Stochastic Optimization with Decision-Dependent Distributions

Lin Xiao
Facebook AI Research

Joint work with
Dmitriy Drusvyatskiy (University of Washington)

Computer Science Seminar at UCLA
February 16, 2021

Classical stochastic optimization

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z\sim\mathcal{P}}\big[\ell(x,z)\big]$$

- stochastic approximation (Robbins & Monro 1951)

$$x_{t+1} = x_t - \eta_t \nabla \ell(x_t, z_t)$$

  dominant algorithmic framework in modern deep learning

- many variants
  - SGD with momentum, Nesterov acceleration
  - AdaGrad, RMSProp, Adam, ...
  - dual averaging
  - clipped stochastic gradient

Classical stochastic optimization

$$\underset{x}{\text{minimize}} \quad \underset{z\sim\mathcal{P}}{\mathbf{E}}\big[\ell(x,z)\big]$$

- stochastic approximation (Robbins & Monro 1951)

$$x_{t+1} = x_t - \eta_t \nabla \ell(x_t, z_t)$$

  dominant algorithmic framework in modern deep learning

- many variants
  - SGD with momentum, Nesterov acceleration
  - AdaGrad, RMSProp, Adam, ...
  - dual averaging
  - clipped stochastic gradient

- convergence guarantees require $\mathcal{P}$ being fixed ($z_t \sim \mathcal{P}$ for all t)

Decision-dependent distributional shift

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \left[\ell(x, z) \right]$$

Decision-dependent distributional shift

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \left[\ell(x, z) \right]$$

many applications in practice

- active interaction (gaming by manipulating features)
  - spam filtering
  - fraud detection
  - detection of abusive content (fake news, hate speech, ...)
- passive feedback
  - banks use classifier to approve load applications, impacting credit score of applications for downstream tasks

Decision-dependent distributional shift

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \left[\ell(x, z) \right]$$

many applications in practice

- active interaction (gaming by manipulating features)
  - spam filtering
  - fraud detection
  - detection of abusive content (fake news, hate speech, . . .)
- passive feedback
  - banks use classifier to approve load applications, impacting credit score of applications for downstream tasks

called **performative prediction** in machine learning context (Perdomo, Zrnic, Mendler-Dünner & Hardt 2020)

Strategic classification

(Hardt, Megiddo, Papadimitriou & Wootters 2016)

two-player online game between

- population of agents, each with feature $a \in \mathbf{R}^m$ and label $b \in \mathbf{R}$
- an institution that deploys classifier h_x, predicts $h_x(a) \approx b$

Strategic classification

(Hardt, Megiddo, Papadimitriou & Wootters 2016)

two-player online game between

- population of agents, each with feature $a \in \mathbf{R}^m$ and label $b \in \mathbf{R}$
- an institution that deploys classifier h_x, predicts $h_x(a) \approx b$

during each round of the game:

- agents adapt features to increase chance of positive classification

$$\hat{a}(h_x, a) := \arg\max_{a'} \{ u(h_x, a') - c(a, a') \}$$

  - $u(h_x, \cdot)$ is utility function of agent
  - $c(a, \cdot)$ is cost of altering features

Strategic classification

(Hardt, Megiddo, Papadimitriou & Wootters 2016)

two-player online game between

- population of agents, each with feature $a \in \mathbf{R}^m$ and label $b \in \mathbf{R}$
- an institution that deploys classifier h_x, predicts $h_x(a) \approx b$

during each round of the game:

- agents adapt features to increase chance of positive classification

$$\hat{a}(h_x, a) := \arg\max_{a'} \left\{ u(h_x, a') - c(a, a') \right\}$$

  - $u(h_x, \cdot)$ is utility function of agent
  - $c(a, \cdot)$ is cost of altering features

- institution can adjust x to minimize classification error

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{(\hat{a},b)\sim\mathcal{D}(x)} [\ell(h_x(\hat{a}), b)]$$

  - can only use random samples of agents

Strategic classification

(Hardt, Megiddo, Papadimitriou & Wootters 2016)

two-player online game between

- population of agents, each with feature $a \in \mathbf{R}^m$ and label $b \in \mathbf{R}$
- an institution that deploys classifier h_x, predicts $h_x(a) \approx b$

during each round of the game:

- agents adapt features to increase chance of positive classification

$$\hat{a}(h_x, a) := \arg\max_{a'} \big\{ u(h_x, a') - c(a, a') \big\}$$

  - $u(h_x, \cdot)$ is utility function of agent
  - $c(a, \cdot)$ is cost of altering features

- institution can adjust x to minimize classification error

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{(\hat{a},b)\sim\mathcal{D}(x)} [\ell(h_x(\hat{a}), b)]$$

  - can only use random samples of agents

(*in practice, agents unlikely play best responses, but* $(\hat{a}, b) \sim \mathcal{D}(x)$)

Optimization model

stochastic optimization with decision-dependent distributions

$$\underset{x}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \left[\ell(x, z) \right] + r(x)$$

- $\mathcal{D}(x)$ decision/state-dependent, accessible by sampling
- $\ell(\cdot, z)$ is a convex loss function
- $r(\cdot)$ is a convex, structure-inducing regularizer

Optimization model

stochastic optimization with decision-dependent distributions

$$\underset{x}{\text{minimize}} \quad \underset{z \sim \mathcal{D}(x)}{\mathbf{E}} \big[\ell(x, z)\big] + r(x)$$

- $\mathcal{D}(x)$ decision/state-dependent, accessible by sampling
- $\ell(\cdot, z)$ is a convex loss function
- $r(\cdot)$ is a convex, structure-inducing regularizer

hard to solve in general: **nonsmooth, nonconvex**

two paths forward:

1. impose structure on $\mathcal{D}(\cdot)$ and solve
   (Ahmed'00, Dupačová'06, Goel-Grossman'06, Hassani et al.'20, ...)
2. settle for an alternative, efficiently computable solution concept
   (Perdomo, Zrnic, Mendler-Dünner & Hardt 2020)

Equilibrium

- notation:

$$f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \nabla f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \nabla \ell(x, z)$$

Equilibrium

- notation:

$$f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \nabla f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \nabla \ell(x, z)$$

- definition of **equilibrium** (Perdomo et al '20):

$$\bar{x} = \operatorname*{argmin}_x \mathop{\mathbf{E}}_{z \sim \mathcal{D}(\bar{x})} \ell(x, z) + r(x)$$

"no incentive to change $\bar{x}$ based only on response $\mathcal{D}(\bar{x})$"

Equilibrium

- notation:

$$f_y(x) = \mathop{\mathbf{E}}_{z\sim\mathcal{D}(y)} \ell(x,z), \qquad \nabla f_y(x) = \mathop{\mathbf{E}}_{z\sim\mathcal{D}(y)} \nabla\ell(x,z)$$

- definition of **equilibrium** (Perdomo et al '20):

$$\bar{x} = \operatorname*{argmin}_x \; f_{\bar{x}}(x) + r(x)$$

"no incentive to change $\bar{x}$ based only on response $\mathcal{D}(\bar{x})$"

Equilibrium

- notation:

$$f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \nabla f_y(x) = \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \nabla \ell(x, z)$$

- definition of **equilibrium** (Perdomo et al '20):

$$\bar{x} = \operatorname*{argmin}_x \; f_{\bar{x}}(x) + r(x)$$

"no incentive to change $\bar{x}$ based only on response $\mathcal{D}(\bar{x})$"

- **algorithmically:** these are fixed points of the map

$$S(y) := \operatorname*{argmin}_x \; f_y(x) + r(x)$$

suggests a fixed-point algorithm (repeated minimization)

Performative prediction

Perdomo, Zrnic, Mendler-Dünner & Hardt (ICML 2020, NeurIPS 2020):

- proposed this framework (performative stable solutions)
- established existence of equilibria
- showed convergence of following algorithms:
  - repeated risk minimization (conceptual)

$$x_{t+1} = \underset{x}{\operatorname{argmin}}\{f_{x_t}(x) + r(x)\}$$

  - projected gradient descent (conceptual)

$$x_{t+1} = \operatorname{prox}_{\eta r}(x_t - \eta \nabla f_{x_t}(x_t))$$

  - projected stochastic gradient (practical)

$$\text{sample } z_t \sim \mathcal{D}(x_t)$$
$$x_{t+1} = \operatorname{prox}_{\eta r}(x_t - \eta \nabla \ell(x_t, z_t))$$

Our contributions

- goal: find equilibrium

$$\bar{x} = \operatorname*{argmin}_{x} \mathop{\mathbf{E}}_{z \sim \mathcal{D}(\bar{x})} \left[f(x, z) \right] + r(x)$$

Our contributions

- goal: find equilibrium

$$\bar{x} = \underset{x}{\operatorname{argmin}} \underset{z \sim \mathcal{D}(\bar{x})}{\mathbf{E}} \big[f(x,z)\big] + r(x)$$

- **Meta Theorem:** Algorithms that sample from $\mathcal{D}(x_t)$ can be viewed as same algorithms applied to the **static problem**

$$\underset{x}{\text{minimize}} \quad \underset{z \sim \mathcal{D}(\bar{x})}{\mathbf{E}} \big[f(x,z)\big] + r(x)$$

with "bias," and the "bias" $\to 0$ linearly as $x_t \to \bar{x}$.

Our contributions

- goal: find equilibrium

$$\bar{x} = \operatorname*{argmin}_x \mathop{\mathbf{E}}_{z \sim \mathcal{D}(\bar{x})} \big[f(x, z) \big] + r(x)$$

- **Meta Theorem:** Algorithms that sample from $\mathcal{D}(x_t)$ can be viewed as same algorithms applied to the **static problem**

$$\operatorname*{minimize}_x \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(\bar{x})} \big[f(x, z) \big] + r(x)$$

  with "bias," and the "bias" $\to 0$ linearly as $x_t \to \bar{x}$.

- sharp convergence guarantees for many popular algorithms:
  - proximal point
  - stochastic gradient
  - clipped stochastic gradient
  - dual averaging

  and their **accelerated** and **proximal** variants

Outline of rest of talk

- notation and assumptions
- two deviation inequalities
- reduction to online convex optimization
- (accelerated) stochastic gradient method
- model-based algorithms

Notation and assumptions

- **strong convexity:** loss $\ell(\cdot, z)$ is α-strongly convex:

$$\ell(x, z) \geq \ell(y, z) + \langle \nabla \ell(y, z), x - y \rangle + \frac{\alpha}{2} \|x - y\|^2$$

- **smoothness**: $\ell(\cdot, \cdot)$ satisfies

$$\|\nabla \ell(x, z) - \nabla \ell(x', z)\| \leq L \cdot \|x - x'\|$$
$$\|\nabla \ell(x, z) - \nabla \ell(x, z')\| \leq \beta \cdot \|z - z'\|$$

- **sensitivity** of $\mathcal{D}(\cdot)$ in Wasserstein-1 distance

$$W_1\big(\mathcal{D}(x), \mathcal{D}(y)\big) \leq \gamma \cdot \|x - y\|$$

conditioning measures:

$$\kappa = \frac{L}{\alpha} \qquad \text{and} \qquad \rho = \frac{\gamma\beta}{\alpha}$$

Interesting regime is $\rho \in (0, 1)$

- repeated risk minimization (RRM)

$$x_{t+1} = \operatorname*{argmin}_x \; f_{x_t}(x) + r(x)$$

theorem (Perdomo et al. 2020):

- if $\rho < 1$, then RRM converges to $\bar{x}$ at linear rate ρ
- if $\rho > 1$, then RRM may diverge

Interesting regime is $\rho \in (0, 1)$

- repeated risk minimization (RRM)

$$x_{t+1} = \operatorname*{argmin}_x \; f_{x_t}(x) + r(x)$$

theorem (Perdomo et al. 2020):
- if $\rho < 1$, then RRM converges to $\bar{x}$ at linear rate ρ
- if $\rho > 1$, then RRM may diverge

- proximal point method (PPM)

$$x_{t+1} = \operatorname*{argmin}_x \; f_{x_t}(x) + r(x) + \frac{1}{2\eta}\|x - x_t\|^2$$

theorem (Drusvyatskiy-X 2020):
If $\rho < 1$, then PPM converges to $\bar{x}$ at linear rate $1 - \frac{1-\rho}{1+(\alpha\eta)^{-1}}$

Interesting regime is $\rho \in (0, 1)$

- repeated risk minimization (RRM)

$$x_{t+1} = \operatorname*{argmin}_x \; f_{x_t}(x) + r(x)$$

theorem (Perdomo et al. 2020):
- if $\rho < 1$, then RRM converges to $\bar{x}$ at linear rate ρ
- if $\rho > 1$, then RRM may diverge

- proximal point method (PPM)

$$x_{t+1} = \operatorname*{argmin}_x \; f_{x_t}(x) + r(x) + \frac{1}{2\eta}\|x - x_t\|^2$$

theorem (Drusvyatskiy-X 2020):
If $\rho < 1$, then PPM converges to $\bar{x}$ at linear rate $1 - \frac{1-\rho}{1+(\alpha\eta)^{-1}}$

empirically: PPM more "distributionally stable"

Numerical illustration

chasing the mean:

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

Numerical illustration

chasing the mean:

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \underset{z \sim \mathcal{D}(x)}{\mathbf{E}} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

- $\alpha = \beta = 1$ and $\gamma = \rho$, thus

$$\rho = \gamma\beta/\alpha$$

- vector field

$$\begin{aligned} \nabla f_y(x) &= x - \mathbf{E}_{z \sim \mathcal{D}(y)}(z) \\ &= \begin{bmatrix} x_1 - \rho y_2 \\ x_2 - \rho y_1 \end{bmatrix} \end{aligned}$$

- equilibrium point: $\bar{x} = (0, 0)$

Numerical illustration

chasing the mean:

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

- $\alpha = \beta = 1$ and $\gamma = \rho$, thus

$$\rho = \gamma\beta/\alpha$$

- vector field

$$\nabla f_y(x) = x - \mathbf{E}_{z \sim \mathcal{D}(y)}(z) = \begin{bmatrix} x_1 - \rho y_2 \\ x_2 - \rho y_1 \end{bmatrix}$$

- equilibrium point: $\bar{x} = (0, 0)$

$\nabla f_x(x)$ versus $\nabla f_{\bar{x}}(x)$

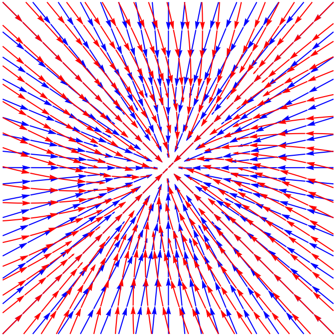


Figure: $\rho = 0.25$

Numerical illustration

chasing the mean:

$$\underset{x\in\mathbf{R}^2}{\text{minimize}} \quad \underset{z\sim\mathcal{D}(x)}{\mathbf{E}} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

- $\alpha = \beta = 1$ and $\gamma = \rho$, thus

$$\rho = \gamma\beta/\alpha$$

- vector field

$$\nabla f_y(x) = x - \mathbf{E}_{z\sim\mathcal{D}(y)}(z) = \begin{bmatrix} x_1 - \rho y_2 \\ x_2 - \rho y_1 \end{bmatrix}$$

- equilibrium point: $\bar{x} = (0, 0)$

$\nabla f_x(x)$ versus $\nabla f_{\bar{x}}(x)$

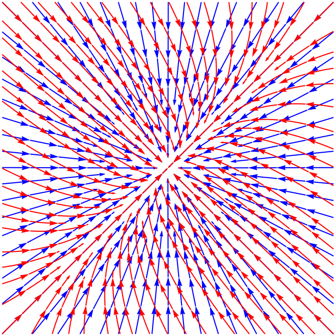


Figure: $\rho = 0.5$

Numerical illustration

chasing the mean:

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \mathop{\mathbf{E}}_{z \sim \mathcal{D}(x)} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

- $\alpha = \beta = 1$ and $\gamma = \rho$, thus

$$\rho = \gamma\beta/\alpha$$

- vector field

$$\nabla f_y(x) = x - \mathbf{E}_{z \sim \mathcal{D}(y)}(z) = \begin{bmatrix} x_1 - \rho y_2 \\ x_2 - \rho y_1 \end{bmatrix}$$

- equilibrium point: $\bar{x} = (0, 0)$

$\nabla f_x(x)$ versus $\nabla f_{\bar{x}}(x)$

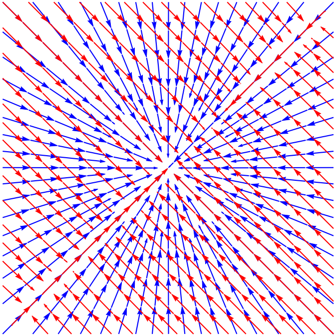


Figure: $\rho = 0.99$

Numerical illustration

chasing the mean:

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \underset{z \sim \mathcal{D}(x)}{\mathbf{E}} \|x - z\|^2 \quad \text{where} \quad \mathcal{D}(x_1, x_2) = N(\rho x_2, \rho x_1), I)$$

- $\alpha = \beta = 1$ and $\gamma = \rho$, thus

$$\rho = \gamma\beta/\alpha$$

- vector field

$$\nabla f_y(x) = x - \mathbf{E}_{z \sim \mathcal{D}(y)}(z) = \begin{bmatrix} x_1 - \rho y_2 \\ x_2 - \rho y_1 \end{bmatrix}$$

- equilibrium point: $\bar{x} = (0, 0)$

$\nabla f_x(x)$ versus $\nabla f_{\bar{x}}(x)$

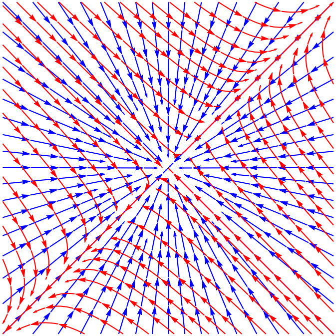


Figure: $\rho = 1.25$

Empirical study: regularization helps!

- RRM: $x_{t+1} = \text{argmin}_x \; f_{x_t}(x) + r(x)$
- PPM: $x_{t+1} = \text{argmin}_x \; f_{x_t}(x) + r(x) + \frac{1}{2\eta}\|x - x_t\|^2$

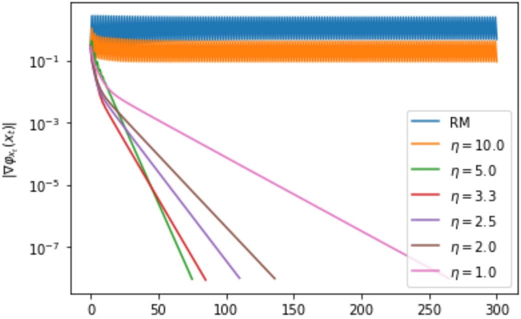


Figure: Strategic classification with $\rho > 1$

Empirical study: regularization helps!

- RRM: $x_{t+1} = \text{argmin}_x \;\; f_{x_t}(x) + r(x)$
- PPM: $x_{t+1} = \text{argmin}_x \;\; f_{x_t}(x) + r(x) + \frac{1}{2\eta}\|x - x_t\|^2$

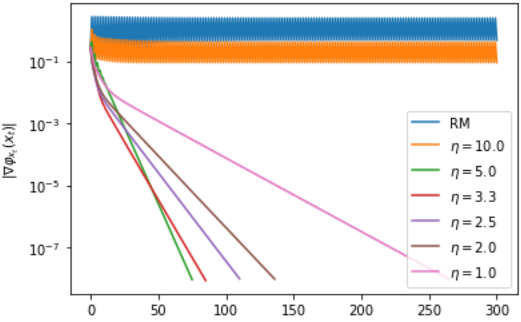


Figure: Strategic classification with $\rho > 1$

(conceptual algorithms, not feasible for practical applications)

Two deviation inequalities

- definitions:

$$f_y(x) := \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \mathcal{G}_y(x, x') := f_y(x) - f_y(x')$$

Two deviation inequalities

- definitions:

$$f_y(x) := \mathop{\mathbf{E}}_{z\sim\mathcal{D}(y)} \ell(x,z), \qquad \mathcal{G}_y(x,x') := f_y(x) - f_y(x')$$

- **lemma** (gradient deviation): for all $y, y' \in \mathbf{R}^d$ it holds:

$$\sup_{x\in\mathbf{R}^d} \|\nabla f_{y}(x) - \nabla f_{y'}(x)\| \le \gamma\beta \cdot \|y - y'\|$$

Two deviation inequalities

- definitions:

$$f_y(x) := \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \mathcal{G}_y(x, x') := f_y(x) - f_y(x')$$

- **lemma** (gradient deviation): for all $y, y' \in \mathbf{R}^d$ it holds:

$$\sup_{x \in \mathbf{R}^d} \|\nabla f_y(x) - \nabla f_{y'}(x)\| \leq \gamma\beta \cdot \|y - y'\|$$

implication: $\texttt{Bias}(x) := \|\nabla f_x(x) - \nabla f_{\bar{x}}(x)\| \leq \gamma\beta \cdot \|x - \bar{x}\|$

Two deviation inequalities

- definitions:

$$f_y(x) := \mathop{\mathbf{E}}_{z \sim \mathcal{D}(y)} \ell(x, z), \qquad \mathcal{G}_y(x, x') := f_y(x) - f_y(x')$$

- **lemma** (gradient deviation): for all $y, y' \in \mathbf{R}^d$ it holds:

$$\sup_{x \in \mathbf{R}^d} \|\nabla f_y(x) - \nabla f_{y'}(x)\| \le \gamma\beta \cdot \|y - y'\|$$

implication: $\texttt{Bias}(x) := \|\nabla f_x(x) - \nabla f_{\bar{x}}(x)\| \le \gamma\beta \cdot \|x - \bar{x}\|$

- **lemma** (gap deviation): all $x, x' \in \mathbf{R}^d$ and $y, y' \in \mathbf{R}^d$ satisfy:

$$\mathcal{G}_y(x, x') - \mathcal{G}_{y'}(x, x') \le \gamma\beta \cdot \|x - x'\| \cdot \|y - y'\|$$

Two deviation inequalities

- definitions:

$$f_y(x) := \mathop{\mathbf{E}}_{z\sim\mathcal{D}(y)} \ell(x,z), \qquad \mathcal{G}_y(x,x') := f_y(x) - f_y(x')$$

- **lemma** (gradient deviation): for all $y, y' \in \mathbf{R}^d$ it holds:

$$\sup_{x\in\mathbf{R}^d} \|\nabla f_{y}(x) - \nabla f_{y'}(x)\| \leq \gamma\beta \cdot \|y - y'\|$$

implication: $\texttt{Bias}(x) := \|\nabla f_x(x) - \nabla f_{\bar{x}}(x)\| \leq \gamma\beta \cdot \|x - \bar{x}\|$

- **lemma** (gap deviation): all $x, x' \in \mathbf{R}^d$ and $y, y' \in \mathbf{R}^d$ satisfy:

$$\mathcal{G}_{y}(x,x') - \mathcal{G}_{y'}(x,x') \leq \gamma\beta \cdot \|x - x'\| \cdot \|y - y'\|$$

implication: $\mathcal{G}_x(x,\bar{x}) - \mathcal{G}_{\bar{x}}(x,\bar{x}) \leq \gamma\beta \cdot \|x - \bar{x}\|^2$

(can be offset by strong convexity)

Online convex optimization

- **a repeated game:** for $t = 1, 2, \ldots$
  - player chooses $x_t \in \mathbf{dom}\, r$
  - nature reveals function ℓ_t and player pays $\ell_t(x_t)$

player's goal: minimize the regret

$$R_t := \sum_{i=1}^{t} \big(\ell_i(x_i) + r(x_i)\big) - \min_x \sum_{i=1}^{t} \big(\ell_i(x) + r(x)\big)$$

Online convex optimization

- **a repeated game:** for $t = 1, 2, \ldots$
  - player chooses $x_t \in \operatorname{dom} r$
  - nature reveals function ℓ_t and player pays $\ell_t(x_t)$

  player's goal: minimize the regret

$$R_t := \sum_{i=1}^{t} \big(\ell_i(x_i) + r(x_i)\big) - \min_x \sum_{i=1}^{t} \big(\ell_i(x) + r(x)\big)$$

- **algorithms:**
  - online prox-gradient (Duchi-Singer 2009)
  - regularized dual averaging (X 2010)
  - follow-the-regularized-leader (FTRL) (McMahan 2011, . . .)
- **guarantees**

$$\left.\begin{array}{l} \ell_t \text{ are } \alpha\text{-strongly convex on } \operatorname{dom} r \\ \ell_t \text{ are } G\text{-Lipschitz on } \operatorname{dom} r \end{array}\right\} \implies R_t = \mathcal{O}\left(\frac{G^2 \log t}{\alpha}\right)$$

Reduction to online convex optimization

- finding **performative equilibrium** is equivalent to

$$\underset{x}{\text{minimize}} \;\; \varphi(x) := \underset{z \sim \mathcal{D}(\bar{x})}{\mathbf{E}} [\ell(x, z)] + r(x)$$

Reduction to online convex optimization

- finding **performative equilibrium** is equivalent to

$$\underset{x}{\text{minimize}} \;\; \varphi(x) := \mathop{\mathbf{E}}_{z \sim \mathcal{D}(\bar{x})}[\ell(x, z)] + r(x)$$

- **theorem** (Drusvyatskiy-X 2020):
Suppose $\rho \in (0, \frac{1}{2})$. Run an online algorithm where in iteration t, nature draws $z_t \sim \mathcal{D}(x_t)$ and declares $\ell_t(x_t) = \ell(x_t, z_t)$. Then

$$\mathbf{E}\left[\varphi\left(\frac{1}{t}\sum_{i=1}^{t} x_i\right) - \varphi(\bar{x})\right] \;\leq\; \frac{\mathbf{E}[R_t]}{(1-2\rho)\, t} \;=\; \mathcal{O}\left(\frac{\log t}{(1-2\rho)\alpha t}\right)$$

Reduction to online convex optimization

- finding **performative equilibrium** is equivalent to

$$\underset{x}{\text{minimize}} \;\; \varphi(x) := \underset{z \sim \mathcal{D}(\bar{x})}{\mathbf{E}} [\ell(x, z)] + r(x)$$

- **theorem** (Drusvyatskiy-X 2020):
Suppose $\rho \in (0, \frac{1}{2})$. Run an online algorithm where in iteration t, nature draws $z_t \sim \mathcal{D}(x_t)$ and declares $\ell_t(x_t) = \ell(x_t, z_t)$. Then

$$\mathbf{E}\left[\varphi\left(\frac{1}{t}\sum_{i=1}^{t} x_i\right) - \varphi(\bar{x})\right] \;\leq\; \frac{\mathbf{E}[R_t]}{(1-2\rho)\,t} \;=\; \mathcal{O}\left(\frac{\log t}{(1-2\rho)\alpha t}\right)$$

- **downside:** strong assumptions (bounded domain, Lipschitz loss)

instead, can analyze algorithms directly, assuming **finite-variance**:

$$\underset{z \sim \mathcal{D}(x)}{\mathbf{E}} \|\nabla \ell(x, z) - \nabla f_x(x)\|^2 \leq \sigma^2, \qquad \forall x$$

Proximal stochastic gradient (SG)

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \mathrm{prox}_{\eta r}(x_t - \eta \nabla \ell(x_t, z_t))$$

Proximal stochastic gradient (SG)

- **algorithm:**

$$\text{sample } z_t \sim \mathcal{D}(x_t)$$
$$x_{t+1} = \text{prox}_{\eta r}(x_t - \eta \nabla \ell(x_t, z_t))$$

- **theorem** (Drusvyatskiy-X 2020, Mendler-Dünner et al. 2020)
  - If $\rho < 1$, proximal SG finds x with $\|x - \bar{x}\|^2 \leq \varepsilon$ using

$$\mathcal{O}\left(\kappa \cdot \log\left(\frac{\|x_0 - \bar{x}\|^2}{\varepsilon}\right) + \frac{\sigma^2}{(1-\rho)^2 \alpha^2 \varepsilon}\right) \quad \text{samples}$$

Proximal stochastic gradient (SG)

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \mathrm{prox}_{\eta r}(x_t - \eta \nabla \ell(x_t, z_t))$$

- **theorem** (Drusvyatskiy-X 2020, Mendler-Dünner et al. 2020)
  - If $\rho < 1$, proximal SG finds x with $\|x - \bar{x}\|^2 \leq \varepsilon$ using

$$\mathcal{O}\left(\kappa \cdot \log\left(\frac{\|x_0 - \bar{x}\|^2}{\varepsilon}\right) + \frac{\sigma^2}{(1-\rho)^2\alpha^2\varepsilon}\right) \quad \text{samples}$$

  - If $\rho < \frac{1}{2}$, proximal SG finds x with $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \leq \varepsilon$ using

$$\mathcal{O}\left(\kappa \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon}\right) + \frac{\sigma^2}{(1-2\rho)\alpha\varepsilon}\right) \quad \text{samples}$$

Proximal stochastic gradient (SG)

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \text{prox}_{\eta r}(x_t - \eta \nabla \ell(x_t, z_t))$$

- **theorem** (Drusvyatskiy-X 2020, Mendler-Dünner et al. 2020)
  - If $\rho < 1$, proximal SG finds x with $\|x - \bar{x}\|^2 \leq \varepsilon$ using

$$\mathcal{O}\left(\kappa \cdot \log\left(\frac{\|x_0 - \bar{x}\|^2}{\varepsilon}\right) + \frac{\sigma^2}{(1-\rho)^2 \alpha^2 \varepsilon}\right) \quad \text{samples}$$

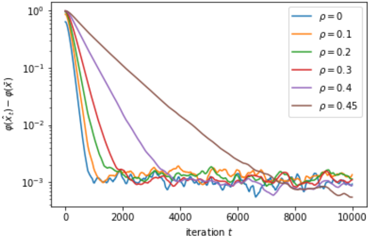
  - If $\rho < \frac{1}{2}$, proximal SG finds x with $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \leq \varepsilon$ using

$$\mathcal{O}\left(\kappa \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon}\right) + \frac{\sigma^2}{(1-2\rho)\alpha\varepsilon}\right) \quad \text{samples}$$

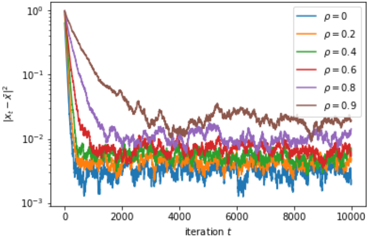
- **remark:** reduces to classical rate if $\rho = 0$ (e.g., Lan 2010)

Numerical experiments of SG method

chasing the mean: SG with constant step size $\eta = 0.01$



(a) function value gap of $\hat{x}_t$



(b) squared distance to $\bar{x}$

- $\varphi(\hat{x}_t) - \varphi(\bar{x})$ decreases linearly to noise level controlled by η
- linear rate degrades as ρ tends to $1/2$
- $\|x_t - \bar{x}\|^2$ decreases linearly to noise level depending on η and ρ

Proximal accelerated stochastic gradient (ASG)

- **algorithm:** (adapted from Kulunchakov-Mairal 2019)

$$\text{sample } z_t \sim \mathcal{D}(y_{t-1}) \text{ and set } g_t = \nabla \ell(y_{t-1}, z_t)$$

$$x_t = \mathrm{prox}_{\eta_t r}(y_{t-1} - \eta g_t)$$

$$y_t = x_t + \frac{1-\sqrt{\eta\alpha(1-2\rho)}}{1+\sqrt{\eta\alpha(1-2\rho)}}(x_t - x_{t-1})$$

Proximal accelerated stochastic gradient (ASG)

- **algorithm:** (adapted from Kulunchakov-Mairal 2019)

$$\text{sample } z_t \sim \mathcal{D}(y_{t-1}) \text{ and set } g_t = \nabla \ell(y_{t-1}, z_t)$$
$$x_t = \mathrm{prox}_{\eta_t r}(y_{t-1} - \eta g_t)$$
$$y_t = x_t + \frac{1-\sqrt{\eta\alpha(1-2\rho)}}{1+\sqrt{\eta\alpha(1-2\rho)}}(x_t - x_{t-1})$$

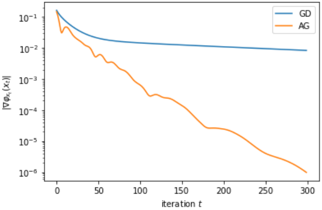
- **theorem** (Drusvyatskiy-X 2020): If $\rho \lesssim \kappa^{-1/4}$, proximal ASG finds x satisfying $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \le \varepsilon$ using

$$\mathcal{O}\left(\sqrt{\kappa} \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon}\right) + \frac{\sigma^2}{\alpha\varepsilon}\right) \qquad \text{samples}$$

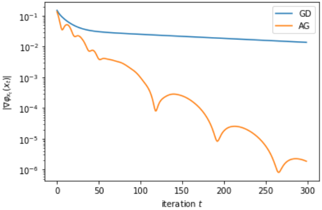
  - $\rho \lesssim \kappa^{-1/4}$ looks suboptimal; can it be improved?
  - somewhat surprising to have acceleration for any $\rho > 0$

- **proof:** technical, using variant of stochastic estimate sequences

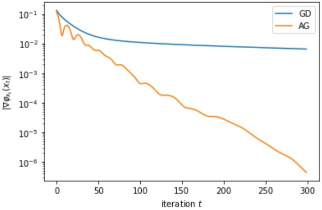
Acceleration works mysteriously well!



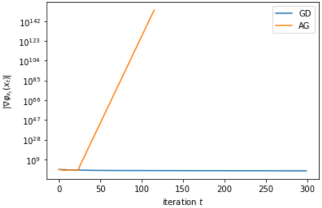
(a) $\gamma = 0$.



(b) $\gamma = 5$.



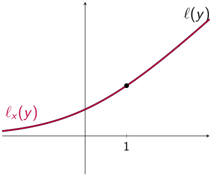
(c) $\gamma = 100$.



(d) $\gamma = 250$.

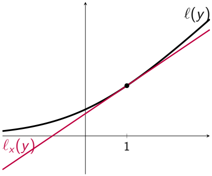
experiments with a strategic classification problem ($\sigma = 0$)

Model-based algorithms



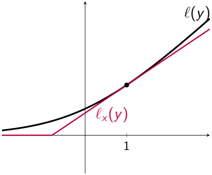
prox−point

$$\ell_x(y) = \ell(y)$$



gradient

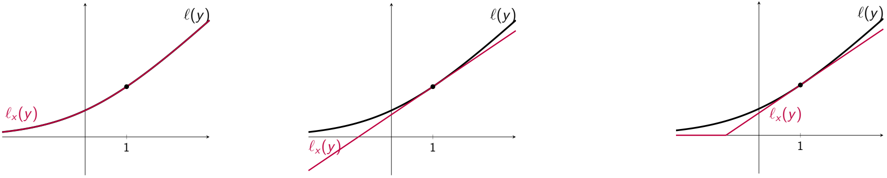
$$\ell_x(y) = \ell(x) + \langle \nabla \ell(x), y - x \rangle$$



clipped gradient

$$\ell_x(y) = (\ell(x) + \langle \nabla \ell(x), y - x \rangle)^+$$

Model-based algorithms



prox−point	gradient	clipped gradient
$\ell_x(y) = \ell(y)$	$\ell_x(y) = \ell(x) + \langle \nabla \ell(x), y - x \rangle$	$\ell_x(y) = (\ell(x) + \langle \nabla \ell(x), y - x \rangle)^+$

algorithm template:

$$\text{sample } z_t \sim \mathcal{D}(x_t)$$

$$x_{t+1} = \underset{y}{\operatorname{argmin}} \left\{ \ell_{x_t}(y, z_t) + r(y) + \frac{1}{2\eta} \|y - x_t\|^2 \right\}$$

(clipped gradient model introduced in Asi-Duchi 2019)

Model-based algorithms

Assumptions: there exist $\alpha_1, \alpha_2 \geq 0$ such that

- **convexity**

$$\ell_x(\cdot, z) \text{ is convex}, \qquad \ell_x(\cdot, z) + r \text{ is } \alpha_1\text{-strongly convex}$$

- **bias/variance**

$$\mathop{\mathbf{E}}_z[\nabla \ell_x(x, z)] = \nabla f_x(x), \qquad \mathop{\mathbf{E}}_z \|\nabla \ell_x(x, z) - \nabla f_x(x)\|^2 \leq \sigma^2$$

- **accuracy**

$$\mathop{\mathbf{E}}_z[\ell_x(x, z)] = f_x(x), \qquad \mathop{\mathbf{E}}_z[\ell_x(y, z)] + \frac{\alpha_2}{2}\|x - y\|^2 \leq f_x(y)$$

Model-based algorithms

Assumptions: there exist $\alpha_1, \alpha_2 \geq 0$ such that

- **convexity**

$$\ell_x(\cdot, z) \text{ is convex}, \qquad \ell_x(\cdot, z) + r \text{ is } \alpha_1\text{-strongly convex}$$

- **bias/variance**

$$\mathop{\mathbf{E}}_z[\nabla \ell_x(x, z)] = \nabla f_x(x), \qquad \mathop{\mathbf{E}}_z \|\nabla \ell_x(x, z) - \nabla f_x(x)\|^2 \leq \sigma^2$$

- **accuracy**

$$\mathop{\mathbf{E}}_z[\ell_x(x, z)] = f_x(x), \qquad \mathop{\mathbf{E}}_z[\ell_x(y, z)] + \frac{\alpha_2}{2}\|x - y\|^2 \leq f_x(y)$$

remark:

- similar assumptions in (Davis-Drusvyatskiy '19, Asi-Duchi '19)
- tighter models $\Rightarrow$ better algorithms (Ryu-Boyd '14, Asi-Duchi '19)

Model-based algorithms

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \operatorname*{argmin}_y \left\{ \ell_{x_t}(y, z_t) + r(y) + \frac{1}{2\eta}\|y - x_t\|^2 \right\}$$

- **theorem** (Drusvyatskiy-X '20)
  - if $\frac{\gamma\beta}{\alpha_1+\alpha_2} < \frac{1}{2}$, algorithm finds x with $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \le \varepsilon$ using

$$\mathcal{O}\left(\frac{L}{\alpha_1 + \alpha_2} \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon} \right) + \frac{\sigma^2}{(\alpha_1 + \alpha_2)\varepsilon} \right) \quad \text{samples}$$

Model-based algorithms

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \operatorname*{argmin}_y \left\{ \ell_{x_t}(y, z_t) + r(y) + \frac{1}{2\eta}\|y - x_t\|^2 \right\}$$

- **theorem** (Drusvyatskiy-X '20)
  - if $\frac{\gamma\beta}{\alpha_1+\alpha_2} < \frac{1}{2}$, algorithm finds x with $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \le \varepsilon$ using

$$\mathcal{O}\left(\frac{L}{\alpha_1 + \alpha_2} \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon} \right) + \frac{\sigma^2}{(\alpha_1 + \alpha_2)\varepsilon} \right) \quad \text{samples}$$

  - if $\frac{\gamma\beta}{\alpha_1+\alpha_2} < 1$, algorithm finds x with $\|x - \bar{x}\|^2 \le \varepsilon$ using

$$\mathcal{O}\left(\frac{L}{\alpha_1 + \alpha_2} \cdot \log\left(\frac{\|x_0 - \bar{x}\|^2}{\varepsilon} \right) + \frac{\sigma^2}{(\alpha_1 + \alpha_2)^2\varepsilon} \right) \quad \text{samples}$$

Model-based algorithms

- **algorithm:** sample $z_t \sim \mathcal{D}(x_t)$

$$x_{t+1} = \underset{y}{\text{argmin}} \left\{ \ell_{x_t}(y, z_t) + r(y) + \frac{1}{2\eta}\|y - x_t\|^2 \right\}$$

- **theorem** (Drusvyatskiy-X '20)
  - if $\frac{\gamma\beta}{\alpha_1+\alpha_2} < \frac{1}{2}$, algorithm finds x with $\mathbf{E}[\varphi(x) - \varphi(\bar{x})] \leq \varepsilon$ using

$$\mathcal{O}\left(\frac{L}{\alpha_1 + \alpha_2} \cdot \log\left(\frac{\varphi(x_0) - \varphi(\bar{x})}{\varepsilon}\right) + \frac{\sigma^2}{(\alpha_1 + \alpha_2)\varepsilon}\right) \quad \text{samples}$$

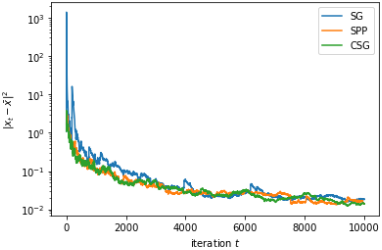
  - if $\frac{\gamma\beta}{\alpha_1+\alpha_2} < 1$, algorithm finds x with $\|x - \bar{x}\|^2 \leq \varepsilon$ using

$$\mathcal{O}\left(\frac{L}{\alpha_1 + \alpha_2} \cdot \log\left(\frac{\|x_0 - \bar{x}\|^2}{\varepsilon}\right) + \frac{\sigma^2}{(\alpha_1 + \alpha_2)^2\varepsilon}\right) \quad \text{samples}$$

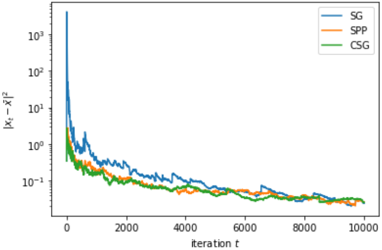
- rates for **stochastic PPM** and **clipped gradient** follow immediately

Numerical experiments of model-based algorithms

example of strategic classification, step size $\eta_t = \frac{2}{\alpha(t+1)}$



(a) $\gamma = 0.1$.



(b) $\gamma = 0.25$.

- all three methods perform similarly asymptotically
- initial stage:
  - SG sensitive to relatively large initial step sizes
  - stochastic PPM and clipped gradient more preferable

  (investigated in Asi-Duchi '19 for fixed distribution)

Inexact repeated minimization (IRM)

deployment of decision rule much more expensive than sampling

Inexact repeated minimization (IRM)

deployment of decision rule much more expensive than sampling

- state-wise algorithm

for $s = 1, 2, \ldots, S$ **do**
  $u_s = x_{s-1,T}$
  for $t = 1, 2, \ldots, T$ **do**
    sample $z_{s,t} \sim \mathcal{D}(u_s)$
    $x_{s,t+1} = \operatorname{argmin}_y \left\{ \ell_{x_{s,t}}(y, z_{s,t}) + r(y) + \frac{1}{2\eta}\|y - x_{s,t}\|^2 \right\}$

Inexact repeated minimization (IRM)

deployment of decision rule much more expensive than sampling

- state-wise algorithm

for $s = 1, 2, \ldots, S$ **do**

$u_s = x_{s-1,T}$

for $t = 1, 2, \ldots, T$ **do**

sample $z_{s,t} \sim \mathcal{D}(u_s)$

$x_{s,t+1} = \text{argmin}_y \left\{ \ell_{x_{s,t}}(y, z_{s,t}) + r(y) + \frac{1}{2\eta}\|y - x_{s,t}\|^2 \right\}$

- Mendler-Dünner, Perdomo, Zrnic, Hardt '20: established "deployments/samples" trade-off for IRM with SG method

- **theorem** (Drusvyatskiy-X '20)
  If $\rho < 1$, can implement IRM with all previous algorithms with same sample efficiency and only $\frac{1}{1-\rho}\log(1/\varepsilon)$ deployments.

Summary

- stochastic optimization with decision-dependent distributions

$$\underset{x}{\text{minimize}} \quad \underset{z \sim \mathcal{D}(x)}{\mathbf{E}} \big[\ell(x,z)\big]$$

tractable solution concept: **equilibrium** (Perdomo et al. '20)

$$\bar{x} = \underset{x}{\operatorname{argmin}} \underset{z \sim \mathcal{D}(\bar{x})}{\mathbf{E}} \big[f(x,z)\big] + r(x)$$

Summary

- stochastic optimization with decision-dependent distributions

$$\underset{x}{\text{minimize}} \quad \underset{z\sim\mathcal{D}(x)}{\mathbf{E}}\big[\ell(x,z)\big]$$

tractable solution concept: **equilibrium** (Perdomo et al. '20)

$$\bar{x} = \underset{x}{\operatorname{argmin}} \underset{z\sim\mathcal{D}(\bar{x})}{\mathbf{E}}\big[f(x,z)\big] + r(x)$$

- **Meta Theorem:** Algorithms that sample from $\mathcal{D}(x_t)$ can be viewed as same algorithms applied to the **static problem**

$$\underset{x}{\text{minimize}} \quad \underset{z\sim\mathcal{D}(\bar{x})}{\mathbf{E}}\big[f(x,z)\big] + r(x)$$

with "bias," and the "bias" $\to 0$ linearly as $x_t \to \bar{x}$.

References

details in the paper:

- "Stochastic optimization with decision-dependent distributions" Drusvyatskiy-X (2020), `arxiv.org/abs/2011.11173`

main references:

- "Performative prediction"
  Perdomo, Zrnic, Mendler-Dünner, Hardt (ICML 2020)
- "Stochastic optimization for performative prediction"
  Mendler-Dünner, Perdomo, Zrnic, Hardt (NeurIPS 2020)
- "Strategic classification"
  Hardt, Megiddo, Papadimitriou, Wootters (ACM ITCS '16)

References

details in the paper:

- "Stochastic optimization with decision-dependent distributions"
Drusvyatskiy-X (2020), `arxiv.org/abs/2011.11173`

main references:

- "Performative prediction"
  Perdomo, Zrnic, Mendler-Dünner, Hardt (ICML 2020)
- "Stochastic optimization for performative prediction"
  Mendler-Dünner, Perdomo, Zrnic, Hardt (NeurIPS 2020)
- "Strategic classification"
  Hardt, Megiddo, Papadimitriou, Wootters (ACM ITCS '16)

Thank you!