Revolutionising B.Tech

Module 2: Convex functions and their properties

Course Name: Numerical Optimization [22CSE304]

Total Hours : 6

Table of Content

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Aim

To understand convex functions and their properties

Objective

a. To understand the functions in machine learning
b. To understand the stochastic gradient descent and its variants
c. To accelerate the gradient methods
d. To understand the proximal methods and their applications

Definition of convex functions

- A convex function is a mathematical function that satisfies the property that the line segment connecting any two points on the graph of the function lies above the graph itself. In other words, a function f: Rn→R is convex if, for any two points x1 and x2 in its domain and for any λ in the interval [0, 1], the following inequality holds:

$$f(\lambda x1+(1-\lambda)x2)\leq\lambda f(x1)+(1-\lambda)f(x2)$$

- Geometrically, this property implies that the function lies below or on the straight line segment connecting any two points on its graph. In other words, the graph of a convex function is "bowed" outward or flat, and it doesn't curve downward.

Definition of convex functions

- Convex functions play a crucial role in optimization and mathematical analysis due to their well-behaved properties.

- Many optimization problems involving convex functions have efficient algorithms and unique global minima, making them particularly amenable to analysis and solution.

- Common examples of convex functions include linear functions, quadratic functions, and exponential functions.

Convex sets and convex combinations

1. **Convex Set:**

- A set C in a vector space is called convex if, for any two points x and y in C, the line segment joining x and y lies entirely within C. Mathematically, C is convex if, for all $x, y \in C$ and for all λ in the interval $[0,1]$, the point $\lambda x+(1-\lambda)y$ is also in C.
- Geometrically, this means that every point on the line segment connecting any two points in the set is also in the set.

2. **Convex Combination:**

- Given a set S of vectors in a vector space, a convex combination of vectors in S is a linear combination of these vectors where the coefficients are non-negative and sum to 1.
- Mathematically, if $v1,v2,\ldots,vk$ are vectors in S and $\lambda1,\lambda2,\ldots,\lambda k$ are non-negative numbers such that $\sum i=1k\lambda i=1$, then the vector $\sum i=1k\lambda ivi$ is a convex combination of $v1,v2,\ldots,vk$.
- Geometrically, a convex combination represents a point in the convex hull of the set S, which is the smallest convex set containing all the points in S.

Jensen's inequality and subdifferentials

- Jensen's inequality is a fundamental result in convex analysis and probability theory. It provides conditions under which the convex transformation of the expected value of a random variable is less than or equal to the expected value of the convex transformation of the random variable.

- Let's denote X as a random variable and g as a convex function. Jensen's inequality can be stated as follows:

$$g(\mathrm{E}[X]) \leq \mathrm{E}[g(X)]$$

- Here, $\mathrm{E}[X]$ represents the expected value of X. The inequality holds if g is a convex function.

- Now, let's discuss subdifferentials in the context of convex analysis. The subdifferential of a convex function at a point is a set that characterizes all possible slopes (gradients) of linear functions that lie below the graph of the convex function at that point.

Jensen's inequality and subdifferentials

- The subdifferential of a convex function f at a point x is denoted as $\partial f(x)$. It is defined as:

$$\partial f(x)=\{v\in \mathrm{R}n \mid f(y)\geq f(x)+\langle v,y-x\rangle,\ \forall y\in \mathrm{R}n\}$$

- Here, $\langle v,y-x\rangle$ represents the inner product between v and $y-x$.
- The subdifferential can be seen as a generalization of the gradient for non-differentiable convex functions. If a convex function is differentiable at a point x, then its subdifferential at that point reduces to a singleton set containing the gradient of the function at x.
- In summary, Jensen's inequality and subdifferentials are both important concepts in convex analysis.
- Jensen's inequality relates the expectation of a convex transformation to the convex transformation of the expectation, while subdifferentials provide a tool for characterizing the subgradients of convex functions at non-differentiable points.

Convex optimization problems - Unconstrained convex optimization problems

- Unconstrained convex optimization problems involve the minimization of a convex objective function without any constraints on the decision variables. In mathematical terms, the general form of an unconstrained convex optimization problem can be expressed as:
- Minimize f(x) - where: x is the vector of decision variables, f(x) is the objective function, and the objective function f(x) is convex.
- Here, convexity implies that the objective function is defined on a convex set and has certain properties, such as the function value being below the chord connecting any two points in the domain. Convex optimization problems have the desirable property that any local minimum is also a global minimum.
- The optimization problem seeks to find the values of the decision variables (vector x) that minimize the objective function. Common examples of unconstrained convex optimization problems include linear programming problems, quadratic programming problems, and more general convex programming problems.

Convex optimization problems - Unconstrained convex optimization problems

- It's important to note that while these algorithms are suitable for convex optimization problems, they may not perform well or may not converge to the global minimum for non-convex problems. In the case of non-convex problems, there are additional challenges and techniques to consider.

Convex optimization problems - Unconstrained convex optimization problems

- Several algorithms can be used to solve unconstrained convex optimization problems. Some common methods include:

1. **Gradient Descent:** Iterative optimization algorithm that moves towards the steepest decrease in the objective function.
2. **Newton's Method:** Iterative optimization algorithm that uses both the gradient and the Hessian matrix of the objective function to find the minimum.
3. **Conjugate Gradient Method:** Iterative optimization algorithm that efficiently combines the advantages of gradient descent and Newton's method.
4. **Quasi-Newton Methods:** Iterative optimization algorithms that approximate the Hessian matrix without explicitly computing it.
5. **Interior-Point Methods:** A class of algorithms that iteratively move towards the interior of the feasible region, converging to the optimal solution.

Linearly constrained convex optimization problems

- Linearly constrained convex optimization problems involve minimizing (or maximizing) a convex objective function subject to linear equality and/or inequality constraints. The general form of such a problem can be expressed as:
- Minimize $f(x)$
- Subject to $Ax=b$

  $Gx \leq h$
- Here:
- x is the vector of decision variables that you want to optimize.
- $f(x)$ is the convex objective function you want to minimize.
- $Ax=b$ represents linear equality constraints.
- $Gx \leq h$ represents linear inequality constraints.

Linearly constrained convex optimization problems

- The matrices A and G and the vectors b and h define the linear constraints. The objective function $f(x)$ is assumed to be convex, which means that its Hessian matrix is positive semidefinite.
- Linearly constrained convex optimization problems have numerous applications in various fields, including engineering, finance, machine learning, and operations research.
- Several methods can be used to solve linearly constrained convex optimization problems, such as:

1. **Interior-point methods:** These methods solve the optimization problem by iteratively moving through the interior of the feasible region.
2. **Sequential quadratic programming (SQP):** SQP methods solve a sequence of quadratic subproblems, which are approximations of the original optimization problem.
3. **Primal-dual interior-point methods:** These methods simultaneously update primal and dual variables and move towards the solution in the interior of the feasible region.
4. **Augmented Lagrangian methods:** These methods solve a sequence of unconstrained optimization problems by adding penalty terms that penalize violations of the constraints.

Linearly constrained convex optimization problems

- It's important to note that the choice of the optimization method depends on various factors, including the problem size, characteristics, and available resources. Additionally, software packages and libraries like CVX, CVXPY, and MATLAB can be used to solve linearly constrained convex optimization problems, providing convenient interfaces for specifying and solving such problems.

Quadratically constrained convex optimization problems

- Quadratically constrained convex optimization (QCQP) problems are mathematical optimization problems that involve the minimization (or maximization) of a convex objective function subject to quadratic constraints. The general form of a QCQP is given by:

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, m \\ & h_j(x) = 0, & j = 1, \ldots, p \\ & q_k(x) \leq 0, & k = 1, \ldots, n_q, \end{array}$$

- where:
- *x* is the optimization variable.
- *f*(*x*) is a convex objective function.
- *gi*(*x*) are convex inequality constraints.
- *hj*(*x*) are equality constraints.
- *qk*(*x*) are quadratic constraints.

Quadratically constrained convex optimization problems

- The quadratic constraints *qk*(*x*) are expressed in the form:

$$q_k(x) = x^T P_k x + 2q_k^T x + r_k \leq 0,$$

- where P_k is a positive semidefinite matrix, q_k is a vector, and r_k is a scalar.

- QCQPs are important in various fields, including control theory, machine learning, finance, and robotics. Due to the convexity of the objective function and constraints, QCQPs have certain desirable properties, and efficient numerical methods can be applied to solve them.

- Common methods for solving QCQPs include interior-point methods, augmented Lagrangian methods, and sequential quadratic programming. Many optimization solvers and modeling languages support QCQPs, making it feasible to solve these problems in practice.

Gradient- based methods for convex optimization - Gradient descent and its variants

- Gradient descent is an iterative optimization algorithm used to minimize the cost or loss function in machine learning and optimization problems. It operates by iteratively moving towards the minimum of the cost function by adjusting the model parameters in the direction of the negative gradient. The negative gradient points in the direction of the steepest decrease in the function, allowing the algorithm to reach a local minimum.
- The basic update rule for gradient descent is given by:

$$\theta_{i+1} = \theta_i - \alpha \cdot \nabla J(\theta_i)$$

- θ_i is the current parameter vector,
- α is the learning rate (step size),
- $\nabla J(\theta_i)$ is the gradient of the cost function J with respect to the parameters θ_i.

Gradient- based methods for convex optimization - Gradient descent and its variants

- Here are some variants of gradient descent:

1. **Batch Gradient Descent:**
   - In batch gradient descent, the entire training dataset is used to compute the gradient of the cost function.
   - It provides a stable convergence but can be computationally expensive for large datasets.
2. **Stochastic Gradient Descent (SGD):**
   - In SGD, only one randomly chosen data point is used to compute the gradient at each iteration.
   - It can be more computationally efficient but may exhibit more variance in convergence.
3. **Mini-Batch Gradient Descent:**
   - Mini-batch gradient descent is a compromise between batch and stochastic gradient descent. It uses a small random subset (mini-batch) of the training data at each iteration.
   - It combines some of the advantages of both batch and stochastic approaches.

Gradient- based methods for convex optimization - Gradient descent and its variants

4. **Momentum:**
   - Momentum helps accelerate SGD in the relevant direction and dampens oscillations.
   - It introduces a moving average of the gradient to the update rule, allowing the algorithm to gain momentum in the direction with consistent gradients.
5. **AdaGrad (Adaptive Gradient Algorithm):**
   - AdaGrad adapts the learning rate for each parameter based on the historical gradients.
   - It performs larger updates for infrequent parameters and smaller updates for frequent parameters.
6. **RMSprop (Root Mean Square Propagation):**
   - RMSprop is an extension of AdaGrad that addresses its aggressive, monotonically decreasing learning rates.
   - It uses a moving average of squared gradients to normalize the learning rates.
7. **Adam (Adaptive Moment Estimation):**
   - Adam combines ideas from momentum and RMSprop.
   - It maintains a moving average of both the gradients and their squared values, adjusting the learning rates accordingly.

Gradient- based methods for convex optimization - Gradient descent and its variants

- These variants aim to improve the convergence speed, handle different types of data, and adaptively adjust learning rates. The choice of the variant depends on the specific characteristics of the optimization problem and the available data.

Stochastic gradient descent

- Stochastic Gradient Descent (SGD) is an optimization algorithm commonly used in machine learning and deep learning to minimize the cost function or loss function during the training of a model. It is a variant of the traditional gradient descent algorithm that is particularly well-suited for large datasets.
- Here's a basic overview of how SGD works:

1. **Gradient Calculation:**
    - In each iteration of the algorithm, a random subset (mini-batch) of the training data is selected.
    - The gradient of the cost function with respect to the model parameters is calculated using only the data points in the mini-batch.
2. **Update Parameters:**
    - The model parameters are updated in the opposite direction of the gradient, aiming to reduce the value of the cost function.
    - The learning rate, which determines the size of the step taken during the update, is multiplied by the gradient to control the step size.
3. **Iterative Process:**
    - Steps 1 and 2 are repeated for multiple iterations or until a convergence criterion is met.

Stochastic gradient descent

- The main advantages of SGD include its ability to handle large datasets more efficiently and its potential to escape local minima due to the stochastic nature of the updates. However, the randomness introduced by using mini-batches can lead to a noisy convergence process.

- There are variations of SGD that incorporate momentum, learning rate schedules, and adaptive learning rates to improve convergence and stability. Some popular variants include Mini-Batch SGD, Momentum SGD, Adagrad, RMSprop, and Adam.

- In summary, SGD is a powerful optimization algorithm that is widely used in training machine learning models, especially in the context of deep learning where large datasets are common.

Momentum methods and Nesterov's method

- Momentum methods and Nesterov's method are optimization algorithms commonly used in machine learning for training deep neural networks. They are variations of the gradient descent algorithm designed to accelerate convergence and improve the optimization process.

1. **Gradient Descent:** The basic idea behind optimization algorithms is to minimize a cost or loss function by adjusting the parameters of a model. Gradient descent is one such optimization algorithm that iteratively updates the model parameters in the direction opposite to the gradient of the cost function with respect to the parameters.

- Mathematically, the update rule for gradient descent is given by:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$

Where:

- θ_t is the parameter vector at iteration t,
- α is the learning rate, and
- $\nabla J(\theta_t)$ is the gradient of the cost function with respect to the parameters.

Momentum methods and Nesterov's method

2. **Momentum Method:** Momentum is a technique that introduces a velocity term to the gradient descent update rule. This helps the optimization algorithm to accelerate in the relevant direction and dampen oscillations.

- The update rule for momentum is given by:

$$v_{t+1} = \beta v_t + (1 - \beta)\nabla J(\theta_t)$$

$$\theta_{t+1} = \theta_t - \alpha v_{t+1}$$

Where:

- v_t is the velocity at iteration t,
- β is the momentum parameter (usually close to 1),
- α is the learning rate, and
- $\nabla J(\theta_t)$ is the gradient of the cost function with respect to the parameters.

Momentum methods and Nesterov's method

3. **Nesterov's Accelerated Gradient (NAG) or Nesterov Momentum:** Nesterov's method is a modification of the momentum method that helps to reduce oscillations and overshooting by adjusting the update rule.

- The update rule for Nesterov's method is given by:

$$v_{t+1} = \beta v_t + (1-\beta)\nabla J(\theta_t - \beta v_t)$$
$$\theta_{t+1} = \theta_t - \alpha v_{t+1}$$

- Nesterov's method calculates the gradient not at the current parameter values but at an adjusted position, incorporating the momentum term. This helps in anticipating the future position of the parameters and adjusting the updates accordingly.

Momentum methods and Nesterov's method

- In summary, momentum methods, including Nesterov's method, enhance gradient descent by introducing a notion of velocity.

- These techniques can help overcome obstacles like oscillations and slow convergence, making the optimization process more efficient.

- Choosing appropriate hyperparameters, such as the learning rate and momentum parameter, is crucial for the success of these methods in practice.

Self Assessments

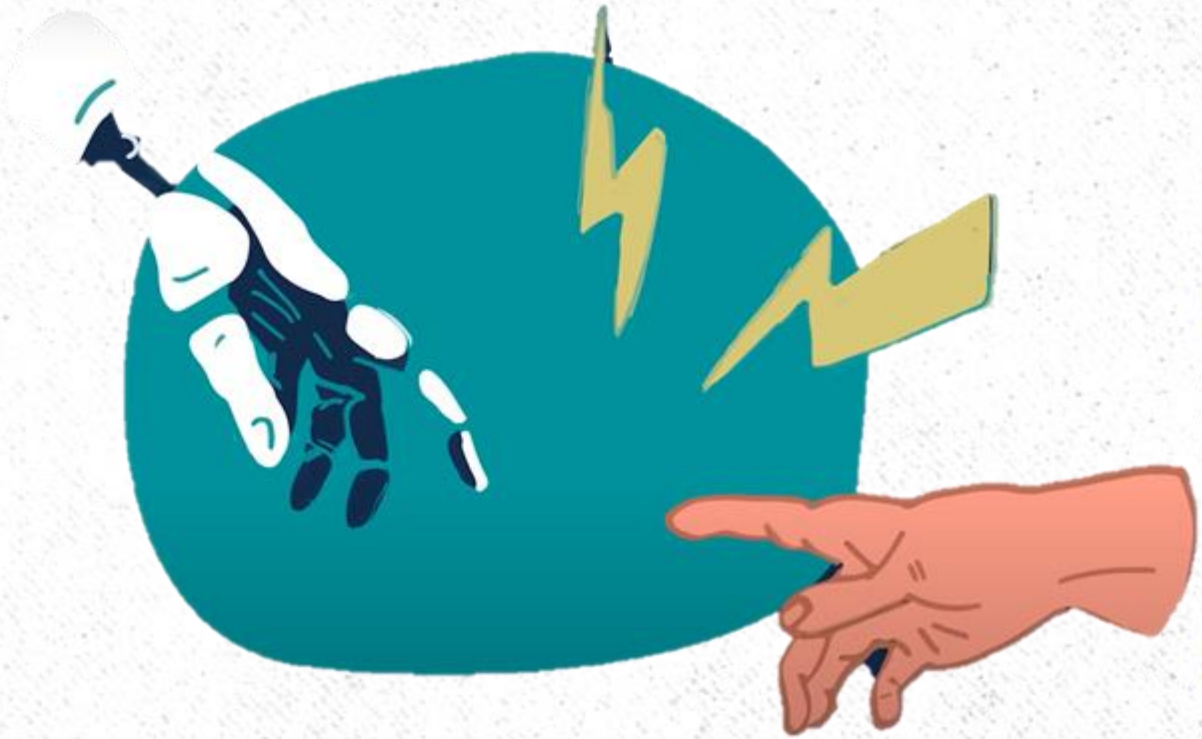
- **Question 1:** What is the definition of a convex function?
- **Question 2:** How do you mathematically express convexity for a real-valued function?
- **Question 3:** Provide a geometric interpretation of convex functions.
- **Question 4:** Explain how the epigraph of a convex function is related to its convexity.
- **Question 5:** State Jensen's inequality and explain its significance in the context of convex functions.
- **Question 6:** Provide an example where Jensen's inequality is applicable.
- **Question 7:** What is the first-order condition for convexity of a differentiable function?
- **Question 8:** Explain how the slope of the tangent line relates to convexity.
- **Question 9:** State the second-order condition for convexity of a twice-differentiable function.
- **Question 10:** How does the positive definiteness of the Hessian matrix relate to convexity?
- **Question 11:** Define a convex set.
- **Question 12:** Explain the relationship between convex sets and convex functions.

Self Assessments

- **Question 13:** What operations preserve convexity of a function?
- **Question 14:** Provide examples of functions that result from applying these operations to convex functions.
- **Question 15:** Define a concave function and highlight the differences between convex and concave functions.
- **Question 16:** Can a function be both convex and concave?
- **Question 17:** How are convex functions utilized in the context of optimization?
- **Question 18:** Explain why convex optimization problems are considered more tractable than non-convex ones.

Activities

- Surprise Quiz in Class



Did You Know?

1. **Optimization Problems:**
    - Convex optimization problems are extensively studied and widely applied in fields such as engineering, finance, and operations research. Many real-world problems can be formulated as convex optimization problems, and efficient algorithms exist to find the global minimum.
2. **Economics and Game Theory:**
    - Convex functions play a crucial role in economic modeling and game theory. Utility functions in economics are often assumed to be convex, representing the idea that more is preferred to less. This assumption helps in analyzing consumer behavior and market equilibrium.
3. **Machine Learning:**
    - Convex optimization is a key component in various machine learning algorithms, particularly in training models. Many optimization problems in machine learning, such as linear regression, logistic regression, and support vector machines, involve convex functions.

Did You Know?

4. **Signal Processing:**
   - In signal processing, convex optimization is used for tasks like signal denoising, image reconstruction, and compressive sensing. Convex formulations are preferred due to the existence of efficient algorithms and the ability to find globally optimal solutions.
5. **Finance:**
   - Convex optimization is applied in portfolio optimization, risk management, and option pricing in finance. It helps in finding optimal investment strategies and managing risk effectively.
6. **Control Systems:**
   - Convex optimization techniques are used in control systems to design controllers that optimize certain performance criteria. Convex formulations are particularly useful for linear time-invariant systems.
7. **Statistics:**
   - Convex functions are employed in statistical estimation problems. The convexity property often simplifies the optimization process and ensures the convergence of optimization algorithms.

Did You Know?

8. **Geometry and Computer Graphics:**
   - Convex functions and sets have applications in computational geometry, collision detection, and computer graphics. Convex hulls, for example, are widely used in geometric algorithms.
9. **Network Optimization:**
   - Convex optimization is applied in network design, resource allocation, and routing problems. It helps in optimizing the flow of resources through a network efficiently.
10. **Mechanical Engineering:**
    - Convex optimization is used in the design and control of mechanical systems. It helps in optimizing the performance of systems subject to various constraints.

Summary

Outcomes:

- Able to know the fundamentals of numerical optimization

Terminal Questions

1. What is the basic idea behind gradient descent?
2. What are the limitations of standard gradient descent algorithms?
3. How does Nesterov Accelerated Gradient (NAG) differ from standard gradient descent?
4. What role does momentum play in accelerating gradient methods?
5. What is the role of learning rate in accelerating gradient methods?
6. How does the Accelerated Gradient Descent algorithm address the problem of oscillations in optimization?

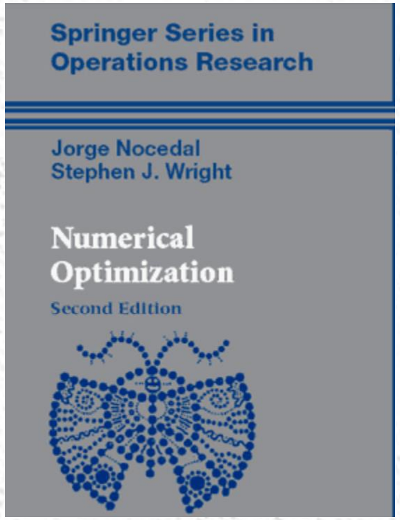
References

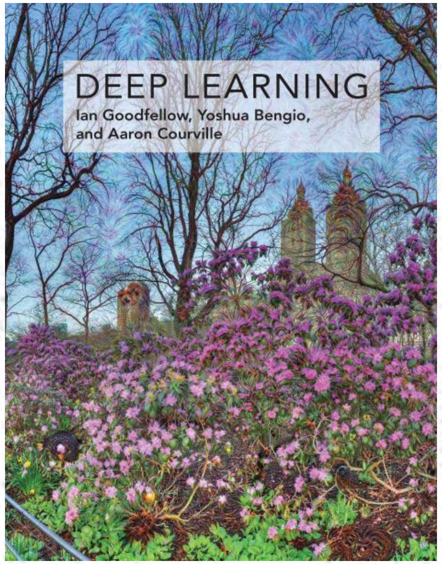
TEXT BOOKS

- "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright
- “Convex Optimization" by Stephen Boyd and Lieven Vandenberghe
- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

REFERENCES

- "Optimization Methods for Large-Scale Machine Learning" by Léon Bottou,Frank E. Curtis, and Jorge Nocedal
- "Nonlinear Programming" by Dimitri P. Bertsekas





Thank you