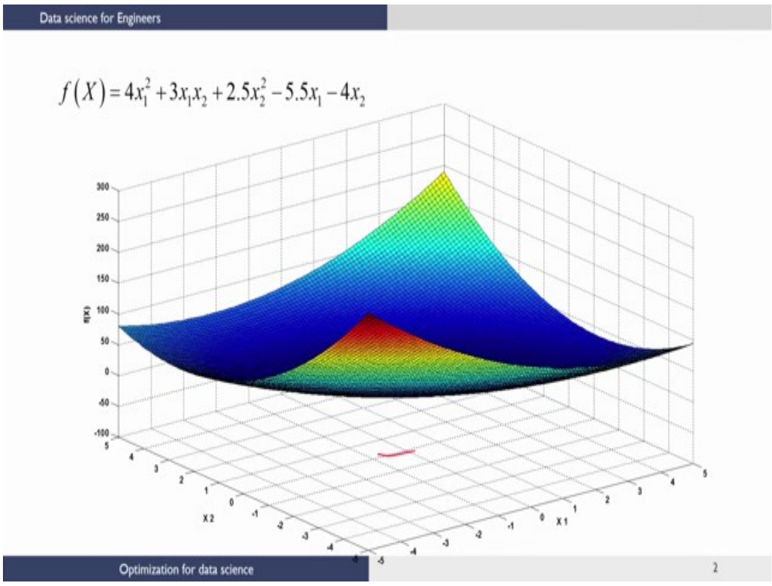
Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 26
Numerical Example Gradient (Steepest) Descent (OR) Learning Rule

Let us continue our lectures on optimization for data science. I am going to start out this lecture by showing you a numerical example of how gradient descent works in optimization. In many cases this is also called the learning rule in machine learning algorithms.

(Refer Slide Time: 00:35)



Data science for Engineers

$f(X)=4x_1^2+3x_1x_2+2.5x_2^2-5.5x_1-4x_2$

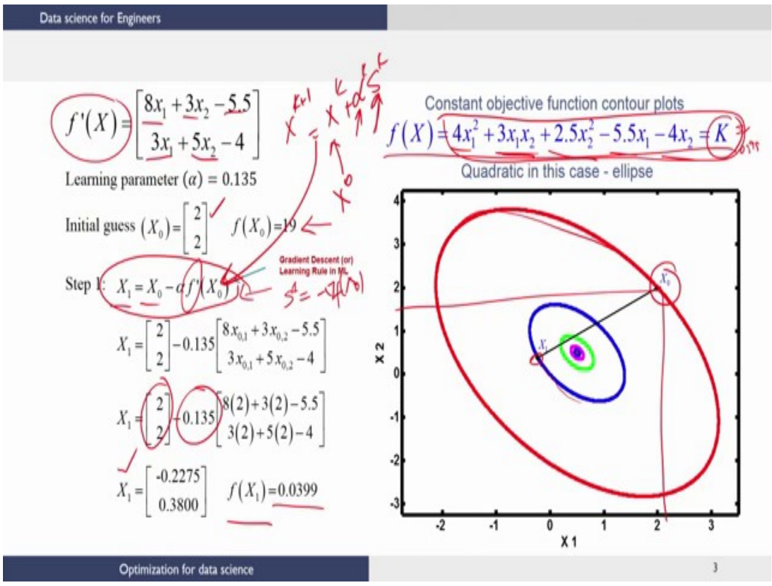
Optimization for data science

2

Let us look at a function f(x) which is 4 x_1^2 + 3 x_1 x_2 + 2.4 $x_2^{\,2}$ - 5.5 x_1 - 4 x_2. We are interested in minimizing this function. As you can notice this is a function of two variables x_1 and x_2. So, there are two decision variables that we can choose values for, to minimize this function.

As I mentioned before if it was a univariate optimization, then you could visualize the picture in two dimensions with the y axis being the objective function value and the x axis being the decision variable. Now, since we have a function with two variables x_1 and x_2 we visualize this in 3 dimensions, you have the two dimensions x_1 and x_2 on the plane below here and you have the z axis which is f(x) which is the objective function value. So, we are trying to look at how an algorithm would minimize this function in a numerical fashion.

(Refer Slide Time: 01:44)



Data science for Engineers

$$f'(X) = \begin{bmatrix} 8x_1 + 3x_2 - 5.5 \\ 3x_1 + 5x_2 - 4 \end{bmatrix}$$

Learning parameter (α) = 0.135

Initial guess $(X_0) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $f(X_0) = 19$

Step 1: $X_1 = X_0 - \alpha f'(X_0)$

Gradient Descent (or) Learning Rule in ML

$$X_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{0,1} + 3x_{0,2} - 5.5 \\ 3x_{0,1} + 5x_{0,2} - 4 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.135 \begin{bmatrix} 8(2) + 3(2) - 5.5 \\ 3(2) + 5(2) - 4 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} \quad f(X_1) = 0.0399$$

$X^{k+1} = X^k + \alpha S^k$

X^0

$S = -\nabla f(x)$

Constant objective function contour plots

$$f(X) = 4x_1^2 + 3x_1x_2 + 2.5x_2^2 - 5.5x_1 - 4x_2 = K$$

Quadratic in this case - ellipse

Optimization for data science

3

Now, notice from the previous lecture we described that while you try to minimize these functions you do what are called contour plots. And if you looked at the previous slide you would notice that x_1 and x_2 are in a plane and the objective function is a value that is projected outside the plane. So, if you think about constant objective function values then what you think about is a constant z value in the previous graph, and a constant z value would be a plane which will be parallel to the plane of the decision variables x_1 and x_2. So, when that cuts the surface that we saw in the previous graph then you have what are called these contour plots.

So, while we are trying to minimize a function which is represented in the z direction, all the changes to the decision variables are being done in A_2-dimensional plane which is shown here. For example, each of these curves that we see here are contour plots and as I mentioned before these contour plots are plots where the objective function takes the same value. So, from the previous slide you would notice that if you were to pass a plane through the surface you would see a curve like this would be traced on the surface, and as you move the surface up and down the size of the contour will increase or decrease correspondingly.

Now, the way most optimization problems work is the following. So, let us say we start with some value for x_1 and x_2. So, we simply guess a value for that and this is what we call as initialization in the optimization. So, let us assume that we initialized this problem at x_0. So, you would notice that this initialization basically says here is your value for x_1 and here is a value for x_2. So, we have picked some x_1, x_2, and we have initialized this problem.

Now, we know that the constant objective function values are contour plots and if we look at this equation here what we are saying is that the function f(x) which we saw from the previous slide. If we were to find a constant value for this function then I have to set th = k. Then I can look at this equation and then say how would this constant contour plot for f(x) look in the x_1, x_2 plane and you would notice that this is quadratic in this case. It is actually going to be an ellipse in for this particular function. So, this ellipse that we trace would be for some particular value of k and our initialization point is here.

Now, the way to interpret this is to say if we were to keep moving on this contour you would make no improvement through your objective function that is your objective function will not decrease because it is a constant contour plot. Now, in gradient descent we wrote this equation where we had $x_{k+1} = x_{k+\alpha}$ k s_k, we said this is the current point, this is the step length and this is the direction in which we should move.

So, let us look at this picture here. When we start we have x_0 which is initialization which is this point. What we need to do is we need to find a direction in which to move and once we find a direction in which we would like to move, then we will find out a learning rule or a learning constant which will take us to the next point. So, initial guess in this case that we have chosen is about 2 2 and f(x) naught value when you substitute this 2 2 is 19.

So, the next step is the following. So, we are going to say x_1 is x_0. Now, if I take this direction as - ∇f which is what we discussed last time which I have written here as f prime then, the equation will be the new point x_1 is x_0 - α times f '(x_0).So, this grad is evaluated at the point that you are currently at. So, the same equation becomes this here. So, just to illustrate the idea of how an optimization approach works we are going to pick some α here, which we have picked as 0.135 here, there is a way in which you can automate this and here we are just using this number.

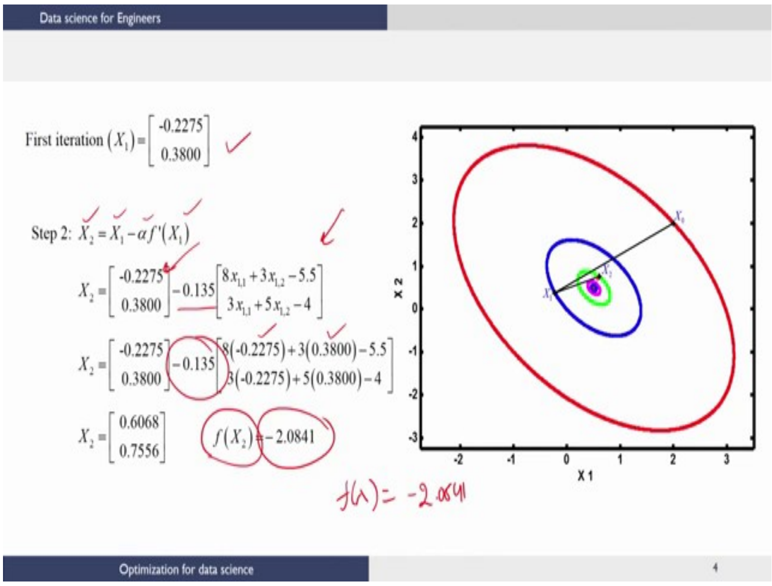
Now, if I take ∇ f, so when I differentiate this function with respect to x_1, I will get from this term $8x_1$, from this term I will get 3 x_2, and from this term I will get - 5.5. So, $\partial f / \partial x_1$ is going to be this term. And $\partial f / \partial x_2$ I am going to get 3 x_1 from this term, 5 x_2 through this and this - 4 I am going to get from here. So, this is $\partial f / \partial x_2$. So, that is your f prime or ∇ f.

Now, what you need to do is, once you identify this if you look at this equation this direction which is a gradient direction has to be evaluated at x_0 and since our original x_0 is 2 2, I am going to substitute the value of 2 2 into these equations. So, I have 8 times 2 + 3 times 2 - 5.5, and 3 times 2 + 5 times 2 - 4 and this is the learning parameter and this is our original point which gives me a v point x_1 after simple computation. And when you compute the function value at x_1 you notice that the original function value was 19. Now, it is come down to this number right here.

So, the point that we are trying to make is this direction is actually a good direction and then you move in the direction and you find that your objective function value decreases which is what which was our original intent because we are trying to minimize this function. And as I mentioned before this, gradient descent is usually called the learning rule. So, when you have parameters let us say that you are trying to learn for a particular problem this keeps adjusting this equation keeps adjusting the parameters till it serves some purpose and this adjustment is usually called the learning rule in machine learning.

So, this is the new point that we are right and if you find out this value which is 0.0399 and then set this k to be this number whatever was f(x) 1 which is 0.0399, then you would notice that the equation form remains the same except this constant has changed. So, this is continuing to be an ellipse, but it is an ellipse that are shrunk from your original ellipse. So, the constant contour plot, this blue plot, is the plot at which f(x) will take a value 0.0399. So, wherever you are on this blue curve or the blue contour the objective function value is the same. So, this is a first step of the learning rule that we see.

(Refer Slide Time: 09:50)


Data science for Engineers

First iteration $(X_1) = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix}$ ✓

Step 2: $X_2 = X_1 - \alpha f'(X_1)$

$$X_2 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{1,1} + 3x_{1,2} - 5.5 \\ 3x_{1,1} + 5x_{1,2} - 4 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} - 0.135 \begin{bmatrix} 8(-0.2275) + 3(0.3800) - 5.5 \\ 3(-0.2275) + 5(0.3800) - 4 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} \qquad f(X_2) = -2.0841$$

$f(x) = -2.0841$

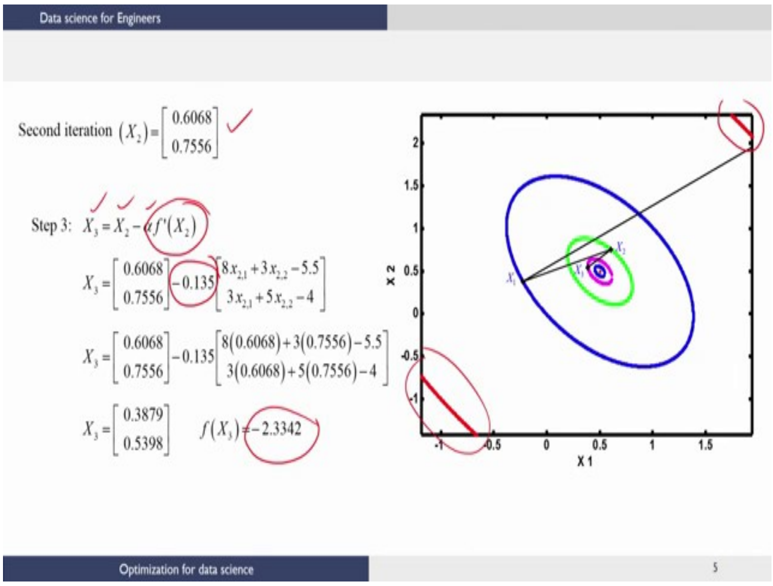
Optimization for data science

4

Now, let us proceed to the next iteration. The next iteration is pretty much exactly the same. What you can see here is now, we start with x_1 which was what we identified from the previous iteration and x_2 is x_1 - α f prime x_1. So, pretty much we are doing the same thing I substitute the value of x_1 here α remains the same and I do the same ∂f /∂x, but now, I evaluate the gradient at the new point x_1. So, if you notice here in the last slide we had put 2 and 2 for these values, but in this you would see I am using the x_1 value - 0.2275, 0.3800 and so on. And in this case the learning rate remains constant, but in more sophisticated algorithms or algorithms where you could actually optimize the size of this learning parameter as we go along in the algorithm.

Nonetheless, the ideas are pretty much the same only that this number will keep changing iteration to iteration. Now, we get a new value x_2 and notice that this new value of f x_2 when substituted into this function $f(x_2)$ give you even smaller objective function value. In fact, the objective function has become negative. So, this new point is shown here as x_2. Now, again much like how we discussed the previous iteration in the last slide, in this iteration if you were to take the function f(x) and then set it = - 2.0841 then that would be again an elliptical contour and that contour is actually described by this green contour.

(Refer Slide Time: 11:53)



Data science for Engineers

Second iteration $(X_2) = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix}$ ✓

Step 3: $X_3 = X_2 - \alpha f'(X_2)$

$$X_3 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{2,1} + 3x_{2,2} - 5.5 \\ 3x_{2,1} + 5x_{2,2} - 4 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} - 0.135 \begin{bmatrix} 8(0.6068) + 3(0.7556) - 5.5 \\ 3(0.6068) + 5(0.7556) - 4 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix} \quad f(X_3) = -2.3342$$

Optimization for data science

5

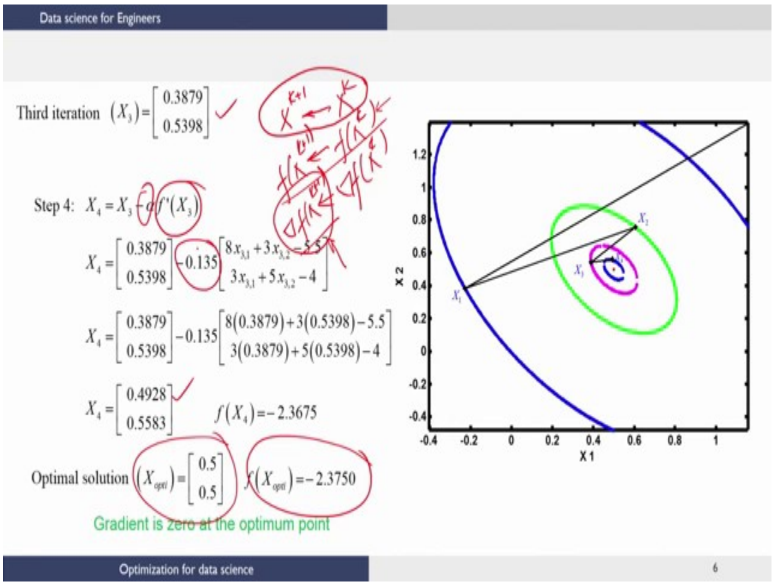
So, one more iteration you can simply follow through the steps same thing here x_2 from the previous slide the new x_3 value which is x_2 - α. Now, again the gradient is evaluated at the new point and the α remains the same and now you notice from the previous slides. Let us go back quickly to the previous slide and see what the value was. The value of the objective function was - 2.0841.

Now, when you look at the new point x_3 the objective function value has decreased even more it has become - 2.3341. So, we notice that at every step of the algorithm the objective function keeps improving for us here in this problem improvement means the objective function value keeps coming down and since at every point and the objective function value keeps coming down our hope is at some point it will hit the minimum value. How do you understand if it is a minimum value or not? It is something that that we will discuss in the next slide.

Nonetheless all I want you to notice is that at every iteration the value of the objective function keeps coming down because we are trying to minimize the objective function, and you can see the kind of improvement you get as we keep zooming down this picture, this was our original red elliptical contour which is kind of going outside the frame. So, this is a big elliptical contour from where we started, but we have improved the objective function to come here.

And just to make the connection between data science and optimization if this objective function were an error in some function that you are approximating your initial error is very large and as we learn or as the machine learns to approximate the function, what it does it keeps improving and it keeps finding out new points which could be the parameter values that that you are trying to find out such that the error keeps decreasing. So, that is what is happening here in this example.

(Refer Slide Time: 14:16)



Data science for Engineers

Third iteration $(X_3)=\begin{bmatrix}0.3879\\0.5398\end{bmatrix}$ ✓

$x^{k+1} \leftarrow x^{k}$

$f(x^{k+1}) \leftarrow f(x^{k})$

$\nabla f(x^{k+1}) \leftarrow \nabla f(x^{k})$

Step 4: $X_4 = X_3 - \alpha f'(X_3)$

$$X_4 = \begin{bmatrix}0.3879\\0.5398\end{bmatrix} - 0.135\begin{bmatrix}8x_{3,1}+3x_{3,2}-5.5\\3x_{3,1}+5x_{3,2}-4\end{bmatrix}$$

$$X_4 = \begin{bmatrix}0.3879\\0.5398\end{bmatrix} - 0.135\begin{bmatrix}8(0.3879)+3(0.5398)-5.5\\3(0.3879)+5(0.5398)-4\end{bmatrix}$$

$$X_4 = \begin{bmatrix}0.4928\\0.5583\end{bmatrix} \quad f(X_4) = -2.3675$$

Optimal solution $(X_{opti})=\begin{bmatrix}0.5\\0.5\end{bmatrix}$ $f(X_{opti}) = -2.3750$

Gradient is zero at the optimum point

Optimization for data science

6

So, you could go through this process. The third iteration maybe gives us this x_3 and then the next iteration gives you an f x_4 value which is this and you can notice that the objective function value has come down.

Also notice a couple of other interesting things that we see as we go along through this optimization procedure. If you noticed, I think the first value was very high for f(x) and after the first iteration the objective function value came down quite a bit, and then we have gradually keep improving the objective function value. And as you get closer and closer to the optimum the gradual improvement in your objective function value in the iterations that come later in the algorithm are going to be less. What I mean is I the very first improvement when we went from x_0 to x_1 was a large improvement. But when we go from x_2 to x_3 and x_3 to x_4, the improvement in the objective function, while the objective function is improving, the improvement amount of improvement keeps decreasing.

And you would expect that because as you get closer and closer to the optimum you know that the optimum point is where the gradient goes to 0. So, as close to the optimum if the function is reasonably continuous then you are going to have derivatives which are very small and if you notice each of these steps, this is one thing that dictates the size of the step you take. And if this is a constant value the size of the step you are going to take is going to come down and also the improvement in your objective function is going to come down. But keep in mind the objective function keeps improving all I am saying is that the amount by which it improves will keep coming down.

So, if you do this for a few more iterations you will get to the optimum point which is this solution 0.5, 0.5 and the function value at this optimum point turns out to be this. Now, the couple of things that I would address here. So, when you write an algorithm like this or when an algorithm like this works you have to tell the machine to stop doing this algorithm at some point. Which is what in optimization terminology called as convergence criteria. And the way the convergence criteria works is the following there are many ways in which you could you could post a convergence criteria and then say this the algorithm has converged.

So, we talked about the decision various values themselves decision variable values themselves. So, there is let us say we are at x_k and we are finding a new variable x_{k+1} we have this we have also the value of the function at x_k and the value of the function at x_{k+1}, and we have also got the gradient of the function at x_k, and the gradient of the function at x_{k+1}. So, what I am trying to show here is that when I am moving from x_k to x_{k+1}, I could compute all of these quantities. So, you could post a convergence criteria on any of these. So, for example, you could say the difference between this and this, in a vector of different sense, if that is becoming smaller and smaller then you might stop your algorithm.

So, the logic behind this is that if you are making minor modifications to your parameters you can keep doing it to try to get to perfect value, but at some point it starts making not much of a difference. So, you could use this norm as we call it which is the difference between these two values at two different iterations as a condition for saying the algorithm converges. That is when this becomes small enough you say the algorithm has converged.

You could also simply take the difference between the objective function values in two iterations for example. When that becomes very very small you could think about saying that the algorithm has converged. Or you could take the derivative at every point and then when the derivative norm becomes very small you could say the algorithm converges. The logic between these two are that in this case we are saying well we are doing this, but we are not really improving our objective function.So, I am going to be happy with whatever I get at some point and then say if you do not improve significantly and what is significant is something that you define I am going to stop the algorithm.

So, you could do that. Or when you do the norm of this you know ultimately at the optimum value you know the gradient has to be 0, that means, the norm of this vector has to be 0. So, when grad f becomes very close to 0 then you could say I have converged my algorithm and I am going to stop the algorithm at that point. So, in typical optimization packages or software there are these various options that you can use to ask for convergence to be detected and the algorithm to stop at that point.

So, this gives you an idea of how the analytical expression that we started with for maximum or minimum is converted into a gradient rule and these are all called as gradient based optimization algorithms. And then we showed you a numerical example of how actually this gradient based optimization algorithm works in practice. We also made the connection between these algorithms and machine learning and as I mentioned before most of the machine learning techniques you can think of them as some form of an optimization algorithm and the gradient descent is one algorithm which is used quite a bit in solving data science problems.

Couple of other things to notice are that the direction for changing your values iteration by iteration, in this case we have taken it as a steepest descent. There are many other ways of doing this you can choose directions in using other ideas that many other algorithms use. So, we here in this introductory course on data science we focused on the most common and the simplest of the search directions which is the negative of the gradient at that point.

And again these algorithms also keep changing the value of the learning parameter or the step length as they would call it in optimization algorithms, iteration to iteration. In this case we have kept that to be a constant just to make sure that we explain the fundamental ideas first before moving on to more complicated concepts. Nonetheless, I just want you to remember that this learning parameter is something that could be changed optimally from iteration to iteration in a given optimization algorithm.

So, with this I hope you have got a reasonable idea of univariate and multi-variate optimization, unconstrained non-linear optimization. What we are going to do in the next lecture is to look at how we can introduce constraints into this formulation, and what effect does a constraint have on the formulation and how do we solve constrained optimization problems. And as I mentioned before these constraints could be of two types, equality constraints and inequality constraints. We will see how we can solve optimization problems with equality constraints and inequality constraints. So, I will see you in the next lecture.

Thank you.