Nonconvex Optimization for Knowledge Discovery and Data Mining

Pan Xu

Quanquan Gu

Department of Computer Science
University of California, Los Angeles

Outline

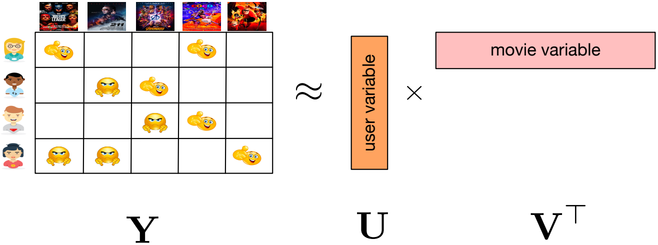
Nonconvexity in Data Mining

Nonconvex Finite-sum Optimization
- Finding First-order Stationary Points
- Finding Local Minima in Nonconvex Optimization
- Finding Local Minima via First-order Algorithms

Structured Nonconvex Problems
- Low Rank Matrix Recovery
- Robust PCA
- Gaussian Graphical Models

References

Collaborative Filtering



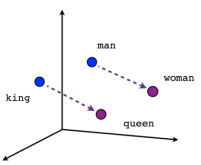
- **Matrix completion:** recover the underlying user-movie score matrix by minimizing the following objective

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2p} \sum_{(j,k)\in\Omega} (\mathbf{U}_{j*}\mathbf{V}_{k*}^\top - Y_{jk})^2$$

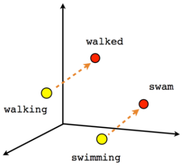
Word Embedding

- **Word2vec:** learn word embeddings by maximizing the following objective [MSC+13]

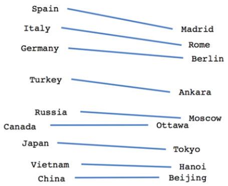
$$\log \sigma\left(\mathbf{u}_{w_O}^\top \mathbf{v}_{w_I}\right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)}\left[\log \sigma\left(-\mathbf{u}_{w_i}^\top \mathbf{v}_{w_I}\right)\right]$$



Male-Female



Verb tense



Country-Capital

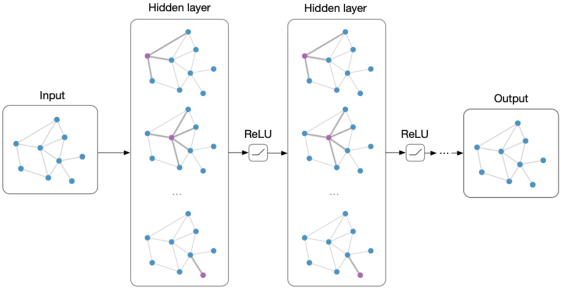
Network embedding

- **Graph convolutional network (GCN):** GCN learns network embeddings using graph-based neural network structure [KW16]

$$\mathbf{Z} = \text{softmax}\Big(\widehat{\mathbf{A}}\text{ReLU}(\cdots\text{ReLU}(\widehat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\cdots)\mathbf{W}^{(L)}\Big),$$

where the weight matrices are trained via solving

$$\min_{\mathbf{W}^{(0)}\ldots\mathbf{W}^{(L)}} \ell(\mathbf{Z}, \mathbf{Y}).$$



Outline

Finite-sum Optimization

- The finite-sum optimization problem:

$$\min f(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}),$$

  where $f_i(\theta)$ and f are nonconvex
- For example, n can be the number of data points.

Goal: find stationary points

- First-order stationary point (FSP): $\|\nabla f(\boldsymbol{\theta})\|_2 = 0$

FSP		ϵ-approximate FSP
$\|\nabla f(\boldsymbol{\theta})\|_2 = 0$	$\Rightarrow$	$\|\nabla f(\boldsymbol{\theta})\|_2 \leq \epsilon$

Outline

Nonconvexity in Data Mining

Nonconvex Finite-sum Optimization
- Finding First-order Stationary Points
- Finding Local Minima in Nonconvex Optimization
- Finding Local Minima via First-order Algorithms

Structured Nonconvex Problems
- Low Rank Matrix Recovery
- Robust PCA
- Gaussian Graphical Models

References

Gradient Descent

Gradient Descent (GD):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla f(\boldsymbol{\theta}_t)$$

- Converge to an ϵ-approximate stationary point within $O(1/\epsilon^2)$ iterations
- **Gradient complexity:** number of gradient computation in order to find an ϵ-approximate stationary point
- Gradient complexity of GD: $O(n/\epsilon^2)$

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD)

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{g}_t$$

- $\mathbf{g}_t = \nabla f_{i_t}(\boldsymbol{\theta}_t)$, i_t: uniformly chosen from $\{1, \ldots, n\}$
- Converge to an ϵ-approximate stationary point within $O(1/\epsilon^4)$ iterations
- Gradient complexity: $O(1/\epsilon^4)$

Adaptive Methods

Partially adaptive momentum estimation method (Padam) [CG18]

$$
\begin{aligned}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t \\
\mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2 \\
\widehat{\mathbf{v}}_t &= \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t) \\
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta_t \frac{\mathbf{m}_t}{\widehat{\mathbf{v}}_t^{p}}
\end{aligned}
$$

$p \in (0, 1/2]$: tuning parameter for improving generalization

- $p = 1/2$, $\Rightarrow$ AMSGrad [RKK18]
- $p = 1/2$, $\max(\cdot)$ removed, $\Rightarrow$ ADAM [KB14]
- $p = 1/2$, $\max(\cdot)$ removed, $\beta_1 = 0$, $\Rightarrow$ RMSprop [HSS12]
- $p = 1/2$, $\max(\cdot)$ removed, $\beta_1 = 0$, $\mathbf{v} = 1/t \sum_{j=1}^{t} \boldsymbol{g}_j^2$, $\Rightarrow$ AdaGrad [DHS11]

Stochastic Variance Reduced Gradient Methods

Stochastic Variance Reduced Gradient (SVRG) [JZ13]

for t = 1, 2, . . . , T
  $\widetilde{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_t$
  Calculate full gradient $\mu = \nabla f(\widetilde{\boldsymbol{\theta}}_0)$
  for k = 0,. . . ,m-1
    Randomly choose i_k from $[n]$
    $\mathbf{g}_k = \mu + \nabla f_{i_k}(\boldsymbol{\theta}_k) - \nabla f_{i_t}(\widetilde{\boldsymbol{\theta}}_0)$
    $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{g}_k$
  end for
  $\boldsymbol{\theta}_{t+1} = \widetilde{\boldsymbol{\theta}}_m$
end for

- Semi-stochastic gradient: snapshot every m iterations
- Reference point, reference gradient
- Linear convergence to global minimum in strongly convex setting

Nonconvex SVRG

Nonconvex SVRG [AZH16, RHS+16]

for t = 1, 2, . . . , T
  $\widetilde{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_t$
  Calculate full gradient $\mu = \nabla f(\widetilde{\boldsymbol{\theta}}_0)$
  for k = 0,. . . ,m-1
    Randomly choose i_k from $[n]$
    $\mathbf{g}_k = \mu + \nabla f_{i_k}(\boldsymbol{\theta}_k) - \nabla f_{i_t}(\widetilde{\boldsymbol{\theta}}_0)$
    $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{g}_k$
  end for
  $\boldsymbol{\theta}_{t+1} = \widetilde{\boldsymbol{\theta}}_m$
end for

- Gradient complexity

$$O\left(n + \frac{n^{2/3}}{\epsilon^2}\right)$$

Stochastically Controlled Stochastic Gradient

Stochastically Controlled Stochastic Gradient (SCSG) [LJCJ17]

for t = 1, 2, . . . , T
  $\widetilde{\theta}_0 = \theta_t$
  $\mu = 1/B \sum_{i \in \widetilde{\mathcal{I}}} \nabla f_i(\widetilde{\theta})$, with $|\widetilde{\mathcal{I}}| = B$
  Generate $m \sim \mathrm{Geom}(B/(B+b))$
  for k = 0,. . . ,m-1
    Randomly choose a subset $\mathcal{I}_k$ of $[n]$, with $|\mathcal{I}_k| = b$
    $\mathbf{g}_k = \mu + 1/b \sum_{i \in \mathcal{I}_k} \big(\nabla f_i(\widetilde{\theta}_k) - \nabla f_i(\widetilde{\theta}_0)\big)$
    $\theta_{k+1} = \theta_k - \eta \mathbf{g}_k$
  end for
  $\theta_{t+1} = \widetilde{\theta}_m$
end for

- Mini-batch gradient in both outer loop and inner loop
- Gradient complexity $O(\min\{n^{2/3}/\epsilon^2, 1/\epsilon^{10/3}\})$
- Geometric distribution not necessary: [LL18]

Comparison on Gradient Complexity

To find an ϵ-approximate first-order stationary point:

$$\|\nabla f(\boldsymbol{\theta})\|_2 \leq \epsilon,$$

the number of stochastic gradient ∇f_i we need to compute is

Algorithm	Gradient Complexity
GD	$O(n\epsilon^{-2})$
SGD	$O(\epsilon^{-4})$
SVRG	$O(n^{2/3}\epsilon^{-2})$
SCSG	$O(\min\{n^{2/3}\epsilon^{-2}, \epsilon^{-10/3}\})$

Revisit SVRG

for $t_0 = 0, \ldots, T_0 - 1$

$\mathbf{g}_{t_0}^{(0)} = 1/B_0 \sum_{i \in \mathcal{I}_0} \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$ ⇒reference gradient

for $t_1 = 0, \ldots, T_1 - 1$

$\mathbf{g}_{t_1}^{(1)} = 1/B_1 \sum_{i \in \mathcal{I}_1} \nabla f_i(\boldsymbol{\theta}_{t_1}^{(1)}) - \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$

$\mathbf{v}_t = \mathbf{g}_{t_0}^{(0)} + \mathbf{g}_{t_1}^{(1)}$, where $t = t_0 T_1 + t_1$

$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{v}_t$

$\boldsymbol{\theta}_{t_1+1}^{(1)} = \boldsymbol{\theta}_{t+1}$

end for

$\boldsymbol{\theta}_{t_0+1}^{(0)} = \boldsymbol{\theta}_{T_1}^{(1)}$ ⇒reference point

end for

- Mini-batch gradients
- Two reference points, two reference gradients
- Can more reference gradients reduce more variance?

Deeper SVRG

for $t_0 = 0, \ldots, T_0 - 1$

$\mathbf{g}_{t_0}^{(0)} = 1/B_0 \sum_{i \in \mathcal{I}_0} \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$

for $t_1 = 0, \ldots, T_1 - 1$

$\mathbf{g}_{t_1}^{(1)} = 1/B_1 \sum_{i \in \mathcal{I}_1} \nabla f_i(\boldsymbol{\theta}_{t_1}^{(1)}) - \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$

for $t_2 = 1, \ldots, T_2 - 1$

$\mathbf{g}_{t_2}^{(2)} = 1/B_2 \sum_{i \in \mathcal{I}_2} \nabla f_i(\boldsymbol{\theta}_{t_2}^{(2)}) - \nabla f_i(\boldsymbol{\theta}_{t_1}^{(1)})$

$\mathbf{v}_t = \mathbf{g}_{t_0}^{(0)} + \mathbf{g}_{t_1}^{(1)} + \mathbf{g}_{t_2}^{(2)}$, where $t = t_0 T_1 T_2 + t_1 T_2 + t_2$

$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{v}_t$

$\boldsymbol{\theta}_{t_2+1}^{(2)} = \boldsymbol{\theta}_{t+1}$

end for

$\boldsymbol{\theta}_{t_1+1}^{(1)} = \boldsymbol{\theta}_{T_2}^{(2)}$

end for

$\boldsymbol{\theta}_{t_0+1}^{(0)} = \boldsymbol{\theta}_{T_1}^{(1)}$

end for

▷ $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2 \subset [n]$: batch sets with sizes B_0, B_1, B_2

Deeper SVRG

for $t_0 = 0, \ldots, T_0 - 1$

$\mathbf{g}_{t_0}^{(0)} = 1/B_0 \sum_{i \in \mathcal{I}_0} \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$ ⇒reference gradient

for $t_1 = 0, \ldots, T_1 - 1$

$\mathbf{g}_{t_1}^{(1)} = 1/B_1 \sum_{i \in \mathcal{I}_1} \nabla f_i(\boldsymbol{\theta}_{t_1}^{(1)}) - \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$

for $t_2 = 1, \ldots, T_2 - 1$

$\mathbf{g}_{t_2}^{(2)} = 1/B_2 \sum_{i \in \mathcal{I}_2} \nabla f_i(\boldsymbol{\theta}_{t_2}^{(2)}) - \nabla f_i(\boldsymbol{\theta}_{t_1}^{(1)})$

$\mathbf{v}_t = \mathbf{g}_{t_0}^{(0)} + \mathbf{g}_{t_1}^{(1)} + \mathbf{g}_{t_2}^{(2)}$, where $t = t_0 T_1 T_2 + t_1 T_2 + t_2$

$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{v}_t$

$\boldsymbol{\theta}_{t_2+1}^{(2)} = \boldsymbol{\theta}_{t+1}$

end for

$\boldsymbol{\theta}_{t_1+1}^{(1)} = \boldsymbol{\theta}_{T_2}^{(2)}$

end for

$\boldsymbol{\theta}_{t_0+1}^{(0)} = \boldsymbol{\theta}_{T_1}^{(1)}$ ⇒reference point

end for

▷ $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2 \subset [n]$: batch sets with sizes B_0, B_1, B_2

Stochastic Nested Variance Reduced Gradient

Reference point $\mathbf{x}_t^{(0)}$
Reference gradient $\mathbf{g}_t^{(0)}$

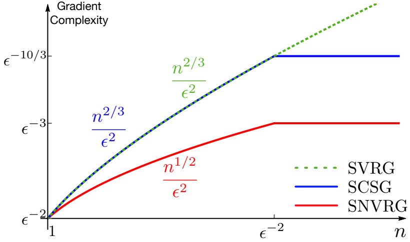
For $t_1 = 1, \ldots, T_1$
 Reference point $\mathbf{x}_t^{(1)}$
 Reference gradient $\mathbf{g}_t^{(1)}$
 ⋮
 ......
 For $t_{K-1} = 1, \ldots, T_{K-1}$
  Reference point $\mathbf{x}_t^{(K-1)}$
  Reference gradient $\mathbf{g}_t^{(K-1)}$
  For $t_K = 1, \ldots, T_K$
   Reference point $\mathbf{x}_t^{(K)}$
   Reference gradient $\mathbf{g}_t^{(K)}$
   update
   $$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \sum_{i=0}^{K} \mathbf{g}_t^{(i)}$$

SNVRG [ZXG18c]

- Gradient complexity:

$$\widetilde{O}\left(\min \left\{ \frac{n^{1/2}}{\epsilon^2}, \frac{1}{\epsilon^3} \right\} \right)$$

- Comparison



Stochastic Path Integrated Differential Estimator

SPIDER [FLLZ18], (SARAH [NLST17] for convex optimization)

for $t_0 = 0, \ldots, T_0 - 1$

  $\mathbf{v}_0 = 1/B_0 \sum_{i \in \mathcal{I}_0} \nabla f_i(\boldsymbol{\theta}_{t_0}^{(0)})$

  for $t_1 = 0, \ldots, T_1 - 1$

    Let $t = t_0 T_1 + t_1$

    $\mathbf{v}_t = \mathbf{v}_{t-1} + 1/B_1 \sum_{i \in \mathcal{I}_1} \nabla f_i(\boldsymbol{\theta}_t) - \nabla f_i(\boldsymbol{\theta}_{t-1})$

    $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{v}_t / \|\mathbf{v}_t\|_2$

  end for

  $\boldsymbol{\theta}_{t_0+1}^{(0)} = \boldsymbol{\theta}_{T_1}$

end for

- Recursive semi-stochastic gradient (Path Integrated)
- All the points in the inner loop are reference points
- A simplified variant without gradient normalization: SpiderBoost [WJZ+18]
- Gradient complexity: $O(n^{1/2}/\epsilon^2)$, the same as SNVRG

Finding a Stationary Point in Nonconvex Optimization

Algorithm	Gradient Complexity
GD	$O(n\epsilon^{-2})$
SGD [GL13]	$O(\epsilon^{-4})$
SVRG [AZH16, RHS+16]	$O(n^{2/3}\epsilon^{-2})$
SCSG [LJCJ17]	$O(\min\{n^{2/3}\epsilon^{-2}, \epsilon^{-10/3}\})$
SPIDER [FLLZ18] SNVRG [ZXG18c]	$\widetilde{O}(\min\{n^{1/2}\epsilon^{-2}, \epsilon^{-3}\})$

Question: can we do better?

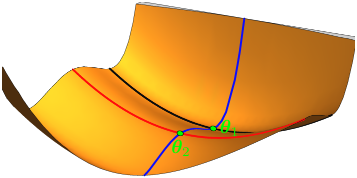
Fundamental Limits:

Lower bound of gradient complexity for finding ϵ-approximate first-order stationary point for nonconvex smooth functions [FLLZ18, ZG19a]:

$$O\left(\frac{\sqrt{n}}{\epsilon^2}\right)$$

Outline

Approximate Second-order Stationary Point



Stationary points (FSP):

- $\boldsymbol{\theta}_1$: local minimum
- $\boldsymbol{\theta}_2$: saddle point

- Second-order Stationary Point (SSP):

$$\|\nabla f(\boldsymbol{\theta})\|_2 = 0, \qquad \lambda_{\min}(\nabla^2 f(\boldsymbol{\theta})) \geq 0$$

- $(\epsilon, \sqrt{\epsilon})$-approximate local minimum:

$$\|\nabla f(\boldsymbol{\theta})\|_2 \leq \epsilon, \qquad \lambda_{\min}(\nabla^2 f(\boldsymbol{\theta})) \geq -\sqrt{\epsilon}$$

Newton Type Methods

Incorporating Hessian information [Ben16, CP77]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \big(\nabla^2 f(\boldsymbol{\theta}_t)\big)^{-1} \nabla f(\boldsymbol{\theta}_t)$$

- Quadratic convergence in convex setting
- Hessian matrix not invertible in nonconvex setting
- $\big(\nabla^2 f(\boldsymbol{\theta}_t)\big)^{-1}$ not well defined

Solution: add regularizer

Cubic Regularized Newton's Methods

Minimize the cubic-regularized second-order Taylor expansion [NP06]

$$\mathbf{h}_t = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \nabla f(\boldsymbol{\theta}_t), \mathbf{h}\rangle + \frac{1}{2}\langle \nabla^2 f(\boldsymbol{\theta}_t)\mathbf{h}, \mathbf{h}\rangle + \frac{M}{6}\|\mathbf{h}\|_2^3,$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbf{h}_t$$

$M > 0$ is a penalty parameter

- $M = 0$, $\Rightarrow$ $\mathbf{h}_t = (\nabla^2 f(\boldsymbol{\theta}_t))^{-1}\nabla f(\boldsymbol{\theta}_t)$, Newton's method
- $\|\mathbf{h}\|_2^3 \rightarrow \|\mathbf{h}\|_2^2$, $\Rightarrow$ $\mathbf{h}_t = (\nabla^2 f(\boldsymbol{\theta}_t) + M\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}_t)$
- $M = 0$, constraint $\{\mathbf{h} : \|\mathbf{h}\|_2 \leq R\}$, $\Rightarrow$ Trust Region method

Oracle Definition

- Second Order Oracle (SO)
  Given an index i and a point $\boldsymbol{\theta}$, one SO call returns a triple:
  $$[f_i(\boldsymbol{\theta}), \nabla f_i(\boldsymbol{\theta}), \nabla^2 f_i(\boldsymbol{\theta})]$$
- Cubic Subproblem Oracle(CSO)
  Given a gradient vector $\mathbf{g}$, a Hessian matrix $\mathbf{H}$ and a positive constant M, one CSO call returns the following minimizer
  $$\mathbf{h}_{\text{sol}} = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \mathbf{g}, \mathbf{h} \rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h} \rangle + \frac{M}{6}\|\mathbf{h}\|_2^3.$$

Cubic Regularized Newton's Methods

Minimize the cubic-regularized second-order Taylor expansion [NP06]

$$\mathbf{h}_t = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \nabla f(\boldsymbol{\theta}_t), \mathbf{h}\rangle + \frac{1}{2}\langle \nabla^2 f(\boldsymbol{\theta}_t)\mathbf{h}, \mathbf{h}\rangle + \frac{M}{6}\|\mathbf{h}\|_2^3,$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbf{h}_t$$

$M > 0$ is a penalty parameter

- Converge to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum within $O(\epsilon^{-3/2})$ iterations
- SO Complexity: $O(n\epsilon^{-3/2})$
- CSO Complexity: $O(\epsilon^{-3/2})$

Subsampled Cubic Regularization Method

Sub-sampled Cubic Regularization (SCR) [KL17, XRKM17]

$$\mathbf{h}_t = \underset{\mathbf{h}\in\mathbb{R}^d}{\operatorname{argmin}} \langle \mathbf{g}_t, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{H}_t\mathbf{h}, \mathbf{h}\rangle + \frac{M}{6}\|\mathbf{h}\|_2^3,$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbf{h}_t$$

Sub-sampled gradient and Hessian matrix:

$$\mathbf{g}_t = 1/B_g \sum_{i\in\mathcal{I}_g} \nabla f_i(\boldsymbol{\theta}_t), \qquad \mathbf{H}_t = 1/B_h \sum_{i\in\mathcal{I}_h} \nabla^2 f_i(\boldsymbol{\theta}_t)$$

- $\mathcal{I}_g, \mathcal{I}_h \subset [n]$ are two index sets with batch sizes B_g and B_h respectively.
- SO complexity: $O(n/\epsilon^{3/2} + 1/\epsilon^{5/2})$ no better than CR
- CSO complexity: $O(1/\epsilon^{3/2})$

Stochastic Variance-Reduced Cubic Regularization

Stochastic Variance-Reduced Cubic (SVRC) [ZXG18d]

for $t_0 = 1, \ldots, T_0$

$\widetilde{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_t$, $\widetilde{\mathbf{g}} = \nabla f(\widetilde{\boldsymbol{\theta}}_0), \widetilde{\mathbf{H}} = \nabla^2 f(\widetilde{\boldsymbol{\theta}}_0)$

for $t_1 = 0, \ldots, T_1 - 1$

$\mathbf{h}_t = \mathrm{argmin}_{\mathbf{h}} \langle \mathbf{v}_t^{s+1}, \mathbf{h} \rangle + 1/2 \langle \mathbf{U}_t^{s+1} \mathbf{h}, \mathbf{h} \rangle + M/6 \|\mathbf{h}\|_2^3$

$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{h}_t$

end for

$\boldsymbol{\theta}_{t+1} = \widetilde{\boldsymbol{\theta}}_m$

end for

Semi-stochastic gradient and Hessian matrix:

$$\mathbf{v}_t = \frac{1}{b_g} \sum_{i_t \in \mathcal{I}_g} \Big(\nabla f_{i_t}(\boldsymbol{\theta}_t) - \nabla f_{i_t}(\widetilde{\boldsymbol{\theta}}_0) + \widetilde{\mathbf{g}} - \big[\nabla^2 f_{i_t}(\widetilde{\boldsymbol{\theta}}_0) - \widetilde{\mathbf{H}}\big][\boldsymbol{\theta}_t - \widetilde{\boldsymbol{\theta}}_0] \Big)$$

$$\mathbf{U}_t = \frac{1}{b_h} \sum_{j_t \in \mathcal{I}_h} \big(\nabla^2 f_{j_t}(\boldsymbol{\theta}_t) - \nabla^2 f_{j_t}(\widetilde{\boldsymbol{\theta}}_0)\big) + \widetilde{\mathbf{H}},$$

Stochastic Variance-Reduced Cubic Regularization

Stochastic Variance-Reduced Cubic (SVRC) [ZXG18d]

To find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum:

- SO complexity:

$$O\left(n + \frac{n^{4/5}}{\epsilon^{3/2}}\right)$$

- CSO complexity: $O(1/\epsilon^{3/2})$

Remarks:

- Cubic sub-problem complexity is the same as previous methods
- Second-order oracle: $(f_i(\boldsymbol{\theta}), \nabla f_i(\boldsymbol{\theta}), \nabla^2 f_i(\boldsymbol{\theta}))$
- Gradient computation: $O(d)$; Hessian matrix computation: $O(d^2)$ $\Rightarrow$ reduce Hessian complexity!

Sample Efficient SVRC

Lite-SVRC: Semi-stochastic gradient and Hessian matrix:

$$\mathbf{v}_t = \frac{1}{b_g} \sum_{i_t \in \mathcal{I}_g} \left(\nabla f_{i_t}(\boldsymbol{\theta}_t) - \nabla f_{i_t}(\widetilde{\boldsymbol{\theta}}_0) + \widetilde{\mathbf{g}} \right)$$

$$\mathbf{U}_t = \frac{1}{b_h} \sum_{j_t \in \mathcal{I}_h} \left(\nabla^2 f_{j_t}(\boldsymbol{\theta}_t) - \nabla^2 f_{j_t}(\widetilde{\boldsymbol{\theta}}_0) \right) + \widetilde{\mathbf{H}}$$

- The same semi-stochastic gradient used in first-order algorithms such as SVRG
- Gradient complexity $O(n/\epsilon^{3/2})$
- Hessian complexity $O(n + n^{2/3}/\epsilon^{2/3})$

Hessian Complexities of Cubic Methods

Algorithm	Gradient Complexity	Hessian Complexity
CR [NP06]	$O(\frac{n}{\epsilon^{3/2}})$	$O(\frac{n}{\epsilon^{3/2}})$
SCR [KL17, XRKM17]	$\widetilde{O}(\frac{n}{\epsilon^{3/2}})$	$\widetilde{O}(\frac{n}{\epsilon^{3/2}})$
SVRC [ZXG18d]	$\widetilde{O}(n + \frac{n^{4/5}}{\epsilon^{3/2}})$	$\widetilde{O}(n + \frac{n^{4/5}}{\epsilon^{3/2}})$
Lite-SVRC [ZXG18b]	$\widetilde{O}(\frac{n}{\epsilon^{3/2}})$	$\widetilde{O}(n + \frac{n^{2/3}}{\epsilon^{3/2}})$
SVRC [WZLL18]	$\widetilde{O}(\frac{n}{\epsilon^{3/2}})$	$\widetilde{O}(n + \frac{n^{2/3}}{\epsilon^{3/2}})$
SRVRC [ZG19b]	$\widetilde{O}(\frac{n}{\epsilon^{3/2}})$	$\widetilde{O}(n + \frac{n^{1/2}}{\epsilon^{3/2}})$

*$\widetilde{O}(\cdot)$ hides logarithm factors.

- Gradient complexity is the same
- Variance reduced Cubic methods have Hessian complexity $O(\sqrt{n}/\epsilon^{3/2})$

Outline

Nonconvexity in Data Mining

Nonconvex Finite-sum Optimization
- Finding First-order Stationary Points
- Finding Local Minima in Nonconvex Optimization
- Finding Local Minima via First-order Algorithms

Structured Nonconvex Problems
- Low Rank Matrix Recovery
- Robust PCA
- Gaussian Graphical Models

References

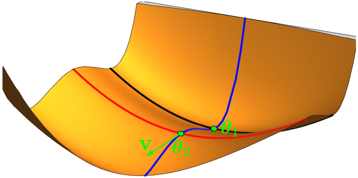
Escape Saddle Points

Cubic Regularization based algorithms (CR, SCR, SVRC, SRVRC, etc.)

- Compute $O(n + \sqrt{n}/\epsilon^{3/2})$ Hessian matrix
- $O(d^2)$ computation per Hessian matrix ☹

Gradient based algorithms (GD, SGD, SVRG, SNVRG, etc.):

- Only need to compute gradient: $O(d)$ per gradient ☺
- Converge to stationary point ⇒ can be a saddle point ☹



- Run SGD/SVRG
- Detect saddle point
  - Yes ⇒ Escape along **v**
- Continue with SGD/SVRG

Finding Local Minima via First-order Oracles

Negative Curvature Direction (NCD): $\mathbf{v}$ such that

$$\mathbf{v}^\top \nabla^2 f(\boldsymbol{\theta})\mathbf{v} \leq -\sqrt{\epsilon}$$

Neon [XRY18], Neon2 [AZL18]

```
for t = 0, 1, ...
    if ||∇f(θ_t)||_2 > ε
        θ_{t+1} = Alg(f, θ_t)                    ▷ SGD/SVRG/SNVRG
    else
        v_t = NCD(f, θ_t)                        ▷ Negative Curvature Direction
        if v_t not found
            return θ_t                           ▷ θ_t is an SSP
        else
            θ_{t+1} = θ_t + η v_t
end for
```

- Turn SGD/SVRG into local minimum finding algorithms
- Saddle points are rare $\Rightarrow$ Hessian matrix computation is not often [YZG17, YXG18]

Complexity for Finding Local Minimum

Algorithm	Gradient Complexity
Pertubed GD [JGN+17]	$O(n\epsilon^{-2})$
Neon+SGD [XRY18, AZL18]	$O(n\epsilon^{-2})$
Neon+SCSG [XRY18, AZL18]	$O(n\epsilon^{-3/2} + n^{2/3}\epsilon^{-2})$
Spider-SFO+ [FLLZ18]	$\widetilde{O}(n^{1/2}\epsilon^{-2})$
Neon+SNVRG [ZXG18a]	

Comparison with CR methods (SRVRC [ZG19b]):

$O(n/\epsilon^{3/2})$ gradient, $O(n^{1/2}/\epsilon^{3/2})$ Hessian

Outline

Nonconvexity in Data Mining

Nonconvex Finite-sum Optimization
- Finding First-order Stationary Points
- Finding Local Minima in Nonconvex Optimization
- Finding Local Minima via First-order Algorithms

Structured Nonconvex Problems
- Low Rank Matrix Recovery
- Robust PCA
- Gaussian Graphical Models

References

Structured Nonconvex Problems

Nonconvex optimization with special structures:

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

- f can have special structures: low-rankness, sparsity.
- Linear rate can be achieved!
- Global minimum can be reached!

Outline

Low Rank Matrix Recovery

Goal: recover a low rank matrix $\mathbf{X}^* \in \mathbb{R}^{d_1 \times d_2}$ with $\mathrm{rank}(\mathbf{X}^*) = r$.

Nonconvex optimization formulation:

$$\min_{\mathbf{U}\in\mathbb{R}^{d_1\times r},\mathbf{V}\in\mathbb{R}^{d_2\times r}} \mathcal{L}_n(\mathbf{U}\mathbf{V}^\top), \text{ subject to } \mathbf{U}\in\mathcal{C}_1, \mathbf{V}\in\mathcal{C}_2,$$

- $\mathcal{L}_n$ is a nonconvex function based on n data observations.
- $\mathcal{C}_1, \mathcal{C}_2$: feasible sets induced by $\mathbf{X}^*$

Specific implications: matrix sensing, matrix completion, and one-bit matrix completion

E.g. the objective function for matrix completion

$$\min_{\mathbf{U}\in\mathbb{R}^{d_1\times r},\mathbf{V}\in\mathbb{R}^{d_2\times r}} \mathcal{L}_\Omega(\mathbf{U}\mathbf{V}^\top) := \frac{1}{2p}\sum_{(j,k)\in\Omega}(\mathbf{U}_{j*}\mathbf{V}_{k*}^\top - Y_{jk})^2$$

Alternating Projected Gradient Pursuit (Main Stage)

$(\mathbf{U}, \mathbf{V})$ is the solution $\Rightarrow (c\mathbf{U}, 1/c\mathbf{V})$ is the solution $\forall c \neq 0$

- Add a regularizer $\mathcal{L}_n(\mathbf{U}\mathbf{V}^\top) + \mathcal{R}(\mathbf{U}, \mathbf{V})$, e.g. $\mathcal{R}(\mathbf{U}, \mathbf{V}) = \|\mathbf{U}^\top\mathbf{U} - \mathbf{V}^\top\mathbf{V}\|_2^2/8$

Alternating projected gradient descent:

$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathcal{C}_1}\Big(\mathbf{U}^t - \eta\big(\nabla_{\mathbf{U}}\mathcal{L}_n(\mathbf{U}^t\mathbf{V}^{t\top}) + \nabla_{\mathbf{U}}\mathcal{R}(\mathbf{U}^t, \mathbf{V}^t)\big)\Big)$$
$$\mathbf{V}^{t+1} = \mathcal{P}_{\mathcal{C}_2}\Big(\mathbf{V}^t - \eta\big(\nabla_{\mathbf{V}}\mathcal{L}_n(\mathbf{U}^t\mathbf{V}^{t\top}) + \nabla_{\mathbf{V}}\mathcal{R}(\mathbf{U}^t, \mathbf{V}^t)\big)\Big),$$

- $\mathcal{P}$: projection operator
- For matrix completion, $\mathcal{C}_i = \{\mathbf{A} \in \mathbb{R}^{d_i \times r} \mid \|\mathbf{A}\|_{2,\infty} \leq \alpha\}$, thus the projection step is very efficient.

Singular Value Projection (Initialization Stage)

- Initialization matters: need to ensure $\mathbf{U}^0\mathbf{V}^{0\top}$ is close to $\mathbf{X}^*$.
- Singular Value Projection:

$$\mathbf{X}_s = \mathcal{P}_r\big(\mathbf{X}_{s-1} - \tau\nabla\mathcal{L}_n(\mathbf{X}_{s-1})\big)$$
$$\big[\bar{\mathbf{U}}^0, \boldsymbol{\Sigma}^0, \bar{\mathbf{V}}^0\big] = \mathrm{SVD}_r(\mathbf{X}_S) \text{ (after } S \text{ steps)}$$

- Initial estimator: $\mathbf{U}^0 = \bar{\mathbf{U}}^0\big(\boldsymbol{\Sigma}^0\big)^{1/2}, \mathbf{V}^0 = \bar{\mathbf{V}}^0\big(\boldsymbol{\Sigma}^0\big)^{1/2}$
- Singular value projection plus alternating projected gradient pursuit allow us to find $\mathbf{X}^*$ in a linear convergence rate[WZG17a]

Stochastic Variance-Reduced Gradient Descent

Loss function

- $F_N(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{n} \mathcal{L}_i(\mathbf{U}\mathbf{V}^\top) + \mathcal{R}(\mathbf{U},\mathbf{V})$, where $\mathcal{L}_i(\mathbf{U}\mathbf{V}^\top) = \sum_{j=1}^{b} \ell_i(\mathbf{U}\mathbf{V}^\top)$

Low rank stochastic variance-reduced gradient (LRSVRG) [WZG17a]:

for $t = 0, 1, \ldots$
  if $t\%m == 0$
    $\widetilde{\mathbf{X}} = \mathbf{U}^t\mathbf{V}^{t\top}$
  Randomly pick $i_t \in \{1, 2, \ldots, n\}$
  $\mathbf{U}^{t+1} = \mathcal{P}_{\mathcal{C}_1}\Big(\mathbf{U}^t - \eta\big(\nabla_{\mathbf{U}} F_{i_t}(\mathbf{U}^t,\mathbf{V}^t) - \nabla\mathcal{L}_{i_t}(\widetilde{\mathbf{X}})\mathbf{V}^t + \nabla\mathcal{L}_N(\widetilde{\mathbf{X}})\mathbf{V}^t\big)\Big)$
  $\mathbf{V}^{t+1} = \mathcal{P}_{\mathcal{C}_2}\Big(\mathbf{V}^t - \eta\big(\nabla_{\mathbf{V}} F_{i_t}(\mathbf{U}^t,\mathbf{V}^t) - \nabla\mathcal{L}_{i_t}(\widetilde{\mathbf{X}})^\top\mathbf{U}^t + \nabla\mathcal{L}_N(\widetilde{\mathbf{X}})^\top\mathbf{U}^t\big)\Big)$
end for

Computational Complexity

Computational complexity for achieving ϵ accuracy $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F \leq \epsilon$ for matrix completion

- Convex relaxation [SRJ04, CT10, RT+11, NW12, GHG16]:

$$O(d^3/\epsilon)$$

- Nonconvex GD [CW15, BKS16, WZG17a, ZL16, PKCS18]:

$$O\big(N\kappa r^3 d \log(1/\epsilon)\big)$$

- Nonconvex LRSVRG [WZG17b]:

$$O\big((N + \kappa^2 b) r^3 d \log(1/\epsilon)\big)$$

Outline

Robust PCA

Goal: recover a low rank matrix and a sparse matrix $\mathbf{X}^*, \mathbf{S}^* \in \mathbb{R}^{d_1 \times d_2}$.

The optimization problem for robust matrix recovery:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{S}} \mathcal{L}_n(\mathbf{U}\mathbf{V}^\top + \mathbf{S}), \text{ subject to } \mathbf{U} \in \mathcal{C}_1, \ \mathbf{V} \in \mathcal{C}_2, \ \mathbf{S} \in \mathcal{K}$$

- $\mathcal{L}_n$ is a nonconvex function
- $\mathcal{C}_1, \mathcal{C}_2$: feasible sets induced by $\mathbf{X}^*$
- $\mathcal{K}$: sparsity induced feasible set

Specific implications: Robust PCA, robust matrix sensing, robust one-bit matrix completion
[NNS+14, CW15, GWL16, YPCC16, CGJ17, ZWG18]

Double Thresholding based Algorithm (Main Stage)

- Introduce the same regularizer $\mathcal{L}_n(\mathbf{U}\mathbf{V}^\top + \mathbf{S}) + \mathcal{R}(\mathbf{U}, \mathbf{V})$.
- Double thresholding based gradient descent:

$$\mathbf{S}^{t+1} = \mathcal{T}_\beta \circ \mathcal{H}_s\big(\mathbf{S}^t - \tau \nabla_{\mathbf{S}} \mathcal{L}_n(\mathbf{U}^t\mathbf{V}^{t\top} + \mathbf{S}^t)\big)$$
$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathcal{C}_1}\Big(\mathbf{U}^t - \eta\big(\nabla_{\mathbf{U}} \mathcal{L}_n(\mathbf{U}^t\mathbf{V}^{t\top} + \mathbf{S}^t) + \nabla_{\mathbf{U}} \mathcal{R}(\mathbf{U}^t, \mathbf{V}^t)\big)\Big)$$
$$\mathbf{V}^{t+1} = \mathcal{P}_{\mathcal{C}_2}\Big(\mathbf{V}^t - \eta\big(\nabla_{\mathbf{V}} \mathcal{L}_n(\mathbf{U}^t\mathbf{V}^{t\top} + \mathbf{S}^t) + \nabla_{\mathbf{V}} \mathcal{R}(\mathbf{U}^t, \mathbf{V}^t)\big)\Big)$$

- Hard thresholding operator $\mathcal{H}_s$: set all but the largest s elements in magnitude to zero.
- Truncation operator $\mathcal{T}_\beta$:

$$[\mathcal{T}_\theta(\mathbf{S})]_{ij} := \begin{cases} S_{ij}, & \text{if } |S_{ij}| \geq |S_{i,*}^{(\theta d_2)}| \text{ and } |S_{ij}| \geq |S_{*,j}^{(\theta d_1)}|, \\ 0, & \text{otherwise.} \end{cases}$$

Singular Value Projection with Hard Thresholding (Initialization Stage)

- Initialization matters: need to ensure initial estimators are close to $\mathbf{X}^*, \mathbf{S}^*$
- Singular value projection for low rank structure.
- Hard thresholding for sparse structure:

$$\mathbf{S}_{\ell+1} = \mathcal{H}_s(\mathbf{S}_\ell - \tau' \nabla_{\mathbf{S}} \mathcal{L}_n(\mathbf{X}_\ell + \mathbf{S}_\ell)).$$

- The whole algorithm can find $\mathbf{X}^*$ and $\mathbf{S}^*$ with a linear convergence rate [ZWG18]

Efficiency and Robustness

Fully observed Robust PCA

Algorithm	Computational Complexity	Robustness
Convex Relaxation [XCS10]	$O(d^3/\epsilon)$	$O(1/r)$
Nonconvex RPCA [NNS+14]	$O(r^2d^2\log(1/\epsilon))$	$O(1/r)$
Alt RPCA [GWL16]	$O(rd^2\log(1/\epsilon))$	$O(1/d)$
Fast RPCA [YPCC16]	$O(rd^2\log(1/\epsilon))$	$O(1/r^{1.5})$
DT RPCA [ZWG18]	$O(rd^2\log(1/\epsilon))$	$O(1/r)$

*Robustness means the fraction of corrupted observations

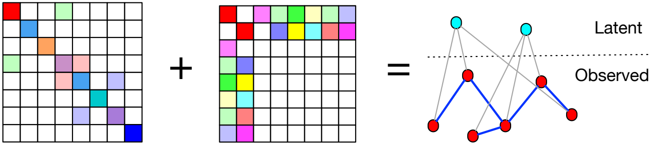
Partially observed Robust PCA

Algorithm	Sample Complexity	Computational Complexity
Fast RPCA [YPCC16]	$O(r^2d\log d)$	$\widetilde{O}(r^4d\log d)$
PG-RMC [CGJ17]	$\widetilde{O}(r^2d\log^2 d)$	$\widetilde{O}(r^3d\log^2 d)$
DT RPCA [ZWG18]	$O(rd\log d)$	$\widetilde{O}(r^3d\log d)$

Outline

Latent Variable Gaussian Graphical Models

Latent Variable Gaussian Graphical Model (LVGGM) [XMG17]



Sparse plus Low-rank Matrix

Gene Regulatory Network

- Data $\boldsymbol{X} = (\boldsymbol{X}^O, \boldsymbol{X}^L) \sim N(\mathbf{0}, \widetilde{\boldsymbol{\Omega}}^*) \Rightarrow \boldsymbol{X}^O \sim N(\mathbf{0}, \boldsymbol{\Omega}^{*-1})$
- $\boldsymbol{\Omega}^* = \mathbf{S}^* + \mathbf{L}^*$: sparse + low rank

Optimization problem:

$$\min_{\mathbf{S},\mathbf{Z}} \quad q_n(\mathbf{S}, \mathbf{Z}) = \operatorname{tr}\left[\widehat{\Sigma}\left(\mathbf{S} + \mathbf{Z}\mathbf{Z}^\top\right)\right] - \log\left|\mathbf{S} + \mathbf{Z}\mathbf{Z}^\top\right|, \quad \text{s.t. } \|\mathbf{S}\|_{0,0} \leq s,$$

- $\widehat{\Sigma} = 1/n \sum_i \boldsymbol{X}_i \boldsymbol{X}_i^\top$: sample covariance matrix
- Negative log-likelihood function. Nonconvex in $(\mathbf{S}, \mathbf{Z})$

Latent Variable Gaussian Graphical Models

Initialization: one step SVD [YPCC16]

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i^\top \quad \Rightarrow \quad \mathbf{S}^{(0)} = \mathcal{H}_s(\widehat{\Sigma}^{-1})$$

$$\text{SVD}(\widehat{\Sigma}^{-1} - \mathbf{S}^{(0)}) = \mathbf{UDU}^\top \quad \Rightarrow \quad \mathbf{Z}^{(0)} = \mathbf{UD}_r^{1/2}$$

Alternating Gradient Descent (AltGD):

$$\mathbf{S}^{t+1} = \mathcal{H}_s\Big(\mathbf{S}^t - \eta \nabla_{\mathbf{S}} q_n(\mathbf{S}^t, \mathbf{Z}^t)\Big)$$

$$\mathbf{Z}^{t+1} = \mathbf{Z}^t - \eta' \nabla_{\mathbf{Z}} q_n(\mathbf{S}^t, \mathbf{Z}^t)$$

- $\mathcal{H}_s$: preserve the s largest magnitudes

Linear convergence to the true parameter up to statistical error [XMG17]

Latent Variable Gaussian Graphical Models

Setting	Method	$\|\boldsymbol{\Omega}^T - \boldsymbol{\Omega}^*\|_F$	Time (s)
$d = 100, r = 2, n = 2000$	PPA	0.7350±0.0359	1.1610
	ADMM	0.7563±0.0298	1.1120
	AltGD	0.6236±0.0669	0.0250
$d = 500, r = 5, n = 10000$	PPA	0.9813±0.0192	35.7220
	ADMM	1.0610±0.0134	25.8010
	AltGD	0.8210±0.0143	0.4800
$d = 1000, r = 8, n = 2.5 \times 10^4$	PPA	1.1639±0.0179	356.7360
	ADMM	1.1869±0.0254	156.5550
	AltGD	0.9021±0.0244	7.4740
$d = 5000, r = 10, n = 2 \times 10^5$	PPA	1.4824±0.0120	33522.0200
	ADMM	1.5012±0.0240	21090.7900
	AltGD	1.3449±0.0084	445.6730

More than 50× speedup than convex methods!

Outline

Summary

General finite-sum optimization

- Find an ϵ-approximate first-order stationary point
- Find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum
- sublinear convergence rate

Structured nonconvex problems

- Low rank matrix recovery, robust PCA, latent variable Gaussian graphical models, etc.
- With a proper initialization, linear convergence to global minimum

Slides, video and other information are available on
https://sites.google.com/view/sdm2019-nonconvex

Outline

References I

Zeyuan Allen-Zhu and Elad Hazan.
Variance reduction for faster non-convex optimization.
In *International Conference on Machine Learning*, pages 699–707, 2016.

Zeyuan Allen-Zhu and Yuanzhi Li.
Neon2: Finding local minima via first-order oracles.
In *Advances in Neural Information Processing Systems*, pages 3720–3730, 2018.

Albert A Bennett.
Newton's method in general analysis.
Proceedings of the National Academy of Sciences, 2(10):592–598, 1916.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi.
Dropping convexity for faster semi-definite optimization.
In *Conference on Learning Theory*, pages 530–582, 2016.

Jinghui Chen and Quanquan Gu.
Closing the generalization gap of adaptive gradient methods in training deep neural networks.
arXiv preprint arXiv:1806.06763, 2018.

Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain.
Nearly optimal robust matrix completion.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 797–805. JMLR. org, 2017.

Ruth F Curtain and Anthony J Pritchard.
Functional analysis in modern applied mathematics, volume 132.
Academic press, 1977.

References II

Emmanuel J Candès and Terence Tao.
The power of convex relaxation: Near-optimal matrix completion.
Information Theory, IEEE Transactions on, 56(5):2053–2080, 2010.

Yudong Chen and Martin J Wainwright.
Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees.
arXiv preprint arXiv:1509.03025, 2015.

John Duchi, Elad Hazan, and Yoram Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang.
Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator.
In *Advances in Neural Information Processing Systems*, pages 686–696, 2018.

Huan Gui, Jiawei Han, and Quanquan Gu.
Towards faster rates and oracle property for low-rank matrix estimation.
In *International Conference on Machine Learning*, pages 2300–2309, 2016.

Saeed Ghadimi and Guanghui Lan.
Stochastic first-and zeroth-order methods for nonconvex stochastic programming.
SIAM Journal on Optimization, 23(4):2341–2368, 2013.

Quanquan Gu, Zhaoran Wang Wang, and Han Liu.
Low-rank and sparse structure pursuit via alternating minimization.
In *Artificial Intelligence and Statistics*, pages 600–609, 2016.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky.
Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.
2012.

References III

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan.
How to escape saddle points efficiently.
In *International Conference on Machine Learning*, pages 1724–1732, 2017.

Rie Johnson and Tong Zhang.
Accelerating stochastic gradient descent using predictive variance reduction.
In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Diederik P. Kingma and Jimmy Ba.
Adam: A method for stochastic optimization.
CoRR, abs/1412.6980, 2014.

Jonas Moritz Kohler and Aurelien Lucchi.
Sub-sampled cubic regularization for non-convex optimization.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1895–1904. PMLR, 2017.

Thomas N Kipf and Max Welling.
Semi-supervised classification with graph convolutional networks.
arXiv preprint arXiv:1609.02907, 2016.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan.
Non-convex finite-sum optimization via scsg methods.
In *Advances in Neural Information Processing Systems*, pages 2348–2358, 2017.

Zhize Li and Jian Li.
A simple proximal stochastic gradient method for nonsmooth nonconvex optimization.
In *Advances in Neural Information Processing Systems*, pages 5564–5574, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
In *Advances in neural information processing systems*, pages 3111–3119, 2013.

References IV

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč.
Sarah: A novel method for machine learning problems using stochastic recursive gradient.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621. JMLR. org, 2017.

Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain.
Non-convex robust pca.
In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

Yurii Nesterov and B. T. Polyak.
Cubic regularization of newton method and its global performance.
Mathematical Programming, 108(1):177–205, 2006.

Sahand Negahban and Martin J Wainwright.
Restricted strong convexity and weighted matrix completion: Optimal bounds with noise.
Journal of Machine Learning Research, 13(May):1665–1697, 2012.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi.
Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably.
SIAM Journal on Imaging Sciences, 11(4):2165–2204, 2018.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola.
Stochastic variance reduction for nonconvex optimization.
In *International Conference on Machine Learning*, pages 314–323, 2016.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar.
On the convergence of adam and beyond.
In *International Conference on Learning Representations*, 2018.

Angelika Rohde, Alexandre B Tsybakov, et al.
Estimation of high-dimensional low-rank matrices.
The Annals of Statistics, 39(2):887–930, 2011.

References V

Nathan Srebro, Jason Rennie, and Tommi S Jaakkola.
Maximum-margin matrix factorization.
In *Advances in neural information processing systems*, pages 1329–1336, 2004.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh.
Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization.
arXiv preprint arXiv:1810.10690, 2018.

Lingxiao Wang, Xiao Zhang, and Quanquan Gu.
A unified computational and statistical framework for nonconvex low-rank matrix estimation.
In *Artificial Intelligence and Statistics*, pages 981–990, 2017.

Lingxiao Wang, Xiao Zhang, and Quanquan Gu.
A unified variance reduction-based framework for nonconvex low-rank matrix recovery.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3712–3721. JMLR. org, 2017.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan.
Stochastic variance-reduced cubic regularization for nonconvex optimization.
arXiv preprint arXiv:1802.07372, 2018.

Huan Xu, Constantine Caramanis, and Sujay Sanghavi.
Robust pca via outlier pursuit.
In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

Pan Xu, Jian Ma, and Quanquan Gu.
Speeding up latent variable gaussian graphical model estimation via nonconvex optimization.
In *Advances in Neural Information Processing Systems*, pages 1933–1944, 2017.

Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney.
Newton-type methods for non-convex optimization under inexact hessian information.
arXiv preprint arXiv:1708.07164, 2017.

References VI

Yi Xu, Jing Rong, and Tianbao Yang.
First-order stochastic algorithms for escaping from saddle points in almost linear time.
In *Advances in Neural Information Processing Systems*, pages 5531–5541, 2018.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis.
Fast algorithms for robust pca via gradient descent.
In *Advances in neural information processing systems*, pages 4152–4160, 2016.

Yaodong Yu, Pan Xu, and Quanquan Gu.
Third-order smoothness helps: Faster stochastic optimization algorithms for finding local minima.
In *Advances in Neural Information Processing Systems*, pages 4526–4536, 2018.

Yaodong Yu, Difan Zou, and Quanquan Gu.
Saving gradient and negative curvature computations: Finding local minima more efficiently.
arXiv preprint arXiv:1712.03950, 2017.

Dongruo Zhou and Quanquan Gu.
Lower bounds for smooth nonconvex finite-sum optimization.
arXiv preprint arXiv:1901.11224, 2019.

Dongruo Zhou and Quanquan Gu.
Stochastic recursive variance-reduced cubic regularization methods.
arXiv preprint arXiv:1901.11518, 2019.

Qinqing Zheng and John Lafferty.
Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent.
arXiv preprint arXiv:1605.07051, 2016.

Xiao Zhang, Lingxiao Wang, and Quanquan Gu.
A unified framework for nonconvex low-rank plus sparse matrix recovery.
In *International Conference on Artificial Intelligence and Statistics*, pages 1097–1107, 2018.

References VII

Dongruo Zhou, Pan Xu, and Quanquan Gu.
Finding local minima via stochastic nested variance reduction.
arXiv preprint arXiv:1806.08782, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu.
Sample efficient stochastic variance-reduced cubic regularization method.
arXiv preprint arXiv:1811.11989, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu.
Stochastic nested variance reduced gradient descent for nonconvex optimization.
In *Advances in Neural Information Processing Systems*, pages 3922–3933, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu.
Stochastic variance-reduced cubic regularized Newton methods.
In *Proceedings of the 35th International Conference on Machine Learning*, pages 5990–5999. PMLR, 2018.