LUND UNIVERSITY

L-3: Optimization methods

Ted Kronvall

Admin

Moodle:

- Register for ”Machine Learning 2019” in Moodle by the 8th of April.
- We will thereafter close enrollment in order to verify the student list for grading of assignments.
- Use enrollment key ”clustering2019”.
- Sign up for the Q & A sessions for Assignment 1.

Admin

Assignment 1:

- Assignment 1 is handed out today!
- Submit your report and solutions for Assignment 1 on Sunday April 14th at the latest.
- The report should be one self-contained pdf document.
- Your code and results should be submitted as supplementary material in a single file.
- No late submissions will be accepted.
- A total of 100 p are awarded, divided over 7 tasks. 70 p are required to pass the assignment.
- If you do not pass, but get 30 p or more, you may do a resubmission, which is due on June 10th. You may only do so twice throughout the course.

Small recap from L-2

- Maximum likelihood estimation (MLE) is the best approach to estimate the model variables, given that the statistical model is known and accurate.
- With linear regression, complicated data structures may be modeled by using non-linear basis functions.
- The mean squared error is the typical choice for measuring how well the model fit the data. In fact, the MSE is the MLE for Gaussian additive noise.
- Ridge regression uses the 2-norm of the model variables as regularization term, preventing overfitting to data.
- LASSO promotes sparsity in the model variables, by soft-thresholding their magnitudes towards zero.
- The bias-variance decomposition shows the trade-off between statistical bias and variance when choosing a model.
- Bayesian regression offers a framework for analyzing the uncertainty in the model variables.

LUND UNIVERSITY

Today

- ~~Basic concepts; tasks, training, validation, regularization~~.
- ~~Probability theory and~~ optimization
- ~~Linear regression~~
- Methods for classification

- Clustering methods
- Neural networks

- Convolutional and recurrent NNs

Generative modeling

- Reinforcement learning
- Natural language processing
- Deep learning for computer vision

Mathematical optimization

- These slides are adapted lecture notes by Stephen Boyd (Stanford) and Pontus Giselsson (Automatic Control LTH)

LUND UNIVERSITY

The optimization problem

Mathematical optimization

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

optimal solution $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

LUND UNIVERSITY

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

LUND UNIVERSITY

Optimization

Optimization can be divided into two classes

- Convex optimization
- Nonconvex optimization

Why this distinction?

- In convex optimization: all local optima are globally optimal
- This is not true for nonconvex optimization

Implications for algorithms:

- Convex: Can perform local search and guarantee global optimality
- This is not the case in nonconvex optimization

Convex vs nonconvex modeling

- Modeling capabilities are richer in the nonconvex setting
- However, cannot guarantee convergence to global optimum

If convex model is accurate enough, use it! If not, use nonconvex.

LUND UNIVERSITY

Optimization in this course

Convex:

- Least squares
- Lasso
- Support vector machines
- Logistic regression

Nonconvex:

- Neural network training (recurrent/convolutional/deep)

LUND UNIVERSITY

Algorithm types and problem dimensions

Problem dimension	Algorithm type
small to medium scale (up to 1'000 variables)	Second-order methods (Newton's method, interior point)
large-scale (up to 100'000 variables)	First-order methods
huge-scale (more than 100'000 variables)	Stochastic, coordinate, parallel asynchronous first-order methods

In machine learning, problems usually large to huge scale
$\Rightarrow$ We will focus on algorithms for such problems

Convex optimization

Convex sets

Convex set

line segment between x_1 and x_2: all points

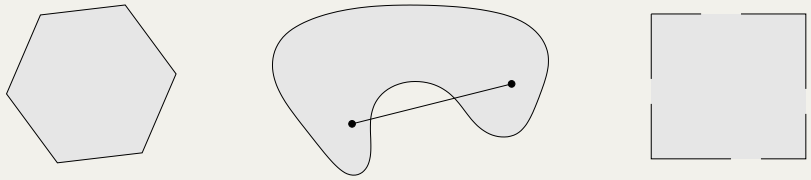
$$x = \theta x_1 + (1 - \theta)x_2$$

with $0 \leq \theta \leq 1$

convex set: contains line segment between any two points in the set

$$x_1, x_2 \in C, \quad 0 \leq \theta \leq 1 \quad \Longrightarrow \quad \theta x_1 + (1 - \theta)x_2 \in C$$

examples (one convex, two nonconvex sets)

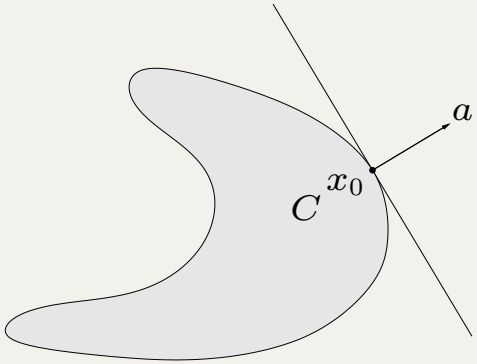


Supporting hyperplane theorem

supporting hyperplane to set C at boundary point x_0:

$$\{x \mid a^T x = a^T x_0\}$$

where $a \neq 0$ and $a^T x \leq a^T x_0$ for all $x \in C$



supporting hyperplane theorem: if C is convex, then there exists a supporting hyperplane at every boundary point of C

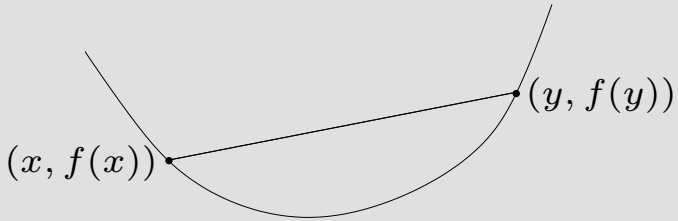
Convex functions

Definition

$f : \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

for all $x, y \in \mathbf{dom}\, f$, $0 \leq \theta \leq 1$



- f is concave if $-f$ is convex
- f is strictly convex if $\mathbf{dom}\, f$ is convex and

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

for $x, y \in \mathbf{dom}\, f$, $x \neq y$, $0 < \theta < 1$

~~Convex~~ functions

Concave

for $x, y \in \mathbf{dom}\, f$, $x \neq y$, $0 < \theta < 1$

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

- f is strictly convex if $\mathbf{dom}\, f$ is convex and
- f is concave if $-f$ is convex

$$f(\theta x + (1-\theta)y) \geq \theta f(x) + (1-\theta)f(y)$$

for all $x, y \in \mathbf{dom}\, f$, $0 \leq \theta \leq 1$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

Definition

$f : \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and

Convex functions

Examples on R

convex:

- affine: $ax + b$ on $\mathbf{R}$, for any $a, b \in \mathbf{R}$
- exponential: e^{ax}, for any $a \in \mathbf{R}$
- powers: x^α on $\mathbf{R}_{++}$, for $\alpha \geq 1$ or $\alpha \leq 0$
- powers of absolute value: $|x|^p$ on $\mathbf{R}$, for $p \geq 1$
- negative entropy: $x \log x$ on $\mathbf{R}_{++}$

concave:

- affine: $ax + b$ on $\mathbf{R}$, for any $a, b \in \mathbf{R}$
- powers: x^α on $\mathbf{R}_{++}$, for $0 \leq \alpha \leq 1$
- logarithm: $\log x$ on $\mathbf{R}_{++}$

Convex functions

Examples on $\mathbf{R}^n$ and $\mathbf{R}^{m\times n}$

affine functions are convex and concave; all norms are convex

examples on $\mathbf{R}^n$

- affine function $f(x) = a^T x + b$
- norms: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \geq 1$; $\|x\|_\infty = \max_k |x_k|$

examples on $\mathbf{R}^{m\times n}$ ($m \times n$ matrices)

- affine function

$$f(X) = \mathbf{tr}(A^T X) + b = \sum_{i=1}^m \sum_{j=1}^n A_{ij} X_{ij} + b$$

- spectral (maximum singular value) norm

$$f(X) = \|X\|_2 = \sigma_{\max}(X) = (\lambda_{\max}(X^T X))^{1/2}$$

Convex functions

First-order condition

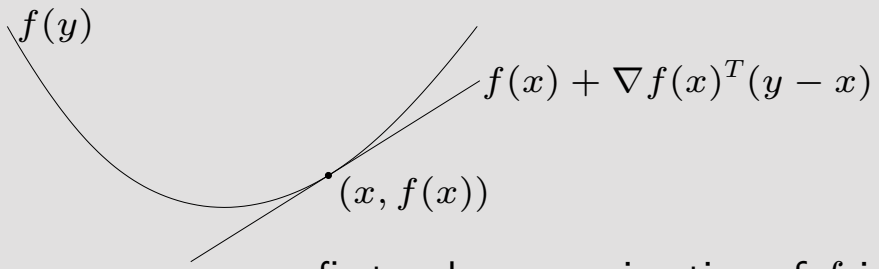
f is **differentiable** if $\mathbf{dom}\, f$ is open and the gradient

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

exists at each $x \in \mathbf{dom}\, f$

1st-order condition: differentiable f with convex domain is convex iff

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for all } x, y \in \mathbf{dom}\, f$$



first-order approximation of f is global underestimator

Convex functions

Second-order conditions

f is **twice differentiable** if $\mathbf{dom}\, f$ is open and the Hessian $\nabla^2 f(x) \in \mathbf{S}^n$,

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \ldots, n,$$

exists at each $x \in \mathbf{dom}\, f$

2nd-order conditions: for twice differentiable f with convex domain

- f is convex if and only if

$$\nabla^2 f(x) \succeq 0 \quad \text{for all } x \in \mathbf{dom}\, f$$

- if $\nabla^2 f(x) \succ 0$ for all $x \in \mathbf{dom}\, f$, then f is strictly convex

Convex functions

Examples

quadratic function: $f(x) = (1/2)x^T Px + q^T x + r$ (with $P \in \mathbf{S}^n$)

$$\nabla f(x) = Px + q, \qquad \nabla^2 f(x) = P$$

convex if $P \succeq 0$

least-squares objective: $f(x) = \|Ax - b\|_2^2$

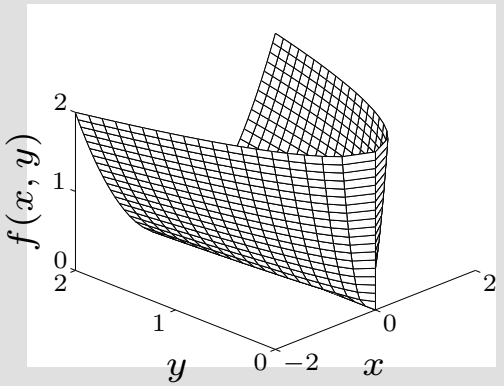
$$\nabla f(x) = 2A^T(Ax - b), \qquad \nabla^2 f(x) = 2A^T A$$

convex (for any A)

quadratic-over-linear: $f(x, y) = x^2/y$

$$\nabla^2 f(x,y) = \frac{2}{y^3}\begin{bmatrix} y \\ -x \end{bmatrix}\begin{bmatrix} y \\ -x \end{bmatrix}^T \succeq 0$$

convex for $y > 0$



Convex functions

Epigraph and sublevel set

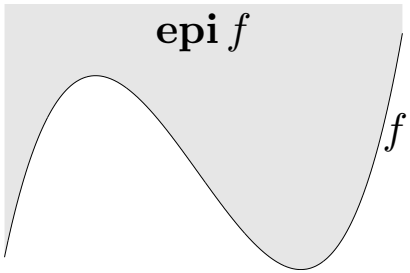
α-sublevel set of $f : \mathbf{R}^n \to \mathbf{R}$:

$$C_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}$$

sublevel sets of convex functions are convex (converse is false)

epigraph of $f : \mathbf{R}^n \to \mathbf{R}$:

$$\mathbf{epi}\, f = \{(x,t) \in \mathbf{R}^{n+1} \mid x \in \mathbf{dom}\, f,\ f(x) \leq t\}$$



f is convex if and only if $\mathbf{epi}\, f$ is a convex set

Convex functions

Operations that preserve convexity

practical methods for establishing convexity of a function

1. verify definition (often simplified by restricting to a line)
2. for twice differentiable functions, show $\nabla^2 f(x) \succeq 0$
3. show that f is obtained from simple convex functions by operations that preserve convexity
   - nonnegative weighted sum
   - composition with affine function
   - pointwise maximum and supremum
   - composition
   - minimization
   - perspective

Optimization problem in standard form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

- $x \in \mathbf{R}^n$ is the optimization variable
- $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the objective or cost function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$, are the inequality constraint functions
- $h_i : \mathbf{R}^n \to \mathbf{R}$ are the equality constraint functions

optimal value:

$$p^\star = \inf\{f_0(x) \mid f_i(x) \leq 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ i = 1, \ldots, p\}$$

- $p^\star = \infty$ if problem is infeasible (no x satisfies the constraints)
- $p^\star = -\infty$ if problem is unbounded below

Optimal and locally optimal points

x is **feasible** if $x \in \mathbf{dom}\, f_0$ and it satisfies the constraints

a feasible x is **optimal** if $f_0(x) = p^\star$; $X_{\rm opt}$ is the set of optimal points

x is **locally optimal** if there is an $R > 0$ such that x is optimal for

$$\begin{array}{ll} \text{minimize (over } z) & f_0(z) \\ \text{subject to} & f_i(z) \leq 0, \quad i = 1, \ldots, m, \quad h_i(z) = 0, \quad i = 1, \ldots, p \\ & \|z - x\|_2 \leq R \end{array}$$

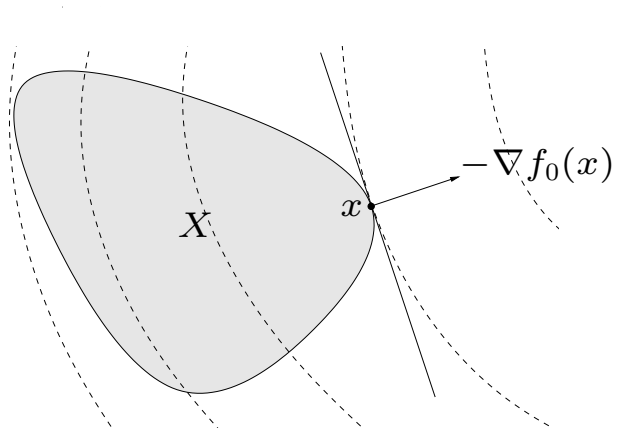
examples (with $n = 1$, $m = p = 0$)

- $f_0(x) = 1/x$, $\mathbf{dom}\, f_0 = \mathbf{R}_{++}$: $p^\star = 0$, no optimal point
- $f_0(x) = -\log x$, $\mathbf{dom}\, f_0 = \mathbf{R}_{++}$: $p^\star = -\infty$
- $f_0(x) = x \log x$, $\mathbf{dom}\, f_0 = \mathbf{R}_{++}$: $p^\star = -1/e$, $x = 1/e$ is optimal
- $f_0(x) = x^3 - 3x$, $p^\star = -\infty$, local optimum at $x = 1$

Optimality criterion for differentiable f_0

x is optimal if and only if it is feasible and

$$\nabla f_0(x)^T(y-x) \geq 0 \quad \text{for all feasible } y$$

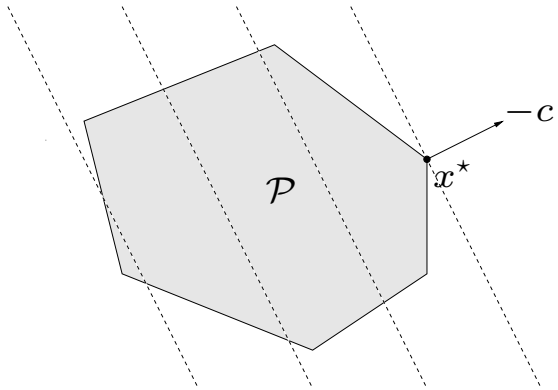


if nonzero, $\nabla f_0(x)$ defines a supporting hyperplane to feasible set X at x

Linear program (LP)

$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

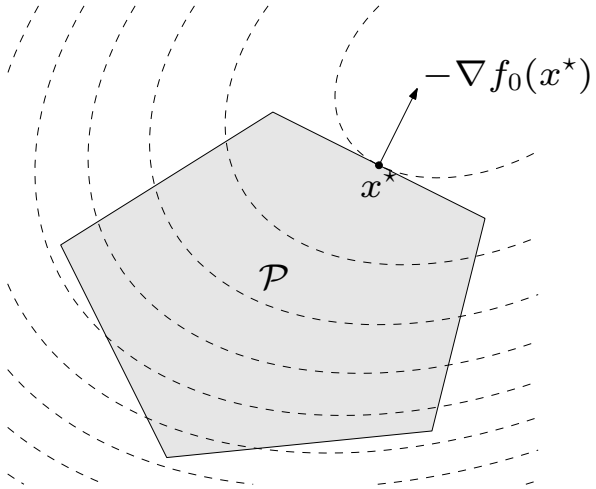
- convex problem with affine objective and constraint functions
- feasible set is a polyhedron



Quadratic program (QP)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

- $P \in \mathbf{S}_+^n$, so objective is convex quadratic
- minimize a convex quadratic function over a polyhedron



Unconstrained optimization

Unconstrained minimization

$$\text{minimize} \quad f(x)$$

- f convex, twice continuously differentiable (hence $\mathbf{dom}\, f$ open)
- we assume optimal value $p^\star = \inf_x f(x)$ is attained (and finite)

unconstrained minimization methods

- produce sequence of points $x^{(k)} \in \mathbf{dom}\, f$, $k = 0, 1, \ldots$ with

$$f(x^{(k)}) \to p^\star$$

- can be interpreted as iterative methods for solving optimality condition

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

Descent methods

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)} \quad \text{with } f(x^{(k+1)}) < f(x^{(k)})$$

- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$
  ($i.e.$, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
repeat

1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Line search

Line search types

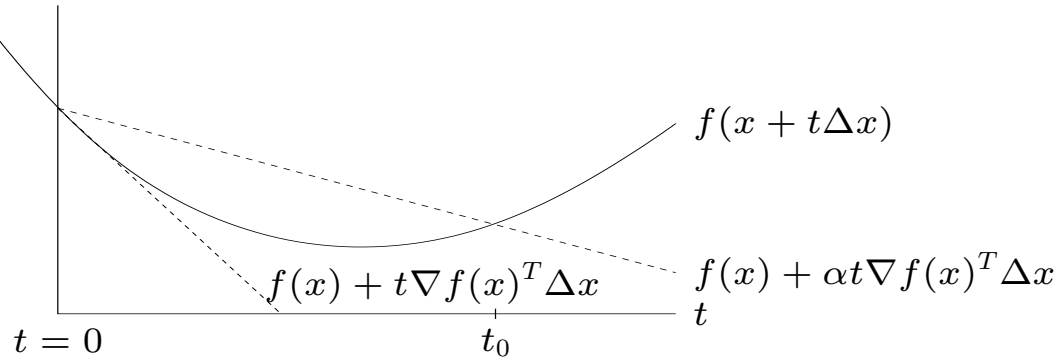
exact line search: $t = \operatorname{argmin}_{t>0} f(x + t\Delta x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$



LUND UNIVERSITY

Derivative-free methods

LUND UNIVERSITY

Coordinate descent

- In coordinate descent, the objective is iteratively minimized per coordinate in the variable vector.
- The algorithm is typically called cyclic coordinate descent (CCD), because the coordinates are updated cyclically.
- Starting at:

$$\mathbf{x}^{(0)} = x_1^{(0)}, \ldots, x_N^{(0)}$$

- The updates are:

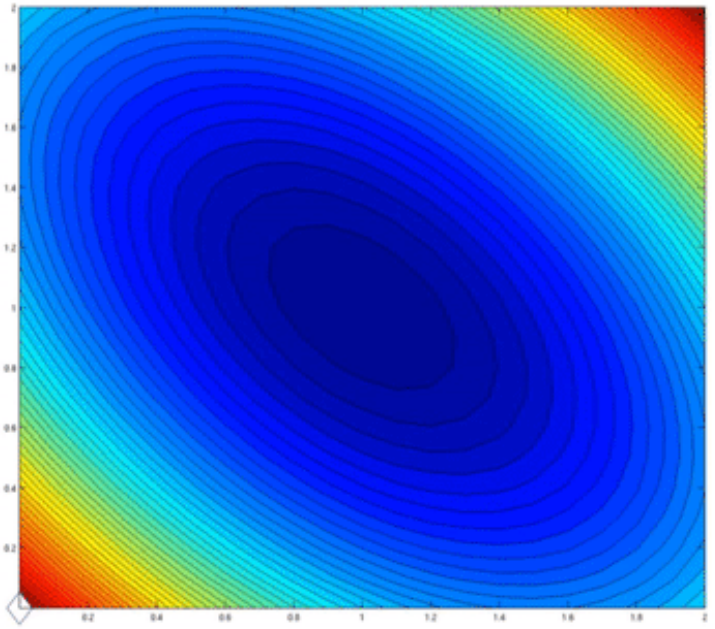
$$x_i^{(k+1)} = \underset{y \in \mathrm{dom} f}{\mathrm{argmin}} \; f(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, y, x_{i+1}^{(k)}, \ldots, x_N^{(k)})$$

- Instead of cycling through the coordinates in the indexed order, one typically randomizes the update order at each cycle.

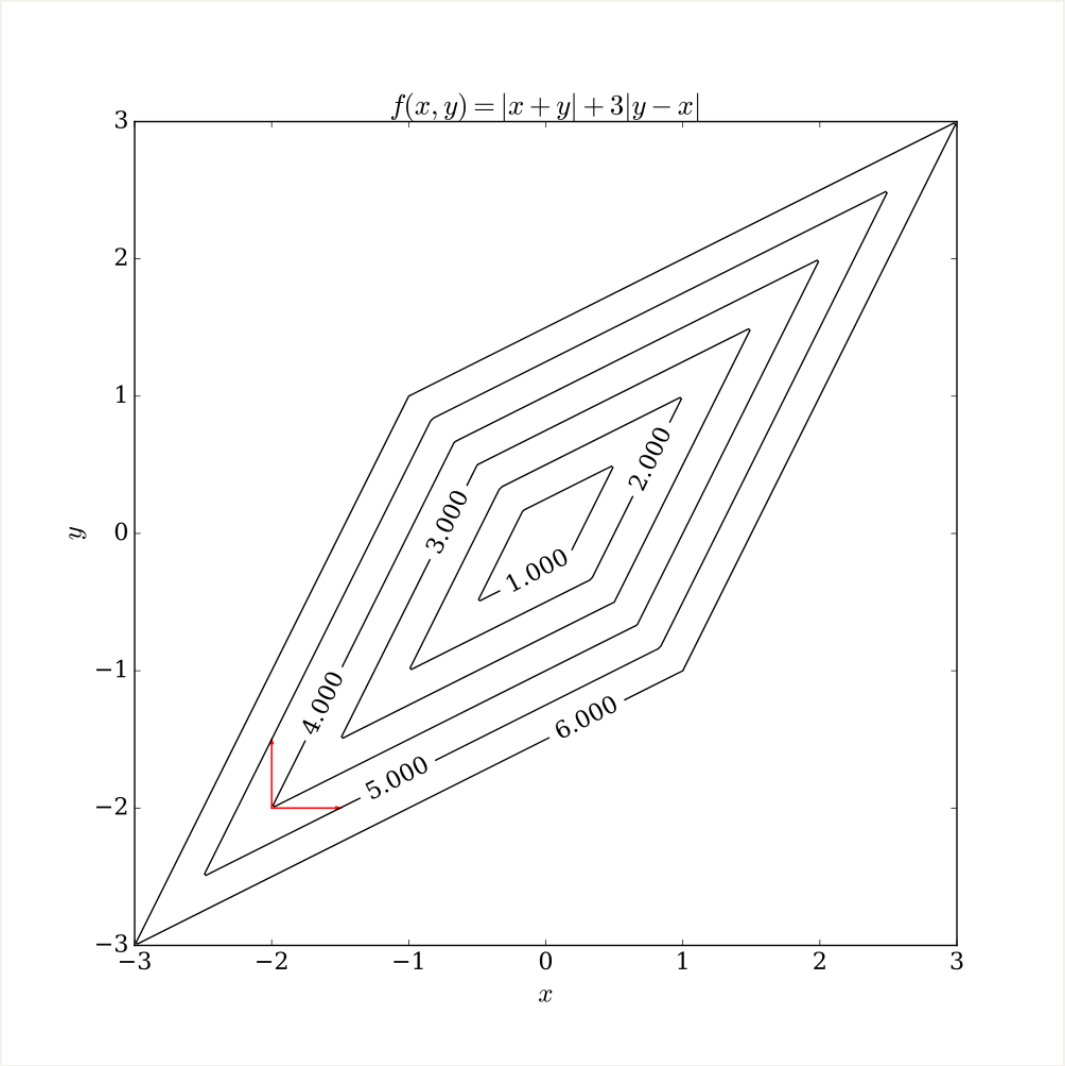
LUND UNIVERSITY

Coordinate descent

$$f(x) = \tfrac{1}{2} x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



Coordinate descent



- For non-smooth objectives, CCD might fail to converge.

First-order methods

Gradient descent method

general descent method with $\Delta x = -\nabla f(x)$

---

given a starting point $x \in \mathbf{dom}\, f$.
repeat

1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

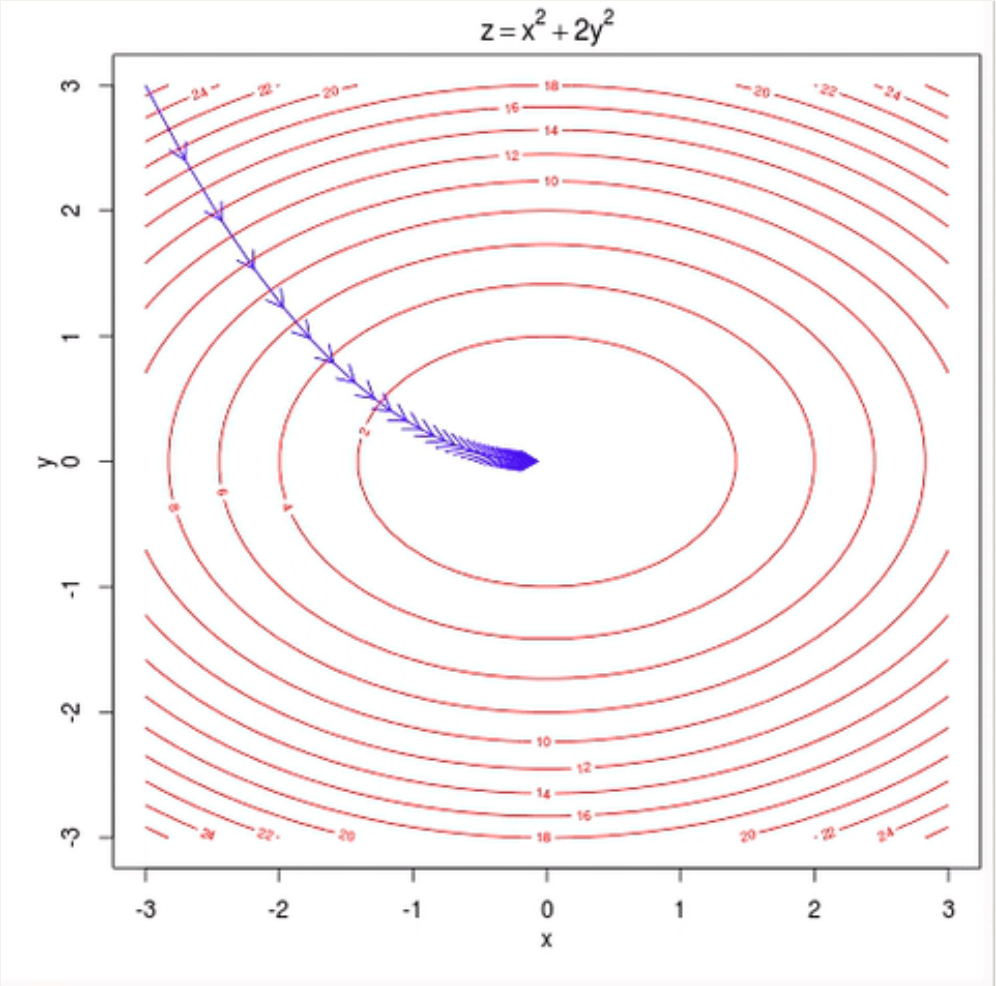
until stopping criterion is satisfied.

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \leq \epsilon$
- convergence result: for strongly convex f,

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

$c \in (0, 1)$ depends on m, $x^{(0)}$, line search type

Gradient descent



Gradient descent

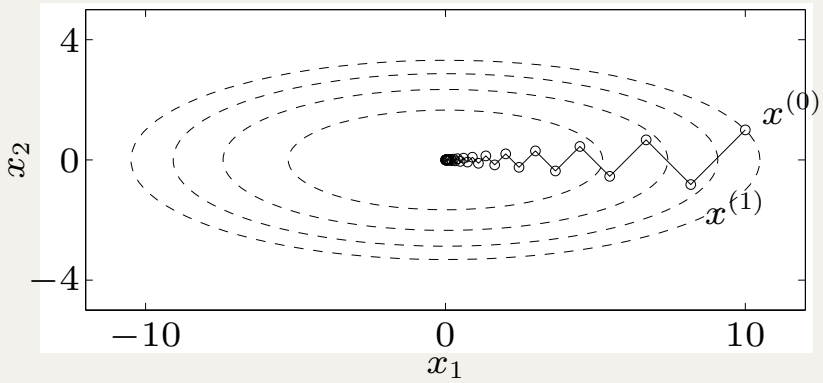
quadratic problem in $\mathbf{R}^2$

$$f(x) = (1/2)(x_1^2 + \gamma x_2^2) \qquad (\gamma > 0)$$

with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1} \right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1} \right)^k$$

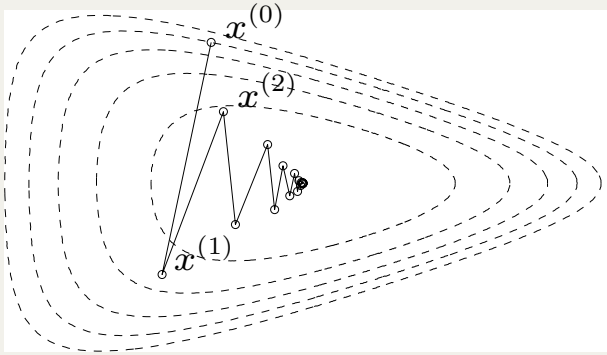
- very slow if $\gamma \gg 1$ or $\gamma \ll 1$
- example for $\gamma = 10$:



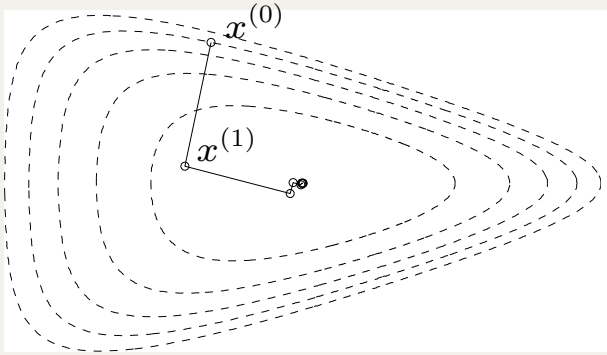
Gradient descent

nonquadratic example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



backtracking line search



exact line search

Steepest descent

Steepest descent method

normalized steepest descent direction (at x, for norm $\|\cdot\|$):

$$\Delta x_{\mathrm{nsd}} = \mathrm{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

interpretation: for small v, $f(x+v) \approx f(x) + \nabla f(x)^T v$;
direction Δx_{nsd} is unit-norm step with most negative directional derivative

(unnormalized) steepest descent direction

$$\Delta x_{\mathrm{sd}} = \|\nabla f(x)\|_* \Delta x_{\mathrm{nsd}}$$

satisfies $\nabla f(x)^T \Delta x_{\mathrm{sd}} = -\|\nabla f(x)\|_*^2$

steepest descent method

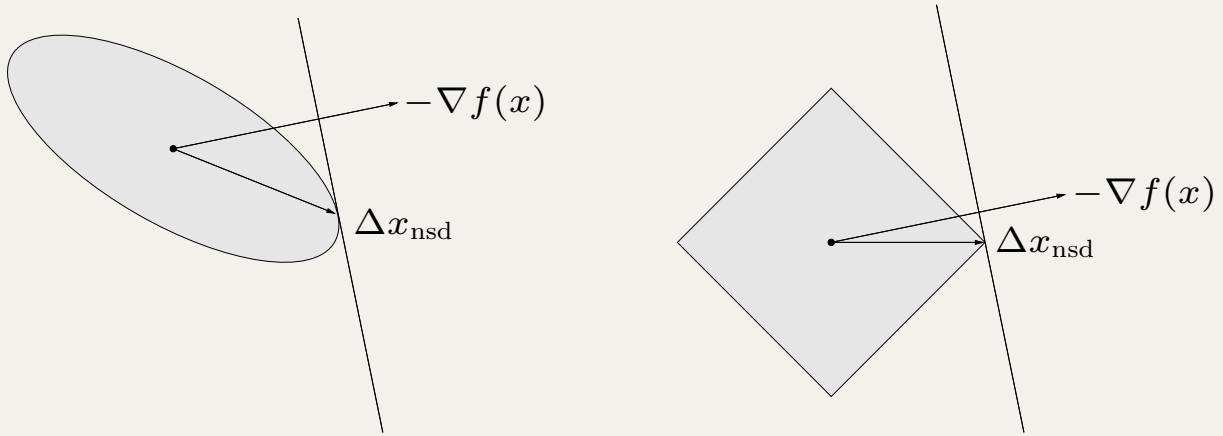
- general descent method with $\Delta x = \Delta x_{\mathrm{sd}}$
- convergence properties similar to gradient descent

Steepest descent

examples

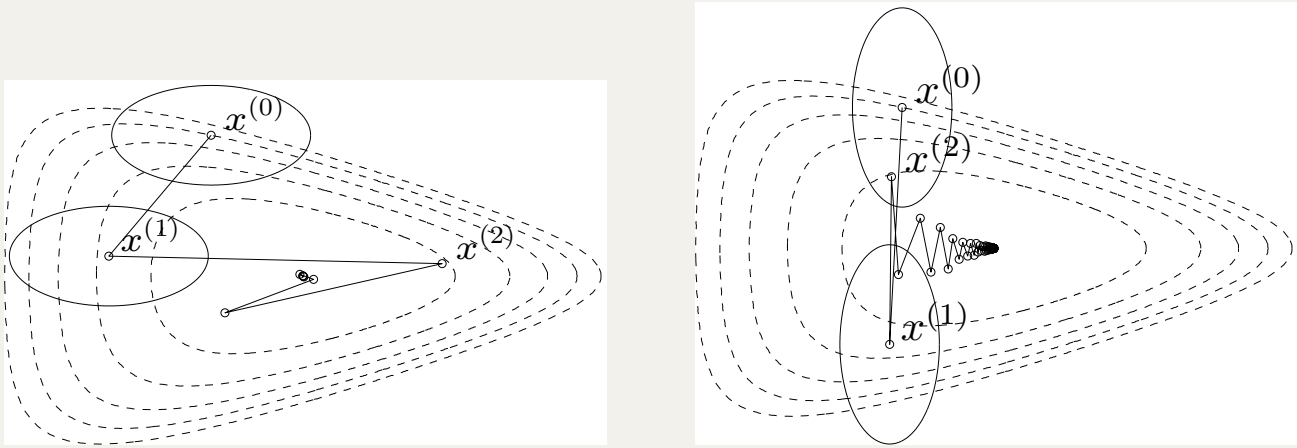
- Euclidean norm: $\Delta x_{\mathrm{sd}} = -\nabla f(x)$
- quadratic norm $\|x\|_P = (x^T P x)^{1/2}$ $(P \in \mathbf{S}^n_{++})$: $\Delta x_{\mathrm{sd}} = -P^{-1}\nabla f(x)$
- ℓ_1-norm: $\Delta x_{\mathrm{sd}} = -(\partial f(x)/\partial x_i)e_i$, where $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$

unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1-norm:



Steepest descent

choice of norm for steepest descent



- steepest descent with backtracking line search for two quadratic norms
- ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$
- equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_P$: gradient descent after change of variables $\bar{x} = P^{1/2}x$

shows choice of P has strong effect on speed of convergence

Second-order methods

Newton's method

Newton step

$$\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

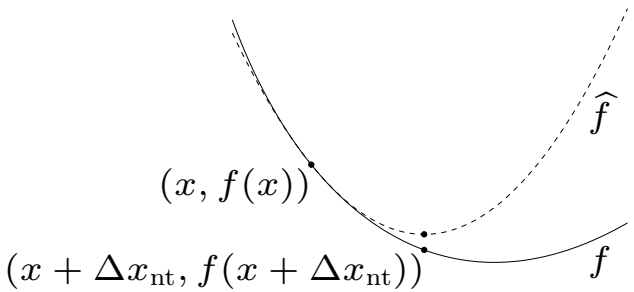
interpretations

- $x + \Delta x_{\mathrm{nt}}$ minimizes second order approximation

$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

- $x + \Delta x_{\mathrm{nt}}$ solves linearized optimality condition

$$\nabla f(x+v) \approx \nabla \widehat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$

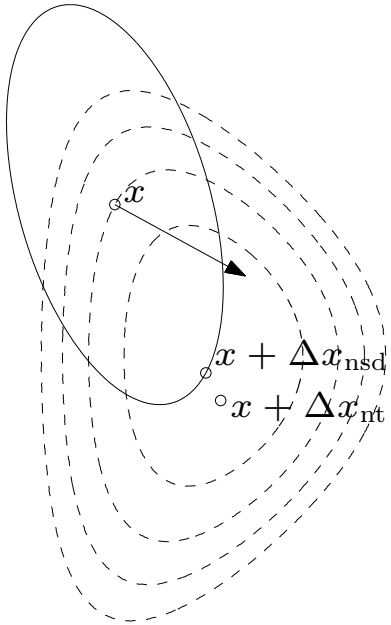


Unconstrained minimization

Newton's method

- Δx_{nt} is steepest descent direction at x in local Hessian norm

$$\|u\|_{\nabla^2 f(x)} = \left(u^T \nabla^2 f(x) u\right)^{1/2}$$



dashed lines are contour lines of f; ellipse is $\{x + v \mid v^T \nabla^2 f(x) v = 1\}$

arrow shows $-\nabla f(x)$

Newton's method

Newton's method

---

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat

1. *Compute the Newton step and decrement.*
   $\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x)$; $\quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$.
2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.

Constrained optimization

Constrained optimization

Lagrangian

standard form problem (not necessarily convex)

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

variable $x \in \mathbf{R}^n$, domain $\mathcal{D}$, optimal value $p^\star$

Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$, with $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$,

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \leq 0$
- ν_i is Lagrange multiplier associated with $h_i(x) = 0$

Constrained optimization

Lagrange dual function

Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$,

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \\ &= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \end{aligned}$$

g is concave, can be $-\infty$ for some λ, ν

lower bound property: if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^\star$

proof: if $\tilde{x}$ is feasible and $\lambda \succeq 0$, then

$$f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu)$$

minimizing over all feasible $\tilde{x}$ gives $p^\star \geq g(\lambda, \nu)$

Constrained optimization

Least-norm solution of linear equations

$$\begin{array}{ll} \text{minimize} & x^T x \\ \text{subject to} & Ax = b \end{array}$$

dual function

- Lagrangian is $L(x, \nu) = x^T x + \nu^T (Ax - b)$
- to minimize L over x, set gradient equal to zero:

$$\nabla_x L(x, \nu) = 2x + A^T \nu = 0 \quad \Longrightarrow \quad x = -(1/2) A^T \nu$$

- plug in in L to obtain g:

$$g(\nu) = L((-1/2) A^T \nu, \nu) = -\frac{1}{4} \nu^T A A^T \nu - b^T \nu$$

  a concave function of ν

lower bound property: $p^\star \geq -(1/4) \nu^T A A^T \nu - b^T \nu$ for all ν

LUND UNIVERSITY

Constrained optimization

The dual problem

Lagrange dual problem

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

- finds best lower bound on $p^\star$, obtained from Lagrange dual function
- a convex optimization problem; optimal value denoted $d^\star$
- λ, ν are dual feasible if $\lambda \succeq 0$, $(\lambda, \nu) \in \mathbf{dom}\, g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \mathbf{dom}\, g$ explicit

Weak and strong duality

weak duality: $d^\star \leq p^\star$

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems

strong duality: $d^\star = p^\star$

- does not hold in general
- (usually) holds for convex problems
- conditions that guarantee strong duality in convex problems are called **constraint qualifications**

Constrained optimization

Complementary slackness

assume strong duality holds, $x^\star$ is primal optimal, $(\lambda^\star, \nu^\star)$ is dual optimal

$$\begin{aligned} f_0(x^\star) = g(\lambda^\star, \nu^\star) &= \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x) + \sum_{i=1}^p \nu_i^\star h_i(x) \right) \\ &\leq f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star f_i(x^\star) + \sum_{i=1}^p \nu_i^\star h_i(x^\star) \\ &\leq f_0(x^\star) \end{aligned}$$

hence, the two inequalities hold with equality

- $x^\star$ minimizes $L(x, \lambda^\star, \nu^\star)$
- $\lambda_i^\star f_i(x^\star) = 0$ for $i = 1, \ldots, m$ (known as complementary slackness):

$$\lambda_i^\star > 0 \Longrightarrow f_i(x^\star) = 0, \qquad f_i(x^\star) < 0 \Longrightarrow \lambda_i^\star = 0$$

Constrained optimization

Karush-Kuhn-Tucker (KKT) conditions

the following four conditions are called KKT conditions (for a problem with differentiable f_i, h_i):

1. primal constraints: $f_i(x) \leq 0$, $i = 1, \ldots, m$, $h_i(x) = 0$, $i = 1, \ldots, p$
2. dual constraints: $\lambda \succeq 0$
3. complementary slackness: $\lambda_i f_i(x) = 0$, $i = 1, \ldots, m$
4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

from page 5–17: if strong duality holds and x, λ, ν are optimal, then they must satisfy the KKT conditions

Convergence

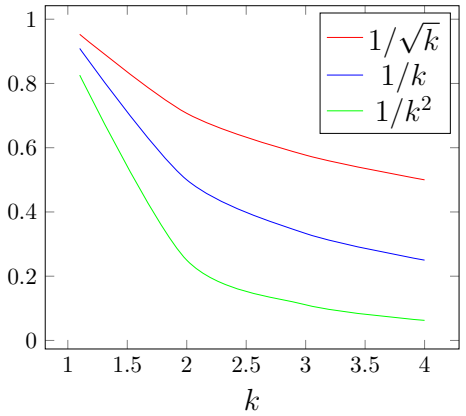
Let $\{\omega_k\} \subset \mathbb{R}^n$ be a sequence converging to ω_*. There are four major types of rates of convergence of interest to us

- SUBLINEAR $\lim_{k\to\infty} \frac{\|\omega_{k+1}-\omega_*\|}{\|\omega_k-\omega_*\|} = 1$

  When $n = 1$ and $\omega_* = 0$, at least three examples will be seen in this course:

$$\frac{1}{\sqrt{k}}, \quad \frac{1}{k}, \quad \frac{1}{k^2}$$

  Sublinear is a slow rate but $\frac{1}{k^2}$ is much faster than $\frac{1}{\sqrt{k}}$.



Convergence

- LINEAR (also known as geometric or exponential convergence):
  $\exists r \in (0,1) \quad \|\omega_{k+1} - \omega_*\| \leq r\|\omega_k - \omega_*\| \quad \forall k$

  When $n = 1$ and $\omega_* = 0$, an example is $(\frac{1}{2})^k$.

  First-order methods (like the gradient or steepest descent method) exhibit sublinear or linear rates. Second-order methods achieve a superlinear rate (quasi-Newton) or a quadratic rate (Newton).

- SUPERLINEAR $\exists\{\eta_k\}_{\eta_k \to 0} \quad \|\omega_{k+1} - \omega_*\| \leq \eta_k\|\omega_k - \omega_*\| \quad \forall k$

  The example for $n = 1$ and $\omega_* = 0$ is $\frac{1}{k!}$

- QUADRATIC $\exists M > 0, \|\omega_{k+1} - \omega_*\| \leq M\|\omega_k - \omega_*\|^2 \quad \forall k$

  Take $10^{(1/2)^k}$ as the example when $n = 1$ and $\omega_* = 0$.

Self study:

- **Goodfellow et al 2016 (Deep Learning):**
  - Chapter 4: 4.3 - 4.5
- **Bishop 2006 (Pattern Recognition and Machine Learning)**
  - Appendix: E
  - Chapter 5: 5.2
- **Hastie et al. 2019 (Elements of Statistical Learning)**
  - -

LUND UNIVERSITY

L-3: Optimization methods

Ted Kronvall