ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

M.Sc. Program in Data Science
Department of Informatics

Optimization Techniques

Convex Optimization

Instructor: G. ZOIS
georzois@aueb.com

Outline

- Convex sets
  - Definitions and basic concepts
- Convex functions
  - Equivalent definitions
  - Advantages when optimizing convex functions
- Convex optimization problems
  - Unconstrained optimization
    - Descent methods
  - Constrained optimization
    - Lagrange duality and the KKT conditions
    - Algorithms

Our goals

- Formulate problems where the objective function or the constraints are not linear
- Understand when can we have efficient algorithms for solving “non-linear” programs
  - What assumptions are needed for the type of constraints or for the objective function?
- Generalize LP duality theory/sensitivity analysis?

Introduction to convex sets and convex functions

Convex Sets

We focus on subsets of R^n for some dimension $n \geq 1$

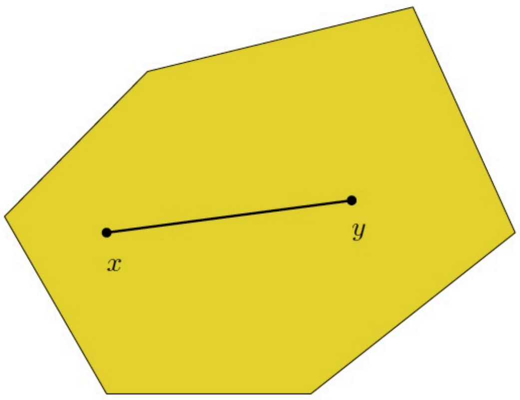
- Points here correspond to n-dimensional vectors
- But intuition from low dimensions very useful

- Given 2 points $x, y \in R^n$, a point z lies on the line that connects x and y if and only if

$$z = ax + (1-a)\, y \text{ for some } a \in [0, 1]$$

Definition: A set $C \subseteq R^n$ is convex if for any 2 points $x, y \in C$, and for any $a \in [0, 1]$, we have that $ax + (1-a)y \in C$

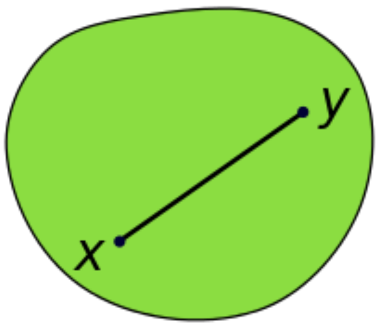
- In geometric terms: the line connecting any 2 points of C, must entirely belong to C

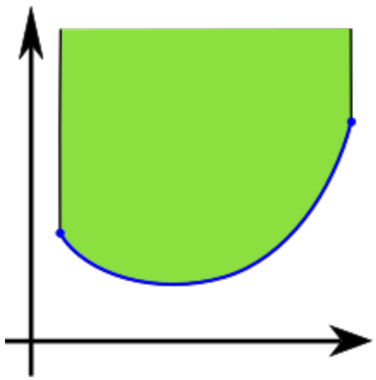


Convex Sets

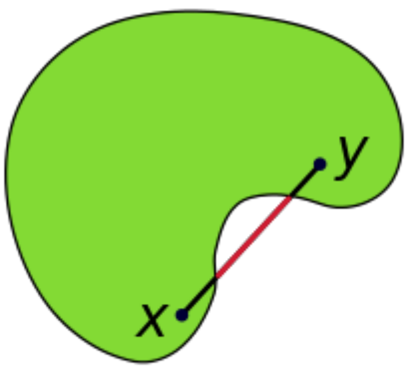
Examples

Convex sets:





Nonconvex sets:



Convex Sets

Further examples of convex sets

1. All of R^n, for any dimension $n \geq 1$

2. The nonnegative orthant: points with all coordinates nonnegative
   - Since $ax + (1\text{-}a)y$ will also have nonnegative coordinates

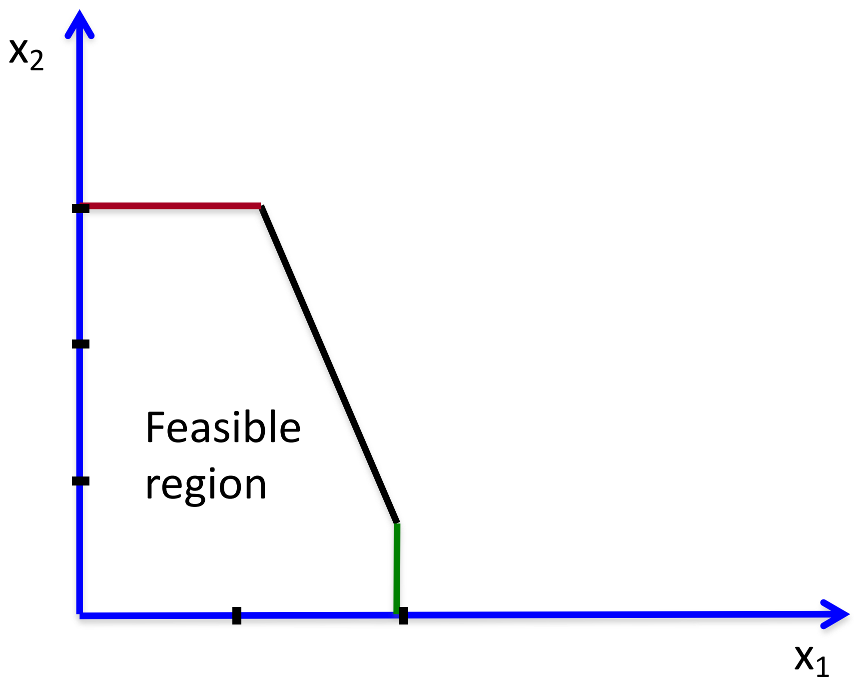
3. The set of points contained within a ball
   - E.g. $\{x: ||x||_2 \leq 1\}$
   - Since
     $||ax + (1\text{-}a)y||_2 \leq ||ax||_2 + ||(1\text{-}a)y||_2 \leq a||x||_2 + (1\text{-}a)||y||_2 \leq 1$

4. Intersections of convex sets

Convex Sets

Further examples of convex sets

5. Feasible region of a linear program
   - Convex polygon in 2 dimensions (when it is bounded)
   - Convexity follows since it is an intersection of halfspaces



Convex Sets

Examples that do not involve R^n

- The exact same definition of convexity can be applied for elements that are not points of R^n

Definition: A real symmetric n x n matrix A is called positive semidefinite (PSD) if for every n-dimensional vector z

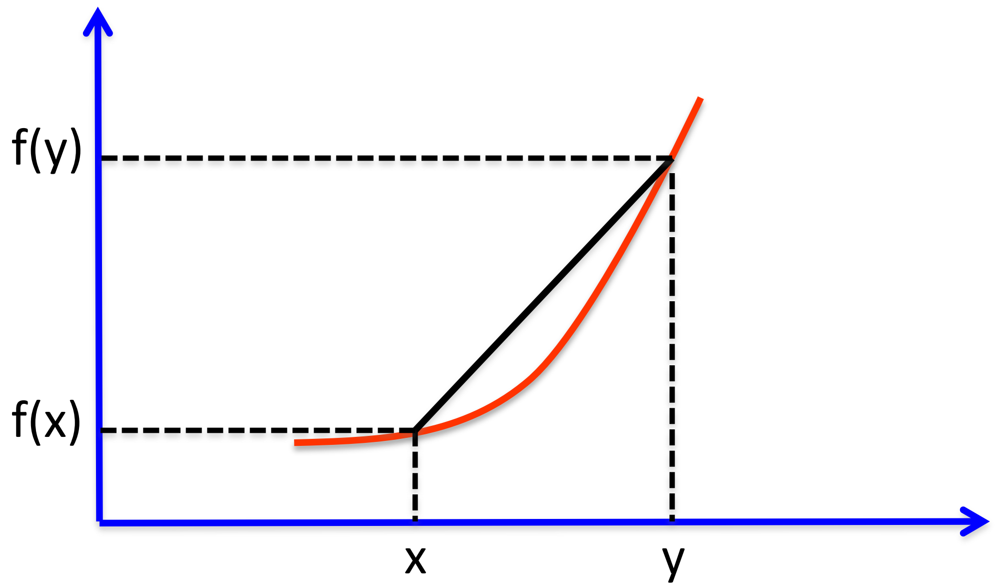
$$z^T \cdot A \cdot z \geq 0$$

Claim: The set of PSD matrices is a convex set
i.e., if A and B are PSD matrices, then $\lambda A + (1-\lambda)B$ is also PSD for any $\lambda \in [0, 1]$

Convex Functions

Definition: A function $f : R^n \to R$ is convex if for any 2 points x, y, and for any $a \in [0, 1]$, we have that

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



- Geometric interpretation: the line connecting any 2 points (x,f(x)) (y,f(y)) must lie on or above the graph of the function

Convex Functions

Examples

- With 1 variable:
  - Exponential functions with base > 1: 2^x, e^x, c^x for $c \geq 1$
  - Polynomial functions: x^3, x^{10}, x^c, for $c \geq 1$
  - Linear functions: we have exact equality in the definition

- With many variables
  - Exponential functions: e^{x+y}, e^{x+y+z},
  - Negative of logarithms: $-\log(x + y)$
  - The sum of convex functions remains convex

Convex Functions

Equivalent definitions

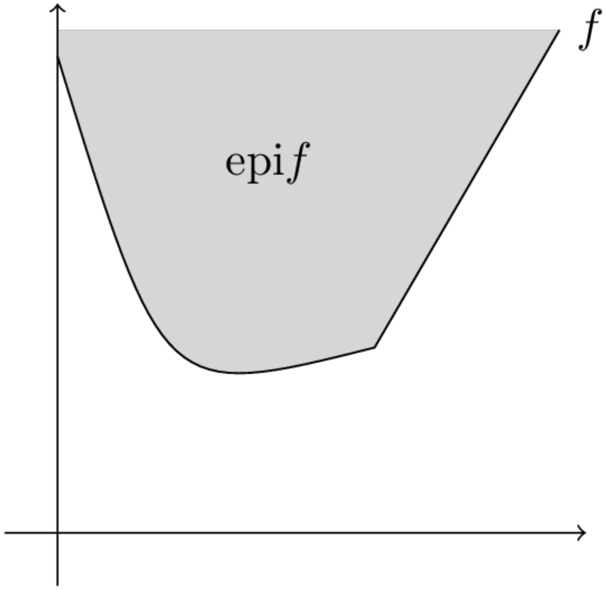
(1)Based on the epigraph

The epigraph of a function f is the set:

$$\text{epi } f = \{ (x, t) : t \geq f(x) \}$$

f is convex if and only if the epigraph of f is a convex set

- Geometric interpretation: the set of points that lie on or above the graph of the function should be a convex set



Convex Functions

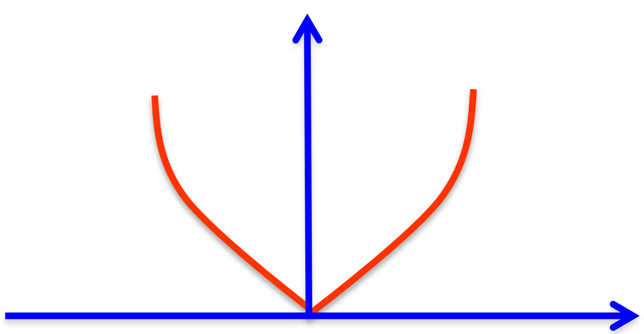
Equivalent definitions

(2) Based on the partial derivatives

Suppose that f is twice differentiable

For functions with 1 variable: f is convex if and only if $f''(x) \geq 0$, for every x

- The first derivative is increasing



e.g. for $f(x) = x^2$, $f''(x) = 2$ for every x

Convex Functions

Equivalent definitions

(2) Based on the partial derivatives

For functions with n variables: Define the Hessian of f at point x as the n x n array:

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

A function f is convex if and only if the Hessian is positive semidefinite for every x

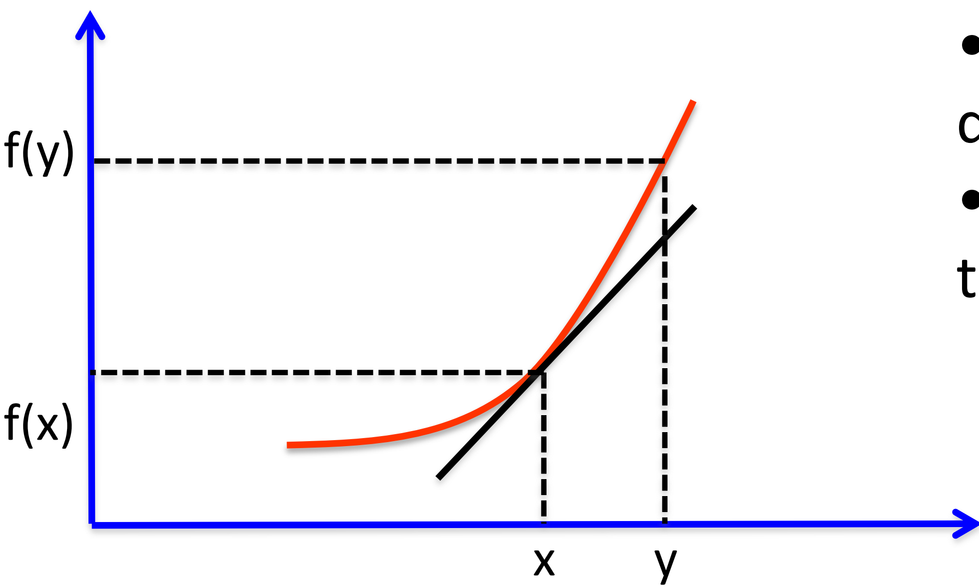
- Easy to see from Slide 13 that this holds for 1 dimension

Convex Functions

Equivalent definitions

(3) Based on the tangents to the graph of f

f is convex if and only if the graph of f lies on or above all its tangents:



Algebraically in 1 dimension:

- Slope of tangent at x = the derivative of f at x
- Hence, if f lies above all its tangents, then for every x, y:

$$f(y) \geq f(x) + f'(x)\,(y-x) \quad (*)$$

Convex Functions

Equivalent definitions

(3) Based on the tangents to the graph of f

For functions with n variables:

•Recall the gradient of a function

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)$$

- The gradient shows the rate of increase/decrease along each dimension just as the derivative does for one variable
- Generalizing (*) for many variables:

$$f(y) \geq f(x) + \nabla f(x)^T \cdot (y-x) \quad (**)$$

One of the most important properties of convex functions

Concave Functions

Sometimes we may also discuss concave functions

Definition: A function $f : R^n \rightarrow R$ is concave if for any 2 points x, y, and for any $a \in [0, 1]$, we have that

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

- If f is concave, -f is convex
- Hence, maximizing a concave function f can be reduced to minimizing a convex function

Convex Optimization Problems

Nonlinear Optimization Problems

General form of optimization problems:

- Both equality and inequality constraints present

$$
\begin{aligned}
&\min \ f(x) \\
&\text{s. t.:} \\
&\qquad g_i(x) \le 0, \quad i = 1, 2, \ldots, m \\
&\qquad h_i(x) = 0, \quad i = 1, 2, \ldots, p
\end{aligned}
$$

We say the above is a convex optimization problem when

- $f(x)$ is a convex function
- Each g_i is a convex function
- Each h_i is an affine function, $h_i = a_i^T x - b_i$

Convex Optimization

Applications of convex optimization:

- Machine learning: linear regression (least squares), classification (logistic regression, support vector machines)
- Statistics: parameter estimation
- Control theory
- Signal processing
- And many many more...

Convex Optimization

- For general non-convex problems, almost no hope
- There is no general approach that can work for any arbitrary optimization problem
- Some families of non-convex problems can be handled
- But when working under assumptions like convexity, or related properties (e.g. strong convexity), we can have guarantees for convergence and running time
- Still however not a standard technology, contrary to LP solvers
  - Commercial availability not as large as for LP solving but gradually changing

Unconstrained Convex Optimization

Unconstrained Optimization

- We start with the easier version that has no constraints

- Suppose we just want to minimize a function $f : R^n \rightarrow R$ without any further constraints
  - Still interesting problem with many applications

- Assumption: f is twice continuously differentiable

- Necessary condition for a point x^* to be a minimum is

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

- BUT: for an arbitrary function f:
  - This is not a sufficient condition, many other points may satisfy this (such as local optima)

Unconstrained Optimization

- We start with the easier version that has no constraints

- Suppose we just want to minimize a function $f : R^n \to R$ without any further constraints
  - Still interesting problem with many applications

- Assumptions from now on (unless otherwise stated)
  - f is **convex**, and twice continuously differentiable
  - The minimum of f is attained (and $\neq +\infty$ or $-\infty$)

Unconstrained Optimization

Why is it nice to be convex:

Theorem: For a convex function f, a point x^* is a global minimum of f if and only if $\nabla f(x^*) = 0$

Proof:

Recall the basic property of convex functions, i.e., inequality (**):

$$f(y) \geq f(x) + \nabla f(x)^T \cdot (y-x) \text{ for any 2 points } x, y$$

- Suppose there exists x^* for which $\nabla f(x^*) = 0$
- Then for every point y, inequality (**) implies $f(y) \geq f(x^*)$
- Hence x^* is a global minimum

Unconstrained Optimization

Why is it nice to be convex:

- The theorem makes our lives much easier (not trivial however)
  - It suffices to find a point where the derivatives become 0
  - Local minima are global minima, as with linear programs (recall the terminating condition of simplex)
  - If we can solve analytically the system $\nabla f(x) = 0$, then no need for an algorithm
- In many cases convexity helps us exploit the geometric intuition we have from polyhedra or linear programming problems

Unconstrained Optimization

Algorithms for convex unconstrained optimization:

- Iterative algorithms, updating a current feasible solution
- They produce a sequence of points $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$ with the property that

$$f(x^{(k)}) \rightarrow p^* \quad \text{as } k \rightarrow \infty$$

  - p^* is what we are after: $p^* = \inf_x f(x)$
  - We may never find the actual optimal solution
  - But we can get very close, in fact arbitrarily close if we allow enough iterations
- We can view these algorithms as iterative methods for solving the system $\nabla f(x) = 0$

Descent Methods

General form of descent methods

- Make a local update towards an appropriate direction
- Stop when $\nabla f(x)$ is close to 0

- Initialization: $k=0$, pick a starting point $x^{(0)}$, and a step size α_0
- Update:
  - Check if stopping criterion satisfied
  - If not, $x^{(k+1)} = x^{(k)} + \alpha_k \Delta x^{(k)}$
  - k++
- Usual stopping criterion: $||\nabla f(x^{(k)})||_2 \le \varepsilon$

Terminology:
- Δx: search direction
- α_k: step size, with $\alpha_k > 0$

Descent Methods

How should we pick the search direction?

- Need to ensure that for every iteration k, $f(x^{(k+1)}) \leq f(x^{(k)})$
- Convexity, i.e. using (**), implies it suffices to enforce that:

$$\nabla f(x^{(k)})^T \cdot \Delta x^{(k)} < 0$$

- Hence: choosing the (negative) gradient itself for the search direction is a safe choice!

The Gradient Descent Method

One of the simplest algorithms in optimization: Descend according to the gradient direction

- Initialization: $k=0$, pick a starting point $x^{(0)}$, and a step size α_0
- Update:
  - Check if stopping criterion satisfied
  - If not, $x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$
  - k++
- Stopping criterion $||\nabla f(x^{(k)})||_2 \leq \varepsilon$
  - If e.g., $||\nabla f(x^{(k)})||_2 \leq (2m\varepsilon)^{1/2}$, where m is a lower bound on the minimum eigenvalue of $H(f,x)$, then $f(x) - p^* \leq \varepsilon$

The Gradient Descent Method

How should we pick the step size α_k?

- First idea: Exact line search
  - Find the minimum value of f along the gradient direction:

$$\alpha_k = \text{argmin}_s\ f(x^{(k)} - s\nabla f(x^{(k)}))$$

  - 1-dimensional problem
  - E.g., we could solve it via Newton's method
  - But often too time consuming in practice

The Gradient Descent Method

How should we pick the step size α_k?

- Second idea: Backtracking line search, an approximate solution to the exact line search
  - Try to approximately minimize f along the ray $x - s\nabla f(x^{(k)})$
  - Essentially make sure the function decreases "enough"
  - Many variants in the literature, e.g.

  Keep setting s:= βs until

  $$f(x - s\nabla f(x^{(k)})) \leq f(x^{(k)}) - \alpha s \cdot ||\nabla f(x^{(k)})||_2^2$$

  for $\beta < 1$, $\alpha < 1/2$

  - Works well in practice

Descent Methods

Example 1:

Consider the function $f(x_1, x_2) = x_1^2 + 2x_2^2 - 2x_1 x_2$

Execute the first 2 steps of gradient descent with exact line search, starting from $x^{(0)} = (1, 1)$

Descent Methods

Example 2:

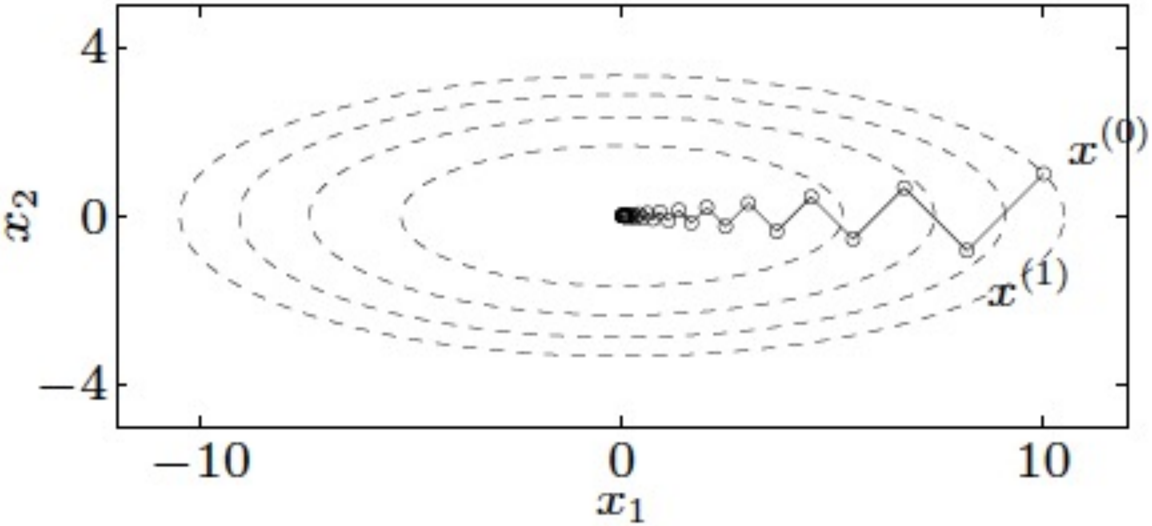
Consider the function $f(x_1, x_2) = \frac{1}{2}(x_1^2 + \gamma x_2^2), \quad \gamma > 0$

Start at $x^{(0)} = (\gamma, 1)$

After k iterations of gradient descent, we get:

$$x^{(k)} = \left(\gamma\left(\frac{\gamma-1}{\gamma+1}\right)^k, \left(-\frac{\gamma-1}{\gamma+1}\right)^k\right)$$

Run with $\gamma = 10$



Convergence analysis

Can we establish convergence properties for the gradient descent method?

- Empirically, it works well on average for convex functions
- Theoretically, upper bounds can be obtained when assuming strong convexity

Definition: A function is strongly convex when there exists $m>0$ such that for any x,

$$H(f, x) \geq m \cdot I$$

- I is the identity matrix

Convergence analysis

- Strong convexity together with (**) implies that there also exists upper bounds on the Hessian
- Hence, there exist m>0 and M>0 such that for every x:

$$m \cdot I \leq H(f, x) \leq M \cdot I$$

- Convergence results on the number of iterations depend on
  - m and M
  - The initial solution $x^{(0)}$
  - The accuracy parameter in the stopping criterion
- Note: we may not be aware of the values for m and M
  - It might be difficult to estimate for some functions
  - So, we may not know how many iterations we need
- Still, these bounds are conceptually useful
  - They provide a guarantee that the method converges

Convergence analysis

- A relatively loose analysis with exact line search
- Theorem: For strongly convex functions, the number of iterations required by the gradient descent method is bounded by

$$\log((f(x^{(0)}) - p^*)/\varepsilon) \,/\, \log(1/c)$$

  where
  - $c = 1 - m/M < 1$
  - $p^* = \min_x f(x)$
  - ε = accuracy parameter (= final suboptimality)
  - $f(x^{(0)}) - p^*$ = initial suboptimality
  - Thus, nominator = log of initial suboptimality to final suboptimality
- Conclusions: The error $f(x^{(k)}) - p^*$ converges to 0 at least as fast as a geometric series
  - i.e., linear convergence
- With backtracking line search, slightly worse bounds can also be established

The Newton Method

A different descent method with favorable performance

- It is instructive to see first the method in 1 dimension
  - When n=1, we search for a point x, where f'(x) = 0
  - Suppose after k iterations, we have reached a point x_k
  - How shall we move to the next iteration and pick x_{k+1}?

Newton's method for n=1:

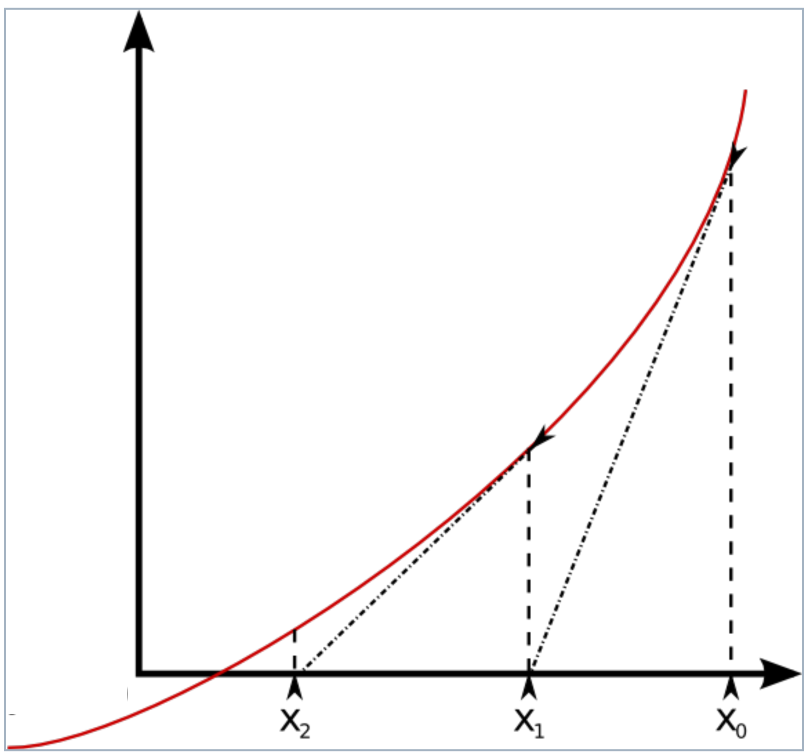
$$x_{k+1} = x_k - (f'(x_k)/f''(x_k))$$

Also referred to as the Newton-Raphson method

The Newton Method

Newton's method for n=1:

$$x_{k+1} = x_k - (f'(x_k)/f''(x_k))$$



Geometric interpretation:

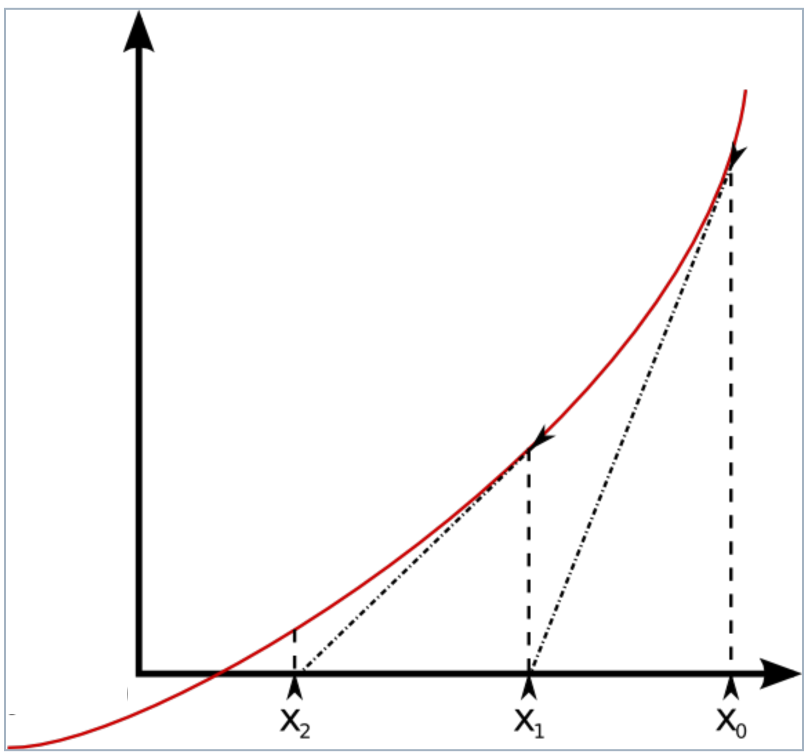
- Consider the plot of the derivative f'
- By convexity the first derivative is an increasing function
- Draw the tangent at x_k
- Slope of the tangent = $f''(x_k)$
- Find the point where the tangent hits the x-axis
- This is given by solving the equation

$$0 = f'(x_k) + f''(x_k)(x - x_k)$$

The Newton Method

Newton's method for n=1:

$$x_{k+1} = x_k - (f'(x_k)/f''(x_k))$$



Algebraic intuition:

- Consider the 2nd order Taylor approximation:

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + f''(x_k)(x_{k+1} - x_k)^2/2$$

- How would we choose to move from x_k to x_{k+1}?
- Set derivative (with respect to x_{k+1}) = 0

$$\Rightarrow x_{k+1} = x_k - f'(x)/f''(x)$$

- X_{k+1} is the minimizer of g
- If f is close to a quadratic function, then the Newton step is close to the best possible

The Newton Method

For many variables, we can generalize the same intuition:

- 2nd order Taylor approximation for a function of n variables
- Now x and δ are n-dimensional vectors

$$f(x+\delta) = f(x) + \nabla f(x)^T \cdot \delta + \tfrac{1}{2}\, \delta^T \cdot H(f, x) \cdot \delta$$

If we try to minimize with respect to δ (=Δx), we get that:

$$\delta = - H(f, x)^{-1}\, \nabla f(x)$$

- Is this a descent direction?
- To be aligned with the convexity of f we need to check that: $-\nabla f(x)^T \cdot \mathbf{(}H(f, x)^{-1} \cdot \nabla f(x)) < 0$,

But the Hessian is a PSD matrix!

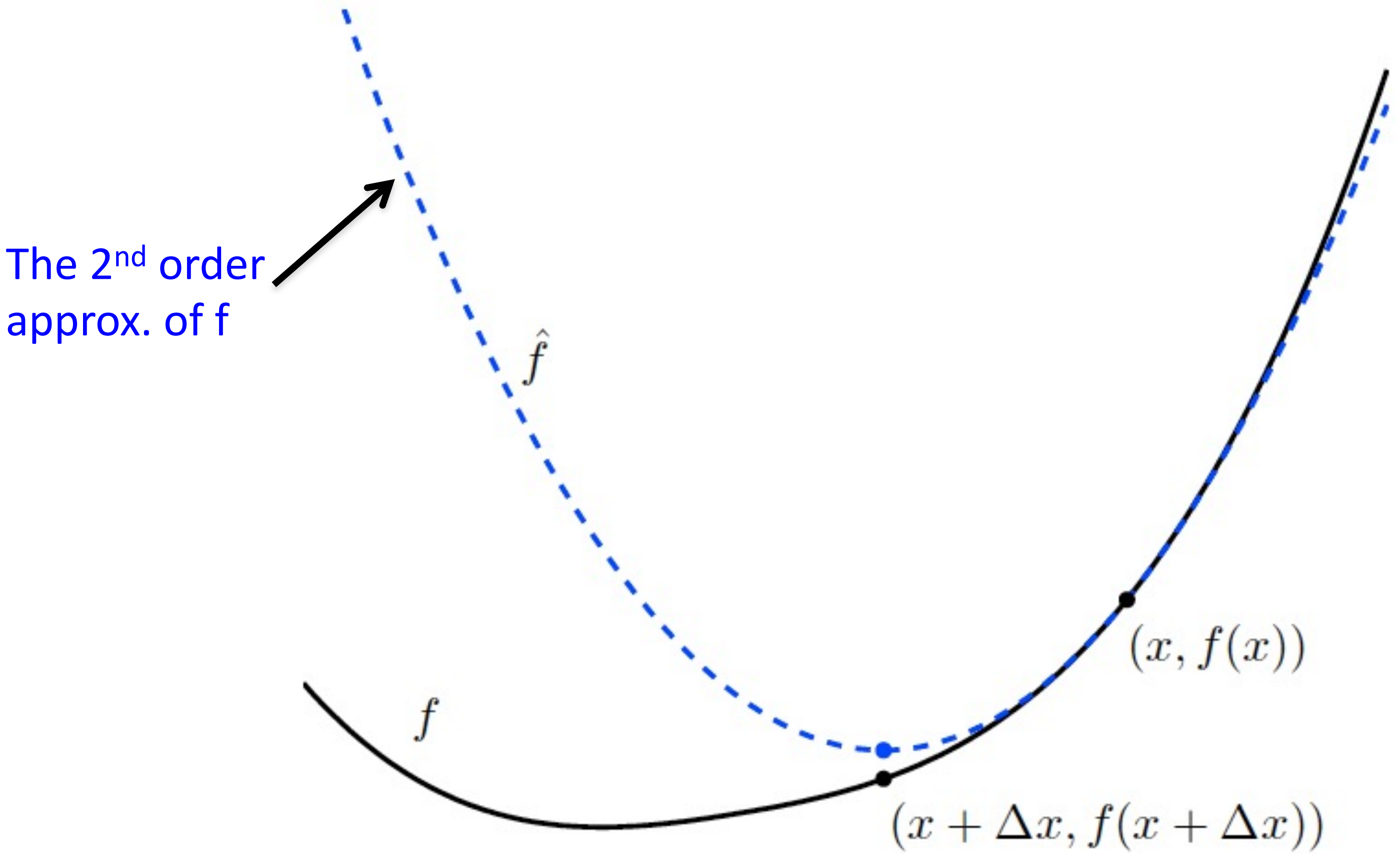
The Newton Method

Summarizing:

- Initialization: $k=0$, pick a starting point $x^{(0)}$, and a step size α_0
- Update:
  - Check if stopping criterion satisfied
  - If not, $x^{(k+1)} = x^{(k)} - \alpha_k H(f, x^{(k)})^{-1} \cdot \nabla f(x^{(k)})$
  - $k{+}{+}$
- Usual stopping criterion:
  - Let $\lambda := (\nabla f(x^{(k)})^T \cdot H(f, x^{(k)}) \cdot \nabla f(x^{(k)}))^{1/2}$
  - Stop when $1/2\lambda^2 \leq \varepsilon$
  - λ is called the Newton decrement
  - Useful parameter for the analysis of the method

The Newton Method

- Progress made using the 2^{nd} order approximation



The Newton Method

- Pros
  - It is fast in general
  - Scales well with problem size
  - Performance not depend on problem parameters (?)
- Cons
  - Cost of computing the Hessian

- Convergence analysis
  - Can be established in a similar way as with gradient descent
  - Theoretical upper bound: proportional to $f(x^{(0)}) - p^*$