X, ξ represents random variables satisfying joint distribution F_ξ.

Introduction to DRO

We start from considering a stochastic optimization problem as follows:

$$\text{maximize}_{\mathbf{x}\in X} \quad \mathbb{E}_{F_\xi}[h(\mathbf{x},\xi)] \tag{1}$$

where $\mathbf{x}$ is the decision variable with feasible region X, ξ represents random variables satisfying joint distribution F_ξ.

- Pros: In many cases, the expected value is a good measure of performance
- Cons: One has to know the exact distribution of ξ to perform the stochastic optimization. Deviant from the assumed distribution may result in sub-optimal solutions. Even know the distribution, the solution/decision is generically risky.

Learning with Noises

"panda"

57.7% confidence

"gibbon"

99.3% confidence

Learning with Noises

"panda"

57.7% confidence

"gibbon"

99.3% confidence

Goodfellow et al. [2014]

Robust Optimization

In order to overcome the lack of knowledge on the distribution, people proposed the following (static) robust optimization approach:

$$\text{maximize}_{\mathbf{x}\in X} \quad \min_{\xi\in\Xi} h(\mathbf{x},\xi) \tag{2}$$

where Ξ is the support of ξ.

Robust Optimization

In order to overcome the lack of knowledge on the distribution, people proposed the following (static) robust optimization approach:

$$\text{maximize}_{\mathbf{x}\in X} \quad \min_{\xi\in\Xi} h(\mathbf{x},\xi) \tag{2}$$

where Ξ is the support of ξ.

- Pros: Robust to any distribution; only the support of the parameters are needed.
- Cons: Too conservative. The decision that maximizes the worst-case pay-off may perform badly in usual cases; e.g., Ben-Tal and Nemirovski [1998, 2000], etc.

Motivation for a Middle Ground

- In practice, although the exact distribution of the random variables may not be known, people usually know certain observed samples or training data and other statistical information.

Motivation for a Middle Ground

- In practice, although the exact distribution of the random variables may not be known, people usually know certain observed samples or training data and other statistical information.

- Thus we could choose an intermediate approach between stochastic optimization, which has no robustness in the error of distribution; and the robust optimization, which admits vast unrealistic single-point distribution on the support set of random variables.

Distributionally Robust Optimization

A solution to the above-mentioned question is to take the following Distributionally Robust Optimization/Learning (DRO) model:

$$\text{maximize}_{\mathbf{x}\in X} \quad \min_{F_\xi \in \mathcal{D}} \mathbb{E}_{F_\xi}[h(\mathbf{x}, \xi)] \tag{3}$$

In DRO, we consider a set of distributions $\mathcal{D}$ and choose one to maximize the expected value for any given $\mathbf{x} \in X$.

Distributionally Robust Optimization

A solution to the above-mentioned question is to take the following Distributionally Robust Optimization/Learning (DRO) model:

$$\text{maximize}_{\mathbf{x}\in X} \quad \min_{F_\xi \in \mathcal{D}} \mathbb{E}_{F_\xi}[h(\mathbf{x}, \xi)] \tag{3}$$

In DRO, we consider a set of distributions $\mathcal{D}$ and choose one to maximize the expected value for any given $\mathbf{x} \in X$.

When choosing $\mathcal{D}$, we need to consider the following:

- Tractability
- Practical (Statistical) Meanings
- Performance (the potential loss comparing to the benchmark cases)

Sample History of DRO

- First introduced by Scarf [1958] in the context of inventory control problem with a single random demand variable.
- Distribution set based on moments: Dupacova [1987], Prekopa [1995], Bertsimas and Popescu [2005], Delage and Y [2009,2010], etc
- Distribution set based on Likelihood/Divergences: Nilim and El Ghaoui [2005], Iyanger [2005], Wang, Glynn and Y [2012], etc
- Distribution set based on Wasserstein ambiguity set: Mohajerin Esfahani and Kuhn [2015], Blanchet et al. [2016], Duchi et al. [2016,17], Gao et al. [2017]
- Axiomatic motivation for DRO: Delage et al. [2017]; Ambiguous Joint Chance Constraints Under Mean and Dispersion Information: Hanasusanto et al. [2017]

DRO with Moment Bounds

Define

$$\mathcal{D} = \left\{ F_\xi \;\middle|\; \begin{array}{l} P(\xi \in \Xi) = 1 \\ (\mathbb{E}[\xi] - \mu_0)^T \Sigma_0^{-1} (\mathbb{E}[\xi] - \mu_0) \le \gamma_1 \\ \mathbb{E}[(\xi - \mu_0)(\xi - \mu_0)^T] \preceq \gamma_2 \Sigma_0 \end{array} \right\}$$

That is, the distribution set is defined based on the support, first and second order moments constraints.

DRO with Moment Bounds

Define

$$\mathcal{D} = \left\{ F_\xi \;\middle|\; \begin{array}{l} P(\xi \in \Xi) = 1 \\ (\mathbb{E}[\xi] - \mu_0)^T \Sigma_0^{-1} (\mathbb{E}[\xi] - \mu_0) \leq \gamma_1 \\ \mathbb{E}[(\xi - \mu_0)(\xi - \mu_0)^T] \preceq \gamma_2 \Sigma_0 \end{array} \right\}$$

That is, the distribution set is defined based on the support, first and second order moments constraints.

Theorem

Under mild technical conditions, the DRO model can be solved to any precision ϵ in time polynomial in $\log(1/\epsilon)$ and the sizes of $\mathbf{x}$ and ξ

Delage and Y [2010]

Confidence Region on F_ξ

Does the construction of $\mathcal{D}$ make a statistical sense?

Confidence Region on F_ξ

Does the construction of $\mathcal{D}$ make a statistical sense?

Theorem

Consider

$$D(\gamma_1, \gamma_2) = \left\{ F_\xi \;\middle|\; \begin{array}{l} P(\xi \in \Xi) = 1 \\ (\mathbb{E}[\xi] - \mu_0)^T \Sigma_0^{-1} (\mathbb{E}[\xi] - \mu_0) \leq \gamma_1 \\ \mathbb{E}[(\xi - \mu_0)(\xi - \mu_0)^T] \preceq \gamma_2 \Sigma_0 \end{array} \right\}$$

where μ_0 and Σ_0 are point estimates from the empirical data (of size m) and Ξ lies in a ball of radius R such that $||\xi||_2 \leq R$ a.s..

Then for $\gamma_1 = O(\frac{R^2}{m} \log(4/\delta))$ and $\gamma_2 = O(\frac{R^2}{\sqrt{m}} \sqrt{\log(4/\delta)})$,

$$P(F_\xi \in D(\gamma_1, \gamma_2)) \geq 1 - \delta$$

DRO with Likelihood Bounds

Define the distribution set by the constraint on the likelihood ratio. With observed Data: $\xi_1, \xi_2, ...\xi_N$, we define

$$\mathcal{D}_N = \left\{ F_\xi \middle| \begin{array}{l} P(\xi \in \Xi) = 1 \\ L(\xi, F_\xi) \geq \gamma \end{array} \right\}$$

where γ adjusts the level of robustness and N represents the sample size.

DRO with Likelihood Bounds

Define the distribution set by the constraint on the likelihood ratio. With observed Data: $\xi_1, \xi_2, \ldots \xi_N$, we define

$$\mathcal{D}_N = \left\{ F_\xi \middle| \begin{array}{l} P(\xi \in \Xi) = 1 \\ L(\xi, F_\xi) \geq \gamma \end{array} \right\}$$

where γ adjusts the level of robustness and N represents the sample size.

For example, assume the support of the uncertainty is finite

$$\xi_1, \xi_2, \ldots \xi_n$$

and we observed m_i samples on ξ_i. Then, F_ξ has a finite discrete distribution $p_1, \ldots, p_n$ and

$$L(\xi, F_\xi) = \sum_{i=1}^{n} m_i \log p_i.$$

Theory on Likelihood Bounds

The model is a convex optimization problem, and connects to many statistical theories:

- Statistical Divergence theory: provide a bound on KL divergence
- Bayesian Statistics with the threshold γ estimated by samples: confidence level on the true distribution
- Non-parametric Empirical Likelihood theory: inference based on empirical likelihood by Owen
- Asymptotic Theory of the likelihood region
- Possible extensions to deal with Continuous Case

Wang, Glynn and Y [2012,2016]

DRO using Wasserstein Ambiguity Set

By the Kantorovich-Rubinstein theorem, the Wasserstein distance between two distributions can be expressed as the minimum cost of moving one to the other, which is a semi-infinite transportation LP.

DRO using Wasserstein Ambiguity Set

By the Kantorovich-Rubinstein theorem, the Wasserstein distance between two distributions can be expressed as the minimum cost of moving one to the other, which is a semi-infinite transportation LP.

Theorem

When using the Wasserstein ambiguity set

$$\mathcal{D}_N := \{F_\xi \mid P(\xi \in \Xi) = 1 \ \& \ d(F_\xi, \hat{F}_N) \leq \varepsilon_N\},$$

where $d(F_1, F_2)$ is the Wasserstein distance function and N is the sample size, the DRO model satisfies the following properties:

- *Finite sample guarantee : the correctness probability $\bar{P}^N$ is high*
- *Asymptotic guarantee : $\bar{P}^\infty(\lim_{N\to\infty} \hat{x}_{\varepsilon_N} = x^*) = 1$*
- *Tractability : DRO is in the same complexity class as SAA*

Mohajerin Esfahani & Kuhn [15, 17], Blanchet, Kang, Murthy [16], Duchi and Namkoong [16]

DRO for Logistic Regression

- Let $\{(\hat{\xi}_i, \hat{\lambda}_i)\}_{i=1}^N$ be a feature-label training set i.i.d. from P, and consider applying logistic regression :

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \ell(x, \hat{\xi}_i, \hat{\lambda}_i) \text{ where } \ell(x, \xi, \lambda) = \ln(1 + \exp(-\lambda x^T \xi))$$

DRO for Logistic Regression

- Let $\{(\hat{\xi}_i, \hat{\lambda}_i)\}_{i=1}^N$ be a feature-label training set i.i.d. from P, and consider applying logistic regression :

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \ell(x, \hat{\xi}_i, \hat{\lambda}_i) \text{ where } \ell(x, \xi, \lambda) = \ln(1 + \exp(-\lambda x^T \xi))$$

- DRO suggests solving

$$\min_x \sup_{F \in \mathcal{D}_N} \mathbb{E}_F[\ell(x, \xi_i, \lambda_i)]$$

with the Wasserstein ambiguity set.

DRO for Logistic Regression

- Let $\{(\hat{\xi}_i, \hat{\lambda}_i)\}_{i=1}^N$ be a feature-label training set i.i.d. from P, and consider applying logistic regression :

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \ell(x, \hat{\xi}_i, \hat{\lambda}_i) \text{ where } \ell(x, \xi, \lambda) = \ln(1 + \exp(-\lambda x^T \xi))$$

- DRO suggests solving

$$\min_x \sup_{F \in \mathcal{D}_N} \mathbb{E}_F[\ell(x, \xi_i, \lambda_i)]$$

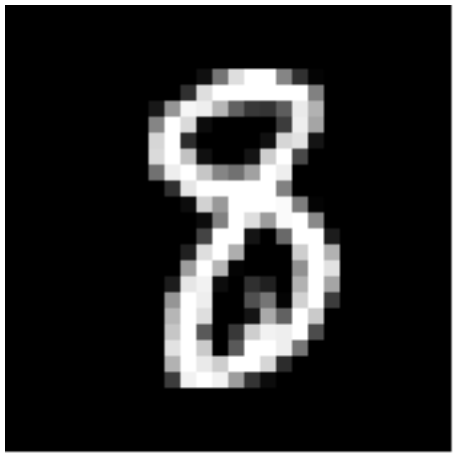
with the Wasserstein ambiguity set.

- When labels are considered to be error free, DRO with $\mathcal{D}_N$ reduces to regularized logistic regression:

$$\min_x \frac{1}{N} \sum_{i=1}^{N} \ell(x, \hat{\xi}_i, \hat{\lambda}_i) + \varepsilon \|x\|_*$$

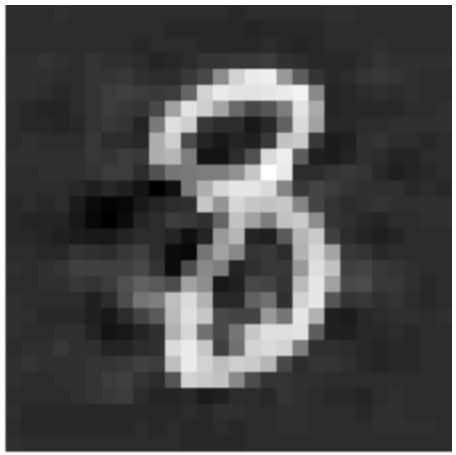
Shafieezadeh Abadeh, Mohajerin Esfahani, & Kuhn, NIPS, [2015]

Result of the DRO Learning

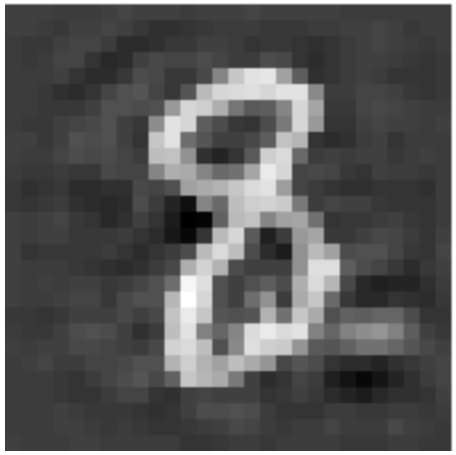


Original

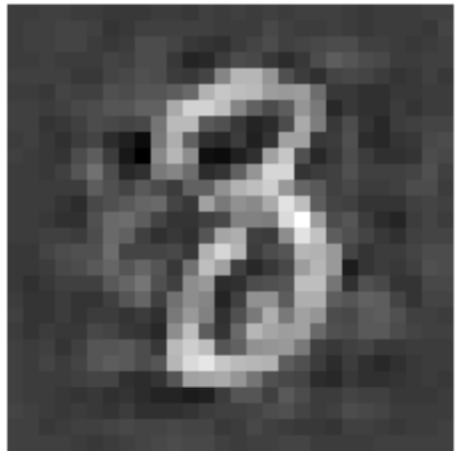
ERM



FGM



IFGM

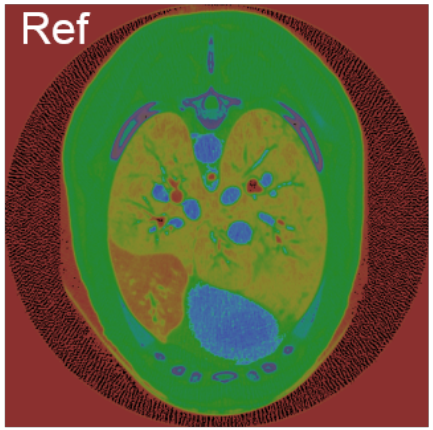


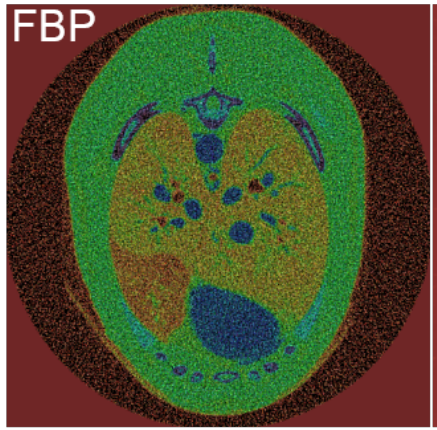
PGM

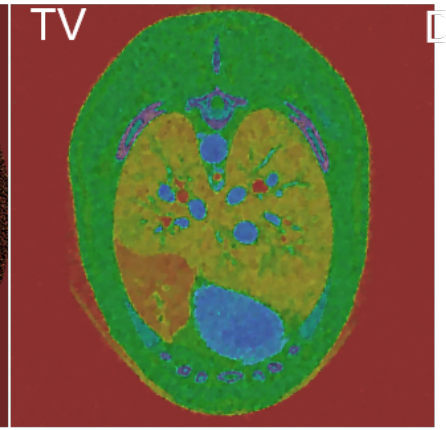
WRM

Sinha, Namkoong and Duchi [2017]

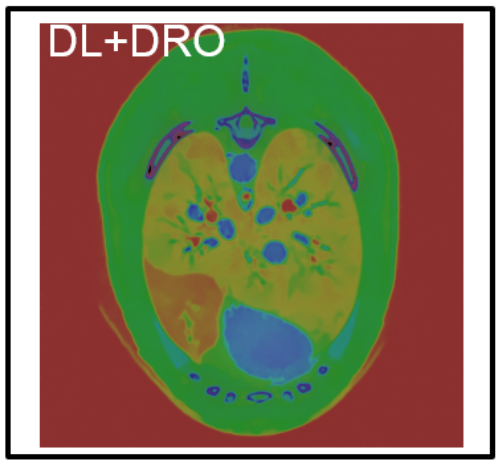
Medical Decision: CT Imaging of Sheep Thorax







DL



Substantial noise reduction

Ref: Filtered Back Projection reconstructions of noise-free data

FBP: FBP reconstructions of noisy data

TV: TV-based reconstruction

DL: Dictionary Learning-based reconstruction

DL+DRO: DL+DRO to encourage low-rankness and robustness

Liu et al. [2017]

Result of the DRO Medical Decision Making

Color CT Imaging of Sheep Thorax

Method	SSIM
FEB	0.37
TV	0.79
DL	0.85
DL+DRO	0.93

The robust method

Best structural similarity (SSIM)

Liu et al. [2017]

Summary of DRO under Moment, Likelihood or Wasserstein Ambiguity Set

- The DRO models yield a solution with a guaranteed confidence level to the possible distributions. Specifically, the confidence region of the distributions can be constructed upon the historical data and sample distributions.

Summary of DRO under Moment, Likelihood or Wasserstein Ambiguity Set

- The DRO models yield a solution with a guaranteed confidence level to the possible distributions. Specifically, the confidence region of the distributions can be constructed upon the historical data and sample distributions.
- The DRO models are tractable, and sometimes maintain the same computational complexity as the stochastic optimization models with known distribution.

Summary of DRO under Moment, Likelihood or Wasserstein Ambiguity Set

- The DRO models yield a solution with a guaranteed confidence level to the possible distributions. Specifically, the confidence region of the distributions can be constructed upon the historical data and sample distributions.
- The DRO models are tractable, and sometimes maintain the same computational complexity as the stochastic optimization models with known distribution.
- This approach can be applied to a wide range of problems, including inventory problems (e.g., newsvendor problem), portfolio selection problems, image reconstruction, machine learning, etc., with reported superior numerical results