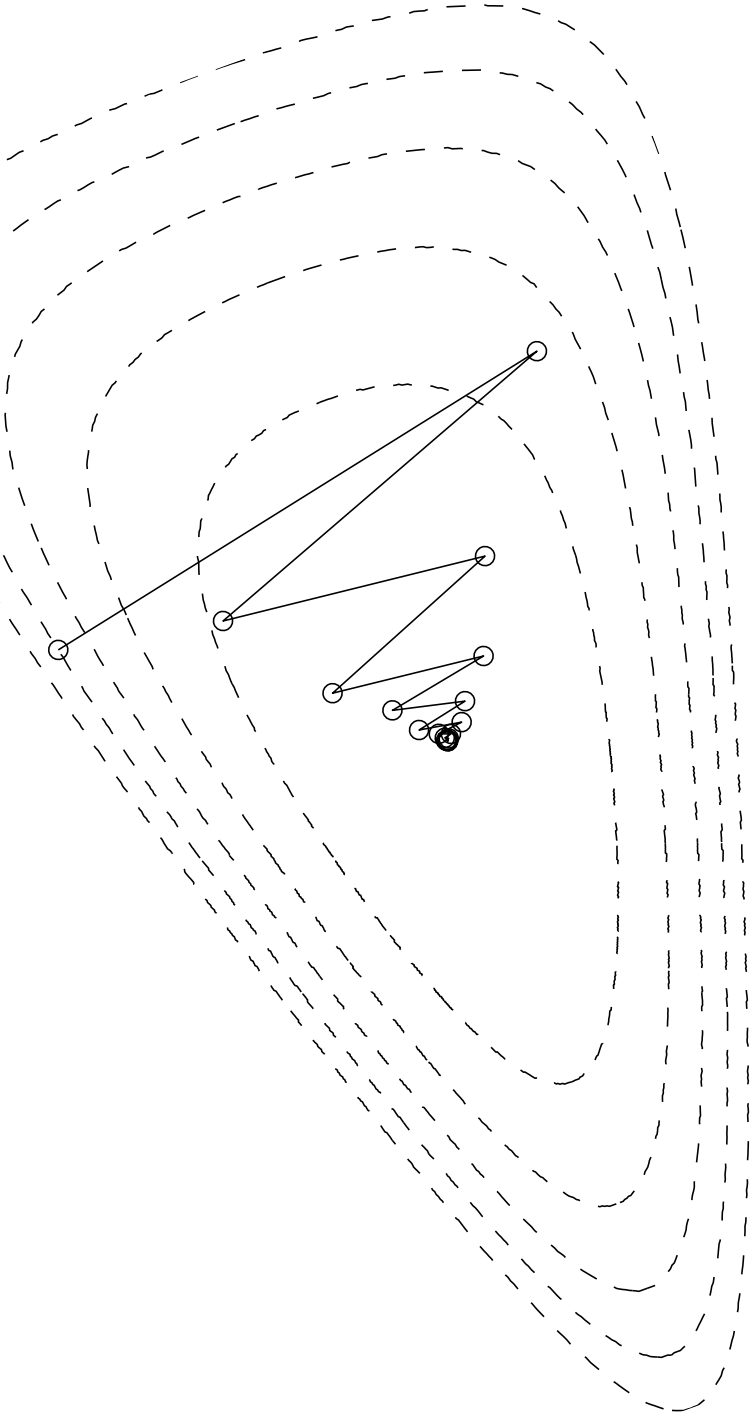


Introduction to Optimization

Convex Problems

Marc Toussaint
U Stuttgart

Planned Outline

- Gradient-based optimization (1st order methods)
  - plain grad., steepest descent, conjugate grad., Rprop, stochastic grad.
  - adaptive stepsize heuristics
- Constrained Optimization
  - squared penalties, augmented Lagrangian, log barrier
  - Lagrangian, KKT conditions, Lagrange dual, log barrier $\leftrightarrow$ approx. KKT
- 2nd order methods
  - Newton, Gauss-Newton, Quasi-Newton, (L)BFGS
  - constrained case, primal-dual Newton
- **Special convex cases**
  - Linear Programming, (sequential) Quadratic Programming
  - Simplex algorithm
  - relation to relaxed discrete optimization
- Black box optimization ("0th order methods")
  - blackbox stochastic search
  - Markov Chain Monte Carlo methods
  - evolutionary algorithms

Function types

- A function is defined **convex** iff

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

  for all $x, y \in \mathbb{R}^n$ and $a \in [0, 1]$.

- A function is **quasiconvex** iff

$$f(ax + (1-a)y) \leq \max\{f(x), f(y)\}$$

  for any $x, y \in \mathbb{R}^m$ and $a \in [0, 1]$.

  ..alternatively, iff every sublevel set $\{x | f(x) \leq \alpha\}$ is convex.

- [Subjective!] I call a function **unimodal** iff it has only 1 local minimum, which is the global minimum

  Note: in dimensions $n > 1$ quasiconvexity is stronger than unimodality

- A general **non-linear** function is unconstrained and can have multiple local minima

$$\text{convex} \subset \text{quasiconvex} \subset \text{unimodal} \subset \text{general}$$

Local optimization

- So far I avoided making explicit assumptions about problem convexity: To emphasize that all methods we considered – except for Newton – are applicable also on non-convex problems.

- The methods we considered are **local** optimization methods, which can be defined as
  – a method that adapts the solution locally
  – a method that is guaranteed to converge to a local minimum only

- Local methods are efficient
  – if the problem is (strictly) unimodal (strictly: no plateaux)
  – if time is critical and a local optimum is a sufficiently good solution
  – if the algorithm is restarted very often to hit multiple local optima

Convex problems

- Convexity is a strong assumption!

- Nevertheless, convex problems are important
  – theoretically (convergence proofs!)
  – for many real world applications

Convex problems

- A constrained optimization problem

$$\min_x \ f(x) \quad \text{s.t.} \quad g(x) \leq 0,\ h(x) = 0$$

  is called **convex** iff
  – f is convex
  – each g_i, $i = 1, .., m$ is convex
  – h is linear: $h(x) = Ax - b,\ A \in \mathbb{R}^{l \times n}, b \in \mathbb{R}^l$

- Alternative definition:

  f convex and feasible region is a convex set

Linear and Quadratic Programs

- Linear Program (LP)

$$\min_x \; c^\top x \quad \text{s.t.} \quad Gx \leq h, \; Ax = b$$

  LP in standard form

$$\min_x \; c^\top x \quad \text{s.t.} \quad x \geq 0, \; Ax = b$$

- Quadratic Program (QP)

$$\min_x \; \frac{1}{2} x^\top Q x + c^\top x \quad \text{s.t.} \quad Gx \leq h, \; Ax = b$$

  where Q is positive definite.

  (One also defines Quadratically Constraint Quadratic Programs (QCQP))

Transforming an LP problem into standard form

- LP problem:

$$\min_x \; c^\top x \quad \text{s.t.} \quad Gx \leq h, \; Ax = b$$

- Define slack variables:

$$\min_{x,\xi} \; c^\top x \quad \text{s.t.} \quad Gx + \xi = h, \; Ax = b, \; \xi \geq 0$$

- Express $x = x^+ - x^-$ with $x^+, x^- \geq 0$:

$$\min_{x^+,x^-,\xi} \; c^\top (x^+ - x^-)$$

$$\text{s.t.} \quad G(x^+ - x^-) + \xi = h, \; A(x^+ - x^-) = b, \; \xi \geq 0, \; x^+ \geq 0, \; x^- \geq 0$$

where $(x^+, x^-, \xi) \in \mathbb{R}^{2n+m}$

- Now this is conform with the standard form (replacing $(x^+, x^-, \xi) \equiv x$, etc)

$$\min_x \; c^\top x \quad \text{s.t.} \quad x \geq 0, \; Ax = b$$

Linear Programming

– Algorithms
– Application: LP relaxtion of discret problems

Algorithms for Linear Programming

- All of which we know!
  – augmented Lagrangian (LANCELOT software), penalty
  – log barrier (“interior point method”, “[central] path following”)
  – primal-dual Newton

- The simplex algorithm, walking on the constraints

  (The emphasis in the notion of *interior* point methods is to distinguish from constraint walking methods.)

- Interior point and simplex methods are comparably efficient
  Which is better depends on the problem

Simplex Algorithm

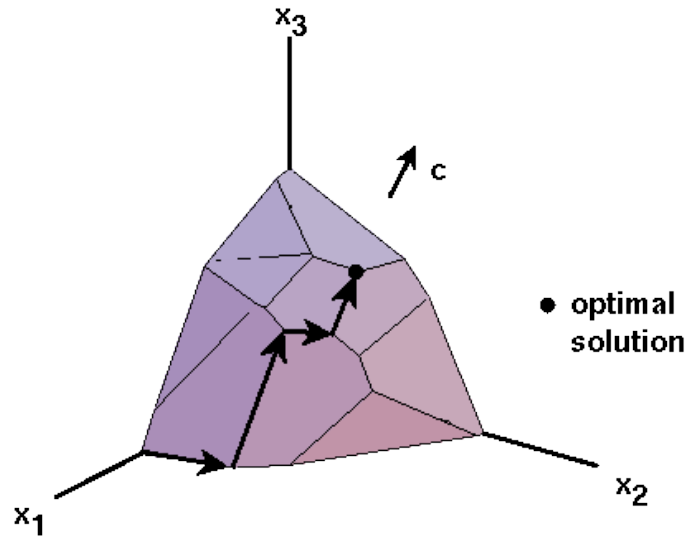
Georg Dantzig (1947)

Note: Not to confuse with the NelderMead method (downhill simplex method)

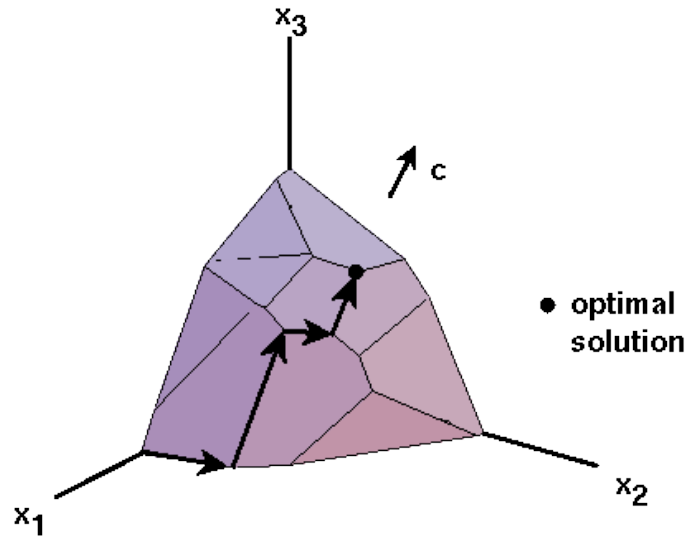
- We consider an LP in standard form

$$\min_x \; c^\top x \quad \text{s.t.} \quad x \geq 0, \; Ax = b$$

- Note that in a linear program the optimum is always situated at a corner



Simplex Algorithm



- The Simplex Algorithm walks along the edges of the polytope, at every corner choosing the edge that decreases $c^\top x$ most
- This either terminates at a corner, or leads to an unconstrained edge ($-\infty$ optimum)

- In practise this procedure is done by “pivoting on the simplex tableaux”

Simplex Algorithm

- The simplex algorithm is often efficient, but in worst case exponential in n and m.

- Interior point methods (log barrier) and, more recently again, augmented Lagrangian methods have become somewhat more popular than the simplex algorithm

LP-relaxations of discrete problems

Integer linear programming

- An integer linear program (for simplicity binary) is

$$\min_x c^\top x \quad \text{s.t.} \quad Ax = b,\ x_i \in \{0, 1\}$$

- Examples:
  - Traveling Salesman: $\min_{x_{ij}} \sum_{ij} c_{ij} x_{ij}$ with $x_{ij} \in \{0, 1\}$ and several more constraints (e.g. rows and columns of x sum to 1)
  - (max) SAT problem: In conjunctive normal form, each clause contributes an additional variable and a term in the objective function; each clause contributes a constraint
    Google: *The Power of Semidefinite Programming Relaxations for MAXSAT*

LP relaxations of integer linear programs

- Instead of solving

$$\min_x c^\top x \quad \text{s.t.} \quad Ax = b,\ x_i \in \{0, 1\}$$

  we solve

$$\min_x c^\top x \quad \text{s.t.} \quad Ax = b,\ x \in [0, 1]$$

- Clearly, the relaxed solution will be a *lower bound* on the integer solution (sometimes also called "outer bound" because $[0, 1] \supset \{0, 1\}$)

- Computing the relaxed solution is interesting
  – as an "approximation" or initialization to the integer problem
  – to be aware of the lower bound (what is achievable)
  – in cases where the optimal relaxed solution happens to be integer

Example: MAP inference in MRFs

- Given integer random variables x_i, $i = 1, .., n$, a pairwise Markov Random Field (MRF) is defined as

$$f(x) = \sum_{(ij) \in E} f_{ij}(x_i, x_j) + \sum_i f_i(x_i)$$

  where E denotes the set of edges.
  (Note: any general (non-pairwise) MRF can be converted into a pair-wise one, blowing up the number of variables)

- Reformulate with different variables

$$b_i(x) = [x_i = x] \ , \quad b_{ij}(x, y) = [x_i = x] \, [x_j = y]$$

  These are $nm + |E|m^2$ binary variables

- The indicator variables need to fulfil the constraints

$$b_i(x), b_{ij}(x, y) \in \{0, 1\}$$

$$\sum_x b_i(x) = 1 \qquad \text{because } x_i \text{ takes eactly one value}$$

$$\sum_y b_{ij}(x, y) = b_i(x) \qquad \text{consistency between indicators}$$

Example: MAP inference in MRFs

- Finding $\max_x f(x)$ of a MRF is then equivalent to

$$\max_{b_i(x), b_{ij}(x,y)} \sum_{(ij)\in E} \sum_{x,y} b_{ij}(x,y)\ f_{ij}(x,y) + \sum_i \sum_x b_i(x)\ f_i(x)$$

  such that

$$b_i(x), b_{ij}(x,y) \in \{0,1\}\ , \quad \sum_x b_i(x) = 1\ , \quad \sum_y b_{ij}(x,y) = b_i(x)$$

- The LP-relaxation replaces the constraint to be

$$b_i(x), b_{ij}(x,y) \in [0,1]\ , \quad \sum_x b_i(x) = 1\ , \quad \sum_y b_{ij}(x,y) = b_i(x)$$

  This set of feasible b's is called **marginal polytope** (because it describes the a space of "probability distributions" that are marginally consistent (but not necessarily globally normalized!))

Example: MAP inference in MRFs

- Solving the original MAP problem is NP-hard
  Solving the LP-relaxation is really efficient

- If the solution of the LP-relaxation turns out to be integer, we've solved the originally NP-hard problem!
  If not, the relaxed problem can be discretized to be a good initialization for discrete optimization

- For binary attractive MRFs (a common case) the solution will always be integer

Quadratic Programming

Quadratic Programming

$$\min_x \ \frac{1}{2} x^\top Q x + c^\top x \quad \text{s.t.} \quad Gx \leq h,\ Ax = b$$

(The dual of a QP is again a QP)

- Efficient Algorithms:
  – Interior point (log barrier)
  – Augmented Lagrangian
  – Penalty

- Highly relevant applications:
  – Support Vector Machines
  – Similar types of max-margin modelling methods

Sequential Quadratic Programming

- We considered general non-linear problems

$$\min_x \; f(x) \quad \text{s.t.} \quad g(x) \leq 0$$

  where we can evaluate $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ and $g(x)$, $\nabla g(x)$, $\nabla^2 g(x)$ for any $x \in \mathbb{R}^n$

$$\rightarrow \text{Newton method}$$

- The standard step direction Δ is $(\nabla^2 f(x) + \lambda \mathbf{I})\, \Delta = -\nabla f(x)$

- Sometimes a better step direction Δ can be found by solving the local QP-approximation to the problem

$$\min_\Delta \; f(x) + \nabla f(x)^\top \Delta + \Delta^\top \nabla^2 f(x) \Delta \quad \text{s.t.} \quad g(x) + \nabla g(x)^\top \Delta \leq 0$$

  The latter is only an optimization problem over Δ and only requires the evaluation of $f(x), \nabla f(x), \nabla^2 f(x), g(x), \nabla g(x)$ once.

Backtracking line search for convex functions

- Line search in general denotes the problem

$$\min_{\alpha \geq 0} f(x + \alpha\Delta)$$

  for some step direction Δ

- The most common line search on convex functions is **backtracking**

Input: start point x, direction Δ, function $f(x)$, parameters $a \in (0, \frac{1}{2})$, $b \in (0, 1)$

Output: x

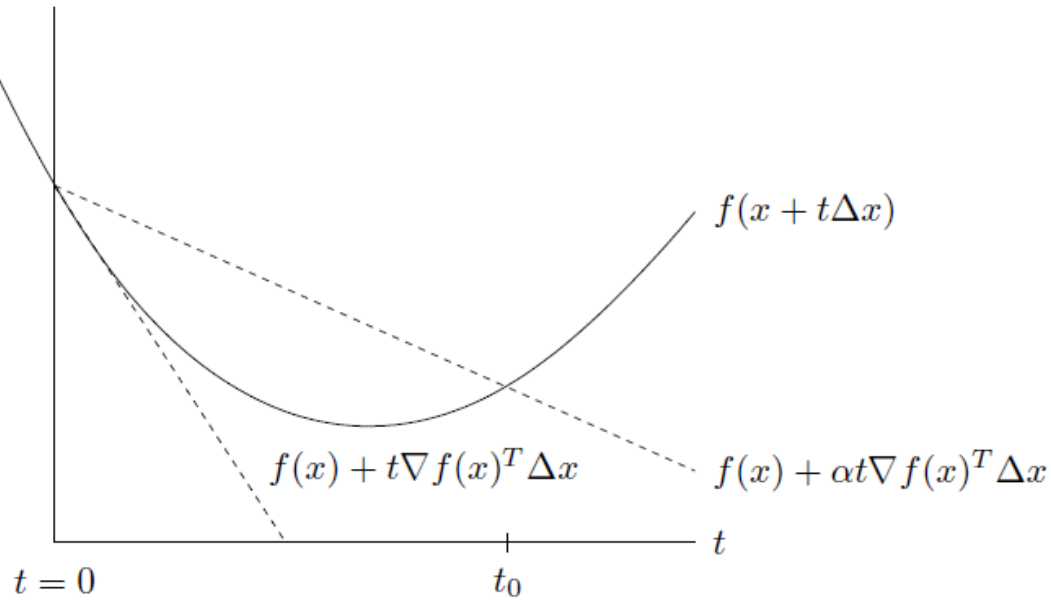
1: initialize $\alpha = 1$
2: **while** $f(x + \alpha\Delta) > f(x) + a\nabla f(x)^\top(\alpha\Delta)$ **do**
3: $\quad t \leftarrow bt$
4: **end while**

---

b describes the stepsize decrement in case of a rejected step
a describes a minimum desired decrease in $f(x)$

- In the 2nd order methods we described, we chose $a = 0$:
  We did not invest into further line search steps if $f(x + \alpha\Delta) \leq f(x)$
- Boyd at al: typically $a \in [0.01, 0.3]$ and $b \in [0.1, 0.8]$

Backtracking line search for convex functions



(From Boyd et al.; notation differs from previous slide.)