1. Overview
2. Administration

Algorithms.

Undergraduate.
This class.

1. Classical.
   Modern.
   Flavor of the week?
2. Cleanly Stated Problems. Shortest Paths, max-flow, MST.
   Vaguely stated problems!
   Address problems; messy or not.
3. Solutions: effective precise bounds!
   Ineffective ..imprecise!
   Analysis sometimes based on modelling world.
   Heuristic, in practice.
5. Techniques tend to be Combinatorial.
   Probabilistic, linear algebra methods, continuous.

Example Problem: clustering.

Example: recommendations.

Sarah Palin likes True Grit (the old one.)
Sarah Palin doesn’t like The Social Network.
Sarah Palin doesn’t like Black Swan.
Sarah Palin likes Sarah Palin on Discovery channel.

Hillary Clinton doesn’t like True Grit (the old one.)
Hillary Clinton likes The Social Network.
Hillary Clinton likes Black Swan.

Should you recommend the discovery channel to Hillary?
What about you?
Are you Hillary? Are you Sarah? A bit of both?

High dimensional data: dimension for each movie.
More than three dimensions!
Nearest neighbors. Principal Components methods.
Topic Models.
Reasoning about these methods.

Image Segmentation

Which region? Normalized Cut: Find $S$, which minimizes

$$\frac{w(S, \overline{S})}{w(S) \times w(\overline{S})}.$$ 

Ratio Cut: minimize

$$\frac{w(S, \overline{S})}{w(S)}.$$ 

$w(S)$ no more than half the weight. (Minimize cost per unit weight that is removed.)
Either is generally useful!
Linear Systems.

Revolution!
Physical Simulation. Airflow.
Solve $Ax = b$.
How long?
$n \times n$ matrix $A$.
Middle School: substitution, adding equations ...
Time: $O(n^2)$.
Now: $\tilde{O}(m)$. Hmmm. What's that tilde?

Other Algorithmic Techniques

Sketching:
Large stream of data: $a_1, a_2, \ldots$
Find digest.
Graphs: Sparse graph.
Data: average, statistics.
Points: center point, k-medians, .
High Dimensional optimization.
Gradient Descent. Convexity.
Linear Algebra.
Eigenvalues.
Semidefinite Programming.
Dueling Subroutines. Duality.
Lower bounding, upper bounder.
Upper uses lower’s evidence to find better solutions.
Lower uses upper’s evidence to prove better lower bounds.

Path Routing.

Given $G = (V,E),$ $(s_1,t_1),\ldots,(s_k,t_k)$, find a set of $k$ paths connecting $s_i$ and $t_i$ and minimize max load on any edge.

Terminology

Routing: Paths $p_1, p_2, \ldots, p_k$, $p_i$ connects $s_i$ and $t_i$.
Congestion of edge, $e$: $c(e)$
number of paths in routing that contain $e$.
Congestion of routing: maximum congestion of any edge.
Find routing that minimizes congestion (or maximum congestion.)

CS270: Administration.

1. Staff:
Satish Rao
Di Wang
3. Assessment.
   3.1 Homeworks (40%).
   Homework 1 out tonight/tomorrow.
   3.2 1 Homework/Midterm (25 %)
   3.3 Project (35%)
   Groups of 2 or 3.
   Connect research to class.
   Or explore/digest a topic from class.
3.4 No Discussion this week.

Algorithms?

Route along any path.
Feasible...but good?
How far from optimal could it be?
(A) It is optimal!
(B) A factor of two.
(C) A factor of $k$, in general.
(C) and (A).

Stupid..also depth first search lexicographically!
Route along shortest path! Duh.
Optimal use of “resources”..or edges.
Shortest Path Routing.
minimizes $\sum_i \ell(p_i)$.
Total congestion: $\sum_e c(e)$
where $c(e)$ congestion of edge.
Why?
Let $\ell(p_i)$ be the length of path $p_i$.

(A) $\sum_i \ell(p_i) = \sum_e c(e)$?
(B) $\sum_i \ell(p_i) > \sum_e c(e)$?
(C) $\sum_i \ell(p_i) < \sum_e c(e)$?

(A). Proof?
Path $i$ uses $\ell(p_i)$ edges.
Edge used by $c(e)$ paths.
Totals be the same.
$\sum_i \ell(p_i) = \sum_e \sum_{p_i \ni e} 1 = \sum_e c(e)$
Shortest path routing minimizes total congestion.

Another problem.

Given $G = (V,E)$, $(s_1,t_1),\ldots,(s_k,t_k)$, find a set of $k$ paths assign one unit of "toll" to edges to maximize total toll for connecting pairs.

Assign $\frac{1}{e}$ on each of 11 edges.
Total toll: $\frac{1}{2} + \frac{1}{2} + \frac{3}{2} = \frac{3}{2}$
Can we do better?
Assign 1/2 on these two edges.
Total toll: $\frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{1}{2}$

Shortest Path Routing and Congestion.

Minimize each path length minimizes total congestion.
Also minimizes average: $\frac{1}{n} \sum_e c(e)$. Just a scaling!
Average load is lower bound on the lowest max congestion!
Shortest path routing minimizes average load.
Does it minimize maximum load?

Toll problem and Routing problem.

Given $G = (V,E)$, $(s_1,t_1),\ldots,(s_k,t_k)$, find a set of $k$ paths assign one unit of "toll" to edges to maximize total toll for connecting pairs.
Possible solution: $\frac{1}{m}$ on each edge.
Toll collected: $\geq \frac{\sum d(s_i,t_i)}{m}$.
Familiar?
Find $d: e \rightarrow R$ with $\sum_e d(e) = 1$ which maximizes

$\sum_{i,j} d(s_i,t_j)$

$d(s_i,t_j)$- shortest path between $s_i$ and $t_j$.
$d(e)$ suggests a weighted average.
Remember uniform average congestion is lower bound on congestion of routing!
Optimal toll solution (weighted average congestion) is lower bound on congestion.

Proving lower bound: notation.

d(e) - toll assigned to edge $e$.
d(p) - total toll assigned to path $p$.
d(u,v) - total assigned to shortest path between $u$ and $v$. d(·) - polymorphic: edges, paths, pairs.
Proving lower bound.

Routing solution: \( p_i \) connects \((s_i, t_i)\) and has length \( d(p_i)\).

- \( c(e) \)- congestion on edge \( e \) under routing.
- Max \( c(e) \)?
- \[ \max_e c(e) \geq \sum_e c(e) d(e) \text{ since } \sum_e d(e) = 1. \]

\[
\sum_i d(p_i) = \sum_i \sum_{p_i} d(e) = \sum_i \sum_{p_i} c(e) d(e) = \sum_i d(p_i) \]
\[
\sum_i d(p_i) = \sum_i \sum_{p_i} d(e) = \sum_i \sum_{p_i} 1 \]
\[
= \sum_i \sum_{p_i} c(e) = \sum_i d(p_i) c(e). \]

Routing solution cost \( \geq \) Any toll solution cost.

Toll is lower bound.

From before:
- Max bigger than minimum weighted average:
- \[ \max_e c(e) \geq \sum_e c(e) d(e) \]
- Total length is total congestion: \[ \sum_e c(e) d(e) = \sum_i d(p_i) \]
- Each path, \( p_i \), in routing has length \( d(p_i) \geq d(s_i, t_i) \).

\[
\max_e c(e) \geq \sum_e c(e) d(e) = \sum_i d(p_i) \geq \sum_i d(s_i, t_i). \]

A toll solution is lower bound on any routing solution.
Any routing solution is an upper bound on a toll solution.

Algorithm.

Assign tolls.
- How to route? Shortest paths!
- Assign routing.
- How to assign tolls? Higher tolls on congested edges.
- Toll: \( d(e) \propto 2^{c(e)} \).

Equilibrium:
- The shortest path routing has \( d(e) \propto 2^{c(e)} \).
- The routing does not change, the tolls do not change.

Getting to equilibrium.

Maybe no equilibrium!

Approximate equilibrium:
- Each path is routed along a path with length within a factor of 3 of the shortest path and \( d(e) \propto 2^{c(e)} \).
- Lose a factor of three at the beginning.
- \[ c_{opt} \geq \sum_i d(s_i, t_i) \geq \frac{1}{3} \sum_i d(p_i) \]
- \[ c_{opt} \geq \sum_i d(s_i, t_i) \geq \frac{1}{3} \sum_i d(p_i) \]
- We obtain \( c_{max} = 3(1 + \frac{1}{m})c_{opt} + 2 \log m \).
- This is worse!
- What do we gain?

How good is equilibrium?

Path is routed along shortest path and \( d(e) \propto 2^{c(e)} \).
- For \( e \) with \( c(e) \leq c_{max} - 2 \log m \):
- \[ 2^{c(e)} \leq 2^{c_{max} - 2 \log m} = \frac{c_{max}}{m} \]

\[
c_{opt} \geq \sum_i d(s_i, t_i) = \sum_e d(e) c(e) \]
\[
= \sum_e \frac{2^{c(e)} c(e)}{\sum_e 2^{c(e)}} \]
\[
= \frac{\sum_e c(e) \cdot 2^{c(e)} + \sum_e c(e) \cdot 2^{c(e)}}{(c_{max} - 2 \log m)^2} \]
\[
\geq \frac{(c_{opt})}{(1 + \frac{1}{m}) c_{opt} + 2 \log m \cdot (1 + \frac{1}{m})} \]
\[
\geq \frac{c_{max} - 2 \log m}{1 + \frac{1}{m}} \]
\[
= (1 + \frac{1}{m}) \cdot c_{opt} + 2 \log m \]
\[
= (1 + \frac{1}{m}) \cdot c_{opt} + 2 \log m \]
\[
(\text{Almost) within } 2 \log m \text{ of optimal!} \]

The end: sort of.

Got to here in class. Feel free to continue reading.

How do we gain?

- Shortest paths!
- Higher tolls on congested edges.
- Has
An algorithm!

Algorithm: reroute paths that are off by a factor of three.
(Note: $d(e)$ recomputed every rerouting.)

$-1$ for $c(e)$

$p: w(p) = X \implies w'(p) = X/2$

$+1$ for $c(e)$

$p': w'(p') \leq X/3 \implies w'(p') \leq 2X/3$

Potential function: $\sum_e w(e), w(e) = 2^{c(e)}$

Moving path:
Divides $w(e)$ along long path (with $w(p)$ of $X$) by two.
Multiplies $w(e)$ along shorter ($w(p) \leq X/3$) path by two.

$-X/2 + X/3 = -X/6$.

Potential function decreases. $\implies$ termination and existence.

Tuning...

Replace $d(e) = (1 + \epsilon)c(e)$.
Replace factor of 3 by $(1 + 2\epsilon)$
$c_{\text{max}} \leq (1 + 2\epsilon)c_{\text{opt}} + 2\log m/\epsilon$. (Roughly)
Fractional paths?

Wrap up.

Dueling players:
Toll player raises tolls on congested edges.
Congestion player avoids tolls.
Converges to near optimal solution!
A lower bound is “necessary” (natural), and helpful (mysterious?)!

Done for the day.....

...see you on Thursday.