ALGORITHMS ON NETWORKS

Vivek Borkar

IIT Bombay

Workshop on Challenges in Networks,

ICTS, Bengaluru

1st Feb., 2024

Outline

1. Background:

   Classical stochastic approximation

2. Distributed algorithms:

   (a) Asynchronous

   (b) Synchronous

   (c) Federated learning

3. Algorithms for consensus in learning:

(a) Tsitsiklis-Bertsekas-Athans paradigm

(b) Extensions

Stochastic approximation (Robbins and Monro '51)

Objective: Find the root of a nonlinear function $h : \mathcal{R}^d \mapsto \mathcal{R}^d$, i.e., solve $h(x) = \theta :=$ the zero vector, given noisy observations.

That is, a black box outputs '$h(x)+$ noise' on input x.

The Robbins-Monro scheme is: Take stepsizes $a(n) > 0$ such that $\sum_n a(n) = \infty$ and $\sum_n a(n)^2 < \infty$. For $n \geq 0$, do:

$$x(n+1) = x(n) + a(n)[h(x(n)) + M(n+1)].$$

Here, $M(n), n \geq 1$, is a martingale difference noise, i.e., integrable random variables adapted to increasing σ-fields $\{\mathcal{F}_n\}$ such that $\sigma(X(m), m \leq n) \subset \mathcal{F}_n \ \forall n$, with

$$E[M(n+1)|\mathcal{F}_n] = \theta.$$

The expression in red is the noisy measurement of $h(x(n))$.

That is, $h(x(n))$ and $M(n+1)$ are not separately known, only their sum is.

This is more general than it appears. A typical algorithm has the form

$$x(n+1) = x(n) + a(n) f(x(n), \xi(n+1))$$

where $\{\xi(n)\}$ are i.i.d. Then set $h(x) := E[f(x, \xi(n))]$ and $M(n+1) := f(x(n), \xi(n+1)) - h(x(n))$.

The classical, purely probabilistic, approach for analyzing this scheme is based on the Siegmund-Robbins theorem on a.s. convergence of 'almost supermartingales' and allied results.

In contrast, I shall describe the so called 'ODE approach' (Meerkov-Derevetsky-Fradkov-Ljung-Benaim, early '70s onwards).

Consider $a(n)$ as a time step. Then the iteration

$$x(n+1) = x(n) + a(n)[h(x(n)) + M(n+1)].$$

is a noisy Euler scheme for the ordinary differential equation (ODE)

$$\dot{x}(t) = h(x(t))$$

with decreasing stepsize.

To make this precise, define the algorithmic time scale:

$$t(0) = 0, t(n) = \sum_{m=0}^{n-1} a(m), m \geq 1.$$

Consider the continuous, piecewise linear interpolation

$$\bar{x}(t) = x(n) + \left(\frac{t - t(n)}{t(n+1) - t(n)} \right) (x(n+1) - x(n)),$$

for $n \geq 0$.

Consider $x^s(t), t \geq s$, for $s \geq 0$ defined by

$$\dot{x}^s(t) = h(x^s(t)), \ t \geq s, \ x^s(s) = \bar{x}(s),$$

i.e., a solution of the ODE on $[s, \infty)$ with initial condition at s matched with $\bar{x}(s)$.

Then using Gronwall inequality, one can prove that

$$\max_{y \in [s, s+T]} \|\bar{x}(y) - x^s(y)\| \to 0 \text{ a.s. } \forall\ T > 0.$$

1. $a(n) \to 0 \implies$ discretization errors go to zero.

2. $\sum_n a(n)^2 < \infty \implies$ under suitable hypotheses, martingale convergence theorem implies that a.s., $\sum_n a(n)M(n+1) < \infty \implies \sum_{m=n}^{\infty} a(m)M(m+1) \to 0$ $\implies$ errors due to noise go to zero, a.s.

3. $\sum_n a(n) = \infty \implies t(n) \uparrow \infty$. (entire time axis covered)

An invariant set A of the ODE is said to be internally chain transitive if given $y, z \in A$, $T > 0$ and $\epsilon > 0$, we can find $n \geq 1$ and points $x_i \in A$, $0 \leq i \leq n$, such that $x_0 = y, x_n = z$, and there exist trajectories $x^i(\cdot), 0 \leq i < n$ of the ODE of duration $\geq T$, such that

$$\|x^i(0) - x_i\| < \epsilon, 0 \leq i < n, \text{ and } \|x^i(T) - x_{i+1}\| < \epsilon, 1 \leq i < n.$$

Main result (Benaim): $x(n)$ converges a.s. to an *internally chain transitive* invariant set of the ODE.

For algorithms, we typically desire convergence a.s. to specific points, which then must be equilibria of the ODE, i.e., zeros of h.

Suppose the only possible ω-limit sets are isolated equilibria. If the equilibria are hyperbolic (i.e., the Jacobian matrix $D_x h$ of h at the equilibrium x does not have eigenvalues on the imaginary axis), they are isolated.

If in addition the noise is 'rich enough' in all directions, one can claim a.s. convergence to stable equilibria of the ODE. (Usually the desired equilibria are stable.).

Benaim, M., 1996. A dynamical system approach to stochastic approximations. *SIAM Journal on Control and Optimization*, 34(2), pp. 437-472.

Benaim, M., 2006. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII* (pp. 1-68). Springer Berlin-Heidelberg.

Borkar, V. S. Stochastic approximation: a dynamical systems viewpoint. Hindustan Publishing Agency (2022) and Springer Nature (2024).

Distributed synchronous algorithms:

N processors/agents sitting on the nodes of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set, $|\mathcal{V}| = N$, and $\mathcal{E}$ is the edge set.

An edge from $i \in \mathcal{V}$ to $j \in \mathcal{V}$ is denoted $(i, j) \in \mathcal{E}$.

The ith node, $1 \leq i \leq N$, computes $x^i(n) \in \mathcal{R}^d, n \geq 0$, based on delayed information from other nodes.

Consider a complete graph.

General scheme:

N concurrent iterations in $\mathcal{R}$ given by* :

for $1 \leq i \leq N$ and $n \geq 0$,

$$x^i(n+1) = x^i(n) \; + \; a(n) I\{i \in Y_n\} \times$$
$$[h^i(x^1(n - \tau_{1i}(n)), x^2(n - \tau_{2i}(n)), \cdots, x^N(n - \tau_{di}(n))$$
$$+ \; M^i(n+1)].$$

Here, $a(n) > 0$ satisfies the 'Robbins-Monro conditions'

$$\sum_n a(n) = \infty, \; \sum_n a(n)^2 < \infty.$$

*Generalization to $\mathcal{R}^d$ is possible.

Here,

1. $0 \leq \tau_{ji}(n) \leq n, i, j \in \mathcal{V}, n \geq 0$, are random delays, such that at time n, ith node ($\approx$ processor / agent) knows $x^j(n - \tau_{ji}(n))$, but not $x^j(m)$ for $m > n - \tau_{ji}(n)$,

2. $Y_n :=$ the set of indices that got updated at time n. The rest retain their previous value (i.e., $i \notin Y_n \Longrightarrow x^i(n+1) = x^i(n)$).

This requires 'time-stamping'. Other models are also possible (e.g., inclusion of data loss corresponding to infinite delay, no time stamping, etc.).

If the delay τ at time n is bounded by some $K < \infty$, it causes an error of $O\left(\sum_{k=n-K}^{n} a(k)\right) \to 0$ as $n \to \infty$. Hence it does not affect the convergence properties, only the speed thereof. More generally, a convenient conditional moment bound suffices.

The fact that not all components are updated at each time is more problematic. It leads to a limiting ODE of the type

$$\dot{x}(t) = \Lambda(t) h(x(t))$$

where $t \mapsto \Lambda(t) \in \mathcal{R}^{d \times d}$ takes values in diagonal $d \times d$ matrices with non-negative diagonal entries.

Intuitively, $\lambda_j(t) :=$ the jth diagonal entry of $\Lambda(t)$ reflects the 'instantaneous relative frequency' of updates of the ith component at time t. This can affect the asymptotic behaviour adversely except in special cases.

For example, for stochastic gradient descent, i.e., $h(x) = -\nabla f(x)$, we have

$$\frac{df}{dt}(x(t)) = -\sum_i \lambda_i(t) \left(\frac{\partial f}{\partial x_i}(x(t))\right)^2 < 0$$

away from critical points as long as $\lambda_i(\cdot)$ remain bounded away from zero.

For this, one needs the 'relative frequencies'

$$\nu(i,n) := \frac{\sum_{m=0}^{n-1} I\{i \in Y_m\}}{n}$$

to remain bounded away from zero a.s. as $n \uparrow \infty$, i.e., all components should be sampled comparably often.

Another case when this works is $h(x) = F(x) - x$ where $\|F(x) - F(y)\|_\infty \leq \alpha \|x - y\|_\infty$ for some $\alpha \in (0,1)$. Then the ODE is

$$\dot{x}(t) = \widetilde{F}_t(x(t)) - x(t)$$

where $\widetilde{F}_t(x) = (I - \Lambda(t))x + \Lambda(t)F(x)$ remains an $\|\cdot\|_\infty$-contraction with a common fixed point.

A common scenario in reinforcement learning is when $\{Y_n\}$ is an irreducible Markov chain on the index set $\{1, 2, \cdots, d\}$,

or,

it is a controlled Markov chain coupled with a control choice that is 'ϵ-greedy', i.e., with probability $1 - \epsilon$, it is the current guess for the optimal choice for the current state, and with probability ϵ, it is uniform over the set of available controls ('exploration' vs 'exploitation').

An alternative is to use sufficiently rapidly decreasing $a(n)$ and replace $a(n)$ in the algorithm by $a(\nu(i,n))$. Then one has

$$\lim_{n\uparrow\infty} \frac{\sum_{m=0}^{n} a(\nu(i,m))I\{i \in Y_m\}}{\sum_{m=0}^{n} a(\nu(j,m))I\{j \in Y_m\}} \to 1 \text{ a.s., } 1 \le i,j \le d,$$

and the time scales of different components are 'matched', leading to the limiting ODE

$$\dot{x}(t) = \frac{1}{d} h(x(t)).$$

Examples are $a(n) = \frac{1}{n}$, $\frac{1}{1+n\log n}$ etc., whereas $\frac{1}{n^{2/3}}$ won't work.

For example, for $a(n) = \frac{1}{n}$, the RHS above is

$$= \frac{\log\left(\sum_{m=0}^{\nu(i,n)} a(m)\right)}{\log\left(\sum_{m=0}^{\nu(j,n)} a(m)\right)} \approx \frac{\log(\nu(i,n)/n) + \log n}{\log(\nu(j,n)/n) + \log n} \to 1$$

when the sampling of components is 'comparably frequent'.

One can think of $\{\nu(i,n)\}$ as the 'local clock' at i. This formulation becomes essential when the computation is fully asynchronous and the 'global clock' $n = 0, 1, 2, \cdots$, can be an artifice as long as the causal dependences are respected.

Federated learning:

Here a central server pools together the outputs of multiple processors and computes a consolidated result, then sends it back to them.

A typical formulation involves periodic updates by the server based on computations which are ready by the time the processors are polled.

The processors are asynchronous and heterogeneous.

Problem of 'stragglers': ameliorated by duplicating data across processors, sometimes combined with coding. Other problems such as changing graph topology etc. may also occur. Also, there can be privacy issues.

Variations such as batch processing, adaptive synchronization and adaptive step-sizes are used.

Joshi, G., 2022. Optimization Algorithms for Distributed Machine Learning. Springer Nature.

Algorithms for consensus:

Tsitsiklis-Bertsekas-Athans model: $\mathcal{N}(i) :=$ the set of neighbours of i. Assume $i \in \mathcal{N}(j) \Longleftrightarrow j \in \mathcal{N}(i)$.

$$x^i(n+1) = \sum_{j \in \mathcal{N}(i)} p(j|i) x^j(n) + a(n)[h^i(x(n)) + M^i(n+1)].$$

Here $P := [[p(j|i)]]_{i,j \in \mathcal{V}}$ is an irreducible stochastic matrix with stationary distribution π.

Tsitsiklis, J., Bertsekas, D. and Athans, M., 1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), pp. 803-812.

1. $\Sigma_{j \in \mathcal{N}(i)} p(j|i) x^j(n)$ is the 'Gossip' component for the averaging / consensus effect that operates on the fast 'natural' time scale. Goes back to:

Degroot, M., 1974. Reaching a consensus. *Journal of American Statistical Association*, 69, pp. 118-121.

2. $a(n)[h^i(x(n) + M^i(n+1)]$ is the 'Learning' component, operating on the slow 'algorithmic' time scale given by

$$t(0) = 0, t(n) = \sum_{m=0}^{n-1} a(m), n \geq 0.$$

For example, $a(n) = \frac{1}{n} \Longrightarrow t(n) \approx \log n$.

Intuition: By itself, gossip forces the iterates to the set of its fixed points, i.e., the one dimensional invariant subspace of constant vectors.

Combined with the slow time scale of the learning scheme, this confines the learning dynamics to this subspace asymptotically, implying consensus.

For example, for $h = -\nabla f$, we get convergence to a common local minimum.

Assume that the iterates are bounded (this needs a proof). Then asymptotically, the individual iterates can be shown to track a common trajectory of the ODE

$$\dot{x}(t) = \sum_i \pi(i) h^i(x(t))$$

where π is the stationary distribution of $P := [[p(j|i)]]$.

This amounts to a dynamic consensus and in case of convergence, consensus, i.e., convergence to a common limit for all processors.

Related models are used for studying opinion dynamics, dynamics of robotic swarms, etc.

Other possibilities exist, see the following for an excellent survey, but a bit dated already.

Nedich, A., 2015. Convergence rate of distributed averaging dynamics and optimization in networks. Foundations and Trends® in Systems and Control, 2(1), pp. 1-100.

Another general model is:

$$\begin{aligned} y^i(n) &= \sum_{j \in \mathcal{N}(i)} p(j|i) x^i(n), \\ x^i(n+1) &= y^i(n) + a(n)[h^i(y(n)) + M^i(n+1)]. \end{aligned}$$

For more, see:

Sayed, A.H., 2014. Adaptation, learning, and optimization over networks. Foundations and Trends® in Machine Learning, 7(4-5), pp. 311-801.

Other variants include polling, i.e., each node (say) i, turns 'on' according to an independent Poisson clock with rate 1, samples one neighbour $j \in \mathcal{N}(i)$ with equal probability, and replaces $x^i(n)$ by

$$x^i(n+1) = y^{ij}(n) + \frac{1}{\nu(i,n)}(-\nabla f_i(y^{ij}(n)) - x^i(n) + M^i(n+1)),$$

$$x^j(n+1) = y^{ij}(n) + \frac{1}{\nu(j,n)}(-\nabla f_j(y^{ij}(n)) - x^j(n) + M^j(n+1)),$$

where $y^{ij}(n) := \frac{1}{2}(x^i(n) + x^j(n))$.

The components of x^i, x^j other than the ith, jth remain unchanged.

This leads to $\pi \approx$ the uniform distribution. This is useful, e.g., for the minimization of a sum of functions, which is of great importance in machine learning. See

Ram, S. S., Nedić, A. and Veeravalli, V. V., 2009. Asynchronous gossip algorithms for stochastic optimization. *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3581-3586.

More generally, any desired π can be obtained by using a Metropolis-Hastings type $p(j|i)$'s.

More generally, one can consider a time-varying sequence of transition probabilities $P_n = [[p_n(j|i)]], n \geq 0$. An important generalization covered thereby is to changing graph topologies, requiring time-dependent selection probabilities $p(j|i)$.

'Polling' based schemes are also possible.

Say that $\pi_n, n \geq 0$, is an absolute probability sequence for $\{P_n\}$ if $\pi_n^T = \pi_{n+1}^T P_n \ \forall n \geq 0$ **(Kolmogorov)**.

Every $\{P_n\}$ has an absolute probability sequence **(Blackwell)**.

Say that $\{P_n\}$ is ergodic if $\exists \phi_n \in \mathcal{P}(S), n \geq 0$, such that

$$\lim_{N \uparrow \infty} \prod_{m=n}^{N} P_m = \mathbf{1}\phi_n^T,$$

where $\mathbf{1} := [1, 1, \cdots, 1]^T$. Then $\{\phi_n\}$ is the unique absolute probability sequence **(Kolmogorov)**.

Under additional conditions on the graph structure, there exists $0 < \beta < 1$ such that for $n \geq n_0 \geq 0$,

$$\|x_n - \phi_{n_0}^T x_{n_0} \mathbf{1}\|^2 \leq \beta^{n-n_0} \|x_{n_0} - \phi_{n_0}^T x_{n_0} \mathbf{1}\|^2.$$

(**Touri**)

Sufficient conditions for ergodicity are given in:

Chatterjee, S. and Seneta, E., 1977. Towards consensus: Some convergence theorems on repeated averaging. Journal of Applied Probability, 14(1), pp.89-97.

Another algorithm for computing arithmetic means is the 'Push-sum' algorithm.

Let $\mathcal{N}_{i,k}^{in}$ denote the in-neighbourhood of node i and d_i^{out} its out-degree, at time k. The scheme is

$$x^i(n+1) = \sum_{j \in \mathcal{N}_{i,k}^{in}} \frac{x^j(k)}{d_j^{out}}, \; y^i(n+1) = \sum_{j \in \mathcal{N}_{i,k}^{in}} \frac{y^j(k)}{d_j^{out}}.$$

with $x^i(0)$'s is as prescribed, and $y^i(0) = 1 \; \forall i$. Then

$$\frac{\sum_{k=0}^{n} x^i(k)}{\sum_{k=0}^{n} y^i(k)} \to \frac{1}{d} \sum_{i=1}^{d} x^i(0).$$

Kempe, D., Dobra, A. and Gehrke, J., 2003. Gossip-based computation of aggregate information. In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pp. 482-491.

Bénézit, F., Blondel, V., Thiran, P., Tsitsiklis, J. and Vetterli, M., 2010. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In Proceedings of the IEEE International Symposium on Information Theory (pp. 1753-1757).

A general scheme for changing graph topology or randomly transmitting agents (without 'gossip'):

$$x^i(n+1) = x^i(n) + \sum_{j \in \mathcal{N}(i)} a(\nu(n,i,j))\xi_{ij}(n) \times$$
$$(h_{ij}(x^1(n - \tau_{1i}(n)), \cdots, x^d(n - \tau_{di}(n))) + M_{ij}(n+1)),$$

where $\xi_{ij}(n) = 1$ if j communicates with i at time n, 0 otherwise. We make suitable assumptions on the delays and stepsizes. If $\xi_{ij}(n)$'s are IID with $E[\xi_{ij}] = q_{ij}$, then the iterates track a time-scaled version of the ODE

$$\dot{x}^i(t) = \sum_{j \in \mathcal{N}(i)} h_j(x(t)).$$

As expected, convergence rates of consensus algorithms depend on the graph topology. They have been worked out for specific classes of graphs.

Nedić, A., Olshevsky, A. and Rabbat, M.G., 2018. Network topology and communication-computation tradeoffs in decentralized optimization. Proceedings of the IEEE, 106(5), pp. 953-976.

One can also use selection probabilities that are modulated by the iterates themselves, i.e., $P_x := [[p_x(j|i)]]$.

Suppose P_x is irreducible $\forall x$ and thus has a unique stationary distribution π_x that is Lipschitz in x. Then the iterates track a common ODE

$$\dot{x}(t) = \sum_i \pi_{x(t)}(i) h^i(x(t)).$$

Example: $h^i(x) = -\nabla f(x^i)$, $\pi_x(i) \propto e^{-\frac{f(x^i)}{T}}$. This scheme augments a distributed gradient descent with simulated annealing like weights, putting a higher weight on the outputs of better performing processors.

A further generalization:

$$x^i(n+1) = f_i(x(n)) + a(n)[h^i(x(n)) + M^i(n+1)],\ n \geq 0,$$

where $f(\cdot) = [f_1(\cdot), \cdots, f_d(\cdots)] : \mathcal{R}^d \mapsto \mathcal{R}^d$ is such that the iterates

$$y(n+1) = f(y(n)) \qquad (*)$$

converge to $C :=$ the set of fixed points of $f(\cdot)$. Then the effect is to restrict the dynamics $\dot{x}(t) = h(x(t))$ to C.

Both these cases are analyzed in :

Mathkar, A.S. and Borkar, V.S., 2016. Nonlinear gossip. SIAM Journal on Control and Optimization, 54(3), pp.1535-1557.

In fact, a further generalization is to replace $(*)$ by a stochastic approximation on a faster time scale induced by stepsizes $\{b(n)\}$ that satisfy, in addition to satisfying the Robbins-Monro conditions, the additional condition:

$a(n) = o(b(n))$, i.e., $\frac{a(n)}{b(n)} \to 0$.

Special case: Consider $(*) :=$ a stochastic approximation version of a distributed version of the Boyle-Dykstra-Han algorithm for projection on the intersection of convex sets. A stochastic approximation version thereof on a faster time scale also works. This leads to projected stochastic approximation.

Shah, S.M. and Borkar, V.S., 2018. Distributed stochastic approximation with local projections. SIAM Journal on Optimization, 28(4), pp. 3375-3401.

Further possibilities:

Can more general 'regularly perturbed' ODEs be put to good use? See, e.g.,

Artstein, Z., Kevrekidis, I. G., Slemrod, M. and Titi, E. S., 2007. Slow observables of singularly perturbed differential equations. *Nonlinearity*, 20(11), 2463-2481.

Finer analysis of the time scale separation, limit theorems, concentration phenomena etc.?

THANK YOU!

BEDANKT!