Applied Mathematics & Optimization
https://doi.org/10.1007/s00245-019-09634-6

Point Process Estimation with Mirror Prox Algorithms

Niao He[1] · Zaid Harchaoui[2] · Yichen Wang[3] · Le Song[3]

Abstract
Point process models have been extensively used in many areas of science and engineering, from quantitative sociology to medical imaging. Computing the maximum likelihood estimator of a point process model often leads to a convex optimization problem displaying a challenging feature, namely the lack of Lipschitz-continuity of the objective function. This feature can be a barrier to the application of common first order convex optimization methods. We present an approach where the estimation of a point process model is framed as a saddle point problem instead. This formulation allows us to develop Mirror Prox algorithms to efficiently solve the saddle point problem. We introduce a general Mirror Prox algorithm, as well as a variant appropriate for large-scale problems, and establish worst-case complexity guarantees for both algorithms. We illustrate the performance of the proposed algorithms for point process estimation on real datasets from medical imaging, social networks, and recommender systems.

Keywords Mirror Prox · Proximal algorithm · Point process · Saddle point problem

✉ Niao He
niaohe@illinois.edu

Zaid Harchaoui
zaid@uw.edu

Yichen Wang
yichen.wang@gatech.edu

Le Song
lsong@cc.gatech.edu

1 Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

2 Department of Statistics, University of Washington, Seattle, WA 10003, USA

3 Department of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Published online: 26 November 2019

The last decade has witnessed a tremendous growth of interests and successes in point process modeling for event data analysis. Applications range from the classical imaging sciences such as nuclear medicine and microscopy using homogeneous Poisson processes (see, e.g., [1] for a comprehensive survey on this topic), stock and option pricing using marked point processes [2,3], all the way to the recent studies of information diffusion over social networks using Hawkes processes [4–8]. A recurring theme of these applications is the requirement of efficient statistical estimation from real-time and large-scale event data. Penalized maximum likelihood estimation has been a mainstream approach to learn estimators for point process models from data. However, existing optimization tools for computing these estimators are far from being optimal and remain a major obstacle to their applicability in practice.

In this work, we present a general form of penalized maximum likelihood estimation for point process models. The problem of interest writes as follows:

$$\min_{x \in \mathbb{R}^n_+} \; f(x) := L(x) + h(x), \;\; \text{where } L(x) := s^T x - \sum\nolimits_{i=1}^{m} c_i \log(a_i^T x). \qquad (1)$$

Here m is the number of observations, $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geq 0\}$ stands for the set of nonnegative vectors, the coefficients $\{c, a_i, i = 1, \ldots m\}$ are given nonnegative values. The component $L(x)$ stands for the negative log-likelihood, and the function $h(x)$ can be a (possibly nonsmooth) convex penalty term used to promote desired structural properties of the solution such as sparsity or low-rank structure. Note that the above problem is well defined and convex. We will refer to this as the *penalized point process likelihood model*. We show later that this general form covers a wide range of likelihood-based objectives for point processes, e.g., Poisson processes, self-exciting and mutual-exciting Hawkes processes.

A key challenge to computing maximum likelihood estimators of point process models lies in the fact that the likelihood function is neither globally Lipschitz continuous nor globally Lipschitz differentiable. We shall explain this in more detail in Sect. 2.1. This makes it fundamentally different from and computationally more difficult than those pertaining to Gaussian processes or generalized linear models. Despite a large body of work on efficient gradient-based (*a.k.a.* first-order) methods for likelihood-based estimation in the literature, ranging from proximal algorithms (see, e.g., [9,10]) to stochastic and incremental algorithms (see, e.g., [11–13]), the overwhelming majority of work assume the log-likelihood to be globally Lipschitz-continuous. Therefore, the application of these algorithms to compute to likelihood-based estimators goes beyond the range of their theoretical guarantees, which may lead to disappointing results in practice. This is indeed evidenced in the results we present in Sect. 4. Hence, there is a need for optimization algorithms, with theoretical guarantees, that could ideally handle both the non-Lipschitzness of the log-likelihood and the potential non-smoothness of the regularization penalty. Another obstacle to computing such point process estimators, especially in the large-scale regime, comes from the computational side, namely, the expensive overhead for computing the gradient based on the entire data.

Related Work Few works have attempted to address the non-Lipschitzness issue of likelihood objectives for point process estimation. [14] propose to add a tolerance ϵ to each logarithmic term, which results in a smooth objective yet with a large Lipschitz constant $L \sim O(1/\epsilon^2)$. [15] instead propose to impose strict positivity constraints $a_i^T x \geq \epsilon, \forall i = 1, \ldots, m$ to the problem, at the cost of computationally expensive projections. Another approach is explored in [16], where the authors exploit the self-concordance nature of the logarithmic term and propose a sophisticated proximal gradient method, yet only with locally linear convergence. In a different line of work, [17] treat this problem as a general non-smooth minimization using the Mirror Descent algorithm [18]. This approach avoids the requirement on Lipschitz continuity of the gradient, but in the sacrifice of having a worse rate of convergence, i.e., $\mathcal{O}(1/\sqrt{t})$. [19] tackle the problem with an instance of the alternating direction method of multipliers (ADMM), which is guaranteed to converge but requires expensive computation of matrix inversion at each iteration. Hence, none of these algorithms are efficient for the general purpose of solving the point process likelihood models, especially at large scale. Recent work [20,21] address a general class of relatively smooth convex problems which includes the problem of interest as a special case, but their approaches are fundamentally different from ours.

Main Contributions We present Mirror Prox algorithms to efficiently compute point process estimators from penalized maximum likelihood objectives. The algorithms hinge upon a saddle point reformulation, circumventing the need for the common Lipschitz-continuity assumptions in the design of first-order methods. The basic algorithm, called the Composite Mirror Prox algorithm, enjoys a $\mathcal{O}(1/t)$ convergence rate in theory, in contrast to the typical $\mathcal{O}(1/\sqrt{t})$ rate of the non-smooth minimization alternative [17]. To tackle large sample and high dimensional problems, we propose a fully randomized block-decomposition variant that enjoys a cheaper cost per iteration. Finally, we present experimental results obtained with the proposed algorithms and competing ones when applied to several applications of point process modeling to medical imaging, social network estimation, and recommendation systems. The results obtained demonstrate the consistent performance of the proposed algorithms when compared to existing methods.

1 Point Process Models

In this section, we consider several examples of point processes models. We then provide a saddle point formulation of penalized maximum likelihood estimation, central to the development of the algorithms in this paper.

1.1 Point Process, Maximum Likelihood Estimation, and Motivating Examples

Point Process This is a widely used probabilistic and statistical model of occurrences of events. The model builds off an intensity function $\lambda(t)$, such that

$$\text{Prob}\left[N(t+dt)-N(t)=1\middle|\mathcal{F}^t\right]=\lambda(t)dt+o(dt), \tag{2}$$
$$\text{Prob}\left[N(t+dt)-N(t)>1\middle|\mathcal{F}^t\right]=o(dt), \tag{3}$$

where $N(t)$ is the corresponding counting process adapted to a filtration $\mathcal{F}^t$. Some important examples of point processes include: (i) homogeneous Poisson process where the intensity function $\lambda(t)=\lambda$ is a constant; (ii) nonhomogeneous Poisson process where $\lambda(t)$ is a general function of t; (iii) self-exciting Hawkes process where $\lambda(t)=v(t)+\int_0^t\gamma(t-\tau)dN(\tau)$; here $v(t)$ is the base intensity of the process and $\gamma(t)$ expresses the positive influence of the past events on the current state; (iv) Cox process, also called doubly stochastic Poisson process with $\lambda(t)$ itself being a stochastic process. These point processes have found a myriad of applications in classical imaging science and modern machine learning areas such as social networks for their capability of capturing temporal dynamics of time series data.

Maximum Likelihood Estimation The associated estimation and prediction problems with these point processes often rely on maximum likelihood estimation. Given a sequence of events $\{t_i\}_{i=1}^N$ from some point process within time interval T with intensity $\lambda(t;\,\theta)$ and model parameter θ, the negative log-likelihood is

$$L(\theta)=\int_0^T\lambda(t;\,\theta)dt-\sum_{i=1}^N\log(\lambda(t_i;\,\theta)).$$

Note that as long as the intensity function $\lambda(t;\,\theta)$ is nonnegative and linear with respect to the parameter θ, the negative likelihood $L(\theta)$ is always well-defined and convex in θ. Maximum likelihood estimation leads to a convex optimization problem. Often times, additional penalty terms are imposed to exploit desired structural properties of the estimator such as sparsity and low rank in high dimensional problems.

Below we give a couple of interesting examples emerging in imaging science and machine learning; see; e.g., [6,19,22–24].

Example 1 *(Poisson Imaging)* Recovering Poissonian images is a classical problem occurring in many medical and astronomical applications. Typically, we observe the number of event counts from independent homogeneous Poisson processes, whose intensity vector is assumed to be a linear combination of an unknown image. Let $c_i, i=1,\ldots,m$ denote the observed counts and $\lambda_i(t)=a_i^Tx, i=1,\cdots,m$ denote the underlying intensity of each dimension, where $x\in\mathbb{R}_+^n$ stands for an unknown image to be learned, and a_i^T is the i-th row of a given observation operator $A\in\mathbb{R}^{m\times n}$. For example, for position emission tomography, each entry a_{ij} of this matrix A stands for the probability that the pair of gamma-quants originating from voxel j ($j=1,\ldots,n$) registered by the i-th ($i=1,\ldots,m$) pair of detectors. Computing the penalized maximum likelihood estimator boils down to solving:

$$\min_{x\in\mathbb{R}_+^n}\ \sum_{i=1}^m a_i^Tx-\sum_{i=1}^m c_i\log(a_i^Tx)+h(x), \tag{4}$$

where $h(x)$ is some penalty function that promotes smoothness of the image, e.g., the total variation regularization term. Obviously, this likelihood model falls into the general problem of our interest in (1).

Example 2 *(Social Network Estimation)* Discovering the latent influence and reciprocating relationships among social communities has been an active research topic in the last decade. A common practice to model the so-called mutual excitation effect among these communities is through multivariate Hawkes processes. Let $N_1(t), \ldots, N_p(t)$ be p-dimensional counting processes such that the intensity of one dimension is affected by the arrivals of other dimensions through:

$$\lambda_i(t;\ \mu, A) := \mu_i + \sum_{j=1}^{p} \int_0^t a_{ij} \cdot k(t-\tau) dN_j(\tau) = \mu_i + \sum_{j=1}^{p} \sum_{n=1}^{N_j(t)} a_{ij} \cdot k(t-\tau_{jn}), \quad (5)$$

for $i = 1, \ldots, p$. Here, the timestamp τ_{jn} is the n-th arrival time of the j-th dimension. The parameter $\mu_i \geq 0$ stands for the base intensity for each dimension. The parameter $a_{ij} \geq 0$ stands for the influence from the j-th dimension to i-th dimension; thus matrix $A = [a_{ij}]$ captures the hidden network structure of social influences. The function $k(\cdot) : \mathbb{R} \to \mathbb{R}$ represents the triggering kernel and captures the decaying effect of the influence; e.g., a common choice is to set $k(t) = \exp(-t/\sigma)$ with some $\sigma > 0$ when $t > 0$ and $k(t) = 0$ when $t \leq 0$. Now given a sequence of events $\{\tau_{in} : i = 1, \ldots, p,\ n = 1, \ldots N_p(T)\}$ observed within time interval $T > 0$, we would like to infer the actual base intensity vector μ and influence matrix A. For example, the sparse maximum likelihood estimation can be formulated as follows [6]:

$$\min_{\mu \geq 0, A \geq 0} \quad \sum_{i=1}^{p} \left(\int_0^T \lambda_i(t;\ \mu, A) dt - \sum_{n=1}^{N_i(T)} \log(\lambda_i(\tau_{in};\ \mu, A)) \right) + \gamma \|A\|_1, \quad (6)$$

where $\gamma > 0$ is some regularization coefficient and the penalty function based on ℓ_1-norm is used to promote sparsity of the influence matrix A. Here $\|A\|_1 := \sum_{i=1}^{p} \sum_{j=1}^{p} |a_{ij}|$. Note that the intensity functions $\lambda_i(t; \mu, A), i = 1, \ldots, p$ are linear in μ and A; hence, the above problem can be viewed as a special case of our general problem of interest in (1).

Example 3 *(Temporal Recommendation System)* Incorporating temporal behaviors of customers into recommendation systems has been recently studied to improve personalized suggestions. A natural way to model the recurrent activities for any user-item pair is using the self-exciting Hawkes process. Let $N_{ij}(t)$ be the counting process associated with each user $i \in I$ and item $j \in J$ with intensity given by

$$\lambda_{ij}(t;\ U, A) = u_{ij} + a_{ij} \sum_{n=1}^{N_{ij}(t)} k(t - \tau_{ijn}). \quad (7)$$

Here $u_{ij} \geq 0$ and $a_{ij} \geq 0$ are the base intensity and self-exciting coefficient for each user-item pair (i, j), which form into the matrices $U = [u_{ij}]$ and $A = [a_{ij}]$. And

τ_{ijn} stands for the n-th arrival time of the events in the counting process. Now given a sequence of events $\{\tau_{ijn} : i \in I,\ j \in J,\ n = 1, \ldots, N_{ij}(T)\}$ within time interval $T > 0$, we would like to estimate the intensity and self-exciting matrices U and A in order to make time-sensitive recommendations. Incorporating low-rank constraints, a penalized maximum likelihood estimation problem can be formulated as follows [24]:

$$\min_{U\geq 0, A\geq 0} \sum_{i\in I}\sum_{j\in J}\left(\int_0^T \lambda_{ij}(t;\ U, A)dt - \sum_{n=1}^{N_{ij}(T)} \log(\lambda_{ij}(\tau_{ijn};\ U, A))\right) + \gamma_1\|U\|_{\text{nuc}} + \gamma_2\|A\|_{\text{nuc}}, \tag{8}$$

where $\gamma_1, \gamma_2 > 0$ are some regularization coefficients and the penalty functions based on the nuclear norm $\|\cdot\|_{\text{nuc}}$ are used to promote low rank of the base intensity matrix U and the self-exciting coefficient matrix A. Here $\|A\|_{\text{nuc}} := \sum_{i=1}^{\min\{|I|,|J|\}} \sigma_i(A)$, where $\sigma_i(A), i = 1, \ldots, \min\{|I|, |J|\}$ are the singular values of matrix A. Again, in this example, the intensity functions $\lambda_{ij}(t;\ U, A)$ are linear in terms of the model parameters U and A. Hence, the above problem can be viewed as as another special case of our general problem of interest in (1).

1.2 Problem Statement

Recall that the goal of this paper is to address problems related to the computation of penalized maximum likelihood estimators of point process models of the form: (1):

$$\min_{x\in\mathbb{R}^n_+} f(x) := L(x) + h(x), \text{ where } L(x) := s^T x - \sum\nolimits_{i=1}^m c_i \log(a_i^T x).$$

As discussed above, the previous examples for Poisson imaging, social network estimation, as well as temporal recommendation systems, can all be characterized as special cases of (1).

Before proceeding, we first define the notion of *Bregman proximal operator* of a convex function. Given input x_0 and ξ, the Bregman proximal operator (we will simply refer to proximal operator below, see [25–27] and references therein) of a convex function $h(x)$ is defined as

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

where $V_\omega(x, x_0) := \omega(x) - \omega(x_0) - \nabla\omega(x_0)^T(x - x_0)$ is the *Bregman divergence* defined by some distance generating function $\omega(x)$. We assume that the function $\omega(x) : \mathbb{R}^n_+ \to \mathbb{R}$ is *compatible*, i.e., continuously differentiable and 1-strongly convex on $\mathbb{R}^n_+$ with respect to a given norm defined on $\mathbb{R}^n$. For instance, the function $\omega(x) = \frac{1}{2}\|x\|_2^2$: $\mathbb{R}^n \to \mathbb{R}$ is compatible with respect to the Euclidean norm $\|\cdot\|_2$ and gives rise to the Euclidean distance. The entropy function $\omega(x) = \sum_{i=1}^n x_i \log(x_i)$: $\mathbb{R}^n_+ \to \mathbb{R}$ is compatible with respect to the ℓ_1-norm $\|\cdot\|_1$ and gives rise to the Kullback-Leibler divergence. Throughout the paper, we make the following assumption below.

Assumption 1 *Function $h(x)$ is convex and is proximal-friendly, i.e., the proximal operator can be exactly computed for some distance generating function $\omega(x)$ associated with norm $\|\cdot\|_x$ on $\mathbb{R}^n_+$.*

Note that the above assumptions hold true for a wide range of sparsity-promoting penalty functions, including those used in the motivating examples. See, e.g., [28,29] for a survey of proximal operators in machine learning and signal processing.

Remark 1 The point process likelihood model considered here is similar but more general than the classical *Poisson linear model* (sometimes coined as Poisson compressed sensing, or Poisson intensity reconstruction) that assumes Poisson noise of linear measurements, i.e., $c_i \sim \text{Poisson}(a_i^T x)$, $i = 1, \ldots, m$. The latter leads to a negative log-likelihood of $L(x) = \sum_{i=1}^m a_i^T x - \sum_{i=1}^m c_i \log(a_i^T x)$, which can be reviewed as a special case of (1), as discussed in Example 1. However, one should note that (1) also covers many other likelihood estimation problems based on general point processes, e.g., those described in Examples 2 and 3 below. These problems cannot necessarily be viewed as a pure Poisson linear model.

Remark 2 The above problem is different from the *Poisson log-linear model* or Poisson regression (a special case of generalized linear model), which assumes Poisson observations $c_i \sim \text{Poisson}(e^{a_i^T x})$, $i = 1, \ldots, m$. This leads to the negative log-likelihood: $\tilde{L}(x) = \sum_{i=1}^m e^{a_i^T x} - c_i(a_i^T x)$, which differs from that of the point process likelihood model considered in this paper. Without additional regularization constraint, the maximum likelihood estimation problem for the Poisson log-linear model is unconstrained, while the one we consider here is constrained with positivity constraints.

Remark 3 A major difference between (1) and the objective for maximum likelihood estimation of Poisson log-linear model lies in the Lipschitz condition of the (negative) likelihood. On any compact set, the function $\tilde{L}(x)$ is globally Lipschitz differentiable, while this is not necessarily true for $L(x)$. In fact, for any compact set $X := \{x \in \mathbb{R}^n_+ : \|x\|_x \leq R\}$ with $R > 0$, there does not exist a finite Lipschitz constant $M > 0$ such that

$$|L(x) - L(x')| \leq M \cdot \|x - x'\|_x, \forall x, x' \in X.$$

This can be easily shown as follows: let $x \in X$, and $x' = \epsilon x \in X$ with $0 < \epsilon < 1$, we see that $|L(x) - L(x')| = |(1-\epsilon)s^T x + \sum_{i=1}^m c_i \log(\epsilon)|$ becomes unbounded when ϵ goes to zero. Hence, although the function L is convex and differentiable, it is i) not globally Lipschitz continuous; ii) not Lipschitz differentiable even on a compact set. As a result, the problem (1) we consider here essentially falls beyond the theoretical grasp of the common gradient-based optimization algorithms, such as (accelerated) proximal gradient method [9,30], (composite) mirror descent [17,31,32], as well as a wide family of coordinate descent algorithms [33–35].

1.3 Saddle Point Formulation

We present our approach to solve this family of non-Lipschitz optimization problems. The approach leads to algorithms enjoying theoretical guarantees with competitive

practical performance. Central to the approach is an equivalent saddle point formulation allowing us to circumvent difficulties arising from the non-Lipschitz of the original objective (1).

Specifically, by invoking the Fenchel representation of the log function $\log(u) = \min_{v\geq 0}\{uv - \log(v) - 1\}$, we can rewrite the problem of interest as

$$\min_{x\in\mathbb{R}^n_+} \max_{v\in\mathbb{R}^m_+} s^T x + \sum\nolimits_{i=1}^m c_i[\log(v_i) - v_i a_i^T x + 1] + h(x).$$

Now replacing $y_i = c_i v_i$, the above problem can be further simplified to the convex-concave saddle point problem:

$$\min_{x\in\mathbb{R}^n_+} \max_{y\in\mathbb{R}^m_+} \psi(x, y) := s^T x - y^T Ax + \sum\nolimits_{i=1}^m c_i \log(y_i) + h(x) + c_0, \tag{9}$$

where the matrix $A = [a_1^T; a_2^T; \ldots; a_m^T]$ and $c_0 = \sum_{i=1}^m c_i(1 - \log(c_i))$ is a constant. The cost function $\psi(x, y)$ is convex in x for any given $y \in \mathbb{R}^m_+$ and is concave in y for any given $x \in \mathbb{R}^n_+$, and moreover, $f(x) = \max_{y\in\mathbb{R}^m_+} \psi(x, y)$. Notice that the cost function is still non-Lipschitz due to the logarithmic term $\sum_{i=1}^m c_i \log(y_i)$. However, a key difference from the negative log-likelihood function is that this non-Lipschitz term, is separable and admits a closed-form solution when computing its proximal operator.

In particular, we can show the following property: let $y^+ = \operatorname{argmin}_{y\in\mathbb{R}^m_+}\left\{\frac{1}{2}\|y\|_2^2 + \langle\eta, y\rangle - \beta\sum_{i=1}^m c_i \log(y_i)\right\}$ given $\eta \in \mathbb{R}^m$ and $\beta > 0$, then y^+ can be computed in closed-form as

$$y_i^+ = Q^\beta(\eta_i) := \big(-\eta_i + \sqrt{\eta_i^2 + 4\beta c_i}\big)/2, \forall i = 1, \ldots, m \tag{10}$$

In other words, the proximal operator of $p(y) := -\sum_{i=1}^m c_i \log(y_i)$ with respect to the usual Euclidean distance can be computed efficiently and naturally lends itself to parallelization. To our best knowledge, this simple yet powerful observation has not been exploited to design algorithms for point process model estimation. An exception is the independent work [36] where related non-negative matrix factorization problems were considered with a similar approach to ours.

Our second key observation is that the above saddle point formulation can be regarded as a *well-structured composite saddle point problem*. This consists of a smooth convex-concave function and two separable penalty functions – a convex penalty, $h(x)$, for variable x and a concave penalty, $-p(y)$, for variable y. The saddle point perspective we present here paves the way to the development of principled and efficient algorithms for point process model estimation using penalized maximum likelihood. In the next section, we first introduce the Composite Mirror Prox algorithm to solve such composite saddle point problems, and then introduce a fully randomized block-decomposition variant for problems with large sample and high dimensions.

Algorithm 1 Composite Mirror Prox (CMP) for Saddle Point Problems

Input: $(x^1, y^1) \in X \times Y$, $\alpha_1 > 0$ and $\alpha_2 > 0$, stepsize $\{\gamma_t\}$

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

2: $\quad \hat{x}^t = \text{argmin}_{x \in X} \{\alpha_1 V_1(x, x^t) + \gamma_t \langle \nabla_x \phi(x^t, y^t), x \rangle + \gamma_t h(x)\}$

3: $\quad \hat{y}^t = \text{argmin}_{y \in Y} \{\alpha_2 V_2(y, y^t) - \gamma_t \langle \nabla_y \phi(x^t, y^t), y \rangle + \gamma_t p(y)\}$

4: $\quad x^{t+1} = \text{argmin}_{x \in X} \{\alpha_1 V_1(x, x^t) + \gamma_t \langle \nabla_x \phi(\hat{x}^t, \hat{y}^t), x \rangle + \gamma_t h(x)\}$

5: $\quad y^{t+1} = \text{argmin}_{y \in Y} \{\alpha_2 V_2(y, y^t) - \gamma_t \langle \nabla_y \phi(\hat{x}^t, \hat{y}^t), y \rangle + \gamma_t p(y)\}$

6: **end for**

Output: $x_T = \sum_{t=1}^T \gamma_t \hat{x}^t / \sum_{t=1}^T \gamma_t$ and $y_T = \sum_{t=1}^T \gamma_t \hat{y}^t / \sum_{t=1}^T \gamma_t$

2 Composite Saddle Point Problem

Consider the following convex–concave composite saddle point problem

$$\min_{x \in X} \max_{y \in Y} \ \psi(x, y) := \phi(x, y) + h(x) - p(y) \tag{11}$$

under the situation

- $X \subseteq E_x$ and $Y \subseteq E_y$ are nonempty closed convex sets in Euclidean spaces E_x, E_y;
- $\phi(x, y)$ is a convex-concave function on $X \times Y$ with Lipschitz continuous gradient;
- $h : X \to \mathbb{R}$ and $p : Y \to \mathbb{R}$ are convex functions, perhaps non-Lipschitz, but are proximal-friendly in the following sense: there exist some distance generating functions $\omega_1(\cdot)$ and $\omega_2(\cdot)$ that are compatible with respect to $(E_x, \|\cdot\|_x)$ and $(E_y, \|\cdot\|_y)$ and the subproblems $\min_{x \in X} \{\alpha\omega_1(x) + \langle \xi, x \rangle + \beta h(x)\}$ and $\min_{y \in Y} \{\alpha\omega_2(y) + \langle \eta, y \rangle + \beta p(y)\}$ are easy to solve for any $\alpha > 0, \beta > 0$ and input $\xi \in E_x, \eta \in E_y$.

In addition, we denote $\overline{\psi}(x) := \sup_{y \in Y} \psi(x, y)$ and $\underline{\psi}(y) := \inf_{x \in X} \psi(x, y)$. We assume saddle point exists and denote as (x^*, y^*). The distance generating functions define the Bregman distances $V_1(x, x') = \omega_1(x) - \omega_1(x') - \nabla\omega_1(x')^T(x - x')$ and $V_2(y, y') = \omega_2(y) - \omega_2(y') - \nabla\omega_2(y')^T(y - y')$ such that $V_1(x, x') \geq \frac{1}{2}\|x - x'\|_x^2$ and $V_2(y, y') \geq \frac{1}{2}\|y - y'\|_y^2$. Given two scalars $\alpha_1 > 0, \alpha_2 > 0$, we can build an aggregated distance generating function on $U = X \times Y$ with $\bar{\omega}(u = [x; y]) = \alpha_1\omega_1(x) + \alpha_2\omega_2(y),$, which is compatible to the induced norm $\|u = [x; y]\| = \sqrt{\alpha_1\|x\|_x^2 + \alpha_2\|y\|_y^2}$. The dual norm of $\|u\|$ is $\|v = [\xi; \eta]\|_* = \sqrt{\alpha_1^{-1}\|\xi\|_{x,*}^2 + \alpha_2^{-1}\|\eta\|_{y,*}^2}$ where $\|\cdot\|_{x,*}$ and $\|\cdot\|_{y,*}$ are the dual norms to $\|\cdot\|_x$ and $\|\cdot\|_y$, respectively.

2.1 Composite Mirror Prox Algorithm

The Composite Mirror Prox (CMP) algorithm was originally introduced in [37] for solving a general class of variational inequalities. We specifically tailor it for solving the composite saddle point problem (11) and present the algorithm in Algorithm 1. The algorithm generalizes the proximal gradient method with Bregman distances from

the usual composite minimization to composite saddle point problems and works "as if" there were no non-smooth terms $h(x)$ and $p(y)$.

Assumption 2 *We assume that $\phi(x, y)$ has $\mathcal{L}$-Lipchitz continuous gradient, namely: $\|\nabla\phi(u) - \nabla\phi(u')\|_* \le \mathcal{L}\|u - u'\|$, where $\nabla\phi(u) = [\nabla_x\phi(x, y); -\nabla_y\phi(x, y)]$.*

For any set $U' \subset U := X \times Y$, let us define the diameter of U' as $\Theta[U'] = \max_{u=[x;y]\in U'}\{\alpha_1 V_1(x, x^1) + \alpha_2 V_2(y, y^1)\}$, where $V_1(x, x^1)$ and $V_2(y, y^1)$ are the Bregman distances associated with $\omega_1(x)$ and $\omega_2(y)$. We have the following results:

Lemma 1 ([37]) *Under Assumption 2 and setting stepsize $0 < \gamma_t \le \mathcal{L}^{-1}$, the candidate solution (x_T, y_T) generated by Composite Mirror Prox leads to the efficiency estimate:*

$$\psi(x_T, y) - \psi(x, y_T) \le \frac{\Theta[X \times Y]}{\sum_{t=1}^T \gamma_t}, \forall x \in X, y \in Y. \tag{12}$$

Moreover, if we set $\gamma_t = \mathcal{L}^{-1}$, then we further have

$$\overline{\psi}(x_T) - \overline{\psi}(x^*) \le \frac{\Theta[\{x^*\} \times Y]\mathcal{L}}{T}. \tag{13}$$

In the situation discussed above, CMP achieves the $\mathcal{O}(1/T)$ convergence rate for solving composite saddle point problems. The rate is known to be unimprovable already in the simple case of bilinear saddle point problems; see, e.g., [18]. We emphasize that this is not the only algorithm available for solving composite saddle point problems; alternative options include primal–dual algorithms [38–41], hybrid proximal extragradient type algorithms [42,43], smoothing proximal gradient [44,45], just to list a few.

CMP shares some similarity with these algorithms, but possesses distinct features in several aspects: (i) unlike primal–dual algorithms, the primal and dual variables are updated simultaneously, thus can easily accommodate parallelism; (ii) the algorithm benefits from the use of non-Euclidean Bregman distances for proximal operators, which could potentially improve the constant factor in the convergence rate in terms of dimension dependence[1]; we point out that in a recent work by [40], the authors have extended the primal–dual method to embrace non-Euclidean Bregman distances as well; (iii) the stepsize can be self-tuned using line-search without requiring a priori knowledge of Lipschitz constant; see details in [37]; and lastly, (iv) unlike the proximal gradient methods based on Nesterov's smoothing, the algorithm does not need to tune any extra smoothness hyperparameter, which is often critical for achieving satisfactory empirical performance. Due to these consideration, we particularly adopt CMP as our working horse to solve the composite saddle point problems. We show later in the numerical experiments that for several specific applications, CMP can slightly outperform the primal–dual algorithm in practice.

[1] Particularly for ℓ_1 minimization, it has been well studied that non-Euclidean gradient methods achieve a nearly optimal dependence on dimension [17,31].

2.2 Fully Randomized Composite Mirror Prox Algorithm

In this section, we introduce a fully randomized variant of the Composite Mirror Prox algorithm, appropriate for solving problems with large samples and high dimensions. Block-decomposition and randomization techniques have been successful in solving high-dimensional convex minimization problems; see, e.g., [33–35,46,47] and reference therein. When considering saddle point problems, a few block-decomposition variants were developed based on primal–dual algorithms, e.g., stochastic primal–dual coordinate descent algorithm [48] and randomized primal dual algorithm [49,50]. However, most of these algorithms make a randomized update for the dual variable while making a full gradient update for the primal variable. Here we present what we call a *fully randomized CMP*. This natural extension of CMP performs randomized block updates for both primal and dual variables.

Problem Setting Consider the following situation in addition to the composite saddle point problem described in (11):

- $x = [x_1; x_2; \ldots; x_{b_1}]$ and $X = X_1 \times X_2 \times \cdots \times X_{b_1}$, where $X_k, k = 1, 2, \ldots, b_1$ are closed convex sets;
- $y = [y_1; y_2; \ldots; y_{b_2}]$ and $Y = Y_1 \times Y_2 \times \cdots \times Y_{b_2}$, where $Y_l, l = 1, 2, \ldots, b_2$ are closed convex sets;
- $h(x) = \sum_{k=1}^{b_1} h_k(x_k)$ is separable and each is proximal-friendly under some distance generating function $\omega_{1,k}(x_k) : X_k \to \mathbb{R}$ that is compatible w.r.t. the norm $\|\cdot\|_{x,k}$ (with dual norm $\|\cdot\|_{x,k,*}$) and induces the Bregman divergence $V_{1,k}(x_k, x_k')$;
- $p(x) = \sum_{l=1}^{b_2} p_l(y_l)$ is separable and each is proximal-friendly under some distance generating function $\omega_{2,l}(y_l) : Y_l \to \mathbb{R}$ that is compatible w.r.t. the norm $\|\cdot\|_{y,l}$ (with dual norm $\|\cdot\|_{y,l,*}$) and induces Bregman divergence $V_{2,l}(y_l, y_l')$.

The fully randomized Composite Mirror Prox algorithm works as follows: at each iteration, a primal block from $\{x_1, x_2, \ldots, x_{b_1}\}$ and a dual block from $\{y_1, y_2, \ldots, y_{b_2}\}$ are randomly selected and performed with block proximal coordinate descent updates. The explicit algorithm is provided below in Algorithm 2. Before deriving the convergence, we first make the following assumption on the function $\phi(x, y)$:

Assumption 3 *For any $k = 1, 2, \ldots, b_1$ and $l = 1, 2, \ldots, b_2$, let $x \in X$, $x' \in X$ be such that $x_{k'} = x_{k'}', k' \neq k$, and let $y \in Y$, $y' \in Y$ be such that $y_{l'} = y_{l'}', l' \neq l$. Namely, only the k-th block between x and x' and l-th block between y and y' differ. We assume that there exist some constants $L_k^{xx} > 0$, $L_l^{xy} > 0$, $L_l^{yy} > 0$, and $L_k^{yx} > 0$ such that the following conditions hold true:*

$$
\begin{aligned}
\|\nabla_{x_k}\phi(x, y) - \nabla_{x_k}\phi(x', y)\|_{x,k,*} &\le L_k^{xx}\|x_k - x_k'\|_{x,k}, \ \forall x, x' \in X, s.t.\ x_{k'} = x_{k'}', k' \neq k,\\
\|\nabla_{x_k}\phi(x, y) - \nabla_{x_k}\phi(x, y')\|_{x,k,*} &\le L_l^{xy}\|y_l - y_l'\|_{y,l}, \ \forall y, y' \in Y, s.t.\ y_{l'} = y_{l'}', l' \neq l,\\
\|\nabla_{y_l}\phi(x, y) - \nabla_{y_l}\phi(x', y)\|_{l,y,*} &\le L_k^{yx}\|x_k - x_k'\|_{x,k}, \ \forall x, x' \in X, s.t.\ x_{k'} = x_{k'}', k' \neq k,\\
\|\nabla_{y_l}\phi(x, y) - \nabla_{y_l}\phi(x, y')\|_{l,y,*} &\le L_l^{yy}\|y_l - y_l'\|_{y,l}, \ \forall y, y' \in Y, s.t.\ y_{l'} = y_{l'}', l' \neq l.
\end{aligned}
$$

Algorithm 2 Fully Randomized CMP for Saddle Point Problems

Input: $(x^1, y^1) \in X \times Y$, $\alpha_1 > 0$ and $\alpha_2 > 0$, stepsize $\{\gamma_t\}$

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

2: Pick k_t uniformly at random in $\{1, \ldots, b_1\}$ and pick l_t uniformly at random in $\{1, \ldots, b_2\}$

3: $\hat{x}^t = \begin{cases} \operatorname{argmin}_{x_k \in X_k} \{\alpha_1 V_{1,k}(x_k, x_k^t) + \gamma_t \langle \nabla_{x_k} \phi(x^t, y^t), x_k \rangle + \gamma_t h_k(x_k)\}, & k = k_t \\ x_k^t, & k \neq k_t \end{cases}$

4: $\hat{y}^t = \begin{cases} \operatorname{argmin}_{y_l \in Y_l} \{\alpha_2 V_{2,l}(y_l, y_l^t) - \gamma_t \langle \nabla_{y_l} \phi(x^t, y^t), y_l \rangle + \gamma_t p_l(y_l)\}, & l = l_t \\ y_l^t, & l \neq l_t \end{cases}$

5: $x^{t+1} = \begin{cases} \operatorname{argmin}_{x_k \in X_k} \{\alpha_1 V_{1,k}(x_k, x_k^t) + \gamma_t \langle \nabla_{x_k} \phi(\hat{x}^t, \hat{y}^t), x_k \rangle + \gamma_t h_k(x_k)\}, & k = k_t \\ x_k^t, & k \neq k_t \end{cases}$

6: $y^{t+1} = \begin{cases} \operatorname{argmin}_{y_l \in Y_l} \{\alpha_2 V_{2,l}(y_l, y_l^t) - \gamma_t \langle \nabla_{y_l} \phi(\hat{x}^t, \hat{y}^t), y_l \rangle + \gamma_t p_l(y_l)\}, & l = l_t \\ y_l^t, & l \neq l_t \end{cases}$

7: **end for**

8: **Output:** $x_T = \sum_{t=1}^T \gamma_t \hat{x}^t / \sum_{t=1}^T \gamma_t$ and $y_T = \sum_{t=1}^T \gamma_t \hat{y}^t / \sum_{t=1}^T \gamma_t$

Further, we assume that the mappings $\nabla_{x_k} \phi(x, y)$ and $\nabla_{y_l} \phi(x, y)$ are bounded on $X \times Y$. More specifically, for any $k = 1, 2, \ldots, b_1$ and $l = 1, 2, \ldots, b_2$, there exist some constants $M_k^x > 0$ and $M_l^y > 0$ such that

$$\|\nabla_{x_k} \phi(x, y)\|_{x,k,*}^2 \leq M_k^x, \ \|\nabla_{y_l} \phi(x, y)\|_{y,l,*}^2 \leq M_l^y,$$

hold true for any $x \in X$, $y \in Y$.

Under the above assumption, we can define $\mathcal{M}^x = \sum_{k=1}^{b_1} M_k^x$, $\mathcal{M}^y = \sum_{l=1}^{b_2} M_l^y$ which can be viewed as uniform bounds for the gradients. Further, we define the following quantities that will be used in the convergence analysis:

$$\begin{aligned} \mathcal{C}^{xx} &= \textstyle\sum_{k=1}^{b_1} D_k^x L_k^{xx} M_k^x, \ \mathcal{C}^{xy} = \sum_{k=1}^{b_1} D_k^x \sum_{l=1}^{b_2} L_l^{xy} M_l^y / b_2, \\ \mathcal{C}^{yy} &= \textstyle\sum_{l=1}^{b_2} D_l^y L_l^{yy} M_l^y, \ \ \mathcal{C}^{yx} = \sum_{l=1}^{b_2} D_l^y \sum_{k=1}^{b_1} L_k^{yx} M_k^x / b_1, \end{aligned}$$

and denote $\mathcal{B}_1 = \frac{1}{2}\mathcal{M}^x + \mathcal{C}^{xx} + \mathcal{C}^{yx}$, $\mathcal{B}_2 = \frac{1}{2}\mathcal{M}^y + \mathcal{C}^{yy} + \mathcal{C}^{xy}$. We have the following convergence result.

Theorem 1 *Let the sequence of step-sizes $\{\gamma_t\}_{t=1}^T$ in the above algorithm satisfy that $1 \geq \gamma_1 \geq \gamma_2 \geq \cdots \gamma_T \geq 0$. Assume that the Assumption 3 holds and assume further that the functions $|h(x)|$ and $|p(y)|$ are bounded above by $\mathcal{B}_0 > 0$ on $X \times Y$. Then the fully randomized CMP algorithm returns a solution (x_T, y_T) that satisfies*

$$\mathbb{E}[\psi(x_T, y^*) - \psi(x^*, y_T)] \leq \frac{\alpha_1 b_1 \Theta_1 + \alpha_2 b_2 \Theta_2 + 4\mathcal{B}_0 + \sum_{t=1}^T \gamma_t^2 \left(\frac{\mathcal{B}_1}{\alpha_1} + \frac{\mathcal{B}_2}{\alpha_2}\right)}{\sum_{t=1}^T \gamma_t},$$

where $\Theta_1 = V_1(x^*, x^1) := \sum_{k=1}^{b_1} V_{1,k}(x_k^*, x_k^1)$ and $\Theta_2 = V_1(y^*, y^1) := \sum_{l=1}^{b_2} V_{2,l}(y_l^*, y_l^1)$. In particular, when setting $\gamma_t \equiv \frac{1}{\sqrt{T}}, \forall t$, $\alpha_1 = \sqrt{\mathcal{B}_1 / b_1 \Theta_1}$ and $\alpha_2 = \sqrt{\mathcal{B}_2 / b_2 \Theta_2}$, this simplifies to

$$\mathbb{E}[\psi(x_T, y^*) - \psi(x^*, y_T)] \leq \frac{2\sqrt{b_1\mathcal{B}_1\Theta_1} + 2\sqrt{b_2\mathcal{B}_2\Theta_2} + 4\mathcal{B}_0}{\sqrt{T}}.$$

The above convergence result implies that with properly selected stepsize, for example, a decaying stepsizes $\gamma_t = \mathcal{O}\left(1/\sqrt{t}\right)$, the fully randomized CMP yields a convergence rate of $\mathcal{O}(1/\sqrt{T})$. This is slower than the batch CMP, but comes with a cheaper cost per iteration, since only one block from the primal variable and one from the dual variable are updated. Notice that the convergence rate is worse than the $\mathcal{O}(1/T)$ convergence rate achieved by the randomized primal dual algorithm (see, e.g., [49]). However, the latter typically requires a full update of the primal variable, whereas the proposed algorithm only updates a random block of the primal variable. Therefore, a slower convergence is somewhat expected. It is also worthwhile to point out that a recent paper by [51] also considered randomized block coordinate schemes for the original mirror prox algorithm in the context of solving variational inequalities. Their result differs from ours in two aspects: first, our algorithm is specifically designed to address composite saddle point problems and able to handle nonsmooth regularization terms; second, our algorithm randomly chooses one block from primal variable and another one from dual variable, while their algorithmic scheme would choose only one block either from the primal or dual variable. Another recent work [52] introduced a variance reduced stochastic primal–dual method which also performs a similar randomized update for the primal and dual variables with $O(1)$ per-iteration cost in the inner loop of their algorithm; however, the algorithm requires both Lipschitz smoothness and strong convexity of the primal objectives, which clearly do not apply to the problem we consider in this paper.

3 Composite Mirror Prox Algorithms for Point Process Models

As shown in Sect. 1.3, the original optimization problem for point process model estimation can be formulated as a saddle point problem (9). The resulting saddle point problem can be solved with the Composite Mirror Prox algorithm (CMP) outlined in Sect. 2 with

$$X = \mathbb{R}^n_+;\ Y = \mathbb{R}^m_+;\ \phi(x, y) = s^T x - y^T A x + c_0;\ p(y) = -\sum\nolimits_{i=1}^m c_i \log(y_i). \quad (14)$$

Based on Assumption 1 and (10), the proximal operators of $h(x)$ and $p(y)$ can be exactly computed when selecting the distance generating functions to be $\omega_1(x) = \omega(x)$ and $\omega_2(y) = \frac{1}{2}\|y\|_2^2$, associated with norms $\|\cdot\|_x$ and $\|\cdot\|_y = \|\cdot\|_2$, respectively. In the wake of this fact, we now apply CMP and its fully randomized block-decomposition variant to solving (9). In particular, we adopt the mixed proximal setup by setting $\omega(u) = \alpha\omega_x(x) + \frac{1}{2}\|y\|_2^2$ and the norm $\|u\| = \sqrt{\alpha\|x\|_x^2 + \|y\|_2^2}$ for some positive number $\alpha > 0$.

Algorithm 3 CMP for Point Process Models

Input: $x^1 \in \mathbb{R}^n_+$, $y^1 \in \mathbb{R}^m_{++}$, $\alpha > 0$, $\gamma_t \geq 0$
1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \hat{x}^t = \mathrm{Prox}_{x^t}^{\gamma_t h/\alpha}\big(\gamma_t(s - A^T y^t)/\alpha\big)$
3: $\quad \hat{y}_i^t = Q^{\gamma_t}(\gamma_t(a_i^T x^t - y_i^t)), i = 1, \ldots, m$
4: $\quad x^{t+1} = \mathrm{Prox}_{x^t}^{\gamma_t h/\alpha}\big(\gamma_t(s - A^T \hat{y}^t)/\alpha\big)$
5: $\quad y_i^{t+1} = Q^{\gamma_t}(\gamma_t(a_i^T \hat{x}^t - \hat{y}_i^t)), i = 1, \ldots, m$
6: **end for**
7: **Output:** $x_T = \sum_{t=1}^T \gamma_t \hat{x}^t / \sum_{t=1}^T \gamma_t$ and $y_T = \sum_{t=1}^T \gamma_t \hat{y}^t / \sum_{t=1}^T \gamma_t$

3.1 The Composite Mirror Prox Algorithm

Applying CMP to solving problem (9) then gives rise to Algorithm 3. In terms of iteration cost, Algorithm 3 is highly efficient. The y-updates can easily be computed in parallel. Below we provide the iteration complexity analysis for the algorithm.

Denote $f(x) = L(x) + h(x)$. Given any subset $X \subset \mathbb{R}^n_+$, let $Y[X] := \{y : y_i = 1/(a_i^T x), i = 1, \ldots, m, x \in X\}$. Clearly, $Y[X] \subset \mathbb{R}^m_{++}$. Following Lemma 1, we arrive at the result below.

Theorem 2 *Assume we have some a priori information on the optimal solution to problem in* (1)*: a convex compact set $X_0 \subset \mathbb{R}^n_+$ containing x_* and a convex compact set $Y_0 \subset \mathbb{R}^m_{++}$ containing $Y[X_0]$. Denote $\Theta[X_0] = \max_{x \in X_0} V_\omega(x, x^1)$ and $\Theta[Y_0] = \max_{y \in Y_0} \frac{1}{2}\|y - y^1\|_2^2$. Let $\mathcal{L} = \|A\|_{x \to 2} := \max_{x \in \mathbb{R}^n_+ : \|x\|_x \leq 1} \{\|Ax\|_2\}$ and let stepsizes in Algorithm 3 satisfy $0 < \gamma_t \leq \sqrt{\alpha}\mathcal{L}^{-1}$ for all $t > 0$. We have*

$$f(x_T) - f(x_*) \leq \frac{\alpha\Theta[X_0] + \Theta[Y_0]}{\sum_{t=1}^T \gamma_t} \tag{15}$$

In particular, by setting $\gamma_t = \sqrt{\alpha}\mathcal{L}^{-1}$ for all t and $\alpha = \Theta[Y_0]/\Theta[X_0]$, one further has

$$f(x_T) - f(x_*) \leq \frac{\sqrt{\Theta[X_0]\Theta[Y_0]}\|A\|_{x \to 2}}{T}. \tag{16}$$

Remark 4 As an immediate result, the algorithm exhibits an overall $\mathcal{O}(1/t)$ rate of convergence, which is better than the $\mathcal{O}(1/\sqrt{t})$ rate for a non-smooth optimization alternative such as MD [17] and matches the rate (best known so far) achieved by the NoLips algorithm recently established in [20]. Aside from achieving same complexity estimates, these two algorithms, CMP and NoLips, are fundamentally distinct from the algorithmic point of view and they behave differently in practice as later illustrated in Sect. 4.1. NoLips is a purely primal algorithm, while CMP builds on solving an equivalent saddle point problem.

Remark 5 Note that Algorithm 3 works without requiring $a_i^T x > 0, \forall i$, or any global Lipschitz continuity of the original objective function. The prior sets X_0 and Y_0 only appear in the theoretical guarantee. Neither set is involved as the Algorithm 3 proceeds. Knowing the geometry of these prior sets can guide the choice of a favorable proximal

setup. We shall provide in Sect. 4 several illustrations of the practical benefits of an appropriate choice of proximal setup. Furthermore, compact prior sets X_0 and Y_0 are easy to construct in practice and in fact readily available for the problem of interest. For example, when $h(x)$ is positive homogeneous, we can set X_0 to be

$$X_0 = \left\{ x \in \mathbb{R}^n_+ : s^T x + h(x) \leq \sum_{i=1}^m c_i \right\}. \tag{17}$$

Clearly, X_0 is convex and compact. The reason why $x_* \in X_0$ is due to the fact:

Proposition 1 *Assume that $h(x)$ is positive homogeneous, i.e., for any $a \in \mathbb{R}$, $h(ax) = |a|h(x)$. Then the optimal solution x_* to problem (1) satisfies*

$$s^T x_* + h(x_*) = \sum\nolimits_{i=1}^m c_i.$$

Proof This is because, for any $t > 0$, tx_* is a feasible solution and the objective at this point is $\phi(t) := L(tx_*) + h(tx_*) = t(s^T x_* + h(x_*)) - \sum_{i=1}^m c_i \log(a_i^T x_*) - \log(t)\sum_{i=1}^m c_i$. By optimality, $\phi'(1) = 0$, i.e. the desired equation holds. □

Remark 6 Theorem 2 implies that the performance of Algorithm 3 is essentially determined by the distance between the initial point (x^1, y^1) to the optimal solution (x_*, y_*). Therefore, if the initial point is close enough to the optima, then one can expect the algorithm to converge quickly. In practice, the optimal choice of $\alpha = \frac{1}{2}\|y^1 - y_*\|_2^2 / V_\omega(x_*, x^1)$ is often unknown. One can instead select α from empirical considerations, for instance by treating α as a hyper-parameter and tuning it during a burn-in phase or with a validation set.

3.2 Fully Randomized Variants for Problems with High Dimension and Large Sample

While the saddle point formulation overcomes the non-Lipschitzness issue of the original objective, it also requires the introduction of m dual variables, where m equals to the number of data points. Hence, if one would consider a problem with a large number of data points, the computation cost of the dual update at each iteration would slow down Algorithm 3. Similarly, if one would consider problems with high dimensions, the computation cost of the primal update at each iteration would also slow it down. Thus, in order to efficiently tackle problems with large samples and high dimensions, we use a block-decomposition strategy with randomization, to develop the fully randomized CMP, updating only a block of primal and dual variables at a time.

With a slight abuse of notation, denote $x = [x_1; \ldots; x_{b_1}]$ and $A = [\check{A}_1, \ldots, \check{A}_b]$, where $x_k \in \mathbb{R}^{n_k}$, $\check{A}_k \in \mathbb{R}^{m \times n_k}$, $k = 1, \ldots, b_1$ such that $n_1 + \ldots + n_{b_1} = n$. Let us further assume that the penalty function $h(x)$ is separable, i.e. $h(x) = \sum_{k=1}^{b_1} h(x_k)$. Let us denote $y = [y_1; \ldots; y_{b_2}]$ and $A = [A_1; \ldots; A_{b_1}]$, where $y_l \in \mathbb{R}^{m_l}$, $A_l \in$

Algorithm 4 Fully Randomized CMP for Point Process Models

Input: $x^1 \in \mathbb{R}^n_+$, $y^1 \in \mathbb{R}^n_{++}$

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

2: Pick k_t uniformly at random in $\{1, \ldots, b_1\}$ and pick l_t uniformly at random in $\{1, \ldots, b_2\}$

3:
$$\hat{x}^t = \begin{cases} \mathrm{Prox}^{\gamma_t h_k/\alpha}_{x^t_k}\left(\gamma_t(s_k - \check{A}_k^T y^t)/\alpha\right), & k = k_t \\ x^t_k, & k \neq k_t \end{cases}$$
$$\hat{y}^t = \begin{cases} Q^{\gamma_t}(\gamma_t(A_l x^t - y^t_l)), & l = l_t \\ y^t_l, & l \neq l_t \end{cases}$$

4:
$$x^{t+1} = \begin{cases} \mathrm{Prox}^{\gamma_t h_k/\alpha}_{x^t_k}\left(\gamma_t(s_k - \check{A}_k^T \hat{y}^t)/\alpha\right), & k = k_t \\ x^t_k, & k \neq k_t \end{cases}$$
$$y^{t+1} = \begin{cases} Q^{\gamma_t}(\gamma_t(A_l \hat{x}^t - y^t_l)), & l = l_t \\ y^t_l, & l \neq l_t \end{cases}$$

5: **end for**

6: **Output:** $x_T = \sum_{t=1}^T \gamma_t \hat{x}^t / \sum_{t=1}^T \gamma_t$ and $y_T = \sum_{t=1}^T \gamma_t \hat{y}^t / \sum_{t=1}^T \gamma_t$

$\mathbb{R}^{m_l \times n}$, $l = 1, \ldots, b_2$ such that $m_1 + \ldots + m_{b_2} = m$. The fully randomized Composite Mirror Prox algorithm tailored to solve (9) is described in Algorithm 4.

Directly applying Theorem 1, we obtain that with properly decaying stepsizes,

$$\mathbb{E}[\psi(x_T, y^*) - \psi(x^*, y_T)] \leq \mathcal{O}\left(\frac{\sqrt{b_1} + \sqrt{b_2}}{\sqrt{T}}\right).$$

Unlike the full batch version, the fully randomized Composite Mirror Prox algorithm exhibits a slower rate of convergence, i.e., $\mathcal{O}(1/\sqrt{T})$, yet with relatively cheaper iteration cost, making it attractive for large scale problems. Note that, without additional assumptions, the above error bound does not necessarily imply a guarantee in terms of function values $\mathbb{E}[f(x_T) - f_*] \leq O(1/\sqrt{T})$. Indeed, establishing such a result can be challenging when considering randomized algorithms for general saddle point problems, as pointed out in [49]. However, if one would make the additional assumption that $\psi(x, y)$ is strongly convex in x over X and strongly concave in y over Y (and the corresponding constants are known), then one could obtain a convergence guarantee in terms of primal function values, moreover with a faster convergence rate; see, e.g., [48,51,53]. We leave this for future investigation.

4 Applications

In this section, we present numerical experiments for the proposed algorithms as applied to the three examples introduced earlier: Poisson imaging, temporal recommendation systems, and social network estimation. For clarity of the exposition and fairness of the comparison, we compare the proposed algorithms to first order optimization algorithms previously considered and used successfully in the context of point process model estimation. Algorithms such as ADMM that require solving expensive subproblems are therefore not considered in the numerical experiments.

4.1 Poisson Imaging

In this experiment, we examine the Poison imaging problem (4). For simplicity, we do not consider any regularization penalty for this particular application, namely, $h(x) = 0$. Invoking Proposition 1, we have $\sum_{j=1}^{n} x_j = \sum_{i=1}^{m} c_i =: \theta$. We can add to problem (4) the above equality constraint without affecting its optimality. Invoking the saddle point formulation in the previous section, solving (4) is equivalent to solving the convex-concave saddle point problem

$$\min_{x \in \Delta_n} \max_{y \in \mathbb{R}^m_{++}} -y^T Ax + \sum_{i=1}^{m} c_i \log(y_i) + \tilde{\theta},$$

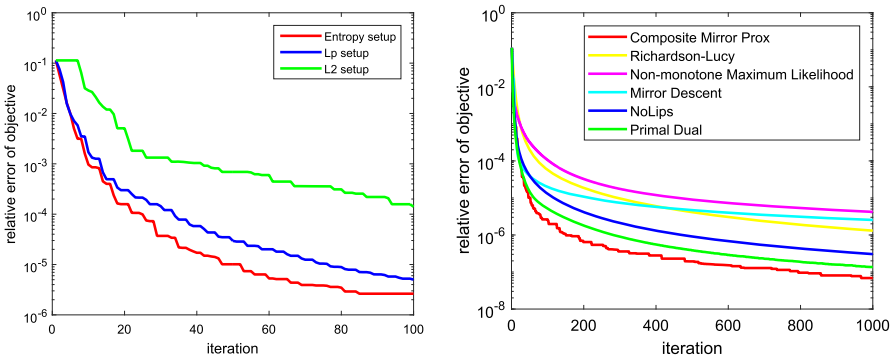
where $\Delta_n = \{x \in \mathbb{R}^n_+ : \sum_{j=1}^{n} x_j = \theta\}$ and $\tilde{\theta} = 2\theta - \sum_{i=1}^{m} c_i \log(c_i)$ is a constant.

Remark 7 Let x_* be the true image. Note that when there is no Poisson noise, $c_i = [Ax_*]_i$ for all i. In this case, the optimal solution y_* corresponding to the y-component of the saddle point problem (4.1) is given by $y_{*,i} = c_i/[Ax_*]_i = 1, \forall i$. Thus, we may hope that under the Poisson noise, the optimal y_* is still close to 1. Assuming that this is the case, the efficiency estimate for T-step CMP algorithm after invoking Theorem 2 when setting $\alpha = r^2 m$ for some $r > 0$ and the distance generating function $\omega(x) = \sum_{j=1}^{n} x_j \log(x_j)$, will be

$$\mathcal{O}(1)\left(\log(n) + \frac{1}{2r^2}\right)\frac{r\theta\sqrt{m}\|A\|_{1\to 2}}{T}.$$

Since A is $m \times n$ stochastic matrix, the Euclidean norms of columns in A are of order $\mathcal{O}(m^{-1/2})$, yielding the efficiency estimate $\mathcal{O}\left(\log(n) \cdot \theta/T\right)$. It is worth pointing out that the dependence on dimension is only logarithmic, making the efficiency estimate nearly optimal in terms of the dependence on dimension. In contrast, Euclidean extragradient methods or primal–dual algorithms would have a worse dependence on dimension, e.g., $\mathcal{O}(\sqrt{n})$ in this case. This difference is clear in the numerical results discussed below. In fact, if we look at what happens in this model when x_* is "uniform", i.e., all entries in x_* are θ/n, the optimal value is $\theta - \theta\log(\theta) + \theta\log(n)$, which is typically of order $\mathcal{O}(\theta)$. This implies that relative to optimal value, the rate of convergence is about $\mathcal{O}(\log(n)/T)$.

Experimental Setup We compare three different choices of distance generating functions (see [31]) to set up the proximal operator for the proposed algorithm: (i) *entropy setup*: $\omega(x) = \sum_{j=1}^{n} x_j \log(x_j)$; (ii) ℓ_2*-setup*: $\omega(x) = \frac{1}{2}\|x\|_2^2$; (iii) ℓ_p*-setup*: $\omega(x) = C\sum_{j=1}^{n} |x_j|^p$, where $C = 2e\log(n)$ and $p = 1 + 1/(2\log(n))$. We compare to several existing algorithms: the classic Richardson-Lucy algorithm [54], Mirror Descent (MD) [17], Non-monotone Maximum Likelihood (NMML) [15], Primal Dual [38], and the recent NoLips algorithm [20]. For the CMP algorithm under each setup, the stepsize is self-tuned using line-search as described in [37], while the the scaling factor $\alpha > 0$ is fined-tuned. More specifically, we run the algorithm among



(a) CMP **(b)** CMP vs. All

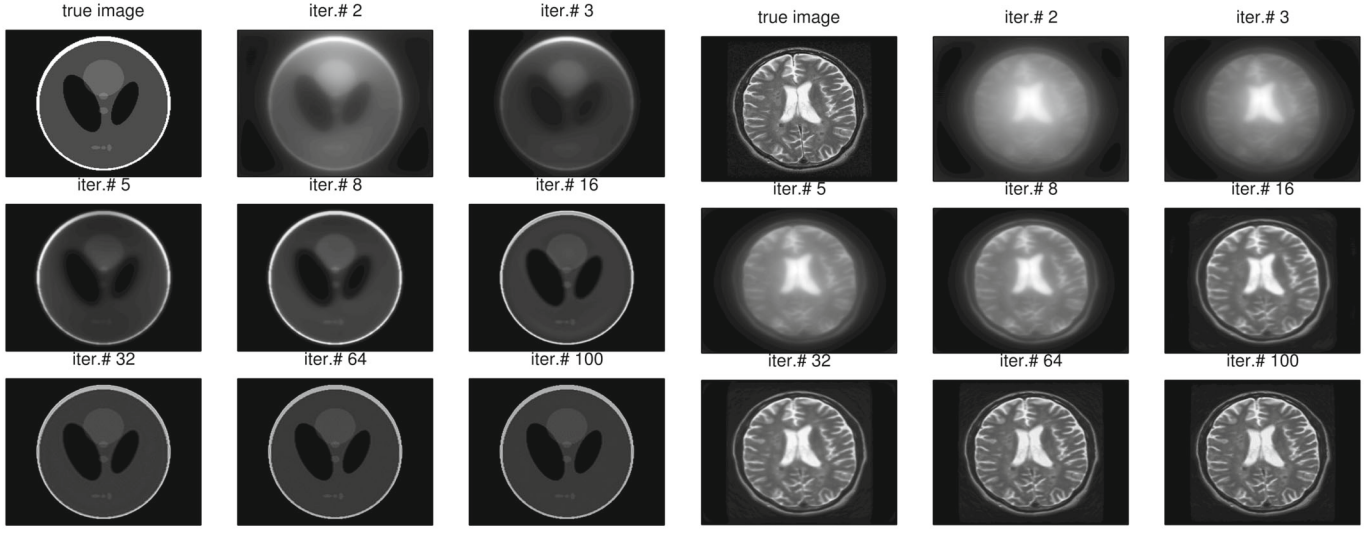
Fig. 1 Poisson imaging reconstruction: **a** convergence behaviors of the CMP algorithm under different proximal setups, **b** convergence comparison among CMP, Richardson-Lucy, Mirror Descent, NMML, and NoLips

a pre-fixed range of values $\{10, 1, 0.5, 0.1, 0.01\}$ of α and chose the one that outputs the best accuracy within 100 iterations. For the Richardson-Lucy algorithm, there is no tuning parameter. For MD, the stepsize is in the order of $O(\gamma/\sqrt{t})$ where γ is also fine-tuned through a given range of values $\{10, 1, 0.5, 0.1, 0.01\}$. For NoLips, the stepsiz $\gamma = \mathcal{O}(1/L)$ (L is the unknown relative Lipschitz constant) is a constant, which is also fine-tuned among the same set of choices. For NMML, we follow the same stepsize used in the original paper (which is explicitly given) and fine tune the corresponding control parameter used in the algorithm. For Primal Dual algorithm, the stepsize is set to be a constant and is also fine-tuned.

Numerical Results We run experiments on several phantom images of size 256×256, and we build the matrix A, which is of size 43530×65536. To evaluate the efficiency, we consider the noiseless situation; hence, the optimal solution and objective value are known. For all algorithms, we evaluate and compare the relative accuracy, i.e., $(\min_{1\le i\le t} f(x_i) - f_*)/f_*$. Results are presented in Fig. 1. From Fig. 1a, we can see that aside from the advantage of preserving positivity without projections, the CMP algorithm under the entropy setup also converges faster than the commonly used ℓ_2 setup. Hence, we adopt the entropy setup for both MD and CMP in the subsequent text. Figure 1b demonstrates that CMP consistently outperforms the competitors including the NoLips and Primal Dual algorithms, even though they attain the same convergence rate theoretically. Figure 2 provides mid-slices of recovery images of the CMP algorithm; it can be seen that CMP is able to provide relatively good recovery of the true images within less than 100 iterations. These results demonstrate that CMP is a competitive algorithm for solving Poisson imaging reconstruction.

4.2 Temporal Recommendation System

We consider the same experimental setup as in (8) for temporal recommendation systems). Here we consider an alternative convex formulation of (8):



(a) Shepp-Logan phantom. **(b)** 2D brain phantom

Fig. 2 Poisson imaging reconstruction: mid-slices of recovery images from the CMP algorithm within 100 iterations

Table 1 Datasets for temporal recommendation systems

Dataset	User	Item	Pair	Event
Synthetic	64	64	2048	2,048,000
Last.fm (small)	297	423	492	31,353
Last.fm (medium)	568	1162	1822	127,724
Last.fm (large)	727	2247	6737	454,375

$$\min_{U\geq 0, A\geq 0, U', A'} L(U, A) + \gamma_1 \|U'\|_{\text{nuc}} + \gamma_2 \|A'\|_{\text{nuc}} + \rho\|U - U'\|_2^2 + \rho\|A - A'\|_2^2, \tag{18}$$

where the negative log-likelihood term is explicitly given by

$$L(U, A) = \sum_{i,j} \Big[T U_{ij} + \sum_{n=1}^{N_{ij}(T)} [A_{ij} g(T - \tau_{ijn}) - \log(U_{ij} + A_{ij} \sum_{l<n} k(\tau_{ijn} - \tau_{ijl}))] \Big].$$

The function $g(t)$ is defined as $g(t) = \int_0^t k(\tau) d\tau$ with $k(\tau) = \exp(-\tau), \tau > 0$. Matrices U', A' are copies of variables U, A to decouple the nuclear norm constraints and the positivity constraints.

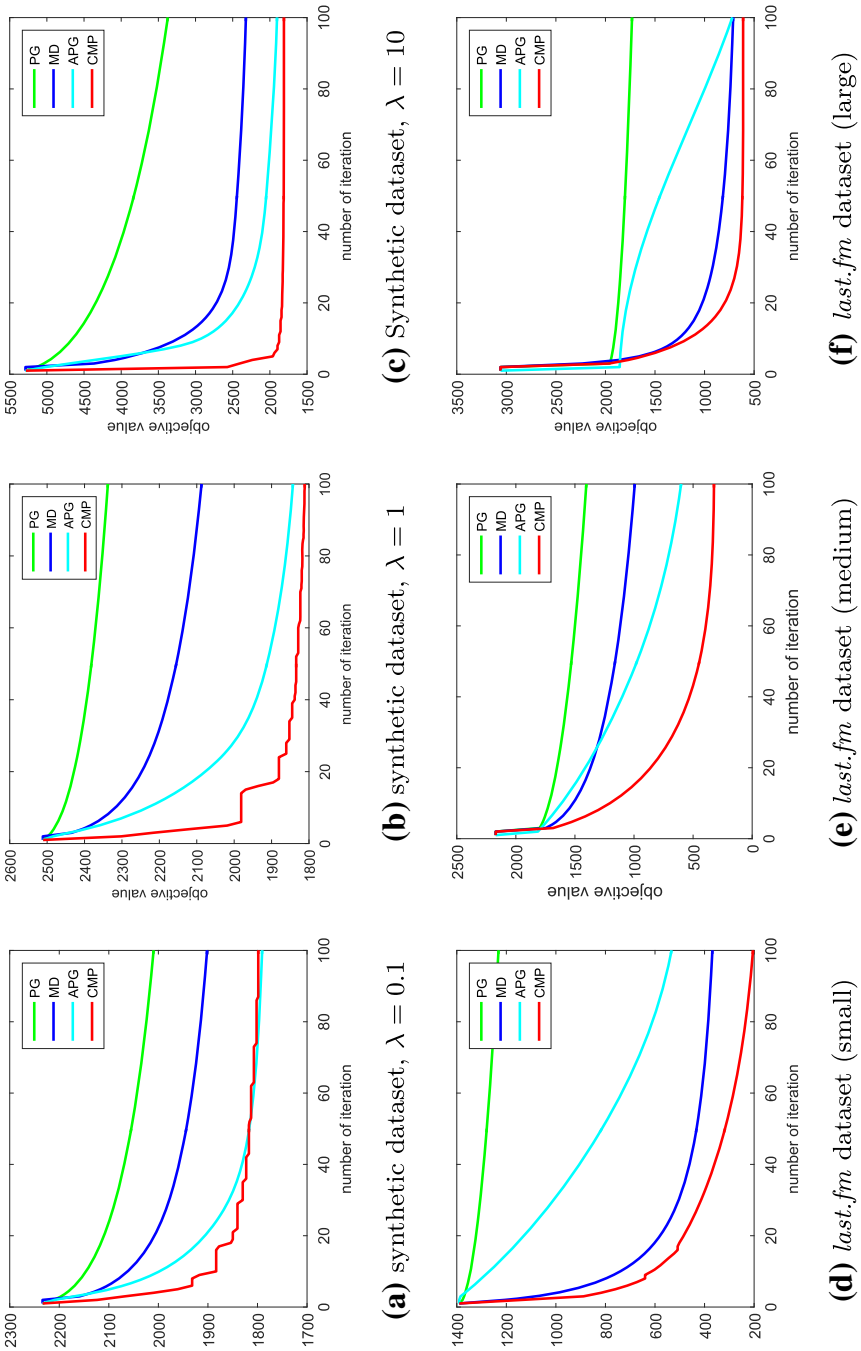
Experimental Setup We compare CMP to several algorithms including Mirror Descent (MD, non-Euclidean setup) for composite objective [32], proximal gradient descent (PG, Euclidean setup) and accelerated proximal gradient (APG, Euclidean setup) [9]. For CMP and MD, we use the following proximal setup for $x = [U, A, U', A']$: $\omega(x) = \sum_{ij} U_{ij} \log U_{ij} + \sum_{ij} A_{ij} \log A_{ij} + \frac{1}{2}\|U'\|_F^2 + \frac{1}{2}\|A'\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. The stepsize of CMP is self-tuned using line-search as described in [37] at each iteration. The stepsize of MD is in the order of $O(\gamma/\sqrt{t})$ where γ is fine-tuned as described earlier. For PG and APG, the stepsizes are selected through back-tracking line-search since the objective is non-globally Lipschitz continuous.

Numerical Results We run the experiments on both synthetic and real-world datasets as described in Table 1. The number of events in the *last.fm* dataset ranges from 30,000 to 500,000. We set the regularization parameters to be the same and range from $\{0.1, 1, 10\}$ and set $\rho = 1$. The results are presented in Fig. 3. Figure 3 again demonstrates that i) using non-Euclidean setup improves the performance ii) when taking into account the non-Lipschitzness, CMP performs better empirically than MD and APG on these problems.

4.3 Social Network Estimation

In the last experiment, we consider the convex problem introduced in (6) for estimating the influence matrix among users in a social network.

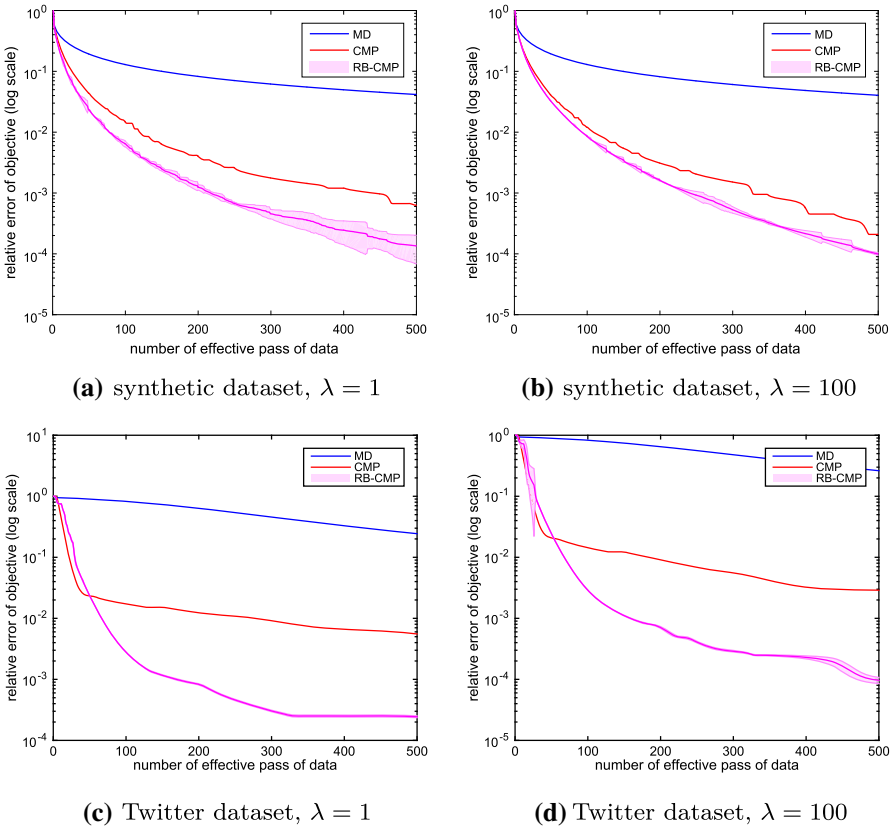
Experimental Setup We test the performance of the proposed fully randomized block CMP (denoted as RB-CMP) and compare it to Composite Mirror Prox (CMP) and



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

(d) *last.fm* dataset (small) **(e)** *last.fm* dataset (medium) **(f)** *last.fm* dataset (large)

Fig. 3 Temporal recommendation system



(a) synthetic dataset, $\lambda = 1$ **(b)** synthetic dataset, $\lambda = 100$

(c) Twitter dataset, $\lambda = 1$ **(d)** Twitter dataset, $\lambda = 100$

Fig. 4 Social network estimation

Mirror Descent (MD). For all algorithms, we use the entropy proximal setup for the x-component: $\omega(\mu, A) = \sum_i \mu_i \log(\mu_i) + \sum_{i,j} a_{ij} \log(a_{ij})$. Note that the primal variable (μ, A) can be naturally decomposed into blocks that corresponds to each user, and the dual variable y can be naturally divided into blocks that corresponds to the datapoints of each user. In our experiment, at each iteration, we randomly pick one user and only update the corresponding blocks in the primal and dual variables using their block gradients. Unlike the full gradient with size $O(n^2 + m)$, the average size of the block gradient effectively reduces to $O(n + m/n)$, where m is the total number of events, and n is the number of users. We compare these algorithms both on synthetic and real Twitter datasets. The synthetic dataset consists of 50 users and 50,000 events. The Twitter dataset consists of 100 users and 98,927 events.

Numerical Results We run the three algorithms with their best-tuned parameters respectively, under different regularization parameters $\lambda \in \{1, 100\}$. For MD and RB-CMP, we fine tune the stepsize as described earlier; for CMP, the stepsize is self-tuned via line-search at each iteration. We evaluate their relative accuracy vs number of effective passes through data. The results are presented in Fig. 4, which indicate that CMP

performs consistently better than MD. Furthermore, the randomized block variant performs even better on large datasets.

5 Conclusion

We introduced a saddle point formulation of penalized maximum likelihood estimation of point process models. This formulation overcomes the issue arising from the non-Lipschitzness of the likelihood objective, which is an obstacle to a direct application of common first order optimization algorithms. The saddle point algorithm we presented enjoys a $\mathcal{O}(1/t)$ convergence rate, in contrast to the typical $\mathcal{O}(1/\sqrt{t})$ rate for non-smooth optimization. We also presented a fully randomized block-decomposition variant, with a slower rate of convergence yet with a cheaper cost per iteration, making it appropriate for problems with both large samples and high dimensions. The two algorithms demonstrated competitive performance on several challenging real-world datasets. The extension of the proposed approach to more general classes of non-Lipschitz problems is an interesting venue for future work.

Acknowledgements This work was first presented at the Fifth International Conference on Continuous Optimization (ICCOPT) in August 2016. This work was supported by NSF CMMI-1761699, NCSA Faculty Fellowship, NSF CCF-1740551, the project Titan (CNRS-Mastodons), the MSR-Inria joint centre, the project Macaron (ANR-14-CE23-0003-01), the program "Learning in Machines and Brains" (CIFAR), and faculty research awards. The authors would like to thank Anatoli Juditsky, Julien Mairal, Arkadi Nemirovski, and Joseph Salmon for fruitful discussions. The authors are also grateful to the reviewers and the editor for their valuable remarks and thoughtful comments.

A Convergence Analysis of Fully Randomized CMP

To prove the above result, we need the technical lemma below.

Lemma 2 *Suppose $U \subset E$ is closed convex on an Euclidean space E, $V(u, u')$ is the Bregman divergence induced by some distance generating function $\omega(x)$ that is 1-strongly convex with respect to some norm $\|\cdot\|$ on U, and $\Psi(u)$ is convex. For any $u \in U$, and $g \in E$, define the prox operator $u^+(g) := argmin_{u' \in U}\{\alpha V(u', u) + \langle g, u'\rangle + \Psi(u')\}$, where $\alpha > 0$. Then*

(i) For a given $g \in E$, denote $u^+ := u^+(g)$. It holds true that

$$\langle g, u^+ - u'\rangle + \Psi(u^+) - \Psi(u') \le \alpha[V(u', u) - V(u', u^+) - V(u^+, u)], \forall u' \in U. \tag{19}$$

(ii) The prox operator is $\frac{1}{\alpha}$-Lipschitz, i.e., $\forall g_1, g_2 \in E$,

$$\|u^+(g_1) - u^+(g_2)\| \le \frac{1}{\alpha}\|g_1 - g_2\|_*,$$

where $\|\cdot\|_*$ is the dual norm to $\|\cdot\|$.

Similar results can be found in [55] and [56]. In fact, this can be proved based on the optimality condition and the generalized triangle inequality of Bregman divergence. For completeness, we provide a proof below.

Proof We first prove (i). By optimality of u^+, we have

$$\langle \alpha[\nabla\omega(u^+) - \nabla\omega(u)] + g + \partial\Psi(u^+), u^+ - u'\rangle \leq 0, \forall u' \in U.$$

By convexity of $\Psi(u)$, we also have $\Psi(u^+) - \Psi(u') \leq \langle \partial\Psi(u^+), u^+ - u'\rangle, \forall u' \in U$. Combining these two inequalities implies that

$$\begin{aligned}\langle g, u^+ - u'\rangle + \Psi(u^+) - \Psi(u') &\leq \alpha\langle \nabla\omega(u^+) - \nabla\omega(u), u' - u^+\rangle \\ &= V(u', u) - V(u', u^+) - V(u^+, u).\end{aligned}$$

The latter equality follows from the definition of Bregman divergence. We now prove (ii). Denoting $u_1 = u^+(g_1)$ and $u_2 = u^+(g_2)$, by optimality of u_1 and u_2, we have

$$\begin{aligned}-[\alpha\nabla\omega(u_1) - \alpha\nabla\omega(u) + g_1] &\in \partial(\Psi + \delta_U(\cdot))(u_1) \\ -[\alpha\nabla\omega(u_2) - \alpha\nabla\omega(u) + g_2] &\in \partial(\Psi + \delta_U(\cdot))(u_2)\end{aligned}$$

where $\delta_U(\cdot)$ is the indicator function of the set U. By monotonicity of the subgradient, it holds that

$$([\alpha\nabla\omega(u_2) - \alpha\nabla\omega(u) + g_2] - [\alpha\nabla\omega(u_1) - \alpha\nabla\omega(u) + g_1])^T (u_1 - u_2) \geq 0.$$

Hence,

$$\langle g_2 - g_1, u_1 - u_2\rangle \geq \alpha\langle \nabla\omega(u_1) - \omega(u_2), u_1 - u_2\rangle \geq \alpha\|u_1 - u_2\|^2.$$

The last inequality comes from the 1-strongly convexity of $\omega(u)$. By Cauchy-Schwarz inequality, we obtain the desired result. □

We now provide the convergence analysis for Theorem 1. Invoking the definition of x^{t+1} and y^{t+1} and Lemma 2, we end up with for any $t = 1, \ldots, T$ and for any $x \in X, y \in Y$:

$$\begin{aligned}&\gamma_t[\langle \nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t),\ x_{k_t}^{t+1} - x_{k_t}\rangle + h_{k_t}(x_{k_t}^{t+1}) - h_{k_t}(x_{k_t})] \\ &\quad\leq \alpha_1[V_{1,k_t}(x_{k_t}, x_{k_t}^t) - V_{1,k_t}(x_{k_t}, x_{k_t}^{t+1}) - V_{1,k_t}(x_{k_t}^{t+1}, x_{k_t}^t)]\,,\end{aligned} \tag{20}$$

and

$$\begin{aligned}&\gamma_t[\langle -\nabla_{y_{l_t}}\phi(\hat{x}^t, \hat{y}^t),\ y_{l_t}^{t+1} - y_{l_t}\rangle + p_{l_t}(y_{l_t}^{t+1}) - p_{l_t}(y_{l_t})] \\ &\quad\leq \alpha_2[V_{2,l_t}(y_{l_t}, y_{l_t}^t) - V_{2,l_t}(y_{l_t}, y_{l_t}^{t+1}) - V_{2,l_t}(y_{l_t}^{t+1}, y_{l_t}^t)]\,,\end{aligned} \tag{21}$$

We first take a look at Eq. (20). This implies that

$$\begin{aligned}
&\gamma_t[\langle \nabla_{x_{k_t}}\phi(x^t, y^t),\ x^t_{k_t} - x_{k_t}\rangle + h_{k_t}(x^t_{k_t}) - h_{k_t}(x_{k_t})] \\
&\quad \le \underbrace{\alpha_1[V_{1,k_t}(x_{k_t}, x^t_{k_t}) - V_{1,k_t}(x_{k_t}, x^{t+1}_{k_t})] + \gamma_t[h_{k_t}(x^t_{k_t}) - h_{k_t}(x^{t+1}_{k_t})]}_{A_t} \\
&\quad \underbrace{-\alpha_1 V_{1,k_t}(x^{t+1}_{k_t}, x^t_{k_t}) + \gamma_t\langle \nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t),\ x^t_{k_t} - x^{t+1}_{k_t}\rangle}_{B_t} \\
&\quad + \underbrace{\gamma_t[\langle \nabla_{x_{k_t}}\phi(x^t, y^t) - \nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t),\ x^t_{k_t} - x_{k_t}\rangle]}_{C_t}.
\end{aligned} \tag{22}$$

First, recall that $V_1(x, x') = \sum_{k=1}^{b_1} V_{1,k}(x_k, x'_k)$. This implies that

$$V_1(x, x^t) - V_1(x, x^{t+1}) = V_{1,k_t}(x_{k_t}, x^t_{k_t}) - V_{1,k_t}(x_{k_t}, x^{t+1}_{k_t}).$$

Similarly, by definition of $h(x) = \sum_k h_k(x_k)$, we have $h(x^t) - h(x^{t+1}) = h_{k_t}(x^t_{k_t}) - h_{k_t}(x^{t+1}_{k_t})$. Combing these two facts, we obtain

$$A_t \le \alpha_1[V_1(x, x^t) - V_1(x, x^{t+1}) + \gamma_t[h(x^t) - h(x^{t+1})]. \tag{23}$$

Second, applying Young's inequality, the term B_t can be bounded as

$$\begin{aligned}
B_t &\le -\alpha_1 V_{1,k_t}(x^{t+1}_{k_t}, x^t_{k_t}) + \frac{\gamma_t^2}{2\alpha_1}\|\nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t)\|^2_{x,k_t,*} + \frac{\alpha_1}{2}\|x^t_{k_t} - x^{t+1}_{k_t}\|^2_{x,k_t} \\
&\le \frac{\gamma_t^2}{2\alpha_1}\|\nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t)\|^2_{x,k_t,*} \le \frac{\gamma_t^2 M^x_{k_t}}{2\alpha_1}.
\end{aligned} \tag{24}$$

Third, using the block-Lipschitzness of the gradient, we can obtain an upper bound of the term C_t as follows:

$$\begin{aligned}
C_t &\le \gamma_t\|\nabla_{x_{k_t}}\phi(x^t, y^t) - \nabla_{x_{k_t}}\phi(\hat{x}^t, \hat{y}^t)\|_{x,k_t,*}\|x^t_{k_t} - x_{k_t}\|_{x,k_t} \\
&\le \gamma_t D^x_{k_t}\left(L^{xx}_{k_t}\|\hat{x}^t_{k_t} - x^t_{k_t}\|_{x,k_t} + L^{xy}_{l_t}\|\hat{y}^t_{l_t} - y^t_{l_t}\|_{y,l_t}\right).
\end{aligned} \tag{25}$$

Invoking the Lipschitzness of the prox mappings, we further have

$$\begin{aligned}
C_t &\le \gamma_t^2 D^x_{k_t}\left(\frac{L^{xx}_{k_t}}{\alpha_1}\|\nabla_{x_{k_t}}\phi(x^t, y^t)\|_{x,k_t,*} + \frac{L^{xy}_{l_t}}{\alpha_2}\|\nabla_{y_{l_t}}\phi(x^t, y^t)\|_{y,l_t,*}\right) \\
&\le \gamma_t^2 D^x_{k_t}\left(\frac{L^{xx}_{k_t}M^x_{k_t}}{\alpha_1} + \frac{L^{xy}_{l_t}M^y_{l_t}}{\alpha_2}\right).
\end{aligned} \tag{26}$$

Note that x^t, y^t are independent of the random variables k_t and l_t. Taking expectation with respect to the probability distributions associated with k_t and l_t in Eq. (22), we show that

$$\begin{aligned}
&\gamma_t \mathbb{E}_{k_t,l_t}[\langle \nabla_{x_{k_t}} \phi(x^t, y^t),\ x^t_{k_t} - x_{k_t}\rangle + h_{k_t}(x^t_{k_t}) - h_{k_t}(x_{k_t})] \\
&= \frac{\gamma_t}{b_1} \sum_{k=1}^{b_1} [\langle \nabla_{x_k} \phi(x^t, y^t),\ x^t_k - x_k\rangle + h_k(x^t_k) - h_k(x_k)] \\
&= \frac{\gamma_t}{b_1} [\langle \nabla_x \phi(x^t, y^t),\ x^t - x\rangle + h(x^t) - h(x)] \,. \qquad (27)
\end{aligned}$$

Combining with Eqs. (23), (24), and (25) and taking expectations, we end up with

$$\gamma_t \mathbb{E}[\langle \nabla_x \phi(x^t, y^t),\ x^t - x\rangle + h(x^t) - h(x)] \le b_1 \cdot \mathbb{E}[A_t] + \gamma_t^2 \left(\frac{\mathcal{M}^x + 2\mathcal{C}^{xx}}{2\alpha_1} + \frac{\mathcal{C}^{xy}}{\alpha_2} \right). \qquad (28)$$

Using a similar analysis for Eq. (21), we obtain

$$\gamma_t \mathbb{E}[-\langle \nabla_y \phi(x^t, y^t),\ y^t - y\rangle + p(y^t) - p(y)] \le b_2 \cdot \mathbb{E}[\tilde{A}_t] + \gamma_t^2 \left(\frac{\mathcal{M}^y + 2\mathcal{C}^{yy}}{2\alpha_2} + \frac{\mathcal{C}^{yx}}{\alpha_1} \right). \qquad (29)$$

Combing Eqs. (28) and (29) and invoking the convex-concavity of the function $\phi(x, y)$, we have

$$\gamma_t \mathbb{E}[\psi(x^t, y) - \psi(x, y^t)] \le b_1 \cdot \mathbb{E}[A_t] + b_2 \cdot \mathbb{E}[\tilde{A}_t] + \gamma_t^2 \left(\frac{\mathcal{B}_1}{\alpha_1} + \frac{\mathcal{B}_2}{\alpha_2} \right). \qquad (30)$$

Taking summation over t and further exploiting the convex-concavity of $\psi(x, y)$, we end up with

$$\mathbb{E}[\psi(x_T, y) - \psi(x, y_T)] \le b_1 \sum_{t=1}^{T} \mathbb{E}[A_t] + b_2 \sum_{t=1}^{T} \mathbb{E}[\tilde{A}_t] + \sum_{t=1}^{T} \gamma_t^2 \left(\frac{\mathcal{B}_1}{\alpha_1} + \frac{\mathcal{B}_2}{\alpha_2} \right). \qquad (31)$$

Note that

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}[A_t] \le \alpha_1 V_1(x, x^1) + \gamma_1 h(x^1) - \gamma_T \mathbb{E}[h(x^T)] \le \alpha_1 V_1(x, x^1) + 2\gamma_1 \mathcal{B}_0, \\
&\sum_{t=1}^{T} \mathbb{E}[\tilde{A}_t] \le \alpha_2 V_2(y, y^1) + \gamma_1 p(y^1) - \gamma_T \mathbb{E}[p(y^T)] \le \alpha_2 V_2(y, y^1) + 2\gamma_1 \mathcal{B}_0.
\end{aligned}$$

Therefore, plugging into $x = x^*$, $y = y^*$, we have arrived at the desired result. □

Remark 8 Note that we did not use the independence of k_t and l_t within each iteration to prove the above results. Moreover, the above convergence analysis can be extended to non-uniform sampling, which we omit for simplicity of presentation.

References

1. Bertero, M., Boccacci, P., Desiderà, G., Vicidomini, G.: Image deblurring with Poisson data: from cells to galaxies. Inverse Probl. **25**(12), 123,006 (2009)
2. Prigent, J.L.: Option pricing with a general marked point process. Math. Oper. Res. **26**(1), 50–66 (2001)
3. Cartea, A.: Derivatives pricing with marked point processes using tick-by-tick data. Quant. Financ. **13**(1), 111–123 (2013)
4. Rajaram, S., Graepel, T., Herbrich, R.: Poisson-networks: a model for structured point processes. In: Proceedings of the 10th international workshop on artificial intelligence and statistics, pp. 277–284. Citeseer (2005)
5. Simma, A., Jordan, M.I.: Modeling events with cascades of Poisson processes. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, pp. 546–555 (2010)
6. Zhou, K., Zha, H., Song, L.: Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, pp. 641–649 (2013)
7. Iwata, T., Shah, A., Ghahramani, Z.: Discovering latent influence in online social activities via shared cascade Poisson processes. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 266–274. ACM (2013)
8. Hall, E.C., Willett, R.M.: Tracking dynamic point processes on networks. IEEE Trans. Inf. Theory **62**(7), 4327–4346 (2016)
9. Nesterov, Y.: Gradient methods for minimizing composite functions. Math. Program. **140**(1), 125–161 (2013)
10. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
11. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**(4), 1574–1609 (2009)
12. Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. Optim. Mach. Learn. **2010**(1–38), 3 (2011)
13. Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Program. **162**(1), 83–112 (2017)
14. Harmany, Z.T., Marcia, R.F., Willett, R.M.: This is SPIRAL-TAP: sparse poisson intensity reconstruction algorithms-theory and practice. IEEE Trans. Image Process. **21**(3), 1084–1096 (2012)
15. Sra, S., Kim, D., Schölkopf, B.: Non-monotonic poisson likelihood maximization. Tech. rep., Max Planck Institute for Biological Cybernetics (2008)
16. Tran-Dinh, Q., Kyrillidis, A., Cevher, V.: Composite self-concordant minimization. J. Mach. Learn. Res. **16**(1), 371–416 (2015)
17. Ben-Tal, A., Margalit, T., Nemirovski, A.: The ordered subsets mirror descent optimization method with applications to tomography. SIAM J. Optim. **12**(1), 79–108 (2001)
18. Nemirovski, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley, New York (1983)
19. Figueiredo, M.A., Bioucas-Dias, J.M.: Restoration of poissonian images using alternating direction optimization. IEEE Trans. Image Process. **19**(12), 3133–3145 (2010)
20. Bauschke, H.H., Bolte, J., Teboulle, M.: A descent Lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. Math. Oper. Res. **42**(2), 330–348 (2016)
21. Haihao Lu, R.M.F., Nesterov, Y.: Relatively smooth convex optimization by first-order methods, and applications. SIAM J. Optim. **28**(1), 333–354 (2018)
22. Mohler, G.O., Short, M.B., Brantingham, P.J., Schoenberg, F.P., Tita, G.E.: Self-exciting point process modeling of crime. J. Am. Stat. Assoc. **106**(493), 100–108 (2011)
23. Kapoor, K., Subbian, K., Srivastava, J., Schrater, P.: Just in time recommendations: Modeling the dynamics of boredom in activity streams. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, pp. 233–242 (2015)
24. Du, N., Wang, Y., He, N., Song, L.: Time-sensitive recommendation from recurrent user activities. In: Proceedings of 28th International Conference on Neural Information Processing Systems, pp. 3492–3500 (2015)
25. Moreau, J.J.: Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. France **93**, 273–299 (1965)

26. Teboulle, M.: Entropic proximal mappings with applications to nonlinear programming. Math. Oper. Res. **17**(3), 670–690 (1992)
27. Bauschke, H.H., Borwein, J.M., Combettes, P.L.: Bregman monotone optimization algorithms. SIAM J. Control Optim. **42**(2), 596–636 (2003)
28. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Found. Trends Mach. Learn. **4**(1), 1–106 (2012)
29. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York (2017)
30. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Process. **18**(11), 2419–2434 (2009)
31. Juditsky, A., Nemirovski, A.: First-order methods for nonsmooth large-scale convex minimization: I. General purpose methods; II. Utilizing problems structure. In: Sra, S., Nowozin, S., Wright, S. (eds.) Optimization for Machine Learning, pp. 121–183. The MIT Press, Cambridge (2011)
32. Duchi, J.C., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: COLT 2010—The 23rd Conference on Learning Theory, Haifa, Israel, June 27–29, 2010, pp. 14–26 (2010)
33. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)
34. Lu, Z., Xiao, L.: On the complexity analysis of randomized block-coordinate descent methods. Math. Program. **152**(1–2), 1–28 (2013)
35. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1–2), 1–38 (2014)
36. Yanez, F., Bach, F.: Primal-dual algorithms for non-negative matrix factorization with the Kullback-Leibler divergence. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2257–2261 (2017)
37. He, N., Juditsky, A., Nemirovski, A.: Mirror prox algorithm for multi-term composite minimization and semi-separable problems. Comput. Optim. Appl. **61**(2), 275–319 (2015)
38. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
39. Yang, T., Mahdavi, M., Jin, R., Zhu, S.: An efficient primal dual prox method for non-smooth optimization. Mach. Learn. **98**(3), 369–406 (2015)
40. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. Math. Program. **159**(1–2), 253–287 (2015)
41. Chen, Y., Lan, G., Ouyang, Y.: Optimal primal-dual methods for a class of saddle point problems. SIAM J. Optim. **24**(4), 1779–1814 (2014)
42. He, Y., Monteiro, R.D.: An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. SIAM J. Optim. **26**(1), 29–56 (2016)
43. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. SIAM J. Optim. **2**, 3 (2009)
44. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
45. Chen, X., Lin, Q., Kim, S., Carbonell, J.G., Xing, E.P.: Smoothing proximal gradient method for general structured sparse regression. Ann. Appl. Stat. **6**(2), 719–752 (2012)
46. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss. J. Mach. Learn. Res. **14**(1), 567–599 (2013)
47. Dang, C.D., Lan, G.: Stochastic block mirror descent methods for nonsmooth and stochastic optimization. SIAM J. Optim. **25**(2), 856–881 (2015)
48. Zhang, Y., Xiao, L.: Stochastic primal-dual coordinate method for regularized empirical risk minimization. J. Mach. Learn. Res. **18**(1), 2939–2980 (2017)
49. Dang, C.D.: Randomized first order methods for convex and nonconvex optimization. PhD Thesis (2015)
50. Gao, X., Xu, Y.Y., Zhang, S.Z.: Randomized primal-dual proximal block coordinate updates. J. Oper. Res. Soc. China **7**(2), 205–250 (2019)
51. Yousefian, F., Nedić, A., Shanbhag, U.V.: On stochastic mirror-prox algorithms for stochastic cartesian variational inequalities: randomized block coordinate and optimal averaging schemes. Set Valued Var. Anal. **26**(4), 789–819 (2016)
52. Tan, C., Zhang, T., Ma, S., Liu, J.: Stochastic primal-dual method for empirical risk minimization with o (1) per-iteration complexity. In: Advances in Neural Information Processing Systems, pp. 8366–8375 (2018)

53. Palaniappan, B., Bach, F.: Stochastic variance reduction methods for saddle-point problems. In: Advances in Neural Information Processing Systems, pp. 1416–1424 (2016)
54. Richardson, W.H.: Bayesian-based iterative method of image restoration. JoSA **62**(1), 55–59 (1972)
55. Ben-Tal, A., Nemirovski, A.: Lectures on modern convex optimization: analysis, algorithms, and engineering applications, vol. 2. SIAM (2001)
56. Beck, A.: First-Order Methods in Optimization, vol. 25. SIAM (2017)