Snake: a Stochastic Proximal Gradient Algorithm for Regularized Problems over Large Graphs

Adil Salim[1], Pascal Bianchi[1], et Walid Hachem[2]

[1]LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France
[2]CNRS / LIGM (UMR 8049), Université Paris-Est Marne-la-Vallée, France

Résumé

A regularized optimization problem over a large unstructured graph is studied, where the regularization term is tied to the graph geometry. Typical examples include the total variation and Laplacian regularizers. In the special case where the graph is a simple path without loops, fast methods are often available. We introduce a novel algorithm, called Snake, which solves the problem over large and possibly unstructured graphs. Snake is a meta-algorithm which take benefits of the existence of fast 1D-methods. This algorithm is an instance of a new general stochastic proximal gradient algorithm, whose convergence is proven. Applications to trend filtering and graph inpainting are provided. Numerical experiments are conducted over large graphs.

Mots-clef : Stochastic optimization, Graphs.

Introduction

Many applications in the fields of machine learning, signal and image restoration, or trend filtering require the solution of the following optimization problem. On an undirected graph $G = (V, E)$ with no self loops, where $V = \{1, \dots, N\}$ represents a set of N nodes ($N \in \mathbb{N}^*$) and E is the set of edges, find

$$\min_{x \in \mathbb{R}^V} F(x) + R(x, \phi), \tag{1}$$

where F is a convex and differentiable function on $\mathbb{R}^V$ representing a data fitting term, and where the function $x \mapsto R(x, \phi)$ represents a regularization term of the form

$$R(x, \phi) = \sum_{\{i,j\} \in E} \phi_{\{i,j\}}(x(i), x(j)) \,,$$

where $\phi = (\phi_e)_{e \in E}$ is a family of convex and symmetric $\mathbb{R}^2 \to \mathbb{R}$ functions. When $\phi_e(x, x') = w_e|x - x'|$ where $w = (w_e)_{e \in E}$ is a vector of positive weights, the function $R(., \phi)$ coincides with the weighted total variation (TV) norm. Another example is the Laplacian regularization $\phi_e(x, x') = (x - x')^2$, or its normalized version obtained by rescaling x and x' by the degrees of each node in e respectively. The Forward-Backward (or proximal gradient) algorithm is one of the most popular approaches towards solving problems of the type of Problem (1). It reads

$$x_{n+1} = \mathrm{prox}_{\gamma R(.,\phi)}(x_n - \gamma \nabla F(x_n)) \,, \tag{2}$$

where $\gamma > 0$ is a fixed step, and where

$$\mathrm{prox}_g(y) = \arg\min_x \left(g(x) + \frac{1}{2}\|x - y\|^2 \right)$$

is the well-known proximity operator applied to the proper, lower semicontinuous (lsc), and convex function g (here $\|\cdot\|$ is the standard Euclidean norm). However, the computation of the proximity operator of $R(., \phi)$ at each iteration is a difficult task when the graph is large.

Our first idea is to consider the function $R(\cdot, \phi)$ as an expectation with respect to a random path. At the moment n, pick a node at random with a probability proportional to the degree of this node. Once this node has been chosen, pick another one at random uniformly among the neighbors of the first node. Repeat the process of choosing neighbors L times, where $L > 0$ is some fixed integer, and denote as $\xi_n \in V^{L+1}$ the path thus obtained at time n. Further, given a path $s = (v_0, v_1, \dots, v_L)$ where $\{v_i, v_{i+1}\} \in E$, write

$$R(x, \phi_s) = \sum_{i=1}^{L} \phi_{\{v_{i-1}, v_i\}}(x(v_{i-1}), x(v_i)) \,.$$

With this at hand, we get with some elementary Markov chain formalism that $R(x, \phi) = (|E|/L)\mathbb{E}R(x, \phi_{\xi_n})$. Thus, Problem 1 can be seen as a problem of minimizing an expectation. We can therefore think of solving

this problem by implementing a *stochastic* version of Algorithm (2) : by generating an independent sequence (ξ_n) of paths chosen as described above, our algorithm would read

$$x_{n+1} = \mathrm{prox}_{\gamma_{n+1}|E|R(\cdot,\phi_{\xi_{n+1}})}(x_n - \gamma_{n+1} L \nabla F(x_n))\,,$$

where the step γ is now replaced with a sequence (γ_n) of decreasing steps in order to alleviate the effect of the randomness incurred by the independent sequence (ξ_n). The above algorithm is still difficult to implement. Indeed, the efficient implementations of the proximity operator over a path often require this path to be *simple*, that is, to bear no repeated node. Among such fast algorithms, let us cite the so-called *taut-string* algorithm [Con13] for total variation minimization, or the fast 1D Laplacian solvers [Spi10]. Thus, we propose to split the path ξ_n into a finite sequence of successive simple paths $\xi_n = (\xi_n^1, \xi_n^2, \ldots)$, and to apply the stochastic proximal gradient algorithm successively on these simple paths, terming this new algorithm as "Snake". The algorithm is a special case of a more general novel stochastic proximal gradient algorithm which is introduced in Section 1.

1 General Stochastic Algorithm

We denote by $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space and by $\mathbb{E}$ the corresponding expectation. We let $(\Xi, \mathcal{X})$ be an arbitrary measurable space. We denote X some Euclidean space. We consider the general problem :

$$\min_{x \in \mathsf{X}} \sum_{i=1}^{L} \mathbb{E}\left(f_i(x, \xi^i) + g_i(x, \xi^i)\right) \tag{3}$$

where L is a positive integer, for all $i = 1, \ldots, L$, $\xi^i : \Omega \to \Xi$ is a random variable (r.v.), for every $i = 1, \ldots, L$, $f_i : \mathsf{X} \times \Xi \to \mathbb{R}$ and $g_i : \mathsf{X} \times \Xi \to \mathbb{R}$ satisfy :

Assumption 1. *For all $i \in \{1, \ldots, L\}$:*

1. *The f_i and g_i are normal convex integrands [Roc69] s.t. for every $x \in \mathsf{X}$, $\mathbb{E}(|f_i(x, \xi^i)|) < \infty$ and $\mathbb{E}(|g_i(x, \xi^i)|) < \infty$.*
2. *For every $s \in \Xi$, $f_i(\cdot, s)$ is differentiable. There exists a measurable map $K_i : \Xi \to \mathbb{R}_+$ s.t. $\mathbb{E}(K_i(\xi_i)^\alpha) < \infty$ for all $\alpha \geq 1$, and s.t. the following holds $\mathbb{P}$-a.e. : for all x, y in X,*

$$\|\nabla f_i(x, \xi_i) - \nabla f_i(y, \xi_i)\| \leq K_i(\xi_i)\|x - y\|\,.$$

For every $i = 1, \ldots, L$ and every $\gamma > 0$, we introduce the mapping $\mathsf{T}_{\gamma,i} : \mathsf{X} \times \Xi \to \mathsf{X}$ defined by

$$\mathsf{T}_{\gamma,i}(x, s) = \mathrm{prox}_{\gamma g_i(\cdot,s)}(x - \gamma \nabla f_i(x, s))\,.$$

We define $\mathsf{T}_\gamma : \mathsf{X} \times \Xi^L \to \mathsf{X}$ by

$$\mathsf{T}_\gamma(\cdot, (s^1, \ldots, s^L)) = \mathsf{T}_{\gamma,L}(\cdot, s^L) \circ \cdots \circ \mathsf{T}_{\gamma,1}(\cdot, s^1)\,.$$

Let ξ be the random vector $\xi = (\xi^1, \ldots, \xi^L)$ with values in Ξ^L and let $(\xi_n : n \in \mathbb{N}^*)$ be a sequence of i.i.d. copies of ξ, defined on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For all $n \in \mathbb{N}^*$, $\xi_n = (\xi_n^1, \ldots, \xi_n^L)$. Finally, let (γ_n) be a positive sequence. Our aim is to analyze the convergence of the iterates (x_n) recursively defined by :

$$x_{n+1} = \mathsf{T}_{\gamma_{n+1}}(x_n, \xi_{n+1})\,, \tag{4}$$

as well as the intermediate variables $\bar{x}_{n+1}^i$ $(i = 0 \ldots L)$ defined by $\bar{x}_{n+1}^0 = x_n$, and $\bar{x}_{n+1}^i = \mathsf{T}_{\gamma_{n+1},i}(\bar{x}_{n+1}^{i-1}, \xi_{n+1}^i)$, $(i = 1 \ldots L)$. Let $\mathcal{Z}$ be the set of minimizers of Problem (3). By our assumptions, a point $x_\star$ belongs to $\mathcal{Z}$ *iff* $0 \in \sum_{i=1}^L \nabla \mathbb{E}(f_i(x_\star, \xi^i)) + \partial \mathbb{E}(g_i(x_\star, \xi^i))$. By [RW82], this means that there exists L integrable mappings $\varphi_1, \ldots, \varphi_L$ s.t $\varphi_i(\xi^i) \in \partial g_i(x_\star, \xi^i)$ -a.e. for all i and s.t.

$$0 = \sum_{i=1}^{L} \mathbb{E}(\nabla f_i(x_\star, \xi^i)) + \mathbb{E}(\varphi_i(\xi^i))\,. \tag{5}$$

When (5) holds, we say that the family $(\nabla f_i(x_\star, \xi^i), \varphi_i(\xi^i))_{i=1\ldots L}$ is a *representation* of the minimizer $x_\star$. In addition, if for some $\alpha \geq 1$ and every $i = 1 \ldots L$, $\mathbb{E}(\|\nabla f_i(x_\star, \xi^i)\|^\alpha) < \infty$ and $\mathbb{E}(\|\varphi(\xi^i)\|^\alpha) < \infty$, we say that the minimizer $x_\star$ admits a α-integrable representation. Finally, let $\partial g_i^0(x, \xi^i)$ be the least norm element in $\partial g_i(x, \xi^i)$.

Assumption 2. *The set $\mathcal{Z}$ is non empty. For every $x_\star \in \mathcal{Z}$, there exists $\varepsilon > 0$ s.t. $x_\star$ admits a $(2+\varepsilon)$-integrable representation $(\nabla f_i(x_\star, \xi^i), \varphi_i(\xi^i))_{i=1\ldots L}$.*

Assumption 3. *For every compact set $\mathcal{K} \subset \mathsf{X}$, there exists $\eta > 0$ such that for all $i = 1 \ldots L$, $\sup_{x \in \mathcal{K}} \mathbb{E}(\|\partial g_i^0(x, \xi^i)\|^{1+\eta}) < \infty$.*

Theorem 1. *Let Assumptions 1–3 hold true. Assume that $\sum \gamma_n = +\infty$, $\sum \gamma_n^2 < \infty$ and $\frac{\gamma_{n+1}}{\gamma_n} \to 1$. There exists a r.v. $X_\star$ s.t. $\mathbb{P}(X_\star \in \mathcal{Z}) = 1$ and s.t. (x_n) converges a.s. to $X_\star$ as $n \to \infty$. Moreover, for every $i = 0 \ldots L-1$, $\bar{x}_n^i$ converges a.s. to $X_\star$.*

The proof follows [BH16] and is omitted.

2 The Snake Algorithm

Let $\ell \geq 1$ be an integer. We refer to a walk of length ℓ over the graph G as a sequence $s = (v_0, v_1, \ldots, v_\ell)$ in $V^{\ell+1}$ such that for every $i = 1, \ldots, \ell$, the pair $\{v_{i-1}, v_i\}$ is an edge of the graph. A walk of length zero is a

single vertex. Let $L \geq 1$. We denote by Ξ the set of all walks over G with length $\leq L$. This is a finite set. Let $\mathcal{X}$ be the set of all subsets of Ξ. We consider the mesurable space $(\Xi, \mathcal{X})$. Let $s = (v_0, v_1, \dots, v_\ell) \in \Xi$ with $0 < \ell \leq L$. We abusively denote by ϕ_s the family of functions $(\phi_{\{v_{i-1}, v_i\}})_{i=1,\dots,\ell}$. We say that a walk is a *simple path* if there is no repeated node that is, all elements in s are different or if s is a single vertex. We assume that when s is a simple path, the computation of $\mathrm{prox}_{R(.,\phi_s)}$ can be done easily.

Formulation of (1) as a stochastic program

Denote by $\deg(v)$ the degree of the node $v \in V$, *i.e.*, the number of neighbors of v in G. Let π be the probability measure on V defined as $\pi(v) = \frac{\deg(v)}{2|E|}$ for all $v \in V$. Define the probability transition kernel P on V^2 as $P(v,w) = \mathbb{1}_{\{v,w\}\in E}/\deg(v)$ if $\deg(v) > 0$, and $P(v,w) = \mathbb{1}_{v=w}$ otherwise, where $\mathbb{1}$ is the indicator function. We refer to a Markov chain (indexed by $\mathbb{N}$) over V with initial distribution π and transition kernel P as an infinite random walk over G. Let $(v_k)_{k\in\mathbb{N}}$ be a infinite random walk over G defined on the canonical probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\Omega = V^{\mathbb{N}}$. Setting an integer $L \geq 1$, we define the random variable ξ from $(v_k)_{k\in\mathbb{N}}$ as $\xi = (v_0, v_1, \dots, v_L)$. It can be shown using elementary Markov chain arguments that $R(x,\phi) = \frac{|E|}{L}\,\mathbb{E}(R(x,\phi_\xi))$. Therefore, Problem (1) is written equivalently

$$\min_{x\in\mathbb{R}^V} LF(x) + |E|\,\mathbb{E}(R(x,\phi_\xi))\,. \tag{6}$$

We now split the random walk ξ into several simple paths. We recursively define a sequence of stopping time $(\tau_i)_{i\in\mathbb{N}}$ as $\tau_0 = 1$ and for all $i \geq 0$,

$$\tau_{i+1} = \min\{k \geq \tau_i :\, v_k \in \{v_{\tau_i - 1}, \cdots, v_{k-1}\}\}$$

if the above set is nonempty, and $\tau_{i+1} = +\infty$ otherwise. We now define the stopping times t_i for all $i \in \mathbb{N}$ as $t_i = \min(\tau_i, L+1)$. Finally, for all $i \in \mathbb{N}^*$ we can consider the random variable ξ^i on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $(\Xi, \mathcal{X})$ defined by

$$\xi^i = (v_{t_{i-1}-1}, v_{t_{i-1}}, \dots, v_{t_i - 1}).$$

We denote by N the smallest integer n such that $t_n = L+1$. We denote by $\ell(\xi^i)$ the length of the walk ξ^i. For every $i = 1 \dots L$, define the functions f_i, g_i on $\mathbb{R}^V \times \Xi$ in such a way that

$$f_i(x, \xi^i) = \ell(\xi^i) F(x) \tag{7}$$

$$g_i(x, \xi^i) = |E|\, R(x, \phi_{\xi^i})\,. \tag{8}$$

Note that when $i > N(\omega)$ then $f_i(x, \xi^i(\omega)) = g_i(x, \xi^i(\omega)) = 0$. It si straightforward to see that $LF(x) = \sum_{i=1}^L \mathbb{E}(f_i(x, \xi^i)$ and $R(x, \phi_\xi) = \sum_{i=1}^N R(x, \phi_{\xi^i}) = |E|^{-1} \sum_{i=1}^L g_i(x, \xi^i)$. In view of (6), **Problem (1) is equivalent to Problem (3) for the function f_i and g_i defined above**. We apply the general algorithm of Section 1 to this special case, and refer to *Snake* as the corresponding algorithm.

Main Algorithm

The corresponding iterations (4) read as $x_{n+1} = \mathsf{T}_{\gamma_{n+1}}(x_n, \xi_{n+1})$ where (ξ_n) are iid copies of ξ. For every $i = 1 \dots L-1$, the intermediate variable $\bar{x}^i_{n+1}$ satisfies $\bar{x}^i_{n+1} = \mathrm{prox}_{\gamma_n g_i(\,.\,,\xi^i_{n+1})}(\bar{x}^{i-1}_n - \gamma_n \nabla f_i(\bar{x}^{i-1}_n, \xi^i_{n+1}))\,.$

Theorem 2. *Let the step sizes (γ_n) be chosen as in Theorem 1. Assume that Problem (1) admits a minimizer. Assume that the convex function F is differentiable and that ∇F is Lipschitz continuous. Then, there exists a r.v. $X_\star$ s.t. $X_\star(\omega)$ is a minimizer of (1) for all ω $\mathbb{P}$-a.e., and s.t. the sequence (x_n) defined above converges a.s. to $X_\star$ as $n \to \infty$. Moreover, for every $i = 0 \dots L-1$, $\bar{x}^i_n$ converges a.s. to $X_\star$.*

The proof of Theorem 2 consists in verifying the assumptions of Theorem 1. The pseudocode is as follows :

---

procedure SNAKE(x_0, L)
  $z \leftarrow x_0, \quad n \leftarrow 0, \quad \ell \leftarrow L$
  $e \leftarrow$RND_ORIENTED_EDGE
  while stopping criterion is not met **do**
    $c, e \leftarrow$ SIMPLE_PATH(e, ℓ)
    $z \leftarrow$ PROX($z - \gamma_n$LGTH(c)$\nabla F(z), c, |E|\gamma_n$)
    $\ell \leftarrow \ell -$ LGTH(c)
    if $\ell = 0$ **then**
      $e \leftarrow$ RND_ORIENTED_EDGE
      $\ell \leftarrow L$
      $n \leftarrow n+1$
    end if
  end while
  return z
end procedure

The above pseudocode calls the following subroutines. If c is a finite walk, $c[-1]$ is the last element of c and LGTH(c) is its length as a walk that is $|c| - 1$. The procedure RND_ORIENTED_EDGE returns a tuple of two nodes randomly chosen (v, w) where $v \sim \pi$ and $w \sim P(v, .)$. For every $x \in \mathbb{R}^V$, every simple path s and every $\alpha > 0$, the procedure PROX(x, s, α) returns the quantity $\mathrm{prox}_{\alpha R(.,\phi_s)}(x)$. The procedure SIMPLE_PATH(e, ℓ) ge-

nerates the first steps of a random walk on G with transition kernel P initialized at the vertex $e[-1]$, and prefaced by the first node in e. It is described below. It uses the procedure UNIFORM_NEIB(v) which returns a random vertex drawn uniformly amongst the neighbors of v.

---

procedure SIMPLE_PATH(e, ℓ)
  $c \leftarrow e$
  $w \leftarrow$ UNIFORM_NEIB($e[-1]$)
  while $w \notin c$ and LGTH(c)$< \ell$ **do**
    $c \leftarrow [c, w]$
    $w \leftarrow$ UNIFORM_NEIB(w)
  end while
  return $c, [c[-1], w]$
end procedure

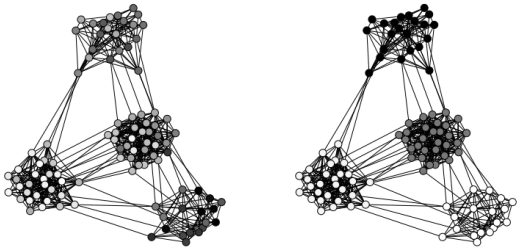
---

3 Trend Filtering on Graphs

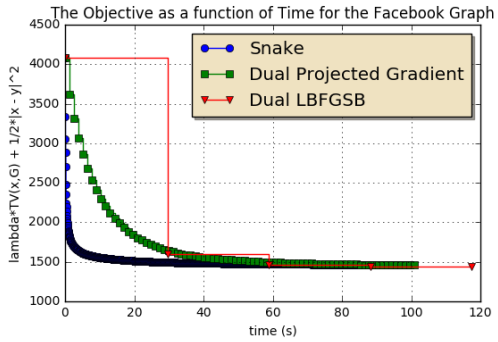
Consider a vector $y \in \mathbb{R}^V$. The Graph Trend Filtering (GTF) estimate on V, is defined in [WSST16] by

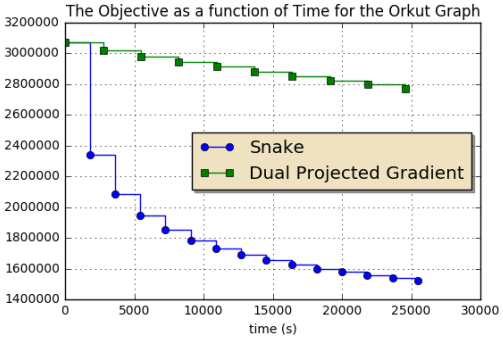
$$\hat{y} = \arg\min_{x \in \mathbb{R}^V} \frac{1}{2}\|x - y\|^2 + \lambda \sum_{\{i,j\} \in E} |x(i) - x(j)|. \quad (9)$$

where $\lambda > 0$. We plot below an example of noisy data y plotted on the vertices of a general graph (left) and the GTF estimate $\hat{y}$ obtained from y (right).



We solve Problem (9) with $F(x) = \frac{1}{2}\|x - y\|^2$ using Snake, and compare with the projected gradient method in the dual, and L-BFGS-B in the dual. In Snake, the 1D-proximity is computed using [Con13]. We consider the Facebook graph which is a network of 4039 nodes and 88234 edges extracted from the Facebook social network and the Orkut graph with 3072441 nodes and 117185083 edges (for the latter, L-BFGS-B generated a memory error). The vector y is sampled according to a standardized Gaussian distribution of dimension $|V|$ and λ is set such that $\mathbb{E}(\frac{1}{2}\|x - y\|^2) = \mathbb{E}(\lambda \sum_{\{i,j\} \in E} |x(i) - x(j)|)$ if x, y are two independent r.v with standardized Gaussian distribution. Initialization is set to y, $\gamma_n = 1/10|E|n$ and $L = |V|$. The following Figures show the objective function as a function of time for each algorithm. We obtain speed-ups over L-BFGS-B and the projected gradient algorithms for the dual problem especially in the first iterations.





Références

[BH16] Pascal Bianchi and Walid Hachem. Dynamical behavior of a stochastic forward–backward algorithm using random monotone operators. *Journal of Optimization Theory and Applications*, 171(1) :90–120, 2016.

[Con13] L. Condat. A direct algorithm for 1d total variation denoising. *IEEE SPL*, 20(11) :1054–1057, 2013.

[Roc69] R Tyrrell Rockafellar. Measurable dependence of convex sets and functions on parameters. *Journal of Mathematical Analysis and Applications*, 28(1) :4–25, 1969.

[RW82] Ralph T Rockafellar and Roger JB Wets. On the interchange of subdifferentiation and conditional expectation for convex functionals. *Stochastics : An International Journal of Probability and Stochastic Processes*, 7(3) :173–182, 1982.

[Spi10] Daniel A Spielman. Algorithms, graph theory, and linear equations in laplacian matrices. In *Proceedings of the ICM*, volume 4, pages 2698–2722, 2010.

[WSST16] Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105) :1–41, 2016.