

IE 510 Applied Nonlinear Programming

Lecture 0: Introduction

Ruoyu Sun

Jan 16, 2018

Outline

- 1 Course Introduction
- 2 Optimization Background
- 3 Examples
- 4 Mathematical Review

Organization

Instructor:

Ruoyu Sun (ruoyus at illinois.edu)

Assistant professor, ISE and CSL

209D Transportation Building

Administrative Details:

Lecture time/location: Tu and Th: 2:00pm - 3:20PM, Enigneering Hall 410C1

The instructor could be on travel in 1-2 occasions

Extended lecture time (for about 30min)?

Office hour: Tu: 3:30 - 4:30pm, Transportation Bld. 209D, or by appointment

TA: Lei Fan (leifan2 at illinois.edu)

Office hour: Thur 12:30-1:30pm

Location: TBD

Organization

Textbook

D. Bertsekas, *Nonlinear Programming*, [Main Textbook](#)

Luenberger and Ye. *Linear and nonlinear programming*. , [Reference](#)

Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*
[Reference](#)

Other relevant materials will be distributed and discussed throughout the course

Syllabus

- 1 Can be found on the Box file
- 2 Some of the key points here
- 3 Numerical grade = homework (35 %) + 1 in-class exam (30 %) + class project (30%) + attendance (5%) + bonus points (up to 10%)
(writing also allowed)
- 4 It is encouraged to submit the homework **electronically**, in **pdf**
- 5 Late homework and exam will **not** be accepted or graded, unless approved by the instructor
- 6 Bonus points = bonus problems in homework/exam and/or excellent project

Components of the course

- ① Homework will be assigned regularly, some of them are mathematical, some of them requires programming
- ② Course slides will be distributed (in the Box file) regularly, but not all results/materials will be on the slides
- ③ One mid-term
- ④ Class project report
- ⑤ No final

More on the course

- 1 Lectures will be delivered using both slides and in-class notes
- 2 **Pre-requisite:** The course is mostly self-contained, no explicit requirements on previous courses on optimization
- 3 **Pre-requisite:** basic knowledge about linear algebra and calculus is required
- 4 Helpful knowledge: probability, numerical linear algebra, complexity theory, machine learning
- 5 **Theoretical or practical?** Mixture. Lots of theoretical analysis, but will try to provide insights/discussion/applications whenever possible
- 6 Happy to take any suggestions/comments about the course/teaching

Project

- ① There is a class project (30% of the numerical grade)
- ② One-person, or multiple-people projects
- ③ **Time line:** 1-page proposal by (roughly) Mar 1;
full report due on early May
- ④ **Option 1:** Apply optimization to practical problems
Examples: recommendation system, object detection,
beamforming design, reinforcement learning, GAN.
I may provide some applications from companies (if possible)
- ⑤ **Option 2:** study of optimization algorithms. It doesn't need to
involve rigorous proofs, but also not pure applications.
 - How is AdaGrad compared to gradient descent?
 - When is cyclic coordinate descent slower than randomized coordinate descent?
 - When does Lagrangian multiplier method diverge?
- ⑥ **Option 3:** work on a theoretical question

Academic Misconduct

- 1 There is zero tolerance on academic misconduct. Individuals suspected of committing academic dishonesty will be reported to the university. Penalty for academic misconduct (up to 100%).
- 2 Collaboration is encouraged, but not copying each others' homeworks

Outline

- 1 Course Introduction
- 2 Optimization Background
- 3 Examples
- 4 Mathematical Review

What is Optimization (in general)

Harry (SE494): What is hypothesis testing?

Common sense concepts – we learn mathematics of them, then forget about the concepts!

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.

McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc.

Japanese companies are relatively bad at this.

What is Optimization (in general)

Harry (SE494): What is hypothesis testing?

Common sense concepts – we learn mathematics of them, then forget about the concepts!

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.

McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc.

Japanese companies are relatively bad at this.

What is Optimization (in general)

Harry (SE494): What is hypothesis testing?

Common sense concepts – we learn mathematics of them, then forget about the concepts!

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.

McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc.

Japanese companies are relatively bad at this.

What is Optimization (in general)

Harry (SE494): What is hypothesis testing?

Common sense concepts – we learn mathematics of them, then forget about the concepts!

What is optimization?

Example 1: physics. Light travels in shortest path

Example 2: AI. Is our brain doing optimization?

Example 3: Operations of companies.

McKincy efficiency manual: to improve the efficiency, you need to optimize the process/allocation/etc. *→ this is optimization*

Japanese companies are relatively bad at this.

What is Optimization (mathematically)

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & x \in X\end{array}$$

(1)

- x : the **decision variable** (discrete, continuous)
- f : the **objective function** (differentiable, convex, linear...)
- X : the **feasible region** (convex, nonempty,...)

Key questions:

- When is the problem **feasible**?
- Does the optimal solution exist?
- How to determine if a feasible x is an **optimal solution**?
- How to **find** an optimal solution? (not by exhaustive search)



Non-continuous functions: hard to optimize
If continuous, better ways than
exhaustive search exist.

How To Use Optimization

Questions

- 1 How to **model** the problem by optimization?

This is not just a math problem.

It's about understanding the core aspects, extract the key elements, figure out the logic.

This is the most challenging part, if you work for a company.

- 2 How to **solve** it?

Focus of traditional optimization course.

It is hard to teach modeling, but we will provide examples to help a bit.

We will teach the methods. Knowing the methods is a great help for modeling.

How To Use Optimization

Regular Questions

- 1 Is my problem easy to solve? (if no, don't spend time on it!)
- 2 Which algorithm should I use?
- 3 How fast should I get my results?
- 4 How do I know that the answer from my computer run is global minimum?

Other Questions

- 1 My cost function is nondifferentiable, what should I do?
- 2 Why does my algorithm become very very slow?

Each question may require a sub-area. Many of them unknown! But knowing the basics helps a lot.

Answer (partially): This course

The Main Topics to be Covered

- Unconstrained Optimization
 - ① Optimality Conditions
 - ② Gradient/Newton's Methods
 - ③ Conjugate Gradient Method (optional).
- Constrained Optimization
 - ① Optimality Conditions
 - ② Descent Methods
 - ③ Conditional Gradient Method
 - ④ Block Coordinate Descent Method
- Lagrangian Multiplier Theory
 - ① Equality Constrained Problems
 - ② Inequality Constrained Problems
 - ③ Linearly Constrained Problems

The Main Topics to be Covered (continued)

- Lagrangian Multiplier Method
 - ① Penalty Method
 - ② Method of Multipliers
- Other Topics (if time permits)
 - ① Proximal gradient method
 - ② Linear systems
 - ③ ADMM
 - ④ Accelerated methods
- How to **apply** the knowledge? Applications in Machine Learning, Signal Processing, Communication and others will be discussed together as the course continues
- For more advanced topics such as mirror descent, smoothing techniques, stochastic optimization, online algorithms, see IE 598 - BIG DATA OPTIMIZATION, Niao He

Difference With Other Courses

Difference with other courses:

Linear programming course: focus on linear programming. Won't talk about general gradient methods in detail

Convex optimization: focus on convex problems, especially linear programming and conic programming. Not so much on nonlinear problems (most machine learning problems are nonlinear)

Machine learning course: won't talk about convergence issue, and constrained problems.

My comment: if you want to understand machine learning algorithms, this course is the best fit (as introductory)

Outline

- 1 Course Introduction
- 2 Optimization Background
- 3 Examples
- 4 Mathematical Review

Example 1: Regression Problem

- ① **Training data sets** (n data points, \mathbf{a}_i is the feature, and b_i is the label)

$$\{\mathbf{a}_i, b_i\}_{i=1, \dots, n}, \mathbf{a}_i \in \mathbb{R}^d, b_i \in \mathbb{R}$$

- ② **Objective:** Learn a predictive function, characterized by $\mathbf{x} \in \mathbb{R}^d, x_0 \in \mathbb{R}$,

$$f(\mathbf{a}; \mathbf{x}) = \mathbf{x}^T \mathbf{a} + x_0 = \tilde{\mathbf{x}}^T \tilde{\mathbf{a}}, \quad \text{with } \tilde{\mathbf{x}} := [\mathbf{x}, x_0], \tilde{\mathbf{a}} := [\mathbf{a}, 1]$$

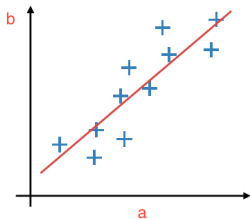
- ③ We have absorbed x_0 in \mathbf{x} and augmenting \mathbf{a}_i 's with extra 1

- ④ Use the cost function

$$\text{Loss}(\mathbf{x}) = \underbrace{\frac{1}{2} \sum_{i=1}^n (\tilde{\mathbf{x}}^T \tilde{\mathbf{a}}_i - b_i)^2}_{\text{squared loss}} = \|\mathbf{A}^T \tilde{\mathbf{x}} - \mathbf{b}\|^2$$

*\hat{x} : predict income b
based on features
(a_1, a_2, a_3, \dots)
↓ ↓
education facebook friends #*

Minimize w.r.t. $\tilde{\mathbf{x}}$: $\tilde{\mathbf{x}}^* = (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}\mathbf{b}$ (closed-form solution!)



Example 1: Regression Problem

- It is the basic regression model; the predicted quantities are **numerical**
- A lot of aspects (including modeling and algorithm) can be improved when the data matrix A becomes “large”

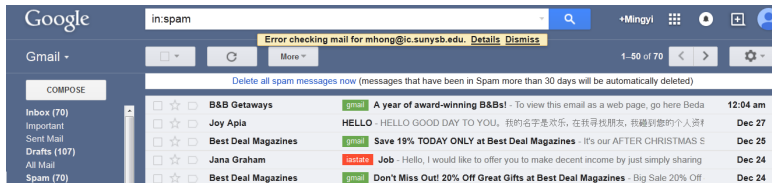
Example 1: Regression Problem

- What if we also want to select **a few** key features that are most important?
LASSO.
- What if the data sets are arriving **sequentially**?
online optimization/learning.
- What if the data sets are **distributed** at different locations?
distributed optimization.
- What if the dimension of the variable is **huge** (x very long, lots of features), while the data is scarce (n is small)?
Large-scale optimization.
- Why we are using the ℓ_2 loss? any other loss function?
- What if the relationship is **nonlinear**?
neural networks, nonlinear least squares

Example 2: Classification

(skip)

- Every email services have automatic spam detection mechanisms
- The basic task is the following
 - ① Given an excerpt of an email, represented by a vector \mathbf{a}_i , where the elements can be words from the email
 - ② Ask the question whether it is a spam or not ($b_i = +1, -1$)

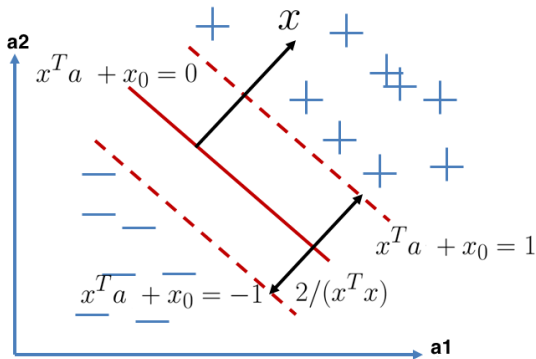


Example 2: Classification (cont.)

- **Training Stage:** data set $\{\mathbf{a}_i, b_i\}_{i=1}^n$, $\mathbf{a}_i \in \mathbb{R}^d$, $b_i \in \{-1, +1\}$
- $b_i = +1$ (spam); $b_i = -1$ (regular)
- **Objective:** learn the classification function
 $f(\mathbf{a}; \mathbf{x}) = \text{sgn}(\mathbf{x}^T \mathbf{a} + x_0)$, where **sgn** is the “sign function”
- **Testing stage**
 - ① Given a feature vector $\hat{\mathbf{a}}$
 - ② Plug in $f(\mathbf{a}; \mathbf{x})$
 - ③ Classify based on the sign (i.e., “+” as spam, “-” as regular)
- Suppose the training data are **linearly separable** (?)
- **Task 1:** Select **two hyperplanes** to separate the data points
- **Task 2:** Try to maximize their distance
- What's a good formulation?

Example 2: Classification (cont.)

- Intuition:** Find the separating plane that is far away from both classes



Example 2: Classification (cont.)

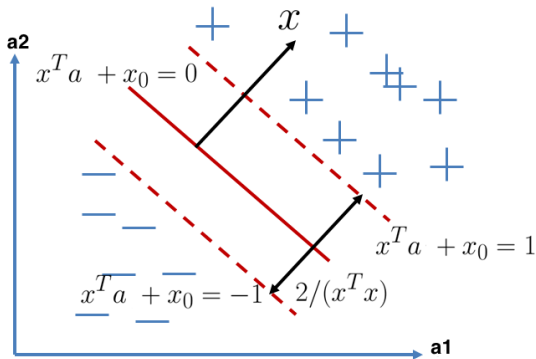
- **Formulation:** Consider the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T \mathbf{x} \\ \text{subject to} \quad & b_i(\mathbf{x}^T \mathbf{a}_i + x_0) \geq 1, \forall i \end{aligned}$$

- One data point, one constraint
- The blue part says there is no classification error
- If $b_i = 1$ (is a spam), then $(\mathbf{x}^T \mathbf{a}_i + x_0) \geq 1$
- If $b_i = -1$ (regular), then $(\mathbf{x}^T \mathbf{a}_i + x_0) \leq 1$

Example 2: Classification (cont.)

- Intuition:** Find the separating plane that is far away from both classes



Example 2: Classification (cont.)

- This is one instance of the **support vector machine (SVM) problem**
- How about non-separable cases?
- How to solve the underlying optimization problem?
- Directly or need some re-formulation?
- What if the decision boundary is nonlinear?
- Data comes in sequentially?
-

Example 3: Training Neural Networks (skip)

- Neural Networks, especially Deep Neural Networks (DNN) become increasingly popular for various machine learning tasks
- Lots of high-profile applications: "Google Brain" by Google, 2012
- 16,000 computer processors and 100 million images, 20,000 distinct items, look for cats; research paper can be found [here](#)
- A nice article about the history of DL and NN on [here](#)



Example 3: Training Neural Network

Finding nonlinear classification boundary?

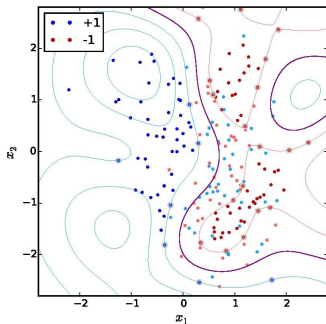
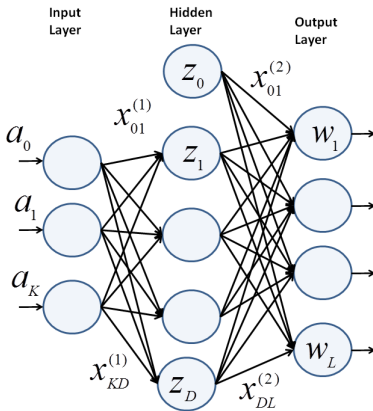


Figure: Nonlinear Classification [Wikipedia]

Example 3: Training Neural Network

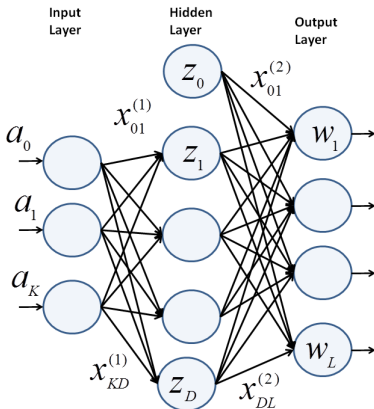
- **Input layer:** Each input **one data point**, K -dimensional: a_0, \dots, a_k
- **Note:** Here we have a **single** data point, a_k denotes its k th element



Example 3: Training Neural Network

- **Hidden layer:** Each node, a nonlinear function g (tanh, sigmoid)
- **Nonlinearly** transform the inputs to outputs

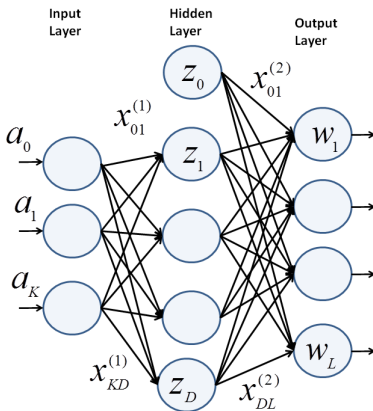
$$z_d = g \left(\sum_{k=1}^K a_k x_{k,d}^{(1)} \right)$$



Example 3: Training Neural Network

- **Output layer:** Each output node, a nonlinear function (sigmoid)

$$w_\ell = h \left(\sum_{d=0}^D z_d x_{d,\ell}^{(2)} \right), \quad \text{if binary classification, } L = 1$$



Example 3: Training Neural Network

- What are the optimization variables here?
- How to setup the optimization problem?
- How to solve the resulting problem?

Example 3: Training Neural Network

- What are the optimization variables here?
- How to setup the optimization problem?
- How to solve the resulting problem?

Example 3: Training Neural Network

regression: $b = a^T x$ ^{↑ variable}
neural-net: $b = w_3 \sigma(w_2 \sigma(w_1 x))$

- What are the optimization variables here? w_1, w_2, w_3, \dots
- How to setup the optimization problem?
- How to solve the resulting problem?

Outline

- 1 Course Introduction
- 2 Optimization Background
- 3 Examples
- 4 Mathematical Review

Overview

- 1 Notations: Sets, functions, derivatives, gradients
- 2 Vectors, matrices
- 3 Norms, sequences, limits, continuity
- 4 Mean value theorems
- 5 Implicit function theorem
- 6 Contraction mappings
- 7 Reference Appendix A, B of the textbook
- 8 Get yourself familiar with them

Notations

① **Sets:** $X, x \in X, X_1 \cup X_2, X_1 \cap X_2$

② **Inf and Sup:**

The supremum of a nonempty set $X \subset \mathbb{R}$ is the **smallest** scalar y :
min

$$y \geq x, \forall x \in X$$

The infimum of a nonempty set $X \subset \mathbb{R}$ is the **largest** scalar y :

$$y \leq x, \forall x \in X$$

If $\sup X \in X$ (or, $\inf \in X$), then we say $\sup X = \max X$ (or, $\inf X = \min X$).

$$\sup\{1/n \mid n \geq 1\} = \overset{1}{?}, \quad \inf\{x \in \mathbb{R} \mid 0 < x < 1\} = \overset{0}{?}$$

inf {1/n | n ≥ 1} = 0

③ **Function:**

$f : X \rightarrow Y$, X is called the domain Y is called the range

Monotonicity \rightarrow **Inverse function** f^{-1} exists

Vectors

\mathbf{x}

- ① **Vector:** a vector $\mathbf{x} = [x_1, \dots, x_n]$ is a **column** of scalars
- ② **Linear combination:** if $\mathbf{x} = [x_1, \dots, x_n]$ and $\mathbf{y} = [y_1, \dots, y_n]$, then the linear combination is given by

$$\alpha \mathbf{x} + \beta \mathbf{y} = (\alpha x_1 + \beta y_1, \alpha x_2 + \beta y_2, \dots, \alpha x_n + \beta y_n)$$

- ③ **Inner product:** $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}' \mathbf{y} = \sum_{i=1}^n x_i y_i$

Question: when is inner product positive, negative, zero?

Orthogonality: $\mathbf{x} \perp \mathbf{y}$ iff (if and only if) $\langle \mathbf{x}, \mathbf{y} \rangle = 0$

- ④ **Linearly Independent:** A set of vectors $\{\mathbf{x}^1, \dots, \mathbf{x}^r\}$ are **linearly independent** if there does not exist a $(\alpha_1, \dots, \alpha_r) \neq 0$ s.t.

$$\langle \mathbf{x}, \mathbf{x} \rangle \quad \sum_{i=1}^r \alpha_i \mathbf{x}^i = \mathbf{0}$$

Vectors

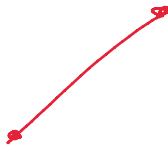
- 1 Basis and dimension of a subspace:
- 2 Orthogonal complement of a subspace S :

$$S^\perp := \{\mathbf{x} \mid \langle \mathbf{x}, \mathbf{y} \rangle = 0, \forall \mathbf{y} \in S\}$$

- 3 **Vector norms:** A norm $\|\mathbf{x}\|$ on \mathbb{R}^n that assigns a **scalar** $\|\mathbf{x}\|$ to every $\mathbf{x} \in \mathbb{R}^n$ that satisfying

- 1 $\|\mathbf{x}\| \geq 0$ for all \mathbf{x}
- 2 $\|c\mathbf{x}\| = |c|\|\mathbf{x}\|$ for all $c \in \mathbb{R}$ and all \mathbf{x}
- 3 $\|\mathbf{x}\| = 0$ iff $\mathbf{x} = 0$
- 4 $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all \mathbf{x}, \mathbf{y}

- 4 Measure some kind of “~~distance~~” *length*



Vectors

1 Common norms

Euclidean Norm : $\|\mathbf{x}\|_2 = (\mathbf{x}'\mathbf{x})^{1/2} = \sqrt{\sum_{i=1}^n x_i^2}$

ℓ_p Norm : $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$ for some $p \geq 1$

ℓ_1 Norm : $\|\mathbf{x}\|_1 = \sum_i |x_i|$

ℓ_∞ Norm : $\|\mathbf{x}\|_\infty = \max_i |x_i|$

$p = \infty$

2 Cauchy-Schwartz inequality: $\langle \mathbf{x}, \mathbf{y} \rangle \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$

related: $\langle \mathbf{x}, \mathbf{y} \rangle \leq \|\mathbf{x}\|_1 \|\mathbf{y}\|_\infty$

·  dual norm

Cauchy-Schwartz inequality

An important inequality about the inner product of two vectors is the **Cauchy-Swartz inequality**

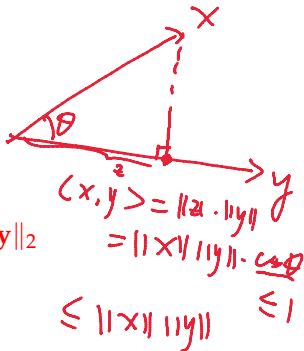
- 1 Characterizes the inner product of two vectors with their norms
- 2 Given two vectors x and y of the same size, we have

$$\langle x, y \rangle \leq \|x\|_2 \|y\|_2$$

- 3 Why this is true? Geometrically?
- 4 Useful fact about inner product

$$\langle x, y \rangle = \cos(\theta) \|x\|_2 \|y\|_2$$

where θ is the **angle** between x and y



Matrices

- 1 For any matrix \mathbf{A} , we use a_{ij} (or A_{ij}) to denote its (i, j) th entry.
- 2 Matrix addition, multiplication, transpose, symmetric matrices $\mathbf{A} = \mathbf{A}'$.

$$[\mathbf{AB}]' = \mathbf{B}'\mathbf{A}', \quad \mathbf{AB} \neq \mathbf{BA}$$

non-commutative

- 3 Let \mathbf{A} be a $m \times n$ matrix.
 - Range of \mathbf{A} : $R(\mathbf{A}) = \{\mathbf{Ax}, \mid \mathbf{x} \in \mathbb{R}^n\}$;
 - Null space of \mathbf{A} : $N(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{Ax} = 0\}$
 - Rank of \mathbf{A} $\text{Rank}(\mathbf{A})$. Full rank matrix \mathbf{A} : $\text{Rank}(\mathbf{A}) = \min\{m, n\}$.
- 4 Inner product:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}(\mathbf{AB}') = \sum_{i,j} A_{ij}B_{ij}$$

where the trace operate is given by

$$\text{Tr}[\mathbf{A}] = \sum_{i=1}^n A_{ii}$$

Square Matrices

- ① Square matrix ($m = n$); Identity matrix \mathbf{I}
- ② Determinant $\det(\mathbf{A})$, inverse \mathbf{A}^{-1} . \mathbf{A}^{-1} exists iff $\det(\mathbf{A}) \neq 0$
- ③ Useful identities: $\det(\mathbf{A}) = \det(\mathbf{A}')$
- ④ Orthogonal matrices: $\mathbf{A}\mathbf{A}' = \mathbf{I}$
- ⑤ (Complex) Eigenvalue λ : $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ for some $\mathbf{x} \neq 0$
- ⑥ Spectral radius: $\rho(\mathbf{A}) = \max_i \{|\lambda_i|\}$, λ_i is an eigenvalue of \mathbf{A} .

Square Matrices

- ① Eigen-decomposition of a symmetric matrix:

$$\mathbf{A} = \mathbf{P}' \mathbf{\Lambda} \mathbf{P}$$

where \mathbf{P} is orthonormal ($\mathbf{P}'\mathbf{P} = \mathbf{I}$), $\mathbf{\Lambda}$ is diagonal and real.

- ② Positive (semi-) definite matrix: $\mathbf{A} \succeq 0$

(symmetric)

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x}$$

(2)

- ③ Property: $\mathbf{A} \succeq 0, \mathbf{B} \succeq 0 \rightarrow \mathbf{A} + \mathbf{B} \succeq 0$; $\mathbf{A} \succeq \mathbf{B} \rightarrow \mathbf{A} - \mathbf{B} \succeq 0$

All eigenvalue of \mathbf{A} are non-negative

- ④ Square root $\mathbf{A}^{1/2}$: $\mathbf{A}^{1/2} := \mathbf{P} \sqrt{\mathbf{\Lambda}} \mathbf{P}'$, where $\mathbf{A} = \mathbf{P} \mathbf{\Lambda} \mathbf{P}'$ is the eigen-decomposition of $\mathbf{A} \succeq 0$.

A “generalization” of **positive number** for the scalars

Single Value Decomposition

① Single value decomposition (SVD):

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}' \in \mathbb{R}^{m \times n}, \quad (3)$$

- $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ rectangular diagonal matrix with diagonals $\sigma_1 \geq \sigma_2 \geq \dots, \sigma_n \geq 0$;
- $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ orthonormal

② Relationship of SVD and ED: σ_i^2 is an eigenvalue of $\mathbf{A}\mathbf{A}'$ (Why?)

$$\mathbf{A}\mathbf{A}' = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}')(\mathbf{V}\mathbf{\Sigma}\mathbf{U}') = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}'$$

③ Condition number: $\kappa(\mathbf{A}) = \sigma_1 / \sigma_n$ very important for optimization!!

Matrices and Norms

1 Norms: *for matrix*

$$= \|\sigma\|_2$$

$$\text{Frobenius Norm : } \|\mathbf{A}\|_F = \left(\sum_{i,j} |A_{ij}|^2 \right)^{1/2} = \left(\sum_i \sigma_i^2 \right)^{1/2} \quad (4)$$

$$\text{Nuclear Norm : } \|\mathbf{A}\|_* = \sum_i \sigma_i \sim \|\sigma\|_1, \quad \sigma = (\sigma_1, \dots, \sigma_n) \quad (5)$$

$$\text{Matrix 2-norm (spectral norm) : } \|\mathbf{A}\|_2 = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_i \sigma_i = \|\sigma\|_\infty \quad (6)$$

2 Useful inequalities:

$$\|\mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2, \quad \|\mathbf{A}\|_* \geq \|\mathbf{A}\|_F \geq \|\mathbf{A}\|_2$$

Relation between $\rho(\mathbf{A})$ and $\|\mathbf{A}\|_2$. $\rho(\mathbf{A}) \leq \|\mathbf{A}\|_2$

Derivatives

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously twice differentiable function.

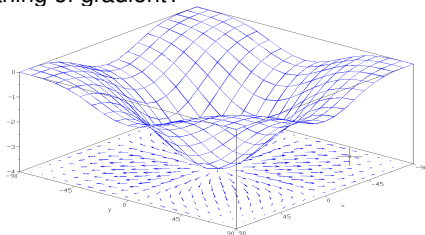
- ① Partial derivative (where \mathbf{e}_i is the i th unit vector of \mathbb{R}^n)

$$\frac{\partial f(\mathbf{x})}{\partial x_i} := \lim_{t \rightarrow 0} \frac{f(\mathbf{x} + t\mathbf{e}_i) - f(\mathbf{x})}{t}$$

- ② Gradient vector (a column vector) [example: $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} - \mathbf{b}$]:

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

Physical meaning of gradient?



Wikipedia: “the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction”

Derivatives

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously twice differentiable function.

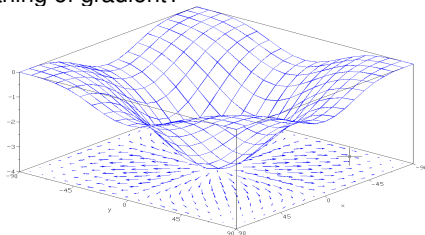
- 1 Partial derivative (where \mathbf{e}_i is the i th unit vector of \mathbb{R}^n)

$$\frac{\partial f(\mathbf{x})}{\partial x_i} := \lim_{t \rightarrow 0} \frac{f(\mathbf{x} + t\mathbf{e}_i) - f(\mathbf{x})}{t}$$

- 2 Gradient vector (a column vector) [example: $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} - \mathbf{b}$]:

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

Physical meaning of gradient?



Wikipedia: “the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction”

Derivatives

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously twice differentiable function.

- ① Hessian matrix:

$$\nabla^2 f = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right] \in \mathbb{R}^{n \times n}$$

e.g.
 $f(x) = 2x_1^2 + 3x_2^2 + 5x_1x_2$
 $\nabla^2 f = \begin{bmatrix} 4 & 5 \\ 5 & 6 \end{bmatrix}$

- ② Taylor expansion:

$$f(\mathbf{y}) - f(\mathbf{x}) = \nabla f(\mathbf{x})'(\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{x} - \mathbf{y})' \nabla^2 f(\mathbf{x})(\mathbf{x} - \mathbf{y}) + o(\|\mathbf{x} - \mathbf{y}\|^2)$$

$$f(\mathbf{x}) = \sum_{i,j=1}^n a_{ij} x_i x_j.$$

Practice:

Q1: How to compute the Hessian of $f(x) = x^T A x$?

Q2: How to compute the Hessian of $f(x, y) = \|M - xy^T\|_F^2$, where $x, y \in \mathbb{R}^n$?

$$x, y \in \mathbb{R}^n.$$

Derivatives: Chain Rule

- ① Scalar case: suppose $f, g : \mathbb{R} \rightarrow \mathbb{R}$ are functions, and $f'(x)$ and $g'(f(x))$ exist, then $h(x) \triangleq g(f(x))$ satisfies

$$h'(x) = g'(f(x))f'(x).$$

Example: $f(x) = \sin x$, $g(y) = y^2$, $h(x) = (\sin x)^2$, then

$$h'(x) = 2 \sin x \cdot \cos x.$$

- ② Vector case: $f : \mathbb{R}^k \rightarrow \mathbb{R}^m$ and $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$, and $f'(x)$ and $g'(f(x))$ exist. Let $h(\mathbf{x}) = g(f(\mathbf{x}))$ (function composition), then

$$\nabla h(\mathbf{x}) = \nabla f(\mathbf{x}) \nabla g(f(\mathbf{x}))$$

Example: $g(z) = z^2/2$, $f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$, then $h(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$

$$\nabla \left(\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \right) = \mathbf{A}^T (\mathbf{A}\mathbf{x} - \mathbf{b}), \quad \nabla^2 f(\mathbf{A}\mathbf{x}) = \mathbf{A}'\mathbf{A}$$
$$(\mathbf{A}\mathbf{x} - \mathbf{b})' = \mathbf{A}'$$

Contraction Mapping

- ① **Lipschitzian Property:** $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfies

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq \underline{\gamma} \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}$$

γ is called the **Lipschitz constant**; the function is called γ -**Lipschitz continuous**

$$\|f(x) - f(y)\| = \|2x - 2y\| \leq 2 \|x - y\|$$

Example 1: $f(x) = 2x$ is 2-Lipschitz continuous;

Example 2: What about $f(x) = Ax$?

$$\begin{pmatrix} 4 & 5 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. f(x) \text{ is } \|A\|_2\text{-Lipschitz continuous}$$

Non-Example 3: What about $f(x) = x^2$?

$$\|x - y\| \cdot \|x + y\| = \|x^2 - y^2\| \leq ? \|x - y\|$$

- ② If $\gamma \leq 1$, then f is called a **non-expansive mapping**

- ③ If $\gamma < 1$, then f is called a **contraction mapping**

Example 1: $f(x) = 2x$ is not a contraction mapping; $f(x) = 0.5x$ is.

Example 2: $f(x) = Ax$ is a contraction mapping iff ?

$$\|A\|_2 = \max_i \sigma_i < 1$$

Fixed Point Theorem

$$x^0 = x, \quad x^1 = f(x^0), \quad x^2 = f(x^1), \quad x^3 = f(x^2), \dots$$

$$\|f(x) - f(y)\| < \|x - y\|,$$



$$1.5, 1.1, 1.01, 1.001, \dots \rightarrow 1$$

1 **Fixed point theorem:** If f is a contraction mapping, then the iterated function sequence ~~converges~~

$$x, f(x), f(f(x)), \dots,$$

converges to (a unique fixed point x^* (independent of x) satisfying

$$x^* = f(x^*).$$

Can only converge to $x^* = f(x^*)$.

$$x^k \rightarrow x^\infty,$$

$$x^{k+1} = f(x^k) \longrightarrow x^\infty = f(x^\infty).$$

Homework

- 1 Getting yourself familiar with mathematical backgrounds
- 2 **Reading:** Appendix A of the textbook.