18.409: An Algorithmist’s Toolkit
Jonathan Kelner

New Room?
- Starting on Tuesday, we may have a different room
- I’ll post any info on the course website and e-mail people on signup sheet

Today
- Administrivia
- Course overview
- Start spectral graph theory
Me

- E-mail: kelner@mit.edu
- Offices:
  - Math: 2-372
  - CSAIL: 32-G618
- Office hours
  - Will be settled once people know their schedules
  - Until then, make an appointment by e-mail
- Course website:
  - http://stellar.mit.edu/S/course/18/fa09/18.409/
  - Easier:
    - http://course.mit.edu/18.409
- Class will have a TA

Course Goals

- To learn a collection of powerful (and interrelated) mathematical techniques for algorithm design
  - Topics picked to be both mathematically interesting and practically useful
  - Will sometimes have long mathematical interludes, but always with proportional algorithmic payoffs
- To be able to apply these tools directly to your research
  - Whether your work is theoretical or applied
  - I'll suggest open research questions whenever possible

Requirements and Grading

- I'll try to keep it light
- Scribe notes
  - Aim for quick turnaround
- Problem sets
  - One per major topic
  - Don't have to do all problems

Collaboration Policy

- **Collaboration is encouraged on problems sets**
- Think about the problems yourself first
- Write solutions individually
- Must understand anything you hand in
- Acknowledge collaborators and outside sources
  - Please try not to use outside sources
Listeners

- I’m happy to have people who aren’t taking the class come as listeners
- **If you are planning on listening regularly, please do so officially**
  - It makes me look popular
  - It makes it easier to re-offer similar courses in the future (and to get the right sized room now)

Prerequisites

- Graduate course, so assuming significant level of mathematical maturity
- Not a lot of formal prerequisites though
  - Multivariate calculus
  - Linear algebra
  - Basic algorithms
  - Basic probability
  - Some algorithms beyond intro would help, but not strictly necessary
- If you’re unsure, come talk to me

Main Topics

- Spectral graph theory (8 lectures)
- Iterative methods for linear algebra (2 lectures)
- Convex geometry (7 lectures)
- Multiplicative weights (2 lectures)
- Lattices and basis reduction (3 lectures)
- LPs and SDPs for approximating NP-hard problems (3 lectures)
Spectral Graph Theory

- Develop mathematical theory
  - Graph Laplacians and their eigenvalues
  - Isoperimetric and Cheeger inequalities

- Connect to other fields
  - Markov processes and random walks
  - Continuous differential and convex geometry
  - Expanders and random graphs

- Applications
  - Graph cutting
  - Clustering
  - Approximate counting
  - Disjoint path problems
  - Routing
  - Graph drawing
  - Coding theory (maybe)
  - How Google works (maybe)

Iterative methods for linear algebra

- Use geometric information to quickly solve linear systems and eigenvalue problems
- Basic iterative methods
- Lanczos algorithm
- Conjugate gradients
- Preconditioning
- Speeding up iterative methods with spectral graph theory

Convex geometry

- Mathematical theory
  - Geometric properties of high-dimensional convex bodies
  - Fritz John's theorem and isotropy
  - Brunn-Minkowski and isoperimetric inequalities
  - Concentration of measure

- Connect to other fields
  - Probability theory and large deviations
  - Spectral graph theory

- Applications
  - Volume computation
  - Convex programming

Multiplicative Weights

- Mathematical theory
  - Multiplicative updates meta-algorithm and its analysis
  - Geometric and interpretation of the method

- Connect to other fields
  - Primal-dual methods
  - Learning theory

- Applications
  - Solving zero-sum games
  - Fast approximation algorithms for graph problems
  - Online algorithms
  - Boosting
  - Complexity theory/pseudorandomness (maybe)
Lattices and basis reduction

Mathematical theory
- Basic properties of lattices
- Minkowski’s theorem
- The LLL algorithm for lattice basis reduction

Connect to other fields
- Computational algebra
- Convex geometry

Applications
- Solving low-dimensional integer programs
- Solving some NP-hard problems in practice
- Breaking cryptosystems

LPs and SDPs for approximating NP-hard problems

- Linear and semidefinite programming relaxations of NP-hard problems
- Rounding techniques
- Primal-dual methods

What is Spectral Graph Theory?

- Associate a matrix to a graph
- Diagonalize the matrix
- Hope something useful happens
Reminders about Eigenvalues

- I'll assume you've seen eigenvalues before
  - I might be willing to give a linear algebra session review or make a handout if there's sufficient interest

**Definition:** Let $M$ be an $n \times n$ matrix. Suppose

$$Mx = \lambda x,$$

$$x \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}$$

We call:
- $x$ an *eigenvector*
- $\lambda$ its *eigenvalue*

Reminders about Eigenvalues (cont.)

- **If $M$ is symmetric:**
  - You can *diagonalize* it:
    $$M = V \Lambda V^T$$
  - $V$ is orthogonal ($V^T V = \text{Id}$)
    - Columns of $V$ are eigenvectors $v_1, \ldots, v_n$
  - $\Lambda$ is diagonal
    - Entries are eigenvalues $\lambda_1, \ldots, \lambda_n$
  - So
    $$M = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

Reminders about Eigenvalues (cont.)

- **If $M$ is symmetric:**
  - If $v$ and $w$ are eigenvectors of $M$ with *different eigenvalues* then $v \cdot w = 0$
  - If $v$ and $w$ have same eigenvalue, any $q=av+bw$ is an eigenvector too
  - Call span of vcts with same eigenvalue an *eigenspace*
  - $M$ has a full orthonormal basis of eigenvectors $v_1, \ldots, v_n$
  - All eigenvalues and eigenvectors are *real*
  - All bets are off if $M$ is not symmetric

Matrices for Graphs

- We'll look at a couple different matrices to associate with graphs
  - Adjacency matrix
  - Random walk matrix
  - Laplacian matrix
  - Normalized Laplacian matrix
- Different ones are useful at different times
- Today
Some Notation
- From now on, unless I say otherwise:
  - $G = (V,E)$ is a graph
  - $G$ is undirected, unweighted, no multiple edges or self loops
  - $n = \text{num vertices}$
  - $m = \text{num edges}$

The Adjacency Matrix
- **Definition**: Adjacency matrix $A = A_G$ is $n \times n$ matrix given by
  
  $A_{i,j} = \begin{cases} 
  1 & \text{if } (i, j) \in E \\
  0 & \text{otherwise} 
  \end{cases}$

- Clearly symmetric
The Adjacency Matrix (cont.)

\[
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & 1 & 0 & 1 \\
2 & 0 & 1 & 0 & 1 \\
3 & 1 & 0 & 0 & 0 \\
4 & 0 & 1 & 1 & 0 \\
5 & 0 & 0 & 0 & 1 \\
\end{array}
\]

The Adjacency Matrix (cont.)

\[
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 0 \\
3 & 0 & 1 & 0 & 1 \\
4 & 1 & 0 & 1 & 0 \\
5 & 0 & 0 & 1 & 0 \\
\end{array}
\]

The Laplacian

- For our purposes, a slightly different matrix will often be nicer
- **Definition:** Laplacian matrix \( L = L_G \) is nxn matrix given by

\[
L_{i,j} = \begin{cases} 
-1 & \text{if } (i, j) \in E \\
\frac{d_i}{d_j} & \text{if } i = j \\
0 & \text{otherwise}
\end{cases}
\]

where \( d_i \) is degree of \( i \)th vertex
- Also clearly symmetric
The Laplacian (cont.)

- Could also say:
  Let $D = D_G$ be diagonal matrix with $i^{th}$ diagonal entry equal to degree of $v_i$

$$L_G = D_G - A_G$$
What Does $L_G$ Do to a Vector?

$$[L_G x]_i = \deg(i) \cdot x(i) - \sum_{(i,j) \in E} x(j)$$
$$= \deg(i) \cdot \left( x(i) - \frac{\sum_{(i,j) \in E} x(j)}{\deg(i)} \right)$$
$$= \deg(i) \cdot (x(i) - \text{average of } x \text{ on } i's \text{ neighbors})$$

Implies that for any graph $G$,

$L_G 1 = 0$

So $x=1$ is an eigenvector with eigenvalue 0.

- We’ll see later that all other eigenvalues are $\geq 0$ ( $>0$ for a connected graph)
- So if $\lambda_1 \geq \cdots \geq \lambda_n$ with respective eigenvectors $v_1, \ldots, v_n$
  - $v_1 = 1$, $\lambda_1 = 0$
  - A lot of info is contained in first few nontrivial eigenvectors.
  - $v_2$ and $v_3$ are vectors and thus each gives a map from $G \rightarrow \mathbb{R}$.
  - Let’s use them as coordinates and see what happens.

MATLAB PICTURES