IE598 Big Data Optimization

Summary
Nonconvex Optimization

Instructor: Niao He
April 16, 2018
This Course

Big Data Optimization

• **Explore** modern optimization theories, algorithms, and big data applications

• **Emphasize** a deep understanding of structure of optimization problems and computation complexity of numerical algorithms

• **Expose to** the frontier of research in large-scale optimization and machine learning
Central Topics

\[
\begin{align*}
\min_x & \quad f_0(x) \\
s.t. & \quad f_i(x) \leq 0, \ i = 1, \ldots, k \\
& \quad h_j(x) = 0, \ j = 1, \ldots, \ell \\
& \quad x \in X
\end{align*}
\]
Convex Optimization

What Did We Cover So Far

“The great watershed in optimization isn’t between linearity and nonlinearity, but convexity and nonconvexity.”

- R. Rockafellar, SIAM Review 1993
Smooth Convex Optimization

$$\min_{x \in X} f(x)$$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iteration Complexity</th>
<th>Iteration Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Convex</td>
<td>Strongly Convex</td>
</tr>
<tr>
<td>GD</td>
<td>$\mathcal{O}(\frac{LD^2}{\epsilon})$</td>
<td>$\mathcal{O}\left(\frac{L}{\mu} \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>AGD</td>
<td>$\mathcal{O}\left(\frac{\sqrt{LD}}{\sqrt{\epsilon}}\right)$</td>
<td>$\mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>PGD</td>
<td>$\mathcal{O}(\frac{LD^2}{\epsilon})$</td>
<td>$\mathcal{O}\left(\frac{L}{\mu} \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>FW</td>
<td>$\mathcal{O}(\frac{LD^2}{\epsilon})$</td>
<td>$\mathcal{O}\left(\frac{LD^2}{\epsilon}\right)$</td>
</tr>
<tr>
<td>BCGD</td>
<td>$\mathcal{O}\left(\frac{bLD^2}{\epsilon}\right)$</td>
<td>$\mathcal{O}\left(\frac{bL}{\mu} \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
</tbody>
</table>

- $L$ : Lipschitz constant of $\nabla f(x)$, $\mu$ : strongly convexity, $D$ : either $\|x_0 - x_*\|_2$ or diameter of set $X$
Example: Logistic Regression

- Logistic Regression

\[
\min_w f(w) := \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2} \|w\|_2^2
\]
Non-smooth Convex Optimization

\[
\min_{x \in X} f(x)
\]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iteration Complexity (Convex Case)</th>
<th>Iteration Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subgradient Descent</td>
<td>( \mathcal{O} \left( \frac{M^2 \cdot |f|<em>2 \cdot \max</em>{x, y \in X} |x - y|_2^2}{\epsilon^2} \right) )</td>
<td>one subgradient one projection</td>
</tr>
<tr>
<td>Mirror Descent</td>
<td>( \mathcal{O} \left( \frac{M^2 \cdot |f|<em>* \cdot \max</em>{x, y \in X} V(x, y)}{\epsilon^2} \right) )</td>
<td>one subgradient one prox-mapping</td>
</tr>
<tr>
<td>Proximal Point Algorithm</td>
<td>( \mathcal{O} \left( \frac{|x_0 - x_*|_2^2}{\epsilon} \right) )</td>
<td>one proximal operator</td>
</tr>
<tr>
<td>Acc Proximal Point Algorithm</td>
<td>( \mathcal{O} \left( \frac{|x_0 - x_*|_2}{\sqrt{\epsilon}} \right) )</td>
<td>one proximal operator</td>
</tr>
</tbody>
</table>

- \( V(x, y) \): Bregman distance w.r.t. some norm \( \| \cdot \| \) defined on \( X \), \( M \): Lipschitz constant of \( f(x) \)
Well-Structured
Non-smooth Convex Optimization

\[ \min_{x \in X} f(x) := \max_{y \in Y} \{ \langle Ax + b, y \rangle - \phi(y) \} \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iteration Complexity (Convex Case)</th>
<th>Iteration Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nesterov’s Smoothing + GD</td>
<td>( \mathcal{O} \left( \frac{|A|^2D_X^2 D_Y^2}{\epsilon^2} \right) )</td>
<td>one gradient of smoothed objective</td>
</tr>
<tr>
<td>Nesterov’s Smoothing + AGD</td>
<td>( \mathcal{O} \left( \frac{|A|D_X D_Y}{\epsilon} \right) )</td>
<td>one gradient of smoothed objective</td>
</tr>
<tr>
<td>Mirror Prox</td>
<td>( \mathcal{O} \left( \frac{L \cdot \max_{z,z' \in X \times Y} V(z,z')}{\epsilon} \right) )</td>
<td>two gradients and two prox-mappings</td>
</tr>
</tbody>
</table>

- \( V(z,z') \): Bregman distance w.r.t. some norm \( \| \cdot \| \) defined on \( X \times Y \), \( D_X, D_Y \): diameter of sets \( X \) and \( Y \)
- \( L \): Lipschitz constant of the gradient of the saddle function
Composite Non-smooth Convex Optimization

\[
\min_x f(x) + g(x)
\]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iteration Complexity (Convex Case)</th>
<th>Iteration Cost</th>
</tr>
</thead>
</table>
| Proximal Gradient                | \( \mathcal{O}\left(\frac{L\|x_0 - x_*\|_2^2}{\epsilon}\right) \) | one gradient of \( f \)  
                             |                     | one proximal operator of \( g \) |
| Accelerated Proximal Gradient    | \( \mathcal{O}\left(\frac{\sqrt{L}\|x_0 - x_*\|_2}{\sqrt{\epsilon}}\right) \) | one gradient of \( f \)  
                             |                     | one proximal operator of \( g \) |
| Douglas-Rachford Splitting      | \( \mathcal{O}\left(\frac{\|x_0 - x_*\|_2^2}{\epsilon}\right) \) | one proximal operator of \( f \)  
                             |                     | one proximal operator of \( g \) |
| Krasnosel’skii-Mann (KM)         | \( \mathcal{O}\left(\frac{\|x_0 - x_*\|_2^2}{\epsilon}\right) \) | one proximal operator of \( f \)  
                             |                     | one proximal operator of \( g \) |

\( L \): Lipschitz constant of \( \nabla f(x) \)
Example: LASSO

\[ \min_x \frac{1}{2} \|Ax - b\|^2_2 + \lambda \|x\|_1 \]
\[ \min_{x \in X} f(x) = \mathbb{E}[F(x, \xi)] \]  

<table>
<thead>
<tr>
<th>Approach</th>
<th>Sample Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Average Approximation (SAA)</td>
<td>( \mathcal{O} \left( \frac{\max_{x \in X} \text{Var}[F(x, \xi)]}{\epsilon^2} \right) )</td>
</tr>
<tr>
<td>Stochastic Approximation (SA)</td>
<td>( \mathcal{O} \left( \frac{\max_{x \in X} \mathbb{E}[|F'(x, \xi)|_2^2]}{\mu^2 \epsilon} \right) )</td>
</tr>
<tr>
<td>(when ( f ) is ( \mu )-strongly convex)</td>
<td></td>
</tr>
<tr>
<td>Mirror Descent SA (when ( f ) is general convex)</td>
<td>( \mathcal{O} \left( \frac{\max_{x \in X} \mathbb{E}[|F'(x, \xi)|<em>2^2] \cdot \max</em>{x, y \in X} V(x, y)}{\epsilon^2} \right) )</td>
</tr>
</tbody>
</table>

- \( V(x, y) \) : Bregman distance w.r.t. some norm \( \| \cdot \| \) defined on \( X \)
Finite Sum Optimization

\[ \min_x f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iteration Complexity (Smooth + Strongly Convex)</th>
<th>Iteration Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>( \mathcal{O} \left( \frac{L}{\mu} \log \left( \frac{1}{\epsilon} \right) \right) )</td>
<td>( O(n) ) gradient</td>
</tr>
<tr>
<td>SGD</td>
<td>( \mathcal{O} \left( \frac{L}{\mu^2 \epsilon} \right) )</td>
<td>( O(1) ) gradient</td>
</tr>
<tr>
<td>SVRG/S2GD</td>
<td>( \mathcal{O} \left( \log \left( \frac{1}{\epsilon} \right) \right) )</td>
<td>( O(n + \frac{L}{\mu}) ) gradient</td>
</tr>
<tr>
<td>SAG/SAGA</td>
<td>( \mathcal{O} \left( \max(n, \frac{L}{\mu}) \log \left( \frac{1}{\epsilon} \right) \right) )</td>
<td>( O(1) ) gradient, ( O(n) ) memory</td>
</tr>
</tbody>
</table>

- Assume each \( f_i \) is \( L \)-smooth and \( f \) is \( \mu \)-strongly convex
Example: Logistic Regression

- Large-scale Logistic Regression

\[
\min_w f(w) := \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2} \|w\|^2
\]
Learning = Generalization + Optimization

Fundamental theorem of machine learning

\[ R(w) = (R(w) - R_s(w)) + R_s(w) \]

Risk = Generalization error + Empirical Risk

- From Moritz Hardt, INFORMS Optimization Society talk in 2018
Convex Optimization

What We Did Not Cover Yet
More First-Order Algorithms

• **Nonsmooth Optimization**
  – Bundle methods (e.g. the level method)
  – Primal-dual methods
  – ...

• **Stochastic Optimization**
  – Dual averaging method
  – Stochastic Frank Wolfe algorithms
  – Stochastic dual coordinate ascent
  – Stochastic ADMM algorithm
  – Streaming SVRG algorithm
  – ...

Beyond First-Order Algorithms

• **Second-Order Methods**
  – Newton method
  – Cubic regularization method
  – Trust region method
  – (Stochastic) Quasi-Newton methods
  – Gauss-Newton method
  – Natural Gradient method
  – ...

• **Zero-Order Methods (Derivative-free)**
  – Fast Differentiation technique
  – Gaussian smoothing
  – Random search
  – ...
Beyond Black-Box

- Methods with linear dimension-dependent convergence
  - Cutting plane methods
  - Center-of-Gravity method
  - Inner and Outer Ellipsoid method
  - Interior Point Method
Parallel and Distributed Algorithms

- Many of the algorithms we learnt can be modified to take advantage of parallel processors and distributed machines
  - Distributed ADMM
  - Async-ADMM
  - Hogwild! /Gossip algorithms
  - Downpour SGD
  - Distributed dual averaging / SVRG/ COCOA/Disco/DSCOVR
  - ....
More on Convex Optimization

- Problems with Convex Structure
  - Convex Minimization
  - Convex-Concave Saddle Point Problems
  - Variational Inequalities
  - Convex Nash Equilibrium
  - Monotone Inclusion Problems
  - Conic Optimization
  - Submodular Optimization
  - Online Learning
  - Multi-Stage Stochastic Programming
  - ...
More Applications in ML

- Aside from supervised learning, many other tasks in ML can be formulated as convex problems
  - Boosting
  - Kernel Machines
  - Bayesian Inference
  - Reinforcement Learning
  - Recommendation Systems
  - Ranking / Choice Models
  - ...
Non-convex Optimization

“Convexity is overrated. Theoretical guarantees are overrated.”
Yann LeCun, 2007
Many practical problems are non-convex and are hard to solve.

- **Nonlinear Optimization**
  - Polynomial optimization
  - Convex equality constraints
  - Eigenvalue problems

- **Integer and Combinatorial Optimization**
  - Traveling salesman problem

- **Optimization under Uncertainty**
  - Robust Optimization
  - Chance Constrained Programming
  - Multi-Stage Stochastic Programming
Non-Convex Applications in ML

Lots of problems in machine learning are indeed non-convex:

- Multi-class classification
- Sparsity learning with non-convex regularization
- Clustering (K-means, PCA, etc)
- Matrix factorization, Dictionary Learning
- Graphical models (MRF, HMM, etc)
- Deep learning
- Deep reinforcement learning
- Generative adversarial network
- ....
Convex vs. Nonconvex Worlds

Local minima = Global minima

NP-hardness: Existence of Exponential Number of Stationary Points
Convex Relaxation

This involves convexifying the constraints or the objectives:

- **LP relaxation of 0-1 integer programs**
  - e.g., min-cut
- **SDP relaxation of quadratic / polynomial optimization**
  - e.g., max-cut, sum of squares, two-layer neural network
- **Convex regularizations**
  - e.g., compressive sensing, matrix completion
- **Convex hulls/ convex envelopes**
  - e.g., Gaussian Homotopy Continuation

Many successfully scenarios, but not yet applicable to deep learning…
Global Optimization

These are often slow and has exponential worse-case performance:

- Branch and Bound
- Simulated Annealing
- Bayesian Optimization
- Graduated optimization
- ....

Not applicable for large-scale problems… Need first-order methods …
For smooth problems, most first-order methods are guaranteed to converge to a stationary point: $\nabla f(x) = 0$.

Convergence criterion: $||\nabla f(x)||^2 \leq \epsilon$

- **Steepest Descent** *(Cartis, Gould & Toint, 2010):* $O\left(\frac{1}{\epsilon}\right)$
- **(Accelerated)Gradient Descent** *(Ghadimi & Lan, 2013):* $O\left(\frac{1}{\epsilon}\right)$
- **Randomized Stochastic Gradient** *(Ghadimi & Lan, 2013):* $O\left(\frac{1}{\epsilon^2}\right)$
- **SAGA** *(Reddi et al, 2016):* $O\left(n + \frac{n^{2/3}}{\epsilon}\right)$

Stationary points could be local minima, local maxima, or saddle points…
Convex-Like Problems

Under some “convex-like” assumptions, FOMs could converge to the global optimal solution:

- **Quasi-convexity**

- **Variational coherence** (Zhou et al., 2017)
  - $VC: \nabla f(x)^T(x - x^*) \geq 0, \forall x \in X, x^* \in X^*$ with equality iff $x \in X^*$

- **Polyak-Lojasiewicz Condition** (Karimi, Nutini, Schmidt, 2016)
  - $PL: ||\nabla f(x)||^2 \geq 2\mu(f(x) - f^*), \forall x$
  - More generally, Kurdyka-Lojasiewicz or error bound condition

Under strong convex-like assumptions, FOMs converge to global optimal solution.
Other Landscapes

- **No saddle points exist**
  - E.g., quasi-convex problems

- **Saddles exist, but they're all strict**
  - E.g., dictionary learning, phase retrieval, tensor decomposition
  - Saddle point can be escaped by using restarting, noisy gradient, second-order methods (trust region, cubic regularization, etc) (Sun, Qu, & Wright, 2015)
  - Gradient descent with random initialization converges to local minima almost surely (Lee et al, 2016)

- **Flat saddles exist, but all local minima are global**
  - E.g., matrix sensing, matrix completion and robust PCA, overparametrized NN
  - Simple method such as SGD converges to global optimum (Ge, Lee, Ma, 2016) (Ge, Jin, Zheng, 2017)

- Local minima are almost as good as global minima.
- Bad local minima exist, but gradient methods converge to good ones

From Moritz Hardt, INFORMS Optimization Society talk in 2018
The Never Ending Endeavour