COMP 414/514: Optimization – Algorithms, Complexity and Approximations

Lecture 1

Overview

$$
\begin{aligned}
\min_{x} \quad & f(x) \\
\text{s.t.} \quad & x \in \mathcal{C}
\end{aligned}
$$

Overview

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

- Different objective classes
- Different strategies within each problem
- Different approaches based on computational capabilities
- Different approaches based on constraints

Overview

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

— Different objective classes
— Different strategies within each problem
— Different approaches based on computational capabilities
— Different approaches based on constraints

And, always having in mind applications in machine learning, AI and signal processing

Motivation (no fancy images included)

Provable efficiency

Motivation (no fancy images included)

Provable efficiency

Lots of data

Motivation (no fancy images included)

Provable efficiency

Harder problems

Lots of data

Motivation (no fancy images included)

More complicated models

Provable efficiency

Harder problems

Lots of data

Motivation

(no fancy images included)

More complicated models

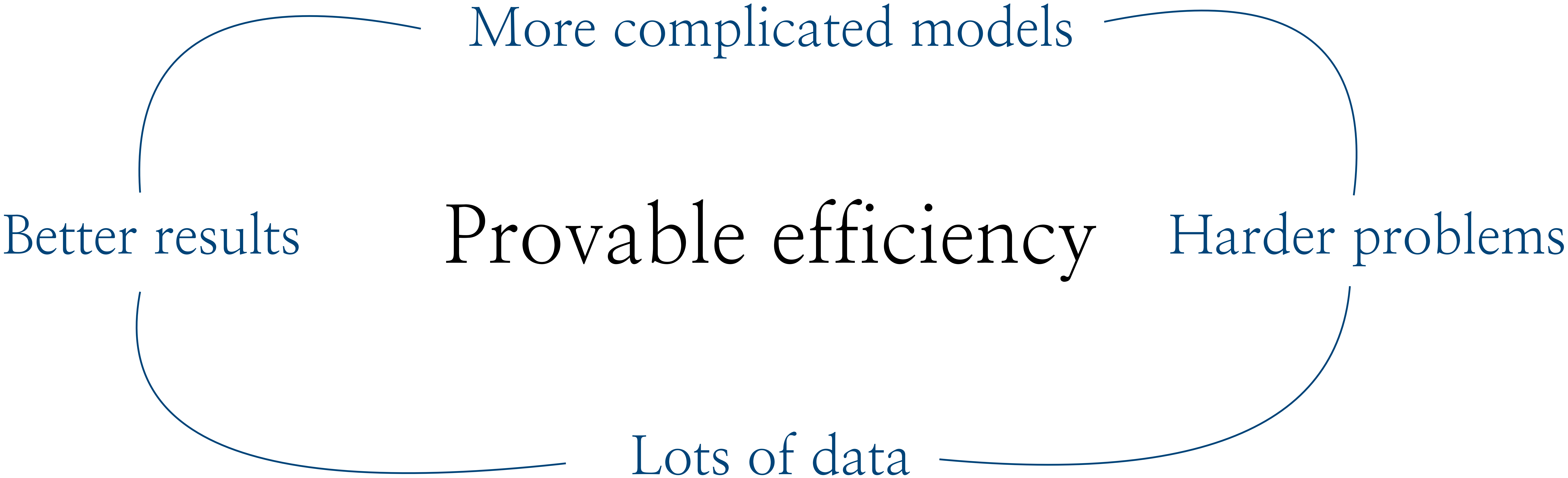
Better results

Provable efficiency

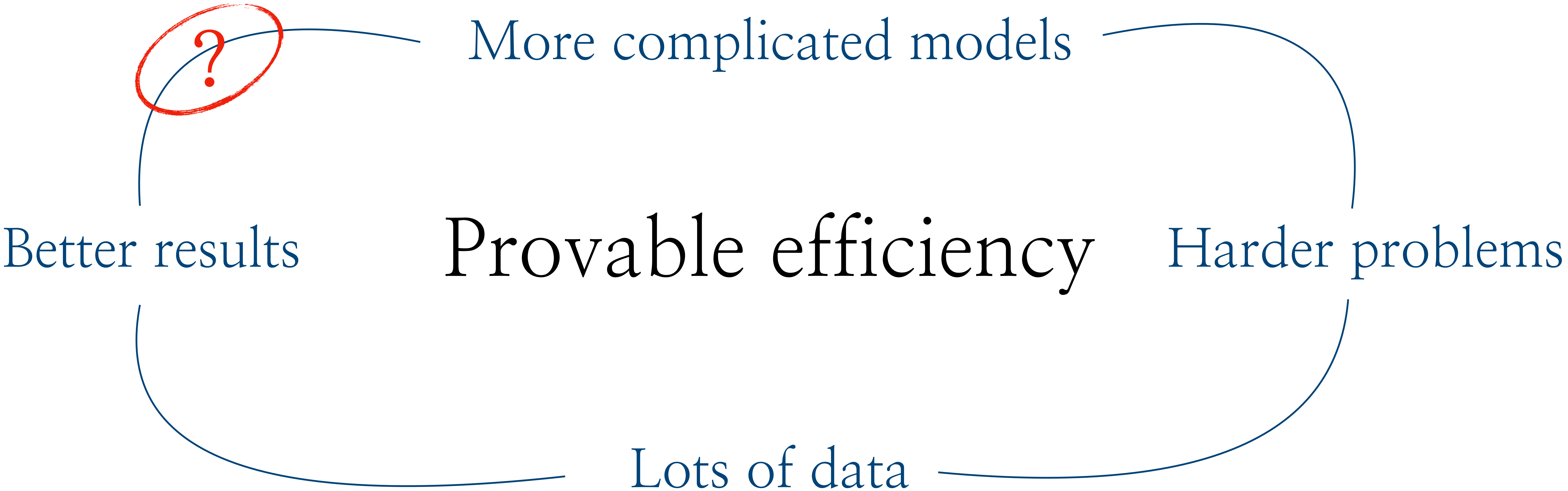
Harder problems

Lots of data

Motivation (no fancy images included)



Motivation (no fancy images included)



Motivation (no fancy images included)

Provable efficiency

"What shall we do?"

Motivation (no fancy images included)

Provable efficiency

"What shall we do?"

Set up algo nicely

Use prior knowledge

Converge faster

Exploit resources

Topics

- Continuous **optimization** (in general)
  - See syllabus
  - Both theory and practice

Topics

- Continuous **optimization** (in general)
  - See syllabus
  - Both theory and practice
- Recent applications that drive research

Topics

- Continuous **optimization** (in general)
  - See syllabus
  - Both theory and practice
- Recent applications that drive research
- When no theory applies, some intuition

Topics NOT covered in this course

- (Mixed) integer programming

(See CMOR)

Topics NOT covered in this course

- (Mixed) integer programming

(See CMOR)

- Combinatorial optimization algorithms

(E.g., Graph algorithms)

Topics NOT covered in this course

- (Mixed) integer programming
  (See CMOR)
- Combinatorial optimization algorithms
  (E.g., Graph algorithms)
- Randomized algorithms
  (See Anshu's course, Maryam's course)

Topics NOT covered in this course

- (Mixed) integer programming (See CMOR)
- Combinatorial optimization algorithms (E.g., Graph algorithms)
- Randomized algorithms (See Anshu's course, Maryam's course)
- Online algorithms, bandits (See Maryam's course)

Topics NOT covered in this course

- (Mixed) integer programming (See CMOR)
- Combinatorial optimization algorithms (E.g., Graph algorithms)
- Randomized algorithms (See Anshu's course, Maryam's course)
- Online algorithms, bandits (See Maryam's course)
- Bayesian algorithms

Topics NOT covered in this course

- (Mixed) integer programming (See CMOR)
- Combinatorial optimization algorithms (E.g., Graph algorithms)
- Randomized algorithms (See Anshu's course, Maryam's course)
- Online algorithms, bandits (See Maryam's course)
- Bayesian algorithms
- Deep learning architectures (See Ankit's course)

Examples

- Least squares / linear regression

(No, we will not re-define it)

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

Examples

- Least squares / linear regression

(No, we will not re−define it)

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{C} \end{aligned} \quad \Rightarrow \quad \min_x \frac{1}{n} \sum_{i=1}^{n} \left(y_i - a_i^\top x \right)^2$$

Examples

- Quantum state tomography from limited samples

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & X \in \mathcal{C} \end{aligned}$$

Examples

- Quantum state tomography from limited samples

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & X \in \mathcal{C} \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min_{X} \quad & \sum_{i=1}^{n} \big(y_i - \mathrm{Tr}(A_i^\top X)\big)^2 \\ \text{s.t.} \quad & \mathrm{Tr}(X) \leq 1 \\ & X \succeq 0 \end{aligned}$$

Examples

- Fleet management

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & X \in \mathcal{C} \end{aligned}$$

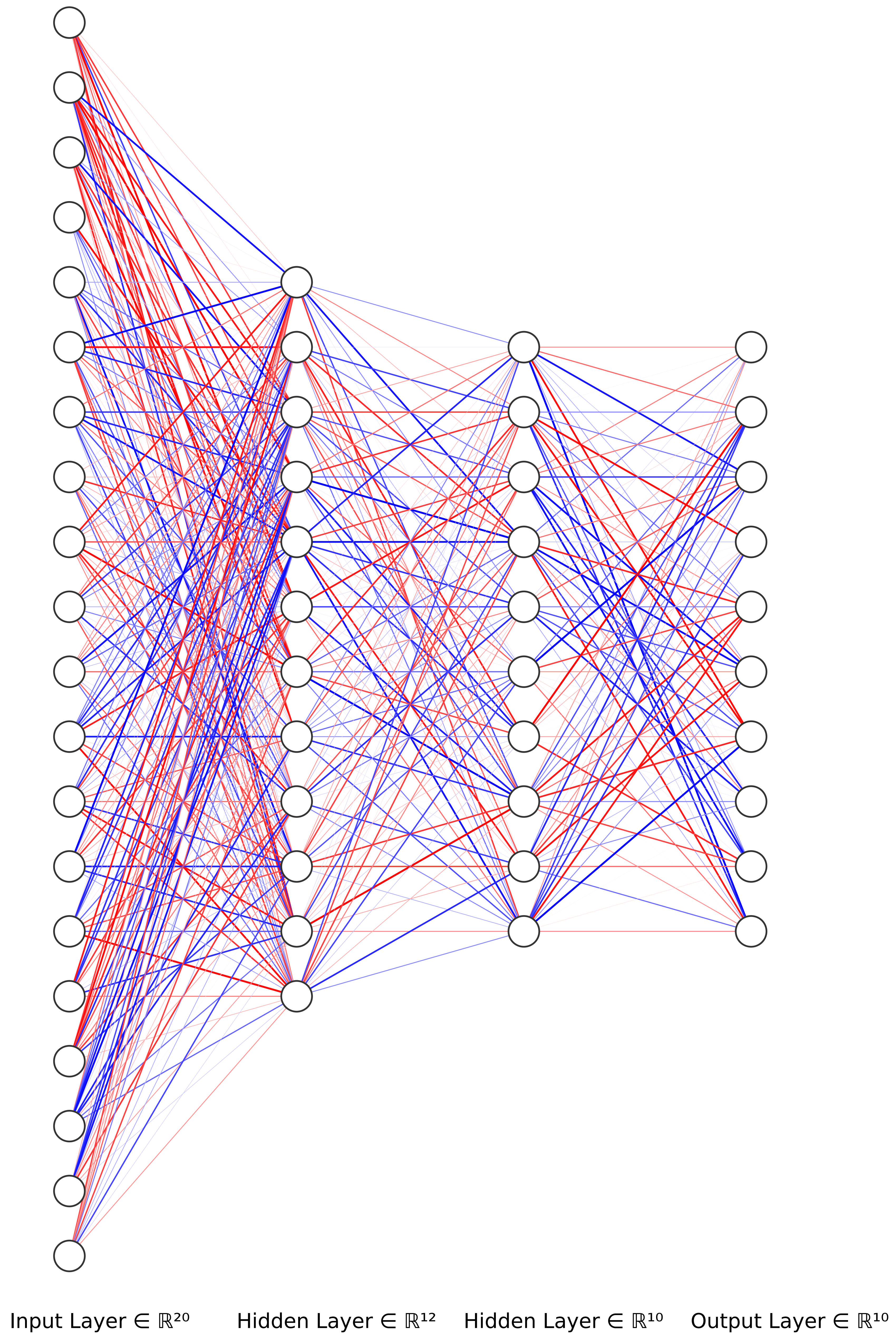
Examples

– Fleet management

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & X \in \mathcal{C} \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min_{x \in \{0,1\}^m, y \in \{0,1\}} \quad & f(y) = \sum_{i \in \mathcal{V}} \sum_{k=1}^{p} d_i (1-q) q^{k-1} y_{ik} \\ \text{s.t.} \quad & \sum_{j \in \mathcal{W}_i} x_j \geq \sum_{k=1}^{p} y_{ik}, i \in \mathcal{V} \\ & \sum_{j \in \mathcal{W}} x_j = p \\ & x_j \leq p_j \end{aligned}$$

Examples

– Neural networks



Examples

– Neural networks

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & X \in \mathcal{C} \end{aligned} \quad \Rightarrow$$

$$\min_{W_i} \quad f(W_1, W_2) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\widehat{y}_i, y_i\right).$$

where

$$\widehat{y}_i = \texttt{softmax}\left(\sigma\left(W_2 \cdot \sigma\left(W_1 \cdot x_i\right)\right)\right)$$

Any questions?

Who is this class for?

- Definitely, for PhD students

Who is this class for?

- Definitely, for PhD students

- Master or even undergraduates that want to start research

(but get in touch with me soon to assess your background)

Who is this class for?

- Definitely, for PhD students
- Master or even undergraduates that want to start research

  (but get in touch with me soon to assess your background)
- Just auditing is fine by me

What is the vision for this course?

- For starters, this will always be an evolving course

(Any feedback is more than welcome)

What is the vision for this course?

- For starters, this will always be an evolving course

(Any feedback is more than welcome)

- My purpose and vision is to introduce a series of optimization courses in the CS (and Duncan Hall's in general) curriculum

What is the vision for this course?

- For starters, this will always be an evolving course

(Any feedback is more than welcome)

- My purpose and vision is to introduce a series of optimization courses in the CS (and Duncan Hall's in general) curriculum

- The vision is for this course to be part of a sequence of courses that will focus on the theory+practice of methods

(I'm also teaching COMP182)

Course format

- Lectures (slides) + whiteboard + in-class code running

(Some lectures have presentations, others will be handwritten)

Course format

- Lectures (slides) + whiteboard + in−class code running

(Some lectures have presentations, others will be handwritten)

- Material broad enough to cover range of problems

(But definitely will not cover each individual's interest at 100%)

Course format

– Lectures (slides) + whiteboard + in–class code running

(Some lectures have presentations, others will be handwritten)

– Material broad enough to cover range of problems

(But definitely will not cover each individual’s interest at 100%)

– Some material inspired from personal research

Course format

- Lectures (slides) + whiteboard + in-class code running

  (Some lectures have presentations, others will be handwritten)

- Material broad enough to cover range of problems

  (But definitely will not cover each individual's interest at 100%)

- Some material inspired from personal research

- "Interlude" lectures to provide some background (if needed)

Course format

- Lectures (slides) + whiteboard + in-class code running

  (Some lectures have presentations, others will be handwritten)

- Material broad enough to cover range of problems

  (But definitely will not cover each individual's interest at 100%)

- Some material inspired from personal research

- "Interlude" lectures to provide some background (if needed)

- Your workload:

  Graduate – HWs, final project
  Undergraduate – HWs, final exam
  (Additional workload: possible midterm, scribing)

Regarding assignments

- Weekly assignments

Regarding assignments

- Weekly assignments
- Some questions are harder than others

Regarding assignments

- Weekly assignments
- Some questions are harder than others
- Some questions might not feel intuitive

("I'm a computer scientist! Why should I care about optimization?")

Regarding assignments

- Weekly assignments
- Some questions are harder than others
- Some questions might not feel intuitive

  ("I'm a computer scientist! Why should I care about optimization?")
- Try to do the best you can

  (There will be a reweighing at the end of the course, only if necessary)

Goals + outcomes

- Learn about research in related fields

Goals + outcomes

- Learn about research in related fields
- Make connections between areas,
  understand how research advances in such areas..

Goals + outcomes

- Learn about research in related fields
- Make connections between areas,
  understand how research advances in such areas..
- Consider possible extensions of these works (project)

Goals + outcomes

- Learn about research in related fields
- Make connections between areas,
  understand how research advances in such areas..
- Consider possible extensions of these works (project)
- Comprehend how optimization is key in ML/AI/SP

Goals + outcomes

- Learn about research in related fields

- Make connections between areas,
  understand how research advances in such areas..

- Consider possible extensions of these works (project)

- Comprehend how optimization is key in ML/AI/SP

- Read and review recent papers

My goals

- Not to judge you on small details in HWs

(But judge whether you have thought about solving the questions)

My goals

- Not to judge you on small details in HWs

  (But judge whether you have thought about solving the questions)

- Spark your interest in research
  where math and practice are combined together

Prerequisites

- If you have taken any ML class, you are good to go

Prerequisites

- If you have taken any ML class, you are good to go
- Basics of calculus, linear algebra, basic knowledge of ML topics

Prerequisites

- If you have taken any ML class, you are good to go
- Basics of calculus, linear algebra, basic knowledge of ML topics
- Programming skills are not necessary

(but might be required, depending on the project selected)

Prerequisites

- If you have taken any ML class, you are good to go
- Basics of calculus, linear algebra, basic knowledge of ML topics
- Programming skills are not necessary

  (but might be required, depending on the project selected)
- A quiz was usually provided for self-assessment, but I decided to make it an additional HW

Grading policy

- 50% HWs
- 50% project/final exam
  (If there will be a midterm, this will change)
- 5%: scribing notes (**bonus**)

Grading policy

- 50% HWs
- 50% project/final exam
  (If there will be a midterm, this will change)
- 5%: scribing notes (**bonus**)

Usually there is scaling in final grades. For me, a good grade is given based on the overall performance of the students: I value self-motivation, being proactive and enthusiasm.

Participation + scribing

- “Παν μετρον αριστον”

(Moderation is key)

Participation + scribing

- "Παν μετρον αριστον" (Moderation is key)
- Piazza is set up but also an email account is available

  (I have set up an email for the course – see the syllabus)

Participation + scribing

- "Παν μετρον αριστον" (Moderation is key)
- Piazza is set up but also an email account is available
  (I have set up an email for the course – see the syllabus)
- A slack channel will be set up for those doing a project

Participation + scribing

- "Παν μετρον αριστον" (Moderation is key)
- Piazza is set up but also an email account is available
  (I have set up an email for the course – see the syllabus)
- A slack channel will be set up for those doing a project
- Scribing is useful for you to understand better the material
  (or even get a better intuition than what was instructed)

Participation + scribing

- "Παν μετρον αριστον" (Moderation is key)
- Piazza is set up but also an email account is available
  (I have set up an email for the course – see the syllabus)
- A slack channel will be set up for those doing a project
- Scribing is useful for you to understand better the material
  (or even get a better intuition than what was instructed)
- Individuals or groups (2–3) of volunteers for each lecture
  (it will depend on the attendance)

Participation + scribing

- "Παν μετρον αριστον" (Moderation is key)
- Piazza is set up but also an email account is available
  (I have set up an email for the course – see the syllabus)
- A slack channel will be set up for those doing a project
- Scribing is useful for you to understand better the material
  (or even get a better intuition than what was instructed)
- Individuals or groups (2–3) of volunteers for each lecture
  (it will depend on the attendance)
- Deliverable in LaTEX
  (template available online)

HWs

– Deliverable in LaTEX

Reviews (when applicable)

- Select papers from a pile of .pdfs that will be provided

(Reviews will be related to the topics currently taught)

Reviews (when applicable)

- Select papers from a pile of .pdfs that will be provided

  (Reviews will be related to the topics currently taught)

- Single page reviews, similar to NIPS/ICML standards:

  (but not random as it usually is now)

  - Comment on novelty, clarity, importance
  - Strengths and weaknesses
  - Main comments + your overall score

Presentations (for final projects)

- How does a project report look like?

Presentations (for final projects)

- How does a project report look like?
- Final exams: necessary for undergrads/optional for grads(?)

Presentations (for final projects)

- How does a project report look like?

- Final exams: necessary for undergrads/optional for grads(?)

- There might be some discussions during the lectures
  Take advantage by asking questions

(depending on the size of the class)

Presentations (for final projects)

- How does a project report look like?

- Final exams: necessary for undergrads/optional for grads(?)

- There might be some discussions during the lectures
  Take advantage by asking questions

(depending on the size of the class)

- Presentation should be at most XX minutes

(tentative – depends on the class size)

Presentations (for final projects)

- How does a project report look like?
- Final exams: necessary for undergrads/optional for grads(?)
- There might be some discussions during the lectures
  Take advantage by asking questions

  (depending on the size of the class)
- Presentation should be at most XX minutes

  (tentative – depends on the class size)
- Grading: slides quality, clarity of main ideas

Presentations (for final projects)

(not certain yet)

(Course website)

Final Project

(Course website)

Final Project

(Course website)

Please come find me the earliest to discuss projects

Final Project

(Course website)

Please come find me the earliest to discuss projects

You should start reading papers soon, so that around mid-way you have a good project proposal

Communication

– I can handle emails in a very responsive way

Communication

- I can handle emails in a very responsive way

- Course email: ricecomp414514@gmail.com
  (please avoid sending emails to my personal account)

Communication

- I can handle emails in a very responsive way

- **Course email**: ricecomp414514@gmail.com
  (please avoid sending emails to my personal account)

- ~~e-Mailing list~~: We canvas + Piazza now
  (So if you are not registered, you will not get updates)

Communication

- I can handle emails in a very responsive way

- **Course email**: ricecomp414514@gmail.com
  (please avoid sending emails to my personal account)

- ~~e-Mailing list~~: We canvas + Piazza now
  (So if you are not registered, you will not get updates)

- **HWs**: will be sent to you via Canvas every week.
  (please do not distribute)

Notes

– I have started preparing notes for this course

Notes

- I have started preparing notes for this course

- Every week I will try to update every chapter; however I would appreciate any help with scribing throughout the semester

(Course website)

Length of each session

- There will be longer or shorter sessions

Length of each session

- There will be longer or shorter sessions

- Each week represents a Chapter

(This might be a optimistic/delusional; some chapters have more "meat" than others)

Length of each session

- There will be longer or shorter sessions

- Each week represents a Chapter

  (This might be a optimistic/delusional; some chapters have more "meat" than others)

- Any feedback is more than welcome

  (e.g., too much material vs. too little material)

Length of each session

- There will be longer or shorter sessions

- Each week represents a Chapter

  (This might be a optimistic/delusional; some chapters have more "meat" than others)

- Any feedback is more than welcome
  (e.g., too much material vs. too little material)

- In case I don't have the time to cover fully a session, I will decide whether you will read it yourself, or I will teach it the next time.

Any questions?

Setting up the background

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase
- Vector in p-dimensions: $x \in \mathbb{R}^p$

$$x = \left[x_1, x_2, \ldots, x_p\right]^\top$$

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase

- Vector in p–dimensions: $x \in \mathbb{R}^p$

$$x = \left[x_1, x_2, \ldots, x_p\right]^\top$$

- Some properties of vectors:

$$x + y = y + x, \quad x, y \in \mathbb{R}^p$$

(Commutative)

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase

- Vector in p-dimensions: $x \in \mathbb{R}^p$

$$x = \left[x_1, x_2, \ldots, x_p\right]^\top$$

- Some properties of vectors:

$$x + y = y + x, \quad x, y \in \mathbb{R}^p$$ (Commutative)

$$(x + y) + x = x + (y + z), \quad x, y, z \in \mathbb{R}^p$$ (Associative)

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase

- Vector in p-dimensions: $x \in \mathbb{R}^p$

$$x = \left[x_1, x_2, \ldots, x_p\right]^\top$$

- Some properties of vectors:

$$x + y = y + x, \quad x, y \in \mathbb{R}^p$$ (Commutative)

$$(x + y) + x = x + (y + z), \quad x, y, z \in \mathbb{R}^p$$ (Associative)

$$0 + x = x, \quad x \in \mathbb{R}^p$$

Vectors

- Notation convention: vectors = lowercase, matrices = uppercase
- Vector in p-dimensions: $x \in \mathbb{R}^p$

$$x = \left[x_1, x_2, \ldots, x_p\right]^\top$$

- Some properties of vectors:

$$x + y = y + x, \quad x, y \in \mathbb{R}^p$$ (Commutative)

$$(x + y) + x = x + (y + z), \quad x, y, z \in \mathbb{R}^p$$ (Associative)

$$0 + x = x, \quad x \in \mathbb{R}^p$$

$$\alpha(x + y) = \alpha x + \alpha y, \quad x, y \in \mathbb{R}^p$$ (Distributive)

Vectors

- Span of a set of vectors:

$$\text{span}\{x_1, x_2, \ldots, x_k\} = \{\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \mid \alpha_i \in \mathbb{R}, i = [1, k]\}$$

Vectors

- Span of a set of vectors:

$$\text{span}\{x_1, x_2, \ldots, x_k\} = \{\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \mid \alpha_i \in \mathbb{R}, i = [1, k]\}$$

- Linear independence:

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k = 0 \quad \Rightarrow \quad \alpha_i = 0, \ \forall i$$

Vectors

- Span of a set of vectors:

$$\text{span}\{x_1, x_2, \ldots, x_k\} = \{\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \mid \alpha_i \in \mathbb{R}, i = [1, k]\}$$

- Linear independence:

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k = 0 \quad \Rightarrow \quad \alpha_i = 0, \ \forall i$$

- How does k compare to p, the vector dimension?

Vectors

- Span of a set of vectors:

$$\text{span}\{x_1, x_2, \ldots, x_k\} = \{\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \mid \alpha_i \in \mathbb{R}, i = [1, k]\}$$

- Linear independence:

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k = 0 \quad \Rightarrow \quad \alpha_i = 0, \ \forall i$$

- How does k compare to p, the vector dimension?

- Inner product:

$$x^\top y = \langle x, y \rangle = \sum_{i=1}^{p} x_i \cdot y_i$$

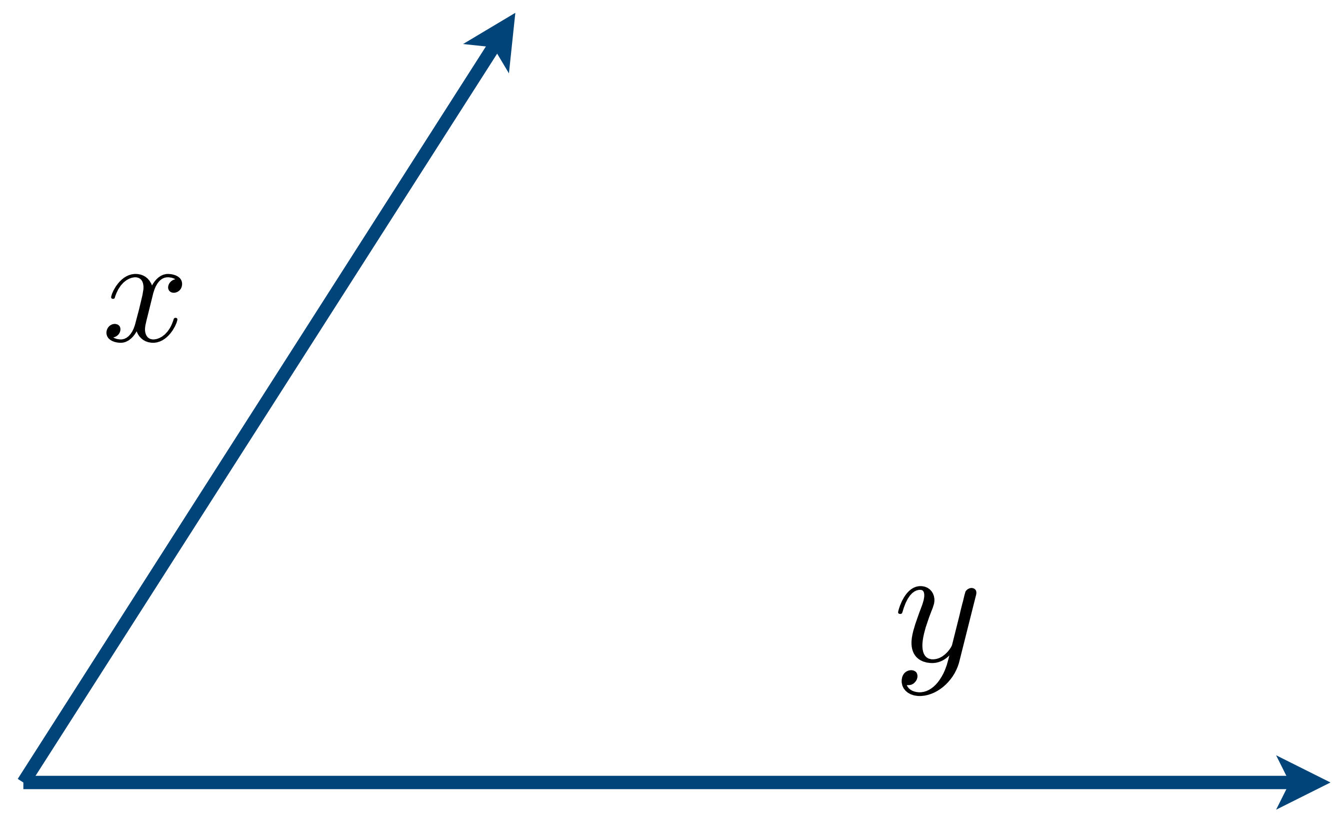
Vectors

– Inner production interpretation:

$$\langle x, y \rangle = \|x\| \cdot \|y\| \cdot \cos\theta$$

Vectors

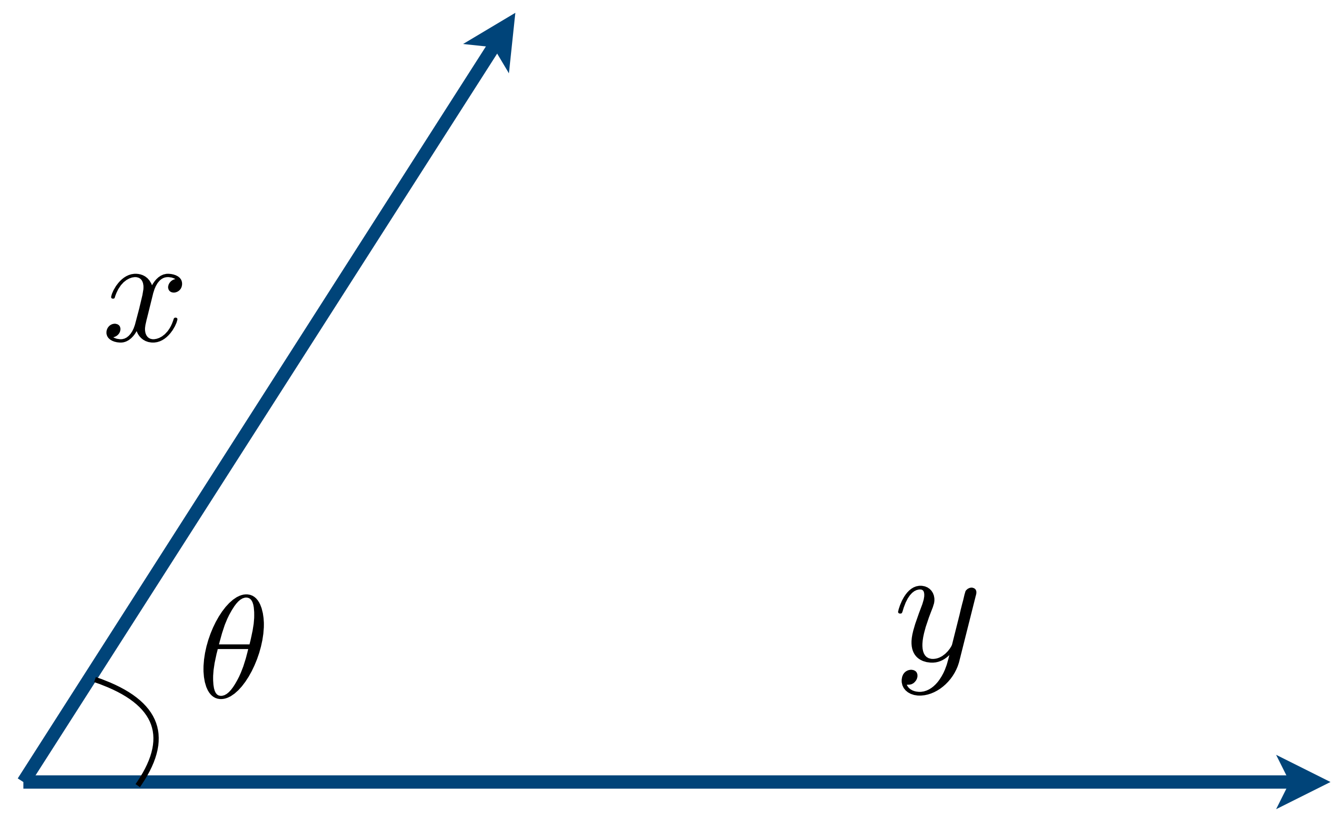
– Inner production interpretation:



$$\langle x, y \rangle = \|x\| \cdot \|y\| \cdot \cos\theta$$

Vectors

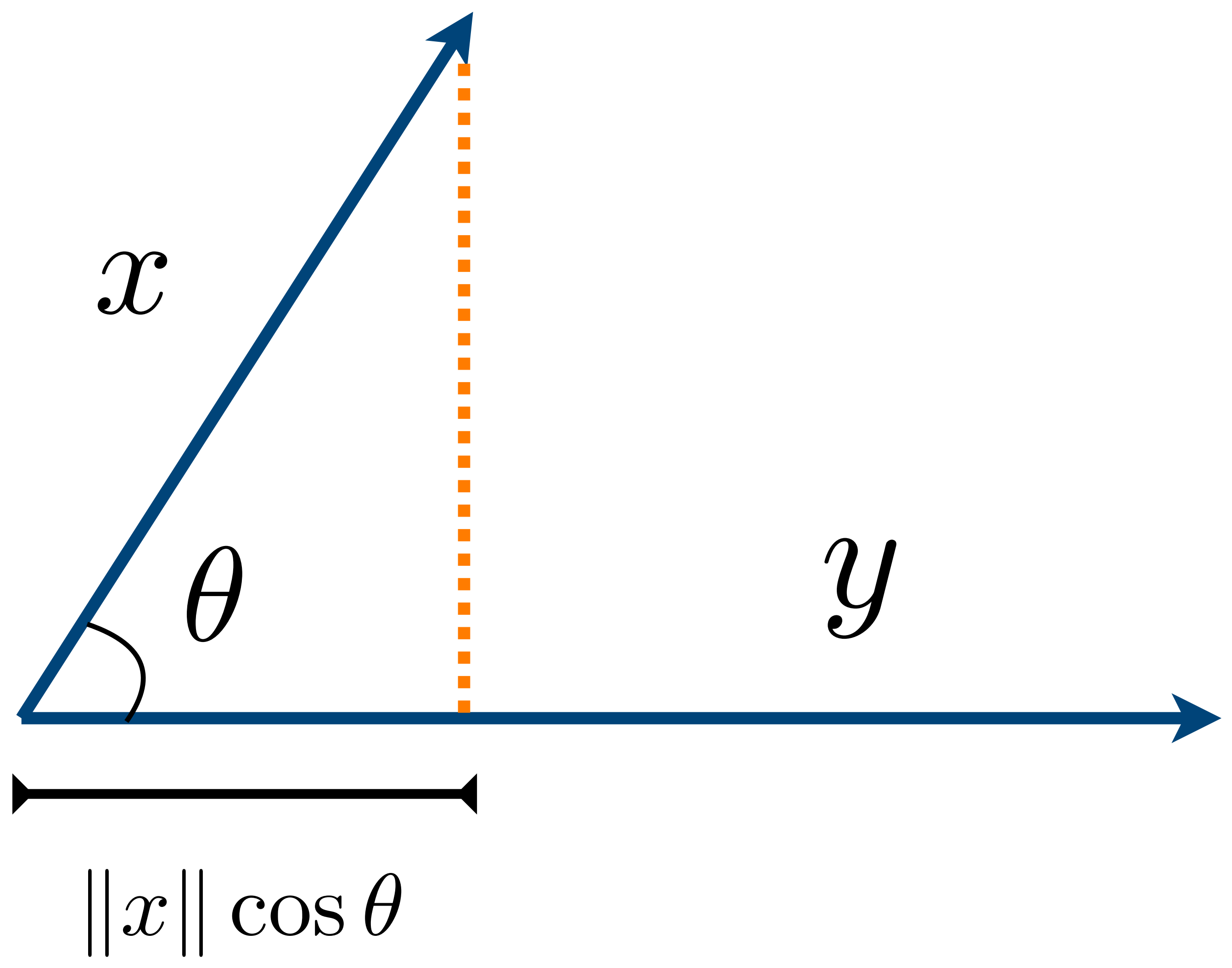
- Inner production interpretation:



$$\langle x, y \rangle = \|x\| \cdot \|y\| \cdot \cos\theta$$

Vectors

– Inner production interpretation:



$$\langle x, y \rangle = \|x\| \cdot \|y\| \cdot \cos\theta$$

Vectors

- Norms = notion of distance in multiple dimensions

Properties:

$$\|x\| \geq 0, \forall x \in \mathbb{R}^p$$

$$\|x\| = 0, \text{ iff } x = 0$$

$$\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in \mathbb{R}$$

$$\|x + y\| \leq \|x\| + \|y\|$$ (Triangle inequality)

$$|x^\top y| \leq \|x\| \|y\|$$ (Cauchy–Schwarz)

Vectors

- Norms = notion of distance in multiple dimensions

Properties:

$$\|x\| \geq 0, \forall x \in \mathbb{R}^p$$
$$\|x\| = 0, \text{ iff } x = 0$$
$$\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in \mathbb{R}$$
$$\|x + y\| \leq \|x\| + \|y\|$$ (Triangle inequality)
$$|x^\top y| \leq \|x\| \|y\|$$ (Cauchy−Schwarz)

- Standard vector norms:

$$\|x\|_2 = \sqrt{\sum_i x_i^2} \qquad \|x\|_1 = \sum_i |x_i| \qquad \|x\|_\infty = \max_i |x_i|$$

Vectors

- Norms = notion of distance in multiple dimensions

Properties:

$$\|x\| \geq 0, \forall x \in \mathbb{R}^p$$
$$\|x\| = 0, \text{ iff } x = 0$$
$$\|\alpha x\| = |\alpha|\|x\|, \forall \alpha \in \mathbb{R}$$
$$\|x + y\| \leq \|x\| + \|y\|$$ (Triangle inequality)
$$|x^\top y| \leq \|x\|\|y\|$$ (Cauchy–Schwarz)

- Standard vector norms:

$$\|x\|_2 = \sqrt{\sum_i x_i^2} \qquad \|x\|_1 = \sum_i |x_i| \qquad \|x\|_\infty = \max_i |x_i|$$

- Famous wanna-be norms: $\|x\|_0 = \text{card}(x)$

Matrices

- Matrix in m, n dimensions: $A \in \mathbb{R}^{m \times n}$

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

Matrices

- Matrix in m, n dimensions: $A \in \mathbb{R}^{m \times n}$

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

- Names: Square, tall, fat, zero, identity, diagonal

Matrices

- Matrix in m, n dimensions: $A \in \mathbb{R}^{m \times n}$

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

- Names: Square, tall, fat, zero, identity, diagonal
- Properties:

$$A + B = B + A,\ \forall A, B \in \mathbb{R}^{m \times n}$$
$$(A + B) + C = A + (B + C),\ \forall A, B, C \in \mathbb{R}^{m \times n}$$
$$A + 0 = 0 + A,\ \forall A \in \mathbb{R}^{m \times n}$$
$$(A + B)^\top = A^\top + B^\top,\ \forall A, B \in \mathbb{R}^{m \times n}$$

Matrices

– Matrix multiplication： $C = AB$ where $C \in \mathbb{R}^{m\times p}$, $A \in \mathbb{R}^{m\times n}$, and $B \in \mathbb{R}^{n\times p}$

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1p} \\ C_{21} & C_{22} & \cdots & C_{2p} \\ \vdots & & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mp} \end{bmatrix} = C = AB = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & & \ddots & \vdots \\ B_{n1} & B_{n2} & \cdots & B_{np} \end{bmatrix}$$

Matrices

- Matrix multiplication: $C = AB$ where $C \in \mathbb{R}^{m\times p}$, $A \in \mathbb{R}^{m\times n}$, and $B \in \mathbb{R}^{n\times p}$

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1p} \\ C_{21} & C_{22} & \cdots & C_{2p} \\ \vdots & & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mp} \end{bmatrix} = C = AB = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & & \ddots & \vdots \\ B_{n1} & B_{n2} & \cdots & B_{np} \end{bmatrix}$$

- Special cases: vector inner product, matrix–vector mult., outer product

Matrices

– Matrix multiplication： $C = AB$ where $C \in \mathbb{R}^{m\times p}$, $A \in \mathbb{R}^{m\times n}$, and $B \in \mathbb{R}^{n\times p}$

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1p} \\ C_{21} & C_{22} & \cdots & C_{2p} \\ \vdots & & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mp} \end{bmatrix} = C = AB = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & & \ddots & \vdots \\ B_{n1} & B_{n2} & \cdots & B_{np} \end{bmatrix}$$

– Special cases： vector inner product, matrix–vector mult., outer product

– Properties：

$$(AB)C = A(BC),\ \forall A, B, C$$

$$\alpha(AB) = (\alpha A)B,\ \forall A, B$$

$$A(B + C) = AB + AC,\ \forall A, B, C$$

$$(AB)^\top = B^\top A^\top,\ \forall, A, B$$

$$AB \neq BA$$

Matrices

- Inner product:

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

Matrices

- Inner product:

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

- Example:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & & \ddots & \vdots \\ B_{m1} & B_{n2} & \cdots & B_{mn} \end{bmatrix}$$

Matrices

- Inner product：

$$\langle A, B \rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

- Example：

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & & \ddots & \vdots \\ B_{m1} & B_{n2} & \cdots & B_{mn} \end{bmatrix} \longrightarrow A^\top B = \begin{bmatrix} \sum_{i=1}^m A_{i1}\cdot B_{i1} & \cdots & \cdots & \cdots \\ \cdots & \sum_{i=1}^m A_{i2}\cdot B_{i2} & \cdots & \cdots \\ \vdots & & \ddots & \vdots \\ \cdots & \cdots & \cdots & \sum_{i=1}^m A_{in}\cdot B_{in} \end{bmatrix}$$

Matrices

- Inner product:

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

- Example:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & & \ddots & \vdots \\ B_{m1} & B_{n2} & \cdots & B_{mn} \end{bmatrix} \longrightarrow A^\top B = \begin{bmatrix} \sum_{i=1}^m A_{i1}\cdot B_{i1} & \cdots & \cdots & \cdots \\ \cdots & \sum_{i=1}^m A_{i2}\cdot B_{i2} & \cdots & \cdots \\ \vdots & & \ddots & \vdots \\ \cdots & \cdots & \cdots & \sum_{i=1}^m A_{in}\cdot B_{in} \end{bmatrix}$$

$$\mathrm{Tr}(A^\top B) = \sum_{i=1}^m A_{i1}\cdot B_{i1} + \sum_{i=1}^m A_{i2}\cdot B_{i2} + \cdots + \sum_{i=1}^m A_{in}\cdot B_{in}$$

Matrices

– Inner product:

$$\langle A, B \rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

– Example:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & & \ddots & \vdots \\ B_{m1} & B_{n2} & \cdots & B_{mn} \end{bmatrix} \longrightarrow A^\top B = \begin{bmatrix} \sum_{i=1}^m A_{i1}\cdot B_{i1} & \cdots & \cdots & \cdots \\ \cdots & \sum_{i=1}^m A_{i2}\cdot B_{i2} & \cdots & \cdots \\ \vdots & & \ddots & \vdots \\ \cdots & \cdots & \cdots & \sum_{i=1}^m A_{in}\cdot B_{in} \end{bmatrix}$$

$$\mathrm{Tr}(A^\top B) = \mathtt{vec}(A)^\top \mathtt{vec}(B) = \langle \mathtt{vec}(A), \mathtt{vec}(B) \rangle$$

Matrices

– Inner product:

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

Matrices

– Inner product：

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

– Rank of a matrix：maximum # of independent columns or rows

Matrices

- Inner product：

$$\langle A, B\rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m\times n}$$

- Rank of a matrix：maximum # of independent columns or rows

- Nullspace of a matrix： $\{x \mid Ax = 0\}$

Matrices

– Inner product：

$$\langle A, B \rangle = \mathrm{Tr}(A^\top B) = \mathrm{Tr}(B^\top A), \forall A, B \in \mathbb{R}^{m \times n}$$

– Rank of a matrix：maximum # of independent columns or rows

– Nullspace of a matrix： $\{x \mid Ax = 0\}$

– Positive semi–definite matrices： $A \succeq 0$

1. $A \in \mathbb{R}^{n \times n}$
2. A is symmetric
3. $x^\top Ax \geq 0,\ \forall x \in \mathbb{R}^n,\ x \neq 0$

Matrices

- Matrix singular value decomposition: $A \in \mathbb{R}^{m \times n}$

$$A = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,\ U \in \mathbb{R}^{m\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{n\times r} \qquad r \leq \{m, n\}$$

Matrices

- Matrix singular value decomposition: $A \in \mathbb{R}^{m\times n}$

$$A = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,\ U \in \mathbb{R}^{m\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{n\times r} \qquad r \leq \{m, n\}$$

- $\mathrm{rank}(A) = r \leq \min\{m, n\}$

Matrices

- Matrix singular value decomposition: $A \in \mathbb{R}^{m \times n}$

$$A = U \Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,\ U \in \mathbb{R}^{m \times r}, \Sigma \in \mathbb{R}^{r \times r}, V \in \mathbb{R}^{n \times r} \qquad r \leq \{m, n\}$$

- $\mathrm{rank}(A) = r \leq \min\{m, n\}$

- $u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$ are the left and right singular vectors

Matrices

- Matrix singular value decomposition: $A \in \mathbb{R}^{m \times n}$

$$A = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,\ U \in \mathbb{R}^{m\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{n\times r} \qquad r \leq \{m, n\}$$

- $\mathrm{rank}(A) = r \leq \min\{m, n\}$
- $u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$ are the left and right singular vectors
- $\Sigma = \texttt{diag}(\sigma_1, \ldots, \sigma_r)$ contains singular values where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$

Matrices

- Matrix singular value decomposition: $A \in \mathbb{R}^{m \times n}$

$$A = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top, \; U \in \mathbb{R}^{m \times r}, \Sigma \in \mathbb{R}^{r \times r}, V \in \mathbb{R}^{n \times r} \qquad r \leq \{m, n\}$$

- $\mathrm{rank}(A) = r \leq \min\{m, n\}$
- $u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$ are the left and right singular vectors
- $\Sigma = \mathtt{diag}(\sigma_1, \ldots, \sigma_r)$ contains singular values where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$
- Left and right singular vectors are orthogonal: $U^\top U = I$ and $V^\top V = I$

Matrices

- Norms:

$$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}$$

(Frobenius norm)

$$\|A\|_* = \sum_{i}^{r} \sigma_i$$

(Nuclear norm)

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

(Spectral norm)

..there are more norms to worry about (e.g., operator norms)
but we will skip them here..

Matrices

– Matrix inverses are defined on square matrices

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank
  - A has empty nullspace

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank
  - A has empty nullspace
  - The equation $Ax = 0$ has only the trivial solution $x = 0$

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank
  - A has empty nullspace
  - The equation $Ax = 0$ has only the trivial solution $x = 0$
  - The linear system $Ax = b$ has a unique solution

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank
  - A has empty nullspace
  - The equation $Ax = 0$ has only the trivial solution $x = 0$
  - The linear system $Ax = b$ has a unique solution
  - The columns and rows of A are linearly independent

Matrices

- Matrix inverses are defined on square matrices
- Matrix inverse definition as a collection of properties:
  - A is full rank
  - A has empty nullspace
  - The equation $Ax = 0$ has only the trivial solution $x = 0$
  - The linear system $Ax = b$ has a unique solution
  - The columns and rows of A are linearly independent
  - There exists a square matrix, A^{-1}, such that $A^{-1}A = AA^{-1} = I$

Overview

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{C} \end{aligned}$$

Overview

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

- The set of points that satisfy the constraint is called the feasible set

Overview

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

- The set of points that satisfy the constraint is called the feasible set
- Finding the point(s) that satisfies the constraint a**nd** minimizes the objective is the task of optimization

Overview

$$\min_{x} \quad f(x)$$

$$\text{s.t.} \quad \sout{x \in \mathcal{C}}$$

Unconstrained optimization

- The set of points that satisfy the constraint is called the feasible set
- Finding the point(s) that satisfies the constraint **and** minimizes the objective is the task of optimization

Disclaimer

Optimization is generally **unsolvable**..

(Closed form expressions vs. Iterative methods)

(Naive solvers that work well on specific cases)

This course focuses on iterative methods

(..or what is the difference to specific, deterministic algorithms)

- General procedure

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

1. Start from an initial point x_0.
2. Given an oracle $\mathcal{O}$, make queries to $\mathcal{O}$.
3. Obtain oracle's answer and exploit such a knowledge to reach to a new point as a putative solution.
4. Repeat steps 2.-3. until we get to a point where we are satisfied, according to a stopping criterion.

This course focuses on iterative methods

(..or what is the difference to specific, deterministic algorithms)

- General procedure

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

1. Start from an initial point x_0.
2. Given an oracle $\mathcal{O}$, make queries to $\mathcal{O}$.
3. Obtain oracle's answer and exploit such a knowledge to reach to a new point as a putative solution.
4. Repeat steps 2.-3. until we get to a point where we are satisfied, according to a stopping criterion.

- Key points that need to be addressed?

This course focuses on iterative methods

(..or what is the difference to specific, deterministic algorithms)

– General procedure

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

1. Start from an initial point x_0.
2. Given an oracle $\mathcal{O}$, make queries to $\mathcal{O}$.
3. Obtain oracle's answer and exploit such a knowledge to reach to a new point as a putative solution.
4. Repeat steps 2.-3. until we get to a point where we are satisfied, according to a stopping criterion.

– Key points that need to be addressed?

– The notion of the Black–Box model

This course focuses on iterative methods

(..or what is the difference to specific, deterministic algorithms)

- General procedure

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{C}$$

1. Start from an initial point x_0.
2. Given an oracle $\mathcal{O}$, make queries to $\mathcal{O}$.
3. Obtain oracle's answer and exploit such a knowledge to reach to a new point as a putative solution.
4. Repeat steps 2.-3. until we get to a point where we are satisfied, according to a stopping criterion.

- Key points that need to be addressed?
- The notion of the Black−Box model

Common types of oracles. Some common types of oracles are:

- *Zeroth-order oracle*: Given a query point x, the oracle only returns $f(x)$.
- *First-order oracle*: Given a query point x, the oracle returns $f(x)$, and its gradient at x, $\nabla f(x)$ (assuming differentiability).
- *Second-order oracle*: Given a query point x, the oracle returns $f(x)$, its gradient $\nabla f(x)$, and the Hessian at x, $\nabla^2 f(x)$ (assuming twice-differentiability).

Conclusion

- We have set up background and notation w.r.t. linear algebra
- We saw a toy example where non-convex operations happen

Conclusion

- We have set up background and notation w.r.t. linear algebra

- We saw a toy example where non-convex operations happen

Next lecture

- Brief introduction to convex optimization and related topics

Demo