1 Escaping local minima with Langevin Monte Carlo (Yuchen Zhang)

Joint work with Moses Charikar and Percy Liang.

The goal is to minimize a nonconvex function $f : K \to \mathbb{R}$. Nonconvex optimization is universal in ML applications: mixture models, matrix/tensor decomposition, neural networks, etc.

The main challenge is to avoid non-optima local minima. Doing this is NP-hard in general, but we can hope to excape shallow local optima.

If we properly initialize gradient descent we converge to local min. It will get stuck at local min because it is a greedy method, only looking at the neighborhood.

A natural approach is to inject noise,

$$x \hookleftarrow x - \eta \cdot (\nabla f(x) + \sigma \cdot w), \quad w \sim N(0, I).$$

One thing to pay attention to is how to choose step size. If you choose very small step size, you lose capability of escaping local min. It's 0-mean so this reduces to regular gradient descent. Choose large enough noise to escape before it gets canceled out. But this causes stability problems: overshooting, divergence, especially if f is not very smooth.

Solutoin: Langevin Monte Carlo. Attracting attention because of connection to successful SGD. Instead of scaling with step size, scale with $\sqrt{}$ step size. It imitates Langevin diffusion in physics. $1/\xi$ controls temperature.

$$x \hookleftarrow x - \eta \cdot \nabla f(x) + \sqrt{\frac{2\eta}{\xi}} w.$$

Note when $\eta \to 0$, $\sqrt{\frac{\eta}{\xi}} \gg \eta$, still able to escape.

Why scale with $\sqrt{\eta}$? Only by this choice does it converge to the stationary distribution, $\mu(x) \propto e^{-\xi f(x)}$. When ξ is large and $x \sim \mu(x)$, x minimizes f. This is good for nonconvex optimization. You need f to be smooth and $e^{-\xi f}$ to be integrable. [1]

I assume f is smooth and bounded on K, and K is compact. It doesn't need to be convex, but I make assumptions on the corner (no sharp corners). I think this is not necessary but is convenient.

For smooth functions and small enogh stepsize, LMC asymptotically converges (Roberts, Tweedie 96). For convex f, LMC converges to μ in polytime (Bubeck 2015, Dalalyan 2016). For nonconvex functions LMC converges in exponential time, for simple 1-D nonconvex functions. This is an area that is underexplored.

On the practical side, LMC is successful in practice.

- Prevents overfitting: logistic regression, ICA (Welling Teh 2011)
- Learn deep nets: neural programmer (Neelakantan 2015), neural RAM (Kurach 2015), GPUs (Kaiser, Sutskever 2015), bidirectional LSTM (Zeyer 2016).

[1] There is a noncompact analysis in 80's, purely asymptotic. For a compact set you can do reflected Brownian motion, or reject steps outside.

How to explain good performance?

LMC can hit a good solution much earlier than it converges to the stationary distribution. Ex. W-shaped function. Sampling means finding all good solutions; optimization means finding one.

The reason we don't have a polytime guarantee is that mixing time is too pessimistic for optimization. We look at hitting times to a set U (target set) instead. If we have an arbitrary defined set, we get into NP-hard problems. Whether we get a polytime bound depends on how we define this set.

Empirical risk can have many local min, but can have shallow local min not present in the global min. We want to get to population local min.

An ε-approximate local min is $\{x \in K : \|\nabla f(x)\| \le \varepsilon, \nabla f(x) \succeq -\sqrt{\varepsilon} I\}$ Can also get by cubic regularized Newton, noisy gradient descent. *Here we can achieve this for population risk even if we only get access to empirical risk!*

We have bad complexity in d in number of iterations.

We only need the Liipschitz constant for population loss function to be small.

We present non-asymptotic analysis for the LMC algorithm (and stochastic version SGLD) on general nonconvex functions, polynomial-time guarantee for hitting certain optimality set, and apply to ERM finding local minima of the population risk, and to learning halfspaces under 0-1 loss (note there is no guarantee on concentration of gradient and Hessian matrices, which are 0 almost everywhere).

First, some definitions and notations.

For $f : K \to \mathbb{R}$, define

$$\mu_f(x) := \frac{e^{-f(x)}}{\int_K e^{-f(x)}} \tag{1}$$

$$\mu_f(\partial A) := \lim_{\varepsilon \to 0} \frac{\mu_f(\{x \in K : d(x, A) \le \varepsilon\}) - \mu_f(A)}{\varepsilon} \tag{2}$$

$$C_f(V) := \inf_{A \subseteq V} \frac{\mu_f(\partial A)}{\mu_f(A)} \tag{3}$$

(restricted Cheeger constant), min ratio betw

If $C_f(V)$ is very small, then there exists some set $A \subseteq V$ with a very narrow exit. If there is a Markov process with stationary distribution μ_f and it was initialized in A, then it takes a long time to move out. This defines geometric property of function, cf. conductance which is property on Markov process.

Restricted Cheeger constant $C_{\xi f}(V)$ measures how fast LMC can escape V. Ex. for the W function, the restricted Cheeger constant for neighborhood of one local min is small.

If you run too long, won't the bias accumulate? I can choose step large enough so we hit before bias accumulates too much. With small Cheeger constant I have to choose small step size.

Theorem 1.1. *For arbitrarily smooth f and $U \subseteq K$, LMC (with some set size and fixed temperature) hits U whp in T iteractions, where*

$$T \le \frac{\text{poly}(d)}{C_{(\xi f)}(K \backslash U)^4}.$$

Initialization is arbitrary. If I choose small step size, I need more iterations. Note that Cheeger constant depends on temperature. Temperature is chosen according to effective dimension of problem, volume of set you want to hit. (Choose proportional to $\frac{1}{d}$ in worst-case. If region around manifold, then you can choose temperature larger.)

Given the function and the set, there exists a step size. Step size: (28) on p. 17.

Lower bound: inversely proportional to conductance. Upper bound is at most square of reciprocal of conductance. This is a mixing time lower bound. If you look at a convex function it is the same thing. The bound we have is square of mixing time for convex case.

Is this an algorithm that can be implemented? There is a theoretical step size, but in practice we choose the step-size by cross-validation.

MT: We go through mixing time lemma, to Gibbs measure, as corollary get suboptimality. Ours is global optimal guarantee. We pay exponentially in the saddle height. We also have $\frac{1}{\varepsilon^4}$. If you choose the step size and temperature, then things become bad. There is a lower bound.

You can make function grow quadratically; choose temperature schedule.

Proof.

- Construct a time-reversible Markov chain; prove hitting time on par with LMC. (LMC is not reversible.)
- Prove hitting time inversely depends on reverse conductance.

  (Look at specification of algorithm, specifics of Markov chain.)
- Restricted conductance is lower bounded by $(C_{(\xi f)}(K\backslash U))^2$.

□

This is a general theorem that doesn't give a polytime guarantee.

Now we look at certain cases where we can show Cheeger constant is not exponenitally small.

A simple class of functions is convex functions. Choosing temperature to be low enough,

Proposition 1.2: If $\xi \geq O\left(\frac{d}{\varepsilon^2}\right)$, $C_{(\xi f)}(K\backslash U)$ is lower bounded by $\Omega(1)$ for U set of ε-approximate global min,

$$U = \left\{x : f(x) \leq \inf_{x\in K} f(x) + \varepsilon\right\}.$$

More interesting case is nonconvex.

Proposition 1.3: If $\xi \geq O\left(\frac{\text{poly(params)}}{\varepsilon^2}\right)$, then C lower bounded by $\Omega(\sqrt{\varepsilon})$, where

$$U := \left\{x : \|\nabla f(x)\|_2 \leq \varepsilon, \nabla^2 f(x) \succeq -\sqrt{\varepsilon} I\right\}.$$

There is implicit dependence on dimension, trace of hessian. Local minimum has 0 gradient, saddle point also has 0 gradient but H is non-PSD. U avoids strict saddle points.

d^4 is impractical but there has to be poly(d) dependence.

If f is nonconvex and uniformly close to F:

Proposition 1.4: $\|f - F\|_\infty := \sup_{x \in K} |f(x) - F(x)| \leq \nu$,

$$C_{(\xi f)}(K \backslash U) \geq e^{-2\xi v} C_{(\xi f)}(K \backslash U).$$

If $\|f - F\|_\infty = O\left(\frac{1}{\xi}\right)$ then $C_{(\xi f)}(K \backslash U) = C_{(\xi F)}(K \backslash U)$.

Ex. 0-1 loss. First define smoothed version on 0-1 loss. Guarantee smoothed version of function is uniformly close. But we cannot guarantee all local minima are close to population global minimum.

Corollary 1.5. *Run LMC on smooth f.*

1. *If F convex, $\|f - F\|_\infty = O\left(\frac{\varepsilon^2}{d}\right)$, then LMC hits ε-approximate global min of F in poly time.*
2. *If F is smooth and $\|f - F\|_\infty = P\left(\frac{\varepsilon^2}{(params)}\right)$, then LMC hits ε-approx local min of F in poly time.*

If f is nonconvex, nonsmooth,

1. define $\widetilde{f}_\sigma(x) = \mathbb{E}_z\, f(x+z)$ for $z \sim N(0, \sigma^2 I)$
2. Run LMC on $\widetilde{f}_\sigma$,
$$\nabla \widetilde{f}_\sigma(x) = \mathop{\mathbb{E}}_z\left[\frac{z}{\sigma^2}(f(x+z) - f(x))\right]$$
(only depends on values of f).

For any $\sigma > 0$, $\widetilde{f}_\sigma$ smooth. We can choose σ small enough so $\left\|f - \widetilde{f}_\sigma\right\|_\infty = O\left(\frac{1}{\xi}\right)$, which implies $C_{(\xi f)}(K \backslash U) \approx C_{(\xi f_\sigma)}(K \backslash U)$.

I care less about computational, more about sample complexity. Choice of σ only affects computational complexity.

Do I need to choose large enough σ so that smoothness is similar to original function? No, only need function value to be close.

In small neighborhood of discontinuity, there constant probability of choosing z that makes it jump to the opposite side. Assume jump is $\leq \frac{1}{\xi}$. (Use upper bound on VC dimension.)

Apply to $f(x) = \frac{1}{n}\sum_{i=1}^n l(x; a_i)$ and $F(x) = \mathbb{E}_{a \sim \mathbb{P}}[l(x; a)]$. Under mild conditions $\|f - F\|_\infty \to 0$ as $n \to \infty$. However, ∇, ∇^2 don't converge unless l is smooth, so GD is unreliable.

Can I just do analysis on the smoothed function?

There are 2 notions of samples. I assume I have a set of n samples, and I can take many passes over it, which is relevant to computation cost.

LMC is not necessarily better than smoothed version of gradient descent—you can also eliminate local minimum.

Add noise in heuristic scaling to gradient in neural networks helps. Would be interesting to compare. People claim LMC is better because they didn't push hard enough in other methods. LMC has enough power to escape, but batch normalization can also avoid getting into bad situation. It's hard to say in practice what is the best procedure.

Learning halfspace with 0-1 loss. LMC learns halfspace in polytime for arbitrary noise level. Better than best previous result (Awasthi 2015, learn halfspace in polytime if noise $\leq 10^{-6}$). Here $l(x;(a,y)) = \mathbb{1}(\operatorname{sign}(\langle x, a \rangle) \neq y)$ for $(a,y) \in \mathbb{R}^d \times \{\pm 1\}$. See Theorem 3 in paper, p. 11. Choose $n \geq \widetilde{O}(1)\frac{d^4}{\delta_0^2 \varepsilon^4}$.

(Here the population risk is not smooth. This is a consequence of a more general theorem that doesn't need smoothness on population risk.)

Summary: LMC asymptotically optimal for nonconvex optimization but convergence rates not well understood. We prove hitting inversely depends on restricted Cheeger constant. We lower bound restricted Cheeger constant for convex functions and smooth nonconvex functions. If $\|f - F\|_\infty$ small, then running LMC on f achieves optimal points of F (stability property). LMC is more reliable in GD in empirical risk minimization because it escapes shallow/tiny local min. It is asymptotically consistent too.