# Perturbed Proximal Gradient Algorithm

Gersende FORT

LTCI, CNRS and Telecom ParisTech
Paris, France

Joint work with

- Yves Atchadé (Univ. Michigan, USA)
- Eric Moulines (Ecole Polytechnique, France)

## Outline

## Problem

Convergence of a perturbed version of an iterative algorithm designed to solve

$$\mathrm{argmin}_{\theta \in \Theta} F(\theta) \qquad \text{with } F(\theta) = f(\theta) + g(\theta)$$

where

- $\Theta$ convex subset of a finite-dimensional Euclidean space with scalar product $\langle , \rangle$ and norm $\| \cdot \|$
- the function $f$:$\Theta \to \mathbb{R}$ is a smooth function

  i.e. $f$ is continuously differentiable and there exists $L > 0$ such that

$$\|\nabla f(\theta) - \nabla f(\theta')\| \leq L \; \|\theta - \theta'\|$$

- the function $g$: $\Theta \to (-\infty, \infty]$ is convex, not identically equal to $+\infty$, and lower semi-continuous

"perturbation" since it is a first-order technique and $\nabla f$ is intractable in many applications.

# Outline

# Example 1: Penalized ML inference in Latent variable models (1/2)

- A vector of observations: $\mathrm{Y}$
- A vector of latent variables: $\mathrm{U}$
- A parametric model indexed by $\theta \in \Theta$

**Minimize the negative log-likelihood:**

$$f(\theta) = -\log p(\mathrm{Y};\theta) = -\log \int p(\mathrm{Y},\mathrm{u};\theta)\mu(\mathsf{d}u) = -\log \int p(\mathrm{Y}|\mathrm{u};\theta)\,\phi(\mathrm{u})\mu(\mathsf{d}u)$$

which is (usually) intractable; same thing for the gradient

$$\nabla f(\theta) = -\int \nabla \log p(\mathrm{Y}|\mathrm{u};\theta)\ \frac{p(\mathrm{Y},\mathrm{u};\theta)}{\int p(\mathrm{Y},\mathrm{x};\theta)\mu(\mathsf{d}x)}\mu(\mathsf{d}u)$$

**with some constraint** $\theta \mapsto g(\theta)$ ($\theta$ in a compact, sparsity constraint on $\theta$, $\cdots$)

# Example 1: Penalized ML inference in Latent variable models (2/2)

For example, logistic regression with random effects, under sparsity constraints

$$\mathbf{U} \sim \mathcal{N}_q(0, I)$$

$$Y_i | \mathbf{U} \stackrel{i.i.d.}{\sim} \text{Ber}\left( \frac{\exp\left(x_i'\beta + \sigma\, z_i'\mathbf{U}\right)}{1 + \exp\left(x_i'\beta + \sigma\, z_i'\mathbf{U}\right)} \right)$$

$$\theta = (\beta, \sigma) \in \mathbb{R}^p \times \mathbb{R}_+$$

$$g(\theta) = \lambda \sum_{i=1}^{p} |\beta_i|$$

In this model,

$$\nabla f(\theta) = \int H_\theta(\mathrm{u})\ \pi_\theta(\mathrm{u}) \mathrm{d}\mathrm{u}$$

$$H_\theta(\mathrm{u}) = \sum_{i=1}^{n} \left( Y_i - \frac{\exp\left(x_i'\beta + \sigma\, z_i'\mathrm{u}\right)}{1 + \exp\left(x_i'\beta + \sigma\, z_i'\mathrm{u}\right)} \right) \begin{bmatrix} x_i \\ z_i'\mathrm{u} \end{bmatrix}$$

$\pi_\theta(\mathrm{u}) = \cdots$ sampled through MCMC / data augmentation Polson et al. (2013); Choi and Hobert (2013)

# Example 2: Network structure estimation

- Observations: $N$ i.i.d. samples $Y_i = (y_1^{(i)}, \cdots, y_p^{(i)})$ from a Gibbs distribution on $\mathbb{X}^p$ ($\mathbb{X}$ finite) with intractable normalizing constant

$$\pi_\theta(\mathrm{y}) = \frac{1}{Z_\theta} \exp\left( \sum_{k=1}^{p} \theta_{kk} B_0(y_k) + \sum_{1 \le j < k \le p} \theta_{jk} B(y_j, y_k) \right)$$

- A parametric model indexed by $\theta \in \mathbb{R}^{p \times p}$, symmetric.

**Minimize the (normalized) negative log-likelihood:**

$$f(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \left( \sum_{k=1}^{p} \theta_{kk} B_0(y_k^{(i)}) + \sum_{1 \le j < k \le p} \theta_{jk} B(y_j^{(i)}, y_k^{(i)}) \right) + \log Z_\theta$$

with the intractable constant $Z_\theta$; same thing for the gradient

$$\nabla f(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \bar{B}(y^{(i)}) + \int \bar{B}(\mathrm{u})\, \pi_\theta(\mathrm{du})$$

**with some constraint** $\theta \mapsto g(\theta)$ ($\theta$ in a compact, sparsity constraint on $\theta$, $\cdots$)

## Example 3: Learning on huge data set

- $f$ is the average of many components

$$f(\theta) = \frac{1}{N} \sum_{i=1}^{N} f_i(\theta)$$

Large sum $\Longrightarrow$ prohibitive computational cost $\Longrightarrow$ incremental methods: stochastic approximation of the gradient

$$\nabla f(\theta) \approx \frac{1}{m} \sum_{k=1}^{m} \nabla f_{I_k}(\theta)$$

# Example 4: Online learning and Stochastic Approximation

- The function $f$ is of the form

$$f(\theta) = \int \bar{f}(\theta; \mathrm{u}) \pi(\mathrm{du})$$

  with an unknown $\pi$

- The user is only provided with random samples from $\pi$, so

$$\nabla f(\theta) \approx \frac{1}{m} \sum_{k=1}^{m} H_\theta(X_k)$$

# Outline

The setting

Examples of problems of the form: $\text{argmin}_\theta \{f(\theta) + g(\theta)\}$

The proximal gradient algorithm

The poster session

# When $\nabla f$ is available: a gradient-based approach

Optimization problem:

$$\text{argmin}_{\theta \in \Theta} \left( \underbrace{f(\theta)}_{C^1 \text{ with Lipschitz gradient}} + \underbrace{g(\theta)}_{\text{not differentiable}} \right)$$

Algorithm: Proximal Gradient Nesterov (2004): iterative procedure

$$\theta_{n+1} = \text{Prox}_{\gamma_{n+1},g} \left( \theta_n - \gamma_{n+1} \nabla f(\theta_n) \right)$$

where

$$\text{Prox}_{\gamma,g}(\tau) = \text{argmin}_{\theta \in \Theta} \left( g(\theta) + \frac{1}{2\gamma} \|\theta - \tau\|^2 \right)$$

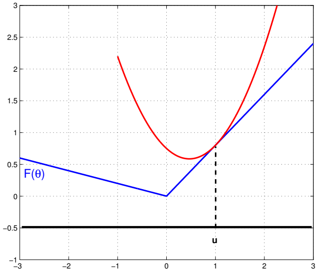
# Proximal Gradient: the intuition

Since $\nabla f$ is Lipschitz (with constant $L$), for any $\gamma \in (0, 1/L]$ and any $u \in \Theta$,

$$f(\theta) + g(\theta) \leq f(u) + g(\theta) + \langle \nabla f(u), \theta - u \rangle + \frac{1}{2\gamma} \|\theta - u\|^2$$

$$\leq C_u + g(\theta) + \frac{1}{2\gamma} \|\theta - (u - \gamma \nabla f(u))\|^2$$

The RHS is a majorizing function s.t.




- for $\theta = u$, it is equal to $(f + g)(u)$.
- for fixed $u$, it is convex (in $\theta$) and possesses an unique minimum.

The Proximal Gradient algorithm is a Majorization-Minimization procedure, satisfying

$$(f + g)(\theta_{n+1}) \leq (f + g)(\theta_n)$$

# The poster session

Proximal Gradient Algorithm $\{\tau_n\}_n$ converges to $\mathrm{argmin}(f+g)$

$$\tau_{n+1} = \mathrm{Prox}_{\gamma_{n+1},g}\left(\tau_n - \gamma_{n+1}\nabla f(\tau_n)\right)$$

In many applications, $\nabla f(\theta)$ unavailable. Hence:

Perturbed Proximal Gradient Algorithm

$$\theta_{n+1} = \mathrm{Prox}_{\gamma_{n+1},g}\left(\theta_n - \gamma_{n+1}H_{n+1}\right)$$

where $H_{n+1}$ is an approximation of $\nabla f(\theta_n)$.

1. Which conditions on the step-size sequence $\gamma_n$ and on the approximation $H_{n+1}$ for the convergence of this algorithm towards the minimizers of $f+g$ ?
2. When $\nabla f(\theta)$ is an integral and $H_{n+1}$ is a Monte Carlo approximation: how many samples when computing $H_{n+1}$ ?
3. The rate of convergence of the exact algorithm is known. Does the Stochastic Proximal Gradient reach the same rate ?

# Not on the poster, the sketch of the proof

- Step 1: for any minimizer $\theta_\star$ of $F$

$$\|\theta_{n+1} - \theta_\star\|^2 \leq \|\theta_n - \theta_\star\|^2 - \gamma_{n+1}\left(F(\theta_{n+1}) - \min F\right) + \gamma_{n+1}\mathsf{noise}_{n+1} \quad (1)$$

- Step 2: Use a (deterministic) Siegmund-Robbins lemma:
  If
  $$\sum_n \gamma_n = \infty, \qquad \sum_n \gamma_{n+1}\,\mathsf{noise}_{n+1} < \infty$$
  then the limiting points are minimizers of $F$.

- Step 3: Use again (1) to show the convergence of $\{\theta_n\}_n$ to a minimizer of $F$.