Fast and Simple Optimization for Non-Lipschitz Poisson Likelihood Models

Niao He

November 17, 2016
A Simple Gaussian Noise Model

Given the observations \((a_i, b_i), i = 1, \ldots, m,\)

\[ b_i \sim \text{Gaussian}(a_i^T x, \sigma^2) \]

- Penalized MLE reduces to solving least-squares regression

\[
\min_x \frac{1}{m} \sum_{i=1}^{m} (a_i^T x - b_i)^2 + h(x)
\]

- Significant body of work: proximal/stochastic/incremental gradient methods, e.g. PG, FISTA, SGD, SVRG, SAGA, etc.
- Rich softwares available: scikit-learn (SGD), glmnet (cyclic CD), etc.
- Most first-order algorithms rely on the smoothness of the loss function, i.e. the Lipschitz differentiability.
A Simple Poisson Noise Model

Given the observations \((a_i, b_i), i = 1, \ldots, m,\)

\[b_i \sim \text{Poisson}(a_i^T x)\]

- Penalized MLE reduces to solving Poisson regression

\[
\min_x \frac{1}{m} \sum_{i=1}^{m} [a_i^T x - b_i \log(a_i^T x)] + h(x)
\]

- **Fundamental difficulty**: loss function is not even Lipschitz continuous!

- Fewer work and softwares are available.
Poisson Likelihood Models

The query "Poisson linear" yields more than 1M hits on GoogleScholar.

- Traditional imaging applications
  - Positron emission tomography (PET)
  - Poisson compressive sensing for solar flare image reconstruction, confocal microscopy image deblurring

- Modern diffusion network applications
  - Hawkes/Cox models for estimating social infectivity, gene regulation, disease diffusion, etc.
  - Hawkes models for time-sensitive recommendation systems
Example I: PET Imaging

- The events (photon counts) registered by the $m$ detectors follow
  \[ w_i \sim \text{Poisson}([Ax]_i), \; i = 1, \ldots, m \]

- $x$ is the density of radioactivity of an object with $n$ voxels

- $A$ is the likelihood matrix known from the geometry of detector

- To recover the density $x$ corresponds to solving the convex optimization problem
  \[ \min_{x \in \mathbb{R}^n_+} \sum_{i=1}^m ([Ax]_i - w_i \log([Ax]_i)). \]

References: [Ben-Tal et al., 2001; Sra et al., 2009; Harmany et al., 2012]
Example II: Network Estimation

- Self-exciting Hawkes process has been widely used to discover the latent influences and hidden network among social communities.

- For each user $u$, the intensity function $\lambda_u(t)$ is given by

$$\lambda_u(x, X|t) = x_u + \sum_{(u', t') \in O, t' < t} X_{u,u'}g(t - t').$$

- $x$ is the base intensity
- $X$ is the influence matrix
- $O = \{(u_i, t_i)\}_{i=1}^m$ are the observations
- $g(t) = ce^{-ct}$ is the triggering kernel

- The latent influence can be learned via the Poisson likelihood model

$$\min_{x \geq 0, X \geq 0} L(\lambda(x, X)) + \lambda_1\|X\|_1 + \lambda_2\|X\|_{nuc}$$

where $L(\lambda) = \int_0^T \lambda(t)dt - \sum_{i=1}^m \log(\lambda(t_i)).$

References: [Mohler et al., 2012; Zhou et al., 2013; Tomoharu et al., 2013]
Example III: Temporal Recommendation System

- Temporal point process has been recently used to incorporate temporal behaviors of customers into recommendation systems.

- For user-item pair \((u, i)\), the intensity function \(\lambda_{u,i}(t)\) is given by

\[
\lambda(X_1, X_2|t) = X_1^{u,i} + X_2^{u,i} \sum_{t' \in O^{u,i}: t' < t} g(t - t')
\]

  - \(X_1\) is the base intensity matrix for all pair
  - \(X_2\) is the self-exciting coefficient for all pair
  - \(O^{(u,i)}\) are the observations for pair \((u, i)\)

- The temporal behavior can be learned via the Poisson likelihood model

\[
\min_{X_1 \geq 0, X_2 \geq 0} L(\lambda(X_1, X_2)) + \lambda_1 \|X_1\|_{nuc} + \lambda_2 \|X_2\|_{nuc}
\]

References: [Du et al., 2015; Kapoor et al., 2015]
The Situation

- Previous objectives can be written in the compact form:

\[
\min_{x \in \mathbb{R}^n_+} L(x) + h(x), \quad \text{with } L(x) = s^T x - \sum_{i=1}^{m} c_i \log(a_i^T x)
\]

- \(s, c\) and \(a_i, i = 1, \ldots m\) are given nonnegatives

- \(h\) is convex, proximal-friendly, i.e. the proximal operator

\[
\text{Prox}^h_{x_0}(\xi) := \arg\min_{x \in \mathbb{R}^n_+} \{ V_\omega(x, x_0) + \langle \xi, x \rangle + h(x) \}
\]

is easy to solve, \(V_\omega(x, x_0) = \omega(x) - \omega(x_0) - \nabla \omega(x_0)^T (x - x_0)\).
The Situation

- Previous objectives can be written in the compact form:

\[
\min_{x \in \mathbb{R}_+^n} L(x) + h(x), \quad \text{with} \quad L(x) = s^T x - \sum_{i=1}^{m} c_i \log(a_i^T x)
\]

- \(s, c\) and \(a_i, i = 1, \ldots, m\) are given nonnegatives
- \(h\) is convex, proximal-friendly, i.e. the proximal operator

\[
\text{Prox}_{x_0}^h(\xi) := \arg\min_{x \in \mathbb{R}_+^n} \{ V_\omega(x, x_0) + \langle \xi, x \rangle + h(x) \}
\]

is easy to solve, \(V_\omega(x, x_0) = \omega(x) - \omega(x_0) - \nabla \omega(x_0)^T (x - x_0)\).

- Few work discussed the non-Lipschitzianity of such objectives:
  - [Harmany et al., 2012]: perturb by \(\log(a_i^T x + \epsilon)\) and apply APG
    - smoothness constant \(L \sim O(1/\epsilon^2)\) too large
  - [Sra et al., 2008]: add constraints \(a_i^T x \geq \epsilon\) and apply projected GD
    - projection could be expensive
  - [Ben-Tal et al., 2001]: treat as nonsmooth problem and apply MD
    - slow convergence \(O(1/\sqrt{t})\)
  - [Teboulle et al., 2016]: NoLips algorithm, \(O(1/t)\) rate
Overview

We introduce a new family of optimization algorithms that
- are simple and fast
- deal with Poisson-likelihood objectives in a principled way
- outperform competing algorithms in practice

<table>
<thead>
<tr>
<th>algorithm</th>
<th>type</th>
<th>guarantee</th>
<th>geometry</th>
<th>convergence</th>
<th>constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>batch</td>
<td>primal</td>
<td>non-Euclidean</td>
<td>$O(M/\sqrt{t})$</td>
<td>$M$ unbounded</td>
</tr>
<tr>
<td>APG</td>
<td>batch</td>
<td>primal</td>
<td>Euclidean</td>
<td>$O(L/t^2)$</td>
<td>$L$ unbounded</td>
</tr>
<tr>
<td>CMP</td>
<td>batch</td>
<td>primal and dual</td>
<td>non-Euclidean</td>
<td>$O(\mathcal{L}/t)$</td>
<td>$\mathcal{L}$ bounded</td>
</tr>
<tr>
<td>RB-CMP</td>
<td>stoch</td>
<td>sad. point gap</td>
<td>non-Euclidean</td>
<td>$O(\mathcal{L}/t)$</td>
<td>$\mathcal{L}$ bounded</td>
</tr>
</tbody>
</table>

Table: Convergence rates of different algorithms for penalized Poisson regression
The Crux: Saddle Point Reformulation

Problem of Interest

\[
\min_{x \in \mathbb{R}^n_+} L(x) + h(x), \text{ with } L(x) = s^T x - \sum_{i=1}^{m} c_i \log(a_i^T x)
\]

Key Observations

- Saddle point representation
\[
\min_{x \in \mathbb{R}^n_+} \max_{y \in \mathbb{R}^m_+} \phi(x, y) := s^T x - y^T A x + \sum_{i=1}^{m} c_i \log(y_i) + h(x)
\]

- Proximal-Friendliness of \( \sum_{i=1}^{m} c_i \log(y_i) \)
\[
y^+ = \arg\min_{y \in \mathbb{R}_+^m} \left\{ \frac{1}{2} \|y\|_2^2 + \langle \eta, y \rangle - \beta \sum_{i=1}^{m} c_i \log(y_i) \right\} = \left[ (-\eta_i + \sqrt{\eta_i^2 + 4/\beta c_i})/2 \right]_{i=1, \ldots, m}
\]

1. A similar technique is used in [Yanez and Bach, 2014] for nonconvex NMF with KL-divergence.
Composite Saddle Point Problem

\[
\min_{u_1 \in U_1} \max_{u_2 \in U_2} \Phi(u_1, u_2) := [\phi(u_1, u_2) + \psi_1(u_1) - \psi_2(u_2)]
\]

- \(\phi(u_1, u_2)\) is smooth convex-concave
- \(\psi_1(u_1), \psi_2(u_2)\) are convex and proximal-friendly
- \(U_1, U_2\) are closed convex sets

Related Work

- Extragradient methods: HPE framework [Tseng, 2009; He & Monterio, 2016], composite Mirror Prox algorithm [He et al, 2015]
Composite Mirror Prox Algorithm

**Composite Mirror Prox (CMP)**

**Input:** $u_i^1 \in U_i$, $\alpha_i > 0$, $i = 1, 2$ and $\gamma_t > 0$

for $t = 1, 2, \ldots, T$ do

\[
\hat{u}_i^t = \min_{u_i \in U_i} \{ \alpha_i V_i(u_i, u_i^t) + \langle \gamma_t \nabla_i \phi(u_i^t), u_i \rangle + \gamma_t \Psi_i(u_i) \}, \quad i = 1, 2
\]

\[
u_i^{t+1} = \min_{u_i \in U_i} \{ \alpha_i V_i(u_i, u_i^t) + \langle \gamma_t \nabla_i \phi(\hat{u}_i^t), u_i \rangle + \gamma_t \Psi_i(u_i) \}, \quad i = 1, 2
\]

end for

Output $u_{i,T} = (\sum_{t=1}^{T} \gamma_t \hat{u}_i^t) / (\sum_{t=1}^{T} \gamma_t)$, $i = 1, 2$.

**Proposition [H.-Nemirovski-Juditsky, 2015]**

Assume $\phi$ is $\mathcal{L}$-Lipchitz differentiable and stepsize $0 < \gamma_t \leq \mathcal{L}^{-1}$, we have

\[
\forall u = [u_1, u_2] \in U : \Phi(u_1, T, u_2) - \Phi(u_1, u_2, T) \leq \frac{\mathcal{L} \cdot \Theta[U]}{T}
\]

where $\Theta[U] = \max_{u \in U} \sum_{i=1}^{2} V_i(u_i, u_i^1)$.
Back to Poisson Likelihood Models

Saddle Point Reformulation

\[
\min_{x \in \mathbb{R}^n_+} \max_{y \in \mathbb{R}^m_+} \phi(x, y) := s^T x - y^T Ax + \sum_{i=1}^{m} c_i \log(y_i) + h(x)
\]

The CMP algorithm enjoys several desiderata when solving the Poisson likelihood models:

- Efficient iteration cost
- Theoretically grounded, we have\( f(x_T) - f_* \leq O\left(\frac{\|A\|_x \to 2}{T}\right)\)
- Self-tuned stepsize without requiring a priori Lipschitz constant
- Versatile in the choice of Bregman distance

**CMP for Penalized Poisson Models**

Input: \(x^1 \in \mathbb{R}^n_+, y^1 \in \mathbb{R}^m_+, \alpha, \gamma_t > 0\)

for \(t = 1, 2, \ldots, T\) do

\[
\hat{x}^t = \text{Prox}_{x^t}^{\gamma_t h/\alpha} \left(\gamma_t (s - A^T y^t) / \alpha\right)
\]

\[
y^t_i = Q^{\gamma_t} \left(\gamma_t (a_i^T x^t - y^t_i)\right), \forall i
\]

\[
x^{t+1} = \text{Prox}_{x^t}^{\gamma_t h/\alpha} \left(\gamma_t (s - A^T \hat{y}^t) / \alpha\right)
\]

\[
y^{t+1}_i = Q^{\gamma_t} \left(\gamma_t (a_i^T x^{t+1} - \hat{y}^{t+1}_i)\right), \forall i
\]

end for
Application I: Positron Emission Tomography

\[
\min_{x \in \mathbb{R}_+^n} \sum_{i=1}^{m} \left( [Ax]_i - w_i \log([Ax]_i) \right).
\]

Shepp-Logan image size $256 \times 256$, matrix $A$ is of size $43530 \times 65536$

\begin{figure}[h]
\centering
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{figure1a}
\caption{CMP under proximal setups}
\end{subfigure}
\hfill
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{figure1b}
\caption{CMP vs. All}
\end{subfigure}
\end{figure}
Application I: Positron Emission Tomography

Reconstruction for MRI brain image
Application II: Temporal Recommendation System

\[
\min_{X_1 \geq 0, X_2 \geq 0} L(X_1, X_2) + \lambda_1 \|X_1\|_{\text{nuc}} + \lambda_2 \|X_2\|_{\text{nuc}}
\]

where \( L(X_1, X_2) = \frac{1}{|O|} \sum_{T^u,i \in O} \ell(T^u,i|X_1, X_2) \) is the log-likelihood.

<table>
<thead>
<tr>
<th>dataset</th>
<th>user</th>
<th>item</th>
<th>pair</th>
<th>event</th>
</tr>
</thead>
<tbody>
<tr>
<td>synthetic</td>
<td>64</td>
<td>64</td>
<td>2048</td>
<td>2048000</td>
</tr>
<tr>
<td>last.fm (small)</td>
<td>297</td>
<td>423</td>
<td>492</td>
<td>31353</td>
</tr>
<tr>
<td>last.fm (medium)</td>
<td>568</td>
<td>1162</td>
<td>1822</td>
<td>127724</td>
</tr>
<tr>
<td>last.fm (large)</td>
<td>727</td>
<td>2247</td>
<td>6737</td>
<td>454375</td>
</tr>
</tbody>
</table>
Application II: Temporal Recommendation System

(a) synthetic, $\lambda = 0.1$
(b) synthetic, $\lambda = 1$
(c) synthetic, $\lambda = 10$

(d) last.fm (small)
(e) last.fm (medium)
(f) last.fm (large)
Problem with Large Sample

One potential drawback for extremely large-sample datasets
- Size of dual variables grows with the number of data points
- Require additional memory and computation cost

The Remedy: randomized block updating rules

<table>
<thead>
<tr>
<th>algorithm</th>
<th>type</th>
<th>guarantee</th>
<th>avg. iteration cost</th>
<th>convergence</th>
<th>constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMP</td>
<td>batch</td>
<td>primal and dual</td>
<td>$O(m)$</td>
<td>$O(\mathcal{L}/t)$</td>
<td>$\mathcal{L}$ bounded</td>
</tr>
<tr>
<td>RB-CMP</td>
<td>stoch</td>
<td>sad. point gap</td>
<td>$O(\ell \cdot m/b)$</td>
<td>$O(\mathcal{L}_\ell/t)$</td>
<td>$\mathcal{L}_\ell$ bounded</td>
</tr>
</tbody>
</table>

Table: Iteration complexity and iteration cost
Block-Decomposition and Randomization

Block-Coordinate Optimization

- High-dimensional minimization/maximization problems, e.g. RBCD [Nesterov, 2012; Richtárik& Takáč, 2014], SDCA [Shwartz and Zhang, 2013], etc.
- Saddle point problems, mostly based on primal-dual framework, e.g. SPDC [Zhang and Xiao, 2015], RPD [Dang and Lan, 2014].
- Various sampling schemes: uniform/non-uniform sampling, arbitrary sampling [Qu & Richtárik, 2016], adaptive sampling [Csiba et al., 2015].

Highlight

- We propose the first randomized block Mirror Prox algorithm that extends previous work in several sense:
  - solves a general class of variational inequalities;
  - uses a general distributed sampling scheme;
  - encompasses many variations with unified analysis.
Randomized Block Mirror Prox

The Situation

Find $u^* \in U : \langle F(u), u - u^* \rangle \geq 0, \forall u \in U$

where $u = [u_1; u_2; \ldots ; u_b]$ and $U = U_1 \times U_2 \times \cdots \times U_b$.

- Let $\{I_1, I_2, \ldots, I_\ell\}$ be a partition of the index set $I = \{1, 2, \ldots, b\}$, each of size $b_k$ such that $b_1 + \cdots + b_\ell = b$.
- We sample multiple blocks ($1 \leq \ell \leq b$), with each block uniformly sampled at random from each partition set.
- We assume that for any subset $K$, there exists $\mathcal{L}_\ell > 0$

$$\|F_K(u) - F_K(u')\|_{K,*} \leq \mathcal{L}_\ell \|u - u'\|_K, \forall u, u' \in U, u_k = u'_k, \forall k \in K$$

- $\ell = 1$, equivalent to fully randomized case
- $\ell = b$, equivalent to fully batch case
Randomized Block Mirror Prox (RB-MP)

for \( t = 1, 2, \ldots, T \) do

Pick a random subset of blocks such that \( K^j_t \in l_j, j = 1, \ldots, \ell \)

\[
\hat{u}^t := \begin{cases} 
\arg\min_{u_k} \{ V_k(u, u^t) + \langle \gamma_t F_k(u^t), x_k \rangle \}, & k \in K_t \\
\hat{u}^t, & k \notin K_t 
\end{cases}
\]

\[
u^{t+1} := \begin{cases} 
\arg\min_{u_k} \{ V_k(u, u^t) + \langle \gamma_t F_k(\hat{u}^t), u_k \rangle \}, & k \in K_t \\
u^t, & k \notin K_t 
\end{cases}
\]

end for

Proposition [H.-Harchaoui-Wang-Song, 2016]

Let the stepsizes \( \gamma_t \) satisfy \( 0 < \gamma_t \leq (\sqrt{2L})^{-1} \). We have

\[
\forall u \in U : \mathbb{E}[\langle F(u), u_T - u \rangle] \leq \bar{b} \cdot \frac{L \Theta[U]}{T}
\]

where \( \bar{b} = \max\{b_1, \ldots, b_\ell\} \).

Note the results can be extended to the composite setting.
Back to Poisson Likelihood Models with Large Sample

Saddle Point Reformulation

\[
\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m_+} \phi(x, y) := s^T x - y^T Ax + \sum_{i=1}^m c_i \log(y_i) + h(x)
\]

- The RB-CMP algorithm enjoys much cheaper iteration cost, while preserves the same convergence rate as CMP algorithm.
- The algorithm encompasses a variety of sampling strategies.
  - \( u = [x_1; \ldots; x_n; y_1; y_2; \ldots; y_m] \)
  - \( u = [[x_1; \ldots; x_n]; [y_1, y_2; \ldots; y_m]] \)
  - \( u = [[x]; [y_1; y_2; \ldots; y_m]] \)
  - \( u = [x; y] \)
- The algorithm shares some similarity with SPDC, RPD (in some case), but are algorithmically different.
Application III: Network Estimation

\[ \min_{x \in \mathbb{R}_+^U, X \in \mathbb{R}_+^{U \times U}} L(x, X) + \lambda \|X\|_1 \]

\[ L(x, X) := \sum_{u=1}^U [Tx_u + \sum_{j=1}^m X_{uu_j} G(T - t_j)] - \sum_{j=1}^m \log(x_{u_j} + \sum_{k: t_k < t_j} X_{u_j u_k} g(t_j - t_k)) \]

- \(m\) is number of events
- \(U\) is number of users
- \(\{(u_i, t_i)\}_{i=1}^m\) are the observations
- \(g(t) = ce^{-ct}\) is the triggering exponential kernel
Application III: Network Estimation

(a) synthetic, $\lambda = 0.01$
(b) synthetic, $\lambda = 1$
(c) synthetic, $\lambda = 100$

(d) Twitter, $\lambda = 0.01$
(e) Twitter, $\lambda = 1$
(f) Twitter, $\lambda = 100$

synthetic (50 users, 50000 events), Twitter (100 users, 98927 events)