COURSE REVIEW

April 27, 2016

Highlights

- Linear programming formulations for approx algo
- Basic probability, randomized rounding
- Intro to eigenvalues, how to compute them (power)
- Eigenvalues of graphs, sparse cuts, random walks
- Directed graphs, pagerank, …
- Optimization, ellipsoid algorithm
- SDP, "vector relaxations", …
- Duality – "linear proofs of infeasibility"
- MWU framework, Gap reductions, Beyond worst case

More advanced, more handwaving

Problems encountered

- Max Cut:
  - $1/2$ approx by local search
  - $1/2$ approx by ... ?
  - 0.878 approx using ...?

- Sparsest cut.

$$\Phi(S) = \frac{E(S,\bar{S})}{|S||\bar{S}|}$$

  - Eigenvalue relaxation
  - Cheeger's ineq.

Connection between eigenvalues & sparsest cut.

Problems encountered (contd.)

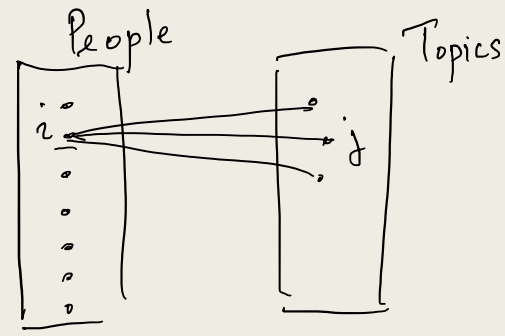
- Set Cover:
  (Vertex cover).
  - Greedy algorithm.
  - Randomized rounding

- Independent Set
  (Clique)
  - SDP relaxation "much better" than LP
  - Hardness amplification.

- 3-SAT, max 3 SAT
  - 7/8 - approx...
  - Reductions (Gap preserving ...)
  - PCP

LP based approximations

- Have a combinatorial problem (e.g. Set Cover, Indep. Set, ...)
- Figure out variables, write "integer program".
- Feasible solution to ILP $\Leftrightarrow$ solution to problem.



<u>Variables</u>: y_i for each person

$y_i \in \{0,1\}$

$$\forall j: \quad \sum_{i:\, i \text{ expert on } T_j} y_j \geq 1$$

RELAX TO $0 \leq y_i \leq 1$

LP based approximations

- Move from fractional solution to LP

  ROUNDING ALGORITHM

  integer solution

- Need to show that solution cost is roughly preserved.

- Writing the right relaxation is key to ensure this.

Basic probablility

- Linearity of Expectation: for any two r.v.s X & Y, $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.
- Markov's inequality: For non-negative R.V. X:

$$\Pr\left[X \geq t \cdot \mathbb{E}[X]\right] \leq 1/t.$$

[Not much larger than expectation].

- Chernoff bounds: Sum of independent r.v.s is tightly concentrated.. [deviation not much more than variance].

Eigenvalues, power method

- Symmetric matrix A $(n \times n)$ has n real e. values and n orthogonal eigenvectors.

$$A = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T$$

- Simple way to compute: power method:
  - start with any x (unit vector)
  - repeatedly do $x \to \dfrac{Ax}{\|Ax\|}$

(Converges to "top" e. vector; Rate depends on "GAP").

Graphs and their spectra

- <u>ADJACENCY MATRIX</u>:

$$A_G = \begin{array}{c} \\ n \end{array}\overset{n}{\begin{bmatrix} 0 & & 1 & \\ & \ddots & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}} \quad \left[1 \text{ if edge } i \sim j\right]$$

- $tr(A_G) = 0 \Rightarrow$ sum of eigenvalues $= 0$.
- max-eigenvalue $= d$ (for d-regular graphs)
- min-eigenvalue $\geq -d$ (equality iff G is ...?)

Cheeger's inequality

- Number of eigenvalues $= d$

  $\equiv$

  Number of connected components.

- If λ_2 is "almost" d, then there is a "sparse" cut. (essentially two connected components)

$$\frac{\lambda_1 - \lambda_2}{2d} \leq \phi(G) \leq \sqrt{\frac{(\lambda_1 - \lambda_2)}{d}} \quad \left[\text{upto consts.}\right]$$

Random walks

- Arbitrary, undirected graphs;

  Walk matrix : $AD^{-1} \equiv D^{1/2}\underbrace{\left(D^{-1/2}AD^{-1/2}\right)}_{\text{Normalized adjacency matrix.}}D^{-1/2}$

- Stationary distribution

- Convergence time $\longleftrightarrow$ eigenvalue gap

  $\updownarrow$

  sparse cuts in G.

NO SPARSE CUTS $\Rightarrow$ FAST CONV. of WALKS

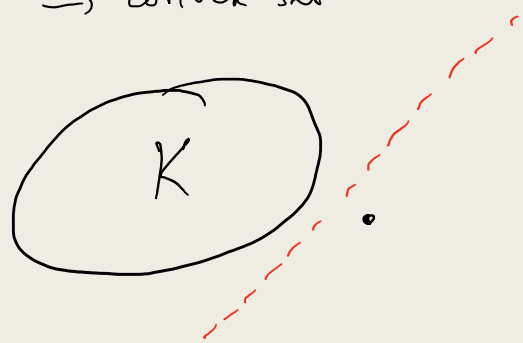
Convex Optimization

General algorithm: GRAD DESCENT

minimize $f(x)$ → convex function

s.t. $x \in K$ → convex set.

- Separation oracles
- Ellipsoid algorithm



"good" convex set with efficient separation oracle ⇒ efficient convex optimization

Semidefinite programming

- Linear optimization over "PSD cone"
  (cone of $n \times n$ matrices with all non-neg. eigenvalues.)

$$\max \quad C \circ X$$

$$\text{s.t.} \quad A_1 \circ X \le b_1 \iff$$

$$\vdots$$

$$A_m \circ X \le b_m$$

$$\max \sum_{i,j} C_{ij} \langle u_i, u_j \rangle$$

$$\vdots$$

"Vector programming" formulation

Semidefinite programming

[Goemans - Williamson]

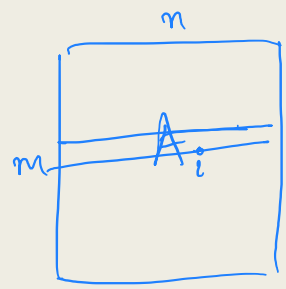
- map vertices to unit vectors in $\mathbb{R}^n$
- maximize $\sum_{u \sim v} \frac{1 - \langle x_u, x_v \rangle}{2}$, s.t. $\|x_u\|^2 = 1$ $\forall u$
- Random hyperplane rounding — gives a 0.878... approximation!

Farkás lemma and duality

- How to show a set of linear inequalities does not have a common feasible solution?

$x \in \mathbb{R}^n$

$$A_1 x \le b_1$$
$$A_2 x \le b_2$$
$$\vdots$$
$$A_m x \le b_m$$

; $A =$ 

<u>Lemma</u>: Infeasible iff $\exists\ y_1, y_2, \ldots, y_m$ s.t.

(a) $\sum_i b_i y_i < 0$ and (b) $y^T A = 0$.

"Linear proof"

Linear programming duality

PRIMAL	DUAL
$\max \; c^T x$	$\min \; b^T y$
$\text{s.t.} \; Ax \le b$	$\text{s.t.} \; y^T A \geq c$
$x \geq 0$	$y \geq 0$

- Dual tries to "certify" that primal OPT $\le$ ...
- For any feasible (resp.) x, y: $c^T x \le b^T y$

DUALITY THEOREM: If both are feasible, optima are equal..

Intro to game theory

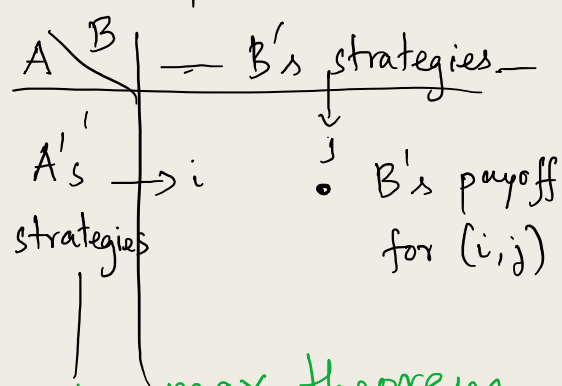
Two player games:

- Actions, payoffs for each player.

Zero sum game: payoff(A) = − payoff(B)

Row player's best move:

$$\arg\min_{a \in A\text{'s strats}} \max_{b \in B} M(a,b)$$



"Mixed strategies", min-max theorem

Multiplicative weight update

GOAL: Decision-making to avoid regret.

Day 1:	w_{11}	w_{12}	...
Day 2:	w_{21}	w_{22}	...
⋮		⋮	

see how well the experts are doing, update weights appropriately..

"Play" a distribution over experts : Can make sure we compete with $\boxed{\text{best expert in hindsight}}$

Hardness of approximation

High level moral: for many approximation problems, solving to an approx factor 'α' is as hard as solving exactly.

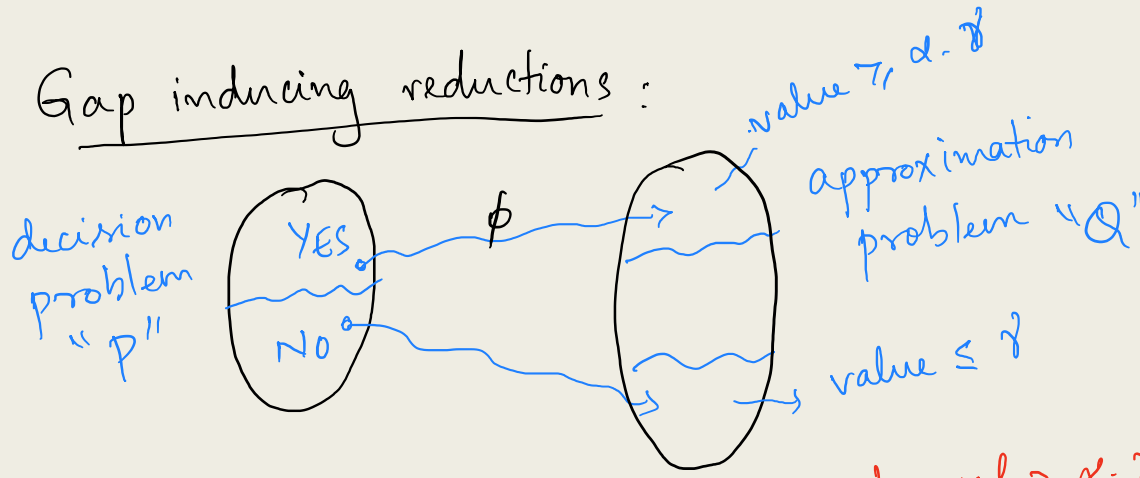
(polynomial time) reduction

Similar in spirit to error correcting codes

Proof goes via a "prover-verifier" view of NP (PCP theorem)

Hardness of approximation

Gap inducing reductions :



Reduction takes YES instances to val $\geq \alpha \cdot \gamma$ and NO instances to val $\leq \gamma$.

Thus: If we can solve problem Q to approx factor 'α', then can solve P exactly!

Beyond worst case

[How to remain hopeful while facing hardness]

- Average case analysis
  - problem is "usually" easy (besides bad regions)
- Smoothed analysis
  - no bad regions; problem easy unless "very unlucky"
- Stability assumptions
- Bi-criteria approximations:
  - relaxing constraints may lead to good approximations..

Thanks,

and Good

Luck!