On Computational Thinking, Inferential Thinking and “Data Science”

Michael I. Jordan
University of California, Berkeley

November 18, 2016

A Job Description, circa 2015

- Your Boss: "*I need a Big Data system that will replace our classic service with a personalized service*"

A Job Description, circa 2015

- Your Boss: "*I need a Big Data system that will replace our classic service with a personalized service*"
- "*It should work reasonably well for anyone and everyone; I can tolerate a few errors but not too many dumb ones that will embarrass us*"

A Job Description, circa 2015

- Your Boss: “*I need a Big Data system that will replace our classic service with a personalized service*”
- “*It should work reasonably well for anyone and everyone; I can tolerate a few errors but not too many dumb ones that will embarrass us*”
- “*It should run just as fast as our classic service*”

A Job Description, circa 2015

- Your Boss: "*I need a Big Data system that will replace our classic service with a personalized service*"
- "*It should work reasonably well for anyone and everyone; I can tolerate a few errors but not too many dumb ones that will embarrass us*"
- "*It should run just as fast as our classic service*"
- "*It should only improve as we collect more data; in particular it shouldn't slow down*"

A Job Description, circa 2015

- Your Boss: "*I need a Big Data system that will replace our classic service with a personalized service*"
- "*It should work reasonably well for anyone and everyone; I can tolerate a few errors but not too many dumb ones that will embarrass us*"
- "*It should run just as fast as our classic service*"
- "*It should only improve as we collect more data; in particular it shouldn't slow down*"
- "*There are serious privacy concerns of course, and they vary across the clients*"

Some Challenges Driven by Big Data

- Big Data analysis requires a thorough blending of computational thinking and inferential thinking

Some Challenges Driven by Big Data

- Big Data analysis requires a thorough blending of computational thinking and inferential thinking
- What I mean by computational thinking
  - abstraction, modularity, scalability, robustness, etc.

Some Challenges Driven by Big Data

- Big Data analysis requires a thorough blending of computational thinking and inferential thinking
- What I mean by computational thinking
  - abstraction, modularity, scalability, robustness, etc.
- Inferential thinking means (1) considering the real-world phenomenon behind the data, (2) considering the sampling pattern that gave rise to the data, and (3) developing procedures that will go “backwards” from the data to the underlying phenomenon

Some Challenges Driven by Big Data

- Big Data analysis requires a thorough blending of computational thinking and inferential thinking
- What I mean by computational thinking
  - abstraction, modularity, scalability, robustness, etc.
- Inferential thinking means (1) considering the real-world phenomenon behind the data, (2) considering the sampling pattern that gave rise to the data, and (3) developing procedures that will go “backwards” from the data to the underlying phenomenon

The Challenges are Daunting

- The core theories in computer science and statistics developed separately and there is an oil and water problem
- Core statistical theory doesn't have a place for runtime and other computational resources
- Core computational theory doesn't have a place for statistical risk

Outline

- Inference under privacy constraints
- Inference under communication constraints
- Lower bounds, the variational perspective and symplectic integration

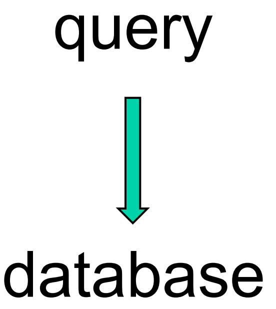
Part I: Inference and Privacy

with John Duchi and Martin Wainwright

Privacy and Data Analysis

- Individuals are not generally willing to allow their personal data to be used without control on how it will be used and now much privacy loss they will incur
- “Privacy loss” can be quantified via differential privacy
- We want to trade privacy loss against the value we obtain from “data analysis”
- The question becomes that of quantifying such value and juxtaposing it with privacy loss

Privacy



Privacy

query

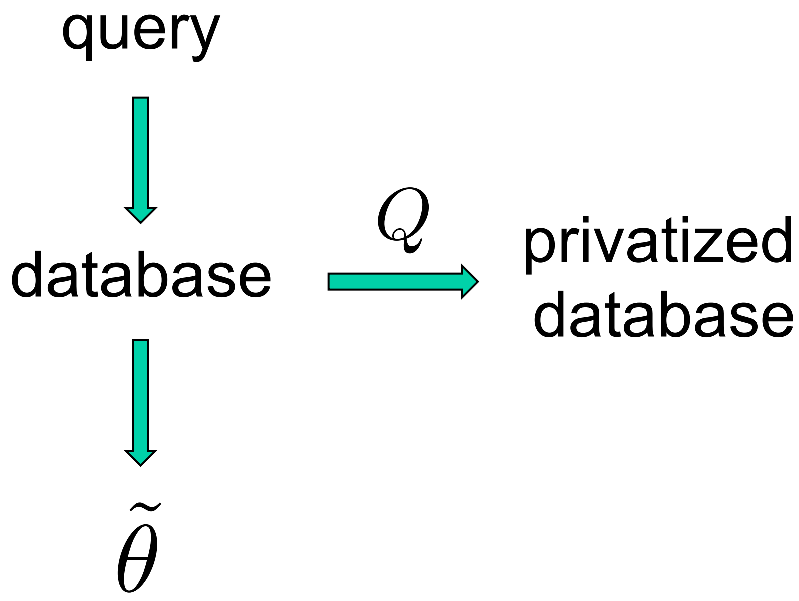
↓

database

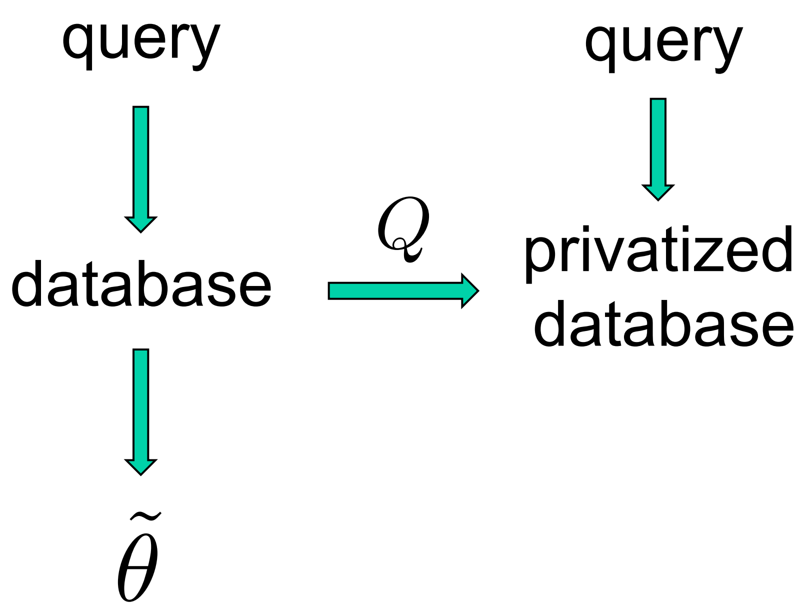
↓

$$\tilde{\theta}$$

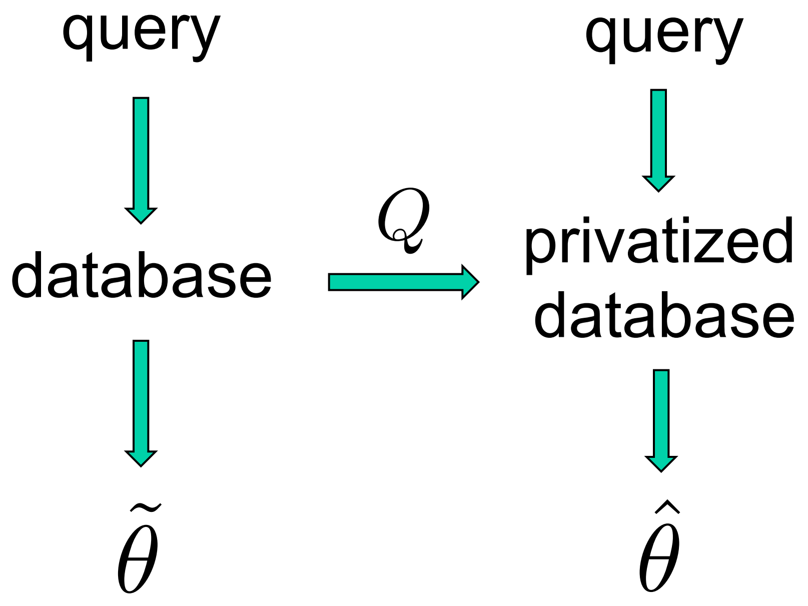
Privacy



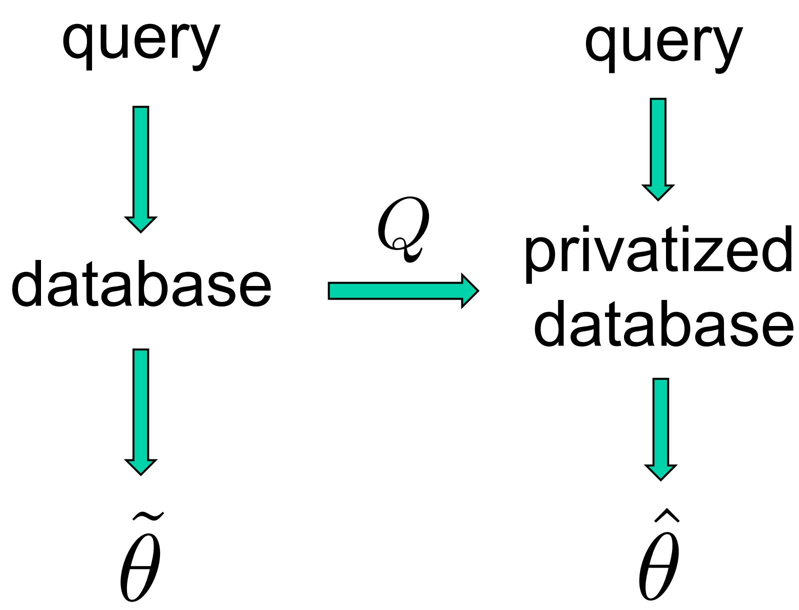
Privacy



Privacy


query
query
Q
database
privatized
database
$\tilde{\theta}$
$\hat{\theta}$

Privacy



Classical problem in differential privacy: show that $\hat{\theta}$ and $\tilde{\theta}$ are close under constraints on Q

Inference

query

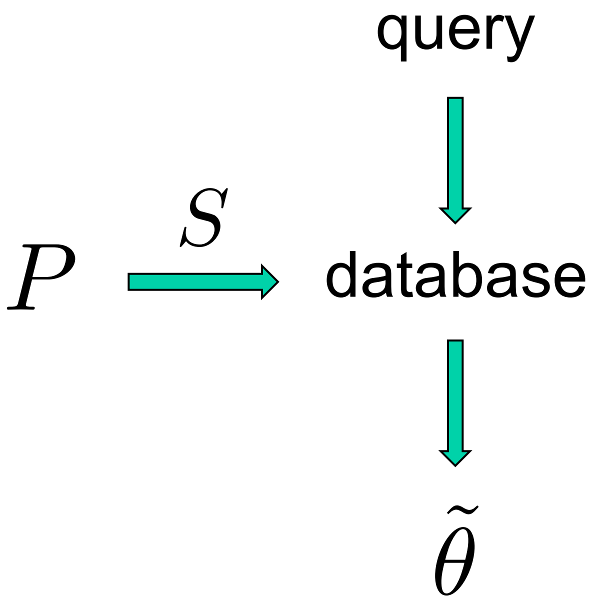
↓

database

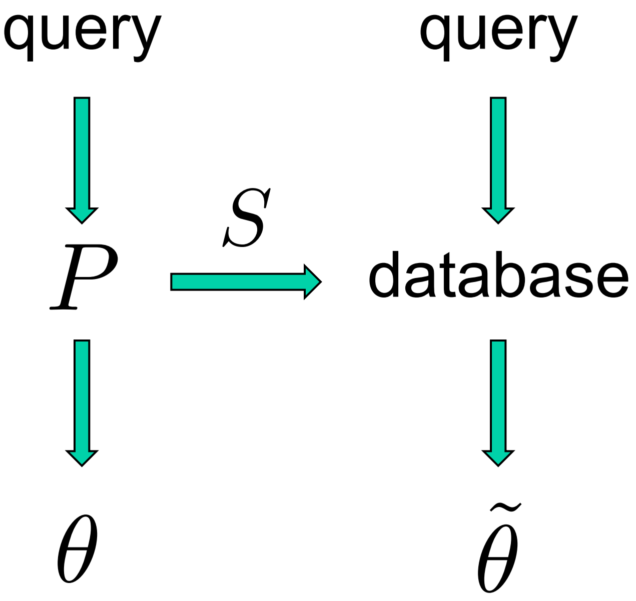
↓

$$\tilde{\theta}$$

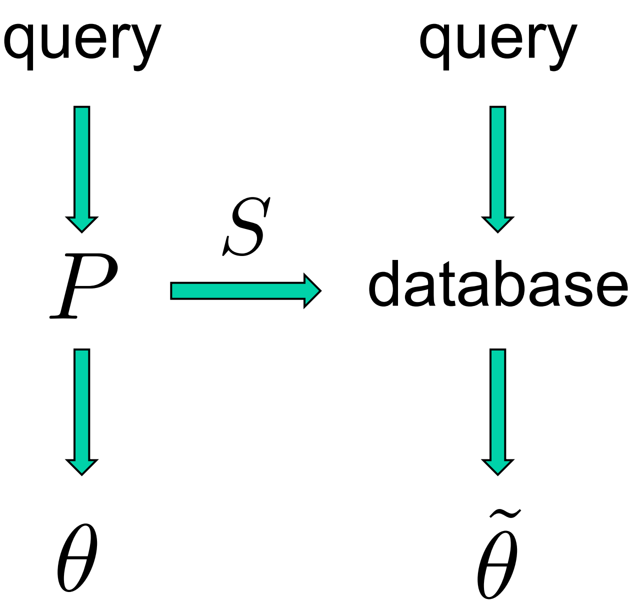
Inference



Inference

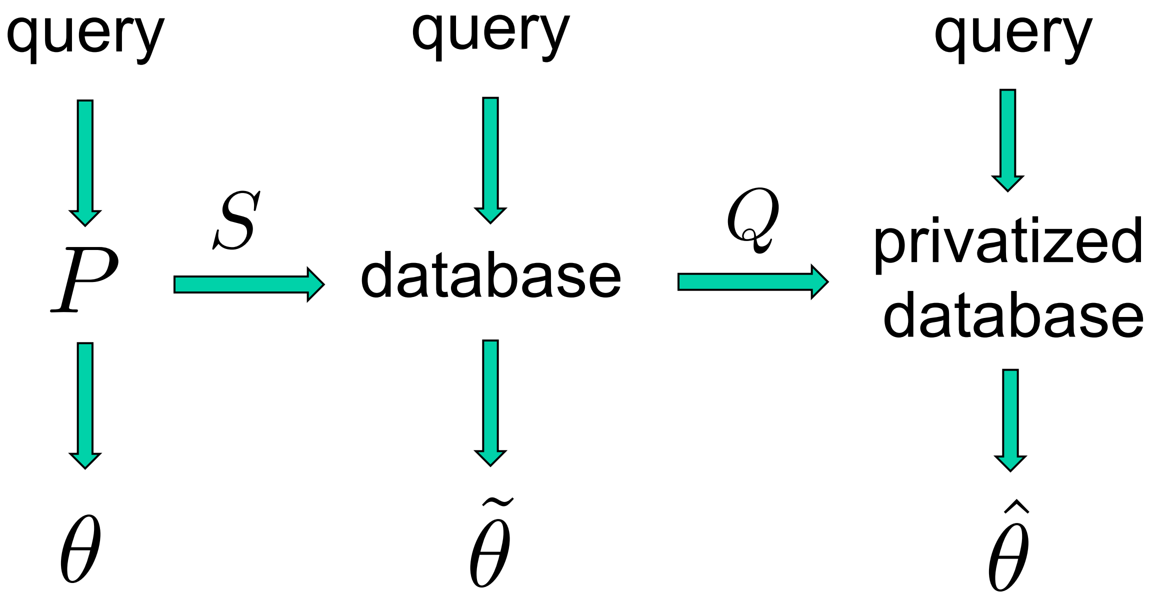


Inference



Classical problem in statistical theory: show that $\tilde{\theta}$ and θ are close under constraints on S

Privacy and Inference



The privacy-meets-inference problem: show that θ and $\hat{\theta}$ are close under constraints on Q and on S

Background on Inference

- In the 1930's, Wald laid the foundations of statistical decision theory
- Given a family of distributions $\mathcal{P}$, a parameter $\theta(P)$ for each $P \in \mathcal{P}$, an estimator $\hat{\theta}$, and a loss $l(\hat{\theta}, \theta(P))$, define the risk:

$$R_P(\hat{\theta}) := \mathbb{E}_P\left[l(\hat{\theta}, \theta(P))\right]$$

Background on Inference

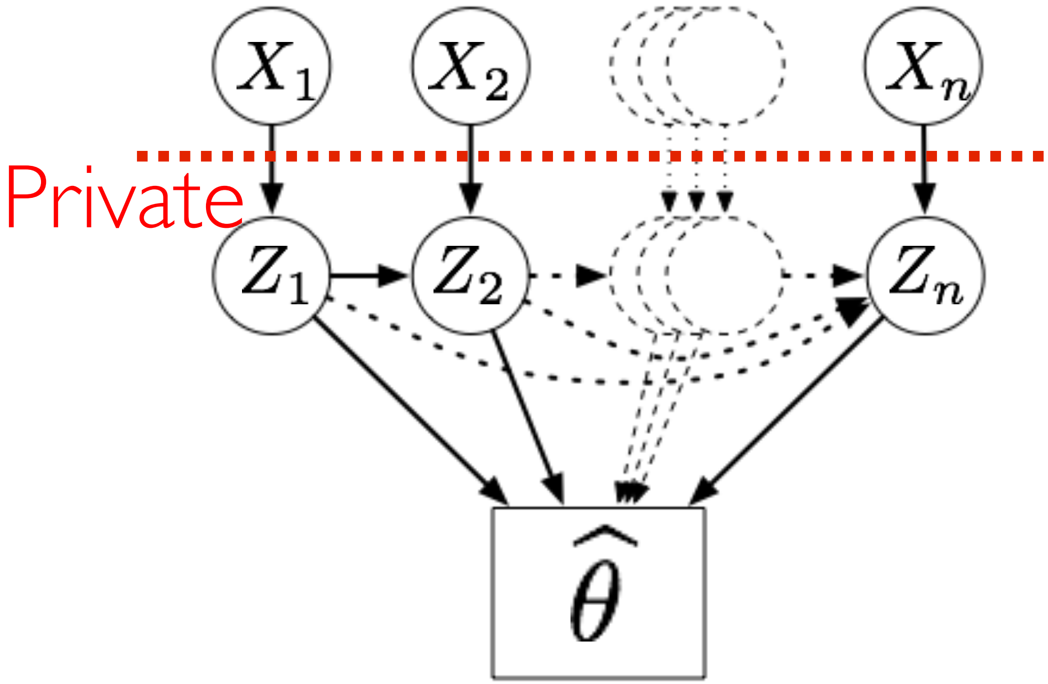
- In the 1930's, Wald laid the foundations of statistical decision theory
- Given a family of distributions $\mathcal{P}$, a parameter $\theta(P)$ for each $P \in \mathcal{P}$, an estimator $\hat{\theta}$, and a loss $l(\hat{\theta}, \theta(P))$, define the risk:

$$R_P(\hat{\theta}) := \mathbb{E}_P\left[l(\hat{\theta}, \theta(P))\right]$$

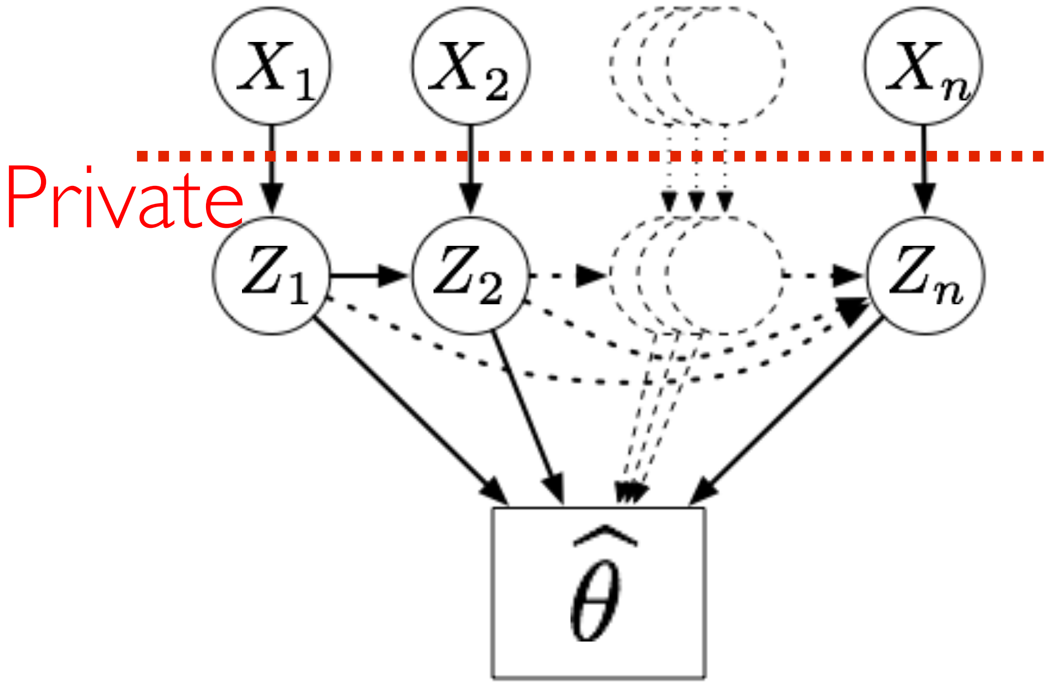
- Minimax principle [Wald, '39, '43]: choose estimator minimizing worst-case risk:

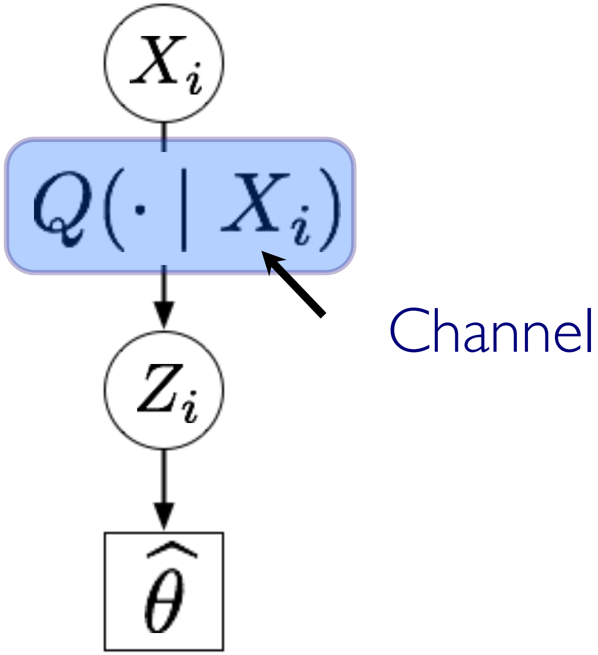
$$\sup_{P \in \mathcal{P}} \mathbb{E}_P\left[l(\hat{\theta}, \theta(P))\right]$$

Local Privacy



Local Privacy





Individuals $i \in \{1, \ldots, n\}$ with private data $X_i \stackrel{\text{iid}}{\sim} P$

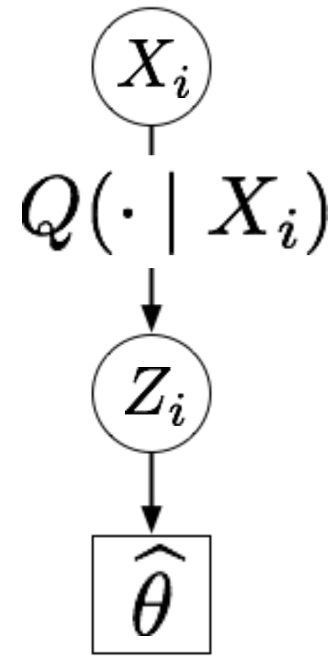
Estimator $Z_1^n \mapsto \widehat{\theta}(Z_1^n)$

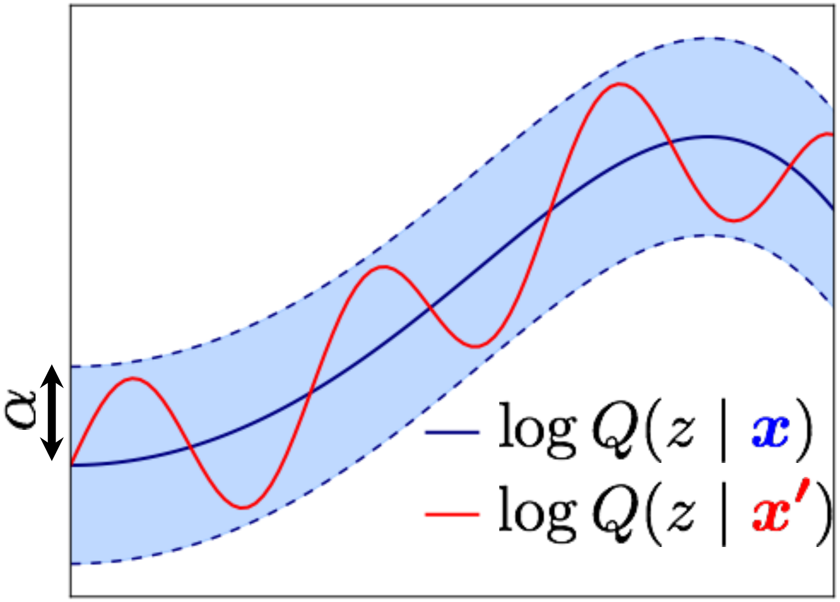
Differential Privacy

Definition: channel Q is α-differentially private if

$$\sup_{S, x \in \mathcal{X}, x' \in \mathcal{X}} \frac{Q(Z \in S \mid x)}{Q(Z \in S \mid x')} \leq \exp(\alpha)$$

[Dwork, McSherry, Nissim, Smith 06]





Given Z, cannot reliably discriminate between x and x'

Private Minimax Risk

- Parameter $\theta(P)$ of distribution
- Family of distributions $\mathcal{P}$
- Loss ℓ measuring error
- Family $\mathcal{Q}_\alpha$ of private channels

α -private Minimax risk

$$\mathfrak{M}_n(\theta(\mathcal{P}), \ell, \alpha) := \inf_{Q \in \mathcal{Q}_\alpha} \inf_{\widehat{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_{P,Q} \left[\ell(\widehat{\theta}(Z_1^n), \theta(P)) \right]$$

Best α-private channel

Minimax risk under privacy constraint

Vignette: Private Mean Estimation

Example: estimate reasons for hospital visits

Patients admitted to hospital for substance abuse

Estimate prevalence of different substances

- 1 Alcohol
- 1 Cocaine
- 0 Heroin
- 0 Cannabis
- 0 LSD
- 0 Amphetamines

Proportions $\boldsymbol{\theta} = \begin{array}{l} \theta_1 = .45 \\ \theta_2 = .32 \\ \theta_3 = .16 \\ \theta_4 = .20 \\ \theta_5 = .00 \\ \theta_6 = .02 \end{array}$

Vignette: Mean Estimation

Consider estimation of mean $\theta(P) := \mathbb{E}_P[X] \in \mathbb{R}^d$, with errors measured in ℓ_∞-norm, for

$$\mathcal{P}_d := \{\text{distributions } P \text{ supported on } [-1,1]^d\}$$

Proposition:

Minimax rate

$$\mathfrak{M}_n(\mathcal{P}_d, \|\cdot\|_\infty) \asymp \min\left\{1, \frac{\sqrt{\log d}}{\sqrt{n}}\right\}$$

(achieved by sample mean)

Vignette: Mean Estimation

Consider estimation of mean $\theta(P) := \mathbb{E}_P[X] \in \mathbb{R}^d$, with errors measured in ℓ_∞-norm, for

$$\mathcal{P}_d := \left\{ \text{distributions } P \text{ supported on } [-1,1]^d \right\}$$

Proposition:

Private minimax rate for $\alpha = O(1)$

$$\mathfrak{M}_n(\mathcal{P}_d, \|\cdot\|_\infty, \alpha) \asymp \min\left\{1, \frac{\sqrt{d \log d}}{\sqrt{n\alpha^2}}\right\}$$

Vignette: Mean Estimation

Consider estimation of mean $\theta(P) := \mathbb{E}_P[X] \in \mathbb{R}^d$, with errors measured in ℓ_∞-norm, for

$$\mathcal{P}_d := \{\text{distributions } P \text{ supported on } [-1,1]^d\}$$

Proposition:

Private minimax rate for $\alpha = O(1)$

$$\mathfrak{M}_n(\mathcal{P}_d, \|\cdot\|_\infty, \alpha) \asymp \min\left\{1, \frac{\sqrt{d \log d}}{\sqrt{n\alpha^2}}\right\}$$

Note: Effective sample size $n \mapsto n\alpha^2/d$

Additional Examples

- Fixed-design regression
- Convex risk minimization
- Multinomial estimation
- Nonparametric density estimation

- Almost always, the effective sample size reduction is:

$$n \mapsto \frac{n\alpha^2}{d}$$

Computation and Inference

- How does inferential quality trade off against classical computational resources such as time and space?

Computation and Inference

- How does inferential quality trade off against classical computational resources such as time and space?
- Hard!

Computation and Inference: Mechanisms and Bounds

- Tradeoffs via convex relaxations
  - linking runtime to convex geometry and risk to convex geometry
- Tradeoffs via concurrency control
  - optimistic concurrency control
- Bounds via optimization oracles
  - number of accesses to a gradient as a surrogate for computation
- Bounds via communication complexity
- Tradeoffs via subsampling
  - bag of little bootstraps, variational consensus Monte Carlo

A Variational Framework for Accelerated Methods in Optimization

with Andre Wibisono and Ashia Wilson

July 12, 2016

Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

- Classical gradient descent:

$$x_{k+1} = x_k - \beta \nabla f(x_k)$$

obtains a convergence rate of $O(1/k)$

Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

- Classical gradient descent:

$$x_{k+1} = x_k - \beta \nabla f(x_k)$$

obtains a convergence rate of $O(1/k)$

- Accelerated gradient descent:

$$\begin{aligned} y_{k+1} &= x_k - \beta \nabla f(x_k) \\ x_{k+1} &= (1 - \lambda_k) y_{k+1} + \lambda_k y_k \end{aligned}$$

obtains the (optimal) convergence rate of $O(1/k^2)$

The acceleration phenomenon

Two classes of algorithms:

- **Gradient methods**
  - Gradient descent, mirror descent, cubic-regularized Newton's method (Nesterov and Polyak '06), etc.
  - Greedy descent methods, relatively well-understood

The acceleration phenomenon

Two classes of algorithms:

- **Gradient methods**
  - Gradient descent, mirror descent, cubic-regularized Newton's method (Nesterov and Polyak '06), etc.
  - Greedy descent methods, relatively well-understood
- **Accelerated methods**
  - Nesterov's accelerated gradient descent, accelerated mirror descent, accelerated cubic-regularized Newton's method (Nesterov '08), etc.
  - Important for both theory (optimal rate for first-order methods) and practice (many extensions: FISTA, stochastic setting, etc.)
  - *Not* descent methods, faster than gradient methods, still mysterious

Accelerated methods

- Analysis using Nesterov's estimate sequence technique
- Common interpretation as "momentum methods" (Euclidean case)

Accelerated methods

- Analysis using Nesterov's estimate sequence technique
- Common interpretation as "momentum methods" (Euclidean case)
- Many proposed explanations:
  - Chebyshev polynomial (Hardt '13)
  - Linear coupling (Allen-Zhu, Orecchia '14)
  - Optimized first-order method (Drori, Teboulle '14; Kim, Fessler '15)
  - Geometric shrinking (Bubeck, Lee, Singh '15)
  - Universal catalyst (Lin, Mairal, Harchaoui '15)
  - ...

Accelerated methods

- Analysis using Nesterov's estimate sequence technique
- Common interpretation as "momentum methods" (Euclidean case)
- Many proposed explanations:
  - Chebyshev polynomial (Hardt '13)
  - Linear coupling (Allen-Zhu, Orecchia '14)
  - Optimized first-order method (Drori, Teboulle '14; Kim, Fessler '15)
  - Geometric shrinking (Bubeck, Lee, Singh '15)
  - Universal catalyst (Lin, Mairal, Harchaoui '15)
  - ...

  But only for strongly convex functions, or first-order methods

Question: What is the underlying mechanism that generates acceleration (including for higher-order methods)?

Accelerated methods: Continuous time perspective

- Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

(and mirror descent is discretization of natural gradient flow)

Accelerated methods: Continuous time perspective

- Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

(and mirror descent is discretization of natural gradient flow)

- Su, Boyd, Candes '14: Continuous time limit of accelerated gradient descent is a second-order ODE

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$$

Accelerated methods: Continuous time perspective

- Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

  (and mirror descent is discretization of natural gradient flow)

- Su, Boyd, Candes '14: Continuous time limit of accelerated gradient descent is a second-order ODE

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$$

- These ODEs are obtained by taking continuous time limits. Is there a deeper generative mechanism?

Our work: A general variational approach to acceleration

A systematic discretization methodology

Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t} \dot{x}, x) - e^{\beta_t} f(x) \right)$$

Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

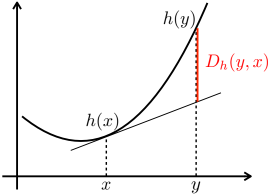
- Function of position x, velocity $\dot{x}$, and time t

Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t} \dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function

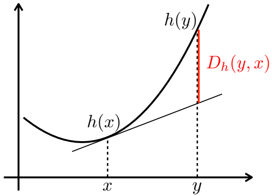


Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function
- f is the convex objective function

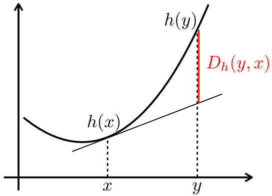


Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t} \dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function
- f is the convex objective function
- $\alpha_t, \beta_t, \gamma_t \in \mathbb{R}$ are arbitrary smooth functions

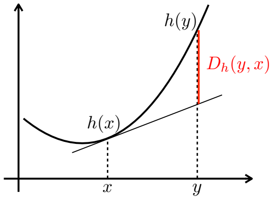


Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t - \alpha_t} \left(\frac{1}{2} \|\dot{x}\|^2 - e^{2\alpha_t + \beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function
- f is the convex objective function
- $\alpha_t, \beta_t, \gamma_t \in \mathbb{R}$ are arbitrary smooth functions
- In Euclidean setting, simplifies to damped Lagrangian

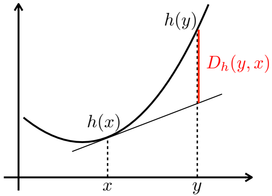


Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t} \dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function
- f is the convex objective function
- $\alpha_t, \beta_t, \gamma_t \in \mathbb{R}$ are arbitrary smooth functions
- In Euclidean setting, simplifies to damped Lagrangian



Ideal scaling conditions:

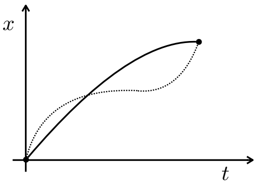
$$\dot{\beta}_t \leq e^{\alpha_t}$$
$$\dot{\gamma}_t = e^{\alpha_t}$$

Bregman Lagrangian

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

Variational problem over curves:

$$\min_X \int \mathcal{L}(X_t, \dot{X}_t, t)\, dt$$



Optimal curve is characterized by **Euler-Lagrange** equation:

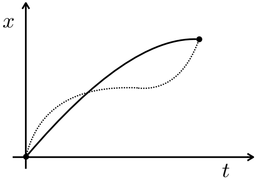
$$\frac{d}{dt}\left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}}(X_t, \dot{X}_t, t) \right\} = \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t)$$

Bregman Lagrangian

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

Variational problem over curves:

$$\min_X \int \mathcal{L}(X_t, \dot{X}_t, t)\, dt$$



Optimal curve is characterized by **Euler-Lagrange** equation:

$$\frac{d}{dt}\left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}}(X_t, \dot{X}_t, t) \right\} = \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t)$$

E-L equation for Bregman Lagrangian under ideal scaling:

$$\ddot{X}_t + (e^{\alpha_t} - \dot{\alpha}_t)\dot{X}_t + e^{2\alpha_t + \beta_t} \left[\nabla^2 h(X_t + e^{-\alpha_t}\dot{X}_t) \right]^{-1} \nabla f(X_t) = 0$$

General convergence rate

Theorem

Theorem Under ideal scaling, the E-L equation has convergence rate

$$f(X_t) - f(x^*) \leq O(e^{-\beta_t})$$

General convergence rate

Theorem

Theorem Under ideal scaling, the E-L equation has convergence rate

$$f(X_t) - f(x^*) \leq O(e^{-\beta_t})$$

Proof. Exhibit a Lyapunov function for the dynamics:

$$\mathcal{E}_t = D_h\left(x^*,\, X_t + e^{-\alpha_t}\dot{X}_t\right) + e^{\beta_t}(f(X_t) - f(x^*))$$

$$\dot{\mathcal{E}}_t = -e^{\alpha_t+\beta_t}D_f(x^*, X_t) + (\dot{\beta}_t - e^{\alpha_t})e^{\beta_t}(f(X_t) - f(x^*)) \;\leq\; 0$$

□

Note: Only requires convexity and differentiability of f, h

Polynomial convergence rate

For $p > 0$, choose parameters:

$$\alpha_t = \log p - \log t$$
$$\beta_t = p \log t + \log C$$
$$\gamma_t = p \log t$$

E-L equation has $O(e^{-\beta_t}) = O(1/t^p)$ convergence rate:

$$\ddot{X}_t + \frac{p+1}{t}\dot{X}_t + Cp^2 t^{p-2}\Big[\nabla^2 h\Big(X_t + \frac{t}{p}\dot{X}_t\Big)\Big]^{-1}\nabla f(X_t) = 0$$

Polynomial convergence rate

For $p > 0$, choose parameters:

$$\alpha_t = \log p - \log t$$
$$\beta_t = p \log t + \log C$$
$$\gamma_t = p \log t$$

E-L equation has $O(e^{-\beta_t}) = O(1/t^p)$ convergence rate:

$$\ddot{X}_t + \frac{p+1}{t}\dot{X}_t + Cp^2 t^{p-2}\left[\nabla^2 h\Big(X_t + \frac{t}{p}\dot{X}_t\Big)\right]^{-1}\nabla f(X_t) = 0$$

For $p = 2$:

- Recover result of Krichene et al with $O(1/t^2)$ convergence rate
- In Euclidean case, recover ODE of Su et al: $\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$

Time dilation property (reparameterizing time)

($p = 2$: accelerated gradient descent)

$$O\left(\frac{1}{t^2}\right): \quad \ddot{X}_t + \frac{3}{t}\dot{X}_t + 4C\Big[\nabla^2 h\Big(X_t + \frac{t}{2}\dot{X}_t\Big)\Big]^{-1}\nabla f(X_t) = 0$$

$$\Big\downarrow \quad \text{speed up time: } Y_t = X_{t^{3/2}}$$

$$O\left(\frac{1}{t^3}\right): \quad \ddot{Y}_t + \frac{4}{t}\dot{Y}_t + 9Ct\Big[\nabla^2 h\Big(Y_t + \frac{t}{3}\dot{Y}_t\Big)\Big]^{-1}\nabla f(Y_t) = 0$$

($p = 3$: accelerated cubic-regularized Newton's method)

Time dilation property (reparameterizing time)

($p = 2$: accelerated gradient descent)

$$O\left(\frac{1}{t^2}\right): \quad \ddot{X}_t + \frac{3}{t}\dot{X}_t + 4C\left[\nabla^2 h\Big(X_t + \frac{t}{2}\dot{X}_t\Big)\right]^{-1}\nabla f(X_t) = 0$$

$$\Big\downarrow \quad \text{speed up time: } Y_t = X_{t^{3/2}}$$

$$O\left(\frac{1}{t^3}\right): \quad \ddot{Y}_t + \frac{4}{t}\dot{Y}_t + 9Ct\left[\nabla^2 h\Big(Y_t + \frac{t}{3}\dot{Y}_t\Big)\right]^{-1}\nabla f(Y_t) = 0$$

($p = 3$: accelerated cubic-regularized Newton's method)

- All accelerated methods are traveling the same curve in space-time at different speeds
- Gradient methods don't have this property
  - From gradient flow to rescaled gradient flow: Replace $\frac{1}{2}\|\cdot\|^2$ by $\frac{1}{p}\|\cdot\|^p$

Time dilation for general Bregman Lagrangian

$$O(e^{-\beta_t}): \quad \text{E-L for Lagrangian } \mathcal{L}_{\alpha,\beta,\gamma}$$

$$\Big\downarrow \quad \text{speed up time: } Y_t = X_{\tau(t)}$$

$$O(e^{-\beta_{\tau(t)}}): \quad \text{E-L for Lagrangian } \mathcal{L}_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$$

where

$$\tilde{\alpha}_t = \alpha_{\tau(t)} + \log \dot{\tau}(t)$$
$$\tilde{\beta}_t = \beta_{\tau(t)}$$
$$\tilde{\gamma}_t = \gamma_{\tau(t)}$$

Time dilation for general Bregman Lagrangian

$$O(e^{-\beta_t}): \quad \text{E-L for Lagrangian } \mathcal{L}_{\alpha,\beta,\gamma}$$

$$\Big\downarrow \quad \text{speed up time: } Y_t = X_{\tau(t)}$$

$$O(e^{-\beta_{\tau(t)}}): \quad \text{E-L for Lagrangian } \mathcal{L}_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$$

where

$$\begin{aligned} \tilde{\alpha}_t &= \alpha_{\tau(t)} + \log \dot{\tau}(t) \\ \tilde{\beta}_t &= \beta_{\tau(t)} \\ \tilde{\gamma}_t &= \gamma_{\tau(t)} \end{aligned}$$

Question: How to discretize E-L while preserving the convergence rate?

Discretizing the dynamics (naive approach)

Write E-L as a system of first-order equations:

$$Z_t = X_t + \frac{t}{p}\dot{X}_t$$

$$\frac{d}{dt}\nabla h(Z_t) = -Cpt^{p-1}\nabla f(X_t)$$

Discretizing the dynamics (naive approach)

Write E-L as a system of first-order equations:

$$Z_t = X_t + \frac{t}{p}\dot{X}_t$$
$$\frac{d}{dt}\nabla h(Z_t) = -Cpt^{p-1}\nabla f(X_t)$$

Euler discretization with time step $\delta > 0$ (i.e., set $x_k = X_t$, $x_{k+1} = X_{t+\delta}$):

$$x_{k+1} = \frac{p}{k+p}z_k + \frac{k}{k+p}x_k$$
$$z_k = \arg\min_z \left\{ Cpk^{(p-1)}\langle \nabla f(x_k), z\rangle + \frac{1}{\epsilon}D_h(z, z_{k-1}) \right\}$$

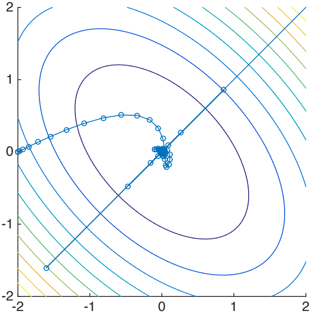
with step size $\epsilon = \delta^p$, and $k^{(p-1)} = k(k+1)\cdots(k+p-2)$ is the rising factorial

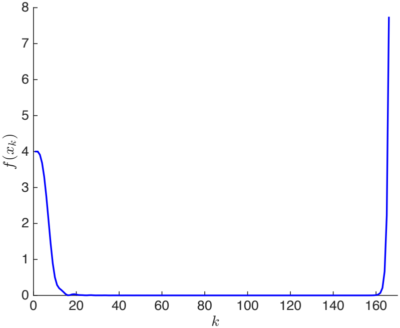
Naive discretization doesn't work

$$x_{k+1} = \frac{p}{k+p} z_k + \frac{k}{k+p} x_k$$

$$z_k = \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(x_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}$$

Cannot obtain a convergence guarantee, and empirically unstable





Modified discretization

Introduce an auxiliary sequence y_k:

$$x_{k+1} = \frac{p}{k+p} z_k + \frac{k}{k+p} y_k$$

$$z_k = \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}$$

Sufficient condition: $\langle \nabla f(y_k), x_k - y_k \rangle \geq M \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|_*^{\frac{p}{p-1}}$

Modified discretization

Introduce an auxiliary sequence y_k:

$$x_{k+1} = \frac{p}{k+p} z_k + \frac{k}{k+p} y_k$$

$$z_k = \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}$$

Sufficient condition: $\langle \nabla f(y_k), x_k - y_k \rangle \geq M \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|_*^{\frac{p}{p-1}}$

Assume h is uniformly convex: $D_h(y, x) \geq \frac{1}{p} \|y - x\|^p$

Theorem

Theorem

$$f(y_k) - f(x^*) \leq O\left(\frac{1}{\epsilon k^p}\right)$$

Note: Matching convergence rates $1/(\epsilon k^p) = 1/(\delta k)^p = 1/t^p$

Proof using generalization of Nesterov's estimate sequence technique

Modified discretization

Introduce an auxiliary sequence y_k:

$$x_{k+1} = \frac{p}{k+p} z_k + \frac{k}{k+p} y_k$$

$$z_k = \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}$$

Sufficient condition: $\langle \nabla f(y_k), x_k - y_k \rangle \geq M \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|_*^{\frac{p}{p-1}} \longleftarrow$ How?

Assume h is uniformly convex: $D_h(y, x) \geq \frac{1}{p} \|y - x\|^p$

Theorem

Theorem

$$f(y_k) - f(x^*) \leq O\left(\frac{1}{\epsilon k^p}\right)$$

Note: Matching convergence rates $1/(\epsilon k^p) = 1/(\delta k)^p = 1/t^p$

Proof using generalization of Nesterov's estimate sequence technique

Higher-order gradient update

Higher-order Taylor approximation of f:

$$f_{p-1}(y; x) = f(x) + \langle \nabla f(x), y - x \rangle + \cdots + \frac{1}{(p-1)!} \nabla^{p-1} f(x)(y - x)^{p-}$$

Higher-order gradient update:

$$y_k = \arg\min_y \left\{ f_{p-1}(y; x_k) + \frac{2}{\epsilon p} \|y - x_k\|^p \right\}$$

Higher-order gradient update

Higher-order Taylor approximation of f:

$$f_{p-1}(y;x) \;=\; f(x) + \langle \nabla f(x), y - x \rangle + \cdots + \frac{1}{(p-1)!} \nabla^{p-1} f(x)(y-x)^{p-}$$

Higher-order gradient update:

$$y_k \;=\; \arg\min_y \left\{ f_{p-1}(y; x_k) + \frac{2}{\epsilon p} \|y - x_k\|^p \right\}$$

Assume f is smooth of order $p-1$:

$$\|\nabla^{p-1} f(y) - \nabla^{p-1} f(x)\|_* \le \frac{1}{\epsilon} \|y - x\|$$

Theorem

Lemma

$$\langle \nabla f(y_k), x_k - y_k \rangle \;\ge\; \frac{1}{4} \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|_*^{\frac{p}{p-1}}$$

Can use this to complete the modified discretization process!

Accelerated higher-order gradient method

$$x_{k+1} = \frac{p}{k+p} z_k + \frac{k}{k+p} y_k$$

$$y_k = \arg\min_y \left\{ f_{p-1}(y; x_k) + \frac{2}{\epsilon p} \|y - x_k\|^p \right\} \quad \leftarrow O\left(\frac{1}{\epsilon k^{p-1}}\right)$$

$$z_k = \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}$$

If $\nabla^{p-1} f$ is $(1/\epsilon)$-Lipschitz and h is uniformly convex of order p, then:

$$f(y_k) - f(x^*) \leq O\left(\frac{1}{\epsilon k^p}\right) \quad \leftarrow \text{ accelerated rate}$$

$p = 2$: Accelerated gradient/mirror descent

$p = 3$: Accelerated cubic-regularized Newton's method (Nesterov '08)

$p \geq 2$: Accelerated higher-order method

Recap: Gradient vs. accelerated methods

How to design dynamics for minimizing a convex function f?

Rescaled gradient flow

$$\dot{X}_t = -\nabla f(X_t) \,/\, \|\nabla f(X_t)\|_*^{\frac{p-2}{p-1}}$$

$$O\left(\frac{1}{t^{p-1}}\right)$$

Higher-order gradient method

$O\left(\frac{1}{\epsilon k^{p-1}}\right)$ when $\nabla^{p-1} f$ is $\frac{1}{\epsilon}$-Lipschitz

matching rate with $\epsilon = \delta^{p-1} \Leftrightarrow \delta = \epsilon^{\frac{1}{p-1}}$

Recap: Gradient vs. accelerated methods

How to design dynamics for minimizing a convex function f?

Rescaled gradient flow	Polynomial Euler-Lagrange equation
$\dot{X}_t = -\nabla f(X_t) \,/\, \|\nabla f(X_t)\|_*^{\frac{p-2}{p-1}}$	$\ddot{X}_t + \frac{p+1}{t}\dot{X}_t + t^{p-2}\left[\nabla^2 h\left(X_t + \frac{t}{p}\dot{X}_t\right)\right]^{-1}\nabla f(X_t) =$
$O\left(\frac{1}{t^{p-1}}\right)$	$O\left(\frac{1}{t^p}\right)$
Higher-order gradient method	**Accelerated higher-order method**
$O\left(\frac{1}{\epsilon k^{p-1}}\right)$ when $\nabla^{p-1} f$ is $\frac{1}{\epsilon}$-Lipschitz	$O\left(\frac{1}{\epsilon k^{p}}\right)$ when $\nabla^{p-1} f$ is $\frac{1}{\epsilon}$-Lipschitz
matching rate with $\epsilon = \delta^{p-1} \Leftrightarrow \delta = \epsilon^{\frac{1}{p-1}}$	matching rate with $\epsilon = \delta^{p} \Leftrightarrow \delta = \epsilon^{\frac{1}{p}}$

Summary: Bregman Lagrangian

- Bregman Lagrangian family with general convergence guarantee

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Polynomial subfamily generates accelerated higher-order methods: $O(1/t^p)$ convergence rate via higher-order smoothness

Summary: Bregman Lagrangian

- Bregman Lagrangian family with general convergence guarantee

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Polynomial subfamily generates accelerated higher-order methods: $O(1/t^p)$ convergence rate via higher-order smoothness
- Exponential subfamily: $O(e^{-ct})$ rate via uniform convexity
- Understand structure and properties of Bregman Lagrangian: Gauge invariance, symmetry, gradient flows as limit points, etc.
- Bregman Hamiltonian:

$$\mathcal{H}(x, p, t) = e^{\alpha_t + \gamma_t} \left(D_{h^*}\left(\nabla h(x) + e^{-\gamma_t} p,\ \nabla h(x)\right) + e^{\beta_t} f(x) \right)$$