Essential Mathematics for Machine Learning

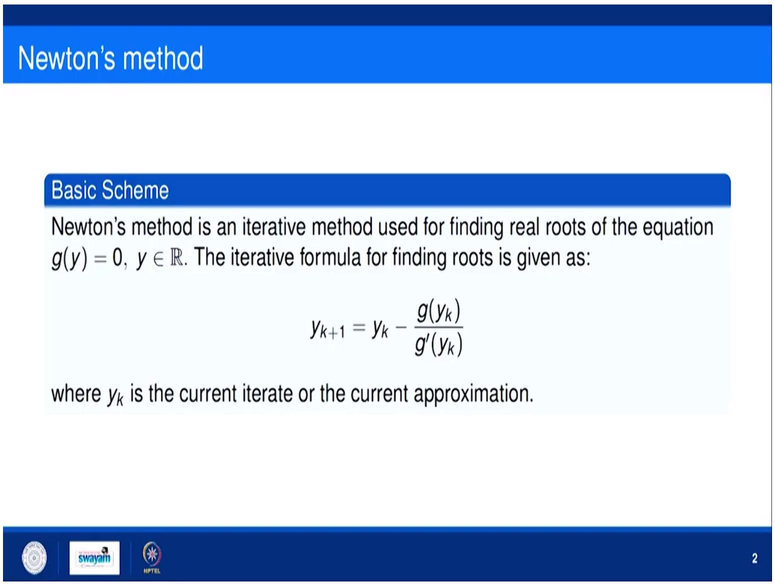
Prof. S. K. Gupta
Department of Mathematics
Indian Institute of Technology, Roorkee

Lecture – 30
Newton's and Penalty Function Methods

Hello friends, welcome to lecture series on Essential Mathematics from Machine Learning. So, in the last lecture we have seen some numerical optimization; some concepts of numerical optimization. We have seen a basic steepest descent method to solve an unconstrained optimization problem.

In this method basically I have already discussed that we have take our direction that is which is negative of gradient of f; in which f decreases most rapidly. And we move from x k to x k plus 1 such that x k plus 1 is nothing, but x k plus alpha k d k; where d k is a direction that is the negative gradient of f at x k and alpha k is optimal step size. In this lecture we will see some more techniques which is required in machine learning.

(Refer Slide Time: 01:19)



Newton's method

Basic Scheme

Newton's method is an iterative method used for finding real roots of the equation $g(y) = 0,\ y \in \mathbb{R}$. The iterative formula for finding roots is given as:

$$y_{k+1} = y_k - \frac{g(y_k)}{g'(y_k)}$$

where y_k is the current iterate or the current approximation.

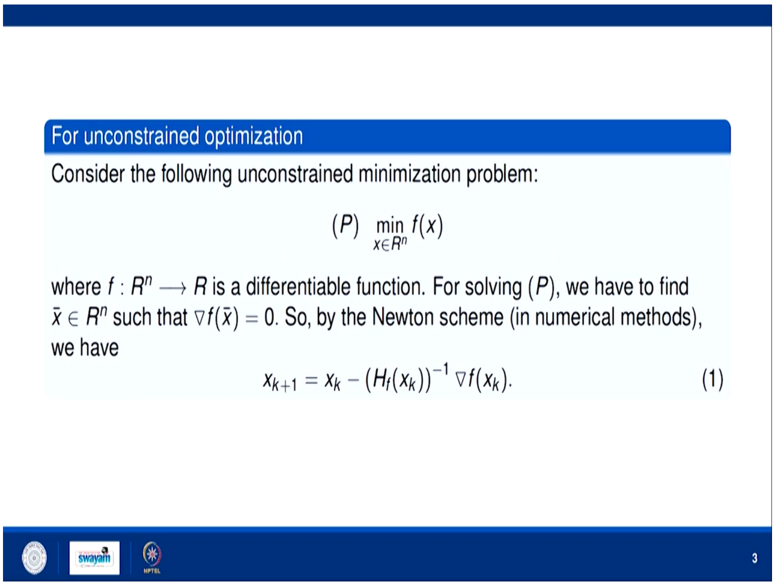
So, first of all Newton's method so what Newton's method is; if we have an equation say y equal to f x or f x equal to 0 we want to solve this equation f x equal to 0 we want to find out the root of this equation. Then how can you find the root of this equation? That we have so many methods in numerical analysis one of them is Newton Raphson method; in which basically here we are having g y equal to 0.

Now, we want to find out root of this equation for which y g y is equal to 0. So, how we can find out? At least approximate root of this equation so that for that we have a recursive algorithm that is given as that this this recursive algorithm is called Newton Raphson method.

So, what this method is basically; we go from one iteration to other iteration in such a way. First we fix our initial guess say that initial guess is y 1 or y 0 you can take y 0 then from y 0 to y 1 if you put k equal to 0 so, that is equal to g y 0 upon g dash y 0.

Of course, g dash y 0 g dash y k should not equal to 0 for any y k this method will work only when g dash y k is not equal to 0 for any y k. So, as this y k plus 1 tends to y k; that means, we are tending towards a solution.

(Refer Slide Time: 02:48)



For unconstrained optimization

Consider the following unconstrained minimization problem:

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

where $f : R^n \longrightarrow R$ is a differentiable function. For solving (P), we have to find $\bar{x} \in R^n$ such that $\triangledown f(\bar{x}) = 0$. So, by the Newton scheme (in numerical methods), we have

$$x_{k+1} = x_k - (H_f(x_k))^{-1} \triangledown f(x_k). \tag{1}$$

Now, if you are if we are having in the same lines if you are having a unconstrained optimization problem; which is minimization of f x subject to x belongs to R n without any restriction on n x, x may be any vector in R n.

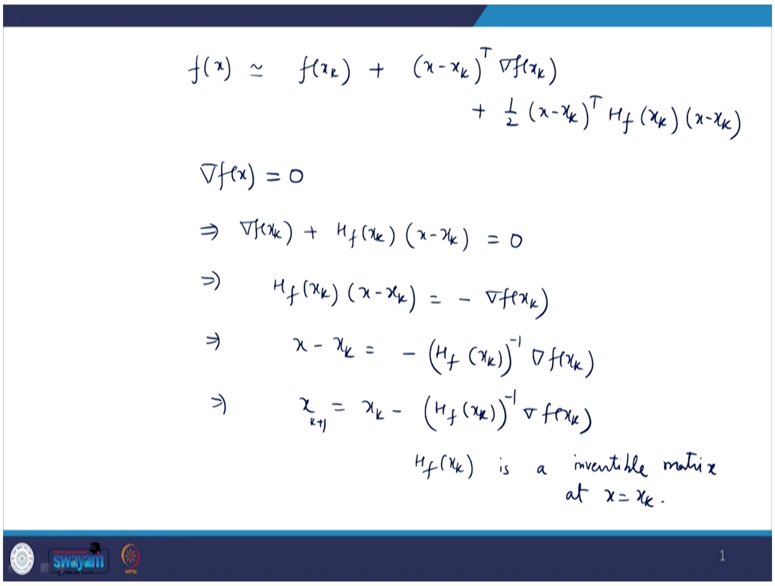
So, how we can find how we can optimize how we can minimize this function f. So, for that now suppose it is given to us as function f is differentiable the differentiable function. Then of course, if you want to maximize if minimize this; that means, we want to find out the root of root of this equation; gradient of f x equal to 0 you want to find out that x bar where gradient equal to 0.

Because if you want to maximize or minimize a function; that means, we have to d y by d x for a single variable function; here it is n variable function. So, for a single variable function how we can find out maxima or minima? We first find d y by d x put it equal to 0 that will give critical points and we find second derivatives we see that where it is maxima or minima or higher order derivatives.

If it is n variable function; so what are those point where it attain minima where gradient is equal to 0. So; that means, instead of solving this problem we have to solve we have to solve this equation we have to solve this actually system of equations; we have to find that x bar where this is equal to 0. So, how can we; how can we solve this equation? How we can find out that x bar where this is equal to 0?

So, that can be find using Newton's scheme and that Newton's scheme is basically it is on the same lines as a Newton Raphson method that is x k plus 1 equal to x k minus Hessian matrix of x k at x k whole inverse into gradient of f x k. Now, how we come how we arrive at this recursive formula how you obtain this? So the derivation is quite easy. So, let us see.

(Refer Slide Time: 04:48)



$$f(x) \simeq f(x_k) + (x-x_k)^T \nabla f(x_k)$$
$$+ \frac{1}{2}(x-x_k)^T H_f(x_k)(x-x_k)$$

$$\nabla f(x) = 0$$

$$\Rightarrow \nabla f(x_k) + H_f(x_k)(x-x_k) = 0$$

$$\Rightarrow \quad H_f(x_k)(x-x_k) = -\nabla f(x_k)$$

$$\Rightarrow \quad x - x_k = -\left(H_f(x_k)\right)^{-1} \nabla f(x_k)$$

$$\Rightarrow \quad x_{k+1} = x_k - \left(H_f(x_k)\right)^{-1} \nabla f(x_k)$$

$H_f(x_k)$ is a invertible matrix at $x = x_k$.

Suppose you approximate this f x by a Taylor series as f x k plus x minus x k whole transpose gradient of f x k plus 1 by 2 x minus x k whole transpose Hessian matrix of f at x k into x minus x k. So, we take a quadratic approximation of this function by the Taylor series expansion.

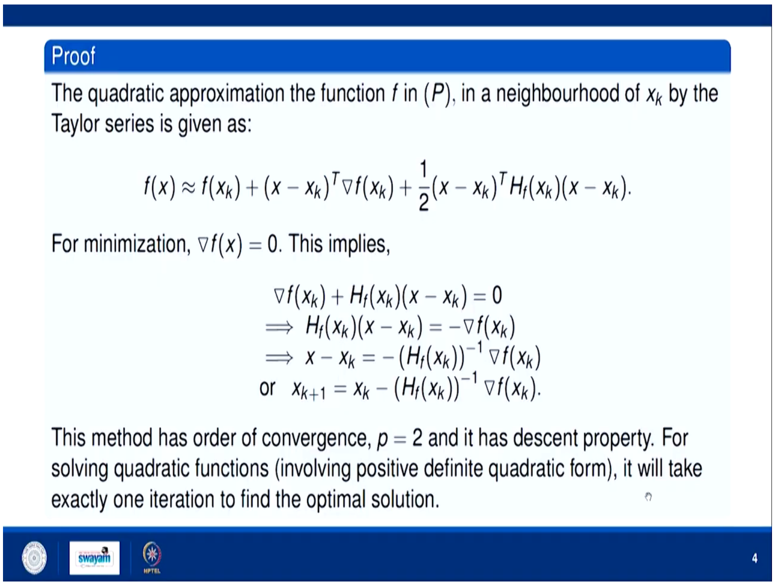
Now, what we want? We want that x where gradient of f x equal to 0 ok, we want this. Now take the gradient of f respect to x both sides and let us see what we will obtain. So, this implies; so this is equal to 0 now this is x k the fix point. So, if this is a fix point; so gradient of this will be 0. Now this is x, x, x into this so when you differentiate this respect to partial differential respect to x. So, that is nothing, but gradient of f x k.

The second term is of course, 0 because x k is fixed and when you differentiate this respect to x k sorry x partially then we will get what? Plus Hessian matrix of f at x k into x minus x k and that is that must be 0. Now this implies Hessian matrix of f at x k into x minus x k is equal to minus of gradient of f x k. And this implies take suppose this is invertible.

Now, if this is invertible; so we can write x minus x k is equals to negative of H f x k whole inverse into gradient of f x k. And this implies x equal to x k minus H f x k; that means, Hessian matrix of f at x k into gradient of f x k. So, this is nothing, but x k plus 1 in the next iteration. So, as soon as this approaches to this x k we say that there that will be the optimal solution of a given problem.

The limitation of this method is that we are we are supposing that Hessian matrix of f at x k is invertible for every x k. If it is not then this method is not applicable ok. So, H f, x k is invertible is a invertible matrix at x equal to x k. So, this this is our supposition.

(Refer Slide Time: 07:39)



Proof

The quadratic approximation the function f in (P), in a neighbourhood of x_k by the Taylor series is given as:

$$f(x) \approx f(x_k) + (x - x_k)^T \nabla f(x_k) + \frac{1}{2}(x - x_k)^T H_f(x_k)(x - x_k).$$

For minimization, $\nabla f(x) = 0$. This implies,

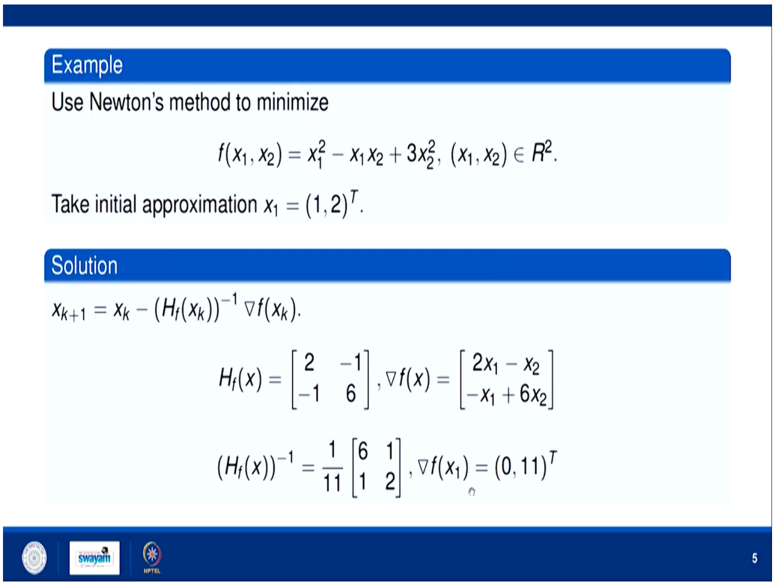
$$\begin{aligned} & \nabla f(x_k) + H_f(x_k)(x - x_k) = 0 \\ \implies & H_f(x_k)(x - x_k) = -\nabla f(x_k) \\ \implies & x - x_k = -\left(H_f(x_k)\right)^{-1} \nabla f(x_k) \\ \text{or} \quad & x_{k+1} = x_k - \left(H_f(x_k)\right)^{-1} \nabla f(x_k). \end{aligned}$$

This method has order of convergence, $p = 2$ and it has descent property. For solving quadratic functions (involving positive definite quadratic form), it will take exactly one iteration to find the optimal solution.

So, basically this I have already explained you. Now this method has the order of convergence 2 that is p is equal to 2 for this method. And it has a descent property that we can easily show. Descent property means f at x k plus 1 is less than f at x k.

For solving quadratic functions involving positive definite quadratic form; it will take exactly one iteration to find out the optimal solution to find the optimal solution exactly one step.

(Refer Slide Time: 08:10)



Example

Use Newton's method to minimize

$$f(x_1, x_2) = x_1^2 - x_1 x_2 + 3x_2^2, \ (x_1, x_2) \in R^2.$$

Take initial approximation $x_1 = (1, 2)^T$.

Solution

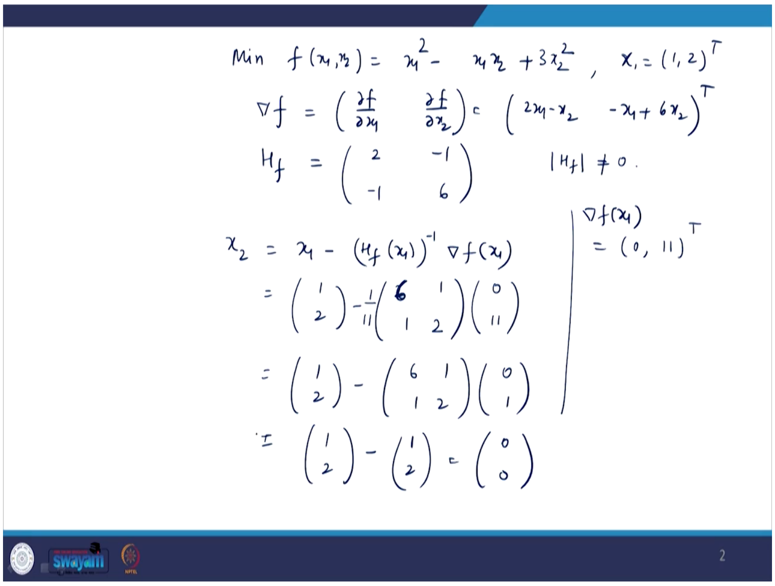
$x_{k+1} = x_k - (H_f(x_k))^{-1} \nabla f(x_k)$.

$$H_f(x) = \begin{bmatrix} 2 & -1 \\ -1 & 6 \end{bmatrix}, \nabla f(x) = \begin{bmatrix} 2x_1 - x_2 \\ -x_1 + 6x_2 \end{bmatrix}$$

$$(H_f(x))^{-1} = \frac{1}{11} \begin{bmatrix} 6 & 1 \\ 1 & 2 \end{bmatrix}, \nabla f(x_1) = (0, 11)^T$$

So, let us see one problem based on this; suppose this is a quadratic expression which you want to minimize it is a unconstrained optimization problem. And suppose initial guess is 1 2 you can take other initial guess also; I am take a I am taken initial guess as 1 comma 2. So, how we can find out the optimal solution of this problem using Newton's method?

(Refer Slide Time: 08:36)



$$\text{Min } f(x_1, x_2) = x_1^2 - x_1 x_2 + 3x_2^2 \ , \quad x_1 = (1, 2)^T$$

$$\nabla f = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \right) = \left(2x_1 - x_2 \quad -x_1 + 6x_2 \right)^T$$

$$H_f = \begin{pmatrix} 2 & -1 \\ -1 & 6 \end{pmatrix} \qquad |H_f| \neq 0 .$$

$$\nabla f(x_1) = (0, 11)^T$$

$$x_2 = x_1 - (H_f(x_1))^{-1} \nabla f(x_1)$$

$$= \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \frac{1}{11} \begin{pmatrix} 6 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 11 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 6 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

So, what is the problem we are having? So, the minimization of f x 1, x 2 which is equal to x 1 square minus 3 x 1, x 2 plus x 2 square 3 x 2 square oh sorry it is 3 x 2 square. And initial guess is 1 comma 2 transverse; so this is the initially guess we are having.

So, first you find gradient of f; gradient of f we del f upon del x 1 del upon del x 2 that will be 2 x 1 minus x 2 and minus x 1 plus 6 x 2 this transpose will be the gradient of f.

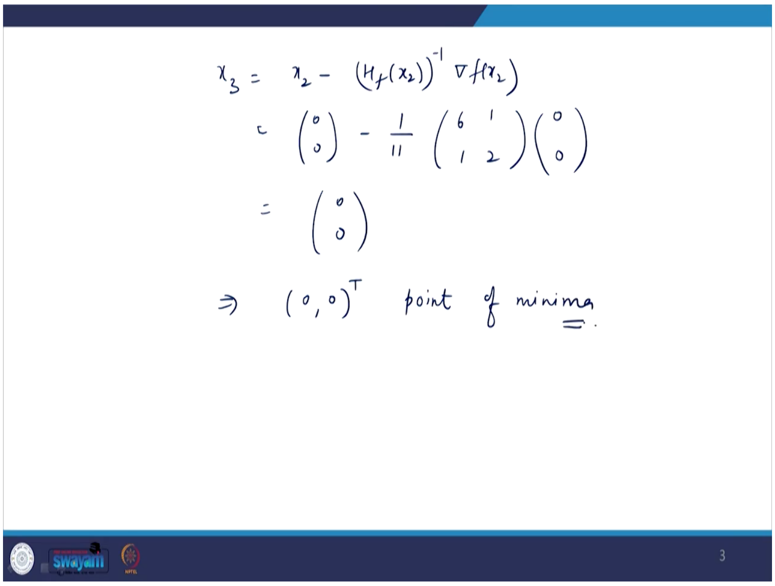
Now, what is Hessian matrix of f? This is 2 minus 1 minus 1, 6 ok. Now this is always invertible this is 12 this is a determinant of h f is always non 0 so for any x k because it is independent of x ok. If you want x 2 x 2 means x 1 minus Hessian matrix of f at x 1 whole inverse gradient of f at x 1 this is why Newton's method Newton's formula.

So, what is gradient of f at x 1? Now gradient of f at x 1; x 1 is x 1 is this this is x 1 basically x 1 is 1, 2. Now, you substitute 1 and 2 so, it is 0 we substitute 1 here 2 here so it is 11 ok. So, now, x 1 is 1 2 minus Hessian matrixes. Now you have to find the inverse of this matrix. So, what is the inverse of this matrix? Inverse of this matrix will be 6, 2 and this is again I think minus 1 minus 1 or 1, 1.

So, it is 1, 1 it is 1, 1 and that is divided by the determinant of this which is 1 upon 11 determinant is 11 here into gradient of f at x 1 that is 0 11. So, this is 1, 2 minus it is 6, 1, 1, 2 and that will be 0, 1. So, this is 1, 2 minus this row this column is 1, this row this column is 2, and that is simply 0, 0. So, x 2 comes out to be 0, 0.

Now if you find x 3.

(Refer Slide Time: 11:17)



$$x_3 = x_2 - \left(H_f(x_2)\right)^{-1} \nabla f(x_2)$$

$$= \begin{pmatrix}0\\0\end{pmatrix} - \frac{1}{11}\begin{pmatrix}6 & 1\\1 & 2\end{pmatrix}\begin{pmatrix}0\\0\end{pmatrix}$$

$$= \begin{pmatrix}0\\0\end{pmatrix}$$

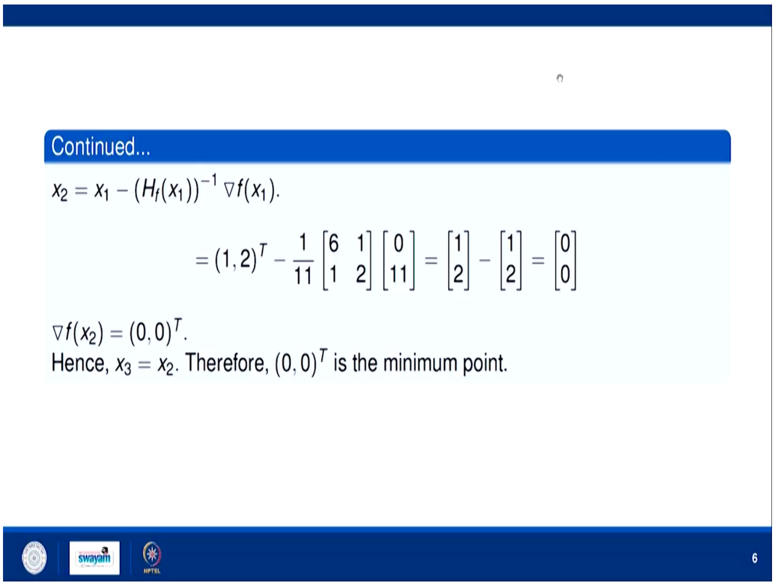
$\Rightarrow (0,0)^T$ point of $\underline{\text{minima}}$.

So, what will be x 3? x 3 will be x 2 minus Hessian matrix of x 2 whole inverse gradient of f x 2. So, x 2 is 0, 0 this is this is same which is 6, 2, 1, 1 and what is gradient of f x 2 at 0, 0? At 0, 0 it is; obviously, 0.

So, when gradient see when gradient of f at x 2 comes out to be 0; that means, it is an point of minima. So, of course, that will a point of minima that which we can verify from here also; so this is 0, 0 so; that means, 0, 0 is the point of minima; it is a point of minima ok.

So, since this this is positive definite form so this method converges only in; one iteration.

(Refer Slide Time: 12:19)



Continued...

$x_2 = x_1 - (H_f(x_1))^{-1} \nabla f(x_1).$

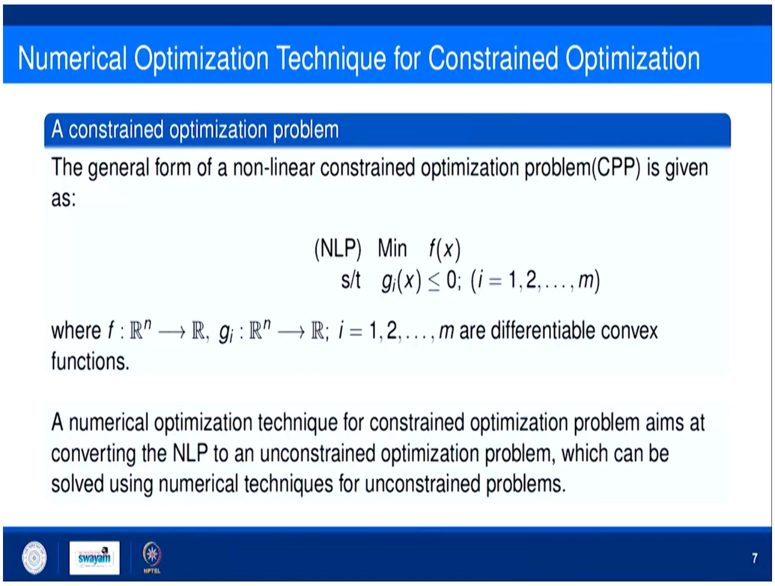
$$= (1,2)^T - \frac{1}{11}\begin{bmatrix}6 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}0\\ 11\end{bmatrix} = \begin{bmatrix}1\\ 2\end{bmatrix} - \begin{bmatrix}1\\ 2\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}$$

$\nabla f(x_2) = (0,0)^T.$

Hence, $x_3 = x_2$. Therefore, $(0,0)^T$ is the minimum point.

So, this is Newton's method for solving such type of problems. Now how can we solve a constrained optimization model? So, here I am discussing one method that is penalty method that; how we can discuss a constrained optimization problem.

(Refer Slide Time: 12:35)



Numerical Optimization Technique for Constrained Optimization

A constrained optimization problem

The general form of a non-linear constrained optimization problem(CPP) is given as:

$$\begin{aligned}(\text{NLP})\quad & \text{Min} \quad f(x)\\ \text{s/t} \quad & g_i(x) \leq 0;\ (i = 1, 2, \ldots, m)\end{aligned}$$

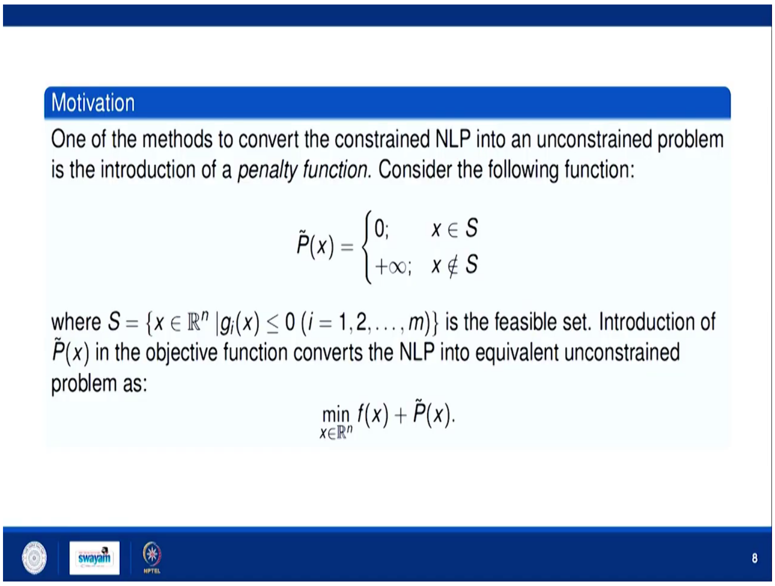
where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$; $i = 1, 2, \ldots, m$ are differentiable convex functions.

A numerical optimization technique for constrained optimization problem aims at converting the NLP to an unconstrained optimization problem, which can be solved using numerical techniques for unconstrained problems.

Now, suppose you are having a non-linear programming problem like this; that is minimization of f x subject to g i x less than equal to 0 i from 1 to m and suppose this f and all g is are differentiable functions. So, a numerical optimization technique for constrained optimization problem aims at converting the NLP to an unconstrained optimization problem which can be solved using numerical technique for unconstrained optimization problem.

If you are having any constrained optimization problem. The main aim of any algorithm first for handling constrained optimization problem is to convert that problem into unconstrained type. If we can convert that problem into unconstrained type, then the usual technique like; descent method or other methods we can apply for solving that problem.

(Refer Slide Time: 13:27)



Motivation

One of the methods to convert the constrained NLP into an unconstrained problem is the introduction of a *penalty function*. Consider the following function:

$$\tilde{P}(x) = \begin{cases} 0; & x \in S \\ +\infty; & x \notin S \end{cases}$$

where $S = \{x \in \mathbb{R}^n \,|g_i(x) \leq 0\,(i = 1, 2, \ldots, m)\}$ is the feasible set. Introduction of $\tilde{P}(x)$ in the objective function converts the NLP into equivalent unconstrained problem as:

$$\min_{x \in \mathbb{R}^n} f(x) + \tilde{P}(x).$$

8

So, one such method is penalty function method. Now what is the penalty function in penalty function what we do? We define a function P x P tilde x which is which we say that it is 0 if x belongs to S here S is a feasible region.

So, S is the feasible region which is consisting of all x such that g i x less than equal to 0 for all i. And if we are saying that it is infinity if x does not belongs to S; does not belongs to x. So, of course, if you minimize this function; if you minimize this function; that means, if x belongs to S then this will be 0, if x belongs to S; that means, x is the feasible region x is the feasible point and then this will be a 0.

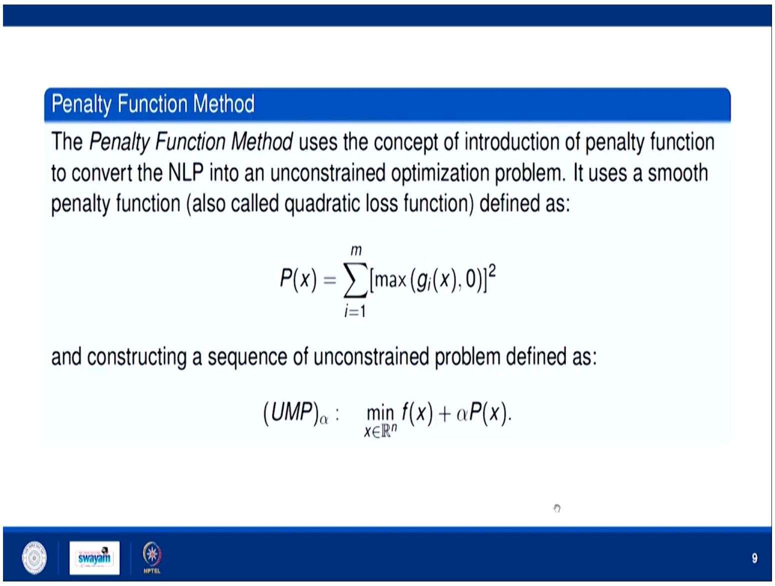
Then minimum of this function and minimum of this function will coincide. That means, NLP the problem NLP and this modified unconstrained optimization problems are equivalent. And if x does not belongs to S then this will tends to infinity.

So, the main aim is this will a minimum f this belongs to if this is 0 and that is possible when x is in feasible region; that means, we are finding a feasible point where we are minimizing simultaneously f x. So, this this unconstrained optimization model is basically equivalent to the problem NLP.

Now the problem here is that this function is not smooth not differentiable it is discontinuous. So, if we want if we have converted this problem into an unconstrained optimization model; if you want to apply the techniques for the numerical techniques for solving unconstrained optimization model then that those methods may not be applicable.

So, how we can make this problem as a smooth problem I mean differentiable problem; so that the techniques of unconstrained optimization problem may work.

(Refer Slide Time: 15:32)



Penalty Function Method

The *Penalty Function Method* uses the concept of introduction of penalty function to convert the NLP into an unconstrained optimization problem. It uses a smooth penalty function (also called quadratic loss function) defined as:

$$P(x) = \sum_{i=1}^{m} [\max(g_i(x), 0)]^2$$

and constructing a sequence of unconstrained problem defined as:

$$(UMP)_\alpha : \quad \min_{x \in \mathbb{R}^n} f(x) + \alpha P(x).$$

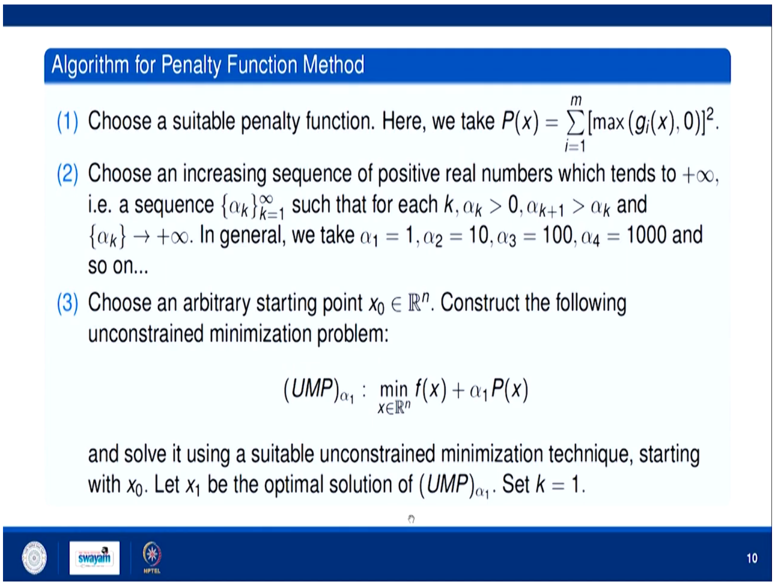
So, what we what we do ? We defined a function P x the penalty function P x as maximization maximum of g i x and 0 this whole square and sum up to i from 1 to m. Of course again if we are minimizing f x plus alpha times P x.

See if x is in S; S means feasible region. If x is an feasible region; that means, g i x is less than equal to 0 for all I, if g i x is less than equal to 0 for all i then the maximum of g i x and 0 will be nothing, but 0 itself. And the sum of squares of 0 is 0 so; that means, this will be 0; that means, it will comes to minimum of f x.

The same problem; that means, feasible x such that minimum of f x; that means, finding the optimal solution of this unconstrained optimization model is same as finding an optimal solution of the constrained optimization problem NLP. Why we are constructed penalty function like this? We have we can construct there are other methods also to find the penalty function this is one of them to make the function smooth ok.

Here we attach alpha also alpha is a sequence of alpha is the sequence basically; you can take alpha as 1, alpha as 10, alpha as 100; we increase alpha we tend it to infinity basically. We tend into infinity; so that f x will approach to that so that this function will approach to f x.

(Refer Slide Time: 17:08)



Algorithm for Penalty Function Method

(1) Choose a suitable penalty function. Here, we take $P(x) = \sum_{i=1}^{m} [\max(g_i(x), 0)]^2$.

(2) Choose an increasing sequence of positive real numbers which tends to $+\infty$, i.e. a sequence $\{\alpha_k\}_{k=1}^{\infty}$ such that for each $k, \alpha_k > 0, \alpha_{k+1} > \alpha_k$ and $\{\alpha_k\} \to +\infty$. In general, we take $\alpha_1 = 1, \alpha_2 = 10, \alpha_3 = 100, \alpha_4 = 1000$ and so on...

(3) Choose an arbitrary starting point $x_0 \in \mathbb{R}^n$. Construct the following unconstrained minimization problem:

$$(UMP)_{\alpha_1} : \min_{x \in \mathbb{R}^n} f(x) + \alpha_1 P(x)$$

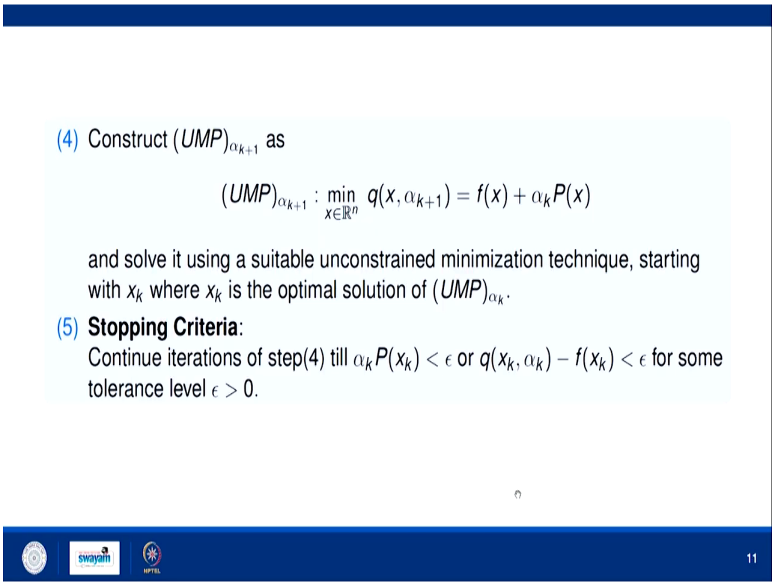
and solve it using a suitable unconstrained minimization technique, starting with x_0. Let x_1 be the optimal solution of $(UMP)_{\alpha_1}$. Set $k = 1$.

So, so this is the mean algorithm basically; what is an algorithm? First of all we define a suitable penalty function; here we have taken penalty function as sum of maximum of g i x and 0 whole square. Then choose an increasing sequence of positive real numbers which tends to infinity that is sequence alpha k, k from 1 to infinity, such that; alpha k greater than 0 and alpha k plus 1 is greater than alpha k. In general we take alpha 1 as first 1, alpha 2 as 10, alpha 3 as 100, alpha 4 as 1000 so on.

Then we choose arbitrary starting point x naught in R n and construct the following unconstrained minimization problem as this ok. And solve it using unconstrained minimization technique you can use steepest descent method or you can use some other technique for solving this unconstrained optimization model.

Now, if x 1 be is the optimal solution of this for alpha 1; then taking that x 1 as an initial S put here alpha 2 and find again the optimal solution of that unconstrained optimization model using any unconstrained minimization technique.

(Refer Slide Time: 18:18)



(4) Construct $(UMP)_{\alpha_{k+1}}$ as

$$(UMP)_{\alpha_{k+1}} : \min_{x\in\mathbb{R}^n} q(x,\alpha_{k+1}) = f(x) + \alpha_k P(x)$$

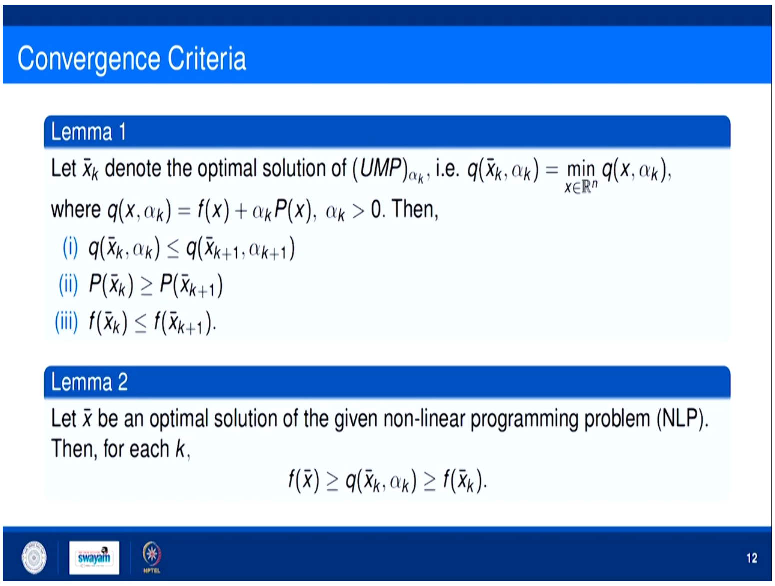
and solve it using a suitable unconstrained minimization technique, starting with x_k where x_k is the optimal solution of $(UMP)_{\alpha_k}$.

(5) **Stopping Criteria**:
Continue iterations of step(4) till $\alpha_k P(x_k) < \epsilon$ or $q(x_k,\alpha_k) - f(x_k) < \epsilon$ for some tolerance level $\epsilon > 0$.

The process will go on till we have either alpha k P k P x k less than epsilon; that means, this is very small penalty function is very small or this q x alpha k minus f x k is very small which one and the same thing basically for some tolerance level alpha epsilon greater than 0.

So, this is the main idea behind this these are main algorithm behind penalty function method.

(Refer Slide Time: 18:51)



Convergence Criteria

Lemma 1

Let $\bar{x}_k$ denote the optimal solution of $(UMP)_{\alpha_k}$, i.e. $q(\bar{x}_k, \alpha_k) = \min_{x \in \mathbb{R}^n} q(x, \alpha_k)$, where $q(x, \alpha_k) = f(x) + \alpha_k P(x)$, $\alpha_k > 0$. Then,

(i) $q(\bar{x}_k, \alpha_k) \leq q(\bar{x}_{k+1}, \alpha_{k+1})$

(ii) $P(\bar{x}_k) \geq P(\bar{x}_{k+1})$

(iii) $f(\bar{x}_k) \leq f(\bar{x}_{k+1})$.

Lemma 2

Let $\bar{x}$ be an optimal solution of the given non-linear programming problem (NLP). Then, for each k,

$$f(\bar{x}) \geq q(\bar{x}_k, \alpha_k) \geq f(\bar{x}_k).$$

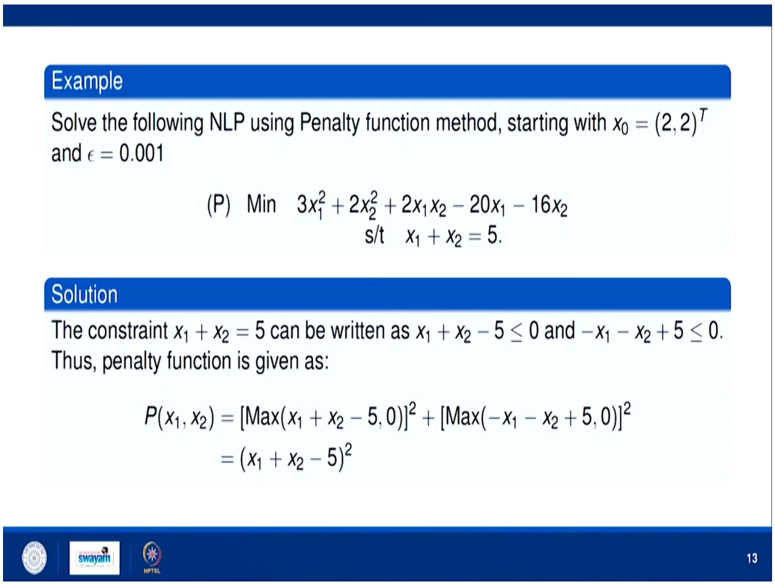
12

How we can say that this is convergent? So, we have various theorems also we are not going to discuss the proof of the theorem; so I am just stating here for your understanding. So, what is the first lemma or the theorem is; let alpha bar let x k bar denote the optimal solution of UMP at alpha equal to alpha k UMP k, UMP is Unconstrained of Minimization Problem.

This is this problem ok; that is this equal to this that is this minimum is attained at x equal to x bar x k bar where q x alpha k is this alpha k greater than 0. Then the first of all at x k plus 1 whatever we have obtained and at alpha k plus 1 this is always greater than equal to this term; that means, this quantity is keep on increasing.

Now, P k x k x k bar is greater than equal to P k x k plus 1 and x k is always less than equal to f k plus 1; that means, this is this we are going to minimize. Now the second lemma is if x let x bar be an optimal solution of the given Non-linear Programming problem N L P then for each k this inequality holds; that means, we are always we are; that means, at x k plus 1 it is always less than equal to f x f x bar where x bar is an optimal solution.

(Refer Slide Time: 20:22)



Example

Solve the following NLP using Penalty function method, starting with $x_0 = (2,2)^T$ and $\epsilon = 0.001$

$$\text{(P)} \quad \text{Min} \quad 3x_1^2 + 2x_2^2 + 2x_1x_2 - 20x_1 - 16x_2$$
$$\text{s/t} \quad x_1 + x_2 = 5.$$

Solution

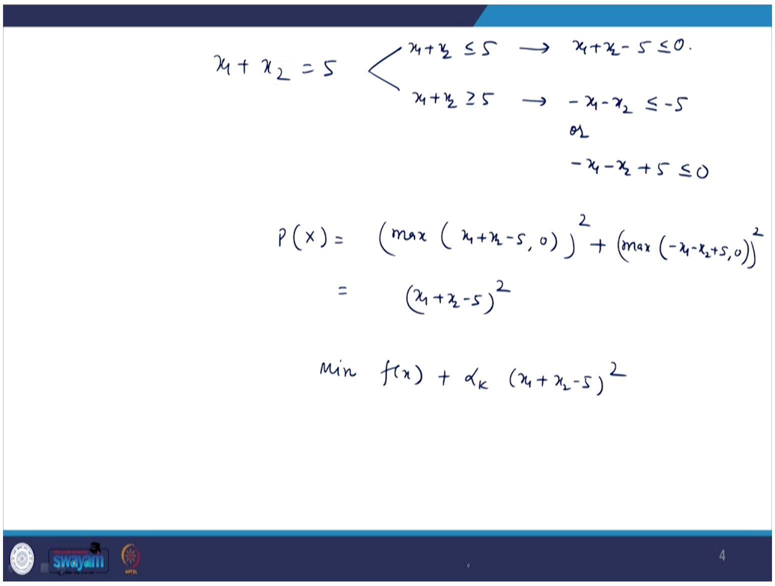
The constraint $x_1 + x_2 = 5$ can be written as $x_1 + x_2 - 5 \leq 0$ and $-x_1 - x_2 + 5 \leq 0$. Thus, penalty function is given as:

$$\begin{aligned} P(x_1, x_2) &= [\text{Max}(x_1 + x_2 - 5, 0)]^2 + [\text{Max}(-x_1 - x_2 + 5, 0)]^2 \\ &= (x_1 + x_2 - 5)^2 \end{aligned}$$

So, let us discuss this example quickly by this problem. We have to solve the NLP using penalty function method starting with x naught equal to 2, 2 and epsilon equal to point 0 0 1. So, let us see what is the problem.

The problem here is we have to minimize 3 x 1 is square plus 2 x 2 square plus 2 x 1 x 2 minus 20 x 1 minus 16 x 2 subject to x 1 plus x 2 equal to 5.

(Refer Slide Time: 20:50)



$$x_1 + x_2 = 5 \quad \begin{cases} x_1+x_2 \le 5 \rightarrow x_1+x_2-5 \le 0. \\ x_1+x_2 \ge 5 \rightarrow -x_1-x_2 \le -5 \end{cases}$$

or

$$-x_1-x_2+5 \le 0$$

$$P(x) = \left(\max\left(x_1+x_2-5, 0\right)\right)^2 + \left(\max\left(-x_1-x_2+5, 0\right)\right)^2$$

$$= (x_1+x_2-5)^2$$

$$\min f(x) + \alpha_K (x_1+x_2-5)^2$$

Now, how we can use penalty function here? x 1 plus x 2 equal to 5 can be written as x 1 plus x 2 less than equal to 5 and x 1 plus x 2 greater than equal to 5. This can be written as minus x 1 minus x 2 less than equal to minus 5 or minus x 1 minus x 2 plus 5 less than equal to 0 and this is basically x 1 plus x 2 minus 5 less than equal to 0.

So, how can we define penalty function now? The penalty function will be defined in the here as maximum of x 1 plus x 2 minus 5 the first constraint 0 whole square plus the second constraint maximum of the second constraint 0 and whole square; that is the penalty function.

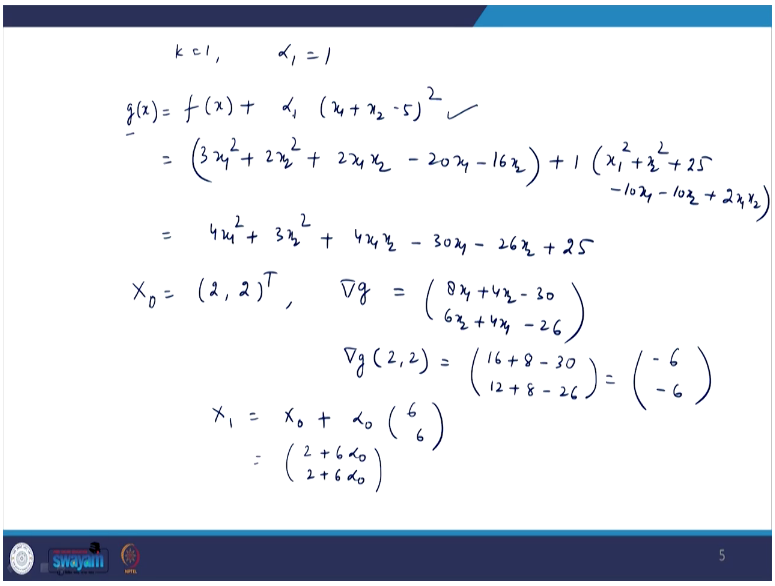
Now, if this is negative if this is suppose this is negative then maximum will be 0 from here and then this will be positive then this is the maximum. And if this is negative then maximum is 0 here and this is positive in either case in any case we are having this as a penalty function ok.

Because if this is negative then we will get 0 here, but on the in the on the same if this is negative then this will be positive. Then the maximum of these two will be this only this is square and if this is negative then from here it is 0 and from here it is x 1 plus x 2 minus 5.

So, in any case we will get the penalty function as x 1 plus x 2 minus 5 whole square. So, what will be our unconstrained problem now? Minimization of f x plus alpha k into x 1 plus x 2 minus 5 whole square; this will be the unconstrained minimization problem which we are having.

Now let us put alpha k equal to like k equal to 1.

(Refer Slide Time: 23:01)



$k=1, \qquad \alpha_1 = 1$

$$g(x) = f(x) + \alpha_1 (x_1 + x_2 - 5)^2 \checkmark$$

$$= (3x_1^2 + 2x_2^2 + 2x_1x_2 - 20x_1 - 16x_2) + 1(x_1^2 + x_2^2 + 25 - 10x_1 - 10x_2 + 2x_1x_2)$$

$$= 4x_1^2 + 3x_2^2 + 4x_1x_2 - 30x_1 - 26x_2 + 25$$

$X_0 = (2, 2)^T, \qquad \nabla g = \begin{pmatrix} 8x_1 + 4x_2 - 30 \\ 6x_2 + 4x_1 - 26 \end{pmatrix}$

$$\nabla g(2,2) = \begin{pmatrix} 16 + 8 - 30 \\ 12 + 8 - 26 \end{pmatrix} = \begin{pmatrix} -6 \\ -6 \end{pmatrix}$$

$$X_1 = X_0 + \alpha_0 \begin{pmatrix} 6 \\ 6 \end{pmatrix} = \begin{pmatrix} 2 + 6\alpha_0 \\ 2 + 6\alpha_0 \end{pmatrix}$$

If you put k equal to 1 in this; take alpha 1 equal to 1 suppose. So, what will be this function? This is say this is g x. So, this is f x plus alpha 1 times x 1 plus x 2 minus 5 whole square.

So, what is f x? f x is given to us as 3 x 1 square plus 2 x 2 square plus 2 x 1 x 2, minus 20 x 1 minus 16 x 2, alpha 1 is 1 into x 1 square plus x 2 square plus 25 minus 10 x 1 minus 10 x 2 plus it is minus 10 x 1 minus 10 x 2 plus 2 x 1 x 2. So, this is nothing but 4 x 1 square plus 3 x 2 square plus 4 x 1 x 2 minus 30 x 1 minus 26 x 2 plus 25.

Now, we have to minimize this we have to minimize this g x; how we can minimize this g x? So, here it is a quadratic form we can directly differentiate it also or we can apply some numerical optimization technique for unconstrained minimization problem. So, we can use steepest descent method also to find optimal solution of this problem.

So, how we can do that? We can take the initial guess as what given to us? Initial guess is 2, 2, 2, 2, is given to us. So, it is x naught which is 2, 2 is given to us. So, you we find gradient of f gradient of f is 8 x 1 plus 4 x 2 minus 30 and here it is 6 x 2 plus 4 x 1 minus 26. And then we find we find gradient of f at here it is g here it is g; so it will be gradient of g. So, gradient of g at we have to find at 2 comma 2.

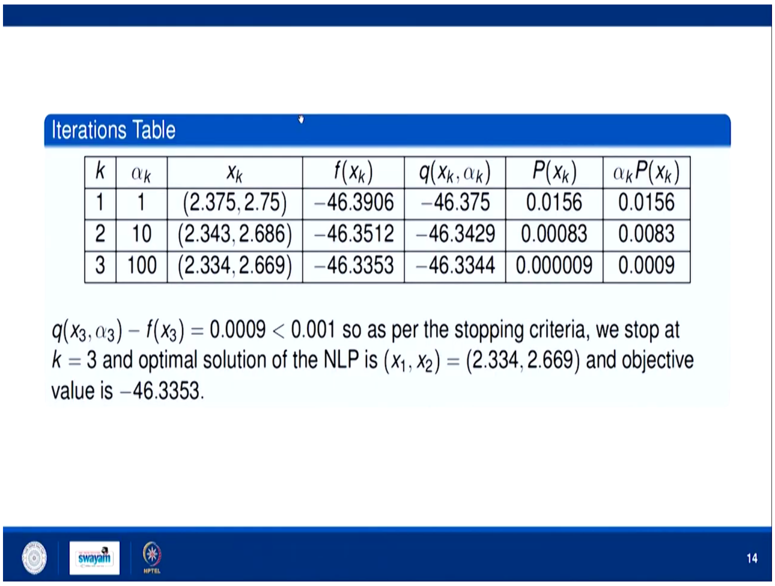
So, that we can find out as; 16 plus 8 minus 30 and it is 12 plus 8 minus 26 and that will be 24 that is minus 12 and that is 20 minus 6; so that will be gradient of g. So, we know the method a method is x 1 equal to x naught plus alpha naught and to minus of gradient of this that is 12 and 6 ok.

So, here it is 30; so it will be 30 itself. So, it is 30 it is 24 minus 30 is minus 6 so it is minus 6. So, here it is comes out to be 6. So, this is now this is 2 plus 6 alpha naught and again 2 plus 6 alpha naught.

So, now, as we do in steepest descent method we will put this x 1 and this x 2 in this function g and we will find out that alpha naught where this attains minima by putting derivative of g respect to alpha naught equal to 0 and the process continued till you get an optimal solution of this problem.

So, what I want to say that basically for once you get a penalty function you define you define your g x; which is unconstrained optimization model like this like this. And changing the value of alpha 1 here I have taken alpha 1 equal to 1 and solve it by steepest descent method; in the same way you will put alpha 1 equal to 10, alpha 2 equal to 10, alpha 3 equal to 100 and so on till you get the required tolerance deliver.

(Refer Slide Time: 27:12)



Iterations Table

k	α_k	x_k	$f(x_k)$	$q(x_k, \alpha_k)$	$P(x_k)$	$\alpha_k P(x_k)$
1	1	(2.375, 2.75)	−46.3906	−46.375	0.0156	0.0156
2	10	(2.343, 2.686)	−46.3512	−46.3429	0.00083	0.0083
3	100	(2.334, 2.669)	−46.3353	−46.3344	0.000009	0.0009

$q(x_3, \alpha_3) - f(x_3) = 0.0009 < 0.001$ so as per the stopping criteria, we stop at $k = 3$ and optimal solution of the NLP is $(x_1, x_2) = (2.334, 2.669)$ and objective value is -46.3353.

swayam
NPTEL
14

So, here when we put alpha 1 alpha k equal to 1 in this problem. Then applying any unconstrained algorithm unconstrained minimization algorithm alpha 1 comes out to be this; this is basically this is basically optimal solution of this problem optimal solution of this problem.

And for this f x k is comes out to be this q x k alpha k means it is g basically g comes out to be this, P x k is this and alpha k P x k is this, because alpha k is 1 here if it is 1 then this is simply seen.

Now, take alpha k equal to 10, alpha 2 equal to 10, if alpha 2 equal to 10 apply the same algorithm, find out the optimal solution same iteration we will get this row. And for alpha k equal to for k equal to 3 alpha will be 100 the optimal solutions comes out to be this.

Now here this is this is 0.0009 which is less than 0.001 as given this problem as a stopping criteria; so we will stop here. So, this is basically a method for solving for solving constrained optimization model.

So, basically in any constrained minimization algorithm the main aim is to convert that problem into an unconstrained minimization problem. And then we will solve that unconstrained of minimization problem by using any such technique. So, these are main motive behind this.

So, this is one illustration or one method similarly we have. So, many other methods also for solving constrained minimization problems. So, we have seen that if you are having unconstrained minimization problem we have various method; the Steepest Descent method, Newton’s method, Conjugate Gradient method, other method also and for solving constrained optimization problem also we have various methods in the literature one of one of them is Penalty Function method.

So, using these techniques using these numerical such techniques we can develop our algorithm; if we are having an optimization models in machine learning algorithms. And that we can solve either by unconstrained optimization methods or by constrained optimization models methods so.

Thank you.