Fast and Simple Optimization for Non-Lipschitz Poisson Likelihood Models

Niao He

Industrial and Enterprise Systems Engineering

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

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

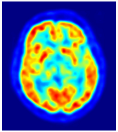
$$\min_x \frac{1}{m} \sum_{i=1}^{m} [a_i^T x - b_i \log(a_i^T x)] + h(x)$$

- Fundamental difficulty: loss function is not even Lipschitz continuous!
- Fewer work and softwares are available.

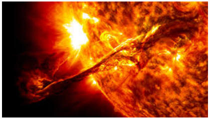
Poisson Likelihood Models

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



PET



detection

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

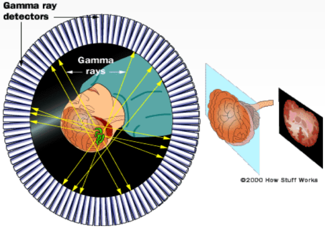
social networks

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

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

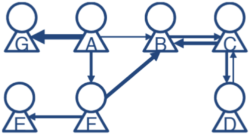
References: [Ben-Tal et al., 2001; Sra et al., 2009; Harmany et al., 2012]

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

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

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



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

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

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

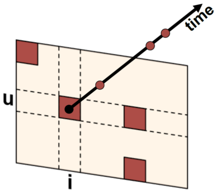
References: [Mohler et al., 2012; Zhou et al., 2013; Tomoharu et al., 2013]

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

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

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



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

$$\min_{X_1 \geq 0, X_2 \geq 0} \; L(\lambda(X_1, X_2)) + \lambda_1 \|X_1\|_{\text{nuc}} + \lambda_2 \|X_2\|_{\text{nuc}}$$

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

The Situation

- Previous objectives can be written in the compact form:

$$\min_{x\in\mathbf{R}^n_+} L(x)+h(x),\ \text{with } L(x)=s^Tx-\sum\nolimits_{i=1}^{m} c_i\log(a_i^Tx)$$

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

$$\mathrm{Prox}^h_{x_0}(\xi) := \mathrm{argmin}_{x\in\mathbf{R}^n_+}\{V_\omega(x,x_0)+\langle\xi,x\rangle+h(x)\}$$

  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:

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

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

$$\mathrm{Prox}^h_{x_0}(\xi) := \mathrm{argmin}_{x \in \mathbf{R}^n_+} \{V_\omega(x, x_0) + \langle \xi, x \rangle + h(x)\}$$

  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

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

Table: Convergence rates of different algorithms for penalized Poisson regression

The Crux: Saddle Point Reformulation

Problem of Interest

$$\min_{x\in\mathbf{R}^n_+} L(x)+h(x), \text{ with } L(x)=s^Tx-\sum_{i=1}^m c_i\log(a_i^Tx)$$

Key Observations

- Saddle point representation[1]
$$\min_{x\in\mathbf{R}^n_+}\max_{y\in\mathbf{R}^m_{++}} \phi(x,y):=s^Tx-y^TAx+\sum_{i=1}^m c_i\log(y_i)+h(x)$$
- Proximal-Friendliness of $\sum_{i=1}^m c_i\log(y_i)$

$$y^+=\operatorname*{argmin}_{y\in\mathbf{R}^m_{++}}\left\{\frac{1}{2}\|y\|_2^2+\langle\eta,y\rangle-\beta\sum_{i=1}^m c_i\log(y_i)\right\}=\left[(-\eta_i+\sqrt{\eta_i^2+4\beta c_i})/2\right]_{i=1,\dots,m}$$

[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

- Primal-dual methods: Arrow-Hurwicz method and its acceleration [Chambolle & Pock, 2011, Lan et al., 2013], Primal-dual prox [Yang et al., 2015], Douglas-Rachford splitting methods [Raguet et al, 2013, etc.]
- 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**

$$\widehat{u}_i^t = \min_{u_i \in U_i} \left\{ \alpha_i V_i(u_i, u_i^t) + \langle \gamma_t \nabla_i \phi(u^t), u_i \rangle + \gamma_t \Psi_i(u_i) \right\}, i = 1,2$$

$$u_i^{t+1} = \min_{u_i \in U_i} \left\{ \alpha_i V_i(u_i, u_i^t) + \langle \gamma_t \nabla_i \phi(\widehat{u}^t), u_i \rangle + \gamma_t \Psi_i(u_i) \right\}, i = 1,2$$

end for

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

Proposition [H.-Nemirovski-Juditsky, 2015]

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

$$\forall u = [u_1, u_2] \in U:\ \Phi(u_{1,T}, u_2) - \Phi(u_1, u_{2,T}) \le \frac{\mathcal{L} \cdot \Theta[U]}{T}$$

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

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

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

CMP for Penalized Poisson Models

Input: $x^1 \in \mathbf{R}^n_+, y^1 \in \mathbf{R}^m_{++}, \alpha, \gamma_t > 0$

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

$\widehat{x}^t = \mathrm{Prox}_{x^t}^{\gamma_t h/\alpha}(\gamma_t(s - A^T y^t)/\alpha)$

$y_i^t = Q^{\gamma_t}(\gamma_t(a_i^T x^t - y_i^t)), \forall i$

$x^{t+1} = \mathrm{Prox}_{x^t}^{\gamma_t h/\alpha}(\gamma_t(s - A^T \widehat{y}^t)/\alpha)$

$y_i^{t+1} = Q^{\gamma_t}(\gamma_t(a_i^T x^t - \widehat{y}_i^t)), \forall i$

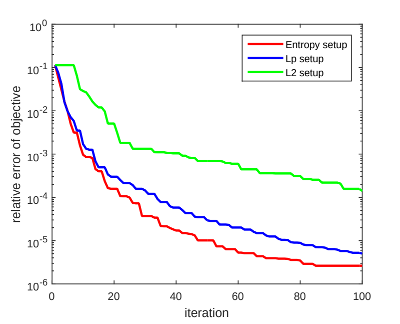
end for

- 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

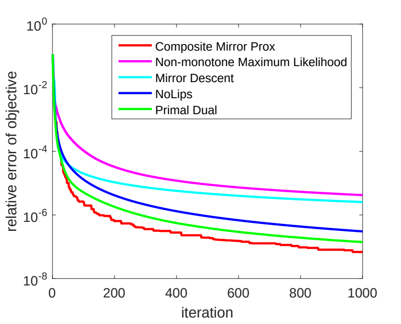
Application I: Positron Emission Tomography

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

Shepp-Logan image size 256×256, matrix A is of size 43530×65536

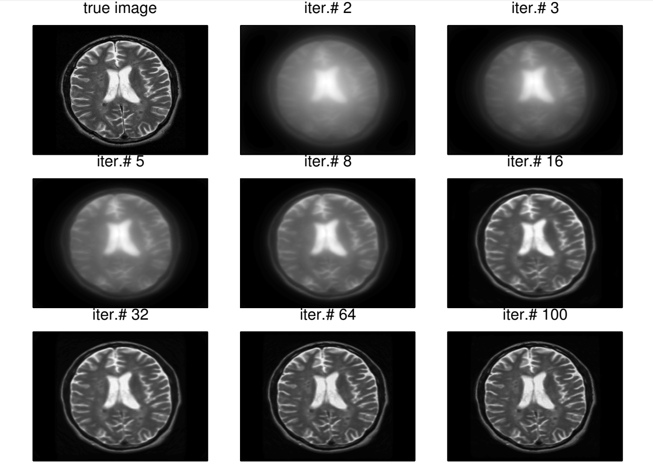


(a) CMP under proximal setups



(b) CMP vs. All

Application I: Positron Emission Tomography



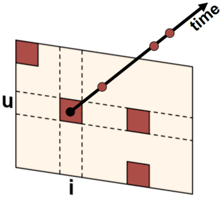
Reconstruction for MRI brain image

Application II: Temporal Recommendation System

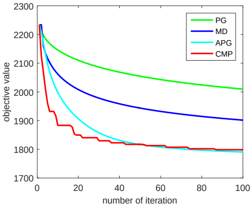
$$\min_{X_1 \geq 0, X_2 \geq 0} L(X_1, X_2) + \lambda_1 \|X_1\|_{\text{nuc}} + \lambda_2 \|X_2\|_{\text{nuc}}$$

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

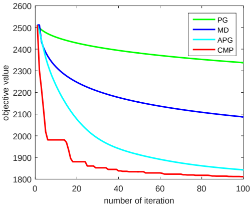
dataset	user	item	pair	event
synthetic	64	64	2048	2048000
last.fm (small)	297	423	492	31353
last.fm (medium)	568	1162	1822	127724
last.fm (large)	727	2247	6737	454375



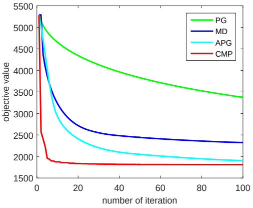
Application II: Temporal Recommendation System



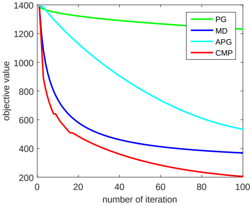
(a) synthetic, $\lambda = 0.1$



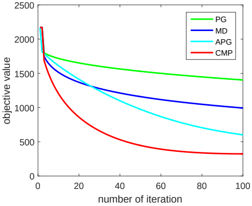
(b) synthetic, $\lambda = 1$



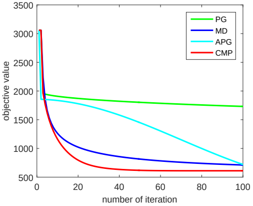
(c) synthetic, $\lambda = 10$



(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

algorithm	type	guarantee	avg. iteration cost	convergence	constant
CMP	batch	primal and dual	$O(m)$	$O(\mathcal{L}/t)$	$\mathcal{L}$ bounded
RB-CMP	stoch	sad. point gap	$O(\ell \cdot m/b)$ $(1 \leq \ell \leq b)$	$O(\mathcal{L}_\ell/t)$	$\mathcal{L}_\ell$ bounded

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

$$\text{Find } u_* \in U : \langle F(u), u - u_* \rangle \geq 0, \forall\, u \in U$$

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

- Let $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell\}$ be a partition of the index set $\mathcal{I} = \{1, 2, \ldots, b\}$, 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

Randomized Block Mirror Prox (RB-MP)

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

Pick a random subset of blocks such that $K_t^j \in I_j, j = 1, \ldots, \ell$

$$\widehat{u}^t := \begin{cases} \mathrm{argmin}_{u_k}\{V_k(u, u^t) + \langle \gamma_t F_k(u^t), x_k\rangle\}, & k \in K_t \\ u_k^t, & k \notin K_t \end{cases}$$

$$u^{t+1} := \begin{cases} \mathrm{argmin}_{u_k}\{V_k(u, u^t) + \langle \gamma_t F_k(\widehat{u}^t), u_k\rangle\}, & k \in K_t \\ u_k^t, & k \notin K_t \end{cases}$$

end for

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

Let the stepsizes γ_t *satisfy* $0 < \gamma_t \leq (\sqrt{2}\mathcal{L}_\ell)^{-1}$. *We have*

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

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

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

- 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

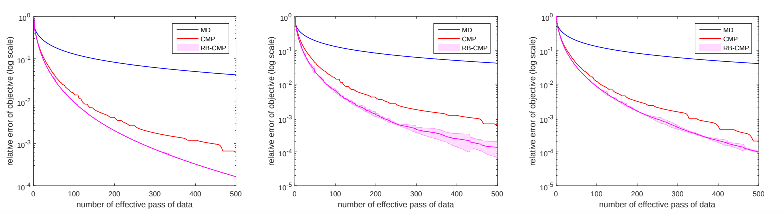
$$\min_{x \in \mathbf{R}_+^U, X \in \mathbf{R}_+^{U \times U}} L(x, X) + \lambda \|X\|_1$$

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

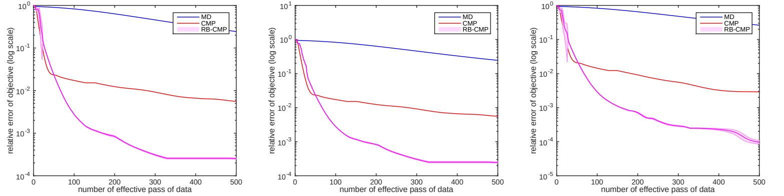
- m is number of events
- U is number of users
- $\{(u_i, t_i)\}_{i=1}^m$ are the observations
- $g(t) = ce^{-ct}$ is the triggering exponential kernel

social networks

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)