Nonconvex problems are …

Nonconvex optimization problem with simple constraints

\[
\begin{align*}
\min & \quad \left( \sum_i a_i z_i - s \right)^2 + \sum_i z_i(1 - z_i) \\
\text{s.t.} & \quad 0 \leq z_i \leq 1, \quad i = 1, \ldots, n.
\end{align*}
\]

**Question:** Is global min of this problem 0 or not?

Does there exist a subset of \( \{a_1, a_2, \ldots, a_n\} \) that sums to \( s \)?

Subset-sum problem, well-known NP-Complete prob.

\[
\begin{align*}
\min & \quad x^\top Ax, \quad x \geq 0
\end{align*}
\]

**Question:** Is \( x=0 \) a local minimum or not?
Nonconvex finite-sum problems

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathcal{DNN}(x_i, \theta)) + \Omega(\theta)
\]
Nonconvex finite-sum problems

\[
\min_{\theta \in \mathbb{R}^d} \quad g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)
\]

Related work

- Original SGD paper \textit{(Robbins, Monro 1951)}
  (asymptotic convergence; no rates)

- SGD with scaled gradients \((\theta_t - \eta_t H_t \nabla f(\theta_t))\) + other tricks:
  space dilation, \textit{(Shor, 1972)}; variable metric SGD \textit{(Uryasev 1988)}; AdaGrad
  \textit{(Duchi, Hazan, Singer, 2012)}; Adam \textit{(Kingma, Ba, 2015)}, and many others...
  (typically asymptotic convergence for nonconvex)

- Large number of other ideas, often for step-size tuning, initialization
  (see e.g., blog post: by S. Ruder on gradient descent algorithms)

Going beyond SGD (theoretically; ultimately in practice too)
Nonconvex finite-sum problems

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

Related work (subset)

- (Solodov, 1997) **Incremental gradient**, smooth nonconvex (asymptotic convergence; no rates proved)
- (Bertsekas, Tsitsiklis, 2000) Gradient descent with errors; **incremental** (see §2.4, Nonlinear Programming; no rates proved)
- (Sra, 2012) **Incremental** nonconvex non-smooth (asymptotic convergence only)
- (Ghadimi, Lan, 2013) SGD for nonconvex stochastic opt. (first non-asymptotic rates to stationarity)
- (Ghadimi et al., 2013) SGD for nonconvex non-smooth stochastic opt. (non-asymptotic rates, but key limitations)
Difficulty of nonconvex optimization

So, try to see how fast we can satisfy this necessary condition

Difficult to optimize, but

$$\nabla g(\theta) = 0$$

necessary condition – local minima, maxima, saddle points satisfy it.
Measuring efficiency of nonconvex opt.

Convex: \[ \mathbb{E}[g(\theta_t) - g^*] \leq \epsilon \] (optimality gap)

Nonconvex: \[ \mathbb{E}[||\nabla g(\theta_t)||^2] \leq \epsilon \] (stationarity gap)

Incremental First-order Oracle (IFO)

\[ (x, i) \rightarrow (f_i(x), \nabla f_i(x)) \]

Measure: \#IFO calls to attain \( \epsilon \) accuracy

(Nesterov 2003, Chap 1; Ghadimi, Lan, 2012)

(Agarwal, Bottou, 2014)

(see also: Nemirovski, Yudin, 1983)
IFO Example: SGD vs GD (nonconvex)

\[
\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)
\]

SGD

\[\theta_{t+1} = \theta_t - \eta \nabla f_i_t(\theta_t)\]

\begin{itemize}
  \item O(1) IFO calls per iter
  \item O(1/\epsilon^2) iterations
  \item **Total**: O(1/\epsilon^2) IFO calls
  \item independent of n
\end{itemize}

(Nghadimi, Lan, 2013, 2014)

GD

\[\theta_{t+1} = x_t - \eta \nabla g(\theta_t)\]

\begin{itemize}
  \item O(n) IFO calls per iter
  \item O(1/\epsilon) iterations
  \item **Total**: O(n/\epsilon) IFO calls
  \item depends strongly on n
\end{itemize}

(Nesterov, 2003; Nesterov 2012)

assuming Lipschitz gradients

\[\mathbb{E}[\|\nabla g(\theta_t)\|^2] \leq \epsilon\]
Nonconvex finite-sum problems

\[ \min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \]

SGD
\[ \theta_{t+1} = \theta_t - \eta \nabla f_{i_t}(\theta_t) \]

GD
\[ \theta_{t+1} = x_t - \eta \nabla g(\theta_t) \]

SAG, SVRG, SAGA, et al.

Do these benefits extend to nonconvex finite-sums?

Analysis depends heavily on convexity (especially for controlling variance)
SVRG/SAGA work again!
(with new analysis)
Nonconvex SVRG

\[ \textbf{for } s=0 \text{ to } S-1 \]
\[ \theta_{0}^{s+1} \leftarrow \theta_{m}^{s} \]
\[ \tilde{\theta}^{s} \leftarrow \theta_{m}^{s} \]

\[ \textbf{for } t=0 \text{ to } m-1 \]

Uniformly randomly pick \[ i(t) \in \{1, \ldots, n\} \]
\[ \theta_{t+1}^{s+1} = \theta_{t}^{s+1} - \eta_{t} \left( \nabla f_{i(t)}(\theta_{t}^{s+1}) - \nabla f_{i(t)}(\tilde{\theta}^{s}) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_{i}(\tilde{\theta}^{s}) \right) \]

end

end

The same algorithm as usual SVRG \textit{(Johnson, Zhang, 2013)}
Nonconvex SVRG

\[ \text{for } s=0 \text{ to } S-1 \]
\[ \theta_0^{s+1} \leftarrow \theta_m^s \]
\[ \tilde{\theta}^s \leftarrow \theta_m^s \]

\[ \text{for } t=0 \text{ to } m-1 \]

Uniformly randomly pick \( i(t) \in \{1, \ldots, n\} \)
\[ \theta_t^{s+1} = \theta_t^{s+1} - \eta_t \left[ \nabla f_{i(t)}(\theta_t^{s+1}) - \nabla f_{i(t)}(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right] \]

end

end
Nonconvex SVRG

for $s = 0$ to $S - 1$

\[ \theta_0^{s+1} \leftarrow \theta_m^s \]

\[ \tilde{\theta}^s \leftarrow \theta_m^s \]

for $t = 0$ to $m - 1$

Uniformly randomly pick $i(t) \in \{1, \ldots, n\}$

\[ \theta_{t+1}^{s+1} = \theta_t^{s+1} - \eta_t \left[ \nabla f_{i(t)}(\theta_t^{s+1}) - \nabla f_{i(t)}(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right] \]

end

end
Nonconvex SVRG

for $s=0$ to $S-1$

\[
\begin{aligned}
\theta_0^{s+1} &\leftarrow \theta_m^s \\
\tilde{\theta}^s &\leftarrow \theta_m^s \\
\end{aligned}
\]

for $t=0$ to $m-1$

Uniformly randomly pick $i(t) \in \{1, \ldots, n\}$

\[
\begin{aligned}
\theta_{t+1}^{s+1} &= \theta_{t+1}^{s+1} - \eta_t \left[ \nabla f_{i(t)}(\theta_{t+1}^{s+1}) - \nabla f_{i(t)}(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right]
\end{aligned}
\]

end

end
Nonconvex SVRG

\begin{align*}
\text{for } s &= 0 \text{ to } S-1 \\
\theta_0^{s+1} &\leftarrow \theta_m^s \\
\tilde{\theta}^s &\leftarrow \theta_m^s \\
\text{for } t &= 0 \text{ to } m-1 \\
\text{Uniformly randomly pick } i(t) &\in \{1, \ldots, n\} \\
\theta_t^{s+1} &= \theta_t^{s+1} - \eta_t \left[ \nabla f_i(t)(\theta_t^{s+1}) - \nabla f_i(t)(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right] \\
\text{end} \\
\text{end}
\end{align*}
Nonconvex SVRG

\begin{align*}
\text{for } s=0 \text{ to } S-1 & \\
\theta_0^{s+1} & \leftarrow \theta_m^s \\
\tilde{\theta}^s & \leftarrow \theta_m^s \\
\text{for } t=0 \text{ to } m-1 & \\
& \text{Uniformly randomly pick } i(t) \in \{1, \ldots, n\} \\
\theta_t^{s+1} & = \theta_t^{s+1} - \eta_t \left[ \nabla f_{i(t)}(\theta_t^{s+1}) - \nabla f_{i(t)}(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right] \\
\text{end} & \\
\text{end} & \\
\Delta_t & \\
\mathbb{E}[\Delta_t] & = 0
\end{align*}
Nonconvex SVRG

for s=0 to S-1
  \( \theta_{0}^{s+1} \leftarrow \theta_{m}^{s} \)  
  \( \tilde{\theta}^{s} \leftarrow \theta_{m}^{s} \)
  
  for t=0 to m-1
    Uniformly randomly pick \( i(t) \in \{1, \ldots, n\} \)
    \( \theta_{t+1}^{s+1} = \theta_{t+1}^{s} - \eta_{t} \left[ \nabla f_{i(t)}(\theta_{t+1}^{s}) - \nabla f_{i(t)}(\tilde{\theta}^{s}) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_{i}(\tilde{\theta}^{s}) \right] \)
  end
end

Full gradient, computed once every epoch
Nonconvex SVRG

1. For $s=0$ to $S-1$
   - Update $\theta^{s+1}_0 \leftarrow \theta^s_m$
   - Update $\tilde{\theta}^s \leftarrow \theta^s_m$

2. For $t=0$ to $m-1$
   - Uniformly randomly pick $i(t) \in \{1, \ldots, n\}$
   - Compute
     \[
     \theta^{s+1}_{t+1} = \theta^{s+1}_t - \eta_t \left[ \nabla f_{i(t)}(\theta^{s+1}_t) - \nabla f_{i(t)}(\tilde{\theta}^s) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\theta}^s) \right]
     \]

Key quantities that determine how the method operates

Full gradient, computed once every epoch
Key ideas for analysis of nc-SVRG

Previous SVRG proofs rely on **convexity to control variance**

Larger step-size $\Rightarrow$ smaller inner loop
(full-gradient computation dominates epoch)

Smaller step-size $\Rightarrow$ slower convergence
(longer inner loop)

(Carefully) trading-off #inner-loop iterations $m$ with step-size $\eta$ leads to lower #IFO calls!

*(Reddi, Hefny, Sra, Poczos, Smola, 2016; Allen-Zhu, Hazan, 2016)*
Faster nonconvex optimization via VR

(Reddi, Hefny, Sra, Poczos, Smola, 2016; Reddi et al., 2016)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Nonconvex (Lipschitz smooth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>$O\left(\frac{1}{\epsilon^2}\right)$</td>
</tr>
<tr>
<td>GD</td>
<td>$O\left(\frac{n}{\epsilon}\right)$</td>
</tr>
<tr>
<td>SVRG</td>
<td>$O\left(n + \frac{n^{2/3}}{\epsilon}\right)$</td>
</tr>
<tr>
<td>SAGA</td>
<td>$O\left(n + \frac{n^{2/3}}{\epsilon}\right)$</td>
</tr>
<tr>
<td>MSVRG</td>
<td>$O\left(\min\left(\frac{1}{\epsilon^2}, \frac{n^{2/3}}{\epsilon}\right)\right)$</td>
</tr>
</tbody>
</table>

Remarks

New results for convex case too; additional nonconvex results
For related results, see also (Allen-Zhu, Hazan, 2016)
Linear rates for nonconvex problems

\[
\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)
\]

The Polyak-Łojasiewicz (PL) class of functions

\[
g(\theta) - g(\theta^*) \leq \frac{1}{2\mu} \|\nabla g(\theta)\|^2
\]

(Polyak, 1963); (Łojasiewicz, 1963)

Examples:

- \(\mu\)-strongly convex \(\Rightarrow\) PL holds

- Stochastic PCA**, some large-scale eigenvector problems

(More general than many other “restricted” strong convexity uses)

(Karimi, Nutini, Schmidt, 2016)

(Attouch, Bolte, 2009)

(Bertsekas, 2016)

proximal extensions; references

more general Kurdya-Łojasiewicz class

textbook, more “growth conditions”
Linear rates for nonconvex problems

\[ g(\theta) - g(\theta^*) \leq \frac{1}{2\mu} \| \nabla g(\theta) \|^2 \quad \text{and} \quad \mathbb{E}[g(\theta_t) - g^*] \leq \epsilon \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Nonconvex</th>
<th>Nonconvex-PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>( O\left(\frac{1}{\epsilon^2}\right) )</td>
<td>( O\left(\frac{1}{\epsilon^2}\right) )</td>
</tr>
<tr>
<td>GD</td>
<td>( O\left(\frac{n}{\epsilon}\right) )</td>
<td>( O\left(\frac{n}{2\mu} \log \frac{1}{\epsilon}\right) )</td>
</tr>
<tr>
<td>SVRG</td>
<td>( O\left(n + \frac{n^{2/3}}{\epsilon}\right) )</td>
<td>( O\left((n + \frac{n^{2/3}}{2\mu}) \log \frac{1}{\epsilon}\right) )</td>
</tr>
<tr>
<td>SAGA</td>
<td>( O\left(n + \frac{n^{2/3}}{\epsilon}\right) )</td>
<td>( O\left((n + \frac{n^{2/3}}{2\mu}) \log \frac{1}{\epsilon}\right) )</td>
</tr>
<tr>
<td>MSVRG</td>
<td>( O\left(\min\left(\frac{1}{\epsilon^2}, \frac{n^{2/3}}{\epsilon}\right)\right) )</td>
<td>-</td>
</tr>
</tbody>
</table>

Variant of \textbf{nc-SVRG} attains this fast convergence!

(Reddi, Hefny, Sra, Poczos, Smola, 2016; Reddi et al., 2016)
Empirical results

CIFAR10 dataset; 2-layer NN
Some surprises!

\[
\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) + \Omega(\theta)
\]

Regularizer, e.g., \( \| \cdot \|_1 \) for enforcing sparsity of weights (in a neural net, or more generally); or an indicator function of a constraint set, etc.
Nonconvex composite objective problems

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) + \Omega(\theta)
\]

\text{convex} \quad \text{nonconvex}

Prox-SGD

\[\theta_{t+1} = \text{prox}_{\lambda_t \Omega} \left( \theta_t - \eta_t \nabla f_i(\theta_t) \right)\]

Prox-SGD convergence not known!*

\[\text{prox}_{\lambda \Omega}(v) := \arg \min_u \frac{1}{2} \|u - v\|^2 + \lambda \Omega(u)\]

prox: soft-thresholding for \( \| \cdot \|_1 \); projection for indicator function

– Partial results: (Ghadimi, Lan, Zhang, 2014)
  (using growing minibatches, shrinking step sizes)

* Except in special cases (where even rates may be available)
Empirical results: NN-PCA

Eigenvecs via SGD: (Oja, Karhunen 1985); via SVRG (Shamir, 2015, 2016); (Garber, Hazan, Jin, Kakade, Musco, Netrapalli, Sidford, 2016); and many more!
Finite-sum problems with nonconvex $g(\theta)$ and params $\theta$ lying on a known manifold

$$\min_{\theta \in \mathcal{M}} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

Example: eigenvector problems (the $||\theta||=1$ constraint)
problems with orthogonality constraints
low-rank matrices
positive definite matrices / covariances
Nonconvex optimization on manifolds

\[ \min_{\theta \in \mathcal{M}} \quad g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \]

(Zhang, Reddi, Sra, 2016)

Related work

- (Udriste, 1994) batch methods; textbook
- (Edelman, Smith, Arias, 1999) classic paper; orthogonality constraints
- (Absil, Mahony, Sepulchre, 2009) textbook; convergence analysis
- (Boumal, 2014) phd thesis, algos, theory, examples
- (Mishra, 2014) phd thesis, algos, theory, examples
- manopt excellent matlab toolbox
- (Bonnabel, 2013) Riemannnian SGD, asymptotic convg.
- and many more!

Exploiting manifold structure yields speedups
Example: Gaussian Mixture Model

\[ p_{\text{mix}}(x) := \sum_{k=1}^{K} \pi_k p_N(x; \Sigma_k, \mu_k) \]

Likelihood \[ \max \prod_i p_{\text{mix}}(x_i) \]

Numerical challenge: positive definite constraint on \( \Sigma_k \)

Riemannian (new)

EM Algo

Cholesky \( L L^T \)

[ Hosseini, Sra, 2015 ]
Careful use of manifold geometry helps!

<table>
<thead>
<tr>
<th>K</th>
<th>EM</th>
<th>R-LBFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17s // 29.28</td>
<td>14s // 29.28</td>
</tr>
<tr>
<td>5</td>
<td>202s // 32.07</td>
<td>117s // 32.07</td>
</tr>
<tr>
<td>10</td>
<td>2159s // 33.05</td>
<td>658s // 33.06</td>
</tr>
</tbody>
</table>

images dataset

d=35,
n=200,000

github.com/utvisionlab/mixest

Riemannian-LBFGS (careful impl.)
Large-scale Gaussian mixture models!

Riemannian SGD for GMMs
(d=90, n=515345)
Larger-scale optimization

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$
Simplest setting: using mini-batches

Idea: Use ‘b’ stochastic gradients / IFO calls per iteration

useful in parallel and distributed settings
increases parallelism, reduces communication

$$\text{SGD} \quad \theta_{t+1} = \theta_t - \frac{\eta_t}{|I_t|} \sum_{j \in I_t} \nabla f_j(\theta_t)$$

For batch size $b$, SGD takes a factor $1/\sqrt{b}$ fewer iterations

(Dekel, Gilad-Bachrach, Shamir, Xiao, 2012)

For batch size $b$, SVRG takes a factor $1/b$ fewer iterations

Theoretical **linear speedup** with parallelism

see also S2GD (convex case): (Konečný, Liu, Richtárik, Takáč, 2015)
Asynchronous stochastic algorithms

\[
\theta_{t+1} = \theta_t - \frac{\eta_t}{|I_t|} \sum_{j \in I_t} \nabla f_j(\theta_t)
\]

- Inherently sequential algorithm
- Slow-downs in parallel/dist settings (synchronization)

Classic results in asynchronous optimization: \textit{(Bertsekas, Tsitsiklis, 1987)}

- Asynchronous SGD implementation (HogWild!)
  Avoids need to sync, operates in a “lock-free” manner
- **Key assumption:** sparse data (often true in ML)

but

It is still SGD, thus has slow sublinear convergence
even for strongly convex functions
Asynchronous algorithms: parallel

Does variance reduction work with asynchrony?

Yes!

ASVRG \textit{(Reddi, Hefny, Sra, Poczos, Smola, 2015)}

ASAGA \textit{(Leblond, Pedregosa, Lacoste-Julien, 2016)}

Perturbed iterate analysis \textit{(Mania et al, 2016)}

– a few subtleties involved
– some gaps between theory and practice
– more complex than async-SGD

\textbf{Bottomline:} on sparse data, can get almost linear speedup due to parallelism ($\pi$ machines lead to $\sim \pi$ speedup)
Asynchronous algorithms: distributed

common parameter server architecture

(Li, Andersen, Smola, Yu, 2014)

Classic ref: (Bertsekas, Tsitsiklis, 1987)

D-SGD:

- workers compute (stochastic) gradients
- server computes parameter update
- can have quite high communication cost

Asynchrony via: servers use delayed / stale gradients from workers

(Nedic, Bertsekas, Borkar, 2000; Agarwal, Duchi 2011) and many others

(Shamir, Srebro 2014) – nice overview of distributed stochastic optimization
Asynchronous algorithms: distributed

To reduce communication, following idea is useful:

Data

\[ D_1 \quad D_2 \quad \ldots \quad D_m \]

Workers

\[ W_1 \quad W_2 \quad \ldots \quad W_m \]

Servers

\[ S_1 \quad \ldots \quad S_k \]

Worker nodes solve compute intensive subproblems

Servers perform simple aggregation (eg. full-gradients for distributed SVRG)

DANE (Shamir, Srebro, Zhang, 2013): distributed Newton, view as having an SVRG-like gradient correction
Asynchronous algorithms: distributed

**Key point:** Use SVRG (or related fast method) to solve suitable subproblems at workers; reduce #rounds of communication; (or just do D-SVRG)

**Some related work**

(Lee, Lin, Ma, Yang, 2015) D-SVRG, and accelerated version for some special cases (applies in smaller condition number regime)

(Ma, Smith, Jaggi, Jordan, Richtárik, Takáč, 2015) CoCoA+: (updates m local dual variables using m local data points; any local opt. method can be used); higher runtime+comm.

(Shamir, 2016) D-SVRG via cool application of without replacement SVRG! regularized least-squares problems only for now

Several more: DANE, DISCO, AIDE, etc.
Summary

- VR stochastic methods for nonconvex problems
- Surprises for proximal setup
- Nonconvex problems on manifolds
- Large-scale: parallel + sparse data
- Large-scale: distributed; SVRG benefits, limitations

If there is a finite-sum structure, can use VR ideas!
Perspectives: did not cover these!

- Stochastic quasi-convex optim. *(Hazan, Levy, Shalev-Shwartz, 2015)*
  
- Nonlinear eigenvalue-type problems *(Belkin, Rademacher, Voss, 2016)*

- Frank-Wolfe + SVRG: *(Reddi, Sra, Poczos, Smola, 2016)*

- Newton-type methods: *(Carmon, Duchi, Hinder, Sidford, 2016); (Agarwal, Allen-Zhu, Bullins, Hazan, Ma, 2016)*;

- many more, including robust optimization,

- infinite dimensional nonconvex problems

- geodesic-convexity for global optimality

- polynomial optimization

- many more… it’s a rich field!
Perspectives

- Impact of non-convexity on generalization
- Non-separable problems (e.g., maximize AUC); saddle point problems; robust optimization; heavy tails
- Convergence theory, local and global
- Lower-bounds for nonconvex finite-sums
- Distributed algorithms (theory and implementations)
- New applications (e.g., of Riemannian optimization)
- Search for other more “tractable” nonconvex models
- Specialization to deep networks, software toolkits