Non-Smooth, Non-Finite, and Non-Convex Optimization

Deep Learning Summer School

Mark Schmidt

University of British Columbia

August 2015

Complex-Step Derivative

Using complex number to compute directional derivatives:

- The usual finite-difference approximation of derivative:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

  - Has $O(h^2)$ error from Taylor expansion,

$$f(x+h) = f(x) + hf'(x) + O(h^2),$$

  - But h can't be too small: cancellation in $f(x+h) - f(x)$.

Complex-Step Derivative

Using complex number to compute directional derivatives:

- The usual finite-difference approximation of derivative:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

  - Has $O(h^2)$ error from Taylor expansion,

$$f(x+h) = f(x) + hf'(x) + O(h^2),$$

  - But h can't be too small: cancellation in $f(x+h) - f(x)$.

- For analytic functions, the complex-step derivative uses:

$$f(x+ih) = f(x) + ihf'(x) + O(h^2),$$

that also gives function and derivative to accuracy $O(h^2)$:

$$\text{real}(f(x+ih)) = f(x) + O(h^2), \quad \frac{\text{imag}(f(x+ih))}{h} = f'(x) + O(h^2),$$

Complex-Step Derivative

Using complex number to compute directional derivatives:

- The usual finite-difference approximation of derivative:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

  - Has $O(h^2)$ error from Taylor expansion,

$$f(x+h) = f(x) + hf'(x) + O(h^2),$$

  - But h can't be too small: cancellation in $f(x+h) - f(x)$.
- For analytic functions, the complex-step derivative uses:

$$f(x+ih) = f(x) + ihf'(x) + O(h^2),$$

  that also gives function and derivative to accuracy $O(h^2)$:

$$\text{real}(f(x+ih)) = f(x)+O(h^2), \quad \frac{\text{imag}(f(x+ih))}{h} = f'(x)+O(h^2),$$

  but no cancellation so use tiny h (e.g., 10^{-150} in minFunc).
- First appearance is apparently Squire & Trapp [1998].

"Subgradients" of Non-Convex functions

- Sub-gradient d of function f at x has

$$f(y) \geq f(x) + d^T(y - x),$$

  for all y and x.
- Sub-gradients always exist for reasonable convex functions.

"Subgradients" of Non-Convex functions

- Sub-gradient d of function f at x has

$$f(y) \geq f(x) + d^T(y - x),$$

  for all y and x.
- Sub-gradients always exist for reasonable convex functions.
- Clarke subgradient or generalized gradient d of f at x

$$f(y) \geq f(x) + d^T(y - x) - \sigma\|y - x\|^2,$$

  for some $\sigma > 0$ and all y near x [Clarke, 1975].
- Exist for reasonable non-convex functions.

Convergence Rate of Stochastic Gradient with Constant Step Size

Mark Schmidt
University of British Columbia

September 5, 2014

Abstract

We show that the basic stochastic gradient method applied to a strongly-convex differentiable function with a constant step-size achieves a linear convergence rate (in function value and iterates) up to a constant proportional the step-size (under standard assumptions on the gradient).

1 Overview and Assumptions

We want to minimize $f(x) = \mathbb{E}[f_i(x)]$, where the expectation is taken with respect to i. The most common case is minimizing a finite sum,

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1.1}$$

as in problems like least squares and logistic regression. With use the iteration

$$x^{k+1} = x^k - \alpha f'_{i_k}(x^k),$$

where i_k is sampled uniformly (and step-size α is the step-size). We will assume that f' is L-Lipschitz, f is μ-strongly convex, $\|f'_i(x)\| \leq C$ for all x and i, that the minimizer is x^*, and $0 < \alpha < 1/2\mu$. We will show that

$$\mathbb{E}[f(x^k) - f(x^*)] \leq (1-2\alpha\mu)^k(f(x^0) - f(x^*)) + O(\alpha),$$
$$\mathbb{E}[\left\|x^k - x^*\right\|^2] \leq (1-2\alpha\mu)^k\left\|x^0 - x^*\right\|^2 + O(\alpha),$$

meaning that the function values and iterates converge linearly up to some error level proportional to α. For the special case of (1.1), Proposition 3.4 in the paper of Nedic and Bertsekas ('Convergence Rates of Incremental Subgradient Algorithms', 2000) gives a similar argument/result but here we also consider the function value and we work with the expectation to get rid of the dependence on n.

2 Useful inequalitites

By L-Lipschitz of f', for all x and y we have

$$f(y) \leq f(x) + f'(x)^T(y-x) + \frac{L}{2}\|y-x\|^2.$$

By μ-strong-convexity of f, for all x and y we have

$$f(y) \geq f(x) + f'(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2.$$

Minimizing both sides in terms of y, by setting $y = x - \frac{1}{\mu}f'(x)$ on the right hand side and using the definition of x^* on the left hand side,

$$f(x^*) \geq f(x) - \frac{1}{\mu}\left\|f'(x)\right\|^2 + \frac{1}{2\mu}\left\|f'(x)\right\|^2 = f(x) - \frac{1}{2\mu}\left\|f'(x)\right\|^2.$$

Also by strong-convexity,

$$f'(x)^T(x-x^*) = (f'(x) - f'(x^*))^T(x-x^*) \geq \mu\|x-x^*\|^2.$$

By definition of i_k and f,

$$\mathbb{E}[f'_{i_k}(x^k)] = f'(x^k).$$

Recall the limit of the geometric series,

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r}, \text{ for } |r| < 1.$$

3 Function Value

$$\begin{aligned}
f(x^{k+1}) &\leq f(x^k) + f'(x^k)^T(x^{k+1}-x^k) + \frac{L}{2}\left\|x^{k+1}-x^k\right\|^2 && (x = x^k,\ y = x^{k+1} \text{ in } L\text{-Lipshitz inequality})\\
&= f(x^k) - \alpha f'(x^k)^T f_{i_k}(x^k) + \frac{L\alpha^2}{2}\left\|f'_{i_k}(x^k)\right\|^2 && (\text{eliminate } (x^{k+1}-x^k) \text{ using definition of } x^{k+1})\\
&\leq f(x^k) - \alpha f'(x^k)^T f_{i_k}(x^k) + \frac{L\alpha^2C^2}{2}. && (\text{use } \left\|f'_i(x^k)\right\| \leq C)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[f(x^{k+1}) - f(x^*)] &\leq f(x^k) - f(x^*) - \alpha f'(x^k)\mathbb{E}[f_{i_k}(x^k)] + \frac{L\alpha^2C^2}{2} && (\text{take expectation WRT } i_k, \text{subtract } f(x^*))\\
&\leq f(x^k) - f(x^*) - \alpha\left\|f'(x^k)\right\|^2 + \frac{L\alpha^2C^2}{2} && (\text{use } \mathbb{E}[f'_{i_k}(x^k)] = f'(x^k)))\\
&\leq f(x^k) - f(x^*) - 2\alpha\mu(f(x^k) - f(x^*)) + \frac{L\alpha^2C^2}{2} && (\text{use } \frac{1}{2\mu}\left\|f'(x^k)\right\|^2 \geq f(x^k) - f(x^*))\\
&= (1-2\alpha\mu)(f(x^k) - f(x^*)) + \frac{L\alpha^2C^2}{2}.
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[f(x^k) - f(x^*)] &\leq (1-2\alpha\mu)^k(f(x^0) - f(x^*)) + \sum_{i=0}^{k}(1-2\alpha\mu)^i\frac{L\alpha^2C^2}{2} && (\text{apply recursively, take total expectation})\\
&\leq (1-2\alpha\mu)^k(f(x^0) - f(x^*)) + \sum_{i=0}^{\infty}(1-2\alpha\mu)^i\frac{L\alpha^2C^2}{2} && (\text{extra terms are positive because } \alpha < 1/2\mu)\\
&= (1-2\alpha\mu)^k(f(x^0) - f(x^*)) + \frac{L\alpha C^2}{4\mu}. && (\text{use that } \sum_{i=0}^{\infty}(1-2\alpha\mu)^i = 1/2\alpha\mu)
\end{aligned}$$

4 Iterates

$$\begin{aligned}
\left\|x^{k+1} - x^*\right\|^2 &= \left\|(x^k - \alpha f'_{i_k}(x^k)) - x^*\right\|^2 && (\text{definition of } x^{k+1})\\
&= \left\|x^k - x^*\right\|^2 - 2\alpha f'_{i_k}(x^k)^T(x-x^*) + \alpha^2\left\|f'_{i_k}(x^k)\right\|^2 && (\text{group } (x^k - x^*), \text{expand})\\
&\leq \left\|x^k - x^*\right\|^2 - 2\alpha f'_{i_k}(x^k)^T(x^k - x^*) + \alpha^2C^2. && (\text{use } \left\|f'_i(x^k)\right\| \leq C)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\left\|x^{k+1} - x^*\right\|^2] &\leq \left\|x^k - x^*\right\|^2 - 2\alpha f'(x^k)^T(x^k - x^*) + \alpha^2C^2 && (\text{take expectation WRT } i_k)\\
&\leq \|x - x^*\|^2 - 2\alpha\mu\|x - x^*\| + \alpha^2C^2 && (\text{use } f'(x)^T(x-x^*) \geq \mu\|x-x^*\|^2)\\
&= (1-2\alpha\mu)\left\|x^k - x^*\right\|^2 + \alpha^2C^2
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\left\|x^k - x^*\right\|^2] &\leq (1-2\alpha\mu)^k\left\|x^0 - x^*\right\|^2 + \sum_{i=0}^{k}(1-2\alpha\mu)^i\alpha^2C^2 && (\text{apply recursively, take total expectation})\\
&\leq (1-2\alpha\mu)^k\left\|x^0 - x^*\right\|^2 + \frac{\alpha C^2}{2\mu}. && (\text{as before, use that } \sum_{i=0}^{k}(1-2\alpha\mu)^i \leq 1/2\alpha\mu).
\end{aligned}$$

Stochastic Variance-Reduced Gradient

SVRG algorithm:

- Start with x_0
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$
  - $x^0 = x_s$
  - for $t = 1, 2, \ldots m$
    - Randomly pick $i_t \in \{1, 2, \ldots, N\}$
    - $x^t = x^{t-1} - \alpha_t (f_{i_t}'(x^{t-1}) - f_{i_t}'(x_s) + d_s)$.
  - $x_{s+1} = x^t$ for random $t \in \{1, 2, \ldots, m\}$.

Stochastic Variance-Reduced Gradient

SVRG algorithm:

- Start with x_0
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$
  - $x^0 = x_s$
  - for $t = 1, 2, \ldots m$
    - Randomly pick $i_t \in \{1, 2, \ldots, N\}$
    - $x^t = x^{t-1} - \alpha_t (f_{i_t}'(x^{t-1}) - f_{i_t}'(x_s) + d_s)$.
  - $x_{s+1} = x^t$ for random $t \in \{1, 2, \ldots, m\}$.

Requires 2 gradients per iteration and occasional full passes, but only requires storing d_s and x_s.

Stochastic Variance-Reduced Gradient

SVRG algorithm:

- Start with x_0
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$
  - $x^0 = x_s$
  - for $t = 1, 2, \ldots m$
    - Randomly pick $i_t \in \{1, 2, \ldots, N\}$
    - $x^t = x^{t-1} - \alpha_t (f_{i_t}'(x^{t-1}) - f_{i_t}'(x_s) + d_s)$.
  - $x_{s+1} = x^t$ for random $t \in \{1, 2, \ldots, m\}$.

Requires 2 gradients per iteration and occasional full passes, but only requires storing d_s and x_s.

Practical issues similar to SAG (acceleration versions, automatic step-size/termination, handles sparsity/regularization, non-uniform sampling, mini-batches).

Review of Part 1 and Motivation for Part 2

Part 1: low iteration cost and linear rate in restrictive setting:

- Objective is smooth.
- Objective is a finite sum.
- Objective is strongly-convex.

Part 2: ty to relax these assumptions.

Outline

Motivation: Sparse Regularization

- Recall the regularized empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^P} \frac{1}{N} \sum_{i=1}^{N} L(x, a_i, b_i) \quad + \quad \lambda r(x)$$

$$\text{data fitting term} \quad + \quad \text{regularizer}$$

- Often, regularizer r is used to encourage sparsity pattern in x.

Motivation: Sparse Regularization

- Recall the regularized empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^P} \frac{1}{N} \sum_{i=1}^{N} L(x, a_i, b_i) \quad + \quad \lambda r(x)$$

$$\text{data fitting term} \quad + \quad \text{regularizer}$$

- Often, regularizer r is used to encourage sparsity pattern in x.
- For example, ℓ_1-regularized least squares,

$$\min_x \|Ax - b\|^2 + \lambda \|x\|_1$$

- Regularizes and encourages sparsity in x

Motivation: Sparse Regularization

- Recall the regularized empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^P} \frac{1}{N} \sum_{i=1}^{N} L(x, a_i, b_i) \quad + \quad \lambda r(x)$$

$$\text{data fitting term} \quad + \quad \text{regularizer}$$

- Often, regularizer r is used to encourage sparsity pattern in x.
- For example, ℓ_1-regularized least squares,

$$\min_x \|Ax - b\|^2 + \lambda \|x\|_1$$

- Regularizes and encourages sparsity in x
- The objective is non-differentiable when any $x_i = 0$.
- Subgradient methods are optimal (slow) black-box methods.

Motivation: Sparse Regularization

- Recall the regularized empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^P} \frac{1}{N} \sum_{i=1}^{N} L(x, a_i, b_i) \quad + \quad \lambda r(x)$$

$$\text{data fitting term} \quad + \quad \text{regularizer}$$

- Often, regularizer r is used to encourage sparsity pattern in x.
- For example, ℓ_1-regularized least squares,

$$\min_x \|Ax - b\|^2 + \lambda \|x\|_1$$

- Regularizes and encourages sparsity in x
- The objective is non-differentiable when any $x_i = 0$.
- Subgradient methods are optimal (slow) black-box methods.
- Are there faster methods for specific non-smooth problems?

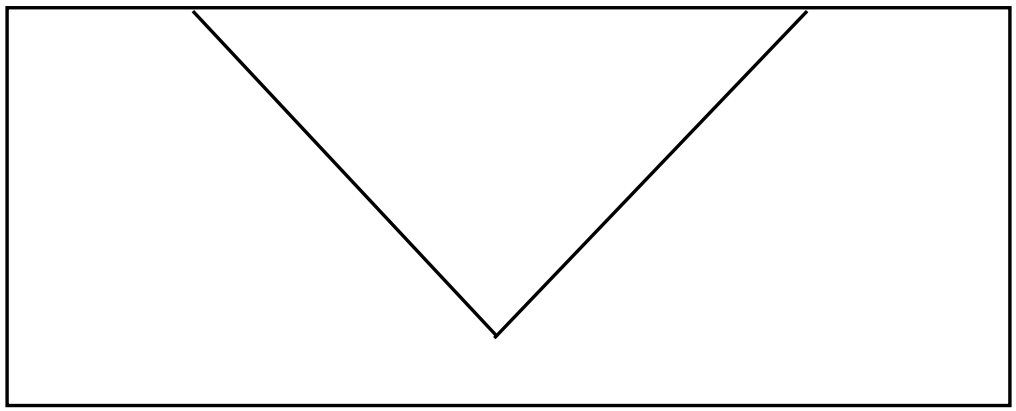
Smoothing Approximations of Non-Smooth Functions

- Smoothing: replace non-smooth f with smooth f_ϵ.
- Apply a fast method for smooth optimization.

Smoothing Approximations of Non-Smooth Functions

- Smoothing: replace non-smooth f with smooth f_ϵ.
- Apply a fast method for smooth optimization.
- Smooth approximation to the absolute value:

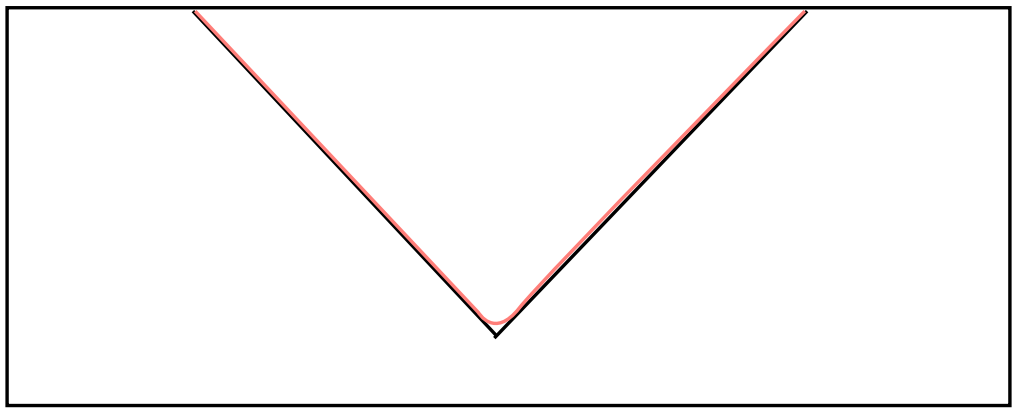
$$|x| \approx \sqrt{x^2 + \nu}.$$



Smoothing Approximations of Non-Smooth Functions

- Smoothing: replace non-smooth f with smooth f_ϵ.
- Apply a fast method for smooth optimization.
- Smooth approximation to the absolute value:

$$|x| \approx \sqrt{x^2 + \nu}.$$



Smoothing Approximations of Non-Smooth Functions

- Smoothing: replace non-smooth f with smooth f_ϵ.
- Apply a fast method for smooth optimization.
- Smooth approximation to the absolute value:

$$|x| \approx \sqrt{x^2 + \nu}.$$

- Smooth approximation to the max function:

$$\max\{a, b\} \approx \log(\exp(a) + \exp(b))$$

- Smooth approximation to the hinge/ReLU loss:

$$\max\{0, x\} \approx \begin{cases} 0 & x \geq 1 \\ 1 - x^2 & t < x < 1 \\ (1-t)^2 + 2(1-t)(t-x) & x \leq t \end{cases}$$

Smoothing Approximations of Non-Smooth Functions

- Smoothing: replace non-smooth f with smooth f_ϵ.
- Apply a fast method for smooth optimization.
- Smooth approximation to the absolute value:

$$|x| \approx \sqrt{x^2 + \nu}.$$

- Smooth approximation to the max function:

$$\max\{a, b\} \approx \log(\exp(a) + \exp(b))$$

- Smooth approximation to the hinge/ReLU loss:

$$\max\{0, x\} \approx \begin{cases} 0 & x \geq 1 \\ 1 - x^2 & t < x < 1 \\ (1-t)^2 + 2(1-t)(t-x) & x \leq t \end{cases}$$

- Generic smoothing strategy: strongly-convex regularization of convex conjugate [Nesterov, 2005].

Discussion of Smoothing Approach

- Nesterov [2005] shows that:
  - Gradient method on smoothed problem has $O(1/\sqrt{t})$ subgradient rate.
  - Accelerated gradient method has faster $O(1/t)$ rate.

Discussion of Smoothing Approach

- Nesterov [2005] shows that:
  - Gradient method on smoothed problem has $O(1/\sqrt{t})$ subgradient rate.
  - Accelerated gradient method has faster $O(1/t)$ rate.
- No results showing improvement in stochastic case.
- In practice:
  - Slowly decrease level of smoothing (often difficult to tune).
  - Use faster algorithms like L-BFGS, SAG, or SVRG.

Discussion of Smoothing Approach

- Nesterov [2005] shows that:
  - Gradient method on smoothed problem has $O(1/\sqrt{t})$ subgradient rate.
  - Accelerated gradient method has faster $O(1/t)$ rate.
- No results showing improvement in stochastic case.
- In practice:
  - Slowly decrease level of smoothing (often difficult to tune).
  - Use faster algorithms like L-BFGS, SAG, or SVRG.
- You can get the $O(1/t)$ rate for $\min_x \max\{f_i(x)\}$ for f_i convex and smooth using *mirror-prox* method [Nemirovski, 2004].
  - See also Chambolle & Pock [2010].

Converting to Constrained Optimization

- Re-write non-smooth problem as constrained problem.

Converting to Constrained Optimization

- Re-write non-smooth problem as constrained problem.
- The problem

$$\min_x f(x) + \lambda \|x\|_1,$$

Converting to Constrained Optimization

- Re-write non-smooth problem as constrained problem.
- The problem

$$\min_x f(x) + \lambda \|x\|_1,$$

is equivalent to the problem

$$\min_{x^+ \geq 0, x^- \geq 0} \quad f(x^+ - x^-) + \lambda \sum_i (x_i^+ + x_i^-),$$

Converting to Constrained Optimization

- Re-write non-smooth problem as constrained problem.
- The problem

$$\min_x f(x) + \lambda \|x\|_1,$$

is equivalent to the problem

$$\min_{x^+ \geq 0, x^- \geq 0} f(x^+ - x^-) + \lambda \sum_i (x_i^+ + x_i^-),$$

or the problems

$$\min_{-y \leq x \leq y} f(x) + \lambda \sum_i y_i, \quad \min_{\|x\|_1 \leq \gamma} f(x) + \lambda\gamma$$

Converting to Constrained Optimization

- Re-write non-smooth problem as constrained problem.
- The problem

$$\min_x f(x) + \lambda \|x\|_1,$$

is equivalent to the problem

$$\min_{x^+ \geq 0, x^- \geq 0} f(x^+ - x^-) + \lambda \sum_i (x_i^+ + x_i^-),$$

or the problems

$$\min_{-y \leq x \leq y} f(x) + \lambda \sum_i y_i, \quad \min_{\|x\|_1 \leq \gamma} f(x) + \lambda\gamma$$

- These are smooth objective with 'simple' constraints.

$$\min_{x \in \mathcal{C}} f(x).$$

Optimization with Simple Constraints

- Recall: gradient descent minimizes quadratic approximation:

$$x^{t+1} = \underset{y}{\text{argmin}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\}.$$

Optimization with Simple Constraints

- Recall: gradient descent minimizes quadratic approximation:

$$x^{t+1} = \underset{y}{\operatorname{argmin}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\}.$$

- Consider minimizing subject to simple constraints:

$$x^{t+1} = \underset{y \in \mathcal{C}}{\operatorname{argmin}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\}.$$

Optimization with Simple Constraints

- Recall: gradient descent minimizes quadratic approximation:

$$x^{t+1} = \operatorname*{argmin}_y \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\}.$$

- Consider minimizing subject to simple constraints:

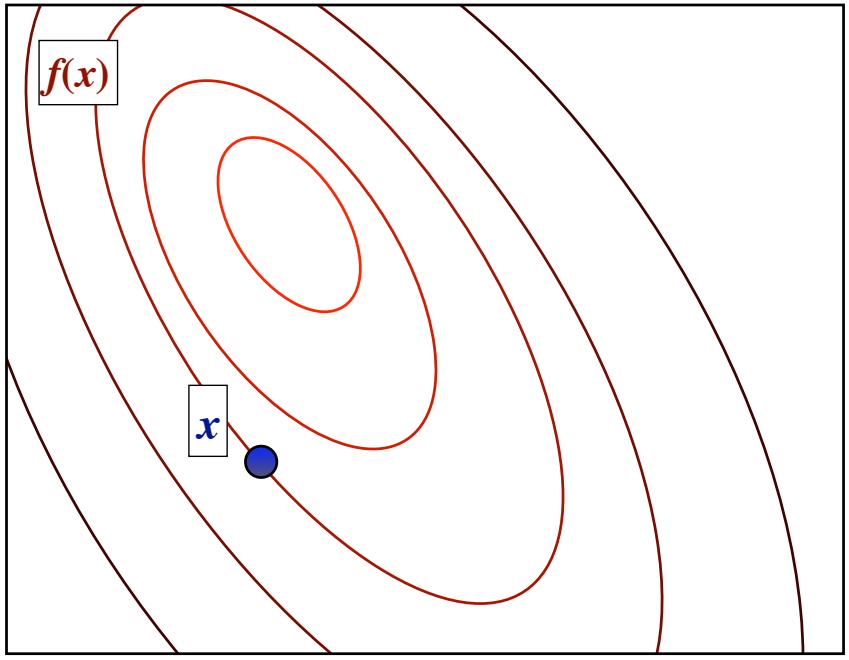
$$x^{t+1} = \operatorname*{argmin}_{y \in \mathcal{C}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\}.$$

- Called projected gradient algorithm:

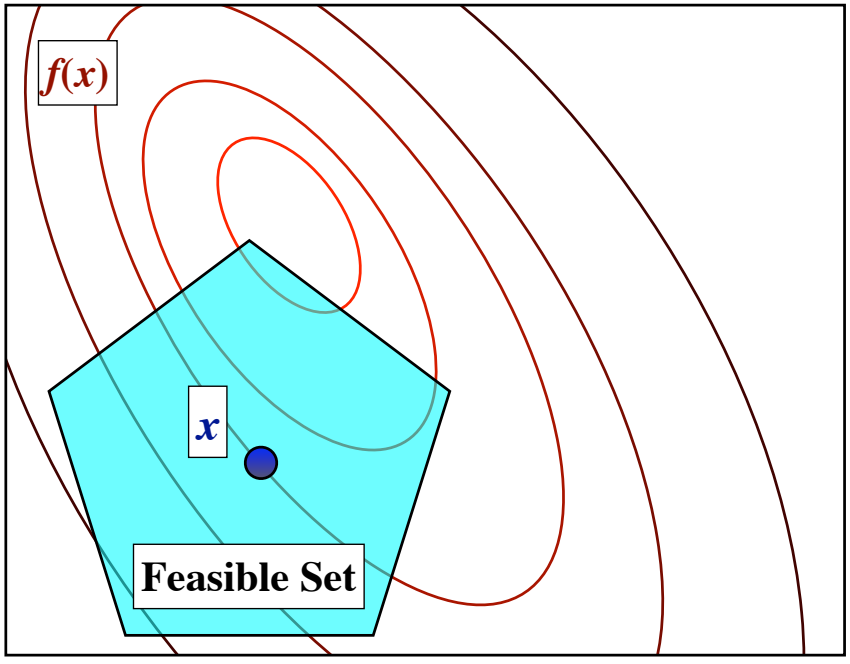
$$x_t^{GD} = x^t - \alpha_t \nabla f(x^t),$$

$$x^{t+1} = \operatorname*{argmin}_{y \in \mathcal{C}} \left\{ \|y - x_t^{GD}\| \right\},$$

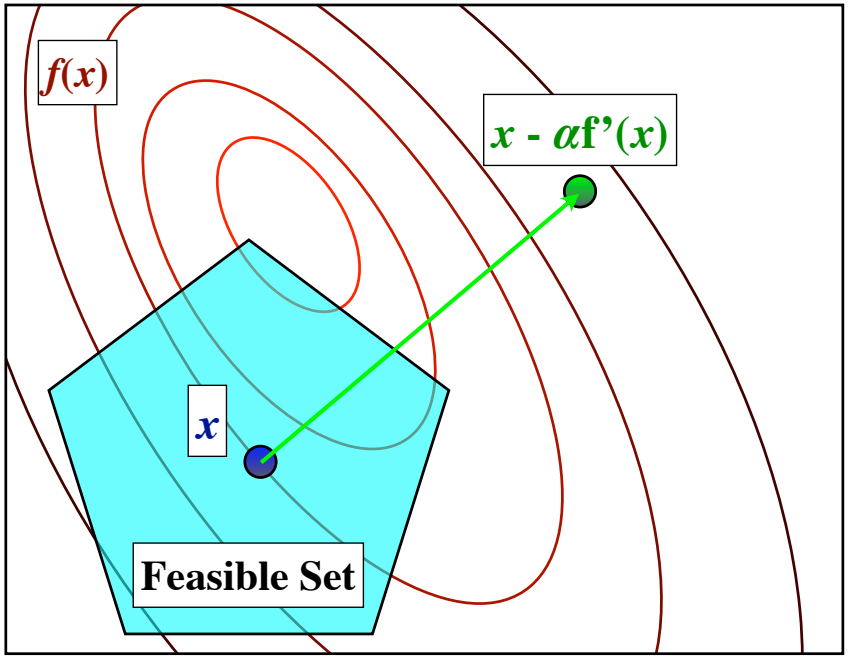
Gradient Projection



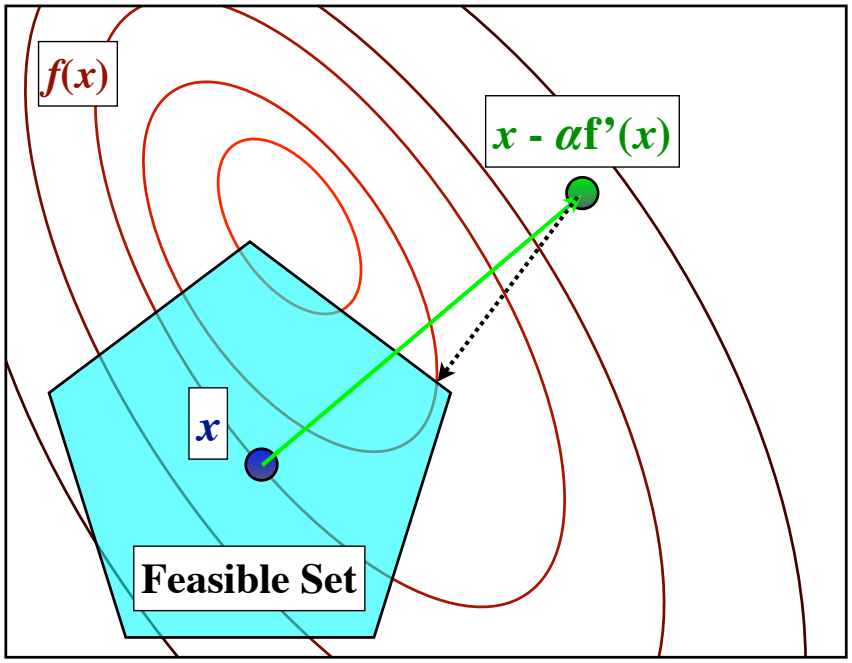
Gradient Projection



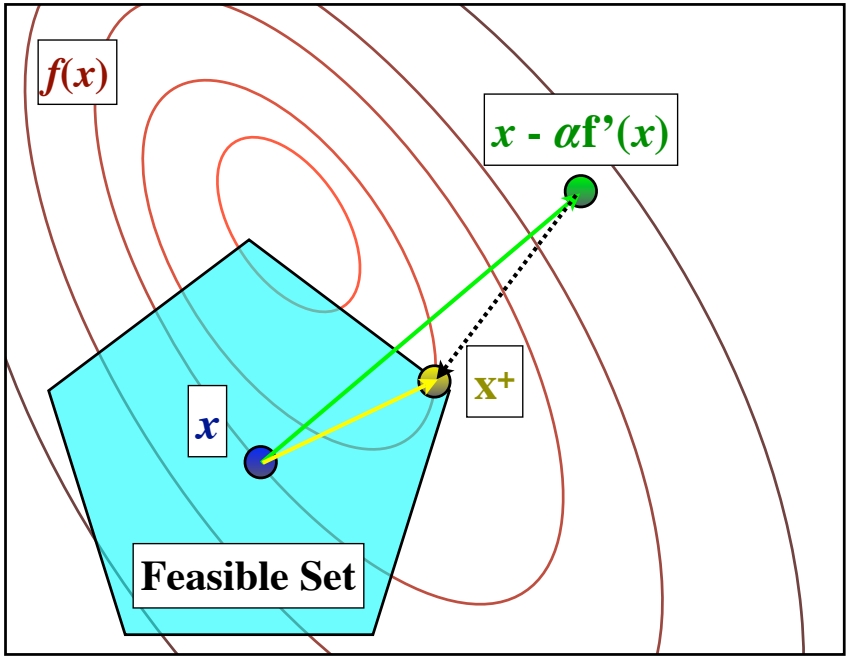
Gradient Projection



Gradient Projection



Gradient Projection



Discussion of Projected Gradient

- Projected gradient has same rate as gradient method!

Discussion of Projected Gradient

- Projected gradient has same rate as gradient method!
- Can do many of the same tricks (i.e. line-search, acceleration, Barzilai-Borwein, SAG, SVRG).

Discussion of Projected Gradient

- Projected gradient has same rate as gradient method!
- Can do many of the same tricks (i.e. line-search, acceleration, Barzilai-Borwein, SAG, SVRG).
- Projected Newton needs expensive projection under $\|\cdot\|_{H_t}$:
  - Two-metric projection methods are efficient Newton-like strategy for bound constraints.
  - Inexact Newton methods allow Newton-like like strategy for optimizing costly functions with simple constraints.

Projection Onto Simple Sets

Projections onto simple sets:

- Bound constraints ($l \leq x \leq u$)
- Small number of linear equalities/inequalities. ($a^T x = b$ or $a^T x \leq b$)
- Norm-balls and norm-cones ($\|x\| \leq \tau$ or $\|x\| \leq x_0$).
- Probability simplex ($x \geq 0, \sum_i x_i = 1$).
- Intersection of disjoint simple sets.

We can solve large instances of problems with these constraints.

Projection Onto Simple Sets

Projections onto simple sets:

- Bound constraints ($l \leq x \leq u$)
- Small number of linear equalities/inequalities. ($a^T x = b$ or $a^T x \leq b$)
- Norm-balls and norm-cones ($\|x\| \leq \tau$ or $\|x\| \leq x_0$).
- Probability simplex ($x \geq 0, \sum_i x_i = 1$).
- Intersection of disjoint simple sets.

We can solve large instances of problems with these constraints.

Intersection of non-disjoint simple sets: Dykstra's algorithm.

Proximal-Gradient Method

- Proximal-gradient generalizes projected-gradient for

$$\min_x f(x) + r(x),$$

where f is smooth but r is a general convex function.

Proximal-Gradient Method

- Proximal-gradient generalizes projected-gradient for

$$\min_x f(x) + r(x),$$

where f is smooth but r is a general convex function.

- Consider the update:

$$x^{t+1} = \operatorname*{argmin}_y \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha} \|y - x^t\|^2 + r(y) \right\}.$$

Proximal-Gradient Method

- Proximal-gradient generalizes projected-gradient for

$$\min_x f(x) + r(x),$$

  where f is smooth but r is a general convex function.

- Consider the update:

$$x^{t+1} = \operatorname*{argmin}_y \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha}\|y - x^t\|^2 + r(y) \right\}.$$

- Applies proximity operator of r to gradient descent on f:

$$x_t^{GD} = x^t - \alpha_t \nabla f(x_t),$$

$$x^{t+1} = \operatorname*{argmin}_y \left\{ \frac{1}{2}\|y - x_t^{GD}\|^2 + \alpha r(y) \right\},$$

- Convergence rates are still the same as for minimizing f.

Proximal Operator, Iterative Soft Thresholding

- The proximal operator is the solution to

$$\text{prox}_r[y] = \underset{x \in \mathbb{R}^P}{\text{argmin}} \ \frac{1}{2}\|x - y\|^2 + r(x).$$

Proximal Operator, Iterative Soft Thresholding

- The proximal operator is the solution to

$$\text{prox}_r[y] = \underset{x \in \mathbb{R}^P}{\text{argmin}} \ \frac{1}{2}\|x - y\|^2 + r(x).$$

- For L1-regularization, we obtain iterative soft-thresholding:

$$x^{t+1} = \text{softThresh}_{\alpha\lambda}[x^t - \alpha \nabla f(x^t)].$$

Proximal Operator, Iterative Soft Thresholding

- The proximal operator is the solution to

$$\text{prox}_r[y] = \underset{x \in \mathbb{R}^P}{\text{argmin}} \ \frac{1}{2}\|x - y\|^2 + r(x).$$

- For L1-regularization, we obtain iterative soft-thresholding:

$$x^{t+1} = \text{softThresh}_{\alpha\lambda}[x^t - \alpha \nabla f(x^t)].$$

- Example with $\lambda = 1$:

Input Threshold Soft-Threshold

$$\begin{bmatrix} 0.6715 \\ -1.2075 \\ 0.7172 \\ 1.6302 \\ 0.4889 \end{bmatrix}$$

Proximal Operator, Iterative Soft Thresholding

- The proximal operator is the solution to

$$\text{prox}_r[y] = \underset{x \in \mathbb{R}^P}{\text{argmin}} \ \frac{1}{2}\|x - y\|^2 + r(x).$$

- For L1-regularization, we obtain iterative soft-thresholding:

$$x^{t+1} = \text{softThresh}_{\alpha\lambda}[x^t - \alpha \nabla f(x^t)].$$

- Example with $\lambda = 1$:

Input	Threshold	Soft-Threshold
$\begin{bmatrix} 0.6715 \\ -1.2075 \\ 0.7172 \\ 1.6302 \\ 0.4889 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1.2075 \\ 0 \\ 1.6302 \\ 0 \end{bmatrix}$	

Proximal Operator, Iterative Soft Thresholding

- The proximal operator is the solution to

$$\text{prox}_r[y] = \underset{x \in \mathbb{R}^P}{\text{argmin}} \ \frac{1}{2}\|x - y\|^2 + r(x).$$

- For L1-regularization, we obtain iterative soft-thresholding:

$$x^{t+1} = \text{softThresh}_{\alpha\lambda}[x^t - \alpha \nabla f(x^t)].$$

- Example with $\lambda = 1$:

Input	Threshold	Soft-Threshold
$\begin{bmatrix} 0.6715 \\ -1.2075 \\ 0.7172 \\ 1.6302 \\ 0.4889 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1.2075 \\ 0 \\ 1.6302 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -0.2075 \\ 0 \\ 0.6302 \\ 0 \end{bmatrix}$

Exact Proximal-Gradient Methods

- For what problems can we apply these methods?

Exact Proximal-Gradient Methods

- For what problems can we apply these methods?
- We can efficiently compute the proximity operator for:
  1. L1-Regularization.
  2. Group ℓ_1-Regularization.

Exact Proximal-Gradient Methods

- For what problems can we apply these methods?
- We can efficiently compute the proximity operator for:
  1. L1-Regularization.
  2. Group ℓ_1-Regularization.
  3. Lower and upper bounds.
  4. Small number of linear constraint.
  5. Probability constraints.
  6. A few other simple regularizers/constraints.

Exact Proximal-Gradient Methods

- For what problems can we apply these methods?
- We can efficiently compute the proximity operator for:
  1. L1-Regularization.
  2. Group ℓ_1-Regularization.
  3. Lower and upper bounds.
  4. Small number of linear constraint.
  5. Probability constraints.
  6. A few other simple regularizers/constraints.
- Can solve these non-smooth/constrained problems as fast as smooth/unconstrained problems!

Exact Proximal-Gradient Methods

- For what problems can we apply these methods?
- We can efficiently compute the proximity operator for:
  1. L1-Regularization.
  2. Group ℓ_1-Regularization.
  3. Lower and upper bounds.
  4. Small number of linear constraint.
  5. Probability constraints.
  6. A few other simple regularizers/constraints.
- Can solve these non-smooth/constrained problems as fast as smooth/unconstrained problems!
- We can again do many of the same tricks (line-search, acceleration, Barzilai-Borwein, two-metric subgradient-projection, inexact proximal operators, inexact proximal Newton, SAG, SVRG).

Alternating Direction Method of Multipliers

- Alernating direction method of multipliers (ADMM) solves:

$$\min_{Ax+By=c} f(x) + r(y).$$

- Alternate between prox-like operators with respect to f and r.

Alternating Direction Method of Multipliers

- Alernating direction method of multipliers (ADMM) solves:

$$\min_{Ax+By=c} f(x) + r(y).$$

- Alternate between prox-like operators with respect to f and r.
- Can introduce constraints to convert to this form:

$$\min_x f(Ax) + r(x) \quad \Leftrightarrow \quad \min_{x=Ay} f(x) + r(y),$$

Alternating Direction Method of Multipliers

- Alernating direction method of multipliers (ADMM) solves:

$$\min_{Ax+By=c} f(x) + r(y).$$

- Alternate between prox-like operators with respect to f and r.
- Can introduce constraints to convert to this form:

$$\min_x f(Ax) + r(x) \quad \Leftrightarrow \quad \min_{x=Ay} f(x) + r(y),$$

$$\min_x f(x) + r(Bx) \quad \Leftrightarrow \quad \min_{y=Bx} f(x) + r(y).$$

Alternating Direction Method of Multipliers

- Alernating direction method of multipliers (ADMM) solves:

$$\min_{Ax+By=c} f(x) + r(y).$$

- Alternate between prox-like operators with respect to f and r.
- Can introduce constraints to convert to this form:

$$\min_x f(Ax) + r(x) \quad \Leftrightarrow \quad \min_{x=Ay} f(x) + r(y),$$

$$\min_x f(x) + r(Bx) \quad \Leftrightarrow \quad \min_{y=Bx} f(x) + r(y).$$

- If prox can not be computed exactly: Linearized ADMM.

Frank-Wolfe Method

- In some cases the projected gradient step

$$x^{t+1} = \operatorname*{argmin}_{y \in \mathcal{C}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\},$$

may be hard to compute.

Frank-Wolfe Method

- In some cases the projected gradient step

$$x^{t+1} = \underset{y \in \mathcal{C}}{\operatorname{argmin}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) + \frac{1}{2\alpha_t} \|y - x^t\|^2 \right\},$$

  may be hard to compute.

- Frank-Wolfe method simply uses:

$$x^{t+1} = \underset{y \in \mathcal{C}}{\operatorname{argmin}} \left\{ f(x^t) + \nabla f(x^t)^T (y - x^t) \right\},$$

  requires compact $\mathcal{C}$, takes convex combination of x^t and x^{t+1}.

- $O(1/t)$ rate for smooth convex objectives, some linear convergence results for strongly-convex [Jaggi, 2013].

Summary

- No black-box method can beat subgradient methods
- For most objectives, you can beat subgradient methods.

Summary

- No black-box method can beat subgradient methods
- For most objectives, you can beat subgradient methods.
- You just need a long list of tricks:
  - Smoothing.
  - Chambolle-Pock.
  - Projected-gradient.
  - Two-metric projection.
  - Proximal-gradient.
  - Proximal-Newton.
  - ADMM
  - Frank-Wolfe.
  - Mirror descent.
  - Incremental surrogate optimization.
  - Solving smooth dual.

Outline

Stochastic vs. Deterministic for Stochastic Objectives

- Consider smooth/strongly-convex stochastic objectives,

$$\min_{x \in \mathbb{R}^D} \mathbb{E}[f_i(x)],$$

including the generalization error in machine learning.

Stochastic vs. Deterministic for Stochastic Objectives

- Consider smooth/strongly-convex stochastic objectives,

$$\min_{x \in \mathbb{R}^D} \mathbb{E}[f_i(x)],$$

  including the generalization error in machine learning.
- Error ϵ has two parts [Bottou & Bousquet, 2007]:

$$\epsilon = (\text{optimization error}) + (\text{estimation error}).$$

(for generalization error, also have model error)

Stochastic vs. Deterministic for Stochastic Objectives

- Consider smooth/strongly-convex stochastic objectives,

$$\min_{x \in \mathbb{R}^D} \mathbb{E}[f_i(x)],$$

  including the generalization error in machine learning.
- Error ϵ has two parts [Bottou & Bousquet, 2007]:

$$\epsilon = (\text{optimization error}) + (\text{estimation error}).$$

  (for generalization error, also have model error)
- Consider two strategies:
  - Generate t samples, then minimize exactly (ERM):
    - Optimization error $= 0$.
    - Estimation error $= \tilde{O}(1/t)$.

Stochastic vs. Deterministic for Stochastic Objectives

- Consider smooth/strongly-convex stochastic objectives,

$$\min_{x \in \mathbb{R}^D} \mathbb{E}[f_i(x)],$$

  including the generalization error in machine learning.
- Error ϵ has two parts [Bottou & Bousquet, 2007]:

$$\epsilon = (\text{optimization error}) + (\text{estimation error}).$$

  (for generalization error, also have model error)
- Consider two strategies:
  - Generate t samples, then minimize exactly (ERM):
    - Optimization error $= 0$.
    - Estimation error $= \tilde{O}(1/t)$.
  - Or just applying stochastic gradient as we go:
    - Optimization error $= O(1/t)$.
    - Estimation error $= \tilde{O}(1/t)$.

Stochastic vs. Deterministic for Stochastic Objectives

- So just go through your data once with stochastic gradient?

Stochastic vs. Deterministic for Stochastic Objectives

- So just go through your data once with stochastic gradient?
- "overwhelming empirical evidence shows that for almost all actual data, the ERM *is* better. However, we have no understanding of why this happens"

[Srebro & Sridharan, 2011]

Stochastic vs. Deterministic for Stochastic Objectives

- So just go through your data once with stochastic gradient?
- "overwhelming empirical evidence shows that for almost all actual data, the ERM *is* better. However, we have no understanding of why this happens"

  [Srebro & Sridharan, 2011]
- Constants matter in learning:
  - SG optimal in terms of sample size.
  - But not other quantities: L, μ, x^0.
  - We care about multiplying test error by 2!

Stochastic vs. Deterministic for Stochastic Objectives

- So just go through your data once with stochastic gradient?
- "overwhelming empirical evidence shows that for almost all actual data, the ERM *is* better. However, we have no understanding of why this happens"

  [Srebro & Sridharan, 2011]
- Constants matter in learning:
  - SG optimal in terms of sample size.
  - But not other quantities: L, μ, x^0.
  - We care about multiplying test error by 2!
- Growing-batch deterministic methods [Byrd et al., 2011].
- Or take t iterations of SAG on fixed $N < t$ samples.
  - Optimization accuracy decreases to $O(\rho^t)$.
  - Estimation accuracy increases to $\tilde{O}(1/N)$.

Stochastic vs. Deterministic for Stochastic Objectives

- So just go through your data once with stochastic gradient?
- "overwhelming empirical evidence shows that for almost all actual data, the ERM *is* better. However, we have no understanding of why this happens"

  [Srebro & Sridharan, 2011]
- Constants matter in learning:
  - SG optimal in terms of sample size.
  - But not other quantities: L, μ, x^0.
  - We care about multiplying test error by 2!
- Growing-batch deterministic methods [Byrd et al., 2011].
- Or take t iterations of SAG on fixed $N < t$ samples.
  - Optimization accuracy decreases to $O(\rho^t)$.
  - Estimation accuracy increases to $\tilde{O}(1/N)$.
- SAG obtains better bounds for difficult optimization problems.

Streaming SVRG

Streaming SVRG algorithm [Frostig et al., 2015]:

- Start with x_0 and initial sample size N

Streaming SVRG

Streaming SVRG algorithm [Frostig et al., 2015]:

- Start with x_0 and initial sample size N
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$ for N fresh samples.
  - $x^0 = x_s$

Streaming SVRG

Streaming SVRG algorithm [Frostig et al., 2015]:

- Start with x_0 and initial sample size N
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$ for N fresh samples.
  - $x^0 = x_s$
  - for $t = 1, 2, \ldots m$
    - Randomly pick 1 fresh sample.
    - $x^t = x^{t-1} - \alpha_t (f_{i_t}'(x^{t-1}) - f_{i_t}'(x_s) + d_s)$.
  - $x_{s+1} = x^t$ for random $t \in \{1, 2, \ldots, m\}$.
  - Increase samples size N.

Streaming SVRG

Streaming SVRG algorithm [Frostig et al., 2015]:

- Start with x_0 and initial sample size N
- for $s = 0, 1, 2 \ldots$
  - $d_s = \frac{1}{N} \sum_{i=1}^{N} f_i'(x_s)$ for N fresh samples.
  - $x^0 = x_s$
  - for $t = 1, 2, \ldots m$
    - Randomly pick 1 fresh sample.
    - $x^t = x^{t-1} - \alpha_t (f_{i_t}'(x^{t-1}) - f_{i_t}'(x_s) + d_s)$.
  - $x_{s+1} = x^t$ for random $t \in \{1, 2, \ldots, m\}$.
  - Increase samples size N.

- Streaming SVRG is optimal in non-asymptotic regime.
- Same variance as ERM (only true for avg(SG) asymptotically).
- Second-order methods are not necessary.

Constant-Step SG under Strong Assumptions

- We can beat $O(1/t)$ under stronger assumptions.

Constant-Step SG under Strong Assumptions

- We can beat $O(1/t)$ under stronger assumptions.
- E.g., Schmidt & Le Roux [2013],

$$\|f_i'(x)\| \leq B\|f'(x)\|.$$

- Crazy assumption: assumes x^* minimizes f_i.

Constant-Step SG under Strong Assumptions

- We can beat $O(1/t)$ under stronger assumptions.
- E.g., Schmidt & Le Roux [2013],

$$\|f_i'(x)\| \leq B\|f'(x)\|.$$

- Crazy assumption: assumes x^* minimizes f_i.
- With $\alpha_t = \frac{1}{LB^2}$, stochastic gradient has

$$\mathbb{E}[f(x^t)] - f(x^*) \leq \left(1 - \frac{\mu}{LB^2}\right)^t [f(x^0) - f(x^*)].$$

- If you expect to over-fit, maybe constant α_t is enough?

Online Convex Optimization

- What if data is not IID?

Online Convex Optimization

- What if data is not IID?
- Addressed by online convex optimization (OCO) framework: [Zinkevich, 2003]
  - At time t, make a prediction x^t.

Online Convex Optimization

- What if data is not IID?
- Addressed by online convex optimization (OCO) framework: [Zinkevich, 2003]
  - At time t, make a prediction x^t.
  - Receive arbitrary convex loss f_t.
- OCO analyzes regret,

$$\sum_{k=1}^{t} f_t(x^t) - f_t(x^*),$$

comparing vs. best fixed x^* for any sequence $\{f_t\}$.

Online Convex Optimization

- What if data is not IID?
- Addressed by online convex optimization (OCO) framework: [Zinkevich, 2003]
  - At time t, make a prediction x^t.
  - Receive arbitrary convex loss f_t.
- OCO analyzes regret,

$$\sum_{k=1}^{t} f_t(x^t) - f_t(x^*),$$

  comparing vs. best fixed x^* for any sequence $\{f_t\}$.
- SG-style methods achieve optimal $O(\sqrt{t})$ regret.
- Strongly-convex losses: $O(\log(t))$ regret [Hazan et al., 2006].

Online Convex Optimization

- What if data is not IID?
- Addressed by online convex optimization (OCO) framework: [Zinkevich, 2003]
  - At time t, make a prediction x^t.
  - Receive arbitrary convex loss f_t.
- OCO analyzes regret,

$$\sum_{k=1}^{t} f_t(x^t) - f_t(x^*),$$

  comparing vs. best fixed x^* for any sequence $\{f_t\}$.
- SG-style methods achieve optimal $O(\sqrt{t})$ regret.
- Strongly-convex losses: $O(\log(t))$ regret [Hazan et al., 2006].
- Variants exist see features first [Cesa-Bianchi et al., 1993.
- Bandit setting: no gradients.

Outline

Two Classic Perspectives of Non-Convex Optimization

Two Classic Perspectives of Non-Convex Optimization

- Local non-convex optimization:
  - Apply method with good properties for convex functions.
  - First phase is getting near minimizer.
  - Second phase applies rates from convex optimization.

Two Classic Perspectives of Non-Convex Optimization

- Local non-convex optimization:
  - Apply method with good properties for convex functions.
  - First phase is getting near minimizer.
  - Second phase applies rates from convex optimization.
  - But how long does the first phase take?

Two Classic Perspectives of Non-Convex Optimization

- Local non-convex optimization:
  - Apply method with good properties for convex functions.
  - First phase is getting near minimizer.
  - Second phase applies rates from convex optimization.
  - But how long does the first phase take?
- Global non-convex optimization:
  - Search for global min for general function class.
  - E.g., search over a sucessively-refined grid.
  - Optimal rate for Lipschitz functions is $O(1/\epsilon^{1/D})$.

Two Classic Perspectives of Non-Convex Optimization

- Local non-convex optimization:
  - Apply method with good properties for convex functions.
  - First phase is getting near minimizer.
  - Second phase applies rates from convex optimization.
  - But how long does the first phase take?
- Global non-convex optimization:
  - Search for global min for general function class.
  - E.g., search over a sucessively-refined grid.
  - Optimal rate for Lipschitz functions is $O(1/\epsilon^{1/D})$.
  - Can only solve low-dimensional problems.
- We'll go over recent local, global, and hybrid results..

Strong Property: Expanding the Second Phase

- Linear convergence proofs usually assume strong-convexity

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Strong Property: Expanding the Second Phase

- Linear convergence proofs usually assume strong-convexity

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

- Which implies the inequality often used in the proofs,

$$\|\nabla f(x)\|^2 \geq 2\mu[f(x) - f^*].$$

Strong Property: Expanding the Second Phase

- Linear convergence proofs usually assume strong-convexity

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

- Which implies the inequality often used in the proofs,

$$\|\nabla f(x)\|^2 \geq 2\mu[f(x) - f^*].$$

- A bunch of weaker assumptions imply this inequality,
  - Essentially strong-convexity.
  - Optimal strong-convexity.
  - Restricted secant inequality.
  - Etc.

Strong Property: Expanding the Second Phase

- Linear convergence proofs usually assume strong-convexity

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

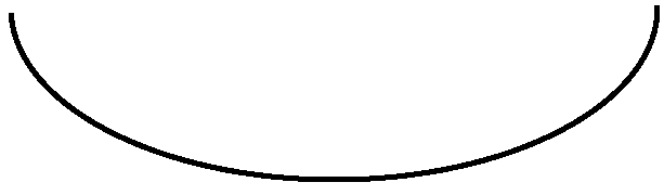
- Which implies the inequality often used in the proofs,

$$\|\nabla f(x)\|^2 \geq 2\mu[f(x) - f^*].$$

- A bunch of weaker assumptions imply this inequality,
  - Essentially strong-convexity.
  - Optimal strong-convexity.
  - Restricted secant inequality.
  - Etc.
- Strong property: Just assume the inequality holds.
  - Special case of Łojasiewicz [1963] inequality.
  - Also introduced in Polyak [1963].
  - Weaker than all the above conditions.
  - Does not imply solution is unique.
    - Holds for $f(Ax)$ with f strongly-convex and $rank(A) \geq 1$.
  - Does not imply convexity.

Global Linear Convergence with the Strong Property

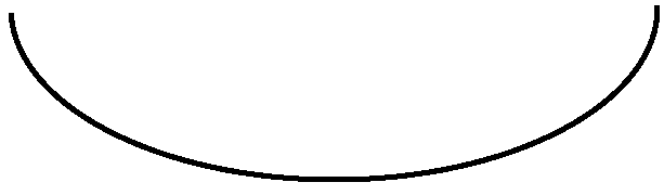
Function satisfying the strong-convexity property:



(unique optimum, convex, growing faster than linear)

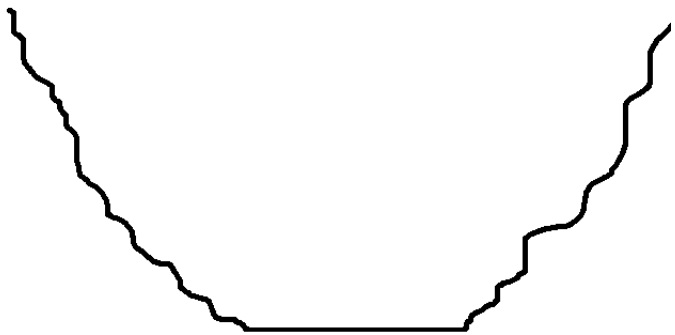
Global Linear Convergence with the Strong Property

Function satisfying the strong-convexity property:



(unique optimum, convex, growing faster than linear)

Function satisfying the strong property:



- Linear convergence rate for this non-convex function.
- Second phase of local solvers is larger than we thought.

General Global Non-Convex Rates?

- For strongly-convex smooth functions, we have

$$\|\nabla f(x^t)\|^2 = O(\rho^t), \quad f(x^t) - f(x^*) = O(\rho^t), \quad \|x_t - x_*\| = O(\rho^t).$$

- For convex smooth functions, we have

$$\|\nabla f(x^t)\|^2 = O(1/t), \quad f(x^t) - f(x^*) = O(1/t).$$

General Global Non-Convex Rates?

- For strongly-convex smooth functions, we have

$$\|\nabla f(x^t)\|^2 = O(\rho^t), \quad f(x^t) - f(x^*) = O(\rho^t), \quad \|x_t - x_*\| = O(\rho^t).$$

- For convex smooth functions, we have

$$\|\nabla f(x^t)\|^2 = O(1/t), \quad f(x^t) - f(x^*) = O(1/t).$$

- For non-convex smooth functions, we have

$$\min_k \|\nabla f(x^k)\|^2 = O(1/t).$$

[Ghadimi & Lan, 2013].

Escaping Saddle Points

- Ghadimi & Lan type of rates could be good or bad news:
  - No dimension dependence (way faster than grid-search).
  - But gives up on optimality (e.g., approximate saddle points).

Escaping Saddle Points

- Ghadimi & Lan type of rates could be good or bad news:
  - No dimension dependence (way faster than grid-search).
  - But gives up on optimality (e.g., approximate saddle points).
- Escaping from saddle points:
  - Classical: trust-region methods allow negative eigenvalues.
  - Modify eigenvalues in Newton's method [Dauphin et al., 2014].
  - Add random noise to stochastic gradient [Ge et al., 2015].

Escaping Saddle Points

- Ghadimi & Lan type of rates could be good or bad news:
  - No dimension dependence (way faster than grid-search).
  - But gives up on optimality (e.g., approximate saddle points).
- Escaping from saddle points:
  - Classical: trust-region methods allow negative eigenvalues.
  - Modify eigenvalues in Newton's method [Dauphin et al., 2014].
  - Add random noise to stochastic gradient [Ge et al., 2015].
  - Cubic regularization of Newton [Nesterov & Polyak, 2006],

$$x^{k+1} = \min_d \left\{ f(x^k) + \langle \nabla f(x^k), d \rangle + \frac{1}{2} d^T \nabla^2 f(x^k) d + \frac{L}{6} \|d\|^3 \right\},$$

  if within ball of saddle point then next step:
    - Moves outside of ball.
    - Has lower objective than saddle-point.

Globally-Optimal Methods for Matrix Problems

Globally-Optimal Methods for Matrix Problems

- Classic: principal component analysis (PCA)

$$\max_{W^T W = I} \|X^T W\|_F^2,$$

and rank-constrained version.
Shamir [2015] gives SAG/SVRG rates for PCA.

Globally-Optimal Methods for Matrix Problems

- Classic: principal component analysis (PCA)

$$\max_{W^T W = I} \|X^T W\|_F^2,$$

  and rank-constrained version.
  Shamir [2015] gives SAG/SVRG rates for PCA.

- Burer & Monteiro [2004] consider SDP re-parameterization

$$\min_{\{X | X \succeq 0, \text{rank}(X) \leq k\}} f(X) \Rightarrow \min_V f(VV^T),$$

  and show does not introduce spurious local minimum.

Globally-Optimal Methods for Matrix Problems

- Classic: principal component analysis (PCA)

$$\max_{W^T W = I} \|X^T W\|_F^2,$$

and rank-constrained version.
Shamir [2015] gives SAG/SVRG rates for PCA.

- Burer & Monteiro [2004] consider SDP re-parameterization

$$\min_{\{X | X \succeq 0, \text{rank}(X) \leq k\}} f(X) \Rightarrow \min_V f(VV^T),$$

and show does not introduce spurious local minimum.

- De Sa et al. [2015]: For class of non-convex problems of the form

$$\min_Y \mathbb{E}[\|A - VV^T\|_F^2].$$

random initialization leads to global optimum.

Globally-Optimal Methods for Matrix Problems

- Classic: principal component analysis (PCA)

$$\max_{W^T W = I} \|X^T W\|_F^2,$$

and rank-constrained version.
Shamir [2015] gives SAG/SVRG rates for PCA.

- Burer & Monteiro [2004] consider SDP re-parameterization

$$\min_{\{X | X \succeq 0, \text{rank}(X) \leq k\}} f(X) \Rightarrow \min_V f(VV^T),$$

and show does not introduce spurious local minimum.

- De Sa et al. [2015]: For class of non-convex problems of the form

$$\min_Y \mathbb{E}[\|A - VV^T\|_F^2].$$

random initialization leads to global optimum.

- Under certain assumptions, can solve UV^T dictionary learning and phase retrieval problems [Agarwal et al., 2014, Candes et al., 2015].
- Certain latent variable problems like training HMMs can be solved via SVD and tensor-decomposition methods [Hsu et al., 2012, Anankumar et al, 2014].

Convex Relaxations/Representations

- Convex relaxations approximate non-convex with convex:
  - Convex relaxations exist for neural nets.
    [Bengio et al., 2005, Aslan et al., 2015].
  - But may solve restricted problem or be a bad approximation.

Convex Relaxations/Representations

- Convex relaxations approximate non-convex with convex:
  - Convex relaxations exist for neural nets.
    [Bengio et al., 2005, Aslan et al., 2015].
  - But may solve restricted problem or be a bad approximation.
- Can solve convex dual:
  - Strong-duality holds for some non-convex problems.
  - Sometimes dual has nicer properties.
  - Efficiently representation/calculation of neural network dual?

Convex Relaxations/Representations

- Convex relaxations approximate non-convex with convex:
  - Convex relaxations exist for neural nets.
    [Bengio et al., 2005, Aslan et al., 2015].
  - But may solve restricted problem or be a bad approximation.
- Can solve convex dual:
  - Strong-duality holds for some non-convex problems.
  - Sometimes dual has nicer properties.
  - Efficiently representation/calculation of neural network dual?
- Exact convex re-formulations of non-convex problems:
  - Laserre [2001].
  - But the size may be enormous.

General Non-Convex Rates

Grid-search is optimal, but can be beaten:

- Convergence rate of Bayesian optimization [Bull, 2011]:
  - Slower than grid-search with low level of smoothness.
  - Faster than grid-search with high level of smoothness:
    - Improves error from $O(1/t^{1/d})$ to $O(1/t^{\nu/d})$.

General Non-Convex Rates

Grid-search is optimal, but can be beaten:

- Convergence rate of Bayesian optimization [Bull, 2011]:
  - Slower than grid-search with low level of smoothness.
  - Faster than grid-search with high level of smoothness:
    - Improves error from $O(1/t^{1/d})$ to $O(1/t^{\nu/d})$.
- Regret bounds for Bayesian optimization:
  - Exponential scaling with dimensionality [Srinivas et al., 2010].
  - Better under additive assumption [Kandasamy et al., 2015].

General Non-Convex Rates

Grid-search is optimal, but can be beaten:

- Convergence rate of Bayesian optimization [Bull, 2011]:
  - Slower than grid-search with low level of smoothness.
  - Faster than grid-search with high level of smoothness:
    - Improves error from $O(1/t^{1/d})$ to $O(1/t^{\nu/d})$.
- Regret bounds for Bayesian optimization:
  - Exponential scaling with dimensionality [Srinivas et al., 2010].
  - Better under additive assumption [Kandasamy et al., 2015].
- Other known faster-than-grid-search rates:
  - Simulated annealing under complicated non-singular assumption [Tikhomirov, 2010].
  - Particle filtering can improve under certain conditions [Crisan & Doucet, 2002].
  - Graduated Non-Convexity for σ-nice functions [Hazan et al., 2014].

Summary

Summary:

- Part 1: Can solve constrained/non-smooth efficiently with a variety of tricks (two-metric, proximal-gradient, dual, etc.).
- Part 2: SG is optimal for learning, but constants matter and finite-sum methods are leading to improved results.
- Part 3: We are starting to be able to understand non-convex problems, but there is a lot of work to do.