Random Pairwise Gossip on Hadamard Manifolds

Anass Bellachehab
Institut Mines-Télécom
Télécom SudParis
CNRS UMR 5157 SAMOVAR
9 rue Charles Fourier, 91000 Evry
France

Jérémie Jakubowicz
Institut Mines-Télécom
Télécom SudParis
CNRS UMR 5157 SAMOVAR
9 rue Charles Fourier, 91000 Evry
France

***Abstract*—In the context of sensor networks, the *consensus* problem is an important problem. The goal is to find a *distributed algorithm* to reach a consensus, *i.e.* some common value shared by all the agents in the network. Such a well known algorithm is the so-called *Random Pairwise Gossip* (RPG) [BGPS06]. This algorithm relies on pairwise *arithmetic averages*. Therefore, it does not apply to some situations of interest where computing arithmetic averages is meaningless. For instance, consensus on axis orientation, or subspace tracking, or camera position, cannot be addressed by RPG. However, all these cases exhibit a common underlying structure: the data belong to some *Riemannian manifold* [DC92]. The goal of this paper is to adapt the RPG algorithm to the Riemannian manifolds framework. Replacing arithmetic average by midpoints for the metric is a natural idea, promoted in this paper. However, due to curvature, and contrarily to the Euclidean case, convergence is no more guaranteed. However, we show that under suitable curvature assumptions – namely, nonpositive curvature – convergence can still be ensured. Numerical experiments validate our approach.**

***Keywords*—consensus, pairwise gossip, Hadamard manifolds**

I. Introduction

Consensus problems are ubiquitous in distributed environments: they appear in database management [Bur06], clock synchronization [SG07], signal estimation in wireless sensor networks [SG08], to cite a few. Consider, for instance, an ad-hoc network of low power sensors, where each sensor has only access to local information of its surrounding environment. Due to hardware and energy constraints, long range communication between sensors is proscribed, and only short distance communications are reliable enough for consideration. The goal is to devise a protocol to achieve consensus among the sensors, without resorting to any central fusion node. If the measurements belong to some vector space, *e.g.* temperatures, speeds, or locations; *Gossip* protocols [BGPS06] are efficient candidates.

However, there are several interesting cases where measurements cannot be added or scaled as vectors. Camera orientations are such an example: it does not make sense to add two orientations. There are several other examples of interest: subspaces, curves, angles have no underlying vector space structure. All these cases are properly addressed by the framework of Riemannian manifolds [DC92]. Sometimes the situation is more subtle: there is a vector space structure, but it is not well adapted to the problem considered. Information geometry [ABNK$^+$87] is such an example, where it has been shown that using the Riemannian structure to compute gradients could give better results than using Euclidean gradients.

Prior work. Consensus on manifolds has already been the subject of several recent works [SSS07], [SS09], [SBS10], [Bon13], [TAV11]. Existing approaches use the following viewpoint: distances between measurement make sense in the Riemmanian setting as much as in the Euclidean setting. Consensus is interpreted as a global minimum of $\Delta = \sum_{v\sim w} d(x_v, x_w)^2$ where the sum runs over all connected agents (notations are clarified below). Hence, tools from optimization on manifolds can be used to address the consensus problem. In [Bon13], [TAV11], a gradient descent is used. In [TAV11] the network is implicitly assumed synchronous, *i.e.* able to perform computations when some common clock ticks. While in [Bon13], the network is assumed asynchronous – which inevitably brings noise – and for convergence to hold, the stepsize goes to 0 with time (in a precise way).

Paper contribution. In this paper we propose an intrinsic and very simple algorithm to achieve consensus in Riemannian manifolds. It is a very natural adaption of the Random Pairwise Gossip (RPG) algorithm [BGPS06] and does not use tools from optimization. Like the RPG, it works in asynchronous networks because like the RPG only two neighbors wake up at a given time. Yet, in our case, it is not necessary to decrease the stepsize. For general manifolds, convergence is hopeless. However, we prove that in the case of nonpositive sectional curvature [DC92] manifolds, referred to as *Hadamard manifolds* in the literature, convergence can be guaranteed. This curvature assumption would exclude the camera orientations manifold which is of positive curvature. But it encompasses several members of the exponential family in information geometry as well as the space of positive definite matrices. Moreover, we prove, and it appears in the numerical experiments performed and reported in this paper, that the proposed algorithm, besides its simplicity, is also efficient. However, unlike the RPG; the proposed algorithm does not necessarily converge towards center of mass. It only converges to an arbitrary consensus state.

The rest of the paper is organized as follows. Section II describes the assumptions made on the network and the data. Section III details the proposed algorithm and state a convergence result along with a convergence speed result. Numerical experiments are provided in section IV and section V concludes the paper.

II. Framework

A. Network

We consider a network of N agents represented by an undirected graph $G = (V, E)$, where $V = \{1, \dots, N\}$ stands for the set of agents and E denotes the set of available communication links between agents. A link $e \in E$ is given by a pair $\{v, w\}$ where v and w are two distinct agents in the network that are able to communicate directly. Note that the graph is assumed undirected, meaning that whenever agent v is able to communicate with agent w, the reciprocal communication is also assumed feasible. This assumption makes sense when communication speed is fast compared to agents movements speed. When a communication link $e = \{v, w\}$ exists between two agents, both agents are said to be neighbors and the link is denoted $v \sim w$. We denote by $\mathcal{N}(v)$ the set of all neighbors of the agent $v \in V$. The communication framework considered here is standard [BGPS06].

The graph is assumed to be connected, which means that for every two agents u, v there exists a finite sequence of agents $w_0 = u, \dots, w_d = v$ such that:

$$\forall 0 \leq i \leq d-1 : \{w_i, w_{i+1}\} \in E$$

This means that each two agents are at least indirectly related.

B. Time

As in [BGPS06], we assume that the time model is asynchronous, *i.e.* that each agent has its own Poisson clock that ticks with a common intensity λ (the clocks are identically made), and moreover, each clock is independent from the other clocks. When an agent clock ticks, the agent is able to perform some computations and wake up some neighboring agents. This time model has the same probability distribution than a global single clock ticking with intensity $N\lambda$ and selecting uniformly randomly a single agent at each tick. This equivalence is described, *e.g.* in [BGPS06]. Notice also that link $e = \{v, w\}$ is not necessarily used by agents v and w at a given time.

C. Data

Each node $v \in V$ stores data represented as an element x_v of a *Riemannian manifold* $\mathcal{M}$ [DC92]. Initially each node v has an initial value $x_v(0)$ and $X_0 = (x_1(0), \dots, x_N(0))$ is the tuple of initial values of the network. We would like to find an interactive algorithm that, from an initial state $x(0)$ takes the system to a *consensus state*; meaning a state of the form $X_\infty = (x_\infty, \dots, x_\infty)$ with: $x_\infty \in \mathcal{M}$. We denote by $x_v(k)$ the value stored by the agent $v \in V$ at the k-th iteration of the algorithm, and $X_k = (x_1(k), \dots, x_N(k))$ the global state of the system at instant k.

D. Manifolds

Let $\mathcal{M}$ be a connected n dimensional Riemannian manifold. For $p \in \mathcal{M}$ we denote by $T_p\mathcal{M}$ the *tangent space* to $\mathcal{M}$ at p [DC92], the tangent space is a vector space with the same dimension as $\mathcal{M}$. A *metric* is a smooth collection of dot products $\langle , \rangle_p$ on $T_p\mathcal{M}$ for $p \in \mathcal{M}$. A smooth curve in $\mathcal{M}$ is a smooth function $c : [0,1] \to \mathcal{M}$, the length of the curve is define as the quantity: $L(c) := \int_0^1 \langle c'(t), c'(t) \rangle_{c(t)} \mathrm{d}t$. Using this definition, the notion of distance between two points x and $y \in \mathcal{M}$ is defined by: $d(x, y) := \inf_{\{c | c(0)=x; c(1)=y\}} L(c)$.

A curve $\gamma(t)$ between x and y verifying: $L(\gamma) = d(x, y)$ is called a *geodesic*. If such a curve exists and is unique, we denote it by $[x, y]$. This notation suggests that the notion of geodesic is the Riemannian analog to that of line segment. The manifold $\mathcal{M}$ is said *complete* if for every couple $(x, y) \in \mathcal{M}^2$ a geodesic between x and y exists.

We also define *the midpoint* of x and y (assuming existence and uniqueness of the geodesic):

$$\left\langle \frac{x+y}{2} \right\rangle := [x, y]\left(\frac{1}{2}\right)$$

It is important to notice that notation $\langle \frac{x+y}{2} \rangle$ denotes the midpoint of $[x, y]$ in the manifold and involves actually no addition nor dilation. Obviously in the Euclidean case the two notions ("midpoint" and "arithmetic mean") coincide.

For $p \in \mathcal{M}$ and σ a 2-dimensional subspace of $T_p\mathcal{M}$ we denote by $K_p(\sigma)$ the *sectional curvature* of σ [DC92]. A manifold is said to be a *Hadamard manifold* if and only if:

$$\forall p \in \mathcal{M}, \sigma \subseteq T_p\mathcal{M} : K_p(\sigma) \leq 0$$

In the following, we assume $\mathcal{M}$ to be a Hadamard manifold. It can be proven that a Hadamard manifold is also a complete manifold, and for all $(x, y) \in \mathcal{M}^2$ the geodesic between x and y is unique.

With these assumptions in mind (connected graph, Poisson clocks, Hadamard manifold), we propose the following consensus algorithm:

III. Algorithm

A. Description

At each count of the virtual global clock one node v is selected uniformly randomly from the set of agents V. The node v then randomly selects a node w from $\mathcal{N}(v)$. Both node v and w then compute and update their value to $\langle \frac{x_v + x_w}{2} \rangle$.

Algorithm Random Pairwise Midpoint

Input: a graph $G = (V, E)$ and the initial nodes configuration $x_v(0), v \in V$

for all $k > 0$ **do**

At instant k, uniformly randomly choose a node v_k from V and a node w_k uniformly randomly from $\mathcal{N}(v_k)$.

Update:

$x_{v_k}(k) = \left\langle \frac{x_{v_k}(k-1) + x_{w_k}(k-1)}{2} \right\rangle$

$x_{w_k}(k) = \left\langle \frac{x_{v_k}(k-1) + x_{w_k}(k-1)}{2} \right\rangle$

$x_v(k) = x_v(k-1)$ for $v \notin \{v_k, w_k\}$

end for

B. Convergence result

Under the assumptions of section II, *i.e.*, homogeneous Poisson clocks, connected and nonpositive sectional curvature for the manifold, we have the following result that is given without proof due to space constraints:

Theorem 1 (Almost sure convergence)**.** *Let $X_k = (x_1(k), ..., x_N(k))$ denote the sequence generated by Algorithm Random Pairwise Midpoint, there exists $x_\infty \in \mathcal{M}$ such that almost surely, $\lim_{k\to\infty} X_k = X_\infty := (x_\infty, \ldots, x_\infty)$*

Let us discuss the nonpositivity assumption on manifold curvature. Indeed, even if the previous algorithm mimics the RPG algorithm, its behavior can be actually quite different in the case of positive curvature. Consider for instance the case where $\mathcal{M} = \{(x, y, z) : x^2 + y^2 + z^2 = 1\}$ is the sphere of dimension 2. It is a Riemmanian manifold as a submanifold of the Euclidean space of dimension 3 with constant positive sectional curvature (all curvature notions are the same in dimension 2). For the network, consider the complete graph K_4 with 4 elements ($V = \{1, 2, 3, 4\}$ and E contains all possible edges).

Assume the initial data is $x_1(0) = x_2(0) = (1, 0, 0)$ and $x_3(0) = x_4(0) = (-1, 0, 0)$.The configuration will remain stuck in this position except if agents 1 or 2 communicates with 3 or 4 in which case the midpoint is not uniquely defined since $(1, 0, 0)$ and $(-1, 0, 0)$ are antipodal points. Even if one tries to perturb the data to get a well defined midpoint, the configuration can go near another pair of antipodal points.

We now define the *disagreement function*.

Definition 1. *Given a configuration $x = (x_1, \ldots, x_N) \in \mathcal{M}^N$ the disagreement function $\Delta(x) = \sum_{v\sim w} d(x_v, x_w)^2$.*

Function Δ measures how much disagreement is left in the network. Indeed, when the network is connected, Δ is 0 if and only if the network is at consensus; and the following result shows that, still under the assumptions of section II, Δ goes to 0 exponentially fast.

Theorem 2 (Convergence speed)**.** *Let $X_k = (x_1(k), ..., x_N(k))$ denote the sequence of random variables generated by Algorithm Random Pairwise Midpoint, there exists $L < 0$ such that,*

$$\limsup_{k\to\infty} \frac{\log \mathbb{E}\Delta(X_k)}{k} \le L$$

The proof shows that one can actually choose constant L independently from the manifold $\mathcal{M}$ and initial configuration x_0: it depends only on the graph structure. However, a lower bound for the $\limsup$ would typically depend on $\mathcal{M}$ and X_0.

The proofs will be provided in an extended version of this paper. We only give here the underlying idea.

Sketch of proof. Both theorems are proved using a comparison argument for triangles in Euclidean and Hadamard geometry. We show that function Δ contracts more when curvature is negative than in the Euclidean case where curvature is zero and both results are well known.

IV. Numerical Simulations

In this section we simulate Algorithm Random Pairwise Midpoint using two examples of Hadamard manifolds. First, we focus on the case of *statistical manifolds* [AN00]. Then, we study an example using positive definite matrices as data.

A. Statistical manifolds

The typical scenario here is the following: sensors fit some parametric model in the exponential family; then, they exchange with the rest of the network and try to find a consensus on the model parameters. Thus, in this scenario, the network is only used after all sensors have processed their data and estimated their parameters. This could be of interest when communication links are not available during estimation, or too costly to use. Two simple models are simulated: (i) the exponential model, suitable for modeling inter-arrival time of customers or serving time or other type of duration and (ii) the Gaussian model which is ubiquitous. Both model fall into the general exponential family; which is equipped with a Riemmanian metric as described below.

Consider an open set $\Theta \subset \mathbb{R}^n$ indexing a family of probability distribution over some set $\mathcal{X}$ with some common reference measure μ; their density function wrt μ is denoted p_θ, with $\theta \in \Theta$. Also denote $l(x;\theta) = \log p_\theta(x)$ the log-likelihood function and to $\theta = (\theta_1, \ldots, \theta_n) \in \Theta$, its $n \times n$ Fisher information matrix with coefficients given by:

$$g_{i,j} = \int_{\mathcal{X}} \left(\frac{\partial l}{\partial \theta^i} \frac{\partial l}{\partial \theta^j} \right) p_\theta(x)\mu(dx)$$

Then (Θ, g) is a Riemmanian manifold.

The numerical examples studied in this paper belong to the *exponential family* (in canonical form), $\theta = (\theta_1, \ldots, \theta_q)$,

$$\log p_\theta(x) = C(x) + \sum_{i=1}^{q} \theta_i F_i(x) - \psi(\theta)$$

for some functions C, ψ and F_i.

Example 1:(Ordinary exponential law)

To recover the iid exponential distribution from the exponential family, set $\mathcal{X} = \{(x_1, \ldots, x_q) : \forall 1 \le i \le q, x_i \ge 0\}$, the parameter space is: $\Theta = \{(\theta_1, \ldots, \theta_q) : \forall 1 \le i \le q, \theta_i \ge 0\}$. Set the functions $\{C, F_1, \ldots, F_q\}$ as: $C(x) = 0, F_i(x) = -x_i$, $\psi(\theta) = \sum_i \log(\theta_i)$, so that to recover the density function: $p_\theta(x) = \prod_i \theta_i^{-1} \exp(-\sum_i \theta_i x_i)$. In this example we find that the geodesic between two elements p_λ and p_θ is of the form $p_{\gamma(t)}$ where: $\forall i \in \{1, \ldots, q\}, t \in [0, 1]$: $\gamma_i(t) = \lambda_i \left(\frac{\theta_i}{\lambda_i}\right)^t$;

Hence, Algorithm Random Pairwise Midpoint writes: $\theta_{v_n}(n) = \theta_{w_n}(n) = \sqrt{\theta_{v_n}(n-1) \circ \theta_{w_n}(n-1)}$ where $\theta_v(n)$ denotes the estimation of θ by agent v at time n, $\circ$ denotes the componentwise product, and $\sqrt{}$ is taken componentwise. In this example, $q = 1$, $\theta_v(0)$ was sampled iid uniformly from $(0, 1)^q$, for the graph of the network we use the complete graph K_N (N=10) and run Algorithm Random Pairwise Midpoint. In figure 1 the blue curve represents the logarithm of the disagreement function with respect to iterations $n \mapsto \log \Delta(X_n)$, we observe a straight line with negative slope which indicates exponential convergence to consensus.

Example 2:(The normal distribution) To recover the normal distribution, set $\mathcal{X} = \mathbb{R}$, $q = 2$, $\Theta = \{(\theta_1, \theta_2) : \theta_1 \ge 0, \theta_2 \in \mathbb{R}\}$, $C(x) = 0$, $F_1(x) = x^2$, $F_2(x) = x$, $\psi(\theta) = \log(\frac{\pi}{\theta_1}) + \frac{\theta_2^2}{\theta_1}$, so that: $p_\theta(x) = \sqrt{\frac{\theta_1}{\pi}} \exp\left(-\theta_1(x - \frac{\theta_2}{2\theta_1})^2 + (\frac{3\theta_2}{4\theta_1})^2\right)$.

The family of Gaussian statistical models has constant negative sectional curvature $K = \frac{-1}{2}$ (for a computation of the metric, Christoffel symbols and sectional curvature see: [ABNK+87, p.190]). In this example we have a 2 dimensional manifold; Algorithm Random Pairwise Midpoint was implemented using a discrete numerical solving technique for the geodesic equations.We use the same graph, Evolution of the log disagreement function $\log \Delta_n$ when n increases is represented by the green curve in Figure 1. One can observe convergence to consensus at an exponential rate.

B. Positive definite matrices

The scenario in this experiment is the following. Each sensor in a network estimates a covariance matrix for some observed multivariate process. Then the network seeks a consensus on these covariance matrices. We implemented the proposed algorithm using known facts from the geometry of positive definite matrices [Lan99, chap. 12]. In particular, [Lan99, p.326]: $d(M,N)^2 = \mathrm{tr}\{\log(MN^{-1})\log(MN^{-1})^T\}$ and $\langle\frac{M+N}{2}\rangle = M^{1/2}(M^{-1/2}NM^{-1/2})^{1/2}M^{1/2}$. It is straightforward to implement Algorithm Random Pairwise Midpoint and compute $\log \Delta\big(M(n)\big)$ at each iteration n; where $M(n) = (M_1(n), \ldots, M_N(n))$ denotes the tuple of positive definite matrix held by the agents $1 \leq v \leq N$ at time n. We generated $M_v(0)$ as $\sum_{k=1}^{p} X_{k,v}X_{k,v}^T$ where $X_{k,v} \sim \mathcal{N}(0, I_q)$ are independent standard multivariate Gaussian vectors of dimension q (in the numerical experiment $q = 5$,and we assume a complete graph). Note that the algorithm proposed is very close to the one proposed in [Bon13] which consists in the iterations $M^{1/2}(M^{-1/2}NM^{-1/2})^{\gamma_n}M^{1/2}$ where γ_n is a sequence of stepsize such that $\sum_n \gamma_n = +\infty$ and $\sum_n \gamma_n^2 < \infty$. In particular stepsize γ_n should go to 0 while in our case it is kept constant at $1/2$. The red and sky blue curves in figure 1 represent the function $\log(\Delta_n)$ for respectively the gradient descent method (implemented with a decreasing step size $\gamma_n = \frac{1}{n}$) and consensus midpoint algorithm; the initialization and graph used for both algorithms being the same, the two curve can be compared so as to deduce that while the consensus midpoint algorithm leads to exponential convergence, the $\log(\Delta_n)$ curve for the gradient descent method seems to converge slower. Actually the fact that it converges slower is coherent with stochastic approximation with decreasing stepsize. Indeed, it is known that, in the Euclidean setting [KY97, chap. 10], for stepsize γ_n, the speed of convergence is of order ${\gamma_n}^{-1/2}$.

V. Conclusion

We have presented an extension to the (RPG) to the Case of Riemannian manifolds in the asynchronous pairwise case. We identified a set of conditions (nonpositive curvature) that guarantees the convergence of the Random Pairwise Midpoint algorithm. Convergence towards an aribitrary consensus state occurs at exponential speed. Our experiments with statistical manifolds and positive definite matrices confirm those results and validate our approach.

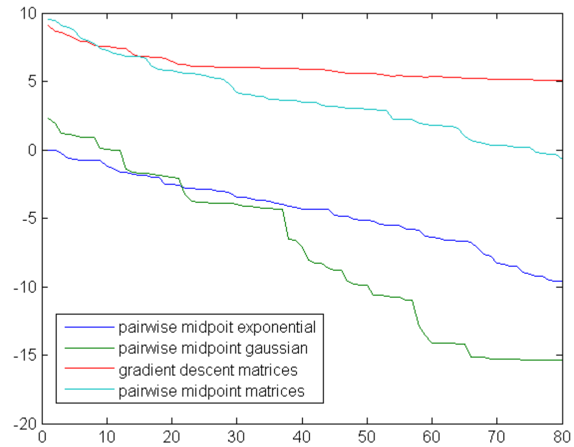

Fig. 1: Plot of $n \mapsto \log \Delta_n$ where n is the iteration index. Because of space constraints all the experiments are reported on the same figure. The point is to show a good match between what we observe in practice and theorems 1 and 2. However absolute values of $\log \Delta$ should not be compared between experiments. We have also simulated the gradient descent algorithm of [Bon13] for symmetric matrices (red curve); we can see that, unlike Algorithm Random Pairwise Midpoint, it does not converge at exponential speed, because of the decreasing stepsize scheme that it implies.

References

[ABNK+87] S.-I. Amari, O. Barndorff-Nielsen, R. Kass, S. Lauritzen, and C. Rao. Differential geometry in statistical inference. *Lecture Notes-Monograph Series*, pages i–240, 1987.

[AN00] S. Amari and H. Nagaoka. *Methods of information geometry*. American Mathematical Society, 2000.

[BGPS06] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[Bon13] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *Automatic Control, IEEE Transactions on*, to appear in Sept. 2013.

[Bur06] Mike Burrows. The chubby lock service for loosely-coupled distributed systems. In *Proceedings of the 7th symposium on Op. sys. design and implem.*, pages 335–350, 2006.

[DC92] M.P. Do Carmo. *Riemannian geometry*. Birkhäuser, 1992.

[KY97] H. Kushner and G. Yin. *Stochastic approximation algorithms and applications*. Springer, 1997.

[Lan99] S. Lang. *Fundamentals of differential geometry*. Springer, 1999.

[SBS10] A. Sarlette, S. Bonnabel, and R. Sepulchre. Coordinated motion design on lie groups. *Automatic Control, IEEE Transactions on*, 55(5):1047–1058, 2010.

[SG07] L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *IEEE Conf. Decision and Control*, pages 2289–2294, 2007.

[SG08] A. Schizas, I.and Ribeiro and G. Giannakis. Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals. *Signal Processing, IEEE Transactions on*, 56(1):350–364, 2008.

[SS09] A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM Journal o.n Control and Optimization*, 48(1):56–76, 2009.

[SSS07] L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the n-torus. *Systems and Control Letters*, 56(5):335–341, 2007.

[TAV11] Roberto Tron, Bijan Afsari, and René Vidal. Average consensus on riemannian manifolds with bounded curvature. In *IEEE Conf. on Decision and Control*, pages 7855–7862, 2011.