Introduction to Randomized Algorithms Part II

Kleinberg and Tardos, Chapter 13; See also lecture notes by Nick Harvey (www.cs.ubc.ca/~nickhar/W15) and Anna Karlin (courses.cs.washington.edu/courses/cse525/13sp)
Outline

• Review: Randomization, Markov's Inequality and probability amplification applied to Max Sat
• Making randomized algorithm deterministic: the method of conditional expectations
• Better algorithms using linear programming and randomized rounding
• How these ideas lead to better deterministic algorithms for Max Cut an Mat Sat
Max Cut: Review

Random-Cut: A randomized approximation algorithm

\[ E[|\text{Random-Cut}(G)|] = \frac{|E|}{2} \]

- Intuition: Each edge has probability 1/2 to cross the cut
- Formal proof uses indicator variables

\[ \Pr[|\text{Random-Cut}(G)| > (1/2-\varepsilon)|E| ] \geq \varepsilon \]

- Reverse Markov's inequality

Probability amplification by repeated trials
**Concentration Inequalities**

*Markov’s Inequality*: Let Y be a real-valued random variable that assumes only nonnegative values. Then for all $a > 0$,

$$Pr[Y \geq a] \leq \frac{E[Y]}{a}$$

*Reverse Markov Inequality*: Let Y be a real-valued random variable that is never larger than B. Then for all $a < B$,

$$Pr[Y \leq a] \leq \frac{E[B-Y]}{(B-a)} = \frac{B-E[Y]}{(B-a)}$$
Given a Boolean formula $\phi$ in conjunctive normal form with $m$ clauses $C_1, \ldots, C_m$ over $n$ variables $x_1, \ldots, x_n$, and a positive weight $w_j$ associated with the clause $C_j$

Find a truth assignment for $x = (x_1, \ldots, x_n)$ that maximizes the sum of the weights of the satisfied clauses.
Given a Boolean formula $\Phi$ in conjunctive normal form with $m$ clauses $C_1, \ldots, C_m$ over $n$ variables $x_1, \ldots, x_n$, and a positive weight $w_j$ associated with the clause $C_j$

Find a truth assignment for $x = (x_1, \ldots, x_n)$ that maximizes the sum of the weights of the satisfied clauses

- Simple randomized algorithm?
- Expected quality guarantee?
- Probability that truth assignment is close to the expected guarantee?
Derandomizing Max Cut and Max Sat

The method of conditional expectations can be used to "derandomize" algorithms such as Random-Cut and Random-Sat.

The solution found by the resulting deterministic algorithm has value at least the expected value of a solution found by the original randomized algorithm.
Suppose $X_1, X_2, \ldots, X_n$ are independent random indicator variables

Let $f : \{0,1\}^n \to \mathbb{R}$ be a function such that

\[
E[ f(X_1,\ldots,X_n) ] \geq \mu
\]

We'd like to deterministically find bits $x_1,\ldots,x_n$ such that

\[
f(x_1,\ldots,x_n) \geq \mu
\]
Method of Conditional Expectations

Notation: Let $E[ f(x_1..x_i,X_{i+1}..X_n) ]$ be the expected value of $f(X_1,...,X_n)$, given that $X_1 = x_1$, ..., $X_i = x_i$

(This notation helps us reason about the expected value of the result produced by an algorithm after $i$ iterations have already taken place)
Method of Conditional Expectations

MCE(f)  // Method of Conditional Expectations

For i = 1 to n
    If ( E[ f(x_1..x_{i-1},0,X_{i+1}..X_n) ] ≥ E[ f(x_1..x_{i-1},1,X_{i+1}..X_n) ] )
        x_i ← 0
    Else
        x_i ← 1

Output \( x_1,\ldots,x_n \)  // \( f(x_1,\ldots,x_n) \geq \mu \)
Method of Conditional Expectations

MCE(f)  // Method of Conditional Expectations

For i = 1 to n

If ( E[ f(x_1..x_{i-1},0,X_{i+1}..X_n) ] \geq E[ f(x_1..x_{i-1},1,X_{i+1}..X_n) ] )
    x_i \leftarrow 0
Else
    x_i \leftarrow 1

Output x_1,...,x_n  // f(x_1,...,x_n) \geq \mu

We can implement MCE(f) efficiently if we can efficiently evaluate the quantities E[ f(x_1..x_i,X_{i+1}..X_n) ]
Claim: For any $i$ between 1 and $n$, if 
$$E[ f(x_1..x_{i-1},X_i..X_n) ] \geq \mu$$

at the start of the $i$th iteration, then 
$$E[ f(x_1..x_i,X_{i+1}..X_n) ] \geq \mu$$

at the end of the $i$th iteration.

Therefore if initially we have that $E[ f(X_1,..,X_n) ] \geq \mu$, we must have that $f(x_1,..,x_n) \geq \mu$ at the end of $n$ iterations.
Max Cut and the Method of Conditional Expectations
Let’s use the Method of Conditional Expectations (MCE) to derandomize Random-Cut

• Let vertex i be considered on the ith iteration of the For loop of Random-Cut
• For fixed bits $x_1, \ldots, x_n$, let $f(x_1, \ldots, x_n) = |\text{Cut}(U)|$, where $U = \{i \mid x_i = 1\}$
• We wish to find $x_1, \ldots, x_n$ for which $f(x_1, \ldots, x_n) \geq |E|/2$
Max Cut and the Method of Conditional Expectations

Let $X_i$ be independent indicator variables, with $X_i = 1$ iff Random-Cut assigns node $i$ to $U$

Then $E[ f(x_1..x_i,X_{i+1}..X_n) ]$ is the expected size of Random-Cut($G$), given that $X_1 = x_1$, ... $X_i = x_i$
Let $X_i$ be independent indicator variables, with $X_i = 1$ iff Random-Cut assigns node $i$ to $U$

Then $E[ f(x_1..x_{i+1}X_{i+1}..X_n) ]$ is the expected size of Random-Cut($G$), given that $X_1 = x_1$, ... $X_i = x_i$

*Can we compute $E[ f(x_1..x_{i+1}X_{i+1}..X_n) ]$ efficiently?*
Max Cut and the Method of Conditional Expectations

Let $X_i$ be independent indicator variables, with $X_i = 1$ iff Random-Cut assigns node $i$ to $U$

Then $E[ f(x_1..x_i,X_{i+1}..X_n) ]$ is the expected size of Random-Cut($G$), given that $X_1 = x_1, ... X_i = x_i$

*Can we compute $E[ f(x_1..x_i,X_{i+1}..X_n) ]$ efficiently?*

Yes: it is the number of edges with both endpoints in $\{1,...,i\}$ that cross the cut, plus half of the number of edges that have at least one endpoint in $\{i+1,...,n\}$
Max Cut and the Method of Conditional Expectations

• The derandomized version of Random-Cut is the simple greedy algorithm that considers nodes in turn places the ith node on whichever side of the cut maximizes the number of edges with an endpoint at i that cross the cut, breaking ties arbitrarily

• Deriving the greedy algorithm from Random-Cut via MCE provides a way of analyzing the greedy algorithm
Max Sat and Max Cut

Approximation ratios of simple randomized algorithms:

Weighted Max Sat: approximation ratio 2
Max Cut: approximation ratio 2
Max Sat and Max Cut

Best known approximation ratios:
Max Sat and Max Cut

Best known approximation ratios:

Weighted Max Sat: approximation ratio 1.299
• Techniques: Linear programming, randomized rounding
• Yannakakis (1992), Goemans-Williamson (1994), Hori et al. (2005)
Max Sat and Max Cut

Best known approximation ratios:

Weighted Max Sat: approximation ratio 1.299
• Techniques: Linear programming, randomized rounding
• Yannakakis (1992), Goemans-Williamson (1994), Hori et al. (2005)

Max Cut: approximation ratio 1.383
• Semidefinite programming, randomized rounding
• Goemans and Williamson (1995)
Max Sat: Integer Program

maximize \[ \sum_{j=1}^{m} w_j z_j \]

subject to \[ \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \quad \forall C_j = \bigvee_{i \in P_j} x_i \lor \bigvee_{i \in N_j} \overline{x_i} \]

\( y_i \) is in \( \{0,1\} \), \( i = 1, \ldots, n \),
\( z_j \) is in \( \{0,1\} \), \( j = 1, \ldots, m \).

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Max Sat: LP Relaxation

\[
\begin{align*}
\text{maximize} & \quad \sum_{j=1}^{m} w_j z_j \\
\text{subject to} & \quad \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \\
& \quad 0 \leq y_i \leq 1, \quad i = 1, \ldots, n, \\
& \quad 0 \leq z_j \leq 1, \quad j = 1, \ldots, m.
\end{align*}
\]

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Max Sat: LP Relaxation

\[ \text{maximize} \quad \sum_{j=1}^{m} w_j z_j \]

subject to

\[ \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \quad \forall C_j = \bigvee_{i \in P_j} x_i \bigvee_{i \in N_j} \bar{x}_i, \]

\[ 0 \leq y_i \leq 1, \quad i = 1, \ldots, n, \]

\[ 0 \leq z_j \leq 1, \quad j = 1, \ldots, m. \]

- Let \((z^*, y^*)\) be an optimal solution to this LP

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Max Sat: LP Relaxation + Randomized Rounding

\[
\text{maximize} \quad \sum_{j=1}^{m} w_j z_j \\
\text{subject to} \quad \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, \quad \forall C_j = \bigvee_{i \in P_j} x_i \lor \bigvee_{i \in N_j} \bar{x}_i,
\]

\[
0 \leq y_i \leq 1, \quad i = 1, \ldots, n, \\
0 \leq z_j \leq 1, \quad j = 1, \ldots, m.
\]

• Let \((z^*, y^*)\) be an optimal solution to this LP

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Max Sat: LP Relaxation + Randomized Rounding

Let \((z^*, y^*)\) be an optimal solution to this LP

Set variable \(x_i\) to true with probability \(f(y_i^*)\), where \(f\) is such that \(1 - 4^{-x} \leq f(x) \leq 4^{x-1}\) in the range \([0,1]\)

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Max Sat: LP Relaxation + Randomized Rounding

The functions $1 - 4^{-x}$ and $4^{x-1}$ in the range $[0,1]$

Williamson: https://people.orie.cornell.edu/dpw/talks/hoffmanfest.pdf
Summary

Randomized algorithms and the method of conditional expectations have been valuable stepping stones to obtaining the best deterministic approximation algorithms for Max Sat, Max Cut and other central NP-hard problems

p.s. please fill out course evaluation forms