Alg-Ml Reading Group

Faster convex optimization with simulated annealing November 5, 2015

Lecturer: Elad Hazan Scribe: Kiran Vodrahalli

Contents

1 Overview

The main point is that there exists a barrier function which is universal in the sense that you can use it for interior point methods for any convex set $\mathcal{K}$. Faster convex optimization gives you $\nu^{1/2}$ versus $n^{1/2}$ and gives you a fasters SDP. This gives you an efficient randomized interior point method for any convex body, which resolves an open question in optimization. They also define the heat path and showed equivalence to the central path in interior point methods. This heat path is also well-defined for non-convex problems so it could be interesting to see what goes on there.

2 Introduction

Today I will be talking about random walk connection with optimization. Here we are relating to methods that were thought to be very distinct: Simulated annealing and interior point methods. I am going to show you they basically do the same thing. This is joint work with Jake Abernethy.

The fundamental problem of convex optimization is to minimize a convex (or linear) function over a convex set with constraints, $\min_{x\in\mathcal{K}} f(x) \to \min_{x\in\mathcal{K}\cap\{f(x)\leq t\}} t$. So without loss of generality you are minimizing a linear function over a convex set. The convex set can be given by linear constraints (LP), semi-definite constraints, a separation oracle, or a membership oracle. The separation oracle is used for large sets, and the most general way is a membership oracle. Given a point, you get to know whether or not it is in the convex set. This talk focuses on the most general case: Convex optimization with a membership oracle.

There are only three polynomial time algorithms: ellipsoid method. It is polynomial time if your approximation is polynomial in $\log(1/\epsilon)$. It is much worse and slow and impractical in the membership oracle model. The other polynomial time algorithm, the most useful one, is interior point methods (Karmarkar, Nesterov-Nemirovski). You also require a barrier. Also, you have random-walk methods (Lavasz-Vempala, Bertsimas-Vempala, Kalai-Vempala). This is the fastest known algorithm for convex optimization in the general model. We want to show that interior

methods and random-walk methods are actually the same thing, and you actually get a faster algorithm where you get a a $\sqrt{n}$ algorithm in the case of semidefinite program, n is the matrix size.

3 Path-following method

Recall that gradient descent is moving in the direction of steepest gradient, and then get projected back to convex set. The folklore theorem says that $f(x_t) \leq f(x^*) + e^{-\mathcal{O}(t)}$: it is polynomial time and converges. The caveat for this method is the projection: We must find the closest point in the set $\mathcal{K}$. This operation is just as difficult or more difficult than the original problem. So it doesn't really help. The ball is easy to project onto, but in other general convex sets it is not clear at all how to do it efficiently. So this setup motivates interior point methods. The idea is to add a super-smooth barrier function. Let us convert our constrained optimization problem to an unconstrained optimization problem to

$$\min_{x \in \mathbb{R}^n} c^T x - \sum_i \log(b_i - A_i x)$$

The main property of the barrier function is that it goes closer to infinity the closer you go to the boundary of the set. Here, $R(x) = \sum_i \log(b_i - A_i x)$. So there are some caveates: gradient descent does not well with high curvature, and barriers give high curvature. Second, the objective is skewed: the barrier distorts the objective. So the first thing we will do is use Newton's method instead of gradient descent, which is more robust to curvature. So this is an easy fix. The second thing to do is control how much distortion the barrier adds: We will scale the objective by t: start at 0 and grow it to infinity:

$$\beta(t) = \min_{x \in \mathbb{R}^n} t * c^T x + R(x)$$

In short I have described the path-following method. We can think of t as a temperature. Iteratively we update t and optimize the new objective. When $t \to \infty$, $\beta(t) \to$solution. Basically you optimize inside ellipse to get a point, then change the ellipse, optimize etc. $\beta(t)$ gives you a path. You do a set of local unconstrained optimization problems and follow the solutions to get the solution.

3.1 The Dikin ellipsoid

Let

$$\|y\|_x = y^T \nabla^2 R(x) y \geq 0$$

Inside the local ellipse, we have a barrier functions. We define the Dikin ellipsoid to be

$$D_1(x) = \{y \text{ such that } \|y - x\|_x \leq 1\}$$

For all x in the set, $D_R(x) \sim \mathcal{K}$. This ellipsoid is always contained inside the set. Inside the Dikin ellipsoid, the function is stronly convex and smooth with respect to the local norm. Also, if you blow up the ellipsoid, it will contain the whole set. Inside the ellipse, you might as well say the Hessian is constant. This gives rise to the following reduction to the gradient method. For well-conditioned functions, we get linear convergence of gradient descent. Inside the Dikin ellipsoid, we are well-conditioned with respect to the local norm, so we converge rapidly inside the ellipse. Then you can change the temperature and repeat.

3.2 Complexity

Each iteration is one iteration of Newton's method and matrix inversion. These are interior point methods. They require this $R(x)$. It is a long-standing question whether we can always construct an efficient universal barrier (we answer yes) - this was previously known only for a few convex sets.

4 Simulated annealing

Simulated annealing is a technique that has been used The idea is to sample with respect to a distribution. This is a heuristic. Here is the Boltzman distribution in general:

$$P_{t,c}(x) = \frac{e^{-f(x)/t}}{\int_{y\in\mathcal{K}} e^{-f(y)/t}dy}$$

For linear functions, replace $f(y)$ with $f(y) = c^T y$. We will consider linear functions. When $t = 0$, we are uniform over $\mathcal{K}$. The idea is to sample down to sample from uniform and then cool down the temperature as you get closer to the optimum. However, as you decrease the temperature, it gets harder to sample. We are minimizing and cooling. The idea is to slowly decrase the temperature, using the previous distribution as a 'warm start'. Then if you have a good estimation of the covariance of the previous distribution, then you can sample decently well. Hit-N-Run is a polynomial time algorithm based on random walks. They sample from this distribution using a sepcial random walk. You sample a random line from $\mathcal{N}(0, \Sigma_{t-1})$ where Σ_{t-1} is the previous covariance matrix. Then you sample with the Boltzmann distribution $P_{t,c}$: If you look at the exponential distribution on this line and sample, then draw a new random line and repeat. We keep updating Σ_t. Vempala et al. prove that it mixes well to get to the Boltzmann distribution. They show as you perform this walk $\mathcal{O}(n^3)$ times, you sample correctly. The only thing that matters is the previous covarianc matrix since things mix quickly.

4.1 The Heat Path

The heat path is the curve of the mean of the Boltzmann distribution. The new observation is that the heat path is the central path for the entropic barrier function. Thus:

$$\mu(t) = \mathbf{E}_{\mathcal{K}:x\sim e^{(c^T x)/t}}[x] \equiv \beta(t) = \min_{x\in\mathbb{R}^n}\{t\cdot c^T x + R(x)\}$$

for $R(x)$ which I will define now. First define

$$A(c) = \log\int_{x\in\mathcal{K}} e^{-c^T x}dx$$

the log partition for the exponential family. Then

$$\nabla A(c) = -\mathbf{E}_{x\sim P_c}[x]$$

$$\nabla^2 A(c) = \mathbf{E}_{x\sim P_c}[(x-\mathbf{E}[x])(x-\mathbf{E}[x])^T]$$

Then the entropic barrier for $\mathcal{K}$ is

$$A^*(x) = \sup_c\{c^T x - A(c)\}$$

We have that as you go towards the barrier, the function goes to infinity and $\nabla^3 A^*(h,h,h) \leq 2(\nabla^2 A^*(h,h))^{3/2}$, where we use $(h,\cdots,h)$ to denote an infinitesmal direction and also $\nabla A^*(h) \leq \sqrt{\nu\nabla^2 A^*(h,h)}$ (in the PSD sense). This can be defined for any convex set, and has been looked at for a long time. ν is the factor by which you need to blow up the set to get the whole set - it is like an isoperimetric constant. Depending on the optimization problem, it depends on the dimension differently. It is optimal in the sense that this barrier gives you $\nu\sim\mathcal{O}(n)$ which cannot be improved. Bubeck-Eldan showed that this is $n+\mathcal{O}(1)$.

4.2 Where can we increase the temperature?

Theorem 4.1. *Temperature schedule suffices to satisfy:* $c_k = t_k * c$.

$$\|P_{c_k} - P_{c_{k+1}}\|_{TV2} = \max\{\|\frac{P_{c_k}}{P_{c_{k+1}}}\|_2, \|\frac{P_{c_{k+1}}}{P_{c_k}}\|_2\} \leq \mathcal{O}(1)$$

The main idea behind this theorem is that it gives the notion of distance that is exactly needed to control the amount of change in the covariance matrix.

We now give our main lemma: For the above, we can have

$$\frac{t_{k+1}}{t_k} = 1 + \frac{\mathcal{O}(1)}{\sqrt{\nu}}$$

If this is the case, then the theorem given by Kalai and Vempala will hold.

Proof. We use the duality of Bregman divergence, which is equivalent to Kullback-Leibler divergence for exponential families. This appears in the book of Jordan and Wainwright. We have

$$KL(P_{c_k}, P_{c_{k+1}}) = D_A(c_k, c_{k+1}) = D_{A^*}(x(c_k), x(c_{k+1}))$$

where A^* denotes the Fenchel dual. These are standard properties of exponential families. Then note by calculation that

$$\log \|\frac{P_{c_{k+1}}}{P_{c_k}}\| = D_A(c_k, c_{k+1})$$

Then we can bound the Bregman divergence by appealing to our intuition about Dikin ellipsoids. We have that this is $\mathcal{O}(1)$. Then the number of Dikin ellipsoids on the path is already bounded by $\nu^{1/2}$ by previous work by Nemirovski, which bounds the total number of temperature updates. □