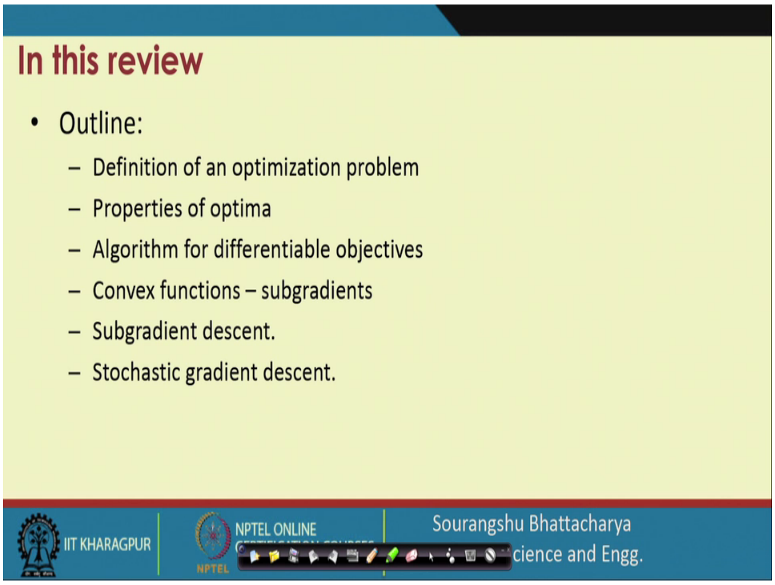
Scalable Data Science
Prof. Sourangshu Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
Background of Optimization

Hello students, welcome to the NPTEL course on Scalable Data Science Lecture number 4. Today we are going to discuss about the Background on Optimization. I am Professor Sourangshu Bhattacharya of Computer Science and Engineering Department IIT, Kharagpur.

(Refer Slide Time: 00:35)



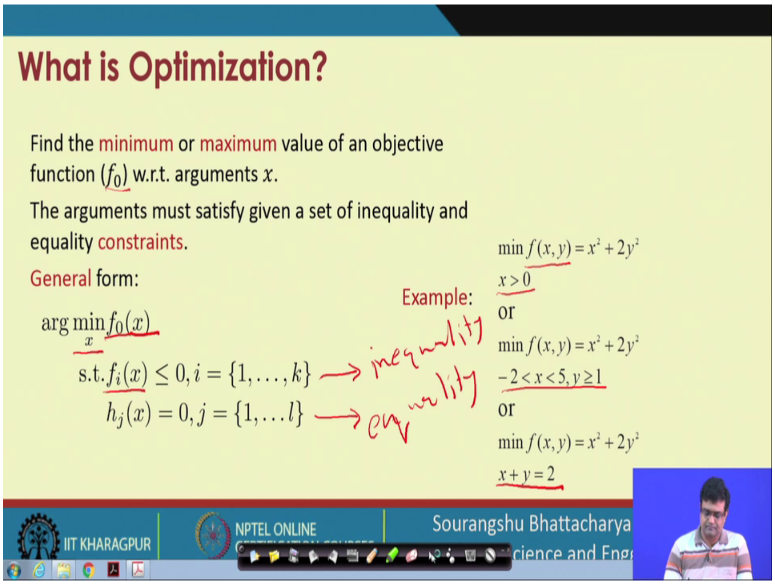
In this review

- Outline:
  - Definition of an optimization problem
  - Properties of optima
  - Algorithm for differentiable objectives
  - Convex functions – subgradients
  - Subgradient descent.
  - Stochastic gradient descent.

IIT KHARAGPUR
NPTEL ONLINE
NPTEL
Sourangshu Bhattacharya
cience and Engg.

So, the outline of today’s lecture is going to be, we are going to discuss definition of an optimization problem, the properties of optima, some algorithms for differentiable objectives. Then we are going to discuss convex functions and the concept of sub gradients, then we are going to briefly describe the subgradient descent algorithm. And finally, we will discuss the stochastic subgradient descent algorithm which is highly useful in the context of machine learning.

(Refer Slide Time: 01:14)



What is Optimization?

Find the minimum or maximum value of an objective function ($\underline{f_0}$) w.r.t. arguments x.

The arguments must satisfy given a set of inequality and equality constraints.

General form:

$$\arg\underline{\min_x}\,\underline{f_0(x)}$$

$$\text{s.t.}\,\underline{f_i(x)} \leq 0, i = \{1,\ldots,k\} \longrightarrow \text{inequality}$$

$$h_j(x) = 0, j = \{1,\ldots l\} \longrightarrow \text{equality}$$

Example:

$\min \underline{f(x,y)} = x^2 + 2y^2$

$\underline{x > 0}$

or

$\min f(x,y) = x^2 + 2y^2$

$\underline{-2 < x < 5, y \geq 1}$

or

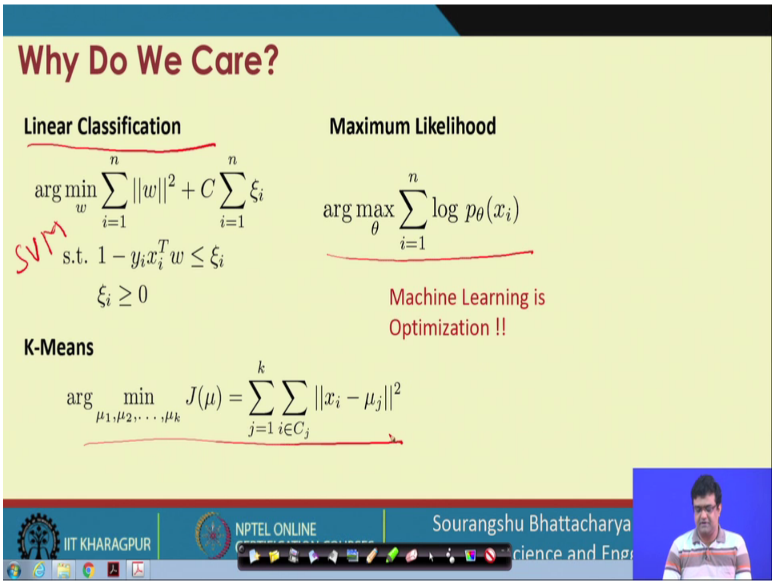
$\min f(x,y) = x^2 + 2y^2$

$\underline{x + y = 2}$

So, what is optimization? So, the optimization is a field that deals with finding the minimum or the maximum value of an objective function, let us say f 0 with respect to the arguments x. So, this is the objective function f 0 of x and we want to either maximize or minimize. In this particular case, we are trying to minimize the objective function with respect to the arguments x. Now, the other condition is that the arguments must satisfy some constraints. So, there can be two types of constraints.

So, this constraints are called inequality constraints, where the constraints are of form fi x is less than or equal to 0 for a certain number of functions, say i is equal to 1 to k and this constraints are called equality constraints. These are inequality constraints and these are equality constraints. This is also called the general form of optimization problem. So, some examples are, for example here f could be function of x and y two variables. And, the function could be something like x square plus 2 y square and the constraints could be something like x is greater than or equal to 0 or for example, the constrain set could be something like x is between minus 2 and 5 and y is greater than 1 or the constraint could something like x plus y should always be 2.

(Refer Slide Time: 03:23)



Why Do We Care?

Linear Classification

$$\arg\min_{w} \sum_{i=1}^{n} ||w||^2 + C\sum_{i=1}^{n} \xi_i$$

SVM

s.t. $1 - y_i x_i^T w \leq \xi_i$

$\xi_i \geq 0$

Maximum Likelihood

$$\arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

Machine Learning is Optimization !!

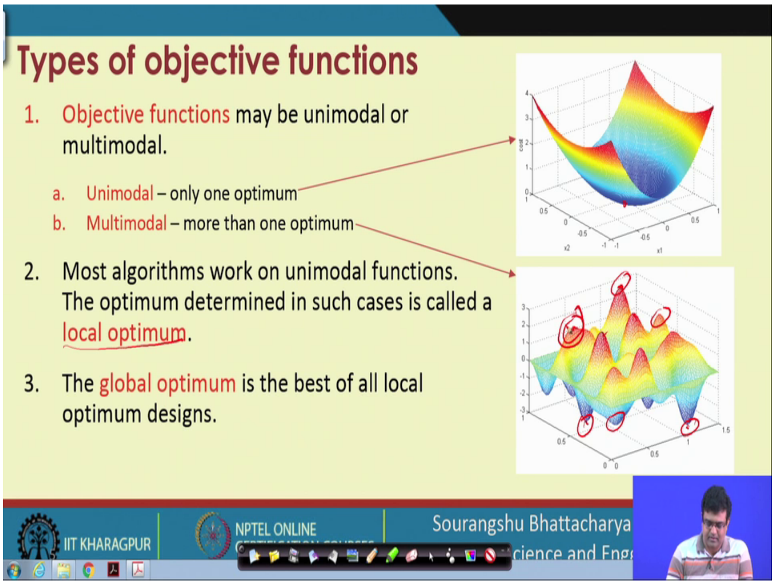
K-Means

$$\arg \min_{\mu_1,\mu_2,\ldots,\mu_k} J(\mu) = \sum_{j=1}^{k}\sum_{i\in C_j} ||x_i - \mu_j||^2$$

So, the question comes in this course why do we need to discuss optimization and the answer is that most of the machine learning problems or many of the machine learning problems involves some form of optimization or the others. So, for example, if we take the problem of linear classification, then it can be solved using an optimization problem like this. So, this formulation of linear classification is sometimes called Support Vector Machines.

Another set of problems which are very common in machine learning is to solve maximum like to estimate parameters of a probabilistic model using the maximum likelihood paradigm and again that happens to be an optimization problem of this form. Again there are problems in unsupervised learning like K-means clustering which also involves solving the optimization problem of this particular form. So, this appears multiple times in many machine learning context, ok.

(Refer Slide Time: 04:40)



Types of objective functions

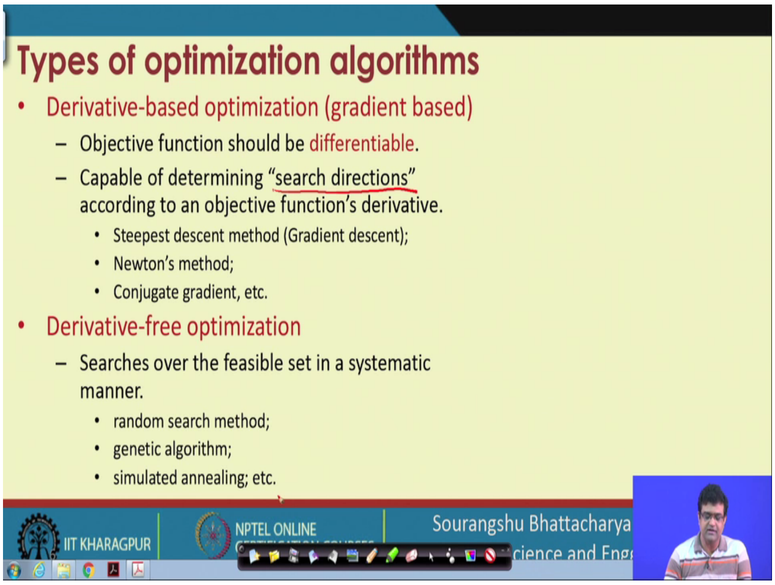
1. Objective functions may be unimodal or multimodal.
   a. Unimodal – only one optimum
   b. Multimodal – more than one optimum
2. Most algorithms work on unimodal functions. The optimum determined in such cases is called a local optimum.
3. The global optimum is the best of all local optimum designs.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Sourangshu Bhattacharya cience and Engi

So, as we have already discussed any optimization problem has some objective functions. So, in the previously described optimization problem, the objective function was the f 0 of x, ok. Now, the first thing that decides how we can solve an optimization problem is the type of objective function. So, there are roughly two types of objective functions we can think of. First one is a unimodal objective function where there is only one optima. So, as you can see here there is only one minima, whereas here there are multiple maxima and multiple minima. So, these are some of the minima and these are some of the maxima, ok. So, this kind of an objective function is called Multimodal Objective Function.

Now, most optimization algorithm work on the assumption that the objective function is a unimodel objective function and in such cases, the optimization algorithms tend to find what is called a local optima. So, something like let say this is this point would still be called a local maxima; even though clearly this point has a higher value of objective function than this point. The global optima is the best or the highest in case of maximization problem and lowest in case of minimization problem. So, the global optima is the best of all local optima.

(Refer Slide Time: 06:42)



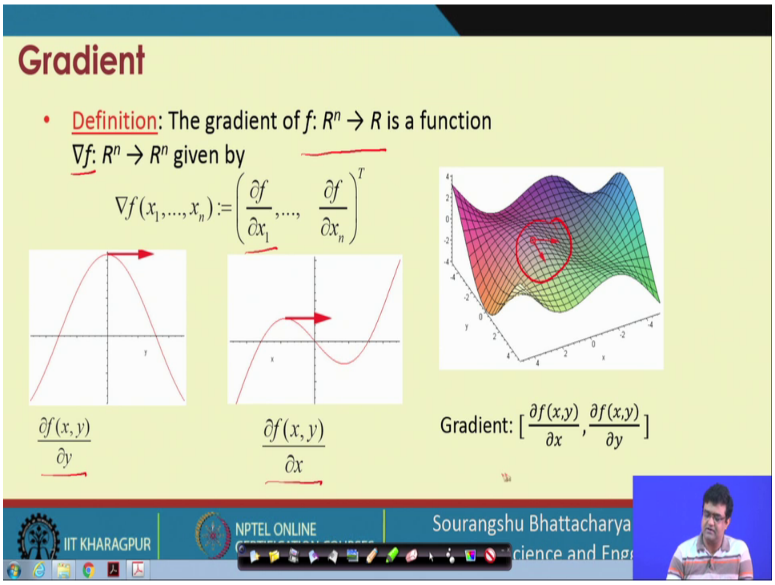
Types of optimization algorithms

- Derivative-based optimization (gradient based)
  - Objective function should be differentiable.
  - Capable of determining "search directions" according to an objective function's derivative.
    - Steepest descent method (Gradient descent);
    - Newton's method;
    - Conjugate gradient, etc.
- Derivative-free optimization
  - Searches over the feasible set in a systematic manner.
    - random search method;
    - genetic algorithm;
    - simulated annealing; etc.

Now, roughly there are all the optimization algorithms can be divided into two classes. The first class is called the derivative based optimization algorithms or also called the gradient based optimization algorithm. So, in order to use this kind of an optimization algorithm, the objective function should be differentiable and in that case the algorithm is able to determine good search directions, ok; so which directions to move according to the objective functions derivative. So, examples of these kinds of methods are Steepest Descent or Gradient Descent Newton's method, Conjugate Gradient, etcetera

These methods are generally much faster and are much more widely used than the second type of objective function. The second type of objective functions or optimization algorithms, sorry do not make such assumptions that the objective function should be differentiable. However, they generate the possible solutions in a systematic manner from the information that they have gathered till now and they search over the entire optimization space, ok. So, some of the methods include Random search, Genetic algorithms, Simulated Annealing etcetera and as you can see many many of these algorithms actually try to find the global optimum rather than the local optima. However, these are generally slower than the derivative based methods.

(Refer Slide Time: 08:35)



Gradient

- Definition: The gradient of $f: R^n \to R$ is a function $\nabla f: R^n \to R^n$ given by

$$\nabla f(x_1,\ldots,x_n) := \left(\frac{\partial f}{\partial x_1},\ldots,\frac{\partial f}{\partial x_n}\right)^T$$

$\frac{\partial f(x,y)}{\partial y}$

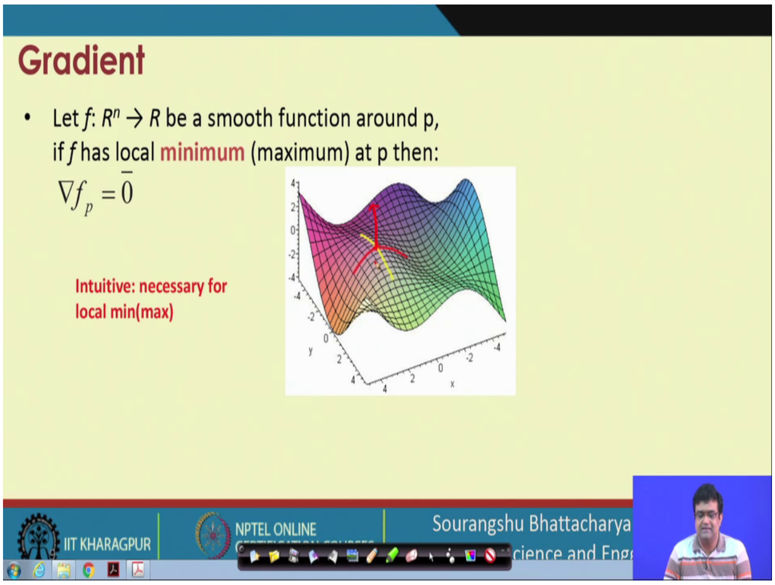
$\frac{\partial f(x,y)}{\partial x}$

Gradient: $[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}]$

So, in this lecture we will mostly concentrate in gradient based algorithms as those are the most practical and more highly used. So, in order to discuss the gradient based algorithms, first we discuss what the gradient is. So, the gradient of any functions f from R n or n dimensional Euclidean space to R is some of is denoted by this delta of or inverse delta of f and it is defined like this.

So, it is a vector of n dimension, where the first dimension is the partial derivative of the function x with or function f with respect to x and so on and so forth, ok. Do this as an example. So, this is the partial derivative with respect to y direction and this could be the partial derivative with respect to x direction and when you put these two together, you get the gradient of the objective function, ok.

(Refer Slide Time: 09:44)



Gradient

- Let f: $R^n \rightarrow R$ be a smooth function around p, if f has local minimum (maximum) at p then:

$$\nabla f_p = \bar{0}$$

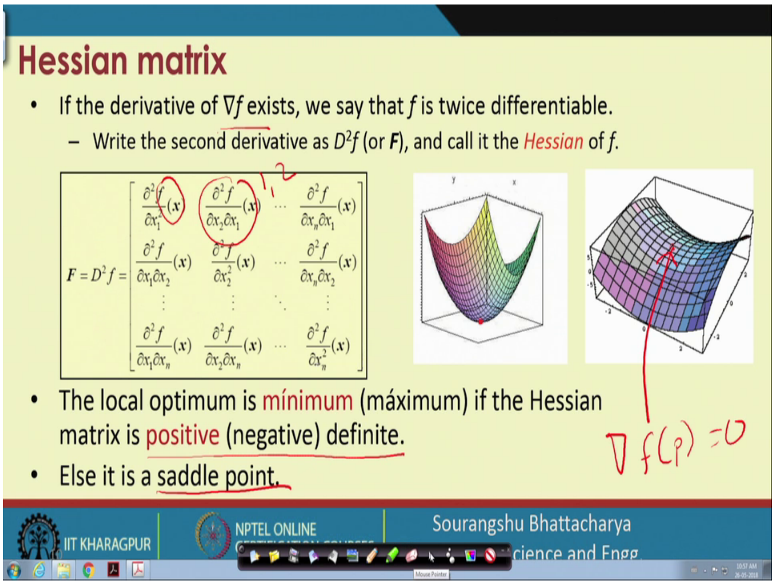
Intuitive: necessary for local min(max)

IIT KHARAGPUR | NPTEL ONLINE | Sourangshu Bhattacharya
cience and Engi

Now, the first result which is one of the most important results of optimization is that for any function f which is a smooth function, basically which sort of means that it is a differentiable function, continuous and differentiable function; f has a local minimum or a maximum at a point p. So, let say this is the point p, f has a local optimum around this point of the gradient of p is equal to 0.

So, if the gradient of p around this point is exactly equal to 0, then this is a local optima, and this is true because what we can see is that gradient of p actually gives the direction of the normal at the point p and if this normal is and also, the gradient of p gives the direction of maximal descent as we shall see the direction of maximal ascent or maximal descent at a certain point p. Hence, if gradient of p is 0, it implies that there is no direction of maximal ascent.

(Refer Slide Time: 11:56)



Hessian matrix

- If the derivative of ∇f exists, we say that f is twice differentiable.
  - Write the second derivative as $D^2 f$ (or $\mathbf{F}$), and call it the *Hessian* of f.

$$F = D^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{bmatrix}$$

1, 2

- The local optimum is mínimum (máximum) if the Hessian matrix is positive (negative) definite.
- Else it is a saddle point.

$\nabla f(p) = 0$

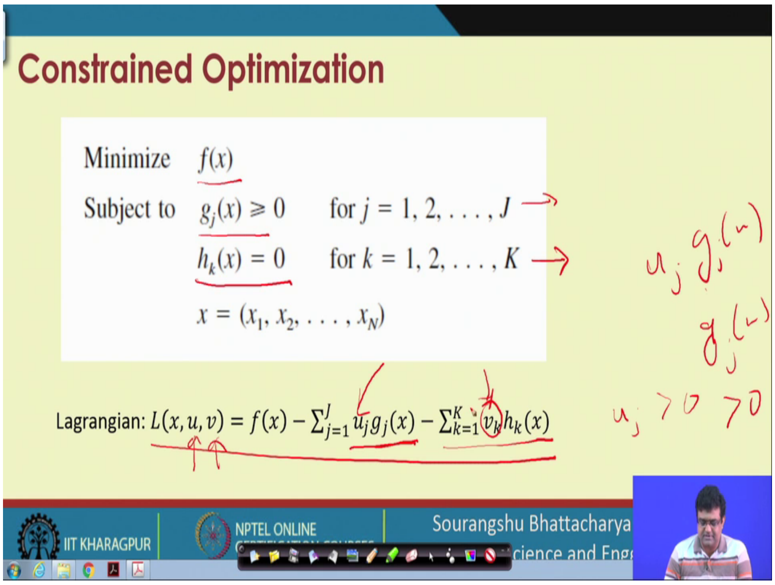
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Sourangshu Bhattacharya
cience and Engg.

Another important quantity in case of differentiable functions is the Hessian matrix. For Hessian, for defining Hessian matrix, firstly the gradient of f has to be a differentiable function. If that is the case, then we can write the matrix of second derivatives of the gradient of x like this. So, the i jth entry of this matrix is derivative of the double derivative, the double partial derivative of f with respect to xi and xj. So, for example, this is the entry 1 2.

So, first row and second column and as you can see it is the derivative of f with respect to x 1 and then, with respect to x 2, and one can show that it does not really matter what order you differentiate and hence, this Hessian matrix is always a symmetric matrix. Now, the important result here is that if the Hessian matrix is positive, semi definite or positive definite rather, then at a certain point when the evaluated at a certain point, then if that point is local optima, then that point will be a local minimum and if it is a negative definite, if Hessian matrix is a negative definite, then it will be local maximum.

Now, there are points called saddle points as shown here, where the derivative or the gradient is 0. So, even at this point the gradient of f at p is 0. However, as you can see it is neither a local optimum nor a local minimum. So, this will be reflected in the Hessian matrix being indefinite.

(Refer Slide Time: 14:37)



Constrained Optimization

Minimize $f(x)$

Subject to $g_j(x) \geq 0$ for $j = 1, 2, \ldots, J$

$h_k(x) = 0$ for $k = 1, 2, \ldots, K$

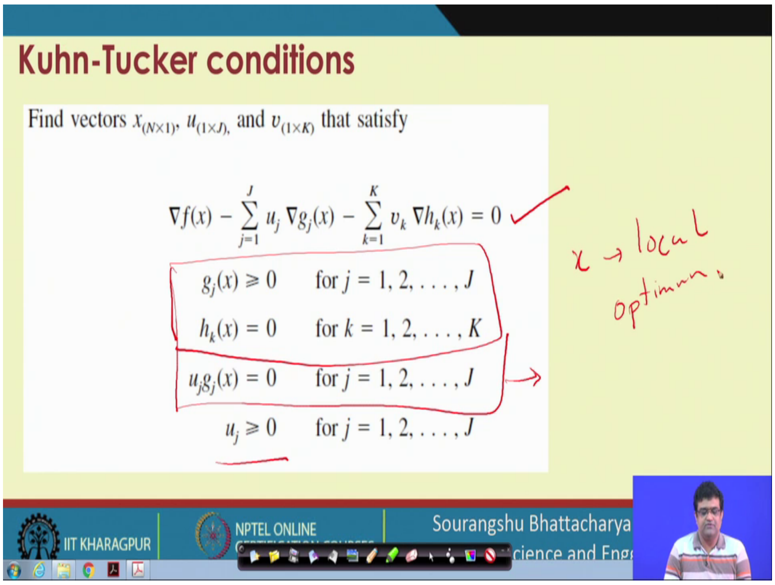
$x = (x_1, x_2, \ldots, x_N)$

Lagrangian: $L(x,u,v) = f(x) - \sum_{j=1}^{J} u_j g_j(x) - \sum_{k=1}^{K} v_k h_k(x)$

Another important issue is how to incorporate constraints into the objective function. So, for example, if we have a general optimization problem of this form minimize f of x with respect to, such that g j of x is greater than or equal to 0 and h k of x is equal to 0. So, these are the inequality constraints and these are the equality constraints. Then, we would like to find the global minimum and one way to find the global minimum is to use the Lagrangian function which says that, so it uses some extra variables. So, in addition to x, the Lagrangian function is also a function of theses vectors u and v which are also called Lagrang multipliers and these extra variables penalize the violation of these constraints.

So, for example, in this case if g j of x is not greater than 0, that is less than 0 and in this case u j is greater than 0, then the objective, the total Lagrangian function will actually increase by u j times g j of x. So, it will increase by u j times g j of x and this could be reduced further by satisfier, by satisfying the constraints that g j of x is actually greater than 0. Similarly, in this case if this constraint is not satisfied, then one can choose v k such that the objective function can be reduced further. So, this is the idea behind. So, this idea is formalized in what is called KKT conditions or Kuhn Tucker conditions and the conditions are given here.

(Refer Slide Time: 17:01)



Kuhn-Tucker conditions

Find vectors $x_{(N\times 1)}$, $u_{(1\times J)}$, and $v_{(1\times K)}$ that satisfy

$$\nabla f(x) - \sum_{j=1}^{J} u_j \nabla g_j(x) - \sum_{k=1}^{K} v_k \nabla h_k(x) = 0 \checkmark$$

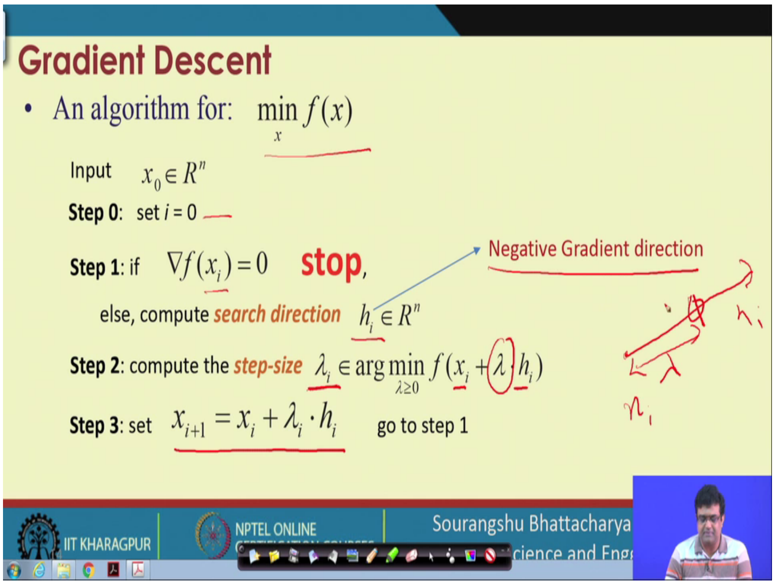
$$\begin{aligned} g_j(x) &\geq 0 && \text{for } j = 1, 2, \dots, J \\ h_k(x) &= 0 && \text{for } k = 1, 2, \dots, K \\ u_j g_j(x) &= 0 && \text{for } j = 1, 2, \dots, J \\ u_j &\geq 0 && \text{for } j = 1, 2, \dots, J \end{aligned}$$

$x \rightarrow$ local optimum.

So, basically this condition is called the First Order condition, which is saying that the derivative of the Lagrangian with respect to x should be 0. And this conditions are satisfied or these two conditions are derived when we take derivative with respect to u j to be greater than or derivative with respect to u j and set them to the appropriate value.

So, the in case of h k, it has to be equal to 0 and in case of g j, it has to be greater than or equal to 0 and the final condition is that what is also called the complimentary slackness condition. This says that basically u j times g j of x should be equal to 0 and this is the condition on the Lagrange multiplier. So, if these conditions are satisfied, the point x is a local optimum, ok. So, this is one of the ways of incorporating constraints while solving an optimization problem.

(Refer Slide Time: 18:39)



Gradient Descent

- An algorithm for: $\min_{x} f(x)$

Input $x_0 \in R^n$

Step 0: set $i = 0$

Step 1: if $\nabla f(x_i) = 0$ **stop**,

else, compute *search direction* $h_i \in R^n$

Negative Gradient direction

Step 2: compute the *step-size* $\lambda_i \in \arg\min_{\lambda \geq 0} f(x_i + \lambda h_i)$

Step 3: set $x_{i+1} = x_i + \lambda_i \cdot h_i$ go to step 1

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

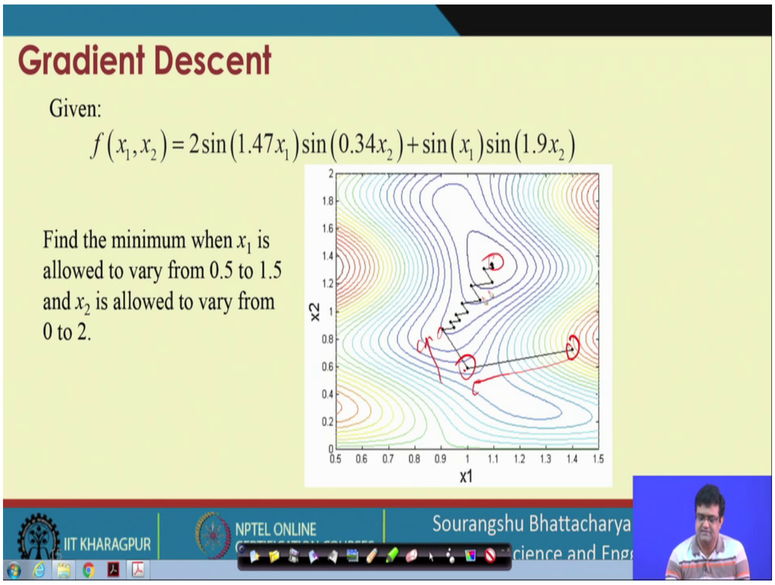
Sourangshu Bhattacharya

cience and Engi

Now, we go into algorithms for solving this optimization problem. So, the first algorithm we discuss is called the Gradient Descent algorithm. So, suppose we have an algorithm function like minimize f of x, the algorithm takes an input x 0 which is an initial point. Now, the algorithm work iteratively or it works in steps and then, the steps are executed many times. So, the first step you set i is equal to 0. Now, if xi which is x 0 in this particular case at this point if the derivative of f of x is 0, then we stop because we have already reached local minima, otherwise you compute a search direction. So, a good search direction is in case of minimization problem is the negative gradient direction, since this is also the direction of steepest descent ok.

So, once we have computed the search direction h i, the next step of the algorithm is to compute a step size lambda i, such that if you move towards the direction h i from x i by an amount lambda i. So, if you are x i, you move in the direction h i by an amount lambda i, you get the new point which is x i plus lambda i times h i and you compute the objective function value f at that point and you minimize this over all lambda and whichever lambda minimizes, this is the new lambda i. Then, you set x i plus 1 is equal to x i plus lambda i h i. So, you basically move to this new point and you set i is equal to i plus 1. So, you proceed in this manner to the next step.

(Refer Slide Time: 21:15)



Gradient Descent

Given:

$$f(x_1,x_2)=2\sin(1.47x_1)\sin(0.34x_2)+\sin(x_1)\sin(1.9x_2)$$

Find the minimum when x_1 is allowed to vary from 0.5 to 1.5 and x_2 is allowed to vary from 0 to 2.

IIT KHARAGPUR

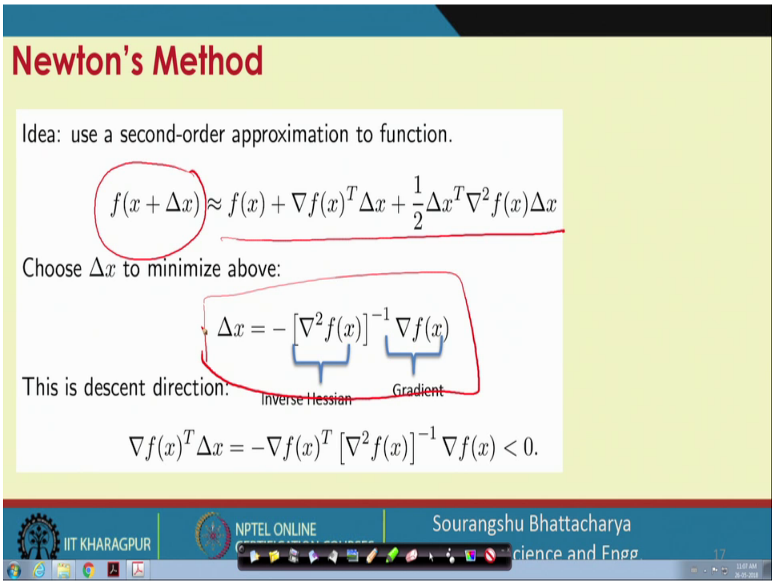
NPTEL ONLINE

Sourangshu Bhattacharya

cience and Engg

So, this shows trace of the optimization algorithm being run. So, what you see here are the contours of the objective function. So, this is the starting point. The algorithm starts from this point and moves in this direction, reaches the next point, then computes the gradient direction, moves in this direction, reaches the next point and keep doing this until it reaches the optima which is this point ok. So, these are the steps of running Gradient Descent Algorithm.

(Refer Slide Time: 22:00)



Newton's Method

Idea: use a second-order approximation to function.

$$f(x+\Delta x)\approx f(x)+\nabla f(x)^T\Delta x+\frac{1}{2}\Delta x^T\nabla^2 f(x)\Delta x$$

Choose Δx to minimize above:

$$\Delta x=-\underbrace{\left[\nabla^2 f(x)\right]^{-1}}_{\text{Inverse Hessian}}\underbrace{\nabla f(x)}_{\text{Gradient}}$$

This is descent direction:

$$\nabla f(x)^T\Delta x=-\nabla f(x)^T\left[\nabla^2 f(x)\right]^{-1}\nabla f(x)<0.$$

IIT KHARAGPUR

NPTEL ONLINE

Sourangshu Bhattacharya

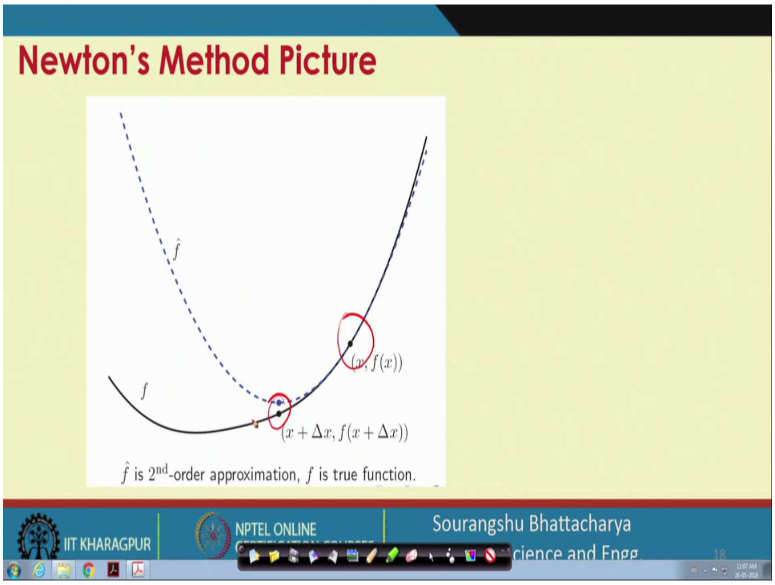
cience and Engg

Now, gradient descent algorithm has a problem. So, what is the problem with gradient descent algorithm? So, the main problem with the gradient descent algorithm is that at a given point even though the gradient direction is giving the direction of the steepest descent or the optimal descent. For example in this point it is pointing to this direction which is the direction of the optimal descent.

However, the actual optima is in this direction. So, we find the optimal point in this step here and then, at this point again the direction of descent is in this direction, pointing in this direction and so on and so forth, ok. So, this is the next point. So, we see that there is a lot of zigzag in behavior which leads to a very slow convergence of the gradient descent algorithm at some point in time, ok. So, the basic problem is that the gradient descent algorithm does not utilize the second order information or the information of Hessian, ok.

So, if we have to, if the function is twice differentiable, then we can use what is called the Newton's method to utilize the second order information and optimize. So, the optimization steps proceed as follows. So, this is an approximation of the function when you move from x to x plus delta x and we wish to minimize this quantity and this quantity is minimized. If we just differentiate this with respect to delta x, then we see that the optimal value of delta x turns out to be this. So, this is the as you can see that this is also a descent direction, ok.

(Refer Slide Time: 24:31)



Newton's Method Picture

$\hat{f}$ is 2^{nd}-order approximation, f is true function.

IIT KHARAGPUR

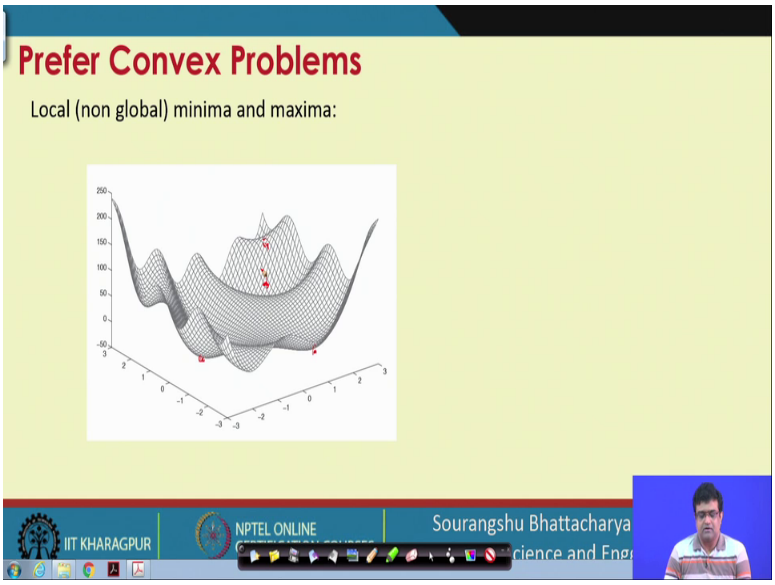
NPTEL ONLINE

Sourangshu Bhattacharya

cience and Engg

So, this is the iteration that is followed by the by Newton's method, that is at any point it takes step of delta x as computed by the formula shown in the previous method and it computes the next x and if we keep doing this, we reach the optimum much faster.

(Refer Slide Time: 24:58)

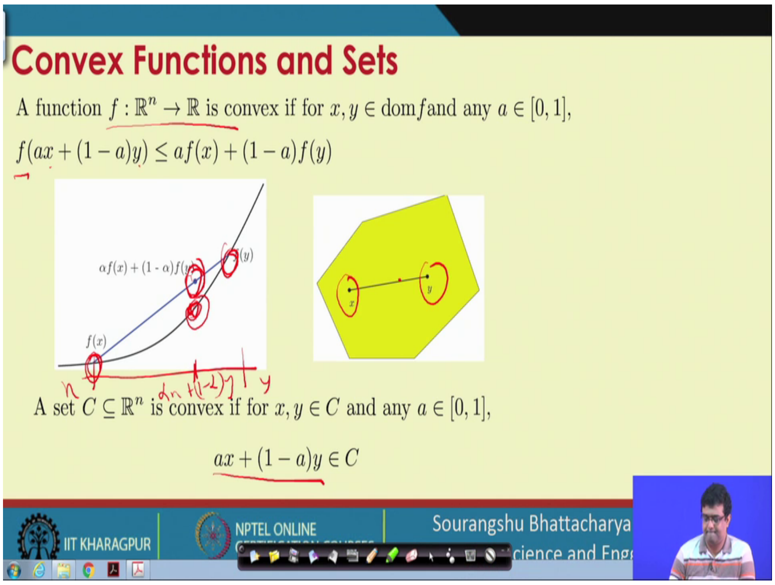


Prefer Convex Problems

Local (non global) minima and maxima:

Now, one of the main problems of a generic optimization problem as we discussed is that we can reach a local optimum, but the local optimum are not always the global optimum, ok. For example, in this function you can see that there are many local optima here ok, but the global optima probably somewhere here ok. A global minima is somewhere here where as this is another minima and there are probably many other minima here, ok. So, how can we tackle this problem?

(Refer Slide Time: 25:46)



Convex Functions and Sets

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if for $x, y \in \mathrm{dom} f$and any $a \in [0, 1]$,

$$f(ax + (1-a)y) \leq af(x) + (1-a)f(y)$$

A set $C \subseteq \mathbb{R}^n$ is convex if for $x, y \in C$ and any $a \in [0, 1]$,

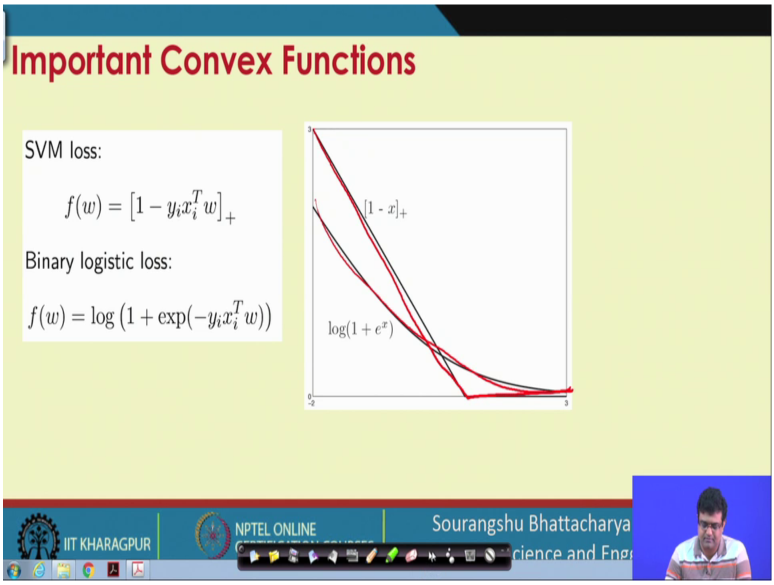
$$ax + (1-a)y \in C$$

So, we can tackle this problem if we restrict our problems to what are called Convex Optimization Problem. So, what are convex optimization problem? First we define the convex functions and the convex sets. So, a convex function is a function of this form. So, a convex function is a function of this form, ok. So, the formal mathematical definition is that a function f of x is convex if for any two points x and y. So, you take any two point x and y and you can compute a point on the line joining x and y using this formula alpha times x plus 1 minus alpha times y.

So, this is the point in the line joining x and y, where alpha is between 0 and 1 and you can compute the function value at this point and you can also compute the interpolation. So, this is the function value at this point, and you can also compute the linear interpolation of f of x and f of y which is this particular value. So, if we see the points x and y here, so this is alpha times x plus 1 minus alpha times y and this is the function value at that point and this is the interpolated value with respect to f fx and fy. So, the function is convex only if the function value is lower than the interpolated value. In other words, you have this kind of a shape where this function value as you can see is lower than the interpolated value.

Another very important notion is that of convex sets. Again if we have two points x and y and take interpolated point on the line joining x and y, then the interpolated point should also belong to a set, ok. If this property is satisfied for all x and y, then the set is called a convex set. So, a convex sets looks like this, ok.

(Refer Slide Time: 28:14)



Important Convex Functions

SVM loss:

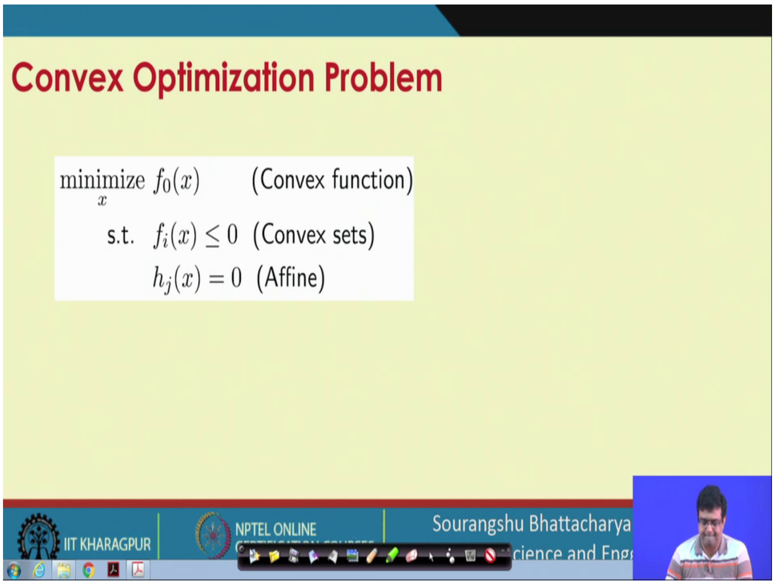
$$f(w) = \left[1 - y_i x_i^T w\right]_+$$

Binary logistic loss:

$$f(w) = \log\left(1 + \exp(-y_i x_i^T w)\right)$$

Now, why do we care about convex functions? So, as it turns out many of the functions that we optimize in machine learning's, so for example, in support vector machine or binary logistic regression are all convex functions, ok. So, they are very important class of functions in machine learning. So, as you can see this is the lost function for support vector machine and this is the last function for binary logistic regression.

(Refer Slide Time: 28:49)

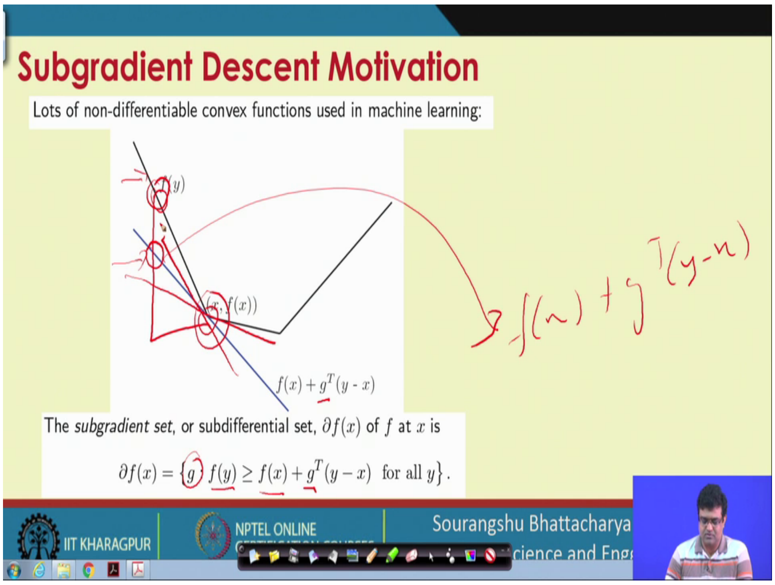


Convex Optimization Problem

$$\begin{aligned} &\underset{x}{\text{minimize}}\ f_0(x) && \text{(Convex function)} \\ &\text{s.t.}\quad f_i(x) \leq 0 && \text{(Convex sets)} \\ &\qquad\ \ h_j(x) = 0 && \text{(Affine)} \end{aligned}$$

So, what is Convex Optimization Problem? A Convex Optimization problem is a problem of this form where your objective function is a convex function and the sets or the feasible sets defined by the constraints are all convex sets.

(Refer Slide Time: 29:14)



Subgradient Descent Motivation

Lots of non-differentiable convex functions used in machine learning:

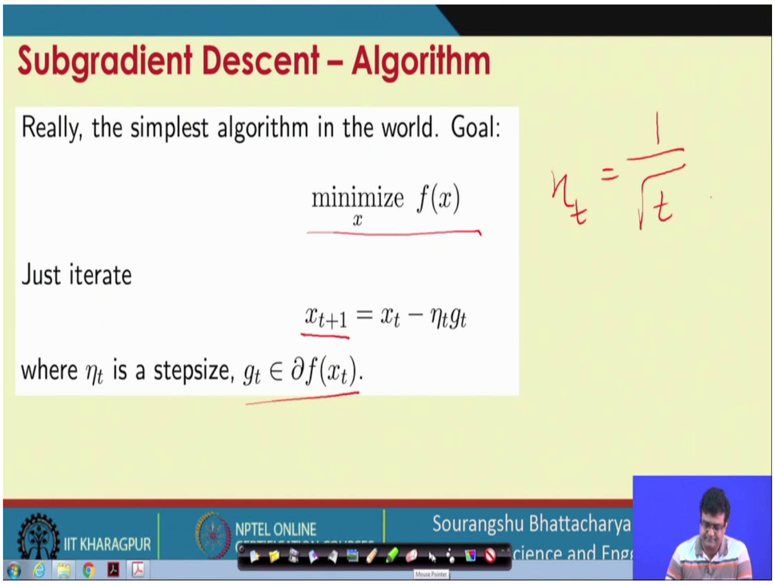
The *subgradient set*, or subdifferential set, $\partial f(x)$ of f at x is

$$\partial f(x) = \{g : f(y) \geq f(x) + g^T(y - x) \text{ for all } y\}.$$

Now, if function is convex even if it is not differentiable, we can still define something similar to a gradient, so called sub-gradient, ok. So, for example, if you look at this function, it is not differentiable, but we can define a sub gradient for this function at this point by. So, we can define a sub gradient of this function as the set of all g. So, all vectors g such that f of y. So, this is f of y, is greater than f of x plus g transpose y minus x. So, f of x plus g transpose y minus x is the interpolation, is the linear interpolation of this function f of x rather linear extrapolation or approximation of this function from this point, ok.

So, this is the value and this is the value of f of y. So, for all directions g such that f of y has to be greater than this linear extrapolation from x based on this g. Such gradients are called sub-gradients. These are basically directions which lie below the convex function. So, these are all the directions that come as sub-gradients. So, now why are these sub gradients important?

(Refer Slide Time: 31:23)



Subgradient Descent – Algorithm

Really, the simplest algorithm in the world. Goal:

$$\underset{x}{\text{minimize}} \; f(x)$$

Just iterate

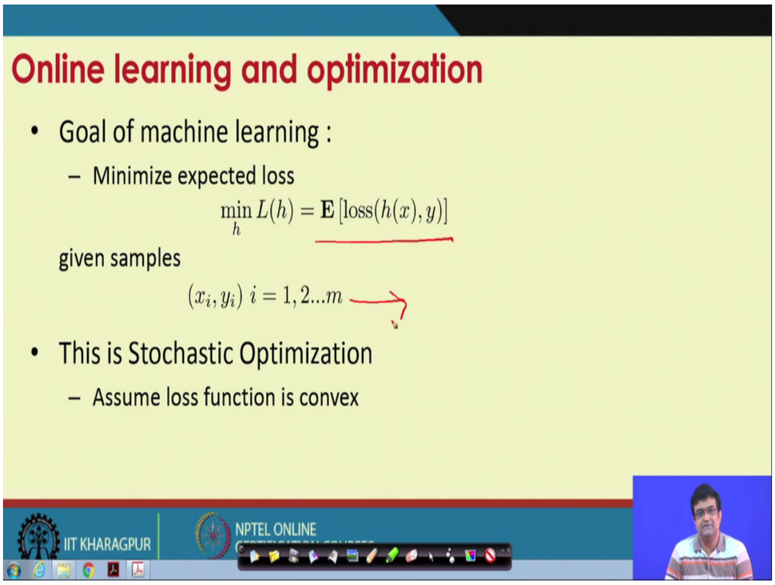
$$x_{t+1} = x_t - \eta_t g_t$$

where η_t is a stepsize, $g_t \in \partial f(x_t)$.

$\eta_t = \frac{1}{\sqrt{t}}$

The sub gradients are important because we can actually just replace, so we can have a very simple algorithm called Sub-gradient Descent Algorithm. So, suppose our goal is to minimize f of x with respect to x, where x is a convex function, but not necessarily differentiable, but if you use this algorithm x t plus 1 is equal to x t plus eta t, where eta t is a step size and step size is usually a decreasing step size. So, eta t can for example be taken as 1 by square root of t, ok. So, this is one way to decrease the step size and if this is the case, then we can still find the optimum of the convex function even though it is not differentiable.

(Refer Slide Time: 32:39)



Online learning and optimization

- Goal of machine learning :
  - Minimize expected loss

$$\min_h L(h) = \mathbf{E}\left[\text{loss}(h(x), y)\right]$$

given samples

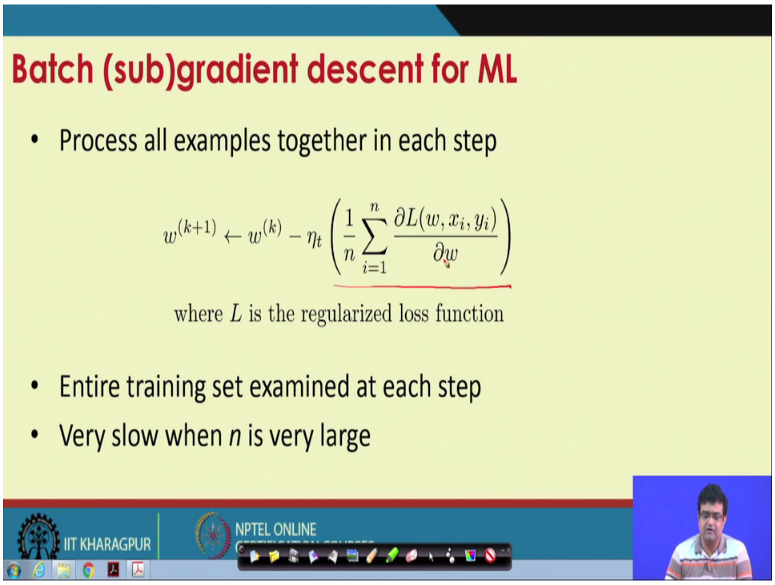
$$(x_i, y_i) \; i = 1, 2...m$$

- This is Stochastic Optimization
  - Assume loss function is convex

Now, finally, we see that we have discussed all these algorithms which are which can optimize various kinds of objective functions. So, one kind of objective function which we encounter a loss in machine learning is the loss minimization function. So, the functions is of this form that you have a loss function which is dependent on the sample data points. Usually we have a lot of sample data points and the loss function is the expectation or the average over the loss at each of this individual sample data points, ok.

So, if the objective function is of this form, then we can use a much faster algorithm than gradient descent or sub gradient descent. So, this faster algorithm is an online algorithm which instead of using all the m samples, to note that the objective function depends all the m data points or all the m samples, however we need not use at each. So, if we use all the m samples to compute the gradient or the objective function at each step, then it will be very time consuming. So, instead of using all samples, we can use a few or even one sample at each iteration and still we will be able to reach the optimum of the total loss function. So, this algorithm is called Batch Gradient Descent.

(Refer Slide Time: 34:32)



Batch (sub)gradient descent for ML

- Process all examples together in each step

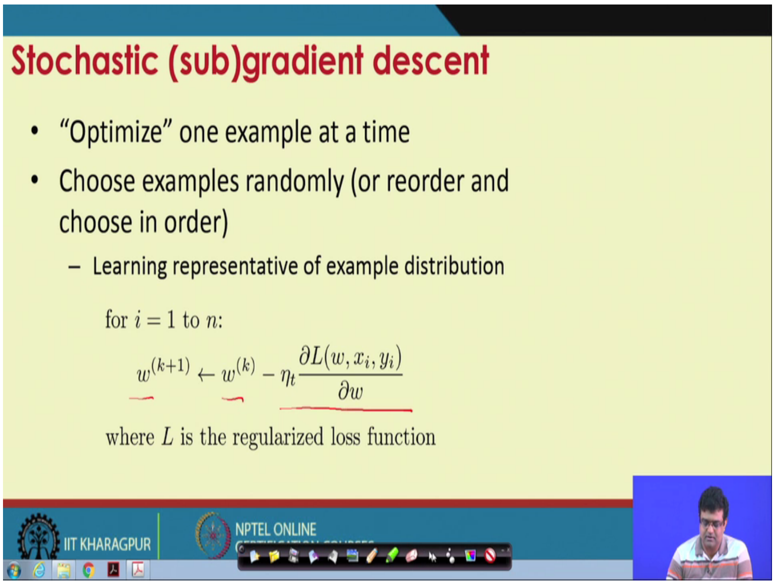
$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \underline{\left(\frac{1}{n} \sum_{i=1}^{n} \frac{\partial L(w, x_i, y_i)}{\partial w} \right)}$$

where L is the regularized loss function

- Entire training set examined at each step
- Very slow when *n* is very large

So, this is batch gradient descent or sub gradient descent algorithm where we are computing the gradient. So, this is the gradient, total gradient for all or average of total gradient for all the data points. Then, we are doing the sub gradient descent, but as we have discussed this is very slow when n is large.

(Refer Slide Time: 35:13)



Stochastic (sub)gradient descent

- "Optimize" one example at a time
- Choose examples randomly (or reorder and choose in order)
  - Learning representative of example distribution

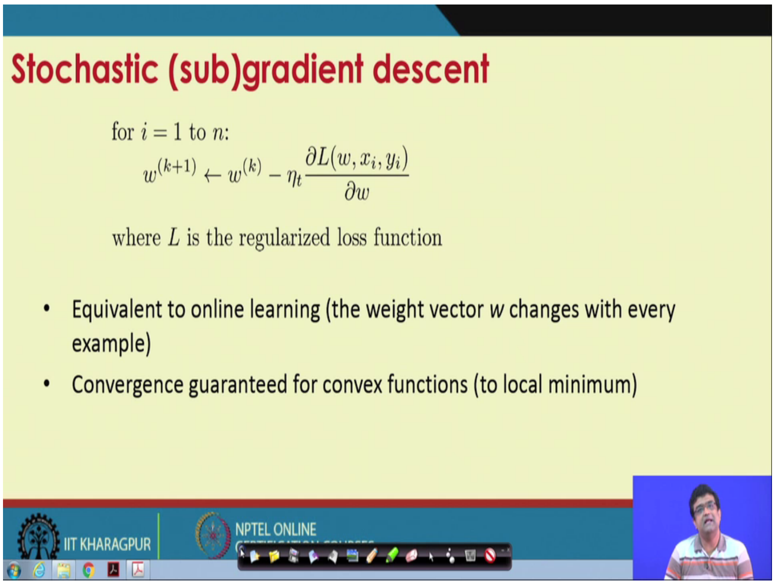
for $i = 1$ to n:

$$\underline{w^{(k+1)}} \leftarrow \underline{w^{(k)}} - \underline{\eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}}$$

where L is the regularized loss function

So, stochastic sub gradient descent is an algorithm which takes only one example at a time and does a similar kind of iterative updates. So, instead of adding the derivative of the total loss, what we have done here is, we have just computed the derivative of the loss calculated using just one data point. So, ith data point and then, we are updating the parameters which are w is in this case using the standard sub gradient descent algorithm or gradient descent algorithm that we have described earlier. It can be shown that if L is a convex loss, then this algorithm also converges to the global optima, ok.

(Refer Slide Time: 36:15)



Stochastic (sub)gradient descent

for $i = 1$ to n:

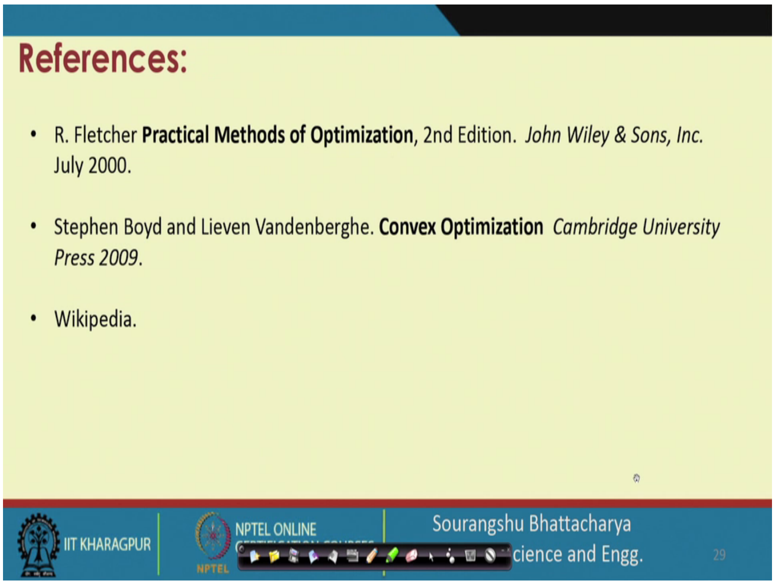
$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

where L is the regularized loss function

- Equivalent to online learning (the weight vector *w* changes with every example)
- Convergence guaranteed for convex functions (to local minimum)

This is equivalent to learning the weights in online fashion, in the sense that when you get one example at a time, you can still you can, so instead of getting the entire set of training data, if you get few training data points, you can still get the optimal model and you can learn the optimal learning function or the optimal prediction function.

(Refer Slide Time: 36:53)



References:

- R. Fletcher **Practical Methods of Optimization**, 2nd Edition. *John Wiley & Sons, Inc.* July 2000.
- Stephen Boyd and Lieven Vandenberghe. **Convex Optimization** *Cambridge University Press 2009.*
- Wikipedia.

So, that brings us to the end of this lecture. So, these are some of the references; Practical Methods of Optimization by Fletcher is a very nice book which describes the derivative based methods and Convex Optimization by Stephen Boyd and Lieven Vandenberghe is a book which describes the convex optimization and techniques.

Thank you.