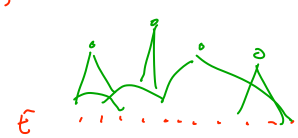
From relaxations to integral solutions (cont.)

Optimization - 10725
Carlos Guestrin
Carnegie Mellon University

April 23rd, 2008

Relaxations and rounding

- What do we do if we don't get integral solutions?
  - because $P \neq NP$ (probably true)
- E.g., set cover problem
  - Ground elements $v \in V$
  - Set of Sets $S \in \mathcal{S}$ $\quad S \subseteq V$
  - Cost for sets c_S
  - Find cheapest collection of subsets that covers all elements



- Integer program and relaxation:

$$\min_x \sum_S c_S x_S \qquad x_S \in \{0,1\} \qquad \forall v \quad \sum_{S: v \in S} x_S \geq 1$$

relax

$$\min_x \sum_S c_S x_S \qquad \sum_{S: v \in S} x_S \geq 1 \quad \forall v \qquad 0 \leq x_S \leq 1$$

- How can we obtain a good integer (rounded) solution?
  - If we set all nonzero x_S to one, then very bad idea... arbitrarily $O(n)$ more expensive solution than OPTIP
  - smart rounding?

Consider a special case…

- Suppose each element in at most k sets
- From inequality constraint:
- Rounding strategy:
- Feasibility:
- Cost of rounded solution:

©2008 Carlos Guestrin

Very simple example of randomized rounding

- Solve set cover relaxation:
- Randomly pick a collection of subsets G
  - For each S, add it to G with (independent) probability x_S
- What's the expected cost of G?
  - I_S indicator of whether set S is in G

©2008 Carlos Guestrin

How many elements do we cover?

- Expected cost of G can be lower than OPT_{IP}
  - Must cover fewer elements
- I_v is indicator of whether element v covered by G
- Expected number of elements covered:

©2008 Carlos Guestrin

How big can cost get?

- Expected cost is lower than OPT_{IP}
  - But how big can actual cost get?
  - (a simple bound here, more interesting bounds using more elaborate techniques)
- Markov Inequality: Let Y be a non-negative random variable
  - Then
- In our example:

©2008 Carlos Guestrin

Randomization & Derandomization

- MAX-3SAT:
  - 3SAT formula:
    - Binary variables $X_1,\ldots,X_n$
    - Conjunction of clauses $C_1,\ldots,C_M$
    - Each clause is a disjunction of three literals on three different variables
  - Want assignment that maximizes number of satisfied formulas

©2008 Carlos Guestrin

Randomized algorithm for MAX-3SAT

- Pick assignment for each X_i independently, at random with prob. 0.5

- Expected number of satisfied clauses:

©2008 Carlos Guestrin

Aside: Probabilistic Method

- Expected number of satisfied clauses:

- Probabilistic method: for any random var. Y, there exists assignment y such that P(y)>0, y≥E[Y]
  - Almost obvious fact
  - Amazing consequences

- For example, in the context of MAX-3SAT:

©2008 Carlos Guestrin

Derandomization

- There exists assignment X that achieves

- In expectation, we get 7/8.M, but can we get it with prob. 1? Without randomization?

- Derandomization: From a randomized algorithm, obtain a deterministic algorithm with same guarantees
  - Today: method of conditional expectations

©2008 Carlos Guestrin

Method of conditional expectations

- Conditional expectation:
- Expectation of the conditional expectation:
  -
- Consider MAX-3SAT:
  - Expectation:
  - Expectation of conditional expectation:

©2008 Carlos Guestrin

Computing conditional expectation

- Conditioning on $X_1=1$:
- General case: $X_1=v_1,\ldots, X_i=v_i$
  - Sum over clauses, I_j is indicator clause j is satisfied

©2008 Carlos Guestrin

Derandomized algorithm for MAX-3SAT

- For i=1,…,n
  - Try X_i=1
    - Compute
  - Try X_i=0
    - Compute
  - Set v_i to best assignment to X_i
- Deterministic algorithm guaranteed to achieve at least 7/8.M

©2008 Carlos Guestrin

Most probable explanation (MPE) in a Markov network

- Markov net:
- Most probable explanation:
- In general, NP-complete problem, and hard to approximate

©2008 Carlos Guestrin

MPE for attractive MNs – 2 classes

- Attractive MN:
  - E.g., image classification

- Finding most probable explanation

- Can be solved by

©2008 Carlos Guestrin

MPE, Attractive MNs, k classes

- MPE for k classes:

- Multiway cut:
  - Graph G, edge weights w_{ij}
  - Finding minimum cut, separate $s_1,\ldots,s_k$

- Multiway cut problem is

©2008 Carlos Guestrin

Multiway cut – combinatorial algorithm

- Very simple alg:
  - For each i=1…k
    - Find cut C_i that separates s_i from rest
  - Discard $\text{argmax}_i\ w(C_i)$, return union of rest
- Algorithm achieves 2-2/k approximation
  - OPT cut A* separates graph into k components
    - No advantage in more than k
  - From A* form $A^*_1,\ldots,A^*_k$, where A^*_i separates s_i from rest
  - Each edge in A* appears in
    - Thus

©2008 Carlos Guestrin

Multiway cut proof

- Thus, for OPT cut A* we have that:
- Each A^*_i separates s_i from rest, thus
- But, can do better, because

©2008 Carlos Guestrin