EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 1

Course information

- EECS 275 Matrix Computation
- Lecture hours: Monday/Wednesday 4:00pm-5:15pm
- Office hours: Tuesday 4:30pm-5:30pm
- Office: SE 258
- Lab session: SE 138, Monday 1:00pm-3:50pm
- Lecture notes and schedule: `http://faculty.ucmerced.edu/mhyang/course/eecs275`
- Email: `mhyang@ucmerced.edu`

What is this course about?

- Cover algorithms and techniques for matrix computation and analysis
- Focus on computational algorithms rather than rigorous mathematical proofs or numerical techniques
- Analyze data represented as vectors and matrices
- Demonstrate their use with applications
- Solving

$$A\mathbf{x} = \mathbf{b},$$
$$A \in \mathbb{R}^{m \times n},\ \mathbf{x} \in \mathbb{R}^{n},\ \mathbf{b} \in \mathbb{R}^{m}$$
$$A = [\mathbf{a}_1, \ldots, \mathbf{a}_n],\ \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},\ \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$
$$\mathbf{b} = \sum_{j=1}^{n} x_j \mathbf{a}_j$$

Examples

- Given a set of images/documents, how do you find a compact representation?
- Given a set of input/output pairs, how do you learn their mapping function?
- Given an image sequence, how do you model the underlying dynamics (temporal correlation)?
- Given a linear system (a set of linear constraints), how do you solve it?
- Given a set of data, how do you determine metrics to compute their similarities?
- Given a set of data, how do you visualize their relationship?
- Given a set of data, how do you cluster them?
- Given a large data set, how do you develop efficient algorithms to analyze data?

Topics

- Fundamentals: Introduction, vector space, vector and matrix norms, orthogonalization, covariance and Gram matrices, multivariate Gaussian
- Matrix decomposition: LU decomposition, QR decomposition, Cholesky decomposition, Schur decomposition, eigen decomposition, singular value decomposition, factorization
- System of equations: least squares, linear programming, linear dynamical system, stochastic matrix, random walk
- Statistical models: principal component analysis, factor analysis
- Eigenvalue problems: Lanczos method, power method
- Matrix inverse: Sherman-Morrison-Woodbury formula, Kailath variant, pseudo inverse, approximation

Topics (cont'd)

- Sequential update: matrix update and downdate, subspace update
- Matrix approximation: sparse matrix approximation, low rank approximation, Nystrom method
- Emerging topics: non-negative matrix factorization, compressive sensing, randomized algorithms, large scale matrix computation
- Applications

Textbook

- Textbook:
  - *Numerical Linear Algebra* by Trefethen and Bau (SIAM Press)
- All the lectures and papers are available on the course website
- Reference:
  - *Matrix Computation* by Gene Golub and Charles Van Loan
  - *Matrix Algebra From a Statistician's Perspective* by David Harville
  - *Matrix Analysis and Applied Linear Algebra* by Carl Meyer
  - *The Matrix Cookbook* by Kaare Petersen and Michael Pedersen

Prerequisites

- Basic knowledge in linear algebra
- Basic knowledge in probability and statistics
- Proficiency in some programming language: MATLAB and others (e.g., C++, R)

Requirements

- Literature review and critique: read conference/journal papers and submit critiques
- Term project: work on term project individually or in two-member groups
- Oral presentations: make one presentation on project overview and progress, as well as one final presentation
- Final project report: submit one project report and code with demos
- Formats and details regarding all the above-mentioned items will be available on the course web site

Grading

- Grading
  - 20% Homework (e.g., exercise, derivation)
  - 10% Midterm presentation
  - 25% Midterm report
  - 15% Final presentation
  - 30% Term project report
- You will get excellent grade if you work hard

What you could expect from this course and me

- Algorithms and techniques in matrix computation and their applications
  - I will discuss the pros/cons of algorithms and demosntrate their merits with their applications to vision and learning problems
- Advice on term project
  - Everyone needs to discuss your project proposal with me
  - You are encouraged to contact me for questions or suggestions regarding your term project

What I expect from you

- Ask questions!
- Send me the typos/mistakes on the lecture notes (I am only human...)
- Think critically about papers
- Discuss project ideas with me
- Work hard on term project

Term project

- Do not simply re-implement existing algorithms
  - Although it is a good way to learn a subject, it is not sufficient to simply reproduce results of others
- Think critically and surpass the state-of-the-art
  - What is the fun of just repeating other's works?
  - Push the envelope
- Numerous ways to improve existing algorithms
  - robustness, efficiency, accuracy
  - simpler approaches with better results
- Publish your research findings!
- Make the code and data sets available
- Suggestions
  - Discuss with your ideas with me and other classmates
  - Work in a small group (at most 2 people)
  - Read papers and write critiques weekly
  - Tinkering with your ideas with experiments
  - Re-search

Some suggested projects

- Human detection:
  - how to locate humans and identify body parts as well as pose? in 2D or 3D? using what features? single or multiple views?
- Human tracking:
  - how to track human body parts? in 2D or 3D? single or multiple views?
- Visual tracking:
  - how to track articulated objects (e.g., arms, fingers)?
  - how to handle occlusions?
  - how to adapt models?
  - how to recover from failure?
  - how to combine detector with trackers?
- Texture synthesis:
  - how to synthesize videos with multiple motion textures (e.g., fire and water)?
  - how to use the learned dynamics to classify visual scenes?

Some suggested projects (cont'd)

- How to learn compact representation for object detection?
- Image features:
  - how to find reliable features for application X?
  - how to extend existing methods into spatio-temporal domains?
- 3D reconstruction:
  - how to infer 3d structure from moving objects under lighting variations?
- Large graph partition
- Small world phenomenon
- Visualization
- Sky is the limit!

Homework

- Get yourself familiar with MATLAB
- Refresh yourself with linear algebra