Lecture 9

- **Introduction to convex optimization**
  - Portfolio optimization revisited
  - Duality and distributed optimization

The first 11 slides are from https://www.stanford.edu/ boyd/cvxbook

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ ($A \in \mathbf{R}^{k \times n}$); less if structured
- a mature technology

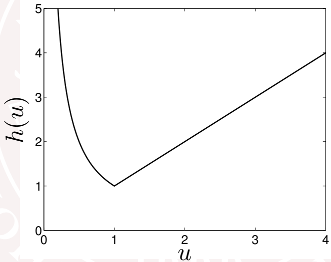
using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility (*e.g.*, including weights, adding regularization terms)

5. use convex optimization: problem is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} h(I_k/I_{\text{des}}) \\ \text{subject to} & 0 \leq p_j \leq p_{\text{max}}, \quad j = 1, \ldots, m \end{array}$$

with $h(u) = \max\{u, 1/u\}$



f_0 is convex because maximum of convex functions is convex

exact solution obtained with effort $\approx$ modest factor $\times$ least-squares effort

Linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \le b_i, \quad i = 1, \ldots, m \end{array}$$

solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \ge n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

Convex optimization problem

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

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

  if $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$

- includes least-squares problems and linear programs as special cases

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

Brief history of convex optimization

theory (convex analysis): ca1900–1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, . . .)
- 1970s: ellipsoid method and other subgradient methods
- 1980s: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s–now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

applications

- before 1990: mostly in operations research; few in engineering
- since 1990: many new applications in engineering (control, signal processing, communications, circuit design, . . .); new problem classes (semidefinite and second-order cone programming, robust optimization)

Examples on R

convex:

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

concave:

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

Introduction

3

Examples of convex sets

- hyperplane and halfspaces
- Euclidean balls and ellipsoids
- norm balls and norm cones
- polyhedra, simplex
- positive semidefinite cone

Linear Fractional Transformation

Linear fractional (or projective) function $f : R^m \to R^n$,

$$f(x) = \frac{Ax + b}{c'x + d}$$

defined on domain

$$\text{dom } f = \{x | c'x + d > 0\}$$

Linear fractional function is line segments preserving:

For $x, y \in$ dom f, $f([x, y]) = [f(x), f(y)]$

Therefore, if S is convex, then $f(S)$ is also convex.

Jensen's inequality

basic inequality: if f is convex, then for $0 \leq \theta \leq 1$,

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

extension: if f is convex, then

$$f(\mathbf{E}\, z) \leq \mathbf{E}\, f(z)$$

for any random variable z

basic inequality is special case with discrete distribution

$$\mathbf{prob}(z = x) = \theta, \qquad \mathbf{prob}(z = y) = 1 - \theta$$

Lecture 9

- ◦ Introduction to convex optimization
- • **Portfolio optimization revisited**
- ◦ Duality and distributed optimization

A Dynamic Portfolio of Assets

A portfolio of assets is modelled as

$$\begin{bmatrix}(x_{t+1})_1 \\ \vdots \\ (x_{t+1})_n\end{bmatrix} = \begin{bmatrix}(r_{t+1})_1 & & \\ & \ddots & \\ & & (r_{t+1})_n\end{bmatrix}\begin{bmatrix}(x_t)_1 + (u_t)_1 \\ \vdots \\ (x_t)_n + (u_t)_n\end{bmatrix}$$

or with vector notation $x_{t+1} = R_{t+1}(x_t + u_t)$. Here

$(x_t)_i$	is the is the value of asset i at time t
$(r_{t+1})_i$	is the vector of asset returns, from period t to period $t+1$
$(u_t)_i$	is the is the value of trades in asset i at time t

Assume that r_t for $t = 1, 2, \dots$ are independent random (vector) variables with known mean $\mathbf{E}r_t = \bar{r}_t$ and covariance $\mathbf{E}(r_t - \bar{r}_t)(r_t - \bar{r}_t)^T = \Sigma_t$.

Notation: $\bar{R}_t = \mathbf{E}R_t = \mathrm{diag}(\bar{r}_t)$.

A Portfolio Optimization Problem

Find a trading policy $u_t = \phi_t(x_t)$ that solves the following optimization problem:

$$\text{Minimize} \quad \mathbf{E} \sum_{t=0}^{T} \ell(x_t, u_t)$$

$$\text{subject to} \quad \begin{cases} x_{t+1} = R_{t+1}(x_t + u_t) \\ u_t = \phi_t(x_t) \end{cases} \quad \text{for } t = 0, 1, \ldots, T-1$$

Notation

$\mathbf{1}$	a column vector where every entry equals one.
$\mathbf{1}^T x_t$	the total value of the portfolio before trading at time t
$\mathbf{1}^T u_t$	the total cash put into the portfolio at time t, *excluding* transaction costs
$\ell(x_t, u_t)$	the total cost at time t, *including* transaction costs discount factors, etc.
$-\ell(x_t, u_t)$	the total revenue at time t
$u_t = \phi_t(x_t)$	The trading policy ϕ_t determines the trades u_t from the portfolio positions x_t

The Stage Cost

$$\ell(x_t, u_t) = \begin{cases} \mathbf{1}^T u_t + \psi(x_t, u_t) & \text{if } x_t + u_t \in \mathcal{C}_t \\ \infty & \text{otherwise} \end{cases}$$

In words:
Minimize investments $\mathbf{1}^T u_t$ plus transaction costs $\psi(x_t, u_t)$, while keeping the portfolio within constraints $x_t + u_t \in \mathcal{C}_t$.

Risk Mitigation

Recall that we keep the portfolio within constraints $x_t + u_t \in \mathcal{C}_t$.

The constraint set $\mathcal{C}_t$ can be chosen to mitigate risk:

- The quadratic constraint $(x_t + u_t)^T \Sigma_{t+1} (x_t + u_t) < \gamma_t$ keeps the variance of the portfolio value below γ_t.
- Negative lower bounds $-\gamma_t \leq x_t$ limit the room for risky short positions

Portfolio Optimization by Model Predictive Control

Minimize $\sum_{\tau=t}^{T} \ell(z_\tau, v_\tau)$

subject to $z_{\tau+1} = \bar{R}_{\tau+1}(z_\tau + v_\tau), \quad \tau = t, \dots, T-1$
$z_t = x_t.$

The optimal sequence $v_t^*, \dots, v_{T-1}^*$ is a plan for future trades over the remaining trading horizon, under the (highly unrealistic) assumption that future returns will be equal to their mean values. Only v_t^* is used for trading. At time $t+1$, a new problem is solved.

Lecture 9

- ◦ Introduction to convex optimization
- ◦ Portfolio optimization revisited
- • **Duality and distributed optimization**

Distributed Optimization

Large scale problems cannot be solved centralized.

- Computational complexity
- Memory constraints
- Communication constraints

Use market mechanisms for distributed optimization!

Linear Programming Example

Product	# of items	Profit / item
Garden Furniture 1	x_1	c_1
Garden Furniture 2	x_2	c_2
Sled 1	x_3	c_3
Sled 2	x_4	c_4

Constraints for sub-division 1:

$$7x_1 + 10x_2 \leq 100 \qquad \text{(Sawing)}$$
$$16x_1 + 12x_2 \leq 135 \qquad \text{(Assembling)}$$

Constraints for sub-division 2:

$$10x_3 + 9x_4 \leq 70 \qquad \text{(Sawing)}$$
$$6x_3 + 9x_4 \leq 60 \qquad \text{(Assembling)}$$

Painting Constraint:

$$5x_1 + 3x_2 + 3x_3 + 2x_4 \leq 45$$

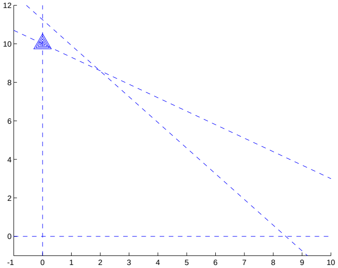
Linear Programming Example

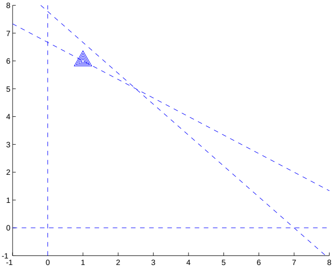
Mathematical formulation:

$$
\begin{aligned}
\text{Maximize} \quad & c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \\
\text{subject to} \quad & 7x_1 + 10x_2 \leq 100 \\
& 16x_1 + 12x_2 \leq 135 \\
& 10x_3 + 9x_4 \leq 70 \\
& 6x_3 + 9x_4 \leq 60 \\
& 5x_1 + 3x_2 + 3x_3 + 2x_4 \leq 45 \\
& x \succeq 0
\end{aligned}
$$

Numerical Results

Optimal solution for Division 1 (left) and Division 2 (right).
Common constraint active (i.e. equality holds).





Dual Variables

Dual variables are the marginal prices for resources:

If the capacity for a resource is increased by 1, the total profit is increased by the corresponding dual variable.

This gives insight to which resource to increase to gain most

Numerical Results

Optimal dual variables and their respective constraints:

Constraint	Dual variable
$7x_1 + 10x_2 \leq 100$	1.04
$16x_1 + 12x_2 \leq 135$	0
$10x_3 + 9x_4 \leq 70$	0
$6x_3 + 9x_4 \leq 60$	0.4
$5x_1 + 3x_2 + 3x_3 + 2x_4 \leq 45$	3.2

Optimal value: $p^* = c^T x^* = 272$

If common (painting) constraint capacity increased to 46, optimal value becomes $272 + 3.2 = 275.2$

Company would gain most by increasing painting capacity

Linear Programming Duality

Linear Program:

$$p^* = \begin{cases} \max\limits_{x} & c^T x \\ \text{subject to} & Ax \preceq b, \, x \succeq 0 \end{cases}$$

where $p^* = c^T x^*$ is the optimal value attained by x^*.

For the constraints $Ax \preceq b$, introduce dual variables $\lambda \succeq 0$ and construct the corresponding dual function $g(\lambda)$:

$$g(\lambda) = \max_{x \succeq 0} \left[c^T x + \lambda^T (b - Ax) \right]$$

The second term in the bracket is non-negative when $Ax \preceq b$. Hence $g(\lambda) \geq p^*$.

Linear Programming Duality cont'd

Tightest upper bound to p^* obtained by minimizing $g(\lambda)$:

$$d^* = \min_{\lambda \succeq 0} g(\lambda) = \min_{\lambda \succeq 0} \max_{x \succeq 0} \left[c^T x + \lambda^T (b - Ax) \right]$$

Optimal value d^* for this min-max problem is attained by $x = x^*$ and $\lambda = \lambda^*$.

Further we have that $p^* = c^T x^* = d^*$. This equality is referred to as *strong duality*

This min-max problem is used later to distribute the optimization

Dual optimal values and d^* can be obtained by solving

$$\begin{aligned} \min_{\lambda} \quad & b^T \lambda \\ \text{subject to} \quad & A^T \lambda \succeq c, \lambda \succeq 0 \end{aligned}$$

Note symmetry to primal problem

Linear Programming Duality

$$
\begin{array}{lll}
\max_{x} & c^T x & \\
\text{with} & Ax \preceq b & \\
 & x \succeq 0 &
\end{array}
\quad = \quad
\begin{array}{ll}
\min_{\lambda} & b^T \lambda \\
\text{with} & A^T \lambda \succeq c \\
 & \lambda \succeq 0
\end{array}
$$

Optimality Conditions

x^* is primal optimal if and only if there exists λ^* such that

$$
\begin{aligned}
Ax^* &\preceq b & A^T\lambda^* &\succeq c \\
\lambda^* &\succeq 0 & x^* &\succeq 0 \\
(A_i x^* - b_i)\lambda_i^* &= 0 & (A_j^T\lambda^* - c_j)x_j^* &= 0
\end{aligned}
$$

These conditions are called the KKT-conditions for this LP-problem

Distribution of LP Example

Solve the LP example

$$
\begin{aligned}
\text{Maximize} \quad & c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \\
\text{subject to} \quad & 7x_1 + 10x_2 \le 100 \\
& 16x_1 + 12x_2 \le 135 \\
& 10x_3 + 9x_4 \le 70 \\
& 6x_3 + 9x_4 \le 60 \\
& 5x_1 + 3x_2 + 3x_3 + 2x_4 \le 45 \\
& x \succeq 0
\end{aligned}
$$

in a distributed fashion using the dual problem

Distribution of LP Example cont'd

Dual problem when constraint with all variables is "dualized":

$$\min_{\lambda\geq 0}\max_{x\geq 0} \quad c^T x + \lambda(45 - 5x_1 + 3x_2 + 3x_3 + 2x_4)$$

subject to

$$7x_1 + 10x_2 \leq 100$$
$$16x_1 + 12x_2 \leq 135$$
$$10x_3 + 9x_4 \leq 70$$
$$6x_3 + 9x_4 \leq 60$$

For fixed $\lambda = \bar{\lambda}$, the inner maximization can be decomposed to two sub-problems (one for each sub-division) P_1 and P_2:

$$P1 : \begin{cases} \max\limits_{x_1\geq 0, x_2\geq 0} & c_1x_1 + c_2x_2 - \bar{\lambda}(5x_1 + 3x_2) \\ \text{s. t.} & 7x_1 + 10x_2 \leq 100 \\ & 16x_1 + 12x_2 \leq 135 \end{cases}$$

$$P2 : \begin{cases} \max\limits_{x_3\geq 0, x_4\geq 0} & c_3x_3 + c_4x_4 - \bar{\lambda}(3x_3 + 2x_4) \\ \text{s. t.} & 10x_3 + 9x_4 \leq 70 \\ & 6x_3 + 9x_4 \leq 60 \end{cases}$$

Distribution Example cont'd

With fixed $x = \bar{x}$ head-quarters can update the dual variable λ to decrease the value of the outer minimization problem:

$$\bar{\lambda}^{+} = \bar{\lambda} - \alpha(45 - 5\bar{x}_1 + 3\bar{x}_2 + 3\bar{x}_3 + 2\bar{x}_4)$$

where α is the step-size, which is chosen so that $\bar{\lambda}^{+} \geq 0$ is maintained.

Motivation, the dual objective with $\bar{\lambda}$ is

$$g(\bar{\lambda}) = p^T\bar{x} + \bar{\lambda}(45 - 5\bar{x}_1 + 3\bar{x}_2 + 3\bar{x}_3 + 2\bar{x}_4)$$

and with $\bar{\lambda}^{+}$:

$$\begin{aligned} g(\bar{\lambda}^{+}) &= p^T\bar{x} + \bar{\lambda}^{+}(45 - 5\bar{x}_1 + 3\bar{x}_2 + 3\bar{x}_3 + 2\bar{x}_4) = \\ &= p^T\bar{x} + \bar{\lambda}(45 - 5\bar{x}_1 + 3\bar{x}_2 + 3\bar{x}_3 + 2\bar{x}_4) - \\ &\quad - \alpha(45 - 5\bar{x}_1 + 3\bar{x}_2 + 3\bar{x}_3 + 2\bar{x}_4)^2 \leq g(\bar{\lambda}) \end{aligned}$$

Distributed Optimization Algorithm

1. Initialize algorithm by $\lambda^{(0)} = 0$ and $x^{(0)} = 0$.
2. For fixed $\lambda = \lambda^{(k)}$ let the sub-divisions solve their respective optimization problems to find the state vector $x^{(k)}$.
3. Define $\lambda^{(k+1)} = \max(0, \lambda^{(k)} - \alpha^{(k)}(45 - 5x_1^{(k)} + 3x_2^{(k)} + 3x_3^{(k)} + 2x_4^{(k)}))$
4. Set $k \leftarrow k + 1$ and go to step 2.

Convergence to optimal value and convergence in dual variables guaranteed with this algorithm, if the step size λ^k is appropriately chosen

Convergence in primal variables guaranteed if objective strictly concave

A Convergence Theorem

Suppose $\|\lambda^{(1)} - \lambda^*\| \leq R$ and consider the iteration

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha_k g^{(k)}$$

where $g^{(k)}$ satisfies the “subgradient” inequality

$$f(\lambda^*) \geq f(\lambda^{(k)}) + (g^{(k)})^T(\lambda^* - \lambda^{(k)}) \qquad \text{for all } \lambda^{(k)}$$

and f satisfies the Lipschitz condition

$$|f(u) - f(v)| \leq G\|u - v\| \qquad \text{for all } u, v$$

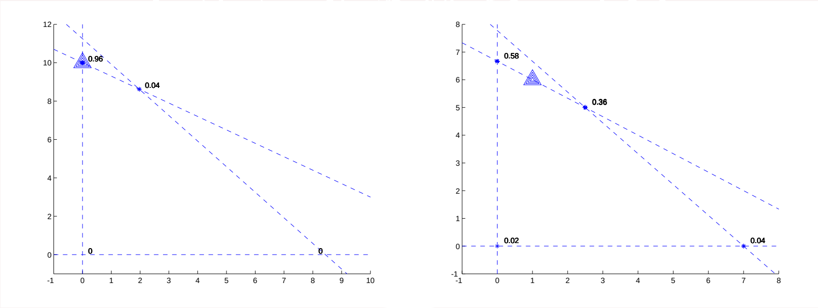
Define $f_{\text{best}}^{(k)} = \min\{f(\lambda^{(1)}), \dots, f(\lambda^{(k)})\}$. Then

$$f_{\text{best}}^{(k)} - f(\lambda^*) \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i}$$

In particular $f_{\text{best}}^{(k)} \to f(\lambda^*)$ as $k \to \infty$ if $\alpha_k = \frac{1}{k}$.

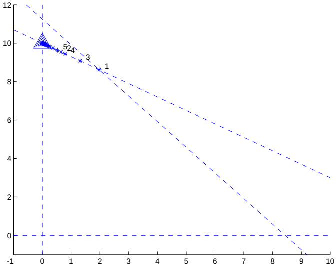
Numerical Results

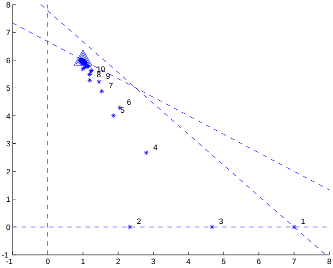
Primal variable iterates (x) for division 1 (left) and division 2 (right) with their respective local constraints. Triangles show optimal solution (which is not in a corner in division 2 due to the constraint with all variables). The numbers show the fraction of iterates in that corner.



Numerical Results

Same as previous slide where a certain convex combination of the solutions is plotted. These converge to the primal optimal solution. The numbers correspond to iterate number.





Comments on Distributed Optimization

- Decomposition scheme is called dual decomposition
- Dual decomposition most useful for large problems with
  - few constraints involving all variables
  - many local constraints
- Applicable to other types of optimization problems as well (such as quadratic problems)

Lecture 8 and 9

Lecture 8

- Linear Programming (LP)
- LP in production planning example
- Model Predictive Control
- A portfolio optimization problem

Lecture 9

- Introduction to convex optimization
- Portfolio optimization revisited
- Duality and distributed optimization