

FPTAS for Counting Monotone CNF

Jingcheng Liu¹ Pinyan Lu²

¹University of California, Berkeley

²Microsoft Research

Overview

- 1 Monotone Formulas
- 2 Approximate Counting
- 3 Correlation Decay Analysis
 - Recurrence and Weitz's approach
 - Adaptively truncating SAW tree
 - Exponential decay of correlations

Monotone CNF

Definition (Monotone CNF)

A CNF formula is *monotone* if every literal appears positively.

Remark (Equivalent definition)

A CNF formula φ (as a Boolean function over $\{0, 1\}^n$) is monotone if $\forall \mathbf{x} \leq \mathbf{y}, \varphi(\mathbf{x}) \leq \varphi(\mathbf{y})$.

Monotone CNF

Definition (Monotone CNF)

A CNF formula is *monotone* if every literal appears positively.

Remark (Equivalent definition)

A CNF formula φ (as a Boolean function over $\{0, 1\}^n$) is monotone if $\forall \mathbf{x} \leq \mathbf{y}, \varphi(\mathbf{x}) \leq \varphi(\mathbf{y})$.

Monotonicity

Example (Monotone increasing of covering and domination)

Let X be a cover; then any $Y \supseteq X$ is also a cover. E.g.:

- Vertex covers
- Edge covers
- Dominating sets
- Set covers

Example (Monotone decreasing of packing)

Let X be a packing; then any $Y \subseteq X$ is also a packing. E.g.:

- Independent sets
- Matching
- Set packings

Monotonicity

Example (Monotone increasing of covering and domination)

Let X be a cover; then any $Y \supseteq X$ is also a cover. E.g.:

- Vertex covers
- Edge covers
- Dominating sets
- Set covers

Example (Monotone decreasing of packing)

Let X be a packing; then any $Y \subseteq X$ is also a packing. E.g.:

- Independent sets
- Matching
- Set packings

Examples

Example (Set Cover)

Universe $U = \{x, y, z\}$, and sets $A = \{x, y\}$, $B = \{x\}$, $C = \{x, z\}$, $D = \{y, z\}$.
 $F \subseteq \{A, B, C, D\}$ is a *set cover* if $\bigcup F = U$.

Sets are variables, and the elements are constraints. In monotone CNF:

$$\underbrace{(A \vee B \vee C)}_{\text{element } x} \wedge \underbrace{(A \vee D)}_{\text{element } y} \wedge \underbrace{(C \vee D)}_{\text{element } z}.$$

Definition (Degree of a variable)

The *degree* of a variable is the number of times it occurs in the formula.

Examples

Example (Set Cover)

Universe $U = \{x, y, z\}$, and sets $A = \{x, y\}$, $B = \{x\}$, $C = \{x, z\}$, $D = \{y, z\}$.
 $F \subseteq \{A, B, C, D\}$ is a *set cover* if $\bigcup F = U$.

Sets are variables, and the elements are constraints. In monotone CNF:

$$\underbrace{(A \vee B \vee C)}_{\text{element } x} \wedge \underbrace{(A \vee D)}_{\text{element } y} \wedge \underbrace{(C \vee D)}_{\text{element } z}.$$

Definition (Degree of a variable)

The *degree* of a variable is the number of times it occurs in the formula.

Overview

1 Monotone Formulas

2 Approximate Counting

3 Correlation Decay Analysis

- Recurrence and Weitz's approach
- Adaptively truncating SAW tree
- Exponential decay of correlations

What is Counting?

Interested in: the number of satisfying assignments to a monotone CNF formula.

Definition (Complexity class #P)

#P: *counting* problems whose *decision* problem is in NP.

Exact counting is generally hard (#P-complete).

Definition (Fully polynomial-time approximation scheme(FPTAS))

An algorithm is an *FPTAS* if, given as input a problem instance of size n with true answer N and a parameter $\epsilon \in (0, 1]$, it outputs a *multiplicative approximation* \hat{N} in time $\text{poly}(n, \frac{1}{\epsilon})$ such that

$$(1 - \epsilon)N \leq \hat{N} \leq (1 + \epsilon)N.$$

What is Counting?

Interested in: the number of satisfying assignments to a monotone CNF formula.

Definition (Complexity class #P)

#P: *counting* problems whose *decision* problem is in NP.

Exact counting is generally hard (#P-complete).

Definition (Fully polynomial-time approximation scheme(FPTAS))

An algorithm is an *FPTAS* if, given as input a problem instance of size n with true answer N and a parameter $\epsilon \in (0, 1]$, it outputs a *multiplicative approximation* \hat{N} in time $\text{poly}(n, \frac{1}{\epsilon})$ such that

$$(1 - \epsilon)N \leq \hat{N} \leq (1 + \epsilon)N.$$

Approximate Counting

Exact and approximate counting problems arise in many applications, e.g.:

- **Combinatorics:** Graph polynomials
- **Statistical physics:** Partition functions of spin systems
- **Game theory:** Dynamics on social networks; Pricing in combinatorial prediction market design
- **Machine learning:** Inference in graphical models

Some Previous Results on Monotone #CSPs

Algorithmic side:

- **Independent set (or vertex cover):** FPTAS if the maximum degree (or connective constant) $\Delta \leq 5$ [Wei'06; SSY'13], and no efficient approximation when $\Delta \geq 6$ unless $\text{RP}=\text{NP}$ [Sly'10].
- **Matching:** FPTAS if the maximum degree (or connective constant) is bounded by a constant [BGK+'07;SSSY'15], otherwise only an *FPRAS* is known [JS'89].
- **Edge cover:** FPTAS [LLL'14].
- **Dominating set:** FP for interval/trapezoid graphs [KOU'11], but #P-hard under AP-reductions even on bipartite graphs [GGL'14].

Hardness side:

Approximating Boolean #CSP is generally hard for non-monotone CSPs, and the classification is complete [DGJR'10].

But the approximability of *monotone* CSP is still open.

Some Previous Results on Monotone #CSPs

Algorithmic side:

- **Independent set (or vertex cover):** FPTAS if the maximum degree (or connective constant) $\Delta \leq 5$ [Wei'06; SSY'13], and no efficient approximation when $\Delta \geq 6$ unless $RP=NP$ [Sly'10].
- **Matching:** FPTAS if the maximum degree (or connective constant) is bounded by a constant [BGK+'07;SSSY'15], otherwise only an *FPRAS* is known [JS'89].
- **Edge cover:** FPTAS [LLL'14].
- **Dominating set:** FP for interval/trapezoid graphs [KOU'11], but #P-hard under AP-reductions even on bipartite graphs [GGL'14].

Hardness side:

Approximating Boolean #CSP is generally hard for non-monotone CSPs, and the classification is complete [DGJR'10].

But the approximability of *monotone* CSP is still open.

Some Previous Results on Monotone #CSPs

Algorithmic side:

- **Independent set (or vertex cover):** FPTAS if the maximum degree (or connective constant) $\Delta \leq 5$ [Wei'06; SSY'13], and no efficient approximation when $\Delta \geq 6$ unless $RP=NP$ [Sly'10].
- **Matching:** FPTAS if the maximum degree (or connective constant) is bounded by a constant [BGK+'07;SSSY'15], otherwise only an *FPRAS* is known [JS'89].
- **Edge cover:** FPTAS [LLL'14].
- **Dominating set:** FP for interval/trapezoid graphs [KOU'11], but #P-hard under AP-reductions even on bipartite graphs [GGL'14].

Hardness side:

Approximating Boolean #CSP is generally hard for non-monotone CSPs, and the classification is complete [DGJR'10].

But the approximability of *monotone* CSP is still open.

Some Previous Results on Monotone #CSPs

Algorithmic side:

- **Independent set (or vertex cover):** FPTAS if the maximum degree (or connective constant) $\Delta \leq 5$ [Wei'06; SSY'13], and no efficient approximation when $\Delta \geq 6$ unless $RP=NP$ [Sly'10].
- **Matching:** FPTAS if the maximum degree (or connective constant) is bounded by a constant [BGK+'07;SSSY'15], otherwise only an *FPRAS* is known [JS'89].
- **Edge cover:** FPTAS [LLL'14].
- **Dominating set:** FP for interval/trapezoid graphs [KOU'11], but #P-hard under AP-reductions even on bipartite graphs [GGL'14].

Hardness side:

Approximating Boolean #CSP is generally hard for non-monotone CSPs, and the classification is complete [DGJR'10].

But the approximability of *monotone* CSP is still open.

Some Previous Results on Monotone #CSPs

Algorithmic side:

- **Independent set (or vertex cover):** FPTAS if the maximum degree (or connective constant) $\Delta \leq 5$ [Wei'06; SSY'13], and no efficient approximation when $\Delta \geq 6$ unless $\text{RP}=\text{NP}$ [Sly'10].
- **Matching:** FPTAS if the maximum degree (or connective constant) is bounded by a constant [BGK+'07;SSSY'15], otherwise only an *FPRAS* is known [JS'89].
- **Edge cover:** FPTAS [LLL'14].
- **Dominating set:** FP for interval/trapezoid graphs [KOU'11], but #P-hard under AP-reductions even on bipartite graphs [GGL'14].

Hardness side:

Approximating Boolean #CSP is generally hard for non-monotone CSPs, and the classification is complete [DGJR'10].

But the approximability of *monotone* CSP is still open.

Our Results

Theorem (Main Theorem)

Counting monotone CNF admits an FPTAS if the maximum degree $\Delta \leq 5$.

Corollaries

FPTAS for counting:

- *set covers (or hitting sets) with maximum set size $d \leq 5$.*
- *hypergraph independent sets with maximum degree $\Delta \leq 5$.*
- *hypergraph edge covers with hypergraph edge size at most 5.*
- *dominating sets in graphs with maximum degree $\Delta \leq 4$.*

* All of the above, except for dominating sets, are tight unless $RP = NP$.

Theorem

Counting 3D matchings admits an FPTAS if the maximum degree $\Delta \leq 4$.

Overview

- 1 Monotone Formulas
- 2 Approximate Counting
- 3 Correlation Decay Analysis
 - Recurrence and Weitz's approach
 - Adaptively truncating SAW tree
 - Exponential decay of correlations

Review of Weitz's approach

Let Z_G be the independent set polynomial. (e.g. number of independent sets)

Quick Fact

- Equivalent to computing $R(G, u) \triangleq \frac{Z_G|_{u=1}}{Z_G|_{u=0}}$.
- An additive approx. of R translates to a multiplicative approx. of Z_G .

Theorem (Weitz'06)

$$R(G, u) = R(T_{SAW}(G), u).$$

Sketch of proof: based on a recursive decomposition:

$$R(G, u) = \prod_{i=1}^d \frac{1}{1 + R(G_i, v_i)},$$

where $d \triangleq \deg_G(u)$, and let $N_G(u)$ be enumerated as $\{v_i\}_{i=1}^d$, then $G_i \triangleq (G - u) - \{v_j\}_{j=1}^{i-1}$.

Review of Weitz's approach

Let Z_G be the independent set polynomial. (e.g. number of independent sets)

Quick Fact

- Equivalent to computing $R(G, u) \triangleq \frac{Z_G|_{u=1}}{Z_G|_{u=0}}$.
- An additive approx. of R translates to a multiplicative approx. of Z_G .

Theorem (Weitz'06)

$$R(G, u) = R(T_{SAW}(G), u).$$

Sketch of proof: based on a recursive decomposition:

$$R(G, u) = \prod_{i=1}^d \frac{1}{1 + R(G_i, v_i)},$$

where $d \triangleq \deg_G(u)$, and let $N_G(u)$ be enumerated as $\{v_i\}_{i=1}^d$, then $G_i \triangleq (G - u) - \{v_j\}_{j=1}^{i-1}$.

Review of Weitz's approach

Let Z_G be the independent set polynomial. (e.g. number of independent sets)

Quick Fact

- Equivalent to computing $R(G, u) \triangleq \frac{Z_G|_{u=1}}{Z_G|_{u=0}}$.
- An additive approx. of R translates to a multiplicative approx. of Z_G .

Theorem (Weitz'06)

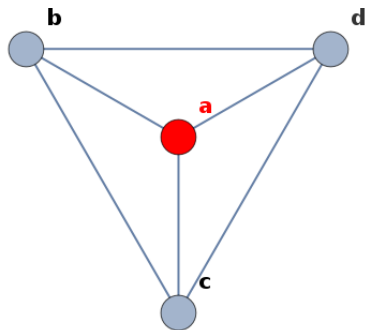
$$R(G, u) = R(T_{SAW}(G), u).$$

Sketch of proof: based on a recursive decomposition:

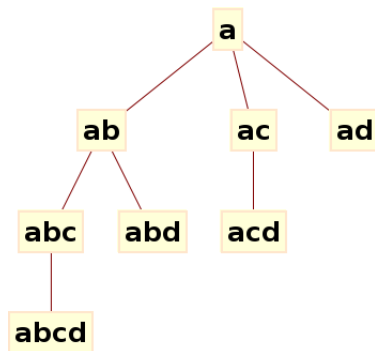
$$R(G, u) = \prod_{i=1}^d \frac{1}{1 + R(G_i, v_i)},$$

where $d \triangleq \deg_G(u)$, and let $N_G(u)$ be enumerated as $\{v_i\}_{i=1}^d$, then $G_i \triangleq (G - u) - \{v_j\}_{j=1}^{i-1}$.

Self-avoiding walk tree



(a) K_4 .



(b) SAW tree of K_4 .

Algorithm (Weitz'06)

Truncate the exponential-sized tree back to polynomial size. When $\Delta \leq 5$, truncating at $\log(n/\epsilon)$ depth gives good approximation.

Recurrence for hypergraphs

Question

For hypergraphs, or monotone CNF formulas C , do we still have

$$R(C, x) = R(T_{SAW}(C), x)?$$

Here recall $R(C, x) \triangleq \frac{Z_C|_{x=0}}{Z_C|_{x=1}}$.

Unfortunately the most natural generalization of T_{SAW} to hypergraphs doesn't work.

Theorem (Sketch)

$$R(C, x) = R'(T_{SAW}(B(C)), x),$$

where

- $B(C)$ is the bipartite graph representation of the hypergraph,
- R' is a two-layer recurrence that alternates between the two partites.

Recurrence for hypergraphs

Question

For hypergraphs, or monotone CNF formulas C , do we still have

$$R(C, x) = R(T_{SAW}(C), x)?$$

Here recall $R(C, x) \triangleq \frac{Z_C|_{x=0}}{Z_C|_{x=1}}$.

Unfortunately the most natural generalization of T_{SAW} to hypergraphs doesn't work.

Theorem (Sketch)

$$R(C, x) = R'(T_{SAW}(B(C)), x),$$

where

- $B(C)$ is the bipartite graph representation of the hypergraph,
- R' is a two-layer recurrence that alternates between the two partites.

Recurrence for hypergraphs

We have the following *two-layer* recursive decomposition:

Theorem (Formal)

$$R(C, x) = \prod_{j=1}^d \left(1 - \prod_{i=1}^{w_j} \frac{R(C_{j,i}, x_{j,i})}{1 + R(C_{j,i}, x_{j,i})} \right),$$

where $d \triangleq \deg_x(C)$, and the d clauses containing x be enumerated as $\{c_j\}_{j=1}^d$. Denote $w_j \triangleq |c_j| - 1$, $C_j \triangleq (C \setminus \{c_k\}_{k \neq j}) \cup \{c_k \setminus x \mid j+1 \leq k \leq d\}$. Let $\{x_{j,i}\}_{i=1}^{w_j}$ be the set of variables in $c_j \setminus x$, and $C_{j,i} \triangleq (C_j \setminus c_j) \setminus \{x_{j,k}\}_{k=1}^{i-1}$.

Truncating the tree of exponential size?

- Truncate at $L \approx \log(n/\varepsilon)$ depth?
 - ☹ The degree of the tree is unbounded! (i.e. w_j is unbounded)
- Truncate more *adaptively*?

Recurrence for hypergraphs

We have the following *two-layer* recursive decomposition:

Theorem (Formal)

$$R(C, x) = \prod_{j=1}^d \left(1 - \prod_{i=1}^{w_j} \frac{R(C_{j,i}, x_{j,i})}{1 + R(C_{j,i}, x_{j,i})} \right),$$

where $d \triangleq \deg_x(C)$, and the d clauses containing x be enumerated as $\{c_j\}_{j=1}^d$. Denote $w_j \triangleq |c_j| - 1$, $C_j \triangleq (C \setminus \{c_k\}_{k \neq j}) \cup \{c_k \setminus x \mid j+1 \leq k \leq d\}$. Let $\{x_{j,i}\}_{i=1}^{w_j}$ be the set of variables in $c_j \setminus x$, and $C_{j,i} \triangleq (C_j \setminus c_j) \setminus \{x_{j,k}\}_{k=1}^{i-1}$.

Truncating the tree of exponential size?

- Truncate at $L \approx \log(n/\varepsilon)$ depth?
 - ☹ The degree of the tree is unbounded! (i.e. w_j is unbounded)
- Truncate more *adaptively*?

Adaptively Truncated Tree

Pretend the tree is a 4-ary branching tree, then do BFS with a budget of visiting at most 4^L nodes. Stop when run out of budget.

Algorithm

Let $L_j = \max(0, L - \lceil \log_4(w_j + 1) \rceil)$.

$$R(C, \mathbf{x}, L) = \prod_{j=1}^d \left(1 - \prod_{i=1}^{w_j} \frac{R(C_{j,i}, \mathbf{x}_{j,i}, L_j)}{1 + R(C_{j,i}, \mathbf{x}_{j,i}, L_j)} \right).$$

Now we take $L \approx \log(n/\varepsilon)$.

But is this a good approximation?..

Adaptively Truncated Tree

Pretend the tree is a 4-ary branching tree, then do BFS with a budget of visiting at most 4^L nodes. Stop when run out of budget.

Algorithm

Let $L_j = \max(0, L - \lceil \log_4(w_j + 1) \rceil)$.

$$R(C, x, L) = \prod_{j=1}^d \left(1 - \prod_{i=1}^{w_j} \frac{R(C_{j,i}, x_{j,i}, L_j)}{1 + R(C_{j,i}, x_{j,i}, L_j)} \right).$$

Now we take $L \approx \log(n/\varepsilon)$.

But is this a good approximation?..

Exponential decay of correlations

Lemma (Correlation decay)

Let $\alpha = 0.981$, C be a monotone CNF formula with maximum degree 5, and x be a variable. Then

$$|R(C, x, L) - R(C, x)| \leq 5\sqrt{6}\alpha^L. \quad (1)$$

Proof by induction?

$$|R(C, x, L + 1) - R(C, x)| \leq \alpha |R(C, x, L) - R(C, x)|. \quad (2)$$

☹ Unfortunately this is not true, even for very simple graph families.

FIX: amortized analysis

Let $\varphi(x) \triangleq 2 \sinh^{-1}(\sqrt{x})$ be a potential function, we analyze $\varphi \circ R \circ \varphi^{-1}$.

Exponential decay of correlations

Lemma (Correlation decay)

Let $\alpha = 0.981$, C be a monotone CNF formula with maximum degree 5, and x be a variable. Then

$$|R(C, x, L) - R(C, x)| \leq 5\sqrt{6}\alpha^L. \quad (1)$$

Proof by induction?

$$|R(C, x, L + 1) - R(C, x)| \leq \alpha |R(C, x, L) - R(C, x)|. \quad (2)$$

☹ Unfortunately this is not true, even for very simple graph families.

FIX: amortized analysis

Let $\varphi(x) \triangleq 2 \sinh^{-1}(\sqrt{x})$ be a potential function, we analyze $\varphi \circ R \circ \varphi^{-1}$.

Exponential decay of correlations

Lemma (Correlation decay)

Let $\alpha = 0.981$, C be a monotone CNF formula with maximum degree 5, and x be a variable. Then

$$|R(C, x, L) - R(C, x)| \leq 5\sqrt{6}\alpha^L. \quad (1)$$

Proof by induction?

$$|R(C, x, L + 1) - R(C, x)| \leq \alpha |R(C, x, L) - R(C, x)|. \quad (2)$$

☹ Unfortunately this is not true, even for very simple graph families.

FIX: amortized analysis

Let $\varphi(x) \triangleq 2 \sinh^{-1}(\sqrt{x})$ be a potential function, we analyze $\varphi \circ R \circ \varphi^{-1}$.

Exponential decay of correlations

Lemma (Correlation decay)

Let $\alpha = 0.981$, C be a monotone CNF formula with maximum degree 5, and x be a variable. Then

$$|R(C, x, L) - R(C, x)| \leq 5\sqrt{6}\alpha^L. \quad (1)$$

Proof by induction?

$$|R(C, x, L + 1) - R(C, x)| \leq \alpha |R(C, x, L) - R(C, x)|. \quad (2)$$

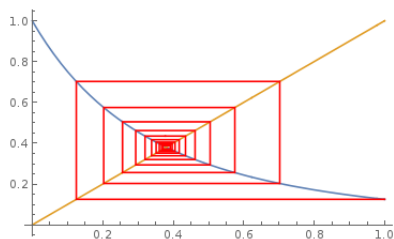
☹ Unfortunately this is not true, even for very simple graph families.

FIX: amortized analysis

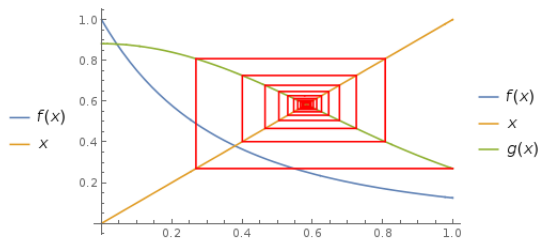
Let $\varphi(x) \triangleq 2 \sinh^{-1}(\sqrt{x})$ be a potential function, we analyze $\varphi \circ R \circ \varphi^{-1}$.

Exponential decay of correlations

Consider a very special case: the graph is a symmetric, infinite d -ary tree.
The recursion: $f(x) = \left(\frac{1}{1+x}\right)^d$. Amortized: $g(x) = \varphi(f(\varphi^{-1}(x)))$.



(a) Original function.



(b) Amortized step-wise decay.

Figure: Graphical dynamics.

Conclusion and Open Questions

- We have a dichotomy for approximately counting monotone CNF.
- We also have a partial result in the set packing region (3D matching).

Questions:

- The limit of the correlation decay technique?
- The classification on approximability of monotone #CSP?
- Could we do better on monotone systems with additional local structures e.g. dominating sets?