# CVXPY x NASA Course 2024

Philipp Schiele Steven Diamond Parth Nobel Akshay Agrawal

June 17, 2024

CVXPY

# Homework Review

# Homework

1. work on Problems 1 - 5 from the notebook (`https://marimo.app/l/3s4nd6`)
2. analyze the following expressions as either DCP-compatible or not. Check your work with `https://dcp.stanford.edu/analyzer`
   2.1 `sqrt(1 + 4 * square(x) + 16 * square(y))`
   2.2 `min(x, log(y)) - max(y, z)`
   2.3 `log(exp(2 * x + 3) + exp(4 * y + 5))`
3. play the DCP quiz at `https://dcp.stanford.edu/quiz`
4. bonus problem 1: figure out how to write the non-DCP functions from q2 as DCP.
5. bonus problem 2:
   *Fuel use as function of distance and speed.* A vehicle uses fuel at a rate $f(s)$, which is a function of the vehicle speed $s$. We assume that $f : \mathbf{R} \rightarrow \mathbf{R}$ is a positive increasing convex function, with $\mathbf{dom}\, f = \mathbf{R}_+$. The physical units of $s$ are m/s (meters per second), and the physical units of $f(s)$ are kg/s (kilograms per second). Let $g(d, t)$ be the total fuel used (in kg) when the vehicle moves a distance $d \geq 0$ (in meters) in time $t > 0$ (in seconds) at a constant speed. Write g in DCP form. *Hint:* Check out the "perspective" atom.

# Q1 Solutions

Solution: DCP analysis

`https://marimo.app/l/bi9huq`

# Q4 and Q5 Solutions

To the board!

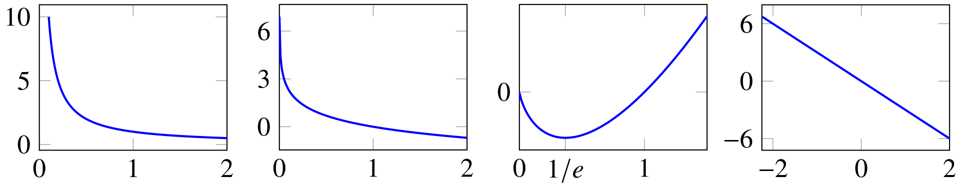
# Key ideas from last time

# Convex functions

$f$ is **convex** if for $\theta \in [0, 1]$,

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

*i.e.*, $f$ have nonnegative (upward) curvature




$f(x) = 1/x$ $\quad$ $f(x) = -\log x$ $\quad$ $f(x) = x \log x$ $\quad$ $f(x) = -3x$

## Examples on $\mathbf{R}$

convex functions:

- affine: $ax + b$ on $\mathbf{R}$, for any $a, b \in \mathbf{R}$
- exponential: $e^{ax}$, for any $a \in \mathbf{R}$
- powers: $x^\alpha$ on $\mathbf{R}_{++}$, for $\alpha \geq 1$ or $\alpha \leq 0$
- powers of absolute value: $|x|^p$ on $\mathbf{R}$, for $p \geq 1$
- positive part (relu): $\max\{0, x\}$

concave functions:

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

# Examples on $\mathbf{R}^n$

convex functions:

- affine functions: $f(x) = a^T x + b$
- any norm, *e.g.*, the $\ell_p$ norms
  - $\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$ for $p \geq 1$
  - $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$
- sum of squares: $\|x\|_2^2 = x_1^2 + \cdots + x_n^2$
- max function: $\max(x) = \max\{x_1, x_2, \ldots, x_n\}$
- softmax or log-sum-exp function: $\log(\exp x_1 + \cdots + \exp x_n)$

# One Rule to Rule Them All

$h(f_1(x), \ldots, f_k(x))$ is convex when $h$ is convex and for each $i$

- $h$ is increasing in argument $i$, and $f_i$ is convex, or
- $h$ is decreasing in argument $i$, and $f_i$ is concave, or
- $f_i$ is affine

There's a similar rule for concave compositions
(just swap convex and concave above).

# DCP Problems

# Convex optimization

convex optimization problem:

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

- variable $x \in \mathbf{R}^n$
- equality constraints are linear
- $f_i$ are **convex**

# 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

# Second-order cone programming

$$
\begin{array}{ll}
\text{minimize} & f^T x \\
\text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m \\
& Fx = g
\end{array}
$$

($A_i \in \mathbf{R}^{n_i \times n}$, $F \in \mathbf{R}^{p \times n}$)

- inequalities are called second-order cone (SOC) constraints:

$$(A_i x + b_i, c_i^T x + d_i) \in \text{second-order cone in } \mathbf{R}^{n_i+1}$$

- for $n_i = 0$, reduces to an LP
- more general LP, QP

# Disciplined convex program

- specify objective as
  - minimize {scalar convex expression}, or
  - maximize {scalar concave expression}
- specify constraints as
  - {convex expression} <= {concave expression} or
  - {concave expression} >= {convex expression} or
  - {affine expression} == {affine expression}
- curvature of expressions are DCP certified, *i.e.*, follow composition rule
- DCP-compliant problems can be automatically transformed to standard forms, then solved

# CVXPY example

**math**:

$$
\begin{array}{ll}
\text{minimize} & \|x\|_1 \\
\text{subject to} & Ax = b \\
& \|x\|_\infty \leq 1
\end{array}
$$

- $x$ is the variable
- $A$, $b$ are given

**CVXPY code**:

```
import cvxpy as cp

A, b = ...

x = cp.Variable(n)
obj = cp.norm(x, 1)
constr = [
  A @ x == b,
  cp.norm(x, 'inf') <= 1,
]
prob = cp.Problem(cp.Minimize(obj), constr)
prob.solve()
```

# How CVXPY works

- starts with your optimization problem $\mathcal{P}_1$
- finds a sequence of equivalent problems $\mathcal{P}_2, \ldots, \mathcal{P}_N$
- final problem $\mathcal{P}_N$ matches a standard form (*e.g.*, LP, QP, SOCP, or SDP)
- calls a specialized solver on $\mathcal{P}_N$
- retrieves solution of original problem by reversing the transformations

$$\mathcal{P}_1 \iff \mathcal{P}_2 \iff \cdots \iff \mathcal{P}_{N-1} \iff \mathcal{P}_N$$

your problem ($\mathcal{P}_1$) — standard problem ($\mathcal{P}_N$)

# Exercise: Write a CVXPY program

Try and write a program that solves

$$\begin{array}{ll} \text{minimize} & \|x - \mathbf{1}\|_2^2 \\ \text{subject to} & \|x\|_1 \leq 1 \end{array}$$

Hint: you will need `sum_squares` and `norm(·, 1)`

# Outline

Transforming problems

# Change of variables

- $\phi : \mathbf{R}^n \to \mathbf{R}^n$ is one-to-one and the optimum is a possible output of $\phi$
- consider (possibly non-convex) problem

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

- change variables to $z$ with $x = \phi(z)$
- can solve equivalent problem

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(z) \\ \text{subject to} & \tilde{f}_i(z) \leq 0, \quad i = 1, \dots, m \\ & \tilde{h}_i(z) = 0, \quad i = 1, \dots, p \end{array}$$

  where $\tilde{f}_i(z) = f_i(\phi(z))$ and $\tilde{h}_i(z) = h_i(\phi(z))$
- recover original optimal point as $x^\star = \phi(z^\star)$

# Example

- **non-convex** problem

$$\begin{array}{ll} \text{minimize} & x_1/x_2 + x_3/x_1 \\ \text{subject to} & x_2/x_3 + x_1 \leq 1 \end{array}$$

  with implicit constraint $x \succ 0$

- change variables using $x = \phi(z) = \exp z$ to get

$$\begin{array}{ll} \text{minimize} & \exp(z_1 - z_2) + \exp(z_3 - z_1) \\ \text{subject to} & \exp(z_2 - z_3) + \exp(z_1) \leq 1 \end{array}$$

  which is **convex**

# Transformation of objective and constraint functions

suppose

- $\phi_0$ is monotone increasing
- $\psi_i(u) \leq 0$ if and only if $u \leq 0$, $i = 1, \ldots, m$
- $\varphi_i(u) = 0$ if and only if $u = 0$, $i = 1, \ldots, p$

standard form optimization problem is equivalent to

$$\begin{array}{lll} \text{minimize} & \phi_0(f_0(x)) & \\ \text{subject to} & \psi_i(f_i(x)) \leq 0, & i = 1, \ldots, m \\ & \varphi_i(h_i(x)) = 0, & i = 1, \ldots, p \end{array}$$

example: minimizing $\|Ax - b\|$ is equivalent to minimizing $\|Ax - b\|^2$

# Converting maximization to minimization

- suppose $\phi_0$ is monotone decreasing
- the maximization problem

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

  is equivalent to the minimization problem

$$\begin{array}{lll} \text{minimize} & \phi_0(f_0(x)) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p \end{array}$$

- **examples**:
  - $\phi_0(u) = -u$ transforms maximizing a concave function to minimizing a convex function
  - $\phi_0(u) = 1/u$ transforms maximizing a concave positive function to minimizing a convex function

# Eliminating equality constraints

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

is equivalent to

$$\begin{array}{ll} \text{minimize (over } z\text{)} & f_0(Fz + x_0) \\ \text{subject to} & f_i(Fz + x_0) \leq 0, \quad i = 1, \dots, m \end{array}$$

where $F$ and $x_0$ are such that $Ax = b \iff x = Fz + x_0$ for some $z$

# Introducing equality constraints

$$\begin{array}{ll} \text{minimize} & f_0(A_0x + b_0) \\ \text{subject to} & f_i(A_ix + b_i) \leq 0, \quad i = 1, \ldots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \text{minimize (over } x, y_i) & f_0(y_0) \\ \text{subject to} & f_i(y_i) \leq 0, \quad i = 1, \ldots, m \\ & y_i = A_ix + b_i, \quad i = 0, 1, \ldots, m \end{array}$$

# Introducing slack variables for linear inequalities

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

is equivalent to

$$
\begin{array}{ll}
\text{minimize (over } x, s\text{)} & f_0(x) \\
\text{subject to} & a_i^T x + s_i = b_i, \quad i = 1, \ldots, m \\
& s_i \geq 0, \quad i = 1, \ldots m
\end{array}
$$

# Epigraph form

standard form convex problem is equivalent to

$$\begin{array}{ll} \text{minimize (over } x, t) & t \\ \text{subject to} & f_0(x) - t \leq 0 \\ & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

# Exercise and Homework

## Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

## Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

## Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

## Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot “short” stocks and have a “long”-only portfolio. Lastly, we don’t want to end up over-invested in a single asset, so, let’s ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot “short” stocks and have a “long”-only portfolio. Lastly, we don’t want to end up over-invested in a single asset, so, let’s ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Q1: Portfolio Optimization

In this problem, we are going to discuss how to use a model of a market to maximize profit. Portfolio optimization is the backbone of quantitative finance and energy allocation.

Imagine we have a budget $B$ (dollars). We can purchase $n$ different assets (e.g. stocks, bonds) with current prices $p$ (dollars per unit), expected returns $\mu$ (dollars per unit), and covariance $\Sigma$ (dollars squared per unit squared). Your goal is to ensure the standard deviation of your portfolio to be less than $S$ (dollars). We are only buying assets, not borrowing them, so we cannot "short" stocks and have a "long"-only portfolio. Lastly, we don't want to end up over-invested in a single asset, so, let's ensure we never put more than $5\%$ of our budget into a single asset.
Using the data in the notebook, `https://marimo.io/p/@cvxpy/port-opt`, find the portfolio that maximizes expected returns.

# Rest of the homework is online!