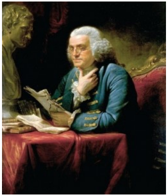
Quiz #2 Details

"I didn't fail the test, I just found 100 ways to do it wrong"

Benjamin Franklin



Quiz #2: Monday, April 15, 4-5:15pm CDT

- In Class
- McBurney Students, ME 3126(?) 4-5:55pm CDT

Today's Docket

- Midterm #2 Details
- Review Course Topics
    - (Network) Duality
    - Multiobjective Optimization and Regularization
    - Convex Optimization
        - Quadratic Functions and Quadratic Programming
        - Cone Programming
    - Integer Programming
        - Relaxations
        - Simple Modeling
- Questions
- Practice Problems?

Second Midterm

- In class—4pm-5:15pm, next Monday, April 15.
- Coverage: Everything in Notes from Lecture 9-Lecture 15
  - Although will need to know basic modeling and duality from first midterm
- (Currently) Five Problems
  1. True/False
  2. Short Answer
  3. Modeling
  4. Duality and Networks
  5. Modeling

Second Midterm

- No Note Sheet Allowed This Time
- No calculators or other electronic devices
- Bring your ID to the exam and place it out. We will be checking IDs during the exam.
  - Also remember to take it with you
- Spread out in the room! (We may ask you to move)

Second Midterm

- No Note Sheet Allowed This Time
- No calculators or other electronic devices
- Bring your ID to the exam and place it out. We will be checking IDs during the exam.
  - Also remember to take it with you
- Spread out in the room! (We may ask you to move)

When Exam is Over...

- I will ask everyone to put their pencils down
- You are not to pick up your pencil again, even to write your name.
- If you pick up your pencil again, I will take off 10% of grade.

Who Says There's No Such Thing?

Free Lunch!

- Eric and Sanjai[a] will take the top three scorers out to lunch!
- Previous winners ineligible.

[a]Prof. Linderoth will also join and pay

Math Modeling—Being More Strict

- You *must* write proper mathematical notation and syntax to get credit. We will give less partial credit.
  - Writing JuMP code `@constraint(m, sum(x) == 1)` is *not* mathematical syntax
- You have to write specifically the set of entities over which you are doing a sum.
  - $\sum_i x_i$ is incorrect. i is an index, not a set
  - $\sum x_i \forall i \in S$ is incorrect
  - $\sum_{i=1}^4 x_i$ or $\sum_{i \in S} x_i$ is proper notation
- $x_k \geq 42 \forall k$ is also incorrect notation. k is an index. For all k *in what set?*

Lecture 9

- Duality / Complementary Slackness
- Network Duality (and Interpretation)
- Max-Flow Min Cut

Lecture 10

- Quadratic Functions/Quadratic Forms
- Spectral Theorem
- PSD Matrices
- Definition of Convex and Concave Functions
- Examples of convex and concave functions

Lecture 11

- Least Squares is a (Convex) Optimization Problem
- Geometry of Least Sqaures
- Regression and Curve fitting
- Multiobjectve and Tradeoffs
- Pareto Cuirves
- Regularization
- Norm balls
- Hierarchical Optimization

Lecture 12

- Ellipsoids
- Degenerate ellipsoids
- Shifted Ellipdoids
- Quadraic Programs: Solution properties (Convex/Nonconvex)
- Where do Quadratic occur?
- Portfolio Optimization

Lecture 13

- Definition of Cone
- Types of Cones

1. Second Order Cone
2. Rotated Second Order Cone
3. Modeling them wuth Julia/JuMP
4. Rational Powers
5. Semidefinite Cone
6. Exponential Cone

Lectures 14 and 15

- Geometry of Feasible Region
- Relaxations and LP relaxation and Convex hull
- IP Modeling: Fixed Costs. Big M
- IP Modeling Variable Lower Bounds
- IP Modeling: "Simple" Logic

Duality review

Every LP has a dual, which is also an LP.

- Every primal constraint corresponds to a dual variable
- Every primal variable corresponds to a dual constraint

Minimization LP	Maximization LP
Nonnegative variable $\geq$	Inequality constraint $\leq$
Nonpositive variable $\leq$	Inequality constraint $\geq$
Free variable	Equality constraint $=$
Inequality constraint $\geq$	Nonnegative variable $\geq$
Inequality constraint $\leq$	Nonpositive variable $\leq$
Equality constraint $=$	Free Variable

Dual of minimum-cost flow problems

$$\begin{aligned} \min_{x} \quad & c^{\mathsf{T}} x && \text{(minimization)} \\ \text{s.t.} \quad & Ax = b && \text{(constraint } =) \\ & x \le q && \text{(constraint } \le) \\ & x \ge 0 && \text{(variable } \ge) \end{aligned}$$

- balance constraints (at nodes)
- capacity constraints (on edges)
- flow variables x (on edges)

$$\begin{aligned} \max_{\mu,\eta} \quad & b^{\mathsf{T}} \mu + q^{\mathsf{T}} \eta && \text{(maximization)} \\ \text{s.t.} \quad & \mu \text{ free} && \text{(variable free)} \\ & \eta \le 0 && \text{(variable } \le) \\ & A^{\mathsf{T}} \mu + \eta \le c && \text{(constraint } \le) \end{aligned}$$

- dual variables μ (at nodes)
- dual variables η (each edge)
- dual constraints (each edge)

- important: A has one '1' and one '-1' per column

Transportation Problem

Primal problem:

- Pick how much commodity flows along each edge of the network to minimize the total transportation cost while satisfying supply/demand constraints.
- If each supply/demand b_i is integral, flows will be integral.

Dual problem:

- Pick the buy/sell price for the commodity at each node of the network to maximize the total profit while ensuring that the prices are competitive.
- If each edge cost c_{ij} is integral, prices will be integral.

Longest path (dual)

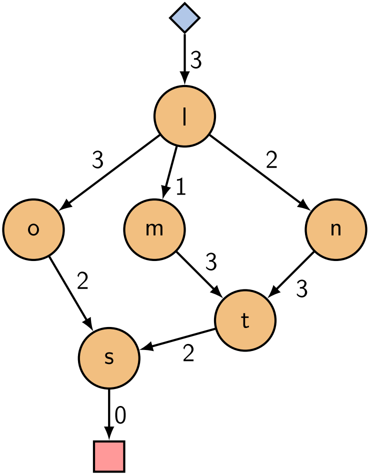
Recall the house-building example, a longest-path problem.

Key players:

- Primal variables: $x_{ij} \in \{0, 1\}$
- Dual constraints: $t_j - t_i \geq c_{ij}$

Complementary slackness:

- if $x_{ij} = 1$ then $t_j - t_i = c_{ij}$ (longest path corresponds to tight time constraints)
- if $t_j - t_i > c_{ij}$ then $x_{ij} = 0$ (this path has slack)



Max-flow summary

Primal problem:

- Each edge of the network has a maximum capacity.
- Pick how much commodity flows along each edge to maximize the total amount transported from the start node to the end node while obeying conservation constraints. This total amount of flow is called the *max flow*.

Dual problem:

- Find a partition of the nodes into two subsets where the first subset includes the start node and the second subset includes the end node.
- Choose the partition that minimizes the sum of capacities of all edges that connect starting subset to ending subset. This total capacity is called the *min cut*.

Cuts

- Given directed graph $G = (V, E)$ with edge capacities $u \in \mathbb{R}^E$, source node $s \in V$ and sink node $t \in V$.
- Let S be *any* set of nodes containing the source but not containing the sink
- The partition of the nodes $(S, \overline{S})$ is called a *cut*.
- The *cut-set* is the set of edges with starting point in S and ending point in $\overline{S}$:

$$\delta(S, \overline{S}) := \{(i, j) \in A : i \in S, j \in \overline{S}\}$$

- The *capacity of a cut* is the sum of the edge capacities of the arcs in its cut-set:

$$\text{cap}(S, \overline{S}) := \sum_{(i,j) \in \delta(S, \overline{S})} u_{ij}$$

Max Flow Min Cut

- Since each unit of flow from $s \to t$ must use an arc from $\delta(S, V \setminus S)$, the capacity of *any* cut-set provides an upper bound on the maximum flow

---

Max-Flow Min-Cut Theorem

For any network, the maximum feasible flow from the source to the sink *equals* the minimum cut capacity for all cuts in the network.

Max Flow – Dual

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} u_{ij}\lambda_{ij} \\
\text{s.t.} \quad & \mu_i - \mu_j + \lambda_{ij} \geq 0 \quad \forall (i,j) \in A \\
& -\mu_s + \mu_t \geq 1 \\
& \mu_i \text{ free} \quad \forall i \in N \\
& \lambda_{ij} \geq 0 \quad \forall (i,j) \in A
\end{aligned}
$$

- We can assume $\mu_s = 0$ in any dual solution
- Integrality property holds for max flow dual.
- There is an optimal dual solution with μ_i is 0 or 1 and λ_{ij} is 0 or 1

Max-Flow Min Cut Theorem (like HW)

- For any cut $(S, \overline{S})$, consider the values

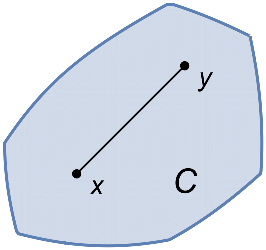
$$\mu'_i = \begin{cases} 1 & i \in \overline{S} \\ 0 & \text{otherwise} \end{cases} \qquad \lambda'_{ij} = \begin{cases} 1 & (i,j) \in A, i \in S, j \in \overline{S} \\ 0 & \text{otherwise} \end{cases}$$

- This "dual solution" is feasible. (check)
- Its objective value sums u_{ij} for $(i,j) \in A$ with $i \in S$ and $j \in \overline{S}$: The capacity of the cut $(S, \overline{S})$.
- The value of any cut (any dual feasible solution) provides an upper bound on the optimal solution value
- By using strong duality, this proves the max flow min cut theorem.

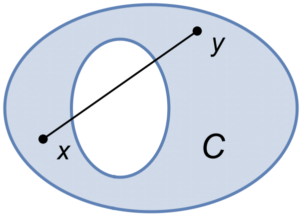
Convex sets

A set $C \subseteq \mathbb{R}^n$ is *convex* if for all $x, y \in C$ and all $0 \leq \alpha \leq 1$, we have: $\alpha x + (1 - \alpha)y \in C$.

- every line segment must be contained in the set
- can include boundary or not
- can be finite or not



convex set



nonconvex set

Quadratic

Any quadratic function $f(x_1, \ldots, x_n)$ can be written in the form $x^\mathsf{T} Q x$ where Q is a **symmetric matrix** ($Q = Q^\mathsf{T}$).

Theorem. Every real symmetric matrix $Q = Q^\mathsf{T} \in \mathbb{R}^{n \times n}$ can be decomposed into a product:

$$Q = U \Lambda U^\mathsf{T}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a real diagonal matrix, and $U \in \mathbb{R}^{n \times n}$ is an orthogonal matrix. i.e. it satisfies $U^\mathsf{T} U = I$.

More Quadratic

For a matrix $Q = Q^\mathsf{T}$, the following are equivalent:

1. $x^\mathsf{T} Q x \geq 0$ for all $x \in \mathbb{R}^n$
2. all eigenvalues of Q satisfy $\lambda_i \geq 0$

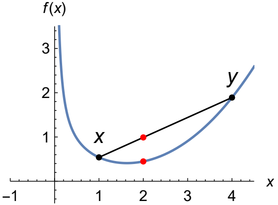
- A matrix with this property is called *positive semidefinite* (PSD). The notation is $Q \succeq 0$.
- If $P \succeq 0$ then $-P \preceq 0$
- If $P \succeq 0$ and $\alpha > 0$ then $\alpha P \succeq 0$
- If $P \succeq 0$ and $Q \succeq 0$ then $P + Q \succeq 0$
- Every $R = R^\mathsf{T}$ can be written as $R = P - Q$ for some appropriate choice of matrices $P \succeq 0$ and $Q \succeq 0$.

Convex functions

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* for all $x, y \in \mathbb{R}^n$ and $0 \le \alpha \le 1$, we have:

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha) f(y)$$

- f is *concave* if $(-f)$ is convex.



Convex Function

- "Holds water"
- All local minima are global minima

Convex and concave functions on $\mathbb{R}$

Convex functions on $\mathbb{R}$

- Affine: $ax + b$.
- Absolute value: $|x|$.
- Quadratic: ax^2 for any $a \geq 0$.
- Exponential: a^x for any $a > 0$.
- Powers: x^α for $x > 0$, $\alpha \geq 1$ or $\alpha \leq 0$.
- Negative entropy: $x \log x$ for $x > 0$.

Concave functions on $\mathbb{R}$

- Affine: $ax + b$.
- Quadratic: ax^2 for any $a \leq 0$.
- Powers: x^α for $x > 0$, $0 \leq \alpha \leq 1$.
- Logarithm: $\log x$ for $x > 0$.

Convex and concave functions

Convex functions on $\mathbb{R}^n$

- Affine: $a^\mathsf{T} x + b$.
- Norms: $\|x\|_2$, $\|x\|_1$, $\|x\|_\infty$
- Quadratic form: $x^\mathsf{T} Q x$ for any $Q \succeq 0$

Concave functions on $\mathbb{R}^n$

- Affine: $a^\mathsf{T} x + b$.
- Quadratic form: $x^\mathsf{T} Q x$ for any $Q \preceq 0$

- For differentiable functions on $\mathbb{R}$,
  - $f(x)$ is convex if $f''(x) \geq 0$,
  - $f(x)$ is concave if $f''(x) \leq 0$,

Building convex functions

1. Nonnegative weighted sum: If $f(x)$ and $g(x)$ are convex and $\alpha, \beta \geq 0$, then $\alpha f(x) + \beta g(x)$ is convex.
2. Composition with an affine function:
   If $f(x)$ is convex, so is $g(x) := f(Ax + b)$
3. Pointwise maximum: If $f_1(x), \ldots, f_k(x)$ are convex, then $g(x) := \max\{f_1(x), \ldots, f_k(x)\}$ is convex.

- N.B.: A composition of two convex functions is not necessarily convex!
  - *Example* in $\mathbb{R}$: $g(x) = |x|$ and $h(x) = x^2 - 1$ are both convex, but $g(h(x)) = |x^2 - 1|$ is not convex.

Optimal tradeoffs

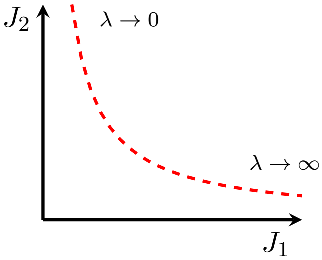
- Suppose $J_1(x)$ and $J_2(x)$ different functions.
- We would like to make **both** J_1 and J_2 small.
- A sensible approach: solve the optimization problem:

$$\underset{x}{\text{minimize}} \quad J_1(x) + \lambda J_2(x)$$

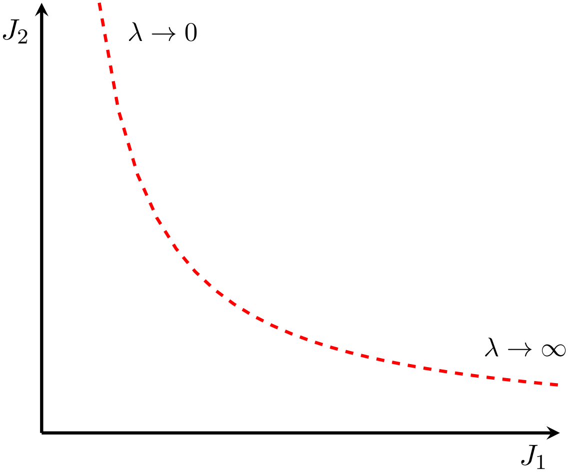
  where $\lambda > 0$ is a (fixed) *tradeoff parameter*.
- Then tune λ to explore possible results.
  - When $\lambda \to 0$, we place more weight on J_1
  - When $\lambda \to \infty$, we place more weight on J_2

Tradeoff analysis

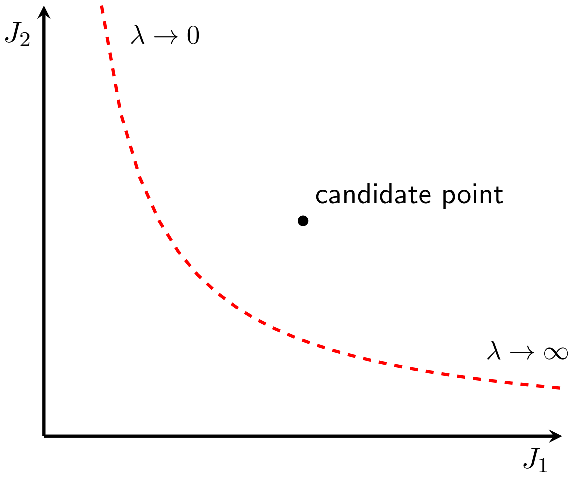
1. Choose values for λ
2. For each λ value, find $\hat{x}_\lambda$ that minimizes $J_1 + \lambda J_2$.
3. For each $\hat{x}_\lambda$, also compute the corresponding $J_1^\lambda := J_1(\hat{x}_\lambda)$ and $J_2^\lambda := J_2(\hat{x}_\lambda)$.
4. Plot $(J_1^\lambda, J_2^\lambda)$ for each λ and connect the dots.



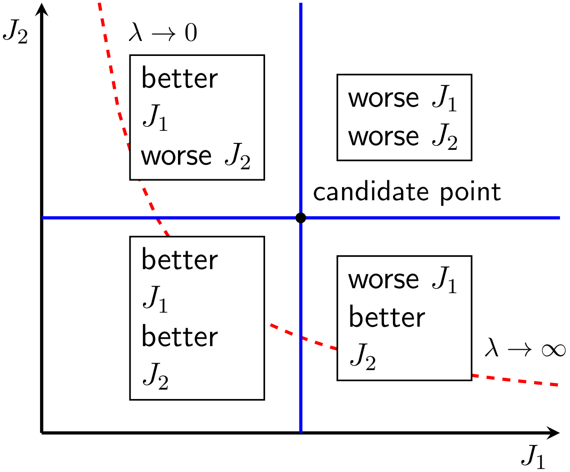
Pareto curve



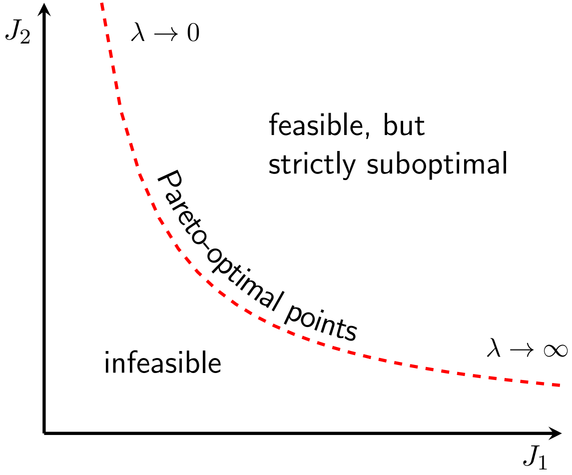
Pareto curve



Pareto curve

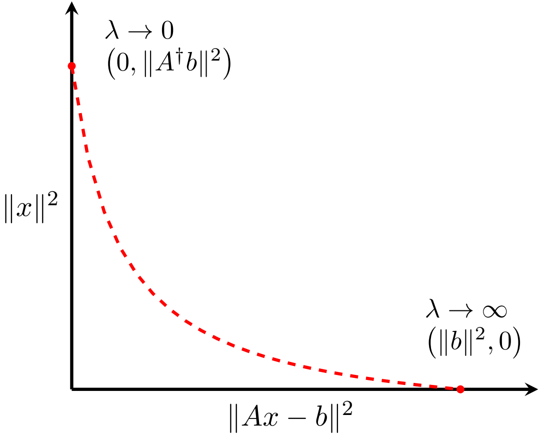


Pareto curve



Tradeoff visualization—Regression

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda \|x\|^2$$



Regularization

Regularization: Additional penalty term added to the cost function to encourage a solution with desirable properties.

Regularized least squares:

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda R(x)$$

- $R(x)$ is the regularizer (penalty function)
- λ is the regularization parameter
- The model has different names depending on $R(x)$.

Regularized least squares turns out to be important in many contexts!

Regularization

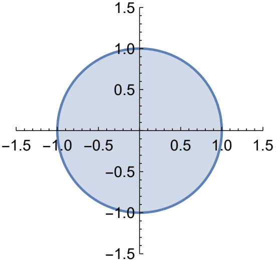
$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda R(x)$$

1. If $R(x) = \|x\|^2 = x_1^2 + x_2^2 + \cdots + x_n^2$
   It is called: L_2 *regularization*, *Tikhonov regularization*, or *Ridge regression* depending on the application. It has the effect of *smoothing* the solution.
2. If $R(x) = \|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$
   It is called: L_1 *regularization* or *LASSO*. It has the effect of *sparsifying* the solution ($\hat{x}$ will have few nonzero entries).
3. $R(x) = \|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$
   It is called L_∞ *regularization* and it has the effect of *equalizing* the solution (makes many components tie for the max absolute value).

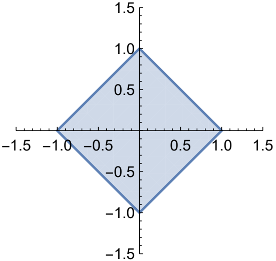
Norm balls

For a norm $\|\cdot\|_p$, the **norm ball** of radius r is the set:

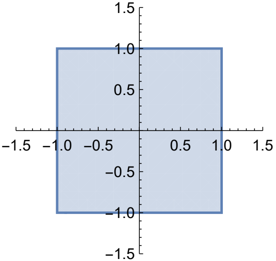
$$B_r = \{x \in \mathbb{R}^n \mid \|x\|_p \leq r\}$$



$$\|x\|_2 \leq 1$$
$$x^2 + y^2 \leq 1$$



$$\|x\|_1 \leq 1$$
$$|x| + |y| \leq 1$$



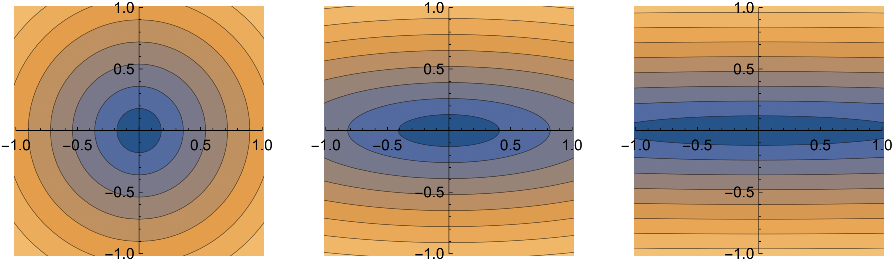
$$\|x\|_\infty \leq 1$$
$$\max\{|x|, |y|\} \leq 1$$

Ellipsoids

- The set of x satisfying $x^{\mathsf{T}} Q x \le b$ is an *ellipsoid* if $Q \succ 0$.
- If $Q \succ 0$, then $x^{\mathsf{T}} Q x \le b \iff \|Q^{1/2} x\|^2 \le b$.
- (Recall that if we write the eigenvalue decomposition $Q = U \Lambda U^T$, then $Q^{1/2} = U \Lambda^{1/2} U^T$, where $\Lambda^{1/2}$ is the diagonal matrix whose diagonal entries are the square roots of the diagonals of Λ.)

Degenerate Ellipsoids

Ellipsoid axes have length $\frac{1}{\sqrt{\lambda_i}}$. When an eigenvalue is close to zero, contours are stretched in that direction.



$x^2 + y^2$ $\qquad$ $\frac{1}{10}x^2 + y^2$ $\qquad$ $\frac{1}{100}x^2 + y^2$

- Warmer colors = larger values
- If $\lambda_i = 0$, then $Q \succeq 0$. The ellipsoid $x^\mathsf{T} Q x \leq 1$ is *degenerate* (stretches out to infinity (is constant) in direction u_i).

Shifted Ellipsoid

- If $Q \succ 0$, then the quadratic form with extra linear term:

$$x^{\mathsf{T}}Qx + r^{\mathsf{T}}x + s$$

  defines an *shifted* ellipsoid, whose center is not at 0.
- This is an ellipse centered at $-\frac{1}{2}Q^{-1}r$ — but its shape is still defined by the matrix Q.

Quadratic programs

Quadratic program (QP) is like an LP, but with quadratic cost:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\mathsf{T} P x + q^\mathsf{T} x + r \\ \text{subject to:} \quad & Ax \le b \end{aligned}$$

Note: Constraints are still linear!

- If $P \succeq 0$, it is a *convex QP*
  - feasible set is a polyhedron
  - solution can be on boundary or in the interior
  - relatively easy to solve
- If $P \not\succeq 0$, it is **very hard** to solve in general.

Example: portfolio optimization

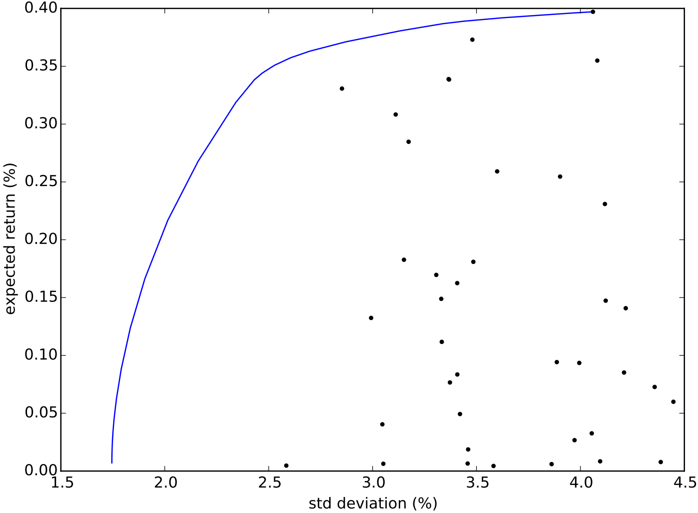
Suppose we buy x_i of asset Z_i. We want:

- A high total return. Maximize $x^\mathsf{T}\mu$.
- Low variance (risk). Minimize $x^\mathsf{T}\Sigma x$.

Pose the optimization problem as a tradeoff:

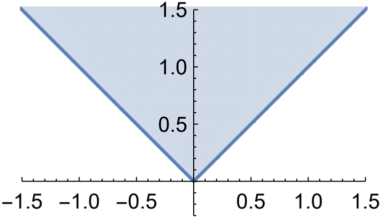
$$\begin{array}{ll} \underset{x}{\text{minimize}} & -x^\mathsf{T}\mu + \lambda x^\mathsf{T}\Sigma x \\ \text{subject to:} & x_1 + \cdots + x_{225} = 1 \\ & x_i \geq 0 \end{array}$$

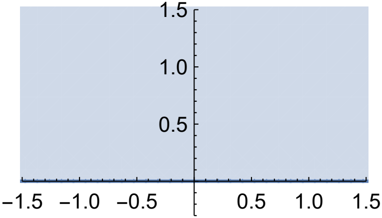
Efficient Frontier



What is a cone?

- A set of points $C \in \mathbb{R}^n$ is called a *cone* if $\alpha x \in C$ whenever $x \in C$ and $\alpha > 0$.
- A cone C is a *convex cone* if it is a convex set: $\lambda x + (1-\lambda)y \in C \ \forall \lambda \in [0,1]$ and $x, y \in C$
- Simple examples: $|x| \leq y$ and $y \geq 0$ are both cones.





Cone Program

- Given m affine mappings $A^i x + b^i, i = 1, \ldots, m$
- Given m cones $C^i, i = 1, \ldots, m$

Cone Program

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & A^i x + b^i \in C^i \quad \forall i = 1, \ldots, m \end{aligned}$$

where C_i is a (closed) convex cone for each $i = 1, \ldots, m$

- Note that you can always *add* extra variables.
- Define new (vectors) of variables $y^i = A^i x + b^i$, and then just say *these variables*: $y^i \in C^i$.

Careful About Bounds

- Suppose you are given a second order cone constraint:

$$\|Ax + b\| \leq c^\mathsf{T} x + d$$

- With $Ax + b \in \mathbb{R}^m$ (so $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$)
- N.B. Note that this implies that $c^\mathsf{T} x + d \geq 0$
- Define vector (and add constraints) $y = Ax + b$
- Define scalar variable (and add constraint) $t = c^\mathsf{T} x + d$. Add $t \geq 0$
- Then you just have that the vector (t, y) is in the $1 + m$ dimensional Lorentz (second-order)/ice cream cone.

Cones

- Second-Order (Lorentz) Cone: (of dimension $n+1$)

$$\mathcal{L} := \{(t, x) \in \mathbb{R}^{1+n} : t \geq \|x\|_2\}$$

- Rotated Second-Order Cone: (of dimension $n+2$)

$$\mathcal{R} := \{(u, t, x) \in \mathbb{R}^{2+n} : 2ut \geq x^{\mathsf{T}} x, u \geq 0, t \geq 0\}$$

- Positive Semidefinite Cone: (of order n)

$$\mathcal{S} := \{X \in \mathbb{R}^{n \times n} : X = X^{\mathsf{T}}, w^T X w \geq 0 \ \forall w \in \mathbb{R}^n\}$$

- Exponential Cone: (always in dimension 3)

$$\mathcal{E} := \{(x, y, z) \in \mathbb{R}^3 : y \exp(x/y) \leq z, y > 0\}$$

Modeling with Cones

- $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- Modeling 3/2 power

$$2ut \geq x^2, 2\frac{1}{8}x \geq u^2 \Rightarrow 4u^2t^2 \cdot \frac{1}{4}x \geq x^4u^2 \Rightarrow t \geq x^{3/2}$$

- You do not need to memorize the 3/2 trick.

Semidefinite program (SDP)

Standard form #1: (looks like the standard form for an LP)

$$
\begin{aligned}
\underset{X}{\text{maximize}} \quad & \langle C, X \rangle \\
\text{subject to:} \quad & \langle A_i, X \rangle \leq b_i \quad \text{for } i = 1, \ldots, m \\
& X \succeq 0
\end{aligned}
$$

Standard form #2:

$$
\begin{aligned}
\underset{x}{\text{maximize}} \quad & c^{\mathsf{T}} x \\
\text{subject to:} \quad & Q_0 + \sum_{i=1}^{m} x_i Q_i \succeq 0
\end{aligned}
$$

Exponential Cone

- The exponential cone is the set of points in $\mathbb{R}^3$ with

$$K_{exp} := \{(x, y, z) : y \exp(x/y) \le z, y > 0\}$$

- It is useful for things modeling exponential and logarithms.
- Epigraph of exponential:

$$t \ge e^x \Leftrightarrow (x, 1, t) \in K_{exp}$$

- Hypograph of log:

$$t \le \log(x) \Leftrightarrow (t, 1, x) \in K_{exp}$$

- Entropy:

$$t \le -x \log(x) \Leftrightarrow t \le x \log(1/x) \Leftrightarrow (t, x, 1) \in K_{exp}$$

No questions on log-sum-exp and geometric programming. (For this exam...)

Hierarchy of complexity

From simplest to most complicated:

1. linear program
2. convex quadratic program
3. convex quadratically constrained quadratic program
4. second-order cone program
5. semidefinite program
6. general convex program

Important notes

- more complicated just means that e.g. every LP is a SOCP (by setting appropriate variables to zero), but a general SOCP cannot be expressed as an LP.
- in general: strive for the simplest model possible

Mixed-Integer Linear Programs

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^\mathsf{T} x \\ \text{subject to:} \quad & Ax \le b \\ & x \ge 0 \\ & x_i \in S_i \end{aligned}$$

where S_i can be:

- The real numbers, $\mathbb{R}$
- The integers, $\mathbb{Z}$
- Boolean / Binary $\{0, 1\}$
- A discrete set, $\{v_1, v_2, \ldots, v_k\}$

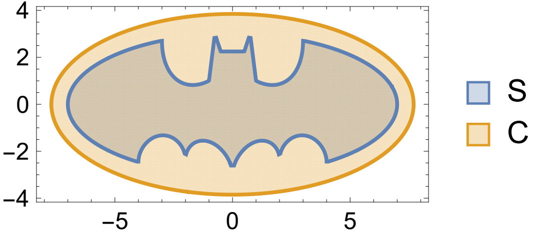
Convex relaxation

$$\underset{x \in S}{\text{minimize}} \quad f(x)$$

Two key (algorithmic) ideas for deadling with nonconvexity:

1. *Function relaxation:* if f is troublesome, bound it with a function that is easier to work with, e.g. a convex function.
2. *Constraint relaxation:* If S is troublesome, find a bigger set that is easier to work with, e.g. a convex set.

Constraint relaxation



- Solve $h_{\text{opt}} = \min_{x \in C} f(x)$ and let $\tilde{x}$ be the optimal x.
- We have the bound: $h_{\text{opt}} = f(\tilde{x}) \leq f_{\text{opt}} \leq f(x)$ for $x \in S$.
- If $\tilde{x} \in S$ then the bound is tight and $f_{\text{opt}} = f(\tilde{x})$.

Pick a convex C so that h_{opt} and $\tilde{x}$ are easy to find!

Common relaxations

1. Boolean constraint:

$$x \in \{0, 1\} \quad \Longrightarrow \quad 0 \leq x \leq 1$$

If x_{opt} is 0 or 1, relaxation is exact.

2. Convex equality:

$$f(x) = 0 \quad \Longrightarrow \quad f(x) \leq 0$$

If $f(x_{\text{opt}}) = 0$, relaxation is exact.

3. A constraint you don't like:

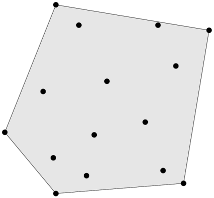
$$x \neq 3 \quad \Longrightarrow \quad \text{just remove the constraint!}$$

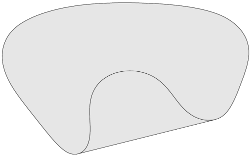
If $x_{\text{opt}} \neq 3$, relaxation is exact.

Convex hull

The **convex hull** of a set S, written $\mathrm{conv}(S)$ is the smallest convex set that contains S.

- (Also): the intersection of all convex sets containing S





Key Idea

Formulations whose (convex) relaxation more closely approximate the convex hull of feasible solutions are "better"

Detour: Logic!

How do we represent: "if $x > 0$ then $z = 1$"?

- Contrapositive: "if $z = 0$ then $x \leq 0$"
- Since $x \geq 0$, this is the same as: "if $z = 0$ then $x = 0$"
- Model logical condition "if $x > 0$ then $z = 1$" as:

$$\boxed{x \leq Mz}$$

  where M is *any* upper bound on the value x can take at the optimal solution: $x_{\text{opt}} \leq M$.
- This is called the "Big M method"
- Choose M using constraints

Recap: Fixed Costs

- Producing x has a fixed cost if the cost has the form:

$$\text{cost} = \begin{cases} f + cx & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

- Define a binary variable $z \in \{0, 1\}$ where:

$$z = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

- The constraint becomes: $x \leq Mz$
  where M is any upper bound of x.
- The cost becomes: $fz + cx$
- Small M's are usually better!

Variable Lower Bounds

- $x = 0$ or $\ell \leq x \leq u$ for $0 < \ell < u$
- Binary variable $z \in \{0, 1\}$ to distinguish the two cases
  - $x > 0 \Rightarrow z = 1 \quad \Leftrightarrow \quad x \leq uz$
  - $z = 1 \Rightarrow \ell \leq x \leq u \quad \Leftrightarrow \quad \ell z \leq x \leq uz$

Homeworks

HW 3

- 3.1: Demonstration of MFMC. Obtaining dual solution from Primal using Complementary Slackness
- 3.2: Regression, Polynomial Fits, and inducing sparsity with $\|\cdot\|_1$ regularizer
- 3.3 Weighted approch to multiobjectuve optimization and Pareto Curve
- 3.4: Quadratic functions—non-PSD matrices not solvable with solvers designed for convex optimization. Can shift spectrum of a matrix by adding constant to diagonal
- Quadratic programming for total energy. Optimization for control of physical systems

Homeworks

HW4

- 4.1: Portfolio Optimization, Expected Value Variance and Std Deviation of return, Holdings balance, Modeling 3/2 rational power with second order cone constraints.
- 4.2: Semidefinite Programming. Can have optimization problems with matrices of variables where matrix is constrained to be a PSD Cone. Doing absolute value trick (splitting) even for matrix of variables
- 4.3: Simple knapsack problem. Julia looping and plotting
- 4.4: Variable lower and upper (fixed-charge) constraints
- 4.5: Fixed charge (and inventory)

Give Yourself a Hand!

We learned a lot of stuff!