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

Adil Salim, Pascal Bianchi and Walid Hachem
LTCI, Telecom ParisTech, Universite Paris-Saclay, 75013, Paris, France.
CNRS / LIGM (UMR 8049), Universite Paris-Est Marne-la-Vallee.
adil.salim,pascal.bianchi@telecom-paristech.fr, walid.hachem@u-pem.fr

Many applications in the fields of multiagent systems, distributed optimization, machine learning on graphs [1], graph theory [2] or multi-task learning, require the solution of the following optimization problem. On an undirected graph $G = (V(G), E(G))$, where $V(G) = \{1, \ldots, N\}$ represents a set of N nodes and $E(G)$ is the set of edges, find

$$\min_{x \in \mathsf{X}^{V(G)}} \sum_{i \in V(G)} f_i(x(i)) + \sum_{\{i,j\} \in E(G)} \phi_{i,j}(x(i), x(j)), \tag{1}$$

where X is an Euclidean space, f_i is a convex differentiable function for all $i \in V(G)$, and ϕ_e is a convex symmetric function for all $e \in E(G)$.

Define the data fitting term $F(x) = \sum_{i \in V(G)} f_i(x(i))$ and the regularization term $R(x, G) = \sum_{\{i,j\} \in E(G)} \phi_{i,j}(x(i), x(j))$. When $R(\cdot, G)$ is the (weighted) Total Variation (TV) norm, $R(\cdot, G) = \sum_{\{i,j\} \in E(G)} |x(i) - x(j)|$, instances of (1) include the Graph Trend Filtering (GTF) context of [1]. In this context, F is set to $F(x) = \frac{1}{2}\|x - y\|^2$ where y is a fixed vector. When $R(\cdot, G)$ is the (weighted and/or normalized) Laplacian regularization, for example $R(\cdot, G) = \sum_{\{i,j\} \in E(G)} |x(i) - x(j)|^2$, instances of Problem (1) include the resolution of linear equations in Laplacian matrix or the resolution of semi-supervised learning problems over graphs [2].

The proximal gradient algorithm is one of the most popular approaches towards solving the regularized Problem (1). This algorithm produces the sequence of iterates

$$x_{n+1} = \operatorname{prox}_{\gamma R(\cdot, G)}(x_n - \gamma \nabla F(x_n)), \tag{2}$$

where $\gamma > 0$ is a fixed step, and where $\operatorname{prox}_{\gamma R(\cdot,G)}(y) = \arg\min_x \left(R(x, G) + \frac{1}{2\gamma}\|x - y\|^2 \right)$ is the well-known proximity operator. Implementing the proximal gradient algorithm requires the computation of $\operatorname{prox}_{\gamma R(\cdot,G)}$. When N is large, the computation of the proximity operator is in general not affordable due to the non separability of the regularization term. However, when G is one-dimensional (1D) (see Figure 1, left) and $R(\cdot, G)$ is the TV norm, the *taut-string* algorithm is an efficient procedure to compute the proximity operator over an 1D-graph [3]. Similar observations can be made for the Laplacian regularization, where, *e.g.*, the discrete cosine transform can be implemented over an 1D-graph. Over large and general graphs, the computation of $\operatorname{prox}_{\gamma R(\cdot,G)}$ is more difficult ([1, 2]).

In this work, an online method called Snake is proposed to solve (1) over a general graph G. It consists in properly selecting random simple paths (*i.e* 1D-subgraphs of G) in the graph and performing the proximal gradient algorithm over these simple paths (see Figure 1, right).

Consider a stationary simple random walk ξ is over G with length $L + 1$. The walk ξ induced a subgraph $\xi = (V(\xi), E(\xi))$ of G (Figure 1). It is proven in [4] that, for every $L \geq 2$, Problem (1) is equivalent to

$$\min_{x \in \mathsf{X}^{V(G)}} \mathbb{E}_\xi \left(\frac{1}{L+1} \sum_{i \in V(\xi)} \frac{1}{\deg(i)} f_i(x(i)) \right) + \mathbb{E}_\xi \left(\frac{1}{L} R(x, \xi) \right). \tag{3}$$

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

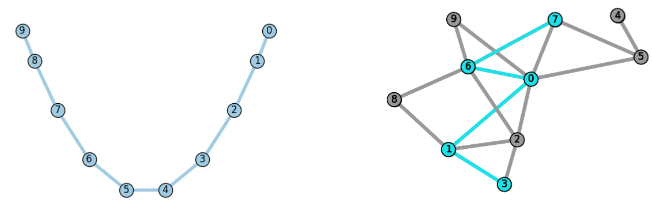

Figure 1: Left: 1D graph. Right: General graph on which is colored the simple path 3-1-0-6-7.

It is then shown, based on recent advances in the study of stochastic proximal algorithm that the Snake algorithm (4)

$$x_{n+1} = \text{prox}_{\frac{\gamma_n}{L} R(\cdot,\xi_{n+1})} \left(x_n - \frac{\gamma_n}{L+1} \sum_{i \in V(\xi_{n+1})} \frac{1}{\deg(i)} \nabla f_i(x_n(i)) \right) \tag{4}$$

converges to a solution of Problem (1). In this algorithm, (γ_n) is a decreasing step size and the ξ_n are copies of random walks stopped when a node is repeated. The Snake algorithm only involves computations of the proximity operators over 1D graphs which can be done efficiently, and only need a local knowledge of the graph topology. Moreover, the parameter L controls the iteration complexity.

Finally, applications of Snake in the GTF context and for solving linear equations in Laplacian matrix over the Facebook and Orkut graphs ([5]) are provided in Figure 2.

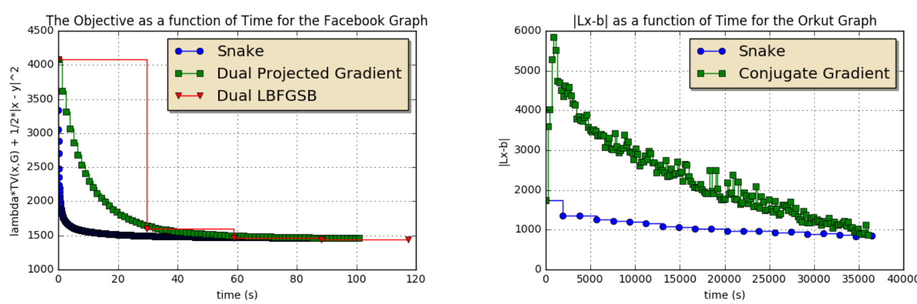

Figure 2: Left: GTF estimation over the Facebook graph. Right: Solving $Lx = b$, where L is the Laplacian matrix of the Orkut graph.

As stochastic gradient algorithm has been implemented to handle large scale data fitting terms, Snake uses a stochastic proximal method to regularize the graph online. This work leaves the door open to the use of stochastic proximal methods to handle large scale regularizations.

References

[1] 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.

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

[3] Alvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. *arXiv preprint arXiv:1411.0589*, 2014.

[4] Adil Salim, Pascal Bianchi, and Walid Hachem. Snake: a stochastic proximal gradient algorithm for regularized problems over large graphs. In preparation, 2017.

[5] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.