# 2022-Ray

---

## Decision-Dependent Risk Minimization in Geometrically Decaying Dynamic Environments

---


- **Date**: 2022
- **Link:** paper
- **Authors:**
  - Ray, Mitas
  - Ratliff, Lillian J
  - Drusvyatskiy, Dmitriy
  - Maryam, Fazel
- **Cites:**
- **Cited by:**
- **Keywords:** #performative-prediction #decision-dependent-risk-min
- **Collections:**
- **Status:** #completed


---

### 0 - Abstract


- Problem description:
  - Perform expected loss minimization.
  - The given data distribution is dependent on the decision-maker's action.
  - The distribution evolves dynamically in time according to a geometric decay process.

- Proposed algorithms
  - Two information settings:
    - Decision-maker has a first order gradient oracle.
    - Decision-maker has a loss function oracle.
  - Key principle:
    - The decision-maker repeatedly deploys a fixed decision repeatedly over the length of an epoch, allowing the environment to sufficiently mix before updating the decision.

- Results:
  - Iteration complexity is shown to match existing rates for first and zero order stochastic gradient methods up to logarithmic factors.

  - Algorithms are evaluated on SF Park dynamic pricing pilot study.

  - Algorithm is shown to improve target occupancy and reduce overall parking rates.


### 1 - Introduction

- Supervised machine learning algorithms are trained on past data under the assumption that the past data is representative of the future.

- This is not the case in examples where the output of the algorithm changes the environment:
  - Online labor markets
  - Predictive policing
  - On-street parking
  - Vehicle sharing markets

- *Performative prediction* (aka decision-dependent risk minimization)- solution to this problem, models the data distribution as being *decision dependent* thereby accounting for feedback induced distributional shift.

- Past work has only focused on static environments; however, most real-world environments are not static.
  - i.e. data distribution changes dynamically in time as the strategic actors adjust their behavior based on the algorithm's change.
  - e.g. if a city changes its street parking pricing, it may take time for drivers to modify their behavior in response. After some time, an equilibrium will be achieved.

- Algorithmic framework:
  - Start of the epoch:
    - Decision-maker selects a decision to deploy repeatedly for the duration of the epoch.
  - During the epoch:
    - The distribution evolves toward a fixed point distribution for the given decision.
  - At the end of the epoch:
    - Decision-maker is updated using a first-order or zeroth-order oracle.
      (Depending on the information setting - gradient or loss function access).

#### 1.1 - Contributions

- Iteration complexity guarantees:

- **Zero Order Oracle** (Algorithm 1)
  - Sample complexity is $\tilde{O}(d^2/\epsilon^2)$.
  - This matches the optimal rate.

- **First Order Oracle** (Algorithm 2)
  - Sample complexity is $\tilde{O}(1/\epsilon)$.
  - This matches the known optimal rate.

- This is achieved by bounding the error between the expected gradient at the fixed point distribution and the stochastic gradient at time $t$.

- Algorithm was shown to produce positive results on the SF Park pilot study.
  - Higher occupancy rates for parking spots.
  - Lower total parking fees.

### 1.2 - Related Work

#### 1.2.1 - Dynamic Decision-Dependent Optimization

- *Recourse* - the decision-maker is able to make a secondary decision after some information has been revealed.

- Dynamic decision-dependent optimization has been extensively studied in the case of recourse.

- This is unlike the setting considered here.

#### 1.2.2 - Reinforcement Learning

- RL is a closely related problem:
  - Agent makes decisions in a dynamic environment that responds to the agent's actions.

- Important difference:
  - Our goal
    - Find the action which optimizes the decision-dependent expected risk at the fixed point distribution.

  - RL's goal
    - Find the policy which is a state-dependent distribution over actions given an accumulated cost.

- In this sense, our setting can be viewed as a special case of general RL.

#### 1.2.3 - Performative prediction

- Naïve strategy - retrain the model after using heuristics to determine when there is sufficient distribution shift.

- None of the works consider dynamic environments.

## 2 - Preliminaries

- Metric space notation:
  - Let $\mathcal{Z}$ be a matric space with metric $d(\cdot,\cdot)$
  - Let $\mathbb{P}(\mathcal{Z})$ denote the set of Radon probability measures $\nu$ on $\mathcal{Z}$ with finite first moment.
  - *Note to self* - I need to understand this part better.

- Decision-maker seeks to solve:

$$\min_{x\in\mathcal{X}} \mathcal{L}(x)$$

  with the expected loss function

$$\mathcal{L}(x) = E_{z\sim\mathcal{D}(x)}\left[\ell(x,z)\right]$$

- Definitions:
  - $\mathcal{X}$ - the decision space in $\mathbb{R}^d$
  - $x$ - the decision made by the decision-maker.
  - $z$ - the response from the environment.
  - $\mathcal{D}(x)$ - the probability distribution from which $z$ is sampled, depends on $x$ and $t$.

- Let $x^*$ denote an optimal decision vector.

- Probability measure:
  - **Main challenge** - the response, characterized as a random variable, $z$ depends not only on the decision $x_t$, but also explicitly on the time step $t$.

  - Let $z$ be governed by the distribution $p_t$.
  - $p_t$ is generated by the process - $p_{t+1} = \mathcal{T}(p_t,\ x_t)$ where

$$\mathcal{T}(p,x) = \lambda p + (1-\lambda)\mathcal{D}(x)$$

    is a geometric decay rate.

  - Interpretation - at each time step $t$, $(1-\lambda)$ fraction of the population becomes aware of the decision $x$.
  - Alternative interpretation - the environment has a memory that is captured by the distribution, as time progresses past observations decay at a rate of $\lambda$.

- See the paper for a list of assumptions for $\ell$ and $\mathcal{D}(x)$.

## 3 - Algorithms and Sample Complexity Analysis

- Recall, the decision-maker holds fixed the decision for $n$ time steps before observing the environment.

### 3.1 - Zero Order Stochastic Gradient Method

- This is the most general information setting in which we assume the decision-maker does not have access to $\mathcal{D}(x)$.
    - In practice, this is likely to be true.

- Stochastic gradient method:
    - Let $\delta > 0$ - this 'wiggles' the decision point around during the epoch.

    - At each epoch $t$, the algorithm...
        - Samples $v_t$ from a $d$-dimensional unit sphere.
        - Queries the environment for $n_t$ iterations with $x_t + \delta v_t$.

    - At the end of the epoch...
        - The loss oracle reveals $\ell(x_t + \delta v_t, z_t)$ where

        $$z_t \sim \lambda^{n_t} p_{t-1} + (1 - \lambda^{n_t})\, \mathcal{D}(x_t + \delta v_t)$$

        Note - this is the accumulated environmental response distribution after $n_t$ repeated selections of $x_t + \delta v_t$.

        - The decision-maker performs the update rule:

        $$x_{t+1} = \text{proj}_{(1-\delta)\mathcal{X}}\,(x_t - \eta \hat{g}_t)$$

        where

        $$\hat{g}_t = \frac{d}{\delta} \ell(x_t + \delta v_t,\ z_t) v_t$$

        is the one-point gradient estimate of the expected loss at $p_t$.
    - *Note to self:* I'm not sure why the projection is taken here? Paper says its to ensure $x$ in the next iteration is in the feasible set.

- Smoothed expected risk:

$$\mathcal{L}^\delta(x) = E_{v\sim B}\Big[E_{z\sim\mathcal{D}(x+\delta v)}\left[\ell(x + \delta v,\ z)\right]\Big]$$

  This is strongly convex.

- See the paper for the sample complexity proof.

### 3.2 - First Order Stochastic Gradient Method

- In this information setting, we assume the decision-maker has access to a parametric description of $\mathcal{D}(x)$.

- Expected loss:

$$\mathcal{L}_t(x) = E_{z\sim p_t} \ell(x_t,\ z)$$

- Differentiating (under a mild smoothness assumption):

$$\nabla \mathcal{L}_t(x) = E_{z\sim p_t}\left[\nabla_x \ell(x, z) + (1 - \lambda^n) A^T \nabla_z \ell(x, z)\right]$$

- Therefore, given a decision $x$, the decision-maker can always sample a response $z \sim p_t$ and form the unbiased estimator of $E_{z\sim p_t}[\hat{g}_t] = \nabla \mathcal{L}_t(x)$:

$$\hat{g}_t = \nabla \ell(x_t, z) = \nabla_x \ell(x, z) + (1 - \lambda^n) A^T \nabla_z \ell(x, z)$$

- Algorithm:
    - For each round $t$:
        - Decision-maker queries the environment with $x_t$ for $n$ steps.
        - Resulting in the distribution, $p_t = \lambda^n p_{t-1} + (1 - \lambda^n)\mathcal{D}(x_t)$.
        - The gradient oracle reveals $\hat{g}_t$
        - And, the update rule is applied $x_{t+1} = \text{proj}_{\mathcal{X}}(x_t - \eta_t \hat{g}_t)$.

- See the paper for the sample complexity proof.

## 4 - Numerical Experiments

- Target metric - between 60 and 80% occupancy rates.

- Rate structure:
    - Operational hours are split into three distinct rate periods.
    - Rates are adjusted on a block-by-block basis.
    - Rates are adjusted based on current occupancy.
    - Only weekday, non-special event parking is considered.

- Problem setup:
    - Decision $x$ is the changes in price from the nominal price relative to SF Park's initial rates.

- Fixed point distribution $z$ is the curb occupancies.

- Dynamic decision-dependent loss:

$$E_{z\sim p_t}\big[(\ell(x,z))\big] = E\big[||z-0.7||^2 + \frac{\nu}{2}||x||^2\big]$$

  where $\nu$ is the regularization parameter.

- Data distribution is defined as follows:

$$z \sim \mathcal{D}(x) \Longleftrightarrow z = \zeta + Ax$$

  - $\zeta$ - follows the distribution $p_0$ which is sampled from the data at the beginning of the pilot study.
  - $A$ is a proxy for price elasticity which is estimated via linear fit between occupancy and price.

### 4.1 - Comparing Performative Optimum to SF Park

- Run the algorithms on a representative block.

- Chosen parameters:
  - $A \approx -0.157$ - a $1 increase in rates results in an approx. 15% decrease in occupancy.
  - $\lambda \approx 0.959$ - decay rate, see appendix for calculation.
  - SF Park initial rate for the block is $3.
  - $\mathcal{X} = [-3,\ 5]$ - decision space - min rate is $0; max rate is $8.
  - $\nu = 1e-3$ - regularization param.
  - Time step and epoch length are varied, in units of weeks.

- Analysis:
  - Choice of $(n,\ T)$ for Algorithm 1:
    - For the sample curb, $n = 8$ appears optimal .
    - Generally lower $n$ and higher $T$ yielded better results.
    - Different optimal $n$ were observed for different curbs.
    - Suggesting a non-uniform price update schedule.

  - Choice of $(n,\ T)$ for Algorithm 2:
    - Opposite is true, generally higher $n$ and lower $T$ yielded better results.
    - This is due to the variance in the query direction for Algorithm 1.

### 4.2 - Redistributing Parking Demand

- Examine a case with a high demand block (Hawthorne St 0) surrounded by lower demand blocks.

- Algorithm is able to reduce the occupancy at the high demand block from 90% to 70% and redistribute this to the lower demand blocks.

- This result is much better than SF Park.

- Interestingly, the algorithm does not perform well on a block where the unconstrained equilibrium price is higher than the max $8, set my SF Park.

## 5 - Discussion and Future Directions

- Future work:
  - Experiment with period dynamics.
  - May be possible to exploit additional structure on $\mathcal{D}(x)$ to improve sample complexity.