From Weighted to Unweighted Model Counting

Supratik Chakraborty[1], Dror Fried[2], **Kuldeep S. Meel[2]**, Moshe Y. Vardi[2]

[1]Indian Institute of Technology Bombay, India
[2]Rice University

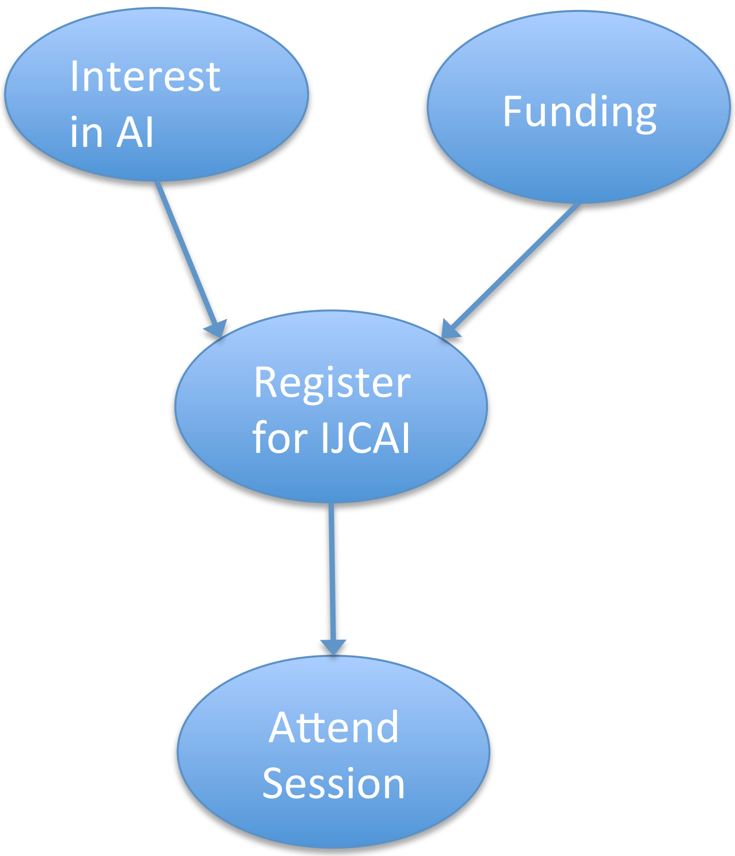
(Author names are ordered alphabetically by last name)

Jul 29, 2015

Internet of Things

How do we infer useful information from the data filled with uncertainty?

Probabilistic Inference



Modeling Attendance in IJCAI Talk

Roth, 1996

Weighted Model Counting

Unweighted Model Counting (UMC)

- Unweighted model counting: Given a Boolean Formula F, count the number of models of F .
- $R_F := \{(a=0, b=1), (a=1, b=0), (a=1, b=1)\}$
- $| R_F | = 3$
- #P-complete
  - #P: Class of counting problem whose decision problems lie in NP

Weighted Model Counting (WMC)

- Given a formula F and weight function W over literals

$$F = (a \vee b)$$

$$W(a=0) = 1/4;\ W(a=1) = 1 - W(a=0) = 3/4$$

$$W(b=0) = 3/8;\ W(b=1) = 5/8$$

- Weight of assignment = Product of weight of literals

$$W(a=0, b=1) = 1/4 * 5/8 = 5/32$$

- Weighted Model Counting: Sum of weight of assignments

Weighted Model Counting (WMC)

- Given a formula F and weight function W over literals

$$F = (a \vee b)$$

$$W(a=0) = 1/4; W(a=1) = 1 - W(a=0) = 3/4$$

$$W(b=0) = 3/8; W(b=1) = 5/8$$

- Weighted Model Counting: Sum of weight of assignments

$$W(R_F) = W(a=0, b=1) + W(a=1, b=0) + W(a=1, b=1)$$

$$= {}^{1}\!/_{4} * {}^{5}\!/_{8} + {}^{3}\!/_{4} * {}^{3}\!/_{8} + {}^{3}\!/_{4} * {}^{5}\!/_{8} = {}^{29}\!/_{32}$$

Problem Statement

- Motivation: Probabilistic Inference can be reduced to Weighted Model Counting (WMC)

- Weighted Model Counting (WMC): Given a formula F and weight function W over literals, compute sum of weight of assignments

- Problem: Design efficient algorithms for WMC

Outline

- Motivation
- Prior work
- Approach: reduce weighted to unweighted counting
- Theoretical Implications
- Experimental Results

Prior Work

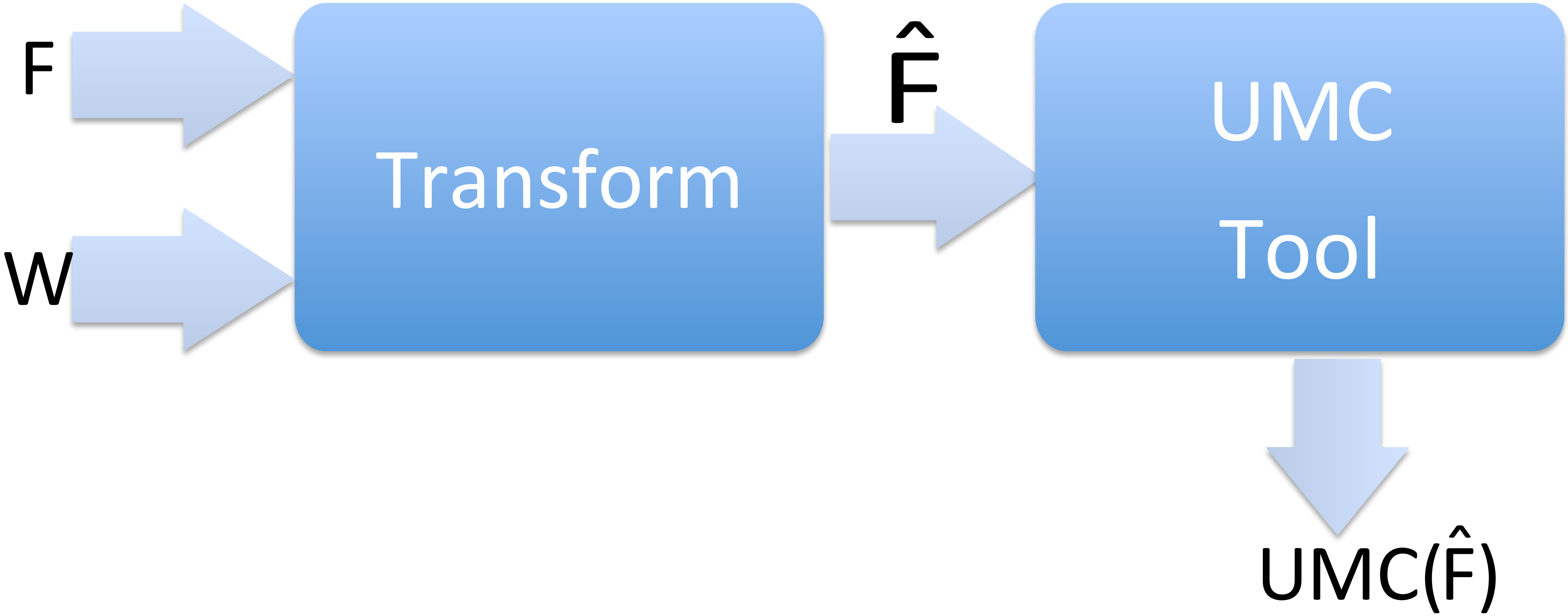
- UMC & WMC are #P-complete (Roth 1996)
- UMC solvers based on:
  - Component Caching
  - BDD-based techniques
  - Clauses learning, no-good learning etc..
- Examples: CDP, Relsat, Cachet, SharpSAT, DSharp

Prior Work

- UMC solvers have been ***manually*** adapted to WMC:
  - Cachet
  - SDD

  (Sang et al., 2005, Choi and Darwiche, 2013, Chakraborty et al., 2014)
- Manual adaptation requires intimate understanding of the UMC implementation techniques

Our Contribution



$$WMC(F, W) = C_F * UMC(\hat{F}) + D_F$$

Our Contributions

- Efficient polynomial time reduction from WMC to UMC
- Allows usage of any UMC solver, viewed as a black box, to compute WMC for a given formula
- Has theoretical guarantees of optimality
- Implementation scales to significantly larger formulas than prior state-of-art WMC solvers

Key Idea

Let $W(X_i = 1) = k_i / 2^{m_i}$ and $\hat{m} = \sum m_i$

1. Construct a formula Ω over $\{X_1, \ldots. X_n, a_1, \ldots a_{\hat{m}}\}$ such that every partial assignment over X_i is extended to $\prod k_i$ satisfying assignments

2. Intersection of F and Ω gives us the desired formula

How to construct Ω ?

Let X= $\{X_1\}; W(X_1 = 1) = 1/4$ and $W(X_2 = 1) = 3/4$

Consider $\Omega := ((X_1 \leftrightarrow (a_1 \wedge a_2)) \wedge (X_2 \leftrightarrow (a_3 \vee a_4))$

Partial assignment $X_1 = 1,\ X_2 = 1$ extends to 3 (= 3*1)

satisfying assignments:

1. $X_1 = 1,\ X_2 = 1,\ a_1 = 1,\ a_2 = 1,\ a_3 = 1,\ a_4 = 1$
2. $X_1 = 1,\ X_2 = 1,\ a_1 = 1,\ a_2 = 1,\ a_3 = 1,\ a_4 = 0$
3. $X_1 = 1,\ X_2 = 1,\ a_1 = 1,\ a_2 = 1,\ a_3 = 0,\ a_4 = 1$

How to construct Ω ?

Let $X = \{X_1\}; W(X_1 = 1) = \frac{1}{4}$ and $W(X_2 = 1) = \frac{3}{4}$

Consider $\Omega := ((X_1 \leftrightarrow (a_1 \wedge a_2)) \wedge (X_2 \leftrightarrow (a_3 \vee a_4))$

More generally,

Let $W(X_i = 1) = \frac{k_i}{2^{mi}}$ and $\hat{m} = \sum m_i$

$\Omega(a_{1,} \ldots . a_{\hat{m}}) := \bigwedge_i (X_i \leftrightarrow \Phi(k_i, m_i))$

where, $\Phi(k_i, m_i)$ is formula over m_i with k_i satisfying assignments

Construction of Unweighted formula

$\hat{F} = F \wedge \Omega$

$W(F) = C_{F^*} |R_{\hat{F}}|$, where C_F is a constant

Theorem1 : Ω can be expressed in CNF in polynomial time and $O(\Sigma m_i^{\ 2})$ size

Theorem 2: If F is in CNF, then $\hat{F}$ can be tranformed into CNF in polynomial time

What about DNF?

$$\tilde{F} = \Omega \rightarrow F \equiv (\neg\Omega \vee F)$$

$$W(F) = C_{F^*} |R_{\tilde{F}}| + D_F$$

Theorem 3: If F is in DNF, then $\tilde{F}$ can be tranformed into DNF in polynomial time

Is the Transformation Optimal?

Theorem 4: Let $W(X_i) = k_i / 2^{m_i}$ and $\hat{m} = \sum m_i$.

Let Reduce(F,W) be an algorithm that returns F′

such that $W(F) = C_{F*} |R_{F'}| + D_{W}$. *Then* F′

has at least n+m̂-k variables where k is is

independent of n and m

Theorem 5: The given tranformation results in

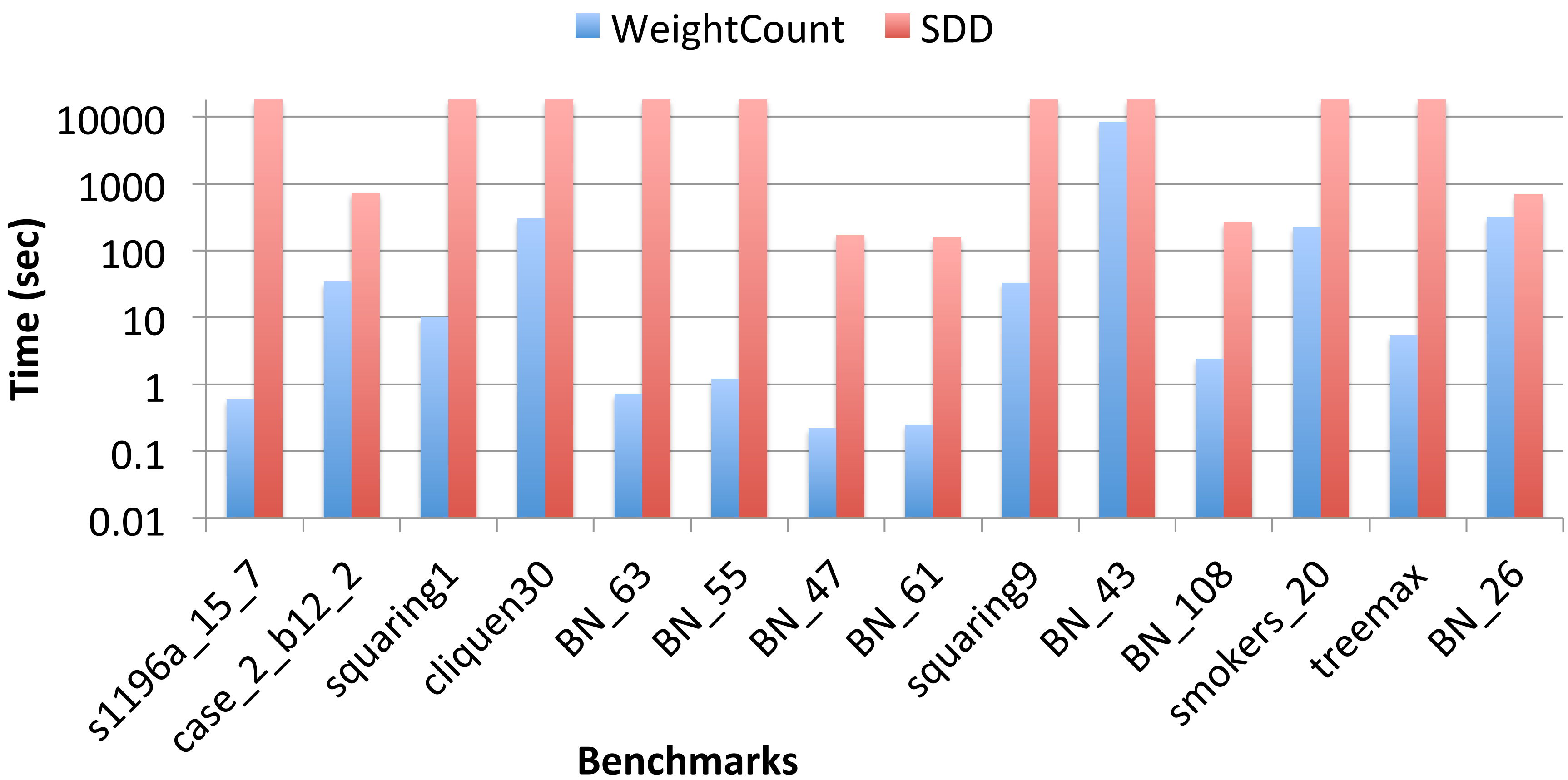
formulas F̂ and F̃ with n+m̂ variables.

Experimental Results

- Experiments over diverse set of benchmarks consisting of
  - Grid networks
  - Ising models
  - Plan recognition
  - Program synthesis
- WMC Solver for comparison: SDD
- UMC Tools: SharpSAT, DSharp

*Cachet's WMC version is broken

Runtime Comparison



Conclusion

- Reduction from probabilistic inference to WMC
- Prior work required **manual** adaptation of UMC techniques to WMC
- Polynomial time transformation from UMC to WMC
- The resulting tool, WeightCount, outperform state-of-the-art counters such as SDD by 1-2 orders of magnitude