Lecture 13

Lecturer: Aleksander Mądry *Scribe: Mina Karzand*

1 Overview

In this lecture, we will develop a variant of gradient descent method that works for any norm $\|\cdot\|$, as opposed to only the ℓ_2-norm $\|\cdot\|_2$. We will then apply this method, with the underlying norm being the ℓ_∞-norm, to the maximum flow problem.

2 Recap of Previous Lectures

Over the last several lectures we were working on applying the multiplicative weights update–based framework to the maximum flow problem. The key element there was designing an (θ, ρ)-oracle that, roughly speaking, computes a very crude, "feasible on average" version of maximum flows. First, we based this oracle on shortest path computations, which correspond to ℓ_1-norm minimization, but this did not result in a spectacular running time improvement. Basically, the width ρ of the resulting oracle, which is the main measure of the "quality" of the oracle, was as large as F^* – the value of the maximum flow, making the running time of the final algorithm be

$$\widetilde{O}\left(m\rho\varepsilon^{-2}\right) = \widetilde{O}\left(mF^*\varepsilon^{-2}\right) = \widetilde{O}\left(mn\varepsilon^{-2}\right).$$

(Note that in unit-capacity graphs F^* can be at most n and, in principle, this bound is fairly tight.)

To obtain a better running time, we resorted to an oracle that is based on electrical flow computations, which correspond to ℓ_2-norm minimization. After applying a simple regularization trick, we quickly established that the width ρ of that oracle can be bounded by, roughly, $\sqrt{\frac{m}{\varepsilon}}$. This gave us a running time of

$$\widetilde{O}\left(m\rho\varepsilon^{-2}\right) = \widetilde{O}\left(m^{\frac{3}{2}}\varepsilon^{-\frac{5}{2}}\right),$$

which matched in the case of sparse graphs the best known 40-year-old bounds that were obtained via classic combinatorial techniques. (These techniques were able to deliver an exact, instead of $(1-\varepsilon)$-approximate, solution though.)

Then, we showed that an additional modification of the oracle together with somewhat non-trivial analysis leads to an improved running time bound of

$$\widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{8}{3}}\right),$$

which enabled us to finally break the sparse-graph $\Omega(n^{\frac{3}{2}})$ running time barrier we mentioned above.

3 Gradient Descent Method for General Norms

Encouraged by all the progress we have made so far on the maximum flow problem, it is natural to ask: how far can we go? In particular, is it possible to compute an $(1-\varepsilon)$-approximation to the maximum flow in, essentially best possible, close to linear time?

To answer this question we need to have a re-look on our approaches so far – especially, the approach we outlined in the previous section. At a high level, what was the root of our success there and what were the obstacles for getting even faster algorithms?

Of course, there is multiple valid answers to this question. However, one answer that should be fairly clear by now is: the key was the ability to work with the"right" geometry. That is, each time we got an

improvement it was because we managed to use the geometries we *can* efficiently optimize in, such as, ℓ_1-geometry or ℓ_2-geometry, and make them "look" more like the geometry that we *want* to efficiently optimize in, i.e., the ℓ_∞-geometry that the maximum flow problem corresponds to. In particular, note that we have that, for any m-dimensional vector x,

$$\frac{1}{m}|x|_1 \leq \|x\|_\infty \leq |x|_1,$$

and

$$\frac{1}{\sqrt{m}}\|x\|_2 \leq \|x\|_\infty \leq \|x\|_2.$$

These norm approximation statements explain, in a sense, the basic bounds on the width ρ that we obtained for the ℓ_1-based and ℓ_2-based oracles. This also tells us why we needed to work hard to overcome the $\sqrt{m}$ bound in the latter case.

However, now that we understand that "this is all about geometry", we should try to take a step back and think of making the choice of geometry we work with more principled and explicit, and then adjust our optimization machinery accordingly. Specifically, we want to re-examine our most basic optimization primitive: the gradient descent method and make it work in more general geometries than just the natural ℓ_2-geometry.

To this end, let us consider a general unconstrained minimization problem

$$\min_x f(x),$$

where f is a general convex objective function.

3.1 L-smoothness Revisited

Recall that, at a very high level, the gradient descent method solves the above minimization problem by generating a sequence of points $x_1, \ldots, x_T$, where each x_i is a progressively better minimizer of f. That is, $f(x_1) > \ldots > f(x_T)$.

Thus, the core of the gradient descent method is the generation of the next point x_{i+1} in the sequence from the current point x_i. The general idea here is to look at the linear approximation of the function f around the current point x according to Taylor expansion. Specifically, in the canonical ℓ_2-variant of gradient descent we used the fact that, for any point y,

$$f(y) = f(x) + \nabla f(x)^T(y - x) + O(\|y - x\|_2^2), \tag{1}$$

where the $O(\|y - x\|_2^2)$ is the "error" of our linear approximation of f measured in the ℓ_2-norm.

From (1) we can see that the further point y is from x in the ℓ_2-norm the less accurate this linear approximation is. In order to understand the exact trade-off here, we introduced the notion of smoothness.

Definition 1 (ℓ_2-based) L-smoothness) *We say that a function f is* L-smooth*, for some $L \geq 0$, iff*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2,$$

for all x and y.

L-smoothness gives us bound on how quickly can the gradient $\nabla f(x)$ change wrt to ℓ_2-norm. This bound enabled us to establish the following lemma that gives us a more precise bound on how accurate the linear approximation (1) is.

Lemma 2 *If f is convex and L-smooth then*

$$0 \leq f(y) - f(x) - \nabla f(x)^T(y - x) \leq \frac{L}{2} \cdot \|x - y\|_2^2,$$

for all x and y.

In other words, the above lemma tells us that L-smoothness of f allows us to put a constant of $\frac{L}{2}$ in front of the quadratic ℓ_2-norm error term in (1).

Now, when we transition to a setting in which we have a given general norm $\|\cdot\|$, the key change that drives the variant of the gradient descent method for that norm is that instead of measuring the error of our local linear approximation of f in the ℓ_2-norm as we did in (1), we measure it in the norm $\|\cdot\|$. That is, we want to have that

$$f(y) = f(x) + \nabla f(x)^T(y-x) + O(\|y-x\|^2). \tag{2}$$

Again, given the approximation (2), one might wonder what is the constant in front of the $\|y-x\|^2$ term or, in fact, if we even can have the error be linear in $\|y-x\|^2$. To answer this questions, we need to generalize first the notion of L-smoothness to general norms.

Definition 3 (L-smoothness) *For a general norm $\|\cdot\|$, we say that a function f is* L-smooth *(wrt to that norm), for some $L \geq 0$, iff*

$$\|\nabla f(x) - \nabla f(y)\|^* \leq L\|x-y\|,$$

for all x and y.

The norm $\|\cdot\|^*$ in the above definition denotes the *dual norm* of $\|\cdot\|$. This norm is defined as

$$\|v\|^* = \max_{u \neq \vec{0}} \frac{v^T u}{\|u\|}. \tag{3}$$

It is not hard to see that

(i) $\|\cdot\|_2^* = \|\cdot\|_2$, i.e., ℓ_2-norm is *self-dual*;

(ii) $\|\cdot\|_\infty^* = |\cdot|_1$;

(iii) and $|\cdot|_1^* = \|\cdot\|_\infty$.

More generally, we have that

$$\|\cdot\|_p^* = \|\cdot\|_q,$$

for any p and q such that $\frac{1}{p} + \frac{1}{q} = 1$. Also, for any vector space, it is the case that

$$(\|\cdot\|^*)^* = \|\cdot\|,$$

that is, the dual norm of the dual norm of $\|\cdot\|$ is $\|\cdot\|$ itself.

One of the key motivations behind introducing the dual norm is that it enables us to generalize the Cauchy-Schwarz inequality to arbitrary norms. Namely, we have that

$$v^T u \leq \|v\|^* \|u\|, \tag{4}$$

for any u and v.

Additionally, observe that self-duality of the ℓ_2-norm makes our general definition of L-smoothness (Definition 3) coincide with the ℓ_2-based definition of the L-smoothness (Definition 1), when $\|\cdot\|$ is the ℓ_2-norm. In fact, we can push this correspondence further and show that our generalized version of L-smoothness enables us to obtain a precise bound on the error of our linear approximation (2) of the function f that is analogous to the one we obtained in Lemma 2 for the ℓ_2-based linear approximation (1).

Lemma 4 *Let $\|\cdot\|$ be a general norm and let f be convex and L-smooth (wrt to that norm) then*

$$0 \leq f(y) - f(x) - \nabla f(x)^T(y-x) \leq \frac{L}{2} \cdot \|x-y\|^2,$$

for all x and y.

Observe that we do not visually differentiate between the smoothness parameters L for different norms – we just make sure that the underlying norm is always clear from the context. It is important to keep in mind though that the value of L for a given function f can change drastically depending on the choice of the underlying norm. In particular, a function f that has very bad smoothness wrt one norm can be very smooth wrt another one, and vice versa. This is another evidence of how crucial the choice of "right" geometry to work in can be.

3.2 Taking a Gradient Improvement Step

Taking a closer look at the linear local approximation (2) of f around a given point x and the bound on the error term provided in Lemma 4, we see that our choice of an improvement step has to balance two factors. On one hand, we want to minimize the linear part of the approximation (2). On the other hand, due to the quadratic error of our approximation, we want to keep the length of our step, as measured in the $\|\cdot\|$ norm, not too large.

In the ℓ_2-norm setting, finding the step that guarantees optimal improvement was pretty obvious. The direction in which to move was $-\nabla f(x)$, which is simply the direction of the steepest decrease of our local linearization of f. The magnitude of that step had to be attuned by the step size parameter of $\frac{1}{L}$, which made the improvement step be exactly $-\frac{1}{L}\nabla f(x)$.

When we move to the setting of general norm $\|\cdot\|$, however, the direction of $-\nabla f(x)$ might not be the optimal one to move in, even though it still is the steepest decrease direction of the local linearization of f. This is so as the norm $\|\cdot\|$, in contrast to the the ℓ_2-norm, might not be rotationally invariant and thus have different sensitivity in different directions.

As a result, the best direction to move towards turns out to be $-(\nabla f(x))^{\#}$, where is a certain projection of the $\nabla f(x)$ given by the following optimization problem

$$v^{\#} := \arg\max_u v^T u - \frac{1}{2}\|u\|^2. \tag{5}$$

(Note that, by definition of $v^{\#}$, taking $y = x - (\nabla f(x))^{\#}$ will be exactly the minimizer of our local upper bound on f from Lemma 4 in case of $L = 1$.)

Observe that in the ℓ_2-norm setting, $v^{\#} = v$, as expected. However, in general, this is not the case. In particular, in the ℓ_∞-norm setting we have that

$$v_i^{\#} = \operatorname{sign}(v_i) \cdot |v|_1, \tag{6}$$

for each coordinate i and, in the ℓ_1-norm setting we have that

$$v_i^{\#} = \begin{cases} \operatorname{sign}(v_{i^*}) \cdot \|v\|_\infty & \text{if } i = i^* \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where i^* is a fixed coordinate such that $|v_{i^*}| = \|v\|_\infty$.

Finally, once we know the best direction in which to move, the following lemma specifies the best step size as well as the overall improvement that taking such step guarantees.

Lemma 5 *Let $\|\cdot\|$ be a general norm and let f be convex and L-smooth wrt that norm. For any point x, we have that*

$$f(y) \le f(x) - \frac{1}{2L}\left(\|\nabla f(x)\|^*\right)^2,$$

where $y := x - \frac{1}{L}(\nabla f(x))^{\#}$.

Note that, again, if $\|\cdot\|$ is the ℓ_2-norm then both the optimal step size and the guaranteed improvement coincide with what we obtained in Lecture 3.

Proof By employing the definition of the projection $(\cdot)^{\#}$ (5), it is not hard to see that, for any vector v and any $L > 0$, we have that

$$\frac{v^{\#}}{L} = \arg\max_u v^T u - \frac{L}{2}\|u\|^2.$$

As a result, we have that

$$\begin{aligned} f(y) &= f\left(x - \frac{1}{L}\left(\nabla f(x)\right)^{\#}\right) \\ &\leq f(x) - \nabla f(x)^T\left(\frac{1}{L}\left(\nabla f(x)\right)^{\#}\right) + \frac{L}{2}\left\|\frac{1}{L}\left(\nabla f(x)\right)^{\#}\right\|^2 \\ &\leq f(x) - \left(\max_u \nabla f(x)^T u - \frac{L}{2}\|u\|^2\right), \end{aligned}$$

where the first inequality follows by Lemma 4.

Now, let u^* be such that

$$\frac{\nabla f(x)^T u^*}{\|u^*\|} = \max_u \frac{\nabla f(x)^T u}{\|u\|}.$$

Wlog we can assume that $\|u^*\| = 1$. By definition of the dual norm (3), we have that

$$\frac{\nabla f(x)^T u^*}{\|u^*\|} = \nabla f(x)^T u^* = \|\nabla f(x)\|^*.$$

Observe that if we take $u' := \frac{u^*\|\nabla f(x)\|^*}{L}$ then

$$\nabla f(x)^T u' - \frac{L}{2}\|u'\|^2 = \frac{\left(\|\nabla f(x)\|^*\right)^2}{L} - \frac{1}{2L}\left(\|\nabla f(x)\|^*\right)^2 = \frac{1}{2L}\left(\|\nabla f(x)\|^*\right)^2.$$

Consequently, we have that

$$f(y) \leq f(x) - \left(\max_u \nabla f(x)^T u - \frac{L}{2}\|u\|^2\right) \leq f(x) - \frac{1}{2L}\left(\|\nabla f(x)\|^*\right)^2,$$

which is what we wanted to prove. ■

In the light of the above lemma, we know that the optimal improvement step at point x is $-\frac{1}{L}\left(\nabla f(x)\right)^{\#}$, which gives rise to the general norm variant of the gradient descent method presented as Algorithm 1.

Algorithm 1 General norm variant of the gradient descent method

$x_1 \leftarrow \vec{0}$
for $i = 1, \ldots, T-1$ **do**
  $x_{i+1} \leftarrow x_i - \frac{1}{L}\left(\nabla f(x_i)\right)^{\#}$
return x_T

Once again, one can check that when we consider the ℓ_2-norm setting this general norm variant of gradient descent method becomes the "classic" gradient descent method we developed in Lecture 3.

3.3 Analysis of the General Norm Variant of Gradient Descent Method

Once we developed our general norm variant of gradient descent method, we want to analyze its performance. This is done in the following theorem.

Theorem 6 *Let $\|\cdot\|$ be a general norm and suppose that f is convex and L-smooth wrt to that norm. If x_T is the output of Algorithm 1 then we have that*

$$f(x_T) - f(x^*) \leq O\left(\frac{L \cdot R^2}{T}\right),$$

where x^* is any optimal minimizer of f, $R := \max_{x:f(x)\leq f(x_1)} \|x - x^*(x)\|$, and

$$x^*(x) := \operatorname*{arg\,min}_{x':f(x')=f(x^*)} \|x - x^*\|$$

is the optimal minimizer of f that is closest to x (wrt to $\|\cdot\|$-norm) .

At the first glance, the above theorem seems to be almost identical to the Theorem 5 that we established in Lecture 3 to analyze the performance of ℓ_2-norm based gradient descent method. The only obvious difference is that the radius parameter R here bounds the distance to the closest optimal solution for *all* the points x with $f(x) \leq f(x_1)$, instead of simply bounding the distance between the (closest) optimal solution and x_1.

However, we want to reiterate again that, even though the parameters L and R look the same, they can have very different values for the same function f depending on the underlying norm we are working in. After all, the whole point of developing such a general norm variant of gradient descent method was exactly to take advantage of the ability to choose the "right" norm, i.e., a norm in which the value of L and R for our objective function f is is as small as possible. In a sense, by choosing the "right" geometry we are able to guide the gradient descent method to make better progress on our optimization problem. As we will see shortly, making a good choice here can have dramatic effect on the performance of gradient descent method.[1]

Proof Our course of action here is very similar to how we established Theorem 5 in Lecture 3. We just need to use the right analogues of all the notions. Specifically, let us define

$$\delta_i := f(x_i) - f(x^*(x_i))$$

to be our measure of progress on minimizing f. (Note that $f(x^*(x_i))$ is the same for all i, as all $x^*(x_i)$s are optimal minimizers of f.)

By convexity of f (see Lemma 4 with $y = x^*(x_i)$ and $x = x_i$), we have that

$$\begin{aligned}\delta_i &= f(x_i) - f(x^*(x_i)) \\ &\leq \nabla f(x_i)^T (x_i - x^*(x_i)) \\ &\leq \|\nabla f(x_i)\|^* \|x_i - x^*(x_i)\|\end{aligned}$$

where the last inequality is the Cauchy-Schwarz inequality (4).

On the other hand, by Lemma 5 (with $x = x_i$ and $y = x_{i+1}$), we know that

$$\begin{aligned}\delta_i - \delta_{i+1} &= f(x_i) - f(x_{i+1}) \\ &\geq \frac{1}{2L} \left(\|\nabla f(x_i)\|^*\right)^2 \\ &\geq \frac{1}{2L} \frac{\delta_i^2}{\|x_i - x^*(x_i)\|^2} \geq \frac{\delta_i^2}{2LR^2},\end{aligned}$$

where we used the fact that $f(x_1) \geq f(x_2) \geq \ldots \geq f(x_i)$ and thus, by definition of R, $\|x_i - x^*(x_i)\| \leq R$.

So, we can conclude that, for any i,

$$\delta_i - \delta_{i+1} \geq \frac{\delta_i^2}{2LR^2},$$

which is exactly the same conclusion we had in the proof of Theorem 5 in Lecture 3.

We can thus just repeat our calculations from there to get the desired end result. That is, observe that, for any i,

$$\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i} = \frac{\delta_i - \delta_{i+1}}{\delta_i \delta_{i+1}} \geq \frac{\delta_i - \delta_{i+1}}{\delta_i^2} \geq \frac{1}{2LR^2}.$$

Summing over all $i = 1, \ldots, T-1$, the left-hand side telescopes and we get

$$\frac{1}{\delta_T} - \frac{1}{\delta_1} \geq \frac{T-1}{2LR^2}. \tag{8}$$

[1] A careful reader might notice that we are ignoring here one aspect: in principle, for a general norm $\|\cdot\|$, computing the projections $(\nabla f(x))^{\#}$ wrt that norm might be computationally non-trivial. However, usually, it turns out that there is some simple closed expression formula for computing these projections. This is the case, for example, for the ℓ_∞ and ℓ_1 norms – see (6) and (7).

Now, to bound δ_1, we just use the L-smoothness of f (Lemma 4 with $x = x^*(x_1)$ and $y = x_1$) to obtain that

$$\delta_1 = f(x_1) - f(x^*(x_1)) \leq \nabla f(x^*)^T(x_1 - x^*(x_1)) + \frac{L}{2}\|x_1 - x^*(x_1)\|^2 \leq \frac{LR^2}{2},$$

where the second inequality follows by noting that $\nabla f(x^*(x_1)) = 0$, as $x^*(x_1)$ is an optimal minimizer of f, and that $R \geq \|x_1 - x^*(x_1)\|$. By plugging this back into (8) and rearranging, the proof of Theorem 6 is complete and we can conclude that indeed

$$\delta_T \leq O\left(\frac{LR^2}{T}\right).$$

■

4 Applying an ℓ_∞-variant of Gradient Descent Method to the Maximum Flow Problem

Now that we developed our general norm variant of gradient descent, we want to apply it to the key problem our interest: the maximum flow problem.

Recall that this problem corresponds to solving the following constrained minimization problem.

$$\begin{aligned} \min \quad & \|h\|_\infty \\ \text{s.t.} \quad & Bh = \chi_{st}, \end{aligned}$$

where Bh, applied to any flow vector h, gives us that flow's demand pattern, and χ_{st} encodes the demand pattern of a unit s-t flow.

Given that this problem is an ℓ_∞-minimization problem, it is pretty clear what the "right" norm for that problem is: the ℓ_∞-norm. But before we can apply the machinery from the previous section, we have to deal with two issues. Firstly, the above formulation is a constrained problem, while above we considered only unconstrained problems. Secondly, somewhat counterintuitively, even though our objective function is the ℓ_∞-norm itself, this objective function is *not* L-smooth, for any finite L, wrt that norm.

4.1 Approximate Affine ℓ_∞-projection

The standard way for dealing with constrained optimization problems in the gradient descent method framework is to simply adjust the algorithm by including a projection in the improvement step update that ensures that after each improvement step we land back in the feasible space. This is, in particular, what we did for the ℓ_2-variant of gradient descent before.

However, we will proceed differently here. That is, we still will use a projection operator, but we will put it directly in our objective function instead of applying it in each step of the gradient descent algorithm. Specifically, we will consider the following optimization problem

$$\min_h \|P(h)\|_\infty.$$

Observe that as long as $P(\cdot)$ is a projection onto the space $\mathcal{F}_{st} := \{h \mid Bh = \chi_{st}\}$ of unit s-t-flows, i.e., for any flow h, we have that $B(P(h)) = \chi_{st}$ and $P(h) = h$, if $h \in \mathcal{F}_{st}$, the above optimization problem is equivalent to our original maximum flow formulation.

Introducing the projection $P(\cdot)$ directly in our objective function will have a number of advantages. First of all, our new problem is unconstrained, so we do not need to develop and analyze the projected variant of gradient descent. More importantly though, it will be easier for us to specify and analyze more concrete properties of that projection. In particular, we want $P(\cdot)$ to be an *α-approximate affine ℓ_∞-projection* onto the space of unit s-t-flows $\mathcal{F}_{st}$. That is, we want that

(a) $P(\cdot)$ is a projection onto $\mathcal{F}_{st}$. That is, $B(P(h)) = \chi_{st}$, for any h, and $P(\cdot)$ is an identity on $\mathcal{F}_{st}$, i.e., $P(h) = h$, whenever $h \in \mathcal{F}_{st}$.

(b) $P(\cdot)$ is α-approximate in the ℓ_∞-norm. That is, for any h,

$$\|P(h) - h\|_\infty \leq \alpha \|\Pi(h) - h\|_\infty, \tag{9}$$

where Π is the exact ℓ_∞-projection onto $\mathcal{F}_{st}$, i.e.,

$$\Pi(h) := \arg\min_{g \in \mathcal{F}_{st}} \|g - h\|_\infty. \tag{10}$$

(c) $P(\cdot)$ is affine. That is, for any h, $P(h)$ is of the form

$$P(h) = \widehat{P}h + \hat{h}, \tag{11}$$

where $\widehat{P}$ is a linear operator (a matrix) and $\hat{h}$ is some fixed flow.

One reasons why we allow $P(\cdot)$ to be only α-approximate, with $\alpha > 1$, instead of being exact is that computing an exact ℓ_∞-projection Π is not easier than solving the maximum flow problem itself. After all, by definition of Π (10), $\Pi(0)$, i.e., applying Π to a trivial all-zero flow, gives exactly the maximum unit s-t-flow.

Also, there is another reason why it is actually necessary to allow $\alpha > 1$. It is not hard to show that the exact ℓ_∞-projection Π *cannot* be affine. So, insisting on $P(\cdot)$ being affine requires $\alpha > 1$.

Finally, observe that requiring $P(\cdot)$ implies that in (11) $\widehat{P}$ is a linear projection onto the space of circulations, i.e., flows h with $Bh = 0$. Also, $\hat{h}$ has to be a fixed unit s-t-flow, equal the projection $P(0)$ of a trivial all-zero flow.

4.2 Applying ℓ_∞-Smoothing

At this point, we are already working with an unconstrained minimization problem, but we still need to deal with the fact that our objective function computes an ℓ_∞-norm, which is not L-smooth.

Fortunately, we already know how to proceed here. Similarly as we did when applying the ℓ_2-variant of gradient descent method to the maximum flow problem, we introduce a smoothing of the ℓ_∞-norm, defined as

$$\operatorname{smax}_\delta(h) := \delta \ln \left(\frac{\sum_e e^{\frac{h_e}{\delta}} + e^{-\frac{h_e}{delta}}}{2m} \right).$$

Recall that in Lecture 4 we proved that $\operatorname{smax}_\delta$ approximates the ℓ_∞-norm fairly well.

Lemma 7 *For any flow h, we have $\|h\|_\infty - \delta \ln(2m) \leq \operatorname{smax}_\delta(h) \leq \|h\|_\infty$.*

So, the smaller the δ the tighter the approximation is. On the other hand, one can show that, similarly to the case of ℓ_2-smoothness, the ℓ_∞-smoothness of smax is inversely proportional to δ.

Lemma 8 *For any $\delta > 0$, the function $\operatorname{smax}_\delta$ is convex and $\frac{1}{\delta}$-smooth wrt ℓ_∞-norm.*

As a result, our optimization problem becomes

$$\min_h \ \operatorname{smax}_\delta(P(h)), \tag{12}$$

and, once again, it will be crucial for us to choose a value of δ that gives us the best trade-off between the smoothness of $\operatorname{smax}_\delta$ and the approximation quality it offers.

4.3 Putting Everything Together

We are now ready to apply the ℓ_∞-variant of gradient descent method to our smoothed unconstrained minimization version of the maximum flow problem (12). Observe that, by Theorem 6, if we apply Algorithm 1 to (12) then, after T iterations, we will obtain a flow x_T such that

$$\operatorname{smax}_\delta(P(x_T)) - \operatorname{smax}_\delta(P(x^*)) \leq O\left(\frac{LR^2}{T}\right), \tag{13}$$

where

$$x^* := \arg\min_x \operatorname{smax}_\delta(P(x)).$$

Note that if h^* is the (scaled) maximum s-t flow, i.e., h^* is unit s-t flow with $\|h^*\|_\infty = \frac{1}{F^*}$, where F^* is the maximum s-t flow value, then, by definition of x^*,

$$\operatorname{smax}_\delta(P(x^*)) \leq \operatorname{smax}_\delta(P(h^*)) \leq \|h^*\|_\infty = \frac{1}{F^*},$$

where the last inequality follows by the fact that $P(\cdot)$ is an identity on unit s-t flows (see property (a) of $P(\cdot)$ above) and Lemma 7.

As a result, plugging the above back into (13), we obtain that if we take $h_T := P(x_T)$ then h_T will be a unit s-t flow and

$$\|h_T\|_\infty \leq \operatorname{smax}_\delta(h_T) + \delta \ln 2m \leq \operatorname{smax}_\delta(P(x^*)) + O\left(\frac{LR^2}{T}\right) + \delta \ln 2m \leq \frac{1}{F^*} + O\left(\frac{LR^2}{T}\right) + \delta \ln 2m, \tag{14}$$

where the first inequality follows by Lemma 7.

In the light of this bound, we just need to estimate the values of L and R as well as choose the values of δ and T so as to ensure that

$$O\left(\frac{LR^2}{T}\right) + \delta \ln 2m \leq \frac{\varepsilon}{2F^*}. \tag{15}$$

Once this is achieved, by (14), we will have that

$$\|h_T\|_\infty \leq \frac{1}{F^*} + O\left(\frac{LR^2}{T}\right) + \delta \ln 2m \leq \frac{1}{F^*} + \frac{\varepsilon}{2F^*} \leq \frac{1}{(1-\varepsilon)F^*},$$

for $\varepsilon \leq \frac{1}{2}$. That is, h_T will be the desired (scaled) $(1-\varepsilon)$-approximate maximum s-t flow.

It is not hard to show that $L \leq \frac{\alpha^2}{\delta}$. That is, we have the following lemma.

Lemma 9 *The function* $\operatorname{smax}_\delta(P(h))$ *is* $\frac{\alpha^2}{\delta}$*-smooth wrt* ℓ_∞*-norm.*

Obtaining bound on R is slightly more involved and it is presented in the following lemma whose proof appears in the appendix.

Lemma 10 *Let* $x_1 = 0$ *and let* $X := \{x \mid \operatorname{smax}_\delta(P(x)) \leq \operatorname{smax}_\delta(P(x_1))\}$ *then, we have that*

$$R := \max_{x \in X} \|x - x^*(x)\|_\infty \leq \frac{(2\alpha+1)}{F^*} + 2\delta \ln 2m,$$

where $x^*(x)$ *is the optimal minimizer of* $\operatorname{smax}_\delta(P(\cdot))$ *that is closest to the point* x *in the* ℓ_∞*-norm.*

Note that, as always, R depends on the choice of our starting point x_1. However, simply taking $x_1 = 0$ turns out to be sufficient.

Also, observe that the obtained bound on R is very small and, in particular, it is proportional to the value of the optimal solution! Recall that in Lecture 4, when we analyzed the ℓ_2-norm diameter the best bound we were able to establish was only $\sqrt{n}$. So, choosing the "right" geometry led to a striking improvement.

Now, taking $\delta := \frac{\varepsilon}{4F^* \ln 2m}$ and using the bounds on L and R that we established in Lemmas 9 and 10, we get that

$$O\left(\frac{LR^2}{T}\right) + \delta \ln 2m \leq O\left(\frac{\alpha^2 F^* \ln 2m \left(\frac{(2\alpha+1)}{F^*} + \frac{\varepsilon}{2F^*}\right)^2}{\varepsilon T}\right) + \frac{\varepsilon}{4F^*} \leq O\left(\frac{\alpha^4 \ln 2m}{\varepsilon T F^*}\right) + \frac{\varepsilon}{4F^*}.$$

Therefore, setting $T := C\frac{\alpha^4 \ln 2m}{\varepsilon^2}$, for a large enough constant C, allows us to satisfy the condition (15). We can thus conclude with the following theorem.

Theorem 11 *For any $0 < \varepsilon \leq \frac{1}{2}$, we can compute an $(1-\varepsilon)$-approximation to the maximum flow problem in time*

$$O\left(\frac{\alpha(P)^4 \ln m}{\varepsilon^{-2}} \left(\tau(P) + m\right)\right),$$

where $\alpha(P)$ is the quality of the affine ℓ_∞-projection P and $\tau(P)$ is the time needed to compute it.

The above theorem is quite remarkable. It allows us to reduce the task of fast computation of an $(1-\varepsilon)$-approximate maximum flow to computing quickly a relatively small number of very crude maximum flow solutions. Basically, ignoring the affinity requirement, computing the projection P can be thought of as solving a maximum flow problem (wrt a given demand vector) up to an approximation $\alpha(P)$. So, if we are aiming for an $(1-\varepsilon)$-approximation maximum flow algorithm that runs in close to linear time, i.e., has its running time be $O(m^{1+o(1)}\varepsilon^{-2})$, then all we have to do is just to construct P that has $\alpha(P) \leq n^{o(1)}$, which is a very loose approximation bound, and can be computed in time $O(m^{1+o(1)})$. Sounds like a much easier task than solving the original $(1-\varepsilon)$-approximation problem from a scratch!

This bootstrapping phenomena, i.e., the ability to leverage a fast algorithm solving a very crude version of our problem to get a fast algorithm for solving that problem up to the desired approximation, is really powerful. We already have seen its power in the context of multiplicative weight update–based framework for solving feasibility questions that we developed in Lecture 10. Now, we see this phenomena in play again.

5 Appendix: Proof of Lemma 10

Let us fix any $\hat{x} \in X$. We will first show that

$$\|P(\hat{x}) - x^*(\hat{x})\|_\infty \leq \frac{(\alpha+1)}{F^*} + 2\delta \ln 2m. \tag{16}$$

To this end, observe that by triangle inequality, we have that

$$\|P(\hat{x}) - x^*(\hat{x})\|_\infty \quad \leq \quad \|P(\hat{x}) - P(x^*(\hat{x}))\|_\infty \leq \|P(\hat{x})\|_\infty + \|P(x^*(\hat{x}))\|_\infty, \tag{17}$$

where the first inequality follows by noticing that if x^* is a minimizer of $\mathrm{smax}_\delta(P(x))$ then so is $P(x^*)$. (Recall that for any projection we have that $P^2(x) = P(x)$.)

By Lemma 7 and the fact that $\hat{x} \in X$, we get that

$$\begin{aligned} \|P(\hat{x})\|_\infty &\leq \mathrm{smax}_\delta(P(\hat{x})) + \delta \ln 2m \leq \mathrm{smax}_\delta(P(x_1)) + \delta \ln 2m \\ &\leq \|P(x_1)\|_\infty + \delta \ln 2m = \|P(0)\|_\infty + \delta \ln 2m \\ &= \|P(0) - 0\|_\infty + \delta \ln 2m \leq \alpha \|\Pi(0) - 0\|_\infty + \delta \ln 2m \\ &= \alpha \|\Pi(0)\|_\infty + \delta \ln 2m = \alpha \|h^*\|_\infty + \delta \ln 2m = \frac{\alpha}{F^*} + \delta \ln 2m, \end{aligned} \tag{18}$$

where we also used (9), the fact that, by definition (10) of Π, $\Pi(0)$ is exactly the (scaled) maximum s-t flow h^*.

Furthermore, by Lemma 7 and the fact that $x^*(P(\hat{x}))$ is a minimizer of $\text{smax}_\delta(P(x))$, we have that

$$\begin{aligned} \|P(x^*(\hat{x}))\|_\infty &\leq \text{smax}_\delta(P(x^*(\hat{x}))) + \delta \ln 2m \leq \min_x \text{smax}_\delta(P(x)) + \delta \ln 2m \\ &\leq \min_x \|P(x)\|_\infty + \delta \ln 2m \leq \|P(h^*)\|_\infty + \delta \ln 2m = \frac{1}{F^*} + \delta \ln 2m, \end{aligned} \tag{19}$$

where we used the fact that $P(h^*) = h^*$ as h^* is a unit s-t-flow. Plugging (18) and (19) into (17), gives us (16).

Now, to see that (16) implies our lemma, note that, if x^* is an optimal minimizer of $\text{smax}_\delta(P(x))$ then the point $\hat{x}^* := P(x^*) + \hat{x} - \widehat{P}(\hat{x})$ is also an optimal minimizer of that function. To see that, observe that the fact that $P(\cdot)$ is an affine projection (see (11)) implies that

$$P(\hat{x}^*) = \widehat{P}\hat{x}^* + \hat{h} = \widehat{P}P(x^*) + \widehat{P}\hat{x} - \widehat{P}^2(\hat{x}) + \hat{h} = \widehat{P}P(x^*) + \hat{h} = P^2(x^*) = P(x^*),$$

where we used the fact that $\widehat{P}^2 = \widehat{P}$, since $\widehat{P}$ is a projection too. Thus, we have that

$$\text{smax}_\delta(P(\hat{x}^*)) = \text{smax}_\delta(P(x^*)),$$

and thus $\hat{x}^*$ is indeed an optimal minimizer of $\text{smax}_\delta(P(x))$ as well.

Consequently, by (16) and triangle inequality, we have that

$$\begin{aligned} \|\hat{x} - x^*(\hat{x})\|_\infty &\leq \|\hat{x} - (P(x^*(\hat{x})) + \hat{x} - \widehat{P}(\hat{x}))\|_\infty = \|\widehat{P}\hat{x} - P(x^*(\hat{x}))\|_\infty \\ &= \|P(\hat{x}) - P(x^*(\hat{x})) - \hat{h}\|_\infty \\ &\leq \|P(\hat{x}) - P(x^*(\hat{x}))\|_\infty + \|\hat{h}\|_\infty \\ &\leq \frac{(\alpha+1)}{F^*} + 2\delta \ln 2m + \|P(0)\|_\infty \leq \frac{(2\alpha+1)}{F^*} + 2\delta \ln 2m, \end{aligned}$$

where the last two inequalities follow by noticing that $\hat{h} = P(0)$ and that $\|P(0)\|_\infty \leq \frac{\alpha}{F^*}$ (see (18)). Lemma 10 follows.