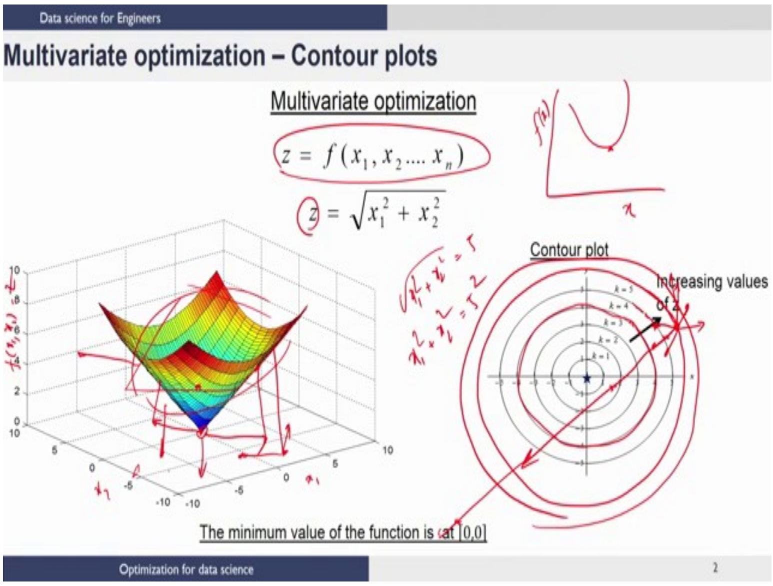
Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 24
Nonlinear Optimization Unconstrained Multivariate Optimization

In the previous lecture we described unconstrained non-linear optimization in the univariate case or when there was only one decision variable. In this lecture we are going to see how this is extended to cases where there are multiple variables that act as decision variables in the optimization problem.

(Refer Slide Time: 00:36)



Data science for Engineers

Multivariate optimization – Contour plots

Multivariate optimization

$z = f(x_1, x_2 .... x_n)$

$z = \sqrt{x_1^2 + x_2^2}$

Contour plot

Increasing values of z

The minimum value of the function is at [0,0]

Optimization for data science

2

So, when you look at these types of problems, a general function z could be some non-linear function of decision variables x_1 to xn. So, there are n variables that we could manipulate or choose to optimize this function z. A simple demonstration of this in a two dimensional case is root of $(x_1^2 + x_2^2)$.

Notice that we could explain univariate optimization using pictures in two dimensions that is because in the x direction we had the decision variable value and in the y direction we had the value of the function. So, you could see something like this and then say this is the minimum and so on. However, when you just extend this problem to two dimensions then you have to have 3-dimensional plots and in dimensions higher than 2, if the decision variables are more than 2 then it is difficult to visualize. So, what we are going to do is we are going to explain some of the main ideas in multivariate optimization through pictures such as the one that I have shown on the left side of the slide and as I mentioned before since even for cases where there are two decision variables we need to go to 3-dimensions and that is simply because if this is x_1 and this is x_2 or this is x_1 and this is x_2, I need a third dimension to describe the value of the function $f(x_1, x_2) = 0$. So, the objective function value becomes the third axis.

So, let us look at how we think about this unconstrained optimization when there are multiple variables. Take this picture for example, if you look at this picture right here on the left hand side of the slide you will notice that the minimum point is somewhere around here, which in this case happens to be 0 0 this is touching the x, y, x_1, x_2 plane and that is a solution minimum point is 0.

Nonetheless if you start moving in the x_1, x_2 plane then when you compute the objective function at different points. Let us say I compute the objective function at this point then this is going to be outside the plane and this is a value of the objective function at this point. And if I compute the objective function at this point it is going to be outside the plane this is going to be the value of the objective function at this point and so on and if I go this direction I might come here and so on.

So, what we are going to do is we are going to be in the space of decision variables and we are going to try and find an optimum solution because those are the values that we are actually choosing. So, for example, if let us say I have a point here on the space of the decision variables then the corresponding objective function value is this and clearly we know that that is not the minimum point.

So, what we need to do is we have to figure out some how to get here. Notice that the point at which the decision variables take values such that the function is a minimum is also in the decision variable space. So, essentially when we keep changing the values for the decision variables we are basically moving in this plane; however, while we are moving this plane we are looking at the values in the z direction to find out whether the point that we have reached is a minimum or not. To better visualize this we draw what are called contour plots which I show on the right hand side of this picture. So, think about a plane that cuts this objective function plot parallel to the x_1, x_2 surface. So, for example, let us say you think of a plane like this which is parallel to this and it is going to cut the objective function plot.

Now, if you have a plane that is parallel to the x_1, x_2 surface then what we are going to see is we are going to have the objective function value be a constant across the plane because when you project it here it is going to be at a particular $f(x_1, x_2)$ value or z value. So, what one could say then is that if I cut this surface with the plane parallel to x_1, x_2 surface then I am going to get what are called contours on the x_1, x_2 surface. So, you want to think about it this way. So, here is a plane that is cutting the surface. So, on the plane wherever the surface is cut you are going to have a contour and what we are going to do is we are going to project that contour onto this x_1, x_2 surface. So, that is the plot here. So, for example, we could take $z = 5$ and then have that plane cut this surface then let us see what the projection of that in x_1, x_2 axis will be.

So, we know that we are going to keep or hold the $z = 5$ as a constant. So, you will get this equation root of $x_1^2 + x_2^2 = 5$ which will give you x_1^2 $x_2^2 = 5^2$. So, this we all realize as equation of a circle centered at the origin with a radius of 5, which is what you see in this plot. Similarly if you say $k = 4$ you will get this contour plot and $k = 3$, $k = 2$ and so on.

Now, an interesting thing to notice is if I start with some decision variable values here and let us say I want to improve my objective function, I know that if I pick a contour like this and then from here if I keep moving on this contour I am not going to make any improvement to my objective function value. I also know that as I go away from this point in this direction, let us say I go to a new point, then that would be on a contour where the value of k is larger than where I was here. So that means, I have increased my objective function. So, if I move in any of these directions I am going to increase my objective function.

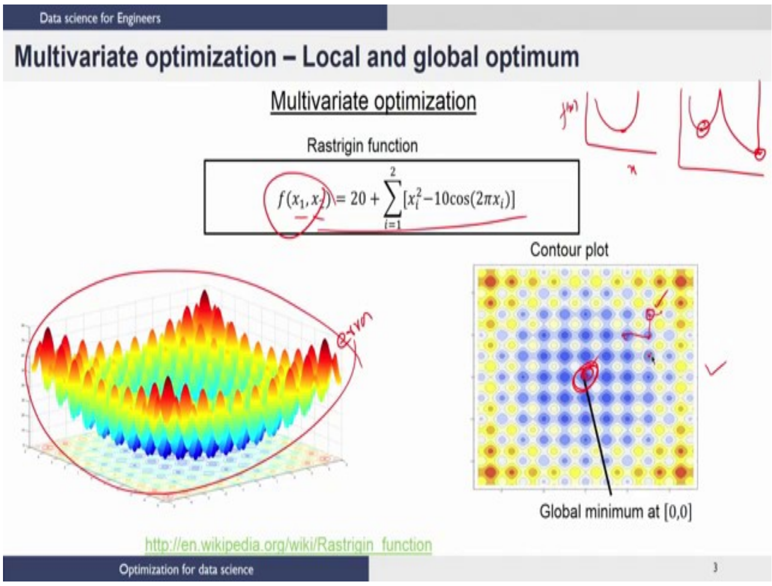
So, the one way in which I should move to decrease my objective function is to move in this direction because however, much I decide to move if I let us say I move here then this is the contour. So, the objective function value on this contour is less than this contour. So, this point is a better point then this one.

So, this is the basic idea about how we optimize this function.

Now, as I speak you would have noticed that there are two decisions that I need to make. The one decision that I need to make is of all of these directions, what direction should I choose? So, I have to sit here and then make a choice about the direction that I need to figure out. And once I choose a particular direction let us say I choose this direction how far should I go in this direction is another decision I should make.

So, for example, if I go here I have made some improvement to my objective function let us say if I go here I have made much more improvement to my objective function, but let us say if I go here I have actually made my objective function worse. So, there are two important things that I need to decide, one is the direction in which I should move in the decision variable surface and once I figure out which direction I should move in how much should I move in the direction. So, those are two important questions that we need to answer.

(Refer Slide Time: 09:59)



Data science for Engineers

Multivariate optimization – Local and global optimum

Multivariate optimization

Rastrigin function

$$f(x_1,x_2) = 20 + \sum_{i=1}^{2}[x_i^2 - 10\cos(2\pi x_i)]$$

Contour plot

Global minimum at [0,0]

http://en.wikipedia.org/wiki/Rastrigin_function

Optimization for data science

3

We will answer these questions when we look at numerical methods of solving these kinds of unconstrained optimization problems. In this lecture what we are going to do is we are going to show you the analytical conditions for minimum in a multivariate problem. Before we do that just like we saw in the univariate case, let us say this is f(x) x and then I have a function like this which has only one minimum which is a global minimum and then I also showed another case where we have a function may be like this where this is one minimum this is one minimum both are minima this is a local minimum and this is a global minimum.

This same thing happens in the multivariate case also. Here is an interesting example of a function where there are two decision variables let us say x_1 and x_2 and the function is of this form. If you plot this in a three d plot you will get this you can see there are many hills and valleys in this and you will notice if we do the projection of this on to the x_1, x_2 surface you will see that there are several minima in this picture and you will see that the global minimum is here.

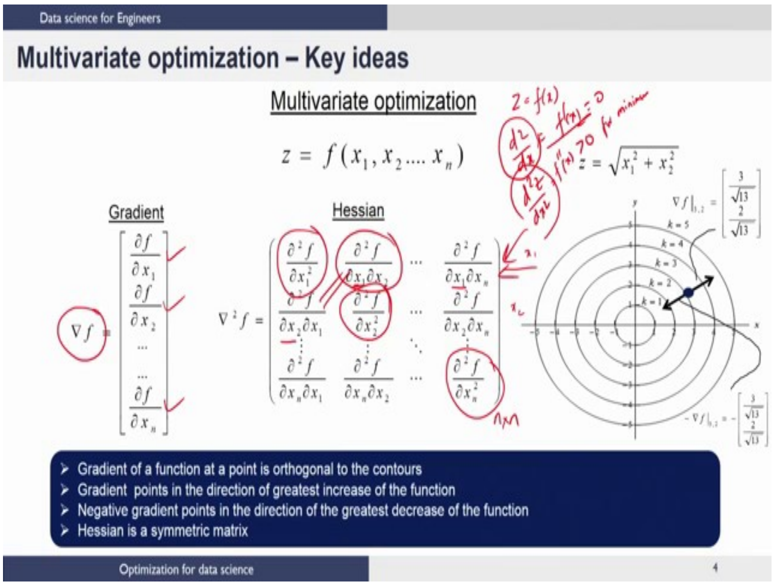
So, you can see how hard it can become in case of functions like this, where if you if you let us say, you start from here, then clearly you know one of the good things to do would be to go to this minimum. And you will be here and from here when you look at the conditions for minimum there will not be any difference between the conditions here and the conditions here in terms of the first order and second order conditions that we talked about in the univariate case we will see what the equivalent conditions are in the multivariate case subsequently.

However, from just those conditions you will not see any difference, nonetheless if you actually compute the objective function value at this point and this point this will be much smaller than this. However, when you are here you have no reason to suspect that a point like this really exists unless you do considerable analysis. So, in cases like this what you will have to do is, you have to see whether you can improve it further, that basically means though you know locally you are very good here you have to do some sacrifice and then try and see whether there are other points which could be better. So, there are algorithms which will let you jump here and then maybe will jump here and so on, but these are all algorithms where it is very difficult in a general case to prove that I will go and hit the global minimum.

Now, remember this is something important to note particularly from a data science viewpoint because let us say this is your error surface. Just as a idea for you to think about how important these concepts are and you are trying to fit a model and the best parameters for the model are here. But a typical optimization algorithm would get stuck anywhere in any of these local optima.

From a data science viewpoint what it means is that the error is not as small as it could be here. However, from the model viewpoint if you were to change the parameters from this value in any direction you change, you will be finding out that the error actually increases in the local region. So, there is very little incentive to improve your objective function value, sorry a very little incentive to move away from this point because locally you are increasing your objective function value. So, ultimately your algorithm might find parameters which while may be acceptable are not the best. So, this is one problem that needs to be solved really to have good efficient data science algorithms.

(Refer Slide Time: 14:54)



Data science for Engineers

Multivariate optimization – Key ideas

Multivariate optimization

$$z = f(x_1, x_2 \ldots x_n)$$

Gradient

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdots \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Hessian

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

$z = f(x)$

$\frac{dz}{dx} = f'(x) = 0$

$f''(x) > 0$ for minimum

$\frac{d^2z}{dx^2}$

x_1

x_2

nM

$$z = \sqrt{x_1^2 + x_2^2}$$

$$\nabla f|_{1,2} = \begin{bmatrix} \frac{3}{\sqrt{13}} \\ \frac{2}{\sqrt{13}} \end{bmatrix}$$

$$-\nabla f|_{1,2} = -\begin{bmatrix} \frac{3}{\sqrt{13}} \\ \frac{2}{\sqrt{13}} \end{bmatrix}$$

- Gradient of a function at a point is orthogonal to the contours
- Gradient points in the direction of greatest increase of the function
- Negative gradient points in the direction of the greatest decrease of the function
- Hessian is a symmetric matrix

Optimization for data science

4

So, let us get back to finding out analytically how we solve this problem. So, if you have a multivariate optimization problem where you have z is f(x_1), x_2 all the way up to xn. And in the univariate case let us just contrast this with the univariate case. So, let us say z = f(x) just one variable. Then remember we said the necessary condition for a minimum is that I should have dz dx = f ‘(x)= 0 and then we said d^2z / dx_2 = f’’(x)> 0 for minimum. So, these are the conditions that we described in the previous lecture.

So, the derivative in a single dimensional case becomes what we call as a gradient in the multivariate case. So, in this case we have dz dx or df dx. However since there are many variables we have many partial derivatives and the gradient of the function f is a vector such that in each component I compute the derivative of the function with respect to the corresponding variable. So, for example, this is ∂f/ ∂x_1 is the first component ∂f /∂x_2 is the second component and ∂f/ ∂xn is the last component. So, this replaces this in the single variable case.

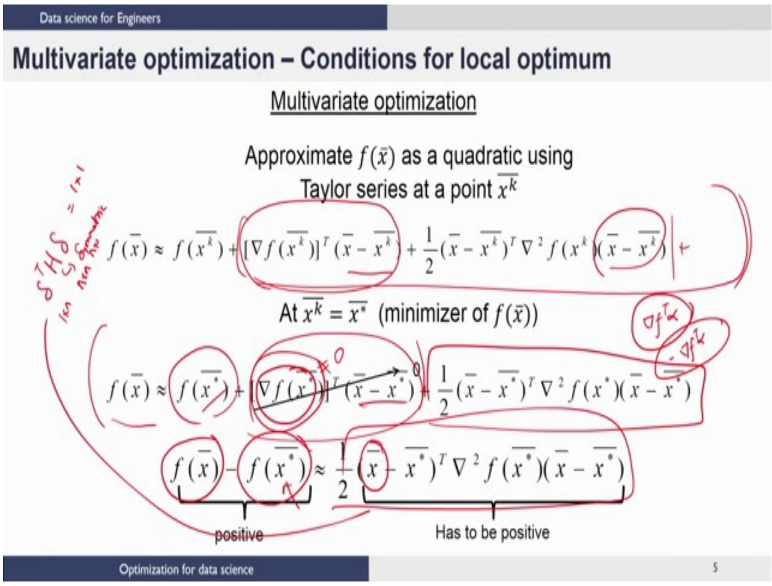
And this is replaced by what we call as a hessian matrix in the multivariate case. So, this is a matrix of dimension n by n. The first component is ∂^2 f /∂x_1^2, the second component is ∂^2 f/ ∂x_1 ∂x_2 and so on, and you fill out the row like this. So, the notation that we have used is we have used the x_1 in the front here and x_2 here for second column, x_3 here for the third column and so on, xn here for the last column.

Now, this is with respect to variable x_1. Now, you can do the same thing with respect to variable x_2. Notice here that x_2 has come before x_1 because it is in the second row and this diagonally is always with respect to the same variable differentiated twice. So, this is ∂^2 f/ ∂x_2^2 ∂^2

f/ ∂xn^2 and so on. Also notice that this hessian will be a symmetric matrix because for most functions this = this and similarly you will have ∂^2 f /∂x_1 ∂x_3 the next term here will be ∂^2 f /∂x_3, ∂x_1 which will be the same. So, the hessian matrix is going to be symmetric. Remember in the linear algebra lecture we said that we will be seeing symmetric matrices quite a bit and here is a symmetric matrix that is of importance from an optimization viewpoint.

Now, what we need to do is we need to see how these conditions in the univariate case translate to the multivariate case.

(Refer Slide Time: 18:36)



Data science for Engineers

Multivariate optimization – Conditions for local optimum

Multivariate optimization

Approximate $f(\bar{x})$ as a quadratic using Taylor series at a point $\overline{x^k}$

$$f(\bar{x}) \approx f(\overline{x^k}) + [\nabla f(\overline{x^k})]^T(\bar{x}-\overline{x^k}) + \frac{1}{2}(\bar{x}-\overline{x^k})^T \nabla^2 f(\overline{x^k})(\bar{x}-\overline{x^k}) +$$

$\delta^T H \delta = (+)$

At $\overline{x^k} = \overline{x^*}$ (minimizer of $f(\bar{x})$)

$$f(\bar{x}) \approx f(\overline{x^*}) + \underbrace{[\nabla f(\overline{x^*})]^T(\bar{x}-\overline{x^*})}_{\to 0} + \frac{1}{2}(\bar{x}-\overline{x^*})^T \nabla^2 f(\overline{x^*})(\bar{x}-\overline{x^*})$$

$\neq 0$

$\nabla f^T \alpha$

$-\nabla f^T \alpha$

$$\underbrace{f(\bar{x}) - f(\overline{x^*})}_{\text{positive}} \approx \frac{1}{2}\underbrace{(\bar{x}-\overline{x^*})^T \nabla^2 f(\overline{x^*})(\bar{x}-\overline{x^*})}_{\text{Has to be positive}}$$

Optimization for data science

5

So, much like what we did in the univariate case, we are going to do a Taylor series approximation and what I have done here is I have just written it till two terms there are more terms here. But what we are going to do is we are going to make the argument that if you make the distance between the point that you are at and the next point that you are going to choose very small, then whatever is the leading term in the sum is going to decide the sign of the whole sum.

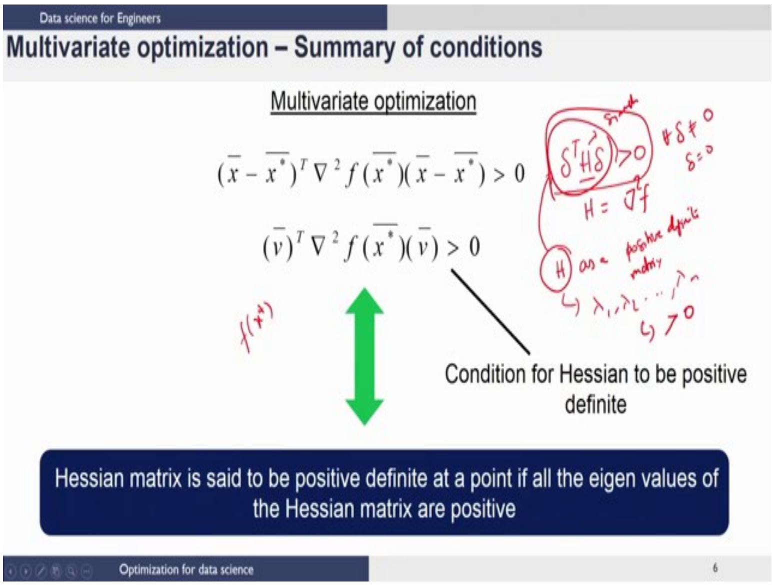
So, in other words if you take this whole thing right here if you keep making this as small as possible or as small as needed then what will happen is, the fact that whether this infinite sum is positive or negative can be identified only by the first term and if that is positive then the whole sum series sum is going to be positive and so on. So, that is the kind of logic that we are going to use again here.

Much like before we said that if I keep making this small I need to only look at this here and much like the univariate case if this does not go to 0, I can make this term either positive or negative. To see this if I take a particular direction and then say δ f ^{T} α. If this turns out to be negative this number, then if I go in the opposite direction of - α I will get ∇ (α) this will be positive; that means, that I will have a point here which can be either larger than this or smaller than this and if I can find a point such that this is smaller than this then this cannot be a minimum.

So, whatever you do unless this goes to 0, I cannot ensure that this is a minimum point. So, the first condition that we will get is that this is 0. And once that is 0 then I am left with the just this term right here and if you notice this term is of the form δ^T the hessian matrix let me use H here δ. We know that this is a symmetric matrix and let us also make sure we understand this clearly the function f is a scalar function and you can see that here also H is an n by n matrix here λ^T will be 1 by n and λ will be n by 1. So, when I do this I will get one by one which is a scalar.

Now, irrespective of this x̅, if this is a positive number then we can say irrespective of whatever direction you take this will always be greater than this in the local region which would qualify this point x star as a minimum point. So, that is the important idea that that you should remember.

(Refer Slide Time: 22:15)



Data science for Engineers

Multivariate optimization – Summary of conditions

Multivariate optimization

$$(\bar{x}-\bar{x}^*)^T \nabla^2 f(\bar{x}^*)(\bar{x}-\bar{x}^*) > 0$$

$$(\bar{v})^T \nabla^2 f(\bar{x}^*)(\bar{v}) > 0$$

Condition for Hessian to be positive definite

Hessian matrix is said to be positive definite at a point if all the eigen values of the Hessian matrix are positive

Optimization for data science

6

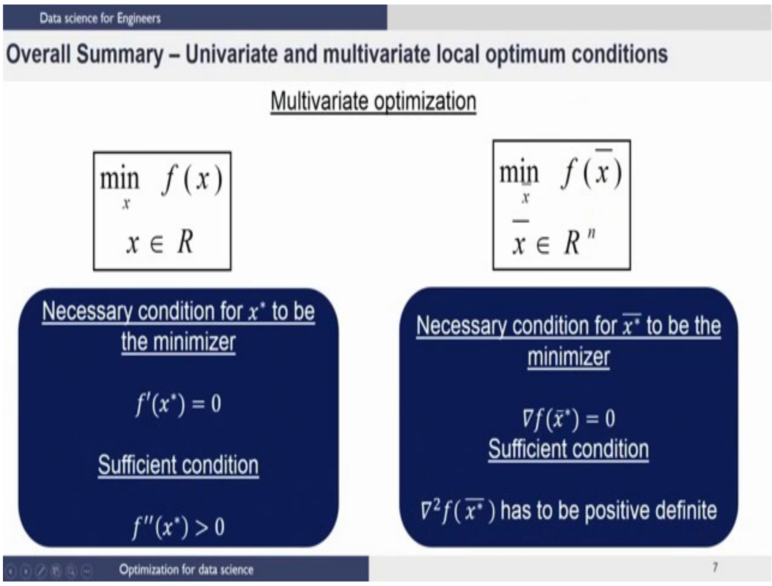
So, we come back to this in the last slide I wrote this as δ^T H δ > 0, H is symmetric. H is basically this second derivative matrix. Now, we did not see this in the linear algebra lectures, but if I need this condition to be satisfied irrespective of whatever δ is then we call this H as a positive definite matrix. So, if H is positive definite then this will be greater than 0 for all δ ! = 0, clearly when δ = 0 this will be = 0.

So, how do I check if a matrix that I compute is positive definite or not. Remember from the linear algebra lecture we said if I have a symmetric matrix, then I will have the eigenvalues as being real. So, symmetric matrices always have real eigenvalues and the eigenvalues could be positive or negative in this case. Now, the linear algebraic result for positive definite matrix is that if this matrix has let us say n eigenvalues, and if all of these eigenvalues are greater than 0 then this matrix is called positive definite.

In other words if all the eigenvalues of this matrix are greater than 0, it is automatically guaranteed that whenever we compute this for any δ direction we will always get a positive quantity. So, this has already been proved. So, if you want this to be positive for any direction why do we want this we want this because we want f(x*) to be the lowest value in its neighborhood and that we said will happen if this is positive for any δ or for every δ this should be positive. That condition can be translated to H being positive definite and H being positive definite can be translated to the condition that λ1 to λn the n eigenvalues of H are strictly greater than 0.

Now, what this does is the following. So, in a multivariate case it gives us a way to identify points that could be optimum points and once we identify those points we can compute this hessian matrix at those points and then computation of the eigenvalues of this hessian matrix would allow us to determine whether the point is a maximum point or a minimum point and so on. So, this is the complete equivalent of what we did in the univariate case.

(Refer Slide Time: 25:38)



Data science for Engineers

Overall Summary – Univariate and multivariate local optimum conditions

Multivariate optimization

$$\min_{x} f(x)$$
$$x \in R$$

$$\min_{\bar{x}} f(\bar{x})$$
$$\bar{x} \in R^{n}$$

Necessary condition for x^* to be the minimizer

$f'(x^*) = 0$

Sufficient condition

$f''(x^*) > 0$

Necessary condition for $\bar{x}^*$ to be the minimizer

$\nabla f(\bar{x}^*) = 0$

Sufficient condition

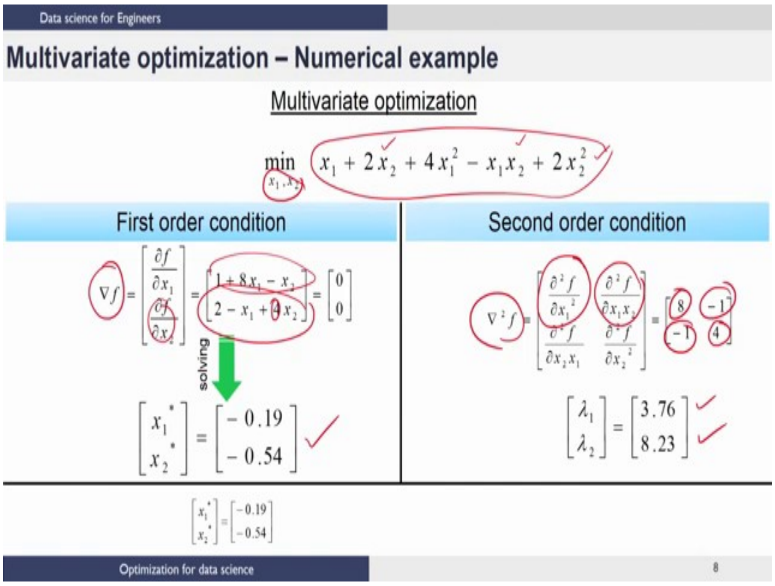
$\nabla^2 f(\bar{x}^*)$ has to be positive definite

Optimization for data science

7

So, to summarize in the univariate case the two conditions are f prime has to be zero and f double prime has to be greater than 0. In the multivariate case these translate to ∇ f = 0 and the Hessian matrix being positive definite.

(Refer Slide Time: 25:57)



Data science for Engineers

Multivariate optimization – Numerical example

Multivariate optimization

$$\min_{x_1, x_2} \; x_1 + 2x_2 + 4x_1^2 - x_1 x_2 + 2x_2^2$$

First order condition

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 8x_1 - x_2 \\ 2 - x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

solving

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix}$$

Second order condition

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ -1 & 4 \end{bmatrix}$$

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 3.76 \\ 8.23 \end{bmatrix}$$

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix}$$

Optimization for data science

8

Let us take a very very simple example and identify an optimum solution.

So, consider this multivariate example. So, there are two decision variables and this is a function in terms of these two decision variables.

So, what you can do is you can first construct this ∇ f vector which is ∂f /∂x$_1$. So, that would be ∂x$_1$/ ∂x$_1$ will be 1, this will be a 0 term this will be 4 time 2 times x$_1$, 8 x$_1$ this would be - x$_2$ and this would be 0. So, the first term is ∂f/ ∂x$_1$ here similarly when you do ∂f /∂x$_2$, I will have a term corresponding to this two. This when differentiated with respect to x$_2$ will go to 0 corresponding to this I will have a - x$_1$. Now, and corresponding to this I left 4 x$_2$ which is what we have here.

So, we have these two equations that we need to solve. So, when we solve this and get let us say one of the solutions x$_1$* x$_2$* is this here. I can check whether this is a maximum point or a minimum point, to do that what I have to do is I have to do this second derivative matrix. So, the way you do the second derivative matrix is the following. So, the first term is ∂^{2} f/ ∂x$_1^2$. So, we already have ∂f /∂x$_1$.

So, if you differentiate it with respect to x 1 you will get this term. So, the only term remaining will be 8 which is what we see here and when we look at this we already have ∂f/ ∂x$_2$. So, we have to differentiate this with respect to x_1. So, the only term remaining will be - 1 which will be here and I already told you this is a symmetric matrix. So, you can simply fill in the - 1 here and to get this term I already have $\partial f/ \partial x_2$ here I differentiate this again with respect to x_2. So, the only thing that will be remaining would be 4 which is here. So, I have this. Now, what I need to do is I need to compute the eigenvalues for this and when I compute the eigenvalues for this I find the eigenvalues to be both positive, that means, that this is a minimum point.

Now, when we look at this equation here there are two equations in two variables and both are linear equations. So, there is going to be only one solution here and it turns out that that solution is a minimum for this function. So, this finishes our lecture on multivariate optimization in the unconstrained case.

What we will do in the lectures that follow, we will look at some numerical methods for solving these types of problems. We will introduce the notions of how to solve these problems when there are constraints. We look at two types of constraints, one are what we call as equality constraints the other type of constraints are inequality constraints. So, we will pick up from here in the next lecture.

Thank you.