From relaxations to integral solutions (cont.)

Optimization - 10725
Carlos Guestrin
Carnegie Mellon University
April 21st, 2010

Relaxations and rounding

What do we do if we don’t get integral solutions?
- Because \( \mathcal{P} \notin \mathcal{NP} \text{ (possibly)} \)
- E.g., set cover problem
  - Ground elements
  - Set of sets
  - Cost for sets \( c_S \)
  - Find cheapest collection of subsets that covers all elements

Integer program and relaxation:

\[
\begin{align*}
\min & \quad \sum_S c_S x_S \\
\text{s.t.} & \quad \forall e \in V, \exists S \in \mathcal{S} \text{ such that } e \in S, \sum_S x_S = 1 \\
\end{align*}
\]

LP relaxation: \( x_S \in [0,1] \)

How can we obtain a good integer (rounded) solution?
- If we set all nonzero \( x_S \) to one, then
- Smarter way to round?
Consider a special case…

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

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
How big can cost get?

- Expected cost is lower than $\text{OPT}_{\text{LP}}$
  - 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:

How many elements do we cover?

- Expected cost of $G$ can be lower than $\text{OPT}_{\text{LP}}$
  - Must cover fewer elements

- $I_v$ is indicator of whether element $v$ covered by $G$
- Expected number of elements covered:
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

Randomized algorithm for MAX-3SAT
- Pick assignment for each $X_i$ independently, at random with prob. 0.5
- Expected number of satisfied clauses:
Aside: Probabilistic Method

- Expected number of satisfied clauses:
  - Probabilistic method: for any random var. Y, there exists assignment y such that \( P(y) > 0, y \geq E[Y] \)
    - Almost obvious fact
    - Amazing consequences
  - For example, in the context of MAX-3SAT:

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
Method of conditional expectations

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

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
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

Most probable explanation (MPE) in a Markov network

- Markov net:

- Most probable explanation:
  - In general, NP-complete problem, and hard to approximate
MPE for attractive MNs – 2 classes

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

- Finding most probable explanation

- Can be solved by

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 – combinatorial algorithm

- Very simple alg:
  - For each \(i=1\ldots k\)
    - 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

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