

# CME 307 / MS&E 311: Optimization

## Introduction

Professor Udell

Management Science and Engineering  
Stanford

April 10, 2023

## Announcements

- ▶ lectures slides on website:  
<https://stanford-cme-307.github.io/website-2023/calendar>
- ▶ recordings posted on canvas
- ▶ office hours posted on website:  
<https://stanford-cme-307.github.io/website-2023/staff>

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

Conic optimization

Integer programming

Convex optimization

Who's here?

## What is an optimization problem?

optimization problem: nonlinear form

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m_1 \\ & h_i(x) = 0, \quad i = 1, \dots, m_2 \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

- ▶ objective  $f_0$
- ▶ inequality constraints  $f_i$
- ▶ equality constraints  $h_i$

## Modularity in optimization

how to optimize:

1. model problem as a mathematical optimization problem
2. identify the properties of the problem
3. use an appropriate solver (or write a new one)

... and iterate:

- ▶ approximate the problem to make it easier
- ▶ solve a sequence of approximated problems that converge to solve the original problem
- ▶ or initialize (“warm-start”) a solver for the original problem with a solution to the approximated problem

## How will you use optimization?

- ▶ model problem to match solver capabilities
- ▶ use solvers as building blocks in larger algorithms
- ▶ write your own solver for a new problem class
- ▶ write a specialized solver for a particular problem

## What will you need to know to use (or develop) optimization?

- ▶ convexity
- ▶ geometry
- ▶ numerical stability
- ▶ complexity
- ▶ reliability
- ▶ stopping conditions

## Optimization courses

compared to

- ▶ EE364a: more algos, less modeling
- ▶ EE364b: more nonconvex, different mix of algos
- ▶ ...

my goal:  $< 30\%$  overlap with either



# Topics

let's look at topics on google sheets:

[https://docs.google.com/spreadsheets/d/1PXv\\_sFkhz5jNAA765kgHanSPoILm2fHzzRNuOOJzpHM](https://docs.google.com/spreadsheets/d/1PXv_sFkhz5jNAA765kgHanSPoILm2fHzzRNuOOJzpHM)

- ▶ what other topics should we add?

# Outline

What topics and why?

**Grading**

Quadratic optimization

Nonlinear optimization

Conic optimization

Integer programming

Convex optimization

Who's here?

## Grading

let's look at the course website:

<https://stanford-cme-307.github.io/website-2023/>

- ▶ for PhD students, the grading is designed to help (not distract from) your research
- ▶ for UG/Masters, this course may push you hard. . .

# Optimization problems

important optimization problem classes:

- ▶ quadratic
- ▶ unconstrained
- ▶ finite-sum
- ▶ linear
- ▶ conic
- ▶ convex
- ▶ nonlinear (with linear or nonlinear constraints)
- ▶ mixed-integer linear programs
- ▶ black-box with (0, 1, or 2)-order oracle

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

Conic optimization

Integer programming

Convex optimization

Who's here?

## Quadratic optimization

a **quadratic optimization** problem is written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \|Ax - b\|^2 := f_0(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where

- ▶  $A \in \mathbf{R}^{m \times n}$ : matrix
- ▶  $b \in \mathbf{R}^m$ : vector

how to solve?

## Quadratic optimization

a **quadratic optimization** problem is written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \|Ax - b\|^2 := f_0(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where

- ▶  $A \in \mathbf{R}^{m \times n}$ : matrix
- ▶  $b \in \mathbf{R}^m$ : vector

how to solve? take gradient and set to 0:

$$\nabla f_0(x) = A^T(Ax - b) = 0$$

$\implies$  linear system solvers also solve quadratic optimization problems

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions



## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if  $m = n$  and  $A$  is full rank

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if  $m = n$  and  $A$  is full rank
- ▶ if  $m < n$  and  $A$  is full rank
  - ▶ solution set is a hyperplane of dimension

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if  $m = n$  and  $A$  is full rank
- ▶ if  $m < n$  and  $A$  is full rank
  - ▶ solution set is a hyperplane of dimension  $n - m$
  - ▶ null space of  $A$ , **nullspace**( $A$ ), is a hyperplane of dimension

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if  $m = n$  and  $A$  is full rank
- ▶ if  $m < n$  and  $A$  is full rank
  - ▶ solution set is a hyperplane of dimension  $n - m$
  - ▶ null space of  $A$ , **nullspace**( $A$ ), is a hyperplane of dimension  $n - m$
  - ▶ solution set is  $\{x : Ax = b\} = \{x_0 + Vz\}$  where columns of  $V \in \mathbf{R}^{n \times n-m}$  span **nullspace**( $A$ )

## Linear algebra review

matrix  $A \in \mathbf{R}^{m \times n}$

- ▶ check matrix calculus results by checking dimensions
- ▶ normal equations  $A^T A x = A^T b$
- ▶ solution to  $Ax = b$  is unique if  $m = n$  and  $A$  is full rank
- ▶ if  $m < n$  and  $A$  is full rank
  - ▶ solution set is a hyperplane of dimension  $n - m$
  - ▶ null space of  $A$ , **nullspace**( $A$ ), is a hyperplane of dimension  $n - m$
  - ▶ solution set is  $\{x : Ax = b\} = \{x_0 + Vz\}$  where columns of  $V \in \mathbf{R}^{n \times n-m}$  span **nullspace**( $A$ )
- ▶  $A^T A$  is symmetric positive semidefinite (proof on board)

## Symmetric positive semidefinite matrices

### Definition

a symmetric matrix  $Q \in \mathbf{R}^{n \times n}$  is **positive semidefinite** (psd) if  $x^T Q x \geq 0$  for all  $x \in \mathbf{R}^n$ .

these matrices are so important that there are many ways to write them! for  $Q \in \mathbf{R}^{n \times n}$ ,

$$Q \in \mathbf{S}_+^n \iff Q \succeq 0 \iff Q = Q^T, \lambda_{\min}(Q) \geq 0$$



## Symmetric positive semidefinite matrices

### Definition

a symmetric matrix  $Q \in \mathbf{R}^{n \times n}$  is **positive semidefinite** (psd) if  $x^T Q x \geq 0$  for all  $x \in \mathbf{R}^n$ .

these matrices are so important that there are many ways to write them! for  $Q \in \mathbf{R}^{n \times n}$ ,

$$Q \in \mathbf{S}_+^n \iff Q \succeq 0 \iff Q = Q^T, \lambda_{\min}(Q) \geq 0$$

$Q \in \mathbf{S}_+^n$  is **symmetric positive definite** (spd) ( $Q \succ 0$ ) if  $x^T Q x > 0$  for all  $x \in \mathbf{R}^n$ .

## Symmetric positive semidefinite matrices

### Definition

a symmetric matrix  $Q \in \mathbf{R}^{n \times n}$  is **positive semidefinite** (psd) if  $x^T Q x \geq 0$  for all  $x \in \mathbf{R}^n$ .

these matrices are so important that there are many ways to write them! for  $Q \in \mathbf{R}^{n \times n}$ ,

$$Q \in \mathbf{S}_+^n \iff Q \succeq 0 \iff Q = Q^T, \lambda_{\min}(Q) \geq 0$$

$Q \in \mathbf{S}_+^n$  is **symmetric positive definite** (spd) ( $Q \succ 0$ ) if  $x^T Q x > 0$  for all  $x \in \mathbf{R}^n$ .

why care about psd matrices  $Q$ ?

- ▶ least-squares objective has a psd  $Q = A^T A$
- ▶ level sets of  $x^T Q x$  are (bounded) ellipsoids
- ▶ the quadratic form  $x^T Q x$  is a metric iff  $Q \succ 0$
- ▶ eigenvalue decomp and svd coincide for psd matrices

## Quadratic program

a **quadratic program** is written as

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} & Ax = b \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where

- ▶  $Q \in \mathbf{R}^{n \times n}$ : symmetric positive semidefinite matrix
- ▶  $c \in \mathbf{R}^n$ : vector

how to solve?

## Quadratic program

a **quadratic program** is written as

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} & Ax = b \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where

- ▶  $Q \in \mathbf{R}^{n \times n}$ : symmetric positive semidefinite matrix
- ▶  $c \in \mathbf{R}^n$ : vector

how to solve? reduce to quadratic optimization problem:

- ▶ (explicit) form solution set  
 $\{x : Ax = b\} = \{x_0 + Vz \mid z \in \mathbf{R}^{n-m}\}$  by computing a solution  $Ax_0 = b$  and a basis  $V$  for the null space of  $A$
- ▶ (implicit) use duality to recast problem as larger linear (KKT) system

## Quadratic program: application

Markowitz portfolio optimization problem:

$$\begin{array}{ll}\text{minimize} & \gamma x^T \Sigma x - \mu^T x \\ \text{subject to} & \sum_i x_i = 1 \\ & Ax = 0 \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where

- ▶  $\Sigma \in \mathbf{R}^{n \times n}$ : asset covariance matrix
- ▶  $\mu \in \mathbf{R}^n$ : asset return vector
- ▶  $\gamma \in \mathbf{R}$ : risk aversion parameter
- ▶ rows of  $A \in \mathbf{R}^{m \times n}$  correspond to other portfolios
  - ▶ ensures new portfolio is independent, e.g., of market returns

# Outline

What topics and why?

Grading

Quadratic optimization

**Nonlinear optimization**

Conic optimization

Integer programming

Convex optimization

Who's here?

## Unconstrained smooth optimization

for  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  ctly differentiable,

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

how to solve?

## Unconstrained smooth optimization

for  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  ctly differentiable,

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

how to solve? approximate as a quadratic problem

$$f(x) \approx f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T H(x_0)(x - x_0)$$

and find solution  $x_{\text{quad}}$  to the quadratic problem.

then set  $x_0 \leftarrow x_{\text{quad}}$  and repeat.



## Finite sum

**finite sum** optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m f_i(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

**key fact:** can approximate gradient using gradient on **minibatch**  $S \subseteq \{1, \dots, m\}$ :

$$\nabla f(x) \approx \frac{1}{|S|} \sum_{i \in S} \nabla f_i(x)$$

examples:

- ▶ statistical learning (logistic regression, SVM)
- ▶ deep learning

## Background: classification

**classification** problem:  $m$  data points

- ▶ feature vector  $a_i \mathbf{R}^n$ ,  $i = 1, \dots, m$
- ▶ label  $b_i \in \{-1, 1\}$ ,  $i = 1, \dots, m$

choose decision boundary  $a^T x = 0$  to separate data points into two classes

- ▶  $a^T x > 0 \implies$  predict class 1
- ▶  $a^T x < 0 \implies$  predict class -1

classification is correct if  $b_i a^T x > 0$

## Background: classification

**classification** problem:  $m$  data points

- ▶ feature vector  $a_i \mathbf{R}^n$ ,  $i = 1, \dots, m$
- ▶ label  $b_i \in \{-1, 1\}$ ,  $i = 1, \dots, m$

choose decision boundary  $a^T x = 0$  to separate data points into two classes

- ▶  $a^T x > 0 \implies$  predict class 1
- ▶  $a^T x < 0 \implies$  predict class -1

classification is correct if  $b_i a^T x > 0$

- ▶ projective transformation transforms affine boundary to linear boundary
- ▶ classification is invariant to scalar multiplication of  $x$

## Logistic regression

(regularized) **logistic regression** minimizes the **finite sum**

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \log(1 + \exp(-b_i a_i^T x)) + r(x) \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where

- ▶  $b_i \in \{-1, 1\}$ ,  $a_i \in \mathbf{R}^n$
- ▶  $r : \mathbf{R}^n \rightarrow \mathbf{R}$  is a **regularizer**, e.g.,  $\|x\|^2$  or  $\|x\|_1$

## Support vector machine

**support vector machine (SVM)** minimizes the **finite sum**

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \max(0, 1 - b_i a_i^T x) + \gamma \|x\|^2 \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where  $b_i \in \{-1, 1\}$  and  $a_i \in \mathbf{R}^n$ .

## Support vector machine

**support vector machine (SVM)** minimizes the **finite sum**

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \max(0, 1 - b_i a_i^T x) + \gamma \|x\|^2 \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where  $b_i \in \{-1, 1\}$  and  $a_i \in \mathbf{R}^n$ . not differentiable!

## Support vector machine

**support vector machine (SVM)** minimizes the **finite sum**

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \max(0, 1 - b_i a_i^T x) + \gamma \|x\|^2 \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where  $b_i \in \{-1, 1\}$  and  $a_i \in \mathbf{R}^n$ . not differentiable!

how to solve?

## Support vector machine

**support vector machine (SVM)** minimizes the **finite sum**

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \max(0, 1 - b_i a_i^T x) + \gamma \|x\|^2 \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$

where  $b_i \in \{-1, 1\}$  and  $a_i \in \mathbf{R}^n$ . not differentiable!

how to solve?

- ▶ use **subgradient** method
- ▶ transform to **conic form**
- ▶ solve **dual** problem instead
- ▶ **smooth** the objective



## Nonlinear optimization

optimization problem: nonlinear form

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \dots, m_1 \\ & h(x) = 0 \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

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

special case: **unconstrained optimization**

## Example: process control

You are the process engineer for a desalination plant that produces drinking water. The plant has a variety of knobs, collected in vector  $x$ , that you can turn to control the process. These control, e.g., how much water is pumped into the plant, how much pressure is used to force the water through filters, and how much of each chemical is added to the water.

- ▶  $f_0(x)$ : cost of water produced
- ▶  $f_i(x)$ : level of each measured impurity in the water
- ▶  $b_i$ : maximum allowable level of each impurity

Given a setting of the knobs, you can observe the cost of water produced and the levels of impurities.

**What is the optimal setting of the knobs?**

## Oracles

an optimization **oracle** is your interface for accessing the problem data:

e.g., an oracle for  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  can evaluate for any  $x \in \mathbf{R}^n$ :

- ▶ **zero-order:**  $f_0(x)$
- ▶ **first-order:**  $f_0(x)$  and  $\nabla f_0(x)$
- ▶ **second-order:**  $f_0(x)$ ,  $\nabla f_0(x)$ , and  $\nabla^2 f_0(x)$

why oracles?

- ▶ can optimize real systems based on observed output (not just models)
- ▶ can use and extend old or complex but trusted code (e.g., NASA, PDE simulations, ...)
- ▶ can prove lower bounds on the oracle complexity of a problem class

source: Nesterov 2004 "Introductory Lectures on Convex Optimization"

## Nonlinear optimization: how to solve?

depends on the oracle:

- ▶ first- or second-order: approximate by a sequence of quadratic problems
- ▶ zero-order: harder, lots of methods
  - ▶ simulated annealing
  - ▶ Bayesian optimization
  - ▶ pseudo-higher-order methods, e.g., compute approximate gradient

## Solution of an optimization problem

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

for  $f : \mathcal{D} \rightarrow \mathbf{R}$ .  $x^*$  is a

- ▶ **local minimizer** if there is a neighborhood  $\mathcal{N}$  around  $x^*$  so that  $f(x) \geq f(x^*)$  for all  $x \in \mathcal{N}$ .
- ▶ **global minimizer** if  $f(x) \geq f(x^*)$  for all  $x \in \mathcal{D}$ .
- ▶ **strict local minimizer** if there is a neighborhood  $\mathcal{N}$  around  $x^*$  so that  $f(x) > f(x^*)$  for all  $x \in \mathcal{N}$ .
- ▶ **isolated local minimizer** if the neighborhood  $\mathcal{N}$  contains no other local minimizers.
- ▶ **unique minimizer** if it is the only global minimizer.

## Solution of an optimization problem

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

for  $f : \mathcal{D} \rightarrow \mathbf{R}$ .  $x^*$  is a

- ▶ **local minimizer** if there is a neighborhood  $\mathcal{N}$  around  $x^*$  so that  $f(x) \geq f(x^*)$  for all  $x \in \mathcal{N}$ .
- ▶ **global minimizer** if  $f(x) \geq f(x^*)$  for all  $x \in \mathcal{D}$ .
- ▶ **strict local minimizer** if there is a neighborhood  $\mathcal{N}$  around  $x^*$  so that  $f(x) > f(x^*)$  for all  $x \in \mathcal{N}$ .
- ▶ **isolated local minimizer** if the neighborhood  $\mathcal{N}$  contains no other local minimizers.
- ▶ **unique minimizer** if it is the only global minimizer.

pictures!

## First order optimality condition

### Theorem

*If  $x^* \in \mathbf{R}^n$  is a local minimizer of a differentiable function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$ , then  $\nabla f(x^*) = 0$ .*

## First order optimality condition

### Theorem

*If  $x^* \in \mathbf{R}^n$  is a local minimizer of a differentiable function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$ , then  $\nabla f(x^*) = 0$ .*

**proof:** suppose by contradiction that  $\nabla f(x^*) \neq 0$ . consider points of the form  $x_\alpha = x^* - \alpha \nabla f(x^*)$  for  $\alpha > 0$ . by definition of the gradient,

$$\lim_{\alpha \rightarrow 0} \frac{f(x_\alpha) - f(x^*)}{\alpha} = -\nabla f(x^*)^\top \nabla f(x^*) = -\|\nabla f(x^*)\|^2 < 0$$

so for any sufficiently small  $\alpha > 0$ , we have  $f(x_\alpha) < f(x^*)$ , which contradicts the fact that  $x^*$  is a local minimizer.



## Second order optimality condition

### Theorem

*If  $x^* \in \mathbf{R}^n$  is a local minimizer of a twice differentiable function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$ , then  $\nabla^2 f(x^*) \succeq 0$ .*

## Second order optimality condition

### Theorem

*If  $x^* \in \mathbf{R}^n$  is a local minimizer of a twice differentiable function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$ , then  $\nabla^2 f(x^*) \succeq 0$ .*

**proof:** similar to the previous proof. use the fact that the second order approximation

$$f(x_\alpha) \approx f(x^*) + \nabla f(x^*)^\top (x_\alpha - x^*) + \frac{1}{2} (x_\alpha - x^*)^\top \nabla^2 f(x^*) (x_\alpha - x^*)$$

is accurate locally to show a contradiction unless  $\nabla^2 f(x^*) \succeq 0$ : if not, there is a direction  $v$  such that  $v^\top \nabla^2 f(x^*) v < 0$ . then  $f(x + \alpha v) < f(x^*)$  for  $\alpha$  arbitrarily small, which contradicts the fact that  $x^*$  is a local minimizer.

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

**Conic optimization**

Integer programming

Convex optimization

Who's here?

## Linear program

a **linear program** is written as

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & b - Ax \geq 0 \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where

- ▶  $A \in \mathbf{R}^{m \times n}$ : matrix
- ▶  $b \in \mathbf{R}^m$ : vector
- ▶  $c \in \mathbf{R}^n$ : vector

how to solve?

## Linear program

a **linear program** is written as

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & b - Ax \geq 0 \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where

- ▶  $A \in \mathbf{R}^{m \times n}$ : matrix
- ▶  $b \in \mathbf{R}^m$ : vector
- ▶  $c \in \mathbf{R}^n$ : vector

how to solve?

- ▶ use the simplex method
- ▶ use a conic solver

## Conic form

**conic form** optimization problem generalizes LP:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & b - Ax \in \mathcal{K},\end{array}$$

where  $\mathcal{K}$  is a **convex cone**:

$$x \in \mathcal{K} \iff rx \in \mathcal{K} \text{ for any } r > 0.$$

examples:

- ▶ zero cone  $\mathcal{K}_0 = \{0\}$
- ▶ positive orthant  $\mathcal{K}_+ = \{x : x_i \geq 0, i = 1, \dots, n\}$
- ▶ second order cone  $\mathcal{K}_{\text{SOC}} = \{(x, t) : \|x\|_2 \leq t\}$
- ▶ positive semidefinite (PSD) cone  
 $\mathcal{K}_{\text{SDP}} = \{X : X = X^T, v^T X v \geq 0, \forall v \in \mathbf{R}^n\}$
- ▶ cartesian products of cones

## Conic form: how to solve?

Morally, conic problems are solved by reducing to a nonlinear optimization problem

- ▶ barrier methods (e.g., interior point methods)
  - ▶ add a barrier term to the objective that goes to infinity when constraints are violated
- ▶ penalty methods (e.g., augmented Lagrangian methods, ADMM, ... )
  - ▶ add a penalty term to the objective that depends on a dual variable
  - ▶ adjust the dual variable to enforce constraints

## Conic form example: nonnegative least squares

$$\begin{array}{ll}\text{minimize} & \|Ax - b\| \\ \text{subject to} & x \geq 0\end{array}$$



$$\begin{array}{ll}\text{minimize} & t \\ \text{subject to} & x \in \mathcal{K}_+ \\ & (Ax - b, t) \in \mathcal{K}_{\text{soc}}\end{array}$$



## Conic form example: SVM

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \max(0, 1 - b_i a_i^T x) + \|x\|^2 \\ \text{variable} & x \in \mathbf{R}^n \end{array}$$
$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & \sum_i s_i + t \\ \text{subject to} & s \geq \mathbf{diag}(b)Ax - 1 \\ & s \geq 0 \\ & t \geq \|x\|^2 \end{array}$$
$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & \sum_i s_i + t \\ \text{subject to} & s - \mathbf{diag}(b)Ax + 1 \in \mathcal{K}_+ \\ & s \in \mathcal{K}_+ \\ & [t \ x; x^T \ I_n] \in \mathcal{K}_{\text{SDP}} \end{array}$$

## Schur complement

Consider the block matrix

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}.$$

- ▶ the **Schur complement** of  $A$  in  $X$  is  $C - B^T A^{-1} B$ .
- ▶  $X \succeq 0$  if and only if  $A \succeq 0$  and  $C - B^T A^{-1} B \succeq 0$ .  
(proof by partial minimization of quadratic form  $(u, v)^T X (u, v)$  over  $u \in \mathbf{R}^m$  for fixed  $v \in \mathbf{R}^n$ )

## Conic form example: semidefinite programming

$$\begin{array}{ll}\text{minimize} & \lambda_{\max}(X) + y^T X^{-1} y \\ \text{subject to} & X \succeq 0\end{array}$$



## Conic form example: semidefinite programming

$$\begin{array}{ll}\text{minimize} & \lambda_{\max}(X) + y^T X^{-1} y \\ \text{subject to} & X \succeq 0\end{array}$$

$$\Updownarrow$$

$$\begin{array}{ll}\text{minimize} & t_1 + t_2 \\ \text{subject to} & t_1 I - X \in \mathcal{K}_{\text{SDP}} \\ & \begin{bmatrix} t_2 & y^T \\ y & X \end{bmatrix} \in \mathcal{K}_{\text{SDP}}\end{array}$$

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

Conic optimization

**Integer programming**

Convex optimization

Who's here?

## Integer programming

**integer linear programming** generalizes linear programming:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & b - Ax \geq 0 \\ \text{variable} & x \in \mathbf{Z}^n\end{array}$$

variants:

- ▶ **mixed integer linear programming (MILP):**  
 $x \in \mathbf{Z}^{n-m} \cup \mathbf{R}^m$
- ▶ **mixed integer nonlinear programming (MINLP):**  
 $x \in \mathbf{Z}^{n-m} \cup \mathbf{R}^m$  and nonlinear objective or constraints

how to solve?

# Integer programming

**integer linear programming** generalizes linear programming:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & b - Ax \geq 0 \\ \text{variable} & x \in \mathbf{Z}^n\end{array}$$

variants:

- ▶ **mixed integer linear programming (MILP):**  
 $x \in \mathbf{Z}^{n-m} \cup \mathbf{R}^m$
- ▶ **mixed integer nonlinear programming (MINLP):**  
 $x \in \mathbf{Z}^{n-m} \cup \mathbf{R}^m$  and nonlinear objective or constraints

how to solve?

- ▶ use Gurobi, CPLEX, ...
- ▶ branch and bound and cut (*i.e.*, a sequence of LPs)
- ▶ use duality to decompose into a sequence of simpler LPs

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

Conic optimization

Integer programming

**Convex optimization**

Who's here?



## Convex sets

### Definition

A set  $S \subseteq \mathbf{R}^n$  is convex if it contains every chord: for all  $\theta \in [0, 1]$ ,  $w, v \in S$ ,

$$\theta w + (1 - \theta)v \in S$$

## Convex sets

### Definition

A set  $S \subseteq \mathbf{R}^n$  is convex if it contains every chord: for all  $\theta \in [0, 1]$ ,  $w, v \in S$ ,

$$\theta w + (1 - \theta)v \in S$$

**Q:** Which of these are convex?

ellipsoid, half moon

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

- **Chords.** it never lies above its chord:  $\forall \theta \in [0, 1], w, v \in \mathbf{R}^n$

$$f(\theta w + (1 - \theta)v) \leq \theta f(w) + (1 - \theta)f(v)$$

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

- ▶ **Chords.** it never lies above its chord:  $\forall \theta \in [0, 1], w, v \in \mathbf{R}^n$

$$f(\theta w + (1 - \theta)v) \leq \theta f(w) + (1 - \theta)f(v)$$

- ▶ **Epigraph.**  $\text{epi}(f) = \{(x, t) : t \geq f(x)\}$  is convex

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

- ▶ **Chords.** it never lies above its chord:  $\forall \theta \in [0, 1], w, v \in \mathbf{R}^n$

$$f(\theta w + (1 - \theta)v) \leq \theta f(w) + (1 - \theta)f(v)$$

- ▶ **Epigraph.**  $\text{epi}(f) = \{(x, t) : t \geq f(x)\}$  is convex
- ▶ **First order condition.** if  $f$  is differentiable,

$$f(v) - f(w) \geq \nabla f(w)^\top (v - w) \quad \forall w, v \in \mathbf{R}^n$$

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

- ▶ **Chords.** it never lies above its chord:  $\forall \theta \in [0, 1], w, v \in \mathbf{R}^n$

$$f(\theta w + (1 - \theta)v) \leq \theta f(w) + (1 - \theta)f(v)$$

- ▶ **Epigraph.**  $\text{epi}(f) = \{(x, t) : t \geq f(x)\}$  is convex
- ▶ **First order condition.** if  $f$  is differentiable,

$$f(v) - f(w) \geq \nabla f(w)^\top (v - w) \quad \forall w, v \in \mathbf{R}^n$$

- ▶ **Second order condition.** If  $f$  is twice differentiable, its Hessian is always psd:

$$\lambda_{\min}(\nabla^2 f(x)) \geq 0 \quad \text{for all } x \in \mathbf{R}^n$$

## Convex functions

a function  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  is convex iff

- ▶ **Chords.** it never lies above its chord:  $\forall \theta \in [0, 1], w, v \in \mathbf{R}^n$

$$f(\theta w + (1 - \theta)v) \leq \theta f(w) + (1 - \theta)f(v)$$

- ▶ **Epigraph.**  $\text{epi}(f) = \{(x, t) : t \geq f(x)\}$  is convex

- ▶ **First order condition.** if  $f$  is differentiable,

$$f(v) - f(w) \geq \nabla f(w)^\top (v - w) \quad \forall w, v \in \mathbf{R}^n$$

- ▶ **Second order condition.** If  $f$  is twice differentiable, its Hessian is always psd:

$$\lambda_{\min}(\nabla^2 f(x)) \geq 0 \quad \text{for all } x \in \mathbf{R}^n$$

**Q:** Which of these are convex?

quadratic, l1, pwl, step, jump, logistic, logistic loss



## Convex optimization

an optimization problem is convex if:

- ▶ **Geometrically:** the feasible set and the epigraph of the objective are convex
- ▶ **NLP:** the objective and inequality constraints are convex functions, and the equality constraints are affine
- ▶ **Conic:** all the cones are convex cones

## Convex optimization

an optimization problem is convex if:

- ▶ **Geometrically:** the feasible set and the epigraph of the objective are convex
- ▶ **NLP:** the objective and inequality constraints are convex functions, and the equality constraints are affine
- ▶ **Conic:** all the cones are convex cones

why convex optimization?

- ▶ relatively complete theory
- ▶ efficient solvers
- ▶ conceptual tools that generalize

duality, stopping conditions, ...

## Convex optimization

an optimization problem is convex if:

- ▶ **Geometrically:** the feasible set and the epigraph of the objective are convex
- ▶ **NLP:** the objective and inequality constraints are convex functions, and the equality constraints are affine
- ▶ **Conic:** all the cones are convex cones

why convex optimization?

- ▶ relatively complete theory
- ▶ efficient solvers
- ▶ conceptual tools that generalize

duality, stopping conditions, ...

- ▶ a function  $f$  is concave if  $-f$  is convex
- ▶ concave maximization results in a **convex** optimization problem

## Local minima are global for convex functions

### Theorem

*If  $x^*$  is a local minimizer of a convex function  $f$ , then  $x^*$  is a global minimizer.*

## Local minima are global for convex functions

### Theorem

*If  $x^*$  is a local minimizer of a convex function  $f$ , then  $x^*$  is a global minimizer.*

**proof:** suppose by contradiction that another point  $x'$  is a global minimizer, with  $f(x') < f(x^*)$ . draw the chord between  $x'$  and  $x^*$ . since the chord lies above  $f$ , every convex combination  $x = \theta x^* + (1 - \theta)x'$  of  $x'$  and  $x^*$  for  $\theta \in (0, 1)$  has a value  $f(x) < f(x^*)$ . this is true even for  $x \rightarrow x^*$ , contradicting our assumption that  $x^*$  is a local minimizer.

## Corollary

### Corollary

*If  $f$  is convex and differentiable and  $\nabla f(x^*) = 0$ , then  $x^*$  is a global minimizer.*

## Corollary

### Corollary

*If  $f$  is convex and differentiable and  $\nabla f(x^*) = 0$ , then  $x^*$  is a global minimizer.*

**Q:** Is a global minimizer of a convex function always unique?

## Corollary

### Corollary

*If  $f$  is convex and differentiable and  $\nabla f(x^*) = 0$ , then  $x^*$  is a global minimizer.*

**Q:** Is a global minimizer of a convex function always unique?

**A:** No. Picture.



## Modern solvers

- ▶ algebraic modeling languages, *e.g.*
  - ▶ JuMP facilitates nonlinear and mixed integer optimization
  - ▶ CVX\* (CVX, CVXPY, Convex.jl, ...) transform a problem into conic form
- ▶ and modern solvers

# Optimization modeling

- ▶ Rocket control
- ▶ Power systems
- ▶ AML

# Outline

What topics and why?

Grading

Quadratic optimization

Nonlinear optimization

Conic optimization

Integer programming

Convex optimization

Who's here?

## Questions for you

respond at [pollev.com/madeleineudell824](https://pollev.com/madeleineudell824)

- ▶ program / major
- ▶ year of program
- ▶ schedule conflicts
- ▶ what other optimization courses have you taken?
- ▶ what optimization techniques do you want to learn?
- ▶ what's your favorite algorithm (or algorithmic trick)