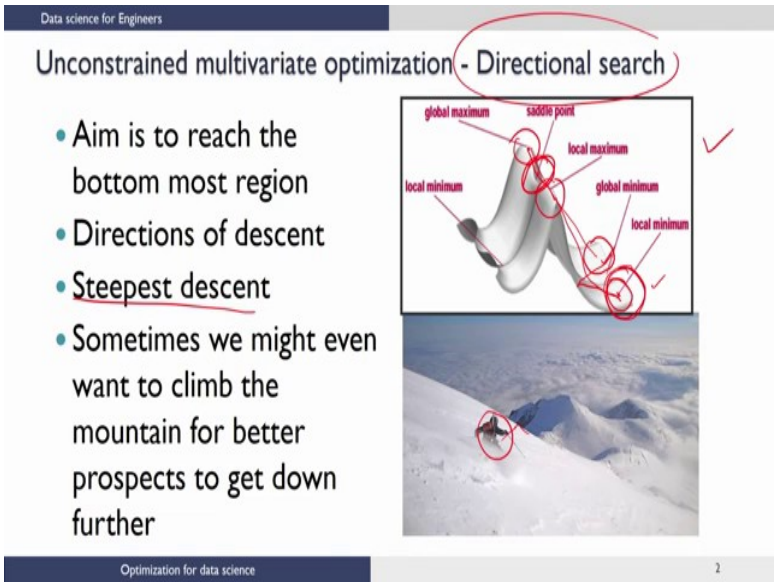
Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 25
Nonlinear Optimization Unconstrained Multivariate Optimization

In this lecture we will continue with Unconstrained Multivariate Optimization. In the last lecture we saw the conditions for a point to be an optimum point a minimum point for multivariate functions.

We described multivariate functions as functions with several decision variables. What we are going to do in this lecture is to look at the same conditions and show how you can numerically solve these optimization problems. And the reason why we are teaching this in a data science course is the following, if you think of any data science algorithm, you can think of it as some form of an optimization algorithm and the techniques that you will see in today's class are also used in solution to those data science problems or data science algorithms. And you will see these numerical methods as what is called as learning rule in machine learning and so on.

(Refer Slide Time: 01:24)



So, let us look at an unconstrained multivariate optimization problem. In unconstrained multivariate optimization problems we are

going to solve these using what we call as a directional search. The idea here is the following, if you are on the top of a mountains keying and you are interested in reaching the bottom most point from where you are pictorially shown through this picture here, you will see that there are several different points in this surface. This is a point which is at the bottom of the hill. So, we call this as a minimum point.

However, this is a local minimum because right next to it there is another point which is even lower which we call as the global minimum. We also see that there is a local maximum here a global maximum here and there are also points such as these which are called the saddle points. When you look at optimization algorithms, the aim is for us to reach this point. We want to avoid points like this because as I described before when we reach these kinds of regions while locally we cannot make our algorithm find anything better we know that globally this is not the best. So, in that sense we could do better.

Nonetheless this is an OK point if it is not very far in terms of the per-formance from a global minimum. However, we want to avoid points like these saddle points and so on, because as you see even in this picture you know saddle points could be very far away from the actual solution.

So, the aim is to reach the bottom most region. Typically what you would do is the following. So, if you are at a particular point here and then you say look let me go to the bottom of the hill as fast as possible, then you would look around and find the direction where you will go down the fastest. So, this is a direction we call as the steepest descent. So, the direction in which I can go down really fast and I will find that direction and then go down the direction.

Now, the way optimization algorithms work is the following, you are at a point you find the steepest descent direction, and then what you do is you keep going in that direction till a sensible amount of time or in this case the length of the step that you take in that direction.

The reason for this is the following, the reason is you could find this as the steepest descent direction and you could keep going in this direction, but let us say beyond this point you really do not know whether this is going to be the steepest descent at that point also. In other words is this going to continue to be the steepest descent till I get to my best solution. Now, that is something that is you cannot guarantee easily.

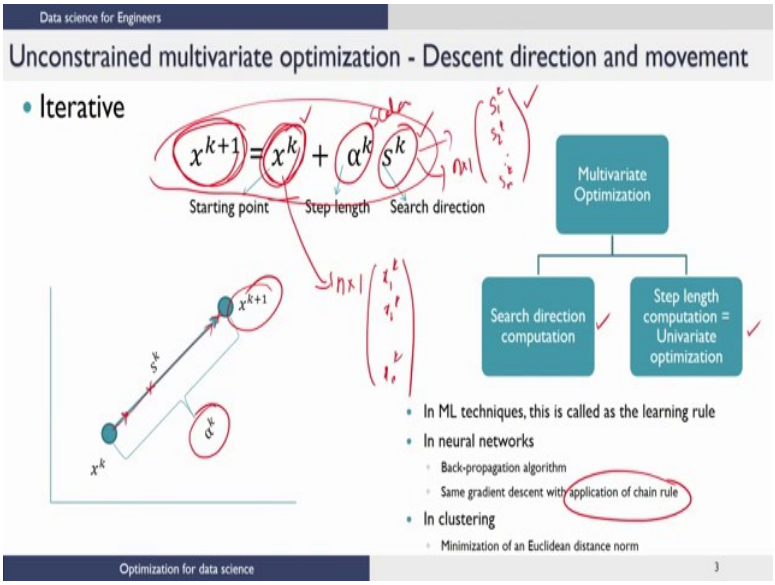
So, smarter strategy would be to find the steepest direction at wherever you are and then find how long you are going to move along this direction, go to the next point in the direction and then at that point reevaluate all the directions, and then find new steepest descent direction. If it turns out that the direction that you are on is continuing to be the steepest descent direction you continue to go on that

direction, if not you find a new direction and then go in the direction. So, that is a basic idea of all steepest descent algorithms.

Now, notice that you know we do the steepest descent and let us say we end up here, then at that point you will find no direction where you can improve your objective function that is you cannot minimize your objective function anymore. In which case you are stuck in the local minimum. So, there are optimization algorithms which, when they try to get out of the local minimum. The only way to do it is let us say if you are here the only way to get to global minimum is to really climb up a little more and then find directions and maybe you will find another direction which takes you here.

In some cases we might construct these optimization algorithms in such a way that you might actually make your objective function worse in the interim, looking for better solutions than your local optimum solution. So, that is what we have written here sometimes we might even climb the mountain to get better perspective. So, for example, if you are here and then say if you look at this you are going to land up here maybe you can go in this direction climb up actually get a better perspective and then come down here. So, some of these are mathematically formulated and solved. So, the basic idea of how these algorithms work is explained in this slide.

(Refer Slide Time: 06:50)



Now, let us see the mathematics behind whatever I described in the previous slide. So, the current point that I am at is what I would call as x_k. So, k stands for the kth iteration. So, at the 0^{th} iteration you will start with x_0, move to a point x_1 and at the first iteration you will start at

x_1 and move to x_2 and so on, till you find the solution that you are happy with.

The discussion that we had in the previous slide is seen here mathematically. So, what we are interested in is finding a new point which is better than x generally. And the notion of steepest descent is the following or for that matter any other search direction is the following. So, I have this point and I am going to move in a direction that I have chosen. If I choose the steepest descent then I choose that direction, if it is some other direction we choose that direction.

And then what I am going to say is I have a current point I have chosen a direction in which I want to move the only other thing that I need to figure out is how much should I move in this direction, and remember from vectors we talked about in the linear algebra portion of this course. If I start from x_k and then keep moving in the direction of s k this is what it will be I will be on this line. Now, the question is this x_{k+1} where is it placed. So, if α is very small it will be here slightly bigger x_{k+1} will be here even bigger x_{k+1} will be here and so on. So, how do I find this step length that I need to take in this direction.

So, notice that something interesting has happened. If I am in a particular point. So, remember this is also vector because we are talking about multivariate problems. So, there are several decision variables. So, this will be an n by 1 vector if I have n decision variables. So, this could be something like x_{1k}, x_{2k} all the way up to xnk and these are the current values for variables x, x2 all the way up to x n which are the decision variables that we are trying to find. Now, for this equation to be correct this is also going to be n by 1 vector the search direction and this is essentially a scalar this is just a number that we need to find out which is a step length.

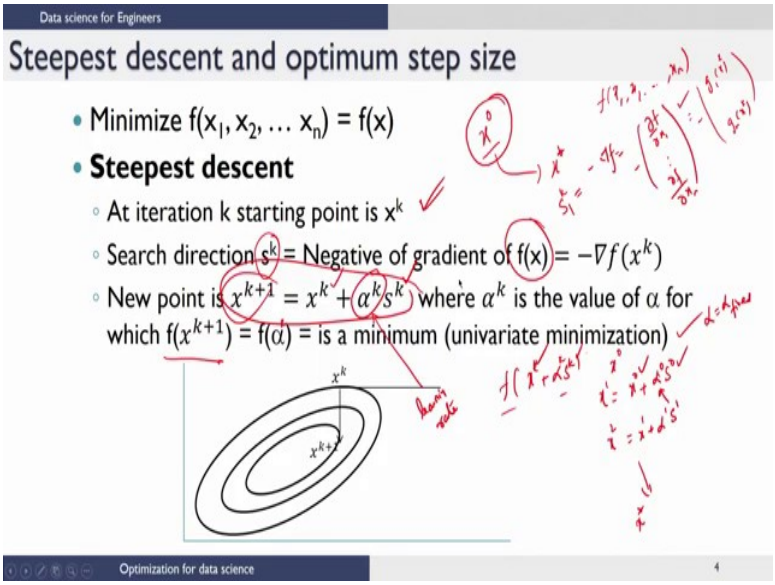
Now, we will show you what the steepest descent direction is, but you figure out this direction somehow. So, this is going to be a direction vector. So, this direction vector will be something like s_{1k}, s_{2k} all the way up to s_{nk}. So, let us assume that we have somehow figured this out, then the real question is what is the step length. So, one idea is to figure out the step length, so that this when substituted into the objective function is an optimum in some sense.

So, that is what we are going to try and do. So, the key take away from this is that if you are at a current point which you know and if you somehow figure out a search direction, then the only thing that you need to then calculate is the step length. And since step length is a scalar what happens is a multivariate optimization problem has been broken down into a search direction computation and finding the best step length in that direction which is a univariate optimization because we are looking for a scalar α.

In general this kind of equation that we see here you will see in many places as we look at machine learning algorithms in clustering, in neural networks and many other places, in machine learning techniques this is called the learning rule. Why is it called the learning rule? It is called the learning rule because you are at a point here and you are going to a new point you are learning to go to a point which is better than wherever you are and I mentioned to you before that we could think of this machine learning algorithms as being optimization problems solutions to optimization problems.

So, if you talk about neural networks one of the well known algorithms is what is called a back propagation algorithm. It turns out that the back propagation algorithm is nothing but the same gradient descent algorithm. However, because of the network and several layers in the network it is basically gradient descent we are including an application of a chain rule which we all know from our high school. Similarly in clustering algorithms you would see that clustering algorithms would turn out to be minimization of a Euclidean distance now.

(Refer Slide Time: 12:16)



So, let us now, focus on the steepest descent and the optimum step size that we need to take. So, the steepest descent algorithm is the following. At iteration k you start at a point xk. Remember with all of these optimization algorithms you would have to start with something called an initialization which is x naught and this is true for your machine learning algorithms also. All of them have to start at some point and depending on where you start, when you go through the sequence of steps in the algorithm you will end up at some point let us

call x*, and in many cases if the problem is non convex that is there are multiple local minima and global minima the point that you will end up is de-pendent on not only the algorithm, but also the initial point that you start with.

That is the reason why in some cases if you run the same algorithm many times and if the choice of the initialization is randomized, every time you might get slightly different results. So, to interpret the difference in the results you have to really think about how the initialization is done. So, that is an important thing to remember later when we when we learn machine learning algorithms.

So, as I said before, we start at this point x_k and then we need to find a search direction and without going into too much detail the steepest descent will turn out to be a search direction s_k which is basically the negative of gradient of f(x), where f(x) is your objective function. So, if f(x) is an objective function of the form with this many decision variables then grad f is basically $\partial f / \partial x_1$ all the way up to $\partial f / \partial x_1$ and negative grad f would be this, and we keep this as the search direction s k = negative grad f and this is called the steepest descent search direction.

The key thing that I really want you guys to notice here is the following, x_k is known, the function is known. So, to get to x_{k+1}, x_k is known since the function is known we know also the grad of f and s k is given as the $-\nabla(f)$evaluated at x_k. So, basically this is going to be let us say some functional form - g1x all the way up to gnx all you are going to do is simply substitute for this x, x_k. So, that basically gives you the search direction. So, this is given this is calculated once this is given.

Then the only thing that I need to find out is this αk and the way they are value for αk is found out is by looking at this $f(x_{k+1})$. Now, substitute this x_{k+1}into this. So, you are going to have $f(x_{k+\alpha})$ sk αk, sk. In this you know this you know this. So, this f is going to simply become a function of α right. So, let me put αk here. So, this is going to be a function of α.

Now, in the previous slide we talked about this and we said α is a scalar. So, it is just one variable. So, this becomes a univariate optimization or a univariate minimization problem. So, this is a critical idea that I want you to understand.

Now, any univariate minimization algorithm can be deployed to find out what αk is. So, you deploy a univariate minimization algorithm find a value for α k and then plug this back in then you have your algorithm for your multivariate optimization which will go something like this. So, you start with x_0 then x_1 is going to be $x_0 + \alpha_0$ s_0. So, x_0 is given based on your initialization, s naught is calculated, α_0

is optimized for, then you go on to x_2 is $x_1 + \alpha_1 s_1$ and so on. And you keep doing this till you use some rule for convergence you say at some point the algorithm is converged. So, that point is what I am going to call as x*.

Connection to machine learning algorithms is the following this α is usually called as the learning rate. You could either optimization find out or you could actually pick a value for this and then say let us run the algorithm let us not optimize for this α_0 α_1 and so on, I will give you a fixed value of α, α_{fixed}. So, you simply run your algorithm with fixed value of α which is x_0 x_1 will be $x_0 + \alpha_{fixed} s_0$, x_2 will be $x_1 + \alpha_{fixed} x_1$ and so on. So, that is also something that you could do. Nonetheless this is the critical equation which is used to optimize an objective function.

In the next lecture we will look at a numerical example that illustrates some of the ideas that we have seen. Now, see you in the next lecture.

Thanks.