

Distributed Optimization and Machine Learning

Prof. Mayank Baranwal

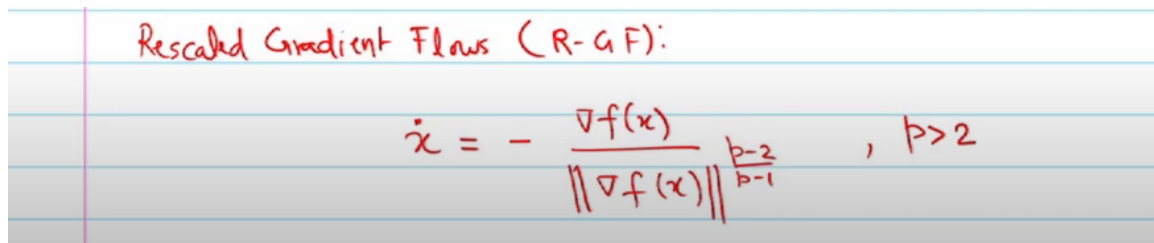
Computer Science & Engineering, Electrical Engineering, Mathematics

Indian Institute of Technology Bombay

Week-5

Lecture - 18: Rescaled Gradient Flow

So, in this same paper by Michael Jordan, they also introduce something called rescaled gradient flow, rescaled gradient flows. And as I said gradient normalization is important. So, this rescale gradient flow was actually scaling the gradient. So, the dynamical system looked something like this. So, usually for simple gradient flow we would have \dot{x} what is negative of gradient of f . So, in rescale gradient flow this turns out to be this divided by the norm of the gradient to the power p minus 2 over p minus 1 with p greater than equal to, strictly greater than 2.



Rescaled Gradient Flows (R-GF):

$$\dot{x} = - \frac{\nabla f(x)}{\|\nabla f(x)\|^{p-1}}^{p-2}, p > 2$$

So, first of all is the dynamical system clear to everyone? So, this dynamical system has the same equilibrium as the simple gradient flow. x equal to x^* , I mean again in this case, in both cases it vanishes at x equal to x^* . So, it has the same equilibrium as simple gradient flow. not just that, the trajectory followed by x is exactly the same as the trajectory followed by.

So, this is just like rescaling the flow, like how quickly you traverse the trajectory, but the trajectory stays the same. So, x as a function of time changes, but the phase portrait of x that remains the same. So, you are just traversing the trajectory much faster than simple gradient flow. it will be exactly the same in continuous time. The moment you discretize then it I mean you cannot see.

So, in fact if I just normalize it with respect to the norm of the gradient that becomes a unit velocity vector field. So, that means the velocity changes, but it that I mean the trajectory that states the same ok. So, for this particular rescale gradient flow. So, Michael

Jordan's group they showed that these convergence rate is order 1 over t to the p. And if you choose p to be very large, you can practically make it converge much faster, right.

Convergence rate is $O\left(\frac{1}{t^p}\right)$.

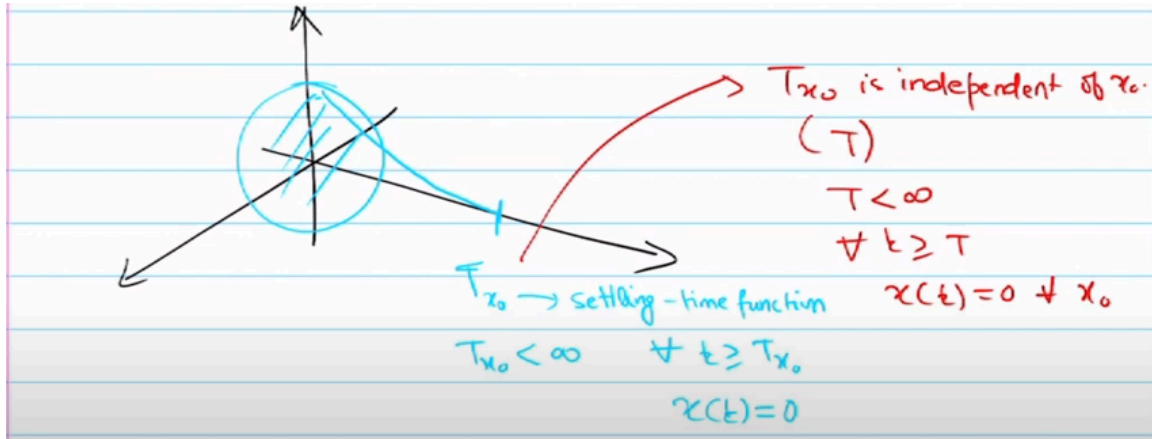
So, that is the idea. But in order to prove this result, again they use some like the Bregman divergence. So, we are going to use Lyapunov theory and we are going to show something remarkable here. We are in fact going to show that this converges in finite time. The same rescaled gradient flow, not just 1 over t to the p.

In a finite time interval T, it will converge to the equilibrium. So, is the setup clear to everyone? So, this is R. So, let me denote this by Rgf. This p greater than 2 is important and we will also look at why it is important. But this essentially sort of is a good departure for us to look at something called finite time or fixed time stability.

It is holder continuous at 0, I mean as in like at x star essentially, I mean it is holder continuous everywhere else it is smooth. It is holder continuous at x star, but it is not non-smooth, non-smooth, it is holder continuous that way. So, usually when you define this kind of dynamical system, you would say that when gradient of f is greater than 0, norm is greater than 0, then you run this, otherwise x dot is equal to 0. So, that is how you can typically define and when in practice when you implement this, you add a small epsilon in the denominator. So, that there is numerical stability, when you implement this in discrete time, but here So this gives rise to two concepts, one is finite time stability and the other is fixed time stability of equilibrium points.

So finite time stability, how many of you know Sanjay Bhatt or heard of Sanjay Bhatt? So he was a faculty here till 2007 in aerospace. He is now at TCS research, but finite time stability was a seminal contribution during his PhD thesis in 98 I think. So, and this fixed time stability is a generalization of finite time stability and this was proposed by Polyakov very recently in fact in 2012. And the idea is with finite time stability, so if I consider the same setup here. So, you start in some ball around the let us say equilibrium.

So, with finite time stability, you are guaranteed to converge to the equilibrium in a finite amount of time T. But this time T is going to be dependent on your initial condition x naught. So, this T is called settling time or settling time function because it is a function of x naught. So, settling time function. and T x naught is in fact finite.



So, strictly and what do we have? So, if origin is the equilibrium, then for every t greater than equal to T_{x_0} , we have that norm of or not just norm, $x(t)$ is equal to 0 for all t greater than this particular time, if origin is equilibrium. I mean if x^* is equilibrium then it would be $x(t) = x^*$ forever. So, what is fixed time then? So, Polyakov generalized this result and showed that with fixed time convergence. So, T_{x_0} is independent of x_0 . So, that means you get initialization independent bound on how quickly you can converge.

So let us call this capital T now because it is independent of x_0 initial condition and Polyakov showed that there exists a fixed time T where independent of the initial condition for every t greater than equal to T , $x(t)$ is equal to 0 for all x_0 . So no matter where you start, you are guaranteed to converge to the optimal solution or in this case the equilibrium in a fixed amount of time. In the finite time stability case, you are guaranteed to converge again in a finite amount of time, but that amount of time that time depends on the initial condition. So, if you are farther from the equilibrium you may require larger time to converge, but that is not the case with fixed time stability where you even if you are farther away from the equilibrium you would still converge to the equilibrium in a fixed amount of time. So, how does these results first of all have a Lyapunov characterization? So, let us look at Lyapunov characterization of finite time stability to get a sense of what this finite time stability is about.

So, we are now going to look at Lyapunov characterization of finite time stability. So, remember when we had $\dot{V} \leq -\alpha V$ let us say $\dot{V} \leq -\alpha V$ is less than equal to negative αV . So, we said that V converges exponentially fast, but x we cannot say anything about x right. So, this talks about exponential convergence of V . So, let us look at a different, let us say if I arrive at a different kind of inequality and I say $\dot{V} \leq -\alpha V^\beta$, $\alpha > 0$, $\beta < 1$.

* Lyapunov characterization of finite-time stability:

$$\dot{V} \leq -V \quad (\text{Exponential convergence of } V)$$

$$\dot{V} \leq -V^\alpha, \quad \alpha < 1$$

$$\int_{V_0}^{V(t)} \frac{dV}{V^\alpha} \leq - \int_0^t dt$$

$$\frac{1}{1-\alpha} [V(t)^{1-\alpha} - V_0^{1-\alpha}] \leq -t$$

So, let us say we arrive at this kind of inequality on time derivative. So, let us see what this, what does this mean? So, that is what is the integral of this term 1 over 1 minus α which basically gives V raised to the 1 minus α is less than equal to V naught 1 minus α minus. So, V is a Lyapunov function, what do we know about the Lyapunov function? Positive definite positive definite positive semi definite whatever, but at least we know that this is always greater than equal to 0 right. So, for this to be greater than equal to 0 , this is an increase like, so this is a constant here. As t keeps on increasing, there would be a point when this term is equal to this term, right.

So, that means this is the point where your Lyapunov function touches 0 , right. Because Lyapunov function. Yeah, but x is a function of t . So, I mean you can have an explicit function of time or you need not even have an explicit function of time. The point is that this particular inequality implies that there would be some point after which this right hand side starts becoming negative right.

And because your Lyapunov function cannot be negative, so this is going to be valid only for time t . So, this T let us say this there exists some time let us call this T_x naught which is a finite time such that V naught 1 minus α is equal to 1 minus α times T_x naught right or T_x naught is equal to So your x naught kind of shows up in your V naught definition. At the point where you start, you get the initial condition for the Lyapunov function or initial value. So that is why you see the dependencies on the initial condition. But we know that for all t greater than equal to t_x naught, first of all your Lyapunov function is going to be 0 .

That means you are going to be at the equilibrium and you are guaranteed to arrive at the equilibrium in time t less than capital T_x , less than equal to capital T_x naught. That

means you have converged in a finite amount of time, if you can show that there exists a Lyapunov function which satisfies this inequality. So, in fact this is the characterization of finite time stability of equilibrium. So, this is the inequality that your Lyapunov function need to satisfy. So, your V of x greater than equal to 0 is strictly greater than 0 when x is not equal to x^* .

and \dot{V} is less than equal to negative V raised to the α with α less than 1 that is the condition for finite-time stability. And not just that this α plays a role why because you can also characterize the settling time function in terms of this right. So, your settling time function or you can rather say that your settling time is also upper bounded by V naught 1 or let me just write it V of x naught 1 minus α ok. So, this is true if I mean if x^* is finite time stable then this is true right. So, this basically gives us idea as to well I mean the choice of Lyapunov function is fine, but this basically tells us how to design this these kind of modified gradient flows.

$$V(t)^{1-\alpha} \leq V_0^{1-\alpha} - (1-\alpha)t$$

$$\exists \text{ some time } T_{x_0} < \infty \text{ s.t. } V_0^{1-\alpha} = (1-\alpha)T_{x_0}$$

$$T_{x_0} = \frac{V_0^{1-\alpha}}{1-\alpha}$$

$$\forall t \geq T_{x_0} \quad V(t) = 0$$

Settling-time function, $T_{x_0} \leq \frac{V(x_0)^{1-\alpha}}{1-\alpha}$

So, now let us try to reanalyze this rescaled gradient flow, but instead of using Bregman divergence as was done in the original paper, we will try to use this particular result on finite time stability of a finite time stability right. So, let us do that. No, as long as V is continuous the settling time because it is I mean you get a upper bound directly in terms of V naught right. So, settling time function is going to be continuous for finite-time stability, for fixed-time stability anyway it is independent of the initial condition. So, the question of continuity does not even show up there.

So, we have the rescale gradient flow here right. Now let us choose the I mean as I said like the choice of Lyapunov function that does not change by much. So, we assume f is μ strongly convex. So, V I can choose to be half that does not change by much that does

not change at all. So V dot turns out to be gradient f transpose h in f times x dot and that is now we are going to be using the RGF or the rescale gradient flow for x dot right.

* Analyzing RGF using Lyapunov Theory:

Assume f is μ -SC:

$$V = \frac{1}{2} \|\nabla f\|^2$$

$$\dot{V} = (\nabla f)^T \nabla^2 f \nabla f$$

$$= -\frac{(\nabla f)^T \nabla^2 f (\nabla f)}{\|\nabla f\|^{\frac{p-2}{p-1}}} \quad , p > 2 \quad (\text{From RGF!})$$

So this becomes p greater 2 ok, just from rgf scale gradient flow. Now, because this is strongly convex, f is strongly convex, Hessian of f is lower bounded by μ times identity. So, I can write this as negative μ divided by the same term Is everyone with me on this? Why? Because since hessian of f is from a strong convexity and this is basically equal to negative μ gradient of f times 2 minus p minus 2 minus 1 which is equal to negative μ . ok which again I am not done because I need to write V dot in terms of V right less than equal to some V raise to the alpha and V by definition is this particular thing.

$$\leq -\frac{\|\nabla f\|^2 \mu}{\|\nabla f\|^{\frac{p-2}{p-1}}} \quad [\because \nabla^2 f \geq \mu I]$$

$$= -\mu \|\nabla f\|^{2 - \frac{p-2}{p-1}}$$

$$= -\mu \|\nabla f\|^{\frac{p}{p-1}} \geq -\mu \|\nabla f\|^2 \cdot \frac{p}{2(p-1)}$$

$$= -\mu (2V)^{\frac{p}{p-1}}$$

So, what I do is minus μ ok. which is minus μ . Now, if p is greater than 2, what is the value of p over 2 times p minus 1? Let us say p is equal to 2.5. So, this term is always less than 1. So, for p greater than 2 by the way this alpha is not just less than 1, it is also supposed to be greater than 0.

So, which means we have shown that \dot{V} is less than equal to $-\mu$ and since this term is less than 1, x^* is finite-time stable. And what is the settling time function? T of x naught turns out to be V of x naught raised to the power $1 - \alpha$ where α is, so choose α is equal to $1 - \mu$ divided by this constant term here which is μ times 2 to the α times $1 - \alpha$. This is your settling time function. So, not only you show that this converges in a finite amount of time, it converges in a finite amount of time which you can also characterize.

For $p > 2$, $0 < \frac{\mu}{2^{p-1}} < 1$

$$\dot{V} \leq -\mu \frac{V^{\frac{p}{2^{p-1}}}}{2^{p-1}} \quad ; \quad \alpha = \frac{\mu}{2^{p-1}}$$

$\Rightarrow x^*$ is finite-time stable,

$$T_{x_0} = \frac{V(x_0)^{1-\alpha}}{\mu 2^{\alpha(1-\alpha)}}$$

So, remember in the earlier case of exponential stability, we could only say that V converges exponentially fast, but x need not converge exponentially fast. But why can we say, because right now this inequality is in terms of V and not in terms of x . Why can we say that x converges in finite time? Because this says that V converges in finite time. How can we conclude that x also converges in finite time from here? Because V is 0 only when x is equal to 0. So, because V has converged in finite time, that means V has converged exactly to 0.

V of x^* is exactly equal to 0. So, that means x has converged to x^* in a finite time, whereas that was not the case earlier. When with exponential stability, I mean we still talk about V converging to 0, V converging to 0 exponentially fast, but V can be an arbitrary function of x right and that may, that may not converge exponentially fast. In this case even if it is an arbitrary function because V of t becomes exactly equal to 0 so that means x of t is also exactly equal to 0 or x^* . Is this clear? So, the same result that was shown in Jordan's work to be order like order 1 over t to the p kind of convergence. You in fact using the simple Lyapunov theory you can get finite time convergence right.

So, it is a much bigger result than like. So, in order to show that this converges arbitrary fast like using their analysis you would have to choose where p which is very large right. So, that this particular term becomes very small. So, you get very fast convergence. But if you use a similar use for the same dynamical system, if you use different tools, you can

show that it is actually much better. I mean you can in fact guarantee convergence in a finite amount of time.

So, how does this result extend to like how can we generalize this to fixed time stability case. So for finite-time stability we knew that \dot{V} is less than or equal to some $c_1 V$ where c_1 is greater than 0 and α_1 is a number between 0 and 1. With this you can guarantee finite time stability right of the equilibrium. If you want to guarantee fixed time stability you would have to add another term to it with α_2 greater than 1 and c_2 greater than 0. So if that is the case, then actually you can show that this is fixed time stable, not just finite time stable, the equilibrium is fixed time stable.

* Fixed-time stability:

$$\dot{V} \leq -c_1 V^{\alpha_1} - c_2 V^{\alpha_2}$$

Equilibrium is fixed-time stable

$c_1 > 0, \alpha_1 \in (0, 1)$
 $c_2 > 0, \alpha_2 > 1$

Settling-time $\rightarrow T \leq \frac{1}{c_1(1-\alpha_1)} + \frac{1}{c_2(\alpha_2-1)}$

So that means no matter where you start you are guaranteed to converge to the equilibrium in a fixed amount of time and that settling time function t it is actually now it is not a function of x naught right it is independent of x naught that is actually upper bounded by $\frac{1}{c_1(1-\alpha_1)} + \frac{1}{c_2(\alpha_2-1)}$ this is the settling time. So, what is the difference? So, how is this particular thing able to guarantee finite time convergence versus why is the other term at like intuitively why is this the other term able to guarantee fixed time convergence? So, let us see what happens when we look at this particular thing right. So, when V is very large or let us say when V is the norm of V or the value of V is actually less than equal to 1. by choosing like by exponentiating it to α , which is a number between 0 and 1, you are making it bigger, right? You are making it bigger. So even when the rate of decrease is actually getting smaller and smaller, you are sort of scaling this up using V raise to the α .

So the moment you hit that unity ball, V equal to 1, right? After that, unlike the exponential convergence case, you are actually scaling things up and that is actually leading to faster convergence. But we do not take care of what happens outside that unity bound with this. In fact, you are sort of slowing it somewhat. But we still get finite time

convergence because we know that the convergence is kind of slow only closer to the optimal solution. So, there you are scaling things up and that basically helps you with faster convergence or in this case finite time convergence.

In case of this particular condition. You have one term to scale things up when you are inside the unit ball, but because α^2 is greater than 1, you actually scale things up when you are outside that unit ball. So, you converge to the unit ball faster first of all and then once you converge to the unit ball, then you converge to the optimal solution or the equilibrium faster because of this term. So, these two different terms, they actually have a role in different regimes. So, if you start somewhere like farther away from the equilibrium, you would have this term basically contributing towards the decrease of the Lyapunov function. The moment you hit the unit ball, this term starts dominating the other term and that is when you would have, you would see that this term would guarantee finite.

So, you converge to the unit ball in a finite amount of time. Once you are in the unit ball, you again converge in a finite amount of time. So, in summary, you will basically converge in a fixed amount of time independent of the initialization. Now, how does this translate to designing the algorithm? So far, before the rescale gradient flow, we were analyzing the algorithms that were already known to us, be it gradient flow or Newton's method and things like that. Rescale gradient flow was a way to somehow engineer a different kind of gradient flow that can show faster convergence.

So, let us look at another rescale gradient flow or another gradient flow similar to rescale gradient flow. But this time our objective is to be able to guarantee fixed time convergence right and not just finite time. So, as I said we did not change much in terms of the Lyapunov function right. Lyapunov function was still the same. So, either you change the algorithm or you change your Lyapunov function so that it satisfies the inequality right.

So, we are going to keep the Lyapunov function as it is, but then we need to be designing our algorithm differently. and now we are going to be using. So, earlier this was the algorithm ok. Now, I am adding another term to it why because I want to get that kind of result with my Lyapunov function V with p greater than 2 and Q is a number between 1 and 2. So, without this second term, we had already shown finite time stability or finite time convergence.

* Another GF (similar to RGF):

$$\dot{x} = - \frac{\nabla f}{\|\nabla f\|^{\frac{p-2}{p-1}}} - \frac{\nabla f}{\|\nabla f\|^{\frac{q-2}{q-1}}}$$

$p > 2$ and $q \in (1, 2)$

$$= -\nabla f \left(\frac{1}{\|\nabla f\|^{\frac{p-2}{p-1}}} + \frac{1}{\|\nabla f\|^{\frac{q-2}{q-1}}} \right)$$

With this other term, the reason we added this other term is because we want to get this kind of inequality on the Lyapunov function and with just one term that is not possible, right. So, that also gives you an idea. Just by looking at what you want to prove or what kind of results or tools that you have, it also gives you an idea as to how you can modify your existing, existing dynamical system, And if I really look at this dynamical system from the perspective of gradient normalization that is nothing but and you normalize this by p minus 2 or p minus 1. So, this is the normalization factor here in the bracket and what you are basically running is again a gradient descent, but you are just normalizing the gradient.

So, let us see how this guarantees fixed time convergence. So, as I said we are going to be choosing the same Lyapunov function. Again we are going to be assuming that f is strongly convex. So, \dot{V} and \dot{x} we are now going to substitute this modified gradient flow.

$$V = \frac{1}{2} \|\nabla f\|^2$$

$$\dot{V} = (\nabla f)^T \nabla^2 f \dot{x}$$

$$= - \frac{(\nabla f)^T \nabla^2 f \nabla f}{\|\nabla f\|^{\frac{p-2}{p-1}}} - \frac{(\nabla f)^T \nabla^2 f \nabla f}{\|\nabla f\|^{\frac{q-2}{q-1}}}$$

$$\leq -\mu(2V)^{\frac{p}{2(p-1)}} - \mu(2V)^{\frac{q}{2(q-1)}}$$

ok. And this is less than equal to minus μ . if you just following the same kind of analysis it is exactly in fact the same, you get $2V^{\frac{p}{2(p-1)}} - \mu 2V^{\frac{q}{2(q-1)}}$

over $2q - 1$ that is using the similar analysis that we followed in for finite time stability that is what we get. So, for the choice of p , so when p is greater than 2, it is basically a number between 0 and 1 and when q is a number between 1 and 2, you can show that you can clearly see that this particular term is actually a number which is greater than 1. Let us take 1.5, so yeah right. So that means what we get is \dot{V} less than equal to some $c_1 V^{\alpha_1} - c_2 V^{\alpha_2}$ and that means this x basically your algorithm is or your optimal solution is fixed time state equilibrium is fixed time stable.

when $p > 2, \frac{p}{2(p-1)} \in (0, 1)$

$q \in (1, 2), \frac{q}{2(q-1)} > 1$

$\dot{V} \leq -c_1 V^{\alpha_1} - c_2 V^{\alpha_2} \quad T \leq \frac{1}{c_1(1-\alpha_1)} + \frac{1}{c_2(\alpha_2-1)}$

So your algorithm converges in a fixed amount of time. independent of the initialization and the settling time t is upper bounded by $\frac{1}{c_1(1-\alpha_1)} + \frac{1}{c_2(\alpha_2-1)}$. So, what did it require us to do? It required us to actually mean we know the result that we want to arrive at right and in the previous exercise also gave us idea as to what kind of gradient normalization one can potentially use. So, if you want to accelerate this further you can use even better normalization because we want to arrive at this condition for fixed-time stability right. So, you have to add this additional term ok. So, this kind of gives you an idea as to how one can design new algorithms right.

And this is all motivated from the kind of tools that we have available in continuous time, the kind of stability theory analysis that one can do. And if you have that thing mapped out, then basically it also helps you not just analyze the existing algorithm, but design the new algorithms right. So you can eventually use this continuous time implement this in discrete time and in fact we have done so while in order to provide guarantee in discrete time is little tricky. Empirically it works much better than things like Adam or RMS prop something that you would have used in for training neural networks. So, this is again like I mean it is still a very active field, I mean in fact most of this work that you are seeing here in today's lecture that has been developed in the last 3 to 4 years or 5 years kind of time.

So, it is a very active research area, but it is a good time to get in this particular field because I mean with sort of LLMs and machine learning models becoming larger and

larger, you actually have to be able to optimize them much faster. And these kind of tools will actually help you design better and better algorithms. Thank you.