TMA947 / MMG621 — Nonlinear optimisation

A course summary

Emil Gustavsson, Michael Patriksson and Caroline Granfeldt

January 2, 2017

Lecture 2 (Convexity)

Convex sets

- $S \subseteq \mathbb{R}^n$ convex set if $\boldsymbol{x}^1, \boldsymbol{x}^2 \in S$ and $\lambda \in (0,1)$ implies that $\lambda \boldsymbol{x}^1 + (1-\lambda)\boldsymbol{x}^2 \in S$.
- The intersection of convex sets is a convex set.
- If the set can be written as $S = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq 0,\ i = 1,\dots,m\}$, where the functions $g_i : \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions, then S is a convex set.
- The *convex hull* of set S is the set of all convex combinations of points in S.
- Caratheodory's theorem: A point $\boldsymbol{x} \in \operatorname{conv} S$, where $S \subseteq \mathbb{R}^n$, can be written as a convex combination of $n+1$ or fewer points of S.
- Polytope = The convex hull of finitely many points.
- Polyhedron = The intersection of finitely many half-spaces.
- Representation theorem: "Polyhedron = Polytope + Polyhedral cone". (Implies "Bounded polyhedron = Polytope").
- An extreme point of a convex set is a point that cannot be expressed as a convex combination of two other points in the set.
- A set $C \subseteq \mathbb{R}^n$ is a cone if $\lambda \boldsymbol{x} \in C$ whenever $\boldsymbol{x} \in C$ and $\lambda > 0$.
- Separation theorem: Either a point lies in a convex set or one can separate the point from the set by a hyperplane.
- Farkas' Lemma: Either a point lies in the polyhedral cone spanned by the columns of a matrix or one can separate it from the cone by a hyperplane.

  Most important fact regarding Farkas' Lemma: To every inconsistent linear system (system that does not have a solution) there exists a corresponding consistent linear system (system that does have a solution).

Convex functions

- $f : S^n \mapsto \mathbb{R}$ is a convex function on S if $f(\lambda \boldsymbol{x}^1 + (1-\lambda)\boldsymbol{x}^2) \leq \lambda f(\boldsymbol{x}^1) + (1-\lambda) f(\boldsymbol{x}^2)$ whenever $\boldsymbol{x}^1, \boldsymbol{x}^2 \in S$ and $\lambda \in (0, 1)$
- A function f is concave if $-f$ is convex.
- The sum of convex functions is a convex function.
- The epigraph of a convex function is a convex set.
- If $f \in C^1$, then f is convex on the convex set S if and only if $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in S$.
- If $f \in C^2$, then f is convex on the convex set S if and only if $\nabla^2 f(\boldsymbol{x}) \succeq 0$ for all $\boldsymbol{x} \in S$.

Convex problem

- A problem is a convex optimization problem if the feasible set is a convex set and the objective function is convex on the feasible set.

Lecture 3 (Optimality conditions)

Consider the problem to

$$\begin{aligned} &\text{minimize} \;\; f(\boldsymbol{x}), \\ &\text{subject to} \;\; \boldsymbol{x} \in S. \end{aligned}$$

- A global minimum is a point which has lowest objective function value on the feasible set.
- A local minimum is a point which has lowest objective function value in a neighborhood of the point.
- Fundamental theorem: Let f be convex on the convex set S. Then every local minimum is also a global minimum.
- Weierstrass' theorem: If S is nonempty and closed, and f is weakly coercive w.r.t. S, then there exists an optimal solution to the problem.

Unconstrained optimization $S = \mathbb{R}^n$

- ($f \in C^1$) If $\boldsymbol{x}^*$ is a local minimum, then $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.
- ($f \in C^2$) If $\boldsymbol{x}^*$ is a local minimum, then $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ and $\nabla^2 f(\boldsymbol{x}^*) \succeq 0$.
- If f is convex then $\boldsymbol{x}^*$ is a global minimum if and only if $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.

Constrained optimization $S \neq \mathbb{R}^n$

- ($f \in C^1$) If $\boldsymbol{x}^*$ is a local minimum, then $\nabla f(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{p} \geq \mathbf{0}$ holds for all feasible directions $\boldsymbol{p}$.
- Suppose S is convex. Then a stationary point $\boldsymbol{x}^*$ is a point fulfilling the following four equivalent statements

$$\nabla f(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^*) \geq 0, \quad \boldsymbol{x} \in S.$$
$$\min_{\boldsymbol{x}\in S} \nabla f(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^*) = 0.$$
$$\boldsymbol{x}^* = \mathrm{Proj}_S[\boldsymbol{x}^* - \nabla f(\boldsymbol{x}^*)].$$
$$-\nabla f(\boldsymbol{x}^*) \in N_S(\boldsymbol{x}^*).$$

  where $N_S(\boldsymbol{x}^*)$ is the normal cone to S at $\boldsymbol{x}^*$.
- If $\boldsymbol{x}^*$ is a local minimum then it is a stationary point.
- If the problem is convex, then a stationary point is a global minimum.

Lecture 4 (Unconstrained optimization)

- Use line search type algorithms. Iteratively update iterates $\boldsymbol{x}_k$ by taking steps α_k in the directions $\boldsymbol{p}_k$.
- Steepest descent: $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$.
- Newtons algorithm: Solve $\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$
- Levenberg-Marquardt: Solve $(\nabla^2 f(\boldsymbol{x}_k) + \gamma I)\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$
- Step lengths α_k can be found using Armijos step length rule. Basic idea: Accept a step length if the decrease in objective function value is at least a portion of the predicted decrease.

Lecture 5/6 (Optimality conditions)

Geometric optimality conditions

- The intuitive necessary optimality condition is that "if a point $\boldsymbol{x}^*$ is a local minimum, it should not be possible to draw a curve starting at $\boldsymbol{x}^*$ inside S such that the objective function f decreases along it".
- The set of all possible curves starting at $\boldsymbol{x}$ and moving inside S is the tangent cone

$$T_S(\boldsymbol{x}) := \{\boldsymbol{p} \mid \exists\{\boldsymbol{x}_k\}, \{\lambda_k\} \geq 0 : \lim \boldsymbol{x}_k = \boldsymbol{x}, \lim \lambda_k(\boldsymbol{x}_k - \boldsymbol{x}) = \boldsymbol{p}\}.$$

- We let $\mathring{F}(\boldsymbol{x}) := \{\boldsymbol{p} \mid \nabla f(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{p} < 0\}$. All the vectors in this set (cone) are descent directions.
- We can now state the intuitive geometric optimality conditions as: If $\boldsymbol{x}^*$ is a local minimum then $T_S(\boldsymbol{x}^*) \cap \mathring{F}(\boldsymbol{x}^*) = \emptyset$.

KKT conditions

- Assume that the feasible set is $S = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, m\}$.
- Let the cone $G(\boldsymbol{x}) = \{\boldsymbol{p} \in \mathbb{R}^n \mid g_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{p} \leq 0,\ i \in \mathcal{I}(\boldsymbol{x})\}$, where $\mathcal{I}(\boldsymbol{x})$ are the active constraints at the point $\boldsymbol{x}$.
- It always holds that $G(\boldsymbol{x}) \subseteq T_S(\boldsymbol{x})$.
- We say that Abadie's constraint qualification holds at a point $\boldsymbol{x} \in S$ if $T_S(\boldsymbol{x}) = G(\boldsymbol{x})$.
- Assuming Abadie's CQ, the geometric optimality conditions can be written as $G(\boldsymbol{x}) \cap \mathring{F}(\boldsymbol{x}) = \emptyset$.

  But this says that a specific linear system is inconsistent. Using Farkas' Lemma, we know that there exists a corresponding consistent system. This system is the KKT system.
- We thus have: If Abadie's CQ holds and $\boldsymbol{x}^*$ is a local minimum, then $\boldsymbol{x}^*$ is a KKT point.
- A point being a KKT point just means that the negative gradient of the objective function can be written as a positive linear combination of the gradients of the active constraints.
- LICQ and Slater's CQ both imply Abadie's CQ.
- If the problem is convex, then every KKT point is a global minimum.

Lecture 7 (Lagrangian duality)

Relaxation

- A relaxation to an optimization problem is a problem where a lower (or equal) objective function is optimized over a larger (or equal) set. In other words,
  1. The original problem's feasible set is a subset of the relaxed problem's feasible set.
  2. The original problem's objective-function is greater than or equal to the relaxed problem's objective-function.
- If the optimal solution to the relaxed problem is feasible in the original problem and has the same objective value, then it is also optimal in the original problem.

Lagrangian relaxation

- Basic idea: Some constraints are complicated, so they are instead added to the objective function with a penalty. A dual problem is then formulated where the objective is to find the optimal penalty parameters.
- $q(\boldsymbol{\mu}) = \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}) + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x})$ is the dual objective function.
- Weak duality: For any feasible $\boldsymbol{x}$ in the primal problem, and any $\boldsymbol{\mu} \geq \mathbf{0}$, it holds that $q(\boldsymbol{\mu}) \leq f(\boldsymbol{x})$.
- If Slater's CQ holds, then also Strong duality: $f^* = q^*$.

Lecture 8/9 (Linear programming)

- Minimize a linear function over a polyhedron.
- If the problem has an optimal solution, then one of the extreme points will be optimal.
- Extreme points of a polyhedron in standard form can be represented as a partition of basic and non-basic variables, i.e., a partition $\boldsymbol{A} = [\boldsymbol{B}, \boldsymbol{N}]$.
- A partition $\boldsymbol{A} = [\boldsymbol{B}, \boldsymbol{N}]$ represents a basic feasible solution (BFS) if $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} \geq \boldsymbol{0}$.
- Degenerate: A BFS is called degenerate if $\boldsymbol{x}_B \not> \boldsymbol{0}$.
- Unboundedness: If $\boldsymbol{B}^{-1}\boldsymbol{N}_j \leq \boldsymbol{0}$, where j is chosen such that the reduced costs for $(\boldsymbol{x}_N)_j$ is negative, then the objective function value is unbounded from below.
- The idea of the simplex method is to iteratively update the partition by replacing one basic variable at a time, until an optimality condition is fulfilled (the reduced costs are all non-negative) or unboundedness is detected.
- Phase I is used for finding a starting BFS (introduce artificial variables which we try to move into the non-basic variables).

Lecture 10 (LP duality)

- If the primal problem has n variables and m constraints, then there exists a corresponding dual problem which has m variables and n constraints.
- Weak duality: For a primal feasible $\boldsymbol{x}$ and a dual feasible $\boldsymbol{y}$, we have that $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$.
- Strong duality: If both problems have feasible solutions, then $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}^* = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*$ for the optimal solutions.
- The dual problem is the same as the Lagrangian dual problem.
- Use canonical form to construct dual problem.
- Complementary slackness: Either the slack or the dual variable corresponding to a constraint is zero.
- Meaning of dual: The dual variable is the measure on how much the optimal value would change if the right-hand side of the constraint is changed. (Now complementary slackness becomes intuitive!)

Lecture 11 (Convex optimization)

– A subgradient $\boldsymbol{p}$ to a function f at a point $\boldsymbol{x}$ is a vector fulfilling $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x})$ for all $\boldsymbol{y}$

– The subdifferential is the set of all subgradients.

– If the function is differentiable, then the subdifferential is the singleton consisting of the gradient of the function.

– A necessary and sufficient optimality condition for unconstrained optimization of a convex function is that the zero vector lies in the subdifferential.

– A subgradient is not necessarily a descent direction. It does, however, cut away half-spaces where the optimal solution does not lie.

Lecture 13 (Feasible direction methods)

– Iterative algorithms: Update points $\boldsymbol{x}_k$ by taking steps α_k in directions $\boldsymbol{p}_k$.

  The directions $\boldsymbol{p}_k$ needs to be feasible directions and the step length must not be too large (the next iterate should stay inside the feasible set).

– Frank-Wolfe algorithm: A search direction is found through the minimization of a linear approximation of the function. The next iterate is taken as the best convex combination of the current point and the optimal solution to the linear approximation.

– Simplicial decomposition: Several optimal solutions to the linear approximations are saved, and the new iterate is taken as the best convex combination of the current point and all saved points.

– Gradient projection algorithm: Let $\boldsymbol{x}_{k+1} = \mathrm{Proj}_S[\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)]$, where the value of the step length α_k is chosen by an approximate line search (such as Armijo).

Lecture 14 (Constrained optimization)

– Basic idea of penalty methods: Replace the constrained optimization problem with an unconstrained one that includes a penalty term representing the constraints. Update the penalty parameter so that the model of the relaxed problem becomes a better and better approximation of the original one.

– Exterior penalty methods: Penalize being infeasible.

– Interior penalty methods: Penalize being close to the constraints (the iterates are always strictly feasible).

– Convergence results of the penalty methods: Under specific assumptions, the iterates converge to KKT points.

– Sequential quadratic programming (SQP): At iterate $\boldsymbol{x}_k$, approximate problem with a QP subproblem using the Lagrangian to find new search direction.

Some Tips & Tricks for the Exam

– Don't start solving problem 1 just because it is the first one. Instead, read through the exam to get an overview and make mental notes on which ones to start with. If you start with something easy, you get a confidence boost and you don't waste too much time on the very toughest ones first (as it could take an unpredictable amount of time to solve those).

– Try to help yourself as much as possible using graphical illustrations; problems are sometimes given in 2D, which means that you can get pictures on what's going on.

– If you are asked to perform a couple of iterations of an algorithm, also here try to illustrate what's going on by graphing it - if it's in 2D, of course.

– Again, when you perform calculations, test that what you do is consistent - such as checking feasibility, that the objective value after a line search is correct, and has in fact decreased if the problem concerns minimisation, and so on.