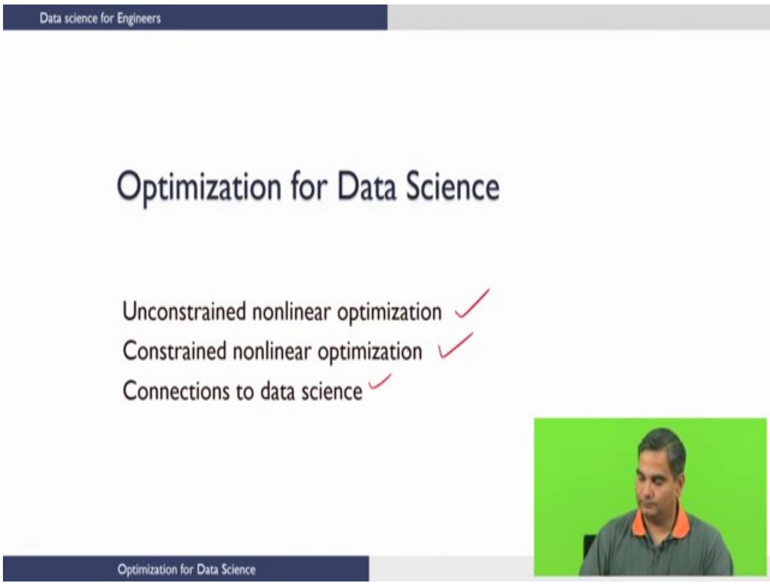
Data science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 23
Optimization for Data Science

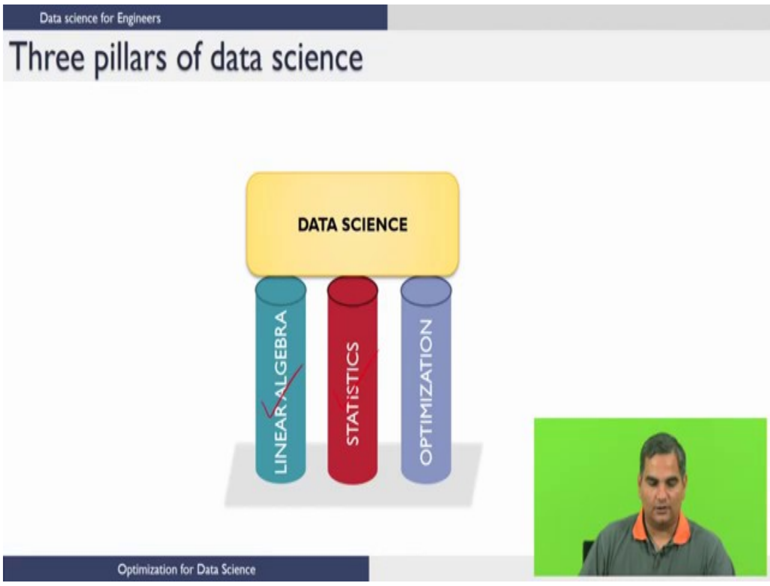
(Refer Slide Time: 00:11)



In this series of lectures now, we will look at the use of optimization for data science. We will startt with a general description of optimization problems and then we will point out the relevance of understanding this field of optimization from a data science perspective.

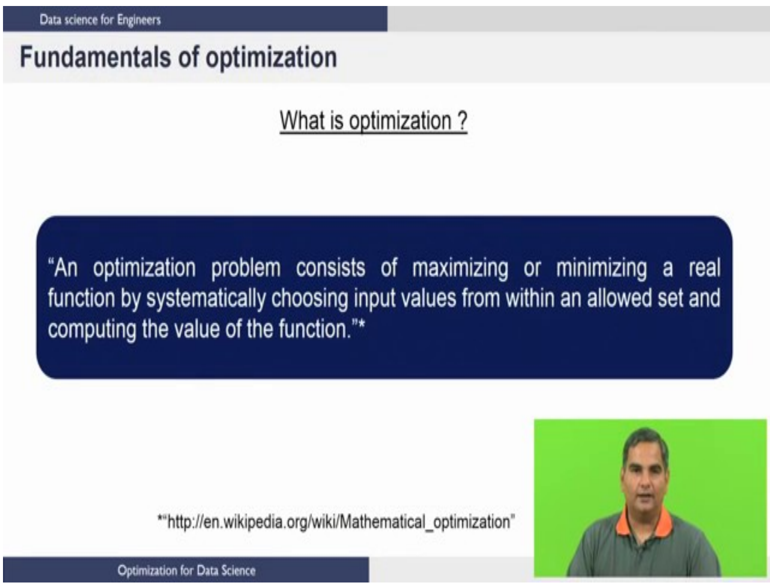
We will also introduce you very very briefly to the various types of optimization problems that people solve. While all of these types of problems have some relevance from a data science perspective, we will focus on two types of optimization problems which are used quite a bit in data science. One is called the unconstrained non-linear optimization and the other one is constrained nonlinear optimization and as I mentioned before we will also describe the connections to data science.

(Refer Slide Time: 01:21)



I would really consider from a mathematical foundations viewpoint that the three pillars for data science that we really need to understand quite well are linear algebra which you already seen before. Following that you saw series of lectures on statistics and the third pillar really is optimization, which is used in pretty much all data science algorithms. And quite a bit of the optimization concepts for one to understand quite well you need a good fundamental understanding of linear algebra which is what we have tried to deliver through the series of lectures on linear algebra.

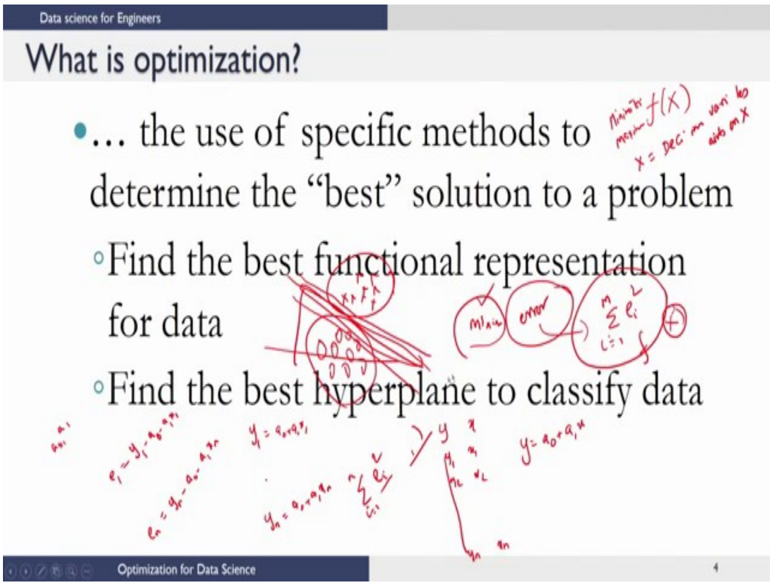
(Refer Slide Time: 02:04)



So, we will start by asking what is optimization and Wikipedia defines optimization as a problem where you maximize or minimize a real function by systematically choosing input values from an allowed set and computing the value of the function, we will more clearly understand what each of these means in the next slide.

Now, when we talk about optimization we are always interested in finding the best solution. So, we will say that I have some functional form that I am interested in and I am trying to find the best solution for this functional form.

(Refer Slide Time: 02:30)



Now, what does best mean? You could either say I am interested in minimizing this functional form or maximizing this functional form. So, this is the function for which we want to find the best solution. And how do I minimize or maximize this functional form? I have to do something to minimize or maximize and the variables that are in my control so that I can maximize or minimize this function or these variables x. so these variables x or call the decision variables.

And in the previous slide we talked about these being in an allowed set. What basically that means, is while I have the ability to choose values for x, so that this function f is either maximized or minimized, there would be some constraints on x which would force us to choose x in only certain regions or certain sets of values for this optimization problem. So, in that sense I have an objective which I am trying to maximize or minimize. I have decision variables which I can choose values for, that will either maximize or minimize the function. However, I might not have complete control over this x there might be some restrictions on x which are the constraints on x which I have to satisfy while I solve this optimization problem.

Now, why is it that we are interested in optimization in data science? So, we talked about two different types of problems, one is what is called a function approximation problem which is what you will see as regression later. So, in that case we were looking for solving for functions with minimum error remember that. Now, the minute I say minimum error, then I have the following minimum error, we said we have to define what this error is somehow, and the minute we say minimum error that basically means we are trying to find something which is the best we are trying to minimize something.

So, this part is already there finding a best for some function. And this error is something that we define. So, for example, if you remember back to our linear algebra lectures we said if there are many equations and they cannot be solved with a given set of variables then we said we could minimize this σ_i = 1 to m e_i^2.

So, this is the function now that we are trying to minimize. And there are the decision variables in that particular case we said the variable values are going to be the decision variable. So, this whole function if I call this as f, this is going to be a function of x - the values that the variables take. So, you already have a situation where you are trying to minimize a function and these are the decision variables. This is completely unconstrained or I have no restrictions on the values of x I choose, I can choose any value of x, I want as long as that value minimizes this or finds a best value for this function.

Now, remember the other case that we saw where we looked at much more variables than equations. In that case again we minimize the norm of the solution as the objective again there is a minimization there is an objective. However we said that optimization problem is constrained by the fact that the solution that I get should satisfy the equation. So, ax = b is a constraint there in which case I am constraining of all the x's that I can take I am constraining to those which would satisfy the equation ax = b. So, this idea of representation is used quite a bit in data science.

Another way to think about the same problem is the following, if I give you data for y and x and let us say you are trying to t a function between y and So, you might say $y = a_0 + a_1 x$. Then what you have here is the following. So, I might give you several samples. So, I have y_1 x_1, y_2 x_2, yn and xn. So, I have given you several samples and I have told you that this is the model that you need to x.

So, if I put each of the sample points into this equation. So, I will get $y_1 = a_0 + a_1 x_1$ and all the way up to $yn = a_0 + a_1 xn$. Clearly you can take the view that there are n equations here, but only two variables. So, there are many more equations and variables. So, I cannot solve all of these equations together. So, what I am going to do is I am going to define an error function which is very similar to what we saw before $y_1 - a_0 - a_1 x_1$ all the way up to $yn - a_0 - an\ xn$. And then I know that I have only two variables that I can identify which are a_0 and a_1; however, there are n equations. So, what I am going to do is I am going to minimize a sum of squared error is something that we talked about.

Now, this error is going to be a function of the two parameters a_0 and a_1. So, these become the decision variables and this becomes a function. And if you have no constraints on what the values can take, what the values these variables can take, then you have an unconstrained optimization problem. This is the type of problem that you would solve in linear regression and in general this is also called as function approximation problem. So, this is one type of problem which is used quite a bit in data science because in many cases we are looking at functional relationships between variables. So, that is one reason why optimization becomes important.

Now, in terms of the other bullet point I have here which is find the best hyper plane to classify this data. This is also something that we had seen before, where we looked at data points and then we said for example, I could have lots of data here corresponding to one class this I described when we are talking about linear algebra and I could have lot of data points here corresponding to another class. Now, I want to find a hyperplane which separates this.

Now, you could ask the question as to which is the best hyperplane that separates this. So, you could say I could draw a hyperplane here or I could draw hyperplane here or I could draw a hyperplane here and so on. Now, which one should I choose and the minute I say which one should I choose. We know that these hyper planes are represented by an equation. Then we say which hyperplane do I choose, then basically it means I am saying which equation do I use, which basically means what are the parameters in the equation that I choose to use. So, I want to find the parameters parameter values that I should use in that equation. So, those become the decision variables the parameters that characterize these hyper planes become the decision variables.

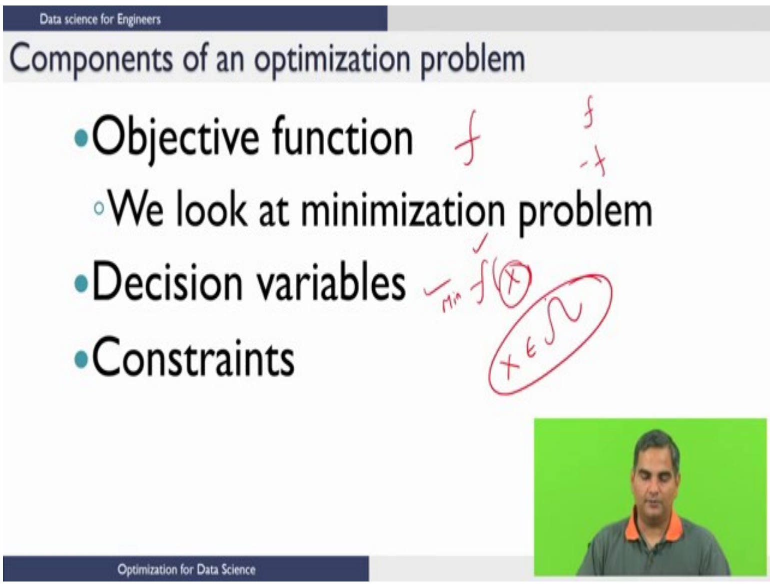
And in this case the function that I am trying to optimize is that when I choose a hyperplane I should not miss classify any data. So, for example, I have to choose a hyperplane in such a way that all of this data is to one half space of the hyperplane and all of this data is to the other half space of the hyperplane. So, you see that again this classification problem becomes an optimization problem.

(Refer Slide Time: 11:41)

So, in summary we can say that almost all machine learning algorithms can be viewed as solutions to optimization problems and it is interesting that even in cases, where the original machine learning technique has a basis derived from other fields for example, from biology and so on one could still interpret all of these machine learning algorithms as some solution to an optimization problem. So, basic understanding of optimization will help us more deeply understand the working of machine learning algorithms, will help us rationalize the working.

So, if you get a result and you want to interpret it, if you had a very deep understanding of optimization you will be able to see why you got the result that you got. And at even higher level of understanding you might be able to develop new algorithms yourselves.

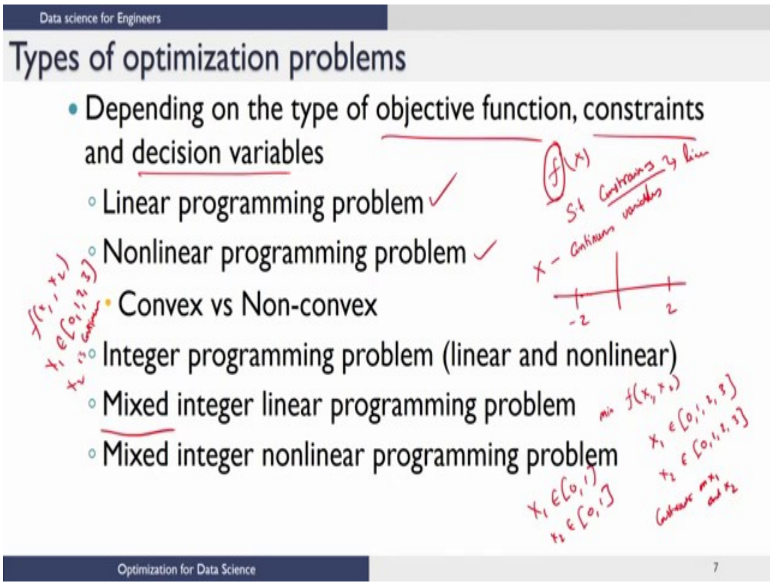
(Refer Slide Time: 12:42)



So, as we have described in quite detail till now, an optimization problem has three components the first component is an objective function f which we are trying to either maximize or minimize. In general we talk about minimization problems this is simply because if I have a maximization problem with f, I can convert it to a minimization problem with - f. So, in without loss of generality we can look at minimization problems. So, that is one component in an optimization problem.

The second component are the decision variables which we can choose to minimize the function. So, I write this as f(x). So, this is a function and these are the decision variables and our goal is to minimize. And the third component is the constraint which basically constrains this X to some set that will be defined as we go along. So, whenever you look at an optimization problem. So, you should look for these three components in an optimization problem. In cases where this is missing we call this as unconstrained optimization problems, in cases where this is there and we have to have the solution satisfy these constraints we call them as constrained optimization problems.

(Refer Slide Time: 14:08)



Now, depending on the type of objective function, type of constraints and the type of decision variables, we will explain what each one of these are, there are different types of optimization problems that we could solve.

For example, if we have the following f(x) subject to some constraints that we are going to impose and if it turns out that this X we use them as continuous variables. What do we mean by continuous variables? These are variables that can take values within a certain range. So, you could say if you have one variable you could say the variable could be between - 2 and 2 for example, or you could simply say it could be any number in the real line then these are condensed variables. What it basically means is within this range I can take any value there is no restriction on the value right X. So, these are continuous variables. So, if you have continuous variables like this, and if the functional form of this f is linear and all the constraints are also linear then I have a type of problem called linear programming problem.

So, in this case the variables are continuous, the objective is linear and the constraints are also linear. Now, if the variable remains continuous however, if either the objective function or the constraints are non-linear functions, then we have what is called a nonlinear programming problem. So, a programming problem becomes non -linear if either the objective or the constraints become non-linear. In general people used to think non-linear programming problems are much harder to solve than linear programming problems which is true in some cases, but really the difficulty in solving non-linear programming problems is mainly related to this notion of convexity. So, whether a non-linear programming problem is convex or non convex is an important idea in identifying how difficult the problem is to solve.

So, this idea of convex and non convex very very briefly without too much detail we will see in the next few slides nonetheless I just wanted to point this out here and also wanted to describe the second type of optimization problem that is of interest which is the nonlinear programming problem.

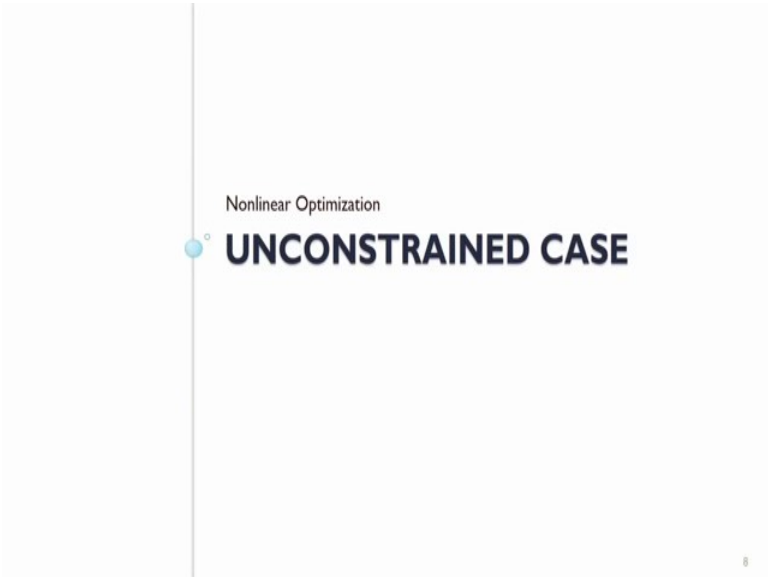
Till now, we have just been talking about the types of objective functions and constraints however, we have always assumed that the decision variables are continuous. In many cases we might want the decision variable not to be continuous, but to be integers. So, for example, I could have an optimization problem where I have f as a function of let us say two variables x_1 and x_2 and I could say minimize this. Now, I could say x_1 is not continuous, but x_1 has to take a value let us say from this integer set µ 0 1 2 3 so on, and x_2 maybe has to also take a value in this set. So, this is called a integer programming problem.

And you could have constraints also on x constraints on x_1 and x_2 could also be there. And if the objective function and constraints are linear then we call this linear integer programming problem, if either the objective or the constraints become non-linear we call them non-linear integer programming problems. One special class of these integer programming problems are binary where if x_1 could only take a value which is 0 or 1 and x_2 could take a value only 0 or 1 we call this as binary integer programming problems.

Now, when you combine variables which are both continuous and integer. So, for example, in this case when I have $f(x_1 , x_2)$ let us say X has to take a value 0 1 2 3 whereas, x_2 is continuous it can take any value let us say within a range then we have what are called mixed programming problems and if both the constraints, and the objective are linear then we have mixed integer linear programming problem and if either the constraints or the objective become non-linear then we have mixed integer non-linear programming problem. So, these are the various types of problems that are of interest.

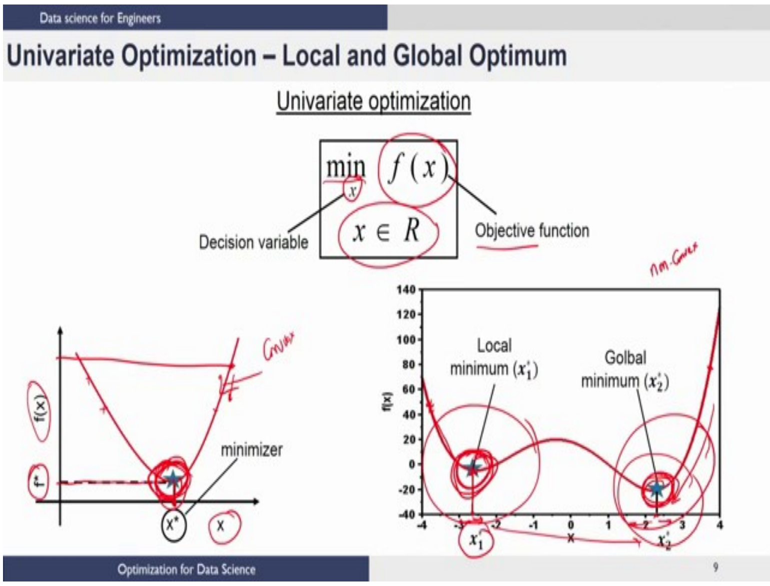
Now, these types of problems have been solved and are of large interest in almost all engineering disciplines. So, we typically solve these problems in for example, in chemical engineering we solve these types of problems routinely for optimizing. Let us say refinery operations or designing optimal equipment and so on, and similarly in all engineering disciplines these optimization problems are used quite heavily. From these lectures viewpoint what we want to point out is to show how we can understand some of these optimization problems and how they are useful in the field of data science.

(Refer Slide Time: 20:48)



So, I am going to start with the simple case of a non-linear optimization problem unconstrained case that is there are no constraints.

(Refer Slide Time: 20:57)



Let us start with a very simple unconstrained optimization problem called an univariate optimization problem and in this slide I am going to explain this univariate optimization problem and the ideas of local and global optimum. So, what do we mean by univariate? When we say it is a univariate optimization problem there is only one decision variable that we are trying to find a value for.

So, when you look at this optimization problem you typically write it in this form where you say I am going to minimize something, this function here, and this function is called the objective function. And the variable that you can use to minimize this function which is called the decision variable is written below like this here x and we also say x is continuous, that is it could take any value in the real number line. And since this is a univariate optimization problem x is a scalar variable and not a vector variable. And whenever we talk about univariate optimization problems, it is easy to visualize that in a 2-D picture like this. So, what we have here is in the x axis we have different values for the decision variable x and in the y axis we have the function value. And when you plot this you can quite easily notice that, this is the point at which this function right here attains its minimum value.

So, the point at which this function attains minimum value can be found by dropping a perpendicular onto the x axis. So, this is actual value of x at which this function takes a minimum value and the value that the function takes at its minimum point can be identified by dropping this perpendicular onto the y axis and this f * is the best value this function could possibly take. So, functions of this type are called convex functions because there is only one minimum here. So, there is no question of multiple minima to choose from. There is only one minimum here and that is given by this.

So, in this case we would say that this minimum is both a local minimum and also a global minimum. We say it is a local minimum because in the vicinity of this point this is the best solution that you can get. And if the solution that we get the best solution that we get in the vicinity of this point is also the best solution globally then we also call it the global minimum.

Now, contrast that with the picture that I have on the right hand side. Now, here I have a function and again it is a univariate optimization problem. So, on the x I have different values of the decision variable on y axis we plot the function. Now, you notice that there are two points where the function attains a minimum and you can see that when we say minimum we automatically actually only mean locally minimum because if you notice this point here in the vicinity of this point this function cannot take any better value from a minimization viewpoint. In other words if I am here and the function is taking this value if I move to the right the function value will increase which basically is not good for us because we are trying to find minimum value, and if I move to my left the function value will again increase which is not good because we are finding the minimum for this function.

What this basically says is the following. This says that in a local vicinity you can never find a point which is better than this. However, if you go far away then you will get to this point here which again from a local viewpoint is the best because if I go in this direction the function increases and if I go in this direction also the function increases, and in this particular example it also turns out that globally this is the best solution. So, while both are local minimum in the sense that in the vicinity they are the best this local minimum is also global minimum because if you take the whole region you still cannot beat this solution.

So, when you have a solution which is the lowest in the whole region then you call that as a global minimum. And these are types of functions which we call as non convex functions where there are multiple local optima and the job of an optimizer is to find out the best solution from the many optimum solutions that are possible.

Now, I just want to make a connection to data science here. Now, this problem of finding the global minimum has been a real issue in several data science algorithms. For example, in the 90s there was a lot of excitement and interest about neural networks and so on, and for a few years lot of research went into neural networks and in many cases it turned out that finding the globally optimum solution was very difficult and in many cases these neural networks trained to local optima which is not good enough for the type of problems that were that being solved.

So, that became a real issue with the notion of neural networks and then in the recent years this problem has been revisited and now there are much better algorithms, and much better functional forms, and much better training strategies, so that you can achieve some notion of global optimality and that is reason why we have these algorithms make a comeback and be very useful.

So, this very simple concept of local and global optimization is a very important challenge in many data science algorithms and we will see those later. I just also want to point out why this becomes a challenge. This becomes a challenge because when you run a data science algorithm depending on where you start the algorithm you will get different solutions if the problems are non-convex.

So, in other words whenever you solve an optimization problem as we will see later, you will start with some initial point and try to keep improving your function value by changing the value of your decision variable. So, for example, if you started here for this problem and the function value is something like this you know that if you want to improve your function value that is it since you are minimizing you want to reduce your function value you have to keep going in this direction. And what will happen is ultimately you will get to this point and then say I cannot improve my objective function anymore. So, this is the best solution that is possible.

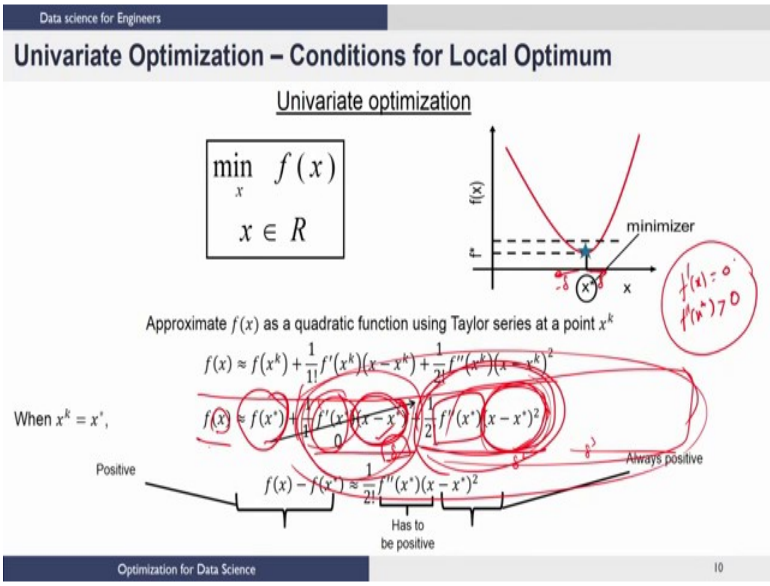
This is how most optimization algorithms work. An important thing to notice here is the respective of whether you start here or here or here or here you are likely to go here depending on your algorithm, you can go there quicker you can go the slower and so on nonetheless whatever is your initialization you are likely to get to the same solution. So, in other words when this optimization algorithm is the backbone of your data science algorithm every time you run the data science algorithm you will get the same solution.

However, notice this picture right here for example, if I started here I want a better my function. So, I will keep improving it and when I come here there is no way to improve it any further. So, I might call this as my best solution and then the data science algorithm will converge. However, if I start here then I would more likely end up here and then I will say this is the best I get and I will stop my data science algorithm.

Now, notice what happens in this case if your data science algorithm is trying to find a value for the decision variable, when you run this once with this initialization you might get this as a solution to your problem, and when you run with this initialization you might get this as a solution to this problem. In other words the algorithm will not give you the same result consistently and more importantly if it is very difficult to find this most of the time your algorithm will give you result which is local minimum, in other words you could do much better, but you are not able to find the solution that does much better.

So, this is an important concept and that you want to understand later when we show you data science algorithms and show you several runs of the same data science algorithm you get several results you might wonder why that is happening and that is due to this problem of initialization.

(Refer Slide Time: 30:48)



Now, let us look at conditions for value to be a minimizer. So, we take the same problem minimize f(x), x element of R. Now, these are conditions that you have seen many times before I am just going to quickly describe how we derive these conditions.

So, if I take f(x) and then let us say I am at a particular point xk, what I can do is I can do a taylor series approximation of this function which we would have seen before in high school and so on. So, let us say this x* is the minimum point and let us see what happens to this Taylor series approximation around this point. So, I am going to say this function f(x) can be approximately written as f(x*) + these. Now, if you notice this expression right here this is a number because this is an x* that I know. So, I simply evaluate f at that x*. So, this is a number, so this is not a function of this x. However, the second term and third term and so on we will all be functions of x. In other words if I change x these are the terms that will change this will remain the same.

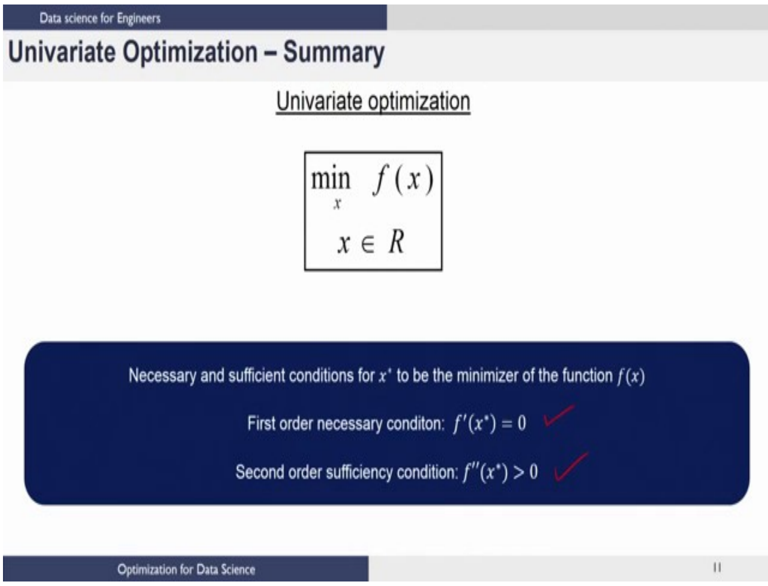
Now, you could see that if you look at this term this is x - x* if you look at this term this is x - x* square and so on. In this univariate case let us call this as δ. So, if I go a δ distance from this - δ here, let us call this x - x* δ. So, if I go in the positive direction I will have a δ, I will have a δ^2, I will have δ^3 and so on. Now, this is a fixed number let us look at this sum of these terms. Now, if you keep reducing δ to smaller and smaller values this is what we explained when we said we are looking at it locally. So, at some point what will happen is δ will become so small that none of these terms will matter the sign of the whole sum will be only depending on this term here. So, if this term is positive this whole sum will be positive and if this term is negative the whole sum will be negative.

Now, you notice this and then if you look at this, if let us say this sign is positive for positive δ then, unfortunately when I go in the negative direction it will become negative because this is again a fixed number if this is positive for δ for - δ this will become negative. That basically means that x* cannot be a minimizer because I can further reduce this function by going to the left.

Now, when δ is positive if this function turns out to be negative then I can go in the to the right and then minimize my function again. So, if this term is not 0 then for sure I will have one direction in which I can go and find a value better than f(x*) locally, which would invalidate our argument that x* is a minimizer. So, basically the only way out for this x* to be minimizer is for this term to be 0 irrespective of x that basically means that f '(x) has to be 0. So, that is the first condition that we usually get f '(x) is 0 and once this is 0 then the Taylor series expansion basically becomes f(x) is f(x*) + the second term third term and so on. By using the same argument when δ becomes smaller and smaller and smaller this term is the only term that will determine the sign of the sum.

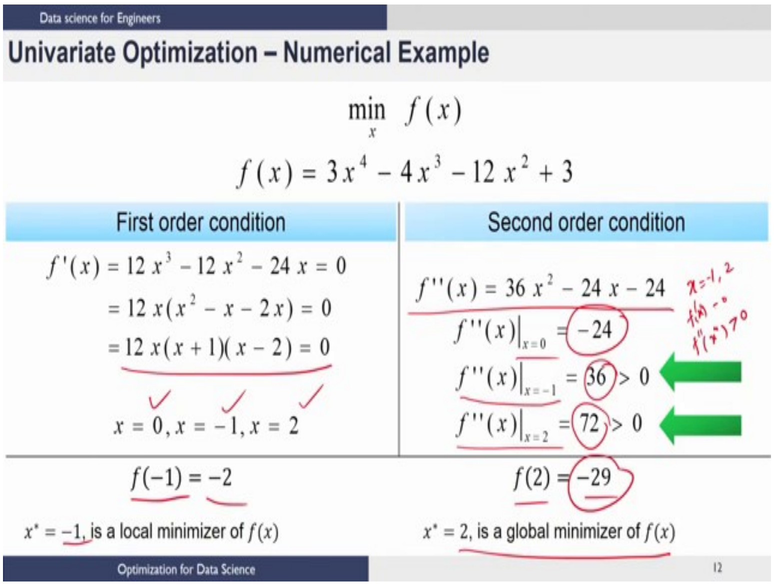
However, notice something very interesting and different here. When we looked at this term this was x - x*, when we look at this term now it is $(x - x^*)^2$. Now, this term, the sign of this term, is dictated only by this quantity here because this is a square and it will always be positive. So, if this f(x) has to be minimized at x* then basically this number f'' (x *) has to be greater than 0 because if this is greater than 0 irrespective of whether you are going to the left or the right this is always positive. So, this will always be a positive contribution; that means, f(x) will always be greater than f(x*) in the local region which should make x* a minimizer. So, that is the important idea in varied optimization and that is the reason why you get these two conditions.

(Refer Slide Time: 35:54)



So, in summary the first order necessary condition as we call it is that the first derivative with respect to x when evaluated at x* has to go to 0. And the second order sufficiency condition as we call it is that then I evaluate the second derivative with respect to x and then evaluated at x* it has to be greater than 0.

(Refer Slide Time: 36:20)



Let us quickly through a see a numerical example to bring all of this ideas together. So, let us take a function f(x) which is of the form $3x^4$ - $4x^3$ - 12 x_2 + 3. Let us first do the first derivative and set it to 0, when we do the first derivative and set it to 0 we get 3 solutions x = 0, x = - 1 and 2.

Now, we want to know which one of this is a minimizer and which one is a local minimizer global minimizer and so on. To do that we look at the second order conditions and then we get f''(x) the second derivative and then we first evaluate it at x = 0. In this case this number turns out to be negative which means that x is a maximum point, not a minimum point. Our interest is in minimization and when we look at this f double prime at - 1 and 2 the only thing we can look for is whether this number is positive or not. The actual numbers do not matter.

So, in this case this is 36 this is 72 in both cases this is greater than 0. So, points x = - 1 and 2 both are minimum points for this function because both of them satisfy the two conditions f '(x*) is 0 and f double prime x* is greater than 0. Now, it is interesting that at this point we cannot say anything more about these two points these numbers do not help we just look whether they are positive or not and of these two points clearly one of them is a local minimum another one is a global minimum. So, the only way to figure out which point is a local minimum which is a global minimum is to actually substitute this into the function and then see what values you get. So, when you substitute - 1 into the function you get - 2 and when you substitute 2 into the function you get - 29. Since we are interested in minimizing the function - 29 is much better than - 2. So, that basically means 2 is a global minimum of this function and - 1 is a local minimizer for f(x).

So, in this lecture we looked at simple univariate unconstrained optimization. And we also looked at why optimization is very important from a data science viewpoint. We will pick up on some of these ideas and then talk about multivariate unconstrained optimization and constrained optimization in the lectures to follow.

Thank you. I will see you again in the next lecture.