AltGDmin for Policy Evaluation in Offline Reinforcement Learning

Komal Krishna Mogilipalepu

Supervised by Prof. Namrata Vaswani

Outline

- Offline Reinforcement Learning
- Policy Evaluation for Offline Reinforcement Learning
- Low-Rank Structure on Markov Decision Process
- Offline Dataset Generation
- Low-Rank Matrix Completion using Alternating Minimization
- Policy Evaluation Algorithm with AltGDmin
- Simulation Results of AltGDmin and Altmin
- Simulation Results of Policy Evaluation with Alternating Minimization
- Conclusion, Future Work, and References
- Acknowledgements

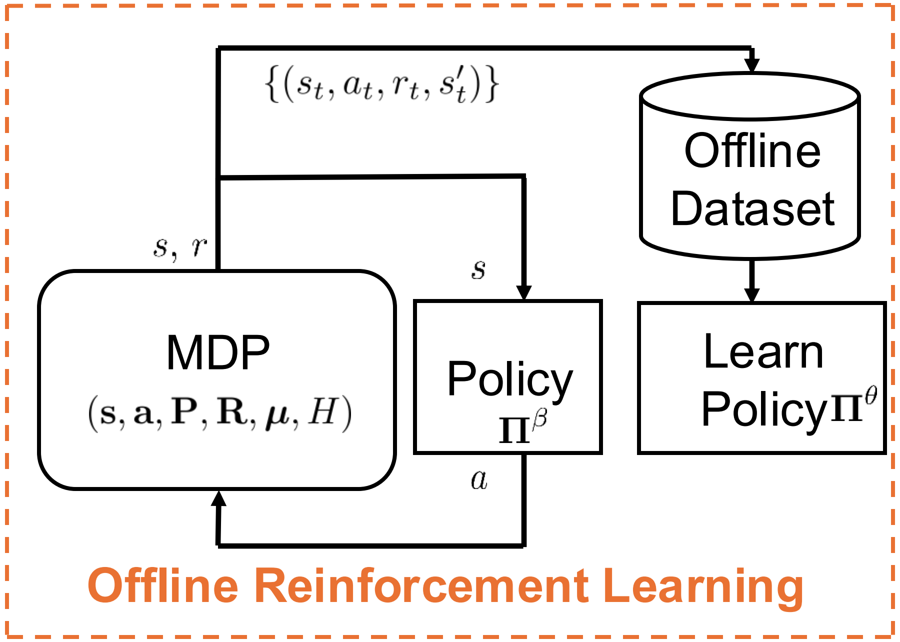
Offline Reinforcement Learning

- Finite discrete Markov decision process (MDP) $\mathcal{M} = (\mathbf{s}, \mathbf{a}, \mathbf{P}, \mathbf{R}, \boldsymbol{\mu}, H)$
  - state space $\mathbf{s}$ such that state $s \in \mathbf{s}$,
  - action space $\mathbf{a}$ such that action $a \in \mathbf{a}$,
  - probability transition kernel $\mathbf{P} = \{\mathbf{P}_t : \mathbf{s} \times \mathbf{a} \times \mathbf{s} \to [0,1]\}_{t \in [H]}$,
  - bounded reward function $\mathbf{R} = \{\mathbf{R}_t : \mathbf{s} \times \mathbf{a} \to [0,1]\}_{t \in [H]}$,
  - initial state distribution $\boldsymbol{\mu}$,
  - horizon H.
- Policy $\mathbf{\Pi} : \mathbf{s} \times \mathbf{a} \to [0,1]$.
- Total expected reward with respect to a policy $\mathbf{\Pi}$ is

$$J^{\mathbf{\Pi}} = \mathbb{E}_{\mathbf{\Pi}}[\textstyle\sum_{t=1}^{H} \mathbf{R}(s_t, a_t) | s_1 \sim \boldsymbol{\mu}].$$

- State-action value in the t-th step according to policy $\mathbf{\Pi}$ is

$$\mathbf{Q}_t^{\mathbf{\Pi}}(s,a) = \mathbb{E}_{\mathbf{\Pi}}[\textstyle\sum_{i=t}^{H} \mathbf{R}_i(s_i, a_i) | s_t = s, a_t = a].$$



$\mathbf{\Pi}^{\beta}$ - unknown behavior policy

$\mathbf{\Pi}^{\theta}$ - target policy

Learn a target policy that generalize the pattern observed in the offline dataset.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Policy Evaluation for Offline Reinforcement Learning

Offline RL → Policy Evaluation → Policy Deployment

- Distributional shift:
  - Evaluating the target policy out of offline data distribution.

- Concentrability coefficient:
  - The coverage of the dataset is measured by the concentrability coefficient $\mathcal{C}^{\mathbf{\Pi}} = \max_{s,a} \frac{\mathbf{D}^{\mathbf{\Pi}}(s,a)}{\widehat{\mathbf{D}}(s,a)}$

    where, $\mathbf{D}^{\mathbf{\Pi}}(s,a)$ - state-action occupancy measure by a policy $\mathbf{\Pi}$ and

    $\widehat{\mathbf{D}}(s,a)$ - empirical measure of a state-action pair observed in the offline data.

  - Full coverage of the dataset $\mathcal{C}^{\mathbf{\Pi}} < \infty \ \forall \ \mathbf{\Pi}$.

  - Partial coverage of the dataset $\mathcal{C}^{\mathbf{\Pi}^\star} < \infty$ for an optimal policy $\mathbf{\Pi}^\star$.

  - Infinite concentrability coefficient $\mathcal{C}^{\mathbf{\Pi}} = \infty \ \forall \mathbf{\Pi}$.

How to evaluate the target policy with infinite concentrability coefficient?

Low-Rank Structure on Markov Decision Process

- **Assumption 1:**

  The reward matrix $\mathbf{R}_t$ has rank at most $d/2\ \forall t$.

- **Assumption 2:**

  The state transition kernel $\mathbf{P}_t$ has following representation

  $$\mathbf{P}_t(s,a,s') = \mathbf{u}_t(s',s)^{\mathrm{T}}\mathbf{w}_t(a)$$
  $$\text{or } \mathbf{P}_t(s,a,s') = \mathbf{u}_t(s)^{\mathrm{T}}\mathbf{w}_t(s',a)\ \forall\, t, s', s, a,$$

  where, $\mathbf{u}_t(\cdot)$ and $\mathbf{w}_t(\cdot)$ are unknown functions, which maps the input to a $d/2$ length vector.

Algorithm 1: Low-Rank Markov decision process

```
LRMDP(S, A, H)
s = [S], a = [A]
μ = [S]/S
reward R = (uniform(0,1)_{s×1} 1^T_{1×a})
Initialize state transition kernel P = [ ]
for t ← 1 to H do
  | Append 1_{s×a×s} * 1/S to P
end
return (s, a, P, R, μ)
```

$[S] = [1,2,3,\ldots,S]$ and $\times$ denotes cartesian product.

- **Assumption:** The state-action value matrix $\mathbf{Q}^{\mathbf{\Pi}}$ is at most rank d for any policy $\mathbf{\Pi}$,

  - alleviates finding the unknown functions $\mathbf{u}_t(\cdot)$ and $\mathbf{w}_t(\cdot)$.

- **Objective:**

  Estimate total expected reward $J^{\mathbf{\Pi}^{\theta}}$ for a target policy $\mathbf{\Pi}^{\theta}$ from the offline dataset that is generated by an unknown behavior policy $\mathbf{\Pi}^{\beta}$.

Offline Dataset Generation

- Uniform transition model

  $\boldsymbol{\mu}_t(s) = 1/n \ \forall \ s \in \mathbf{s}$ here, $n = S = A$,

  and $\mathbf{P}_t(\cdot|s,a) = 1/n \ \forall s \in \mathbf{s},\ a \in \mathbf{a}$.

- Uniform policy

  - for every t and $\forall \ s \in \mathbf{s}$, target policy $\mathbf{\Pi}_t^{\theta}(\cdot|s)$ sample an action a uniformly from the action subset $\mathbf{a}_t^{\theta}$ of size m, which itself sampled uniformly from action space $\mathbf{a}$ i.e.

  $$\mathbf{\Pi}_t^{\theta} = \{1,\, 0\}_{S\times A} \sim \ \text{Bernoulli}\ \left(\tfrac{m}{n}\right) * \tfrac{1}{m},$$

  - generate behavior policy from the same target policy model independently i.e.

  $$\mathbf{\Pi}_t^{\beta} = \{1,\, 0\}_{S\times A} \sim \ \text{Bernoulli}\ \left(\tfrac{m}{n}\right) * \tfrac{1}{m}.$$

- The uniform policy model resulting to an infinite concentrability coefficient

  $$\mathcal{C}^{\mathbf{\Pi}} = \max_{s,a} \frac{\mathbf{D}^{\mathbf{\Pi}_t^{\theta}}(s,a)}{\mathbf{D}^{\mathbf{\Pi}_t^{\beta}}(s,a)} = \infty.$$

Algorithm 2: dataset generation

Input: number of states S and actions A, number of trajectories K, horizon H, action subset size m.

Output: dataset $\mathcal{D}$

Initialize dataset $\mathcal{D} = [\,]$
$(\mathbf{s}, \mathbf{a}, \mathbf{P}, \mathbf{R}, \boldsymbol{\mu}) = \mathtt{MDP}(S, A, H)$
for $k \leftarrow 1$ ***to*** K **do**
  Initialize trajectory $\tau = [\,]$
  $s_0 \sim \boldsymbol{\mu}(\mathbf{s})$
  for $t \leftarrow 1$ ***to*** H **do**
    $a \sim \text{uniform}(\mathbf{a}_t^{\theta}),\ \text{where}\ |\mathbf{a}_t^{\theta}| = m,\ \text{and}\ \mathbf{a}_t^{\theta} \subseteq \mathbf{a}$
    $r = \mathbf{R}_t(s_0, a)$
    $s_1 \sim \mathbf{P}_t(\mathbf{s}|s_0, a)$
    Append (s_0, a, r, s_1) to τ
    $s_0 \leftarrow s_1$
  end
  Append trajectory τ to dataset $\mathcal{D}$
end

- Offline data $\mathcal{D} = \{(s_t^k, a_t^k, r_t^k, s_t'^k)\} \ \forall \ t \in H, k \in K.$

Low-Rank Matrix Completion using Alternating Minimization

- Compose a low-rank matrix $\mathbf{X} = \mathbf{U}\mathbf{B}$ where, $\mathbf{U} \in \mathbb{R}^{n \times r} \sim \mathcal{N}(\mathbf{0}_r, \mathbf{I}_{r \times r})$ and $\mathbf{B} \in \mathbb{R}^{r \times q} \sim \mathcal{N}(\mathbf{0}_r, \mathbf{I}_{r \times r})$.

- Observations $\mathbf{Y} := \mathbf{M} \circ (\mathbf{X} + \mathbf{N})$,

  where, $\mathbf{M} \in \{0,1\}^{n \times q}$ is a Bernoulli matrix with probability p and $\mathbf{N} \in \mathbb{R}^{n \times q} \sim \sigma\mathcal{N}(0,1)$ with standard deviation σ.

- Task: retrieve $\mathbf{X}$ from $\mathbf{Y}$.

- Optimization problem: $\min_{\mathbf{B}, \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}} \|\mathbf{Y} - \mathbf{M} \circ \mathbf{U}\mathbf{B}\|_F^2$.

- Metric: subspace distance measure between $\mathbf{U}^{(I)}$ and $\mathbf{U}$,

  $$\mathrm{SD}(\mathbf{U}^{(I)}, \mathbf{U}) = \|(\mathbf{I} - \mathbf{U}^{(I)}\mathbf{U}^{(I)^{\mathrm{T}}})\mathbf{U}\|_F.$$

- Altmin implement the least squares instead of the gradient descent for updating $\mathbf{U}$.

- We consider $c = 0.1$ for all our experiments.

Algorithm 3: AltGDmin

Input: observations $\mathbf{Y}$, mask $\mathbf{M}$, rank r, stepsize η, and iterations I.
Output: $\mathbf{U}^{(I)}$, $\mathbf{B}^{(I)}$

1. Initialize $\mathbf{U}^{(0)}$ by first r left singular vectors of $\mathbf{Y}$
2. **for** $i \leftarrow 1$ *to* I **do**
3. $\quad \mathbf{B}_{:,k}^{(i)} = (\mathbf{U}_{\mathbf{m}_k,:}^{(i-1)^{\mathrm{T}}} \mathbf{U}_{\mathbf{m}_k,:}^{(i-1)})^{-1} \mathbf{U}_{\mathbf{m}_k,:}^{(i-1)^{\mathrm{T}}} \mathbf{Y}_{\mathbf{m}_k,k} \ \forall\ k \in [q]$
4. $\quad \mathbf{U} \leftarrow \mathbf{U}^{(i-1)} - \eta * 2(\mathbf{M} \circ (\mathbf{U}^{(i-1)}\mathbf{B}^{(i)}) - \mathbf{Y})\mathbf{B}^{(i)^{\mathrm{T}}}$
5. $\quad \mathbf{U}^{(i)} \leftarrow \mathrm{QR}(\mathbf{U})$
6. **end**

where, step size $\eta = \frac{cp}{\|\mathbf{Y}\|_2^2}$ with constant c.

Abbasi, A. A., Moothedath, S., & Vaswani, N. (2023, September). Fast Federated Low Rank Matrix Completion. In *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 1-6). IEEE.
Jain, P., Netrapalli, P., & Sanghavi, S. (2013, June). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing* (pp. 665-674).

Policy Evaluation with AltGDmin

- In the t-th horizon,
  find $\mathbf{N}_t(s,a) = \sum_{k\in[K]} \mathbb{1}_{\{(s_t^k, a_t^k)=(s,a)\}}\ \forall (s,a) \in \mathbf{s}\times\mathbf{a}$,
  estimate $\widehat{\mathbf{P}}_t(s'|s,a) = \frac{1}{\mathbf{N}_t(s,a)} \sum_{k\in K} \mathbb{1}_{\{(s_t^k, a_t^k, s_{t+1}^k)=(s,a,s')\}}$,
  get $\mathbf{R}_t(s,a) = r(s,a)\ \forall (s,a) \in supp(\mathbf{N}_t)$ from dataset $\mathcal{D}$.
- Find, $\mathbf{Z}_t(s,a)$ using $\widehat{\mathbf{P}}_t$ and $\mathbf{R}_t\ \ \forall (s,a) \in supp(\mathbf{N}_t)$ through the Bellman update equation, and $\forall\ (s,a) \in \mathbf{s}\times\mathbf{a} \setminus supp(\mathbf{N}_t)$ estimate with AltGDmin.
- Iterate backward from horizon H to 1.
- Finally, find estimated total expected reward $\widehat{J}$.
- True total expected reward

$$J^{\mathbf{\Pi}^\theta} \leftarrow \sum_{s,a} \mathbf{D}_1^{\mathbf{\Pi}^\theta}(s,a)\mathbf{Q}_1^{\mathbf{\Pi}^\theta}(s,a).$$

Algorithm 4: policy evaluation with AltGDmin

Input: dataset $\mathcal{D}$, target policy $\mathbf{\Pi}^\theta$, initial state distribution $\boldsymbol{\mu}_1$, horizon H, number of states S and actions A, `AltGDmin`, rank d, stepsize η, iterations I, and $\mathbf{D}_1^{\mathbf{\Pi}^\theta}$.

Output: $\widehat{J}$

```
Initialize Q̂^{Π^θ}_{H+1} ← 0_{S×A}
Number of trajectories K = len(D)
for t ← H to 1 do
    Initialize N_t ← 0_{S×A}, P̂_t ← 0_{S×A×S}, and R_t ← 0_{S×A}
    for k ← 1 to K do
        (s, a, r, s')   = D(k, t)
        N_t(s, a)      += 1
        P̂_t(s, a, s') += 1
        R_t(s, a)       = r
    end
    P̂_t(s'|s, a) ← P̂_t(s, a, s')/N_t(s, a) ∀ (s, a) ∈ supp(N_t)
    Z_t(s, a)   = R_t(s, a) +
        Σ_{s',a'} P̂_t(s'|s, a) Π^θ_{t+1}(a'|s') Q̂^{Π^θ}_{t+1}(s', a') ∀ (s, a) ∈ supp(N_t)
    U, B = AltGDmin(Z_t, supp(N_t), d, η, T)
    Q̂^{Π^θ}_t ← UB
end
Ĵ ← Σ_{s,a} D^{Π^θ}_1(s, a) Q̂^{Π^θ}_1(s, a)
```

$$\implies |\widehat{J} - J^{\mathbf{\Pi}^\theta}| = |\langle \mathbf{D}_1^{\mathbf{\Pi}^\theta}, \widehat{\mathbf{Q}}_1^{\mathbf{\Pi}^\theta} - \mathbf{Q}_1^{\mathbf{\Pi}^\theta}\rangle|$$

$$= |\sum_{t=1}^{H} \langle \mathbf{D}_t^{\mathbf{\Pi}^\theta}, \widehat{\mathbf{Q}}_t^{\mathbf{\Pi}^\theta} - \mathbf{Y}_t\rangle|$$ where, $\langle \mathbf{D}, \mathbf{Q}\rangle = \sum_{s,a} \mathbf{D}(s,a)\mathbf{Q}(s,a)$ and $\mathbf{Y}_t$ is population version of $\mathbf{Z}_t$.

Xi, X., Yu, C., & Chen, Y. Matrix Estimation for Offline Evaluation in Reinforcement Learning with Low-Rank Structure. In *3rd Offline RL Workshop: Offline RL as a"Launchpad"*.

Simulation Results of Noise-free AltGDmin

➢ Observations:

- For a fixed rank, as the number of observations increases the subspace distance measure converge to 10^{-15} with fewer iterations.

- For a fixed number of observations, as the rank reduces, the subspace distance measure converge to 10^{-15} with fewer iterations.

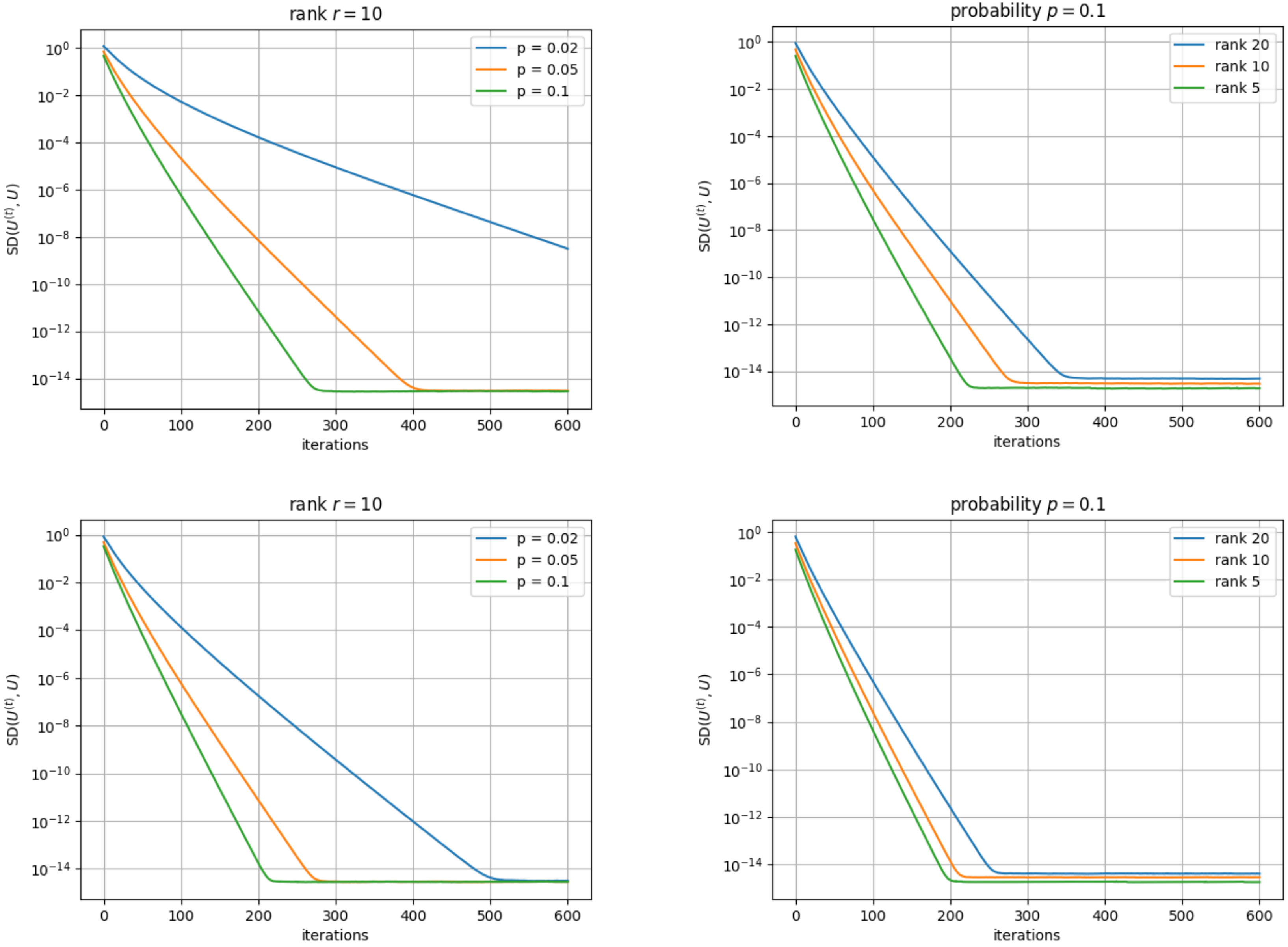


Fig.1: $(3000, 5000)$ and $(5000, 10000)$ low-rank matrix completion using AltGDmin

Simulation Results of Noise-free Altmin

- Observations:
  - For a fixed rank, as the number of observations increases the subspace distance measure converge to 10^{-15} with fewer iterations.
  - For a fixed number of observations, as the rank reduces, the subspace distance measure converge to 10^{-15} with fewer iterations.

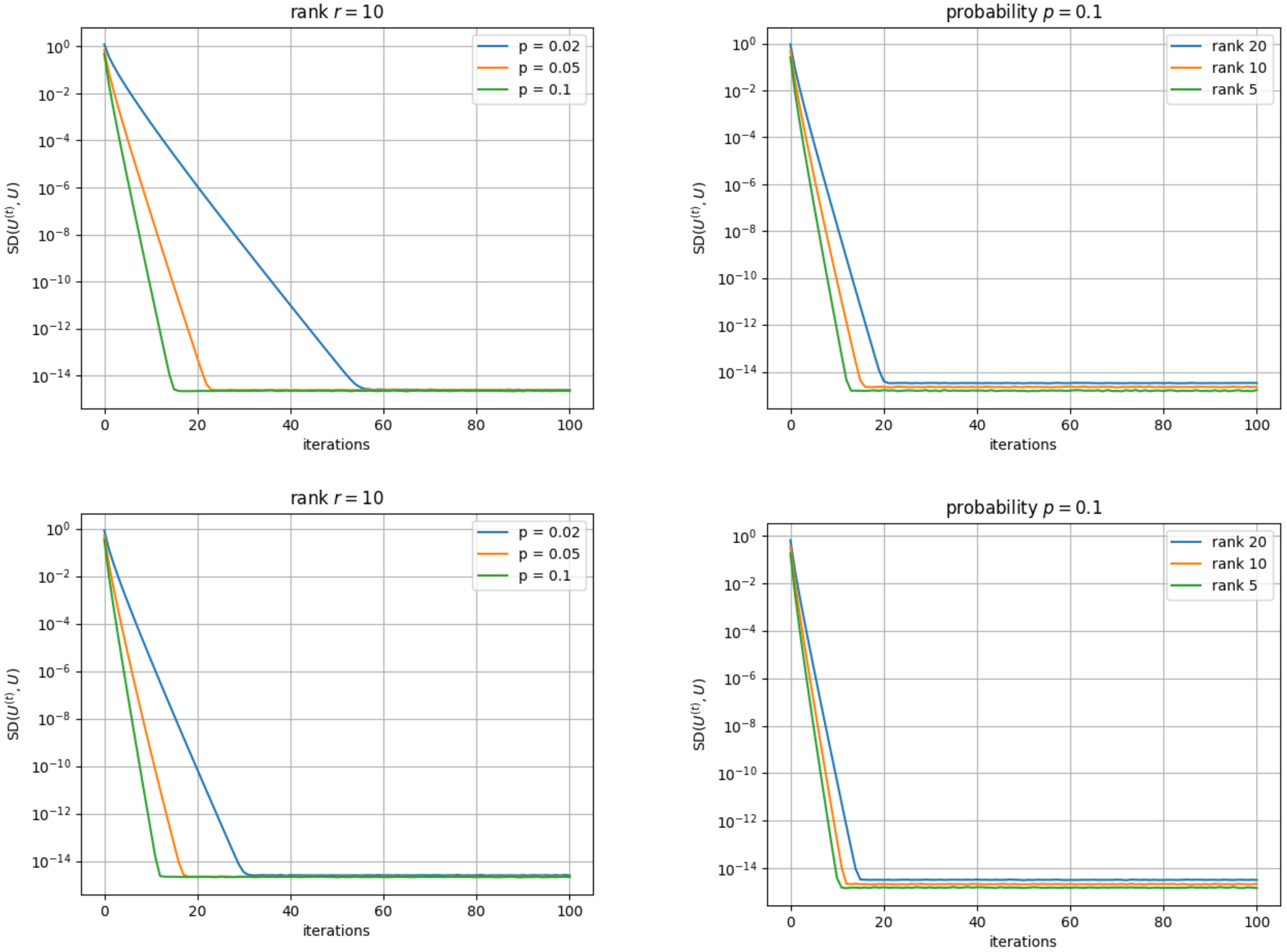


Fig.2: $(3000, 5000)$ and $(5000, 10000)$ low-rank matrix completion using AltGDmin

Simulation Results of Noisy AltGDmin

- Observations:
  - For a fixed rank, as the number of observations increases the subspace distance measure converge to 10^{-11} with fewer iterations.
  - For a fixed number of observations, as the rank increases, the subspace distance measure takes more iterations to converge to 10^{-11}.

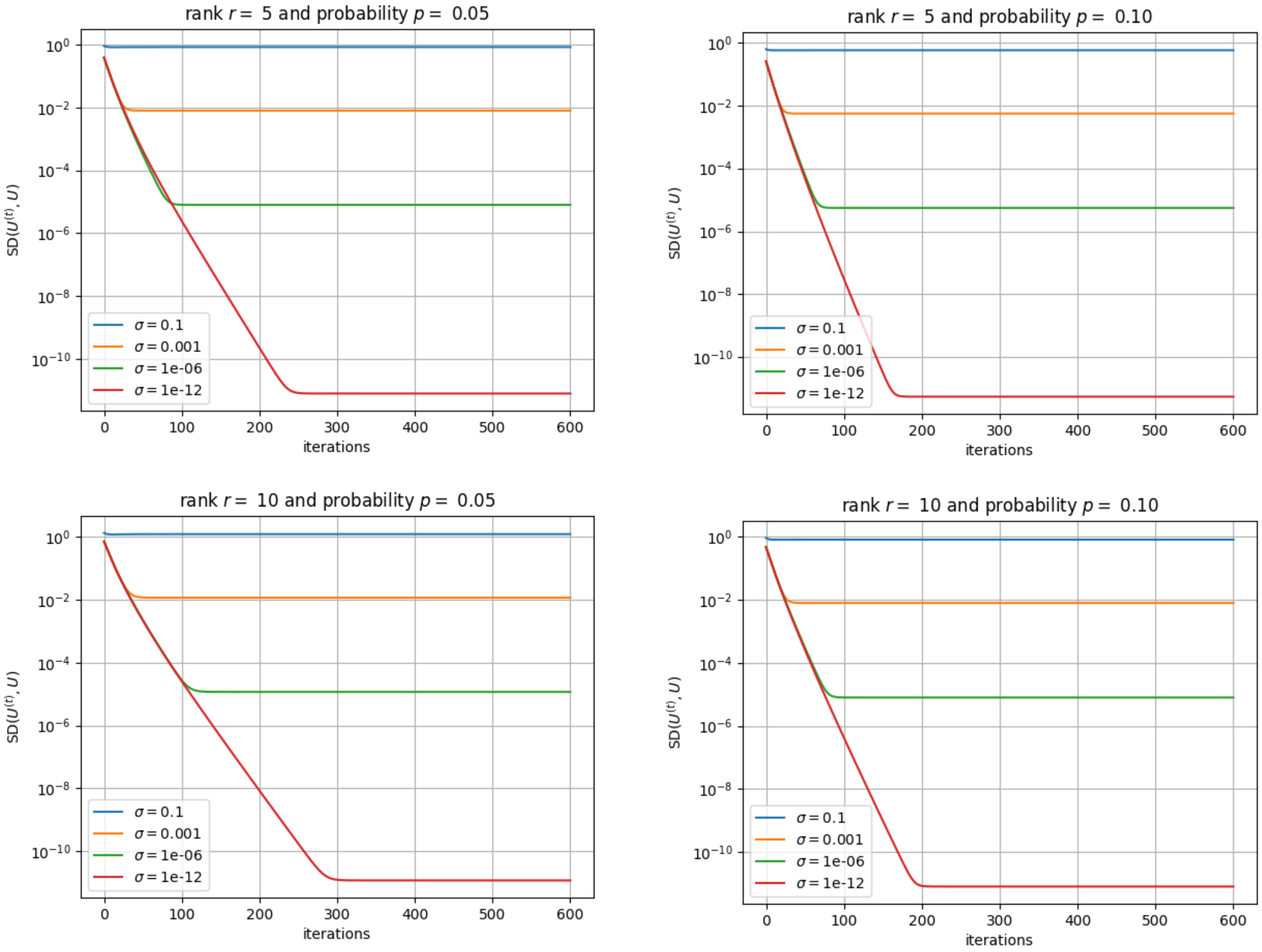


Fig.3: $(3000, 5000)$ noisy low-rank matrix completion using AltGDmin

Simulation Results of Noisy Altmin

- Observations:
  - For a fixed rank, as the number of observations increases the subspace distance measure converge to 10^{-11} with fewer iterations.
  - For a fixed number of observations, as the rank increases, the subspace distance measure takes more iterations to converge to 10^{-11}.

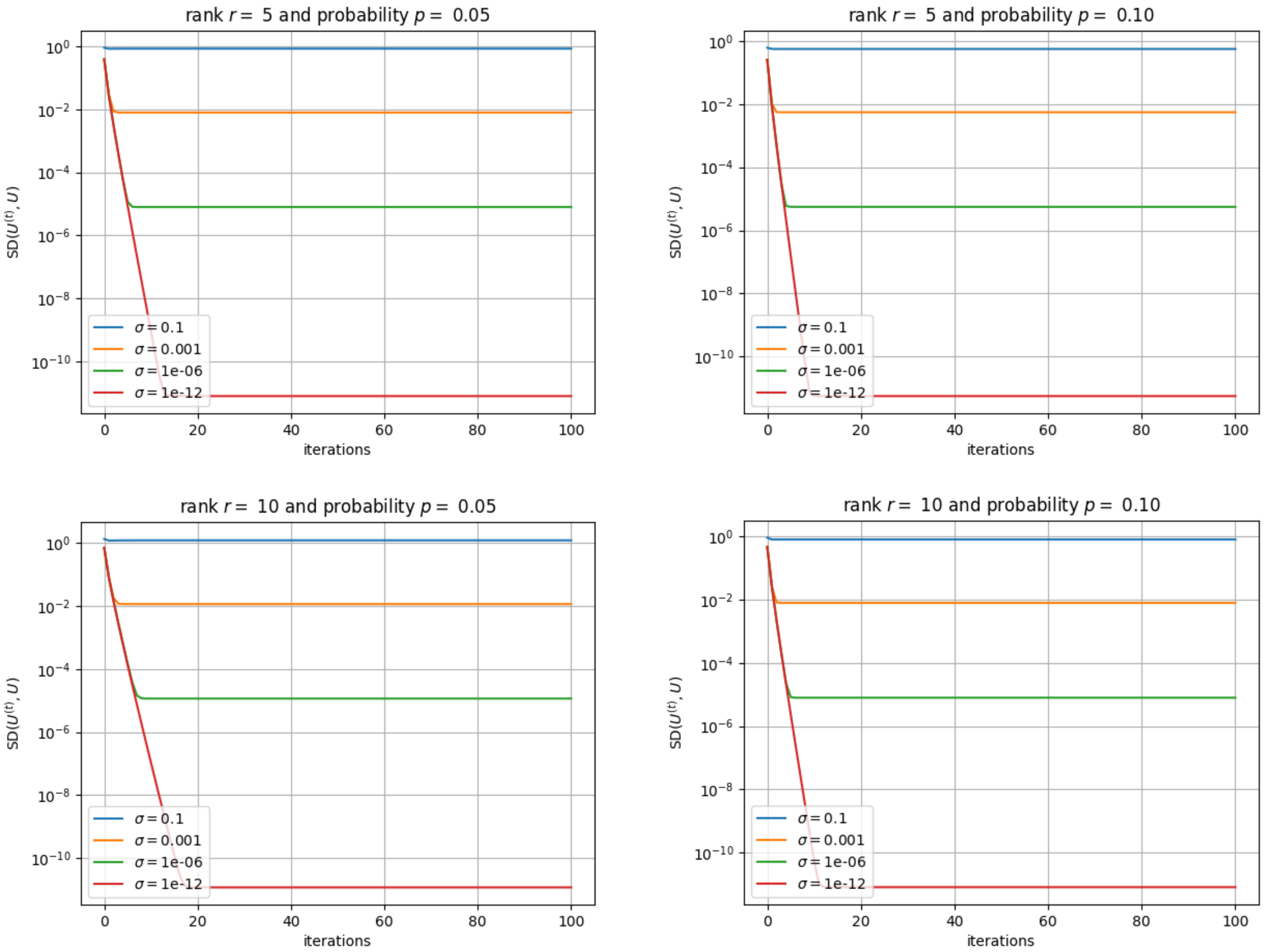


Fig.4: $(3000, 5000)$ noisy low-rank matrix completion using AltGDmin

AltGDmin and Altmin Comparison

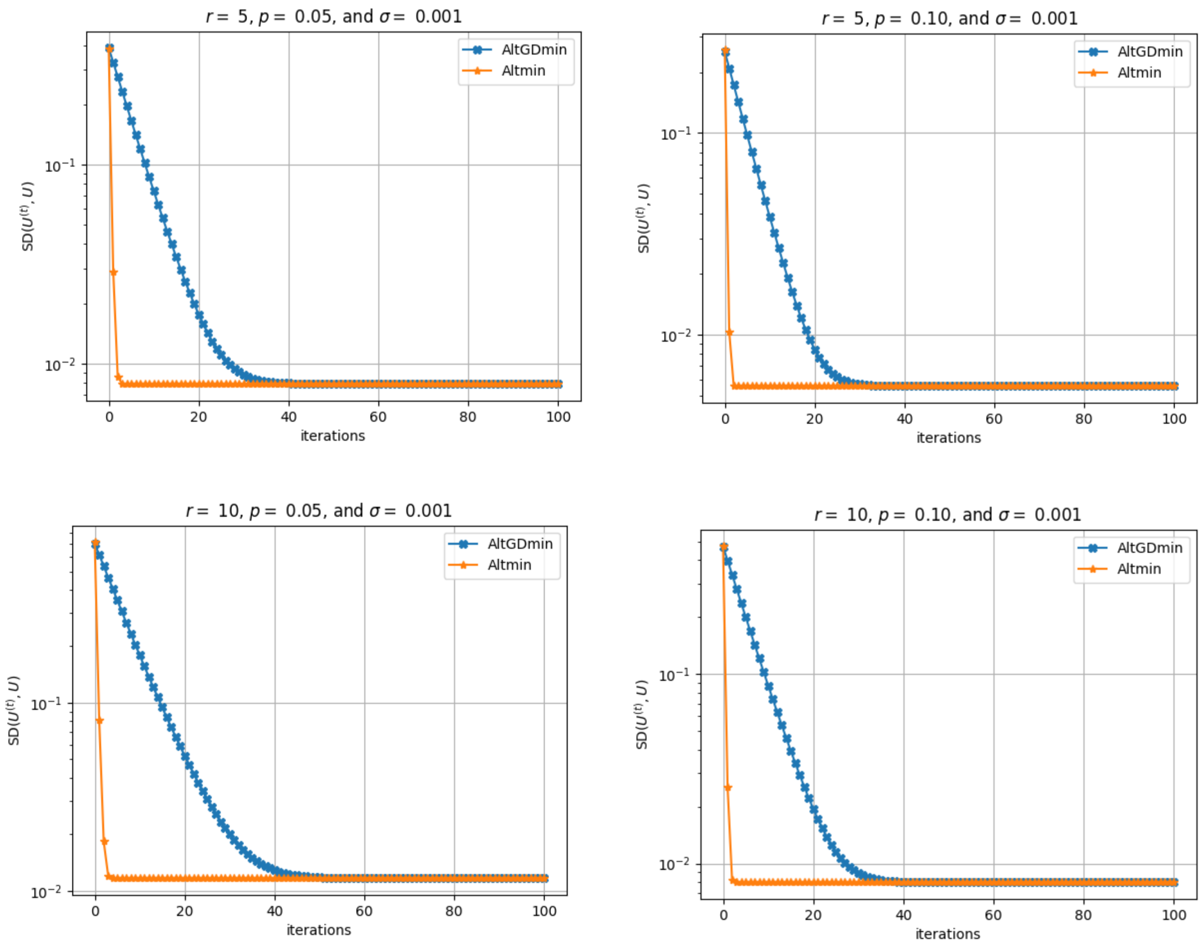


Fig.6(a): noise standard deviation $\sigma = 0.001$

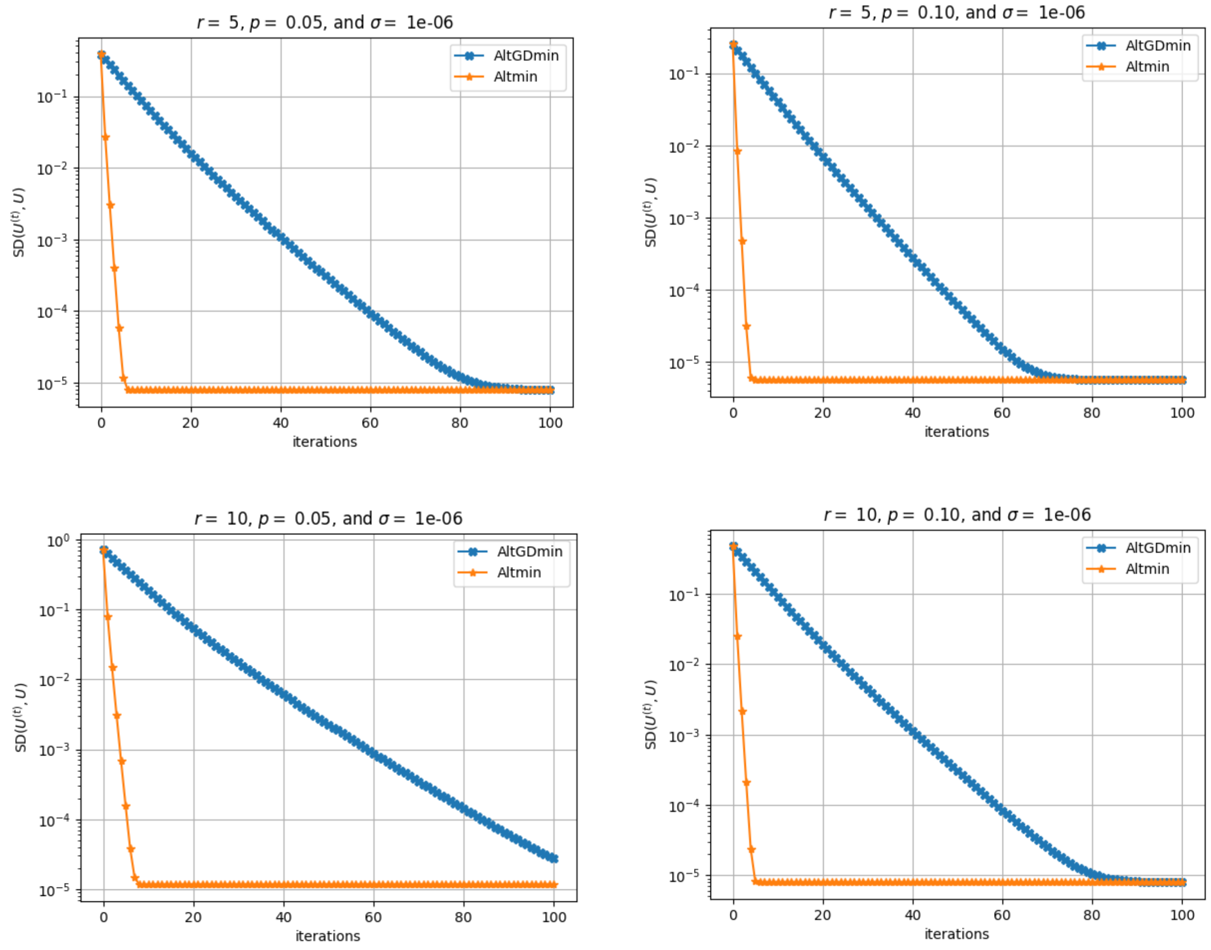


Fig.6(b): noise standard deviation $\sigma = 1e-06$

Fig.6: Comparing AltGDmin and Altmin for $(3000, 5000)$ noisy LRMC

- Altmin takes fewer iterations to converge compared to AltGDmin.

Simulation Results of Policy Evaluation with Alternating Minimization

- For action subset size $m = 300$ Algorithm 3 do not use matrix completion.

- Observations:
  - For every value of action subset size m the error increases with horizon H.
  - For action subset size $m = 300$, as the trajectories K increase from 5000 to 10000 the error reduces.
  - For action subset size $m \geq n/2$, as the trajectories K increase from 5000 to 10000 the error reduces for Altmin.
  - For action subset size $m = 250$, the error is almost increasing linearly with horizon H.

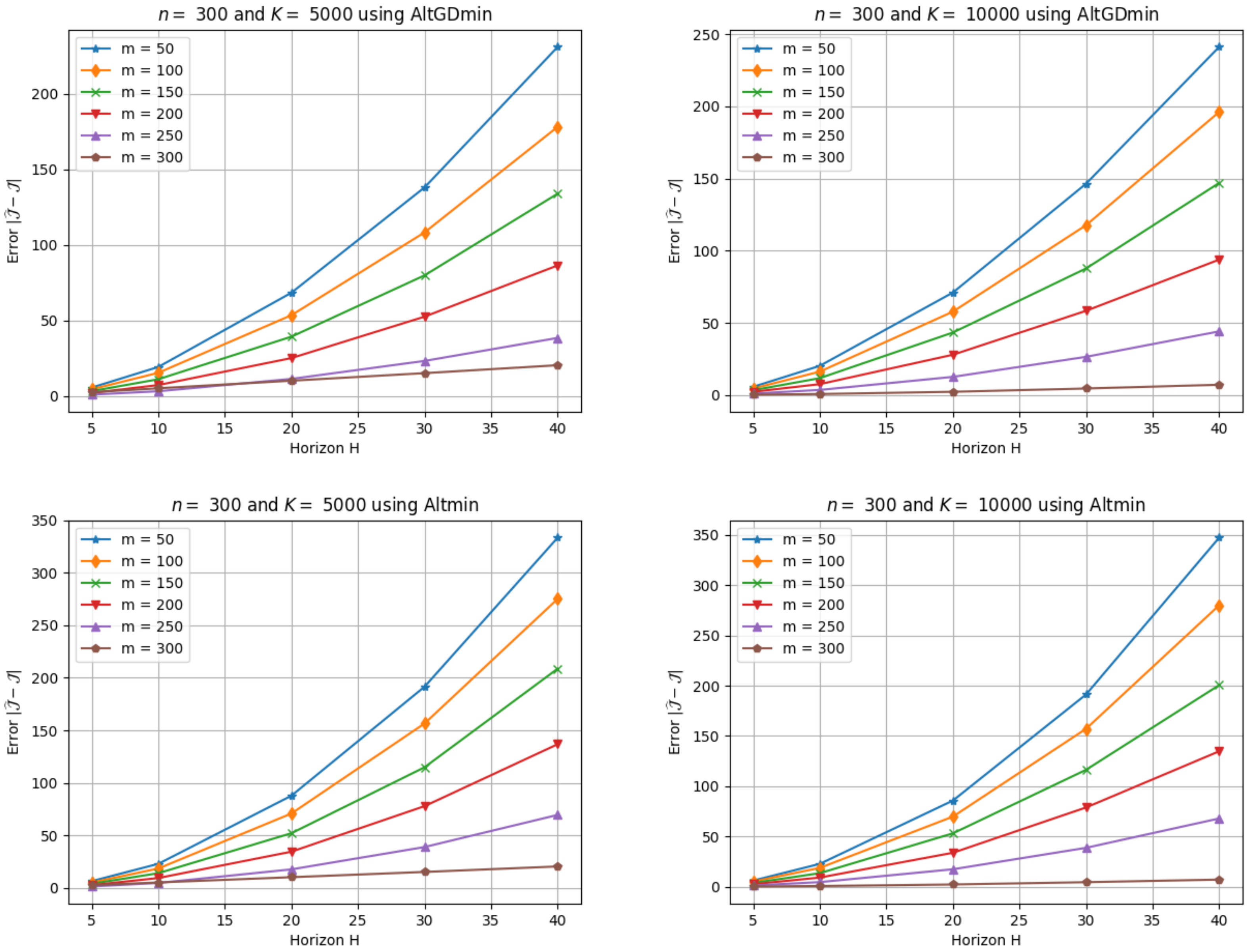


Fig.7. Policy Evaluation using AltGDmin and Altmin

Simulation Results of Policy Evaluation with Alternating Minimization

- Observations:
  - For action subset size $m = 300$, as the trajectories K increase from 5000 to 10000 the error reduces significantly.
  - For action subset size $m \geq n/2$, as the trajectories K increase from 5000 to 10000 the error reduces for Altmin.

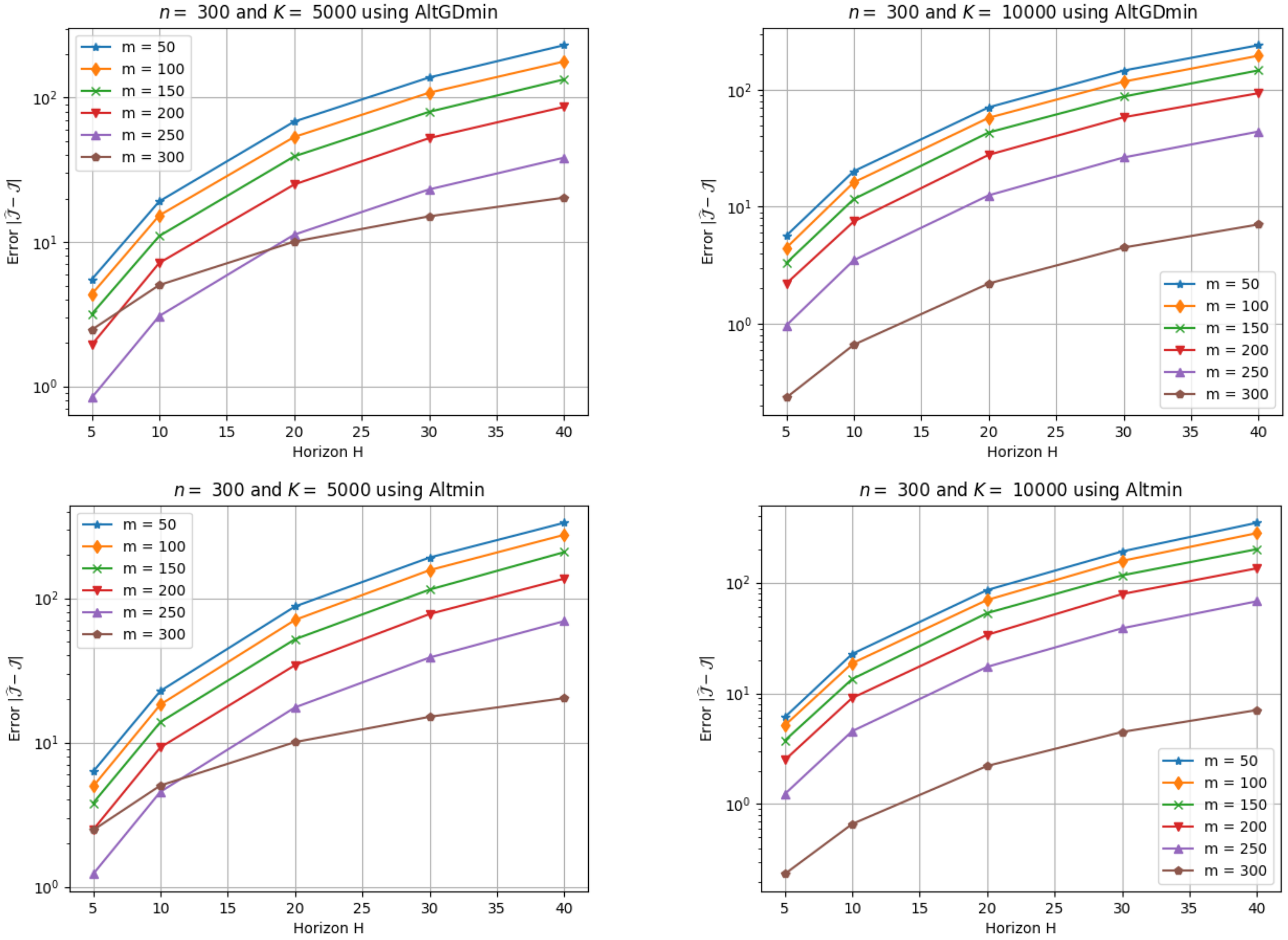


Fig.8. Policy Evaluation using AltGDmin and Altmin

Comparison of AltGDmin and Altmin for Policy Evaluation

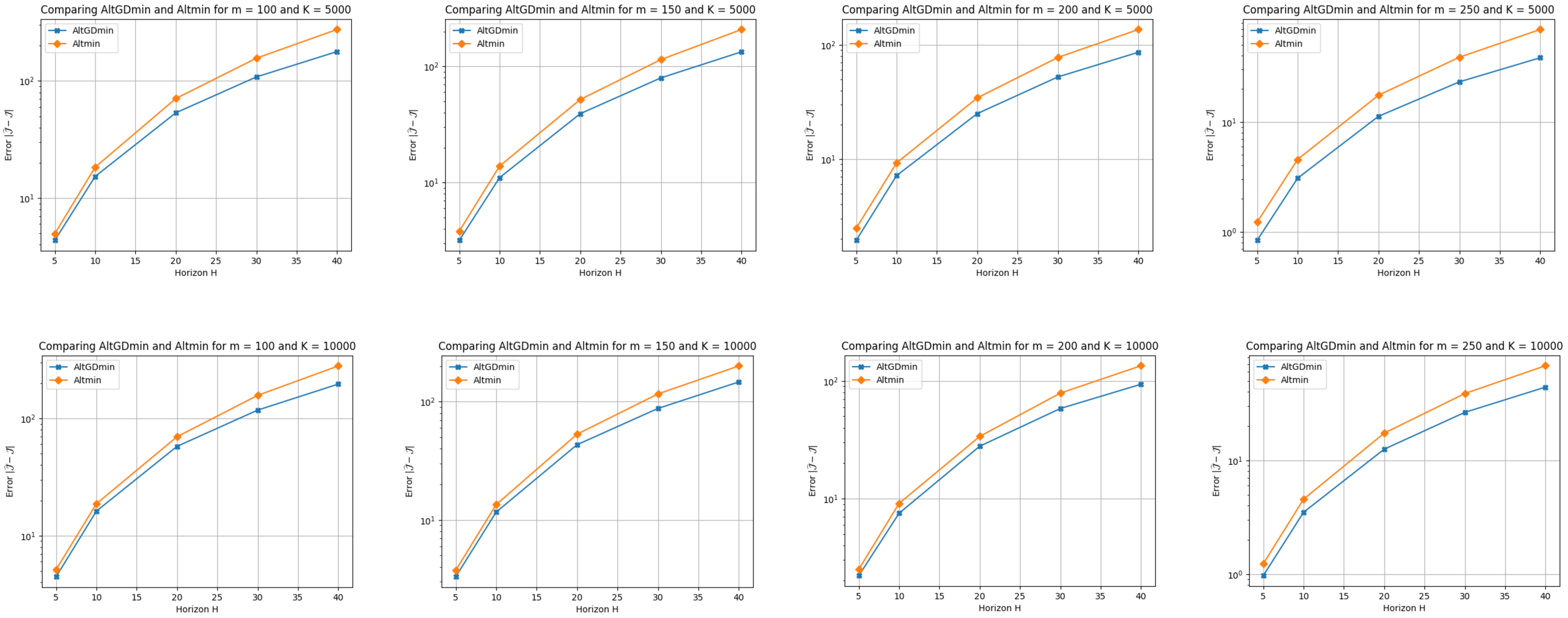


Fig.9. Comparing AltGDmin and Altmin for Policy Evaluation

- Observation: AltGDmin based policy evaluation outperform the Altmin one for every action subset size m and trajectory K.

Conclusion

- Practically feasible policy evaluation algorithm using alternating minimization for offline reinforcement learning.
- Alternating minimization based policy evaluation has a finite sample error bound.
- AltGDmin based policy evaluation outperform the Altmin one significantly.

Future Work

- We will provide theoretical guarantees showing that AltGDmin is significantly better than Altmin.

References

- Abbasi, A. A., Moothedath, S., & Vaswani, N. (2023, September). Fast Federated Low Rank Matrix Completion. In *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 1-6). IEEE.
- Jain, P., Netrapalli, P., & Sanghavi, S. (2013, June). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing* (pp. 665-674).
- Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643.*

- Liu, Y., Swaminathan, A., Agarwal, A., & Brunskill, E. (2020). Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, *33*, 1264-1274.
- Munos, R., & Szepesvári, C. (2008). Finite-Time Bounds for Fitted Value Iteration. *Journal of Machine Learning Research*, *9*(5).
- Nayer, S. S., & Vaswani, N. (2022, June). Fast low rank column-wise compressive sensing. In *2022 IEEE International Symposium on Information Theory (ISIT)* (pp. 3285-3290). IEEE.
- Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., & Russell, S. (2021). Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, *34*, 11702-11716.
- Uehara, M., & Sun, W. (2021). Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*.
- Xi, X., Yu, C., & Chen, Y. Matrix Estimation for Offline Evaluation in Reinforcement Learning with Low-Rank Structure. In *3rd Offline RL Workshop: Offline RL as a"Launchpad"*.

Acknowledgements

- I am thankful to National Science Foundation (NSF) for supporting on this project.
- I am thankful to my advisor Prof. Namrata Vaswani for her consistent support and for teaching the course High Dimensional Probability and Linear Algebra.
- I am thankful to the faculty Dr. Shana Moothedath and Dr. Sung Yell Song for being a part of my program of study committee and for the helpful discussions on Reinforcement Learning and Applied Linear Algebra topics.
- I am thankful to the faculty Prof. Aditya Ramamoorthy for clarifying all my doubts on the topics of Convex Optimization.
- I am thankful to the faculty Prof. Pavan Kumar Aduri, Dr. Hongyang Gao, and Dr. Cheng Huang for teaching the courses Algorithms for Large Datasets, Introduction to Machine Learning, and Deep Learning for Theory and Practice.
- I am thankful to the faculty Jenna Bertilson, Ben Godard, and Febriana for teaching Academic Writing and Speaking.
- I am thankful to the Electrical and Computer Engineering department for the Teaching Assistant ship.
- I am thankful to Ruoyu, Silpa, Ankit, Jiabin, Ahmed, Sifat and Vrindha for helping and supporting.
- I am thankful to Dr. Kim, Vicky, Stacy, Shahin, Jason, Lynne, Sara, Tony and ECpE department staff for the administrative support.
- I am thankful to Post. Doc Kiran, Aditya Kar, Souradeep, Jaydeep, Nabila, and Aditya for their help and support.

Thank You!