A Quick Intro to (Weighted) Model Counting

Why Probabilistic Reasoning?

- Probability is our best framework for dealing with uncertainty.
- We need probabilistic reasoning for integrating not only data from uncertain physical sources but also from the variety of machine learning algorithms.

Why SAT-Based Techniques?

- Together with (M)ILP, SAT is one of the most mature technologies for solving complex reasoning problems.

Content

- In this short lecture, we will learn about the following problems and their applications in **probabilistic reasoning**:
  - SAT (you probably know this one)
  - (Weighted) Model counting (#SAT)

Quick Refresher: Propositional Logic

- Propositional formulas are constructed (using the standard rules) from propositional variables, constants 0 (false) and 1 (true), logical connectives $\vee$ (disjunction), $\wedge$ (conjunction), $\neg$ (negation) and parentheses.

Quick Refresher: Propositional Logic

- Propositional formulas are constructed (using the standard rules) from propositional variables, constants 0 (false) and 1 (true), logical connectives $\vee$ (disjunction), $\wedge$ (conjunction), $\neg$ (negation) and parentheses.
- For instance $(a \vee b) \wedge c$ is a propositional formula.

Quick Refresher: Propositional Logic

- Propositional formulas are constructed (using the standard rules) from propositional variables, constants 0 (false) and 1 (true), logical connectives $\vee$ (disjunction), $\wedge$ (conjunction), $\neg$ (negation) and parentheses.

- For instance $(a \vee b) \wedge c$ is a propositional formula.

- A formula is called “clause” if it is a disjunction of variables or their negations, e.g. $(a \vee \neg b)$ is a clause (*note: a single variable or its negation are also clauses*).

CNF and DNF

- A formula is in **CNF** form (conjunctive normal form) if it is a conjunction of clauses.
- For example:

$(a \vee b) \wedge c$ is in CNF

$(a \vee (b \wedge d)) \wedge c$ is not in CNF

CNF and DNF

- A formula is in **CNF** form (conjunctive normal form) if it is a conjunction of clauses.

- A formula is in **DNF** form (disjunctive normal form) if it is a disjunction of conjunctions of variables or their negations.

- For example:

  $(a \wedge b) \vee (a \wedge \neg c)$ is in **DNF**

Satisfiability (SAT)

- **SAT Problem:** Is there an assignment of truth values to the propositional variables which would make the formula evaluate to true?

- For example:

  Is $(a \vee b) \wedge c$ *satisfiable?*

Satisfiability (SAT)

- **SAT Problem:** Is there an assignment of truth values to the propositional variables which would make the formula evaluate to true?
- For example: ??

  Is $(a \vee b) \wedge c$ *satisfiable?*

Satisfiability (SAT)

- **SAT Problem:** Is there an assignment of truth values to the propositional variables which would make the formula evaluate to true?

- For example:

  Is *(a ∨ b) ∧ c satisfiable? Yes, it is, e.g. by setting (a=1,b=0,c=1) or (a=0,b=1,c=1) or (a=1,b=1,c=1).*

How Hard is “DNF SAT”?

Reminder (DNF): $(a \wedge b) \vee (a \wedge \neg c) \vee (d \wedge \neg c)$.

How Hard is “DNF SAT”?

Reminder (DNF): $(a \wedge b) \vee (a \wedge \neg c) \vee (d \wedge \neg c)$.

Easy: **P**

How Hard is CNF SAT?

Reminder (CNF): $(a \vee b \vee \neg c) \wedge (\neg a \vee c)$.

How Hard is CNF SAT?

Reminder (CNF): *(a ∨ b ∨ ¬c) ∧ (¬a ∨ c).*

Hard: **NP**-complete

How Hard is CNF SAT?

Reminder (CNF): $(a \vee b \vee \neg c) \wedge (\neg a \vee c)$.

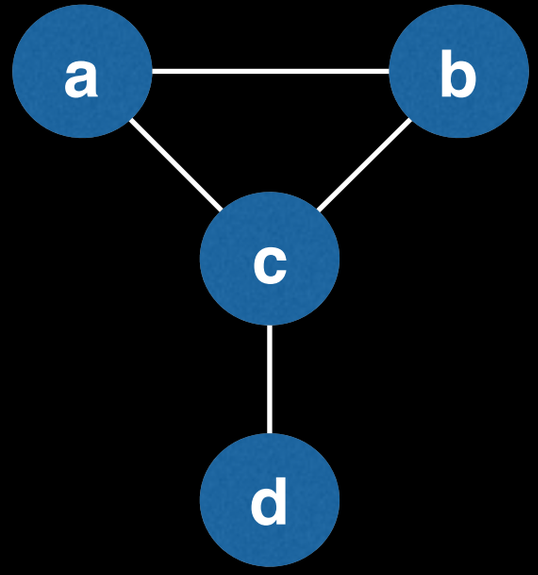
Hard: **NP**-complete

But often very fast in practice using modern SAT solvers!

Probabilistic Reasoning Using Model Counting

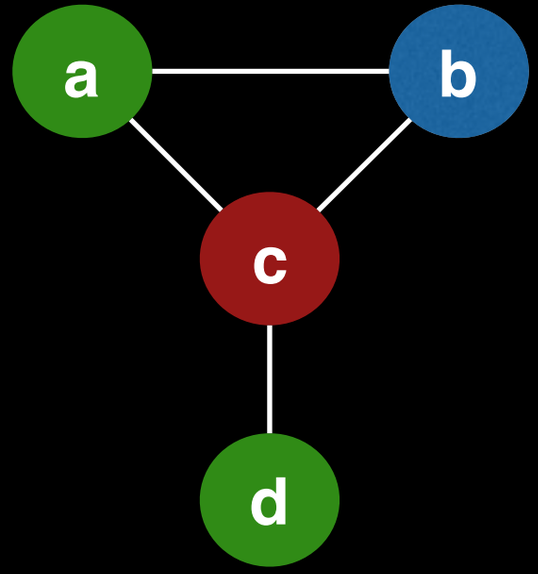
Simple Example

- We want to colour the following graph using 3 colours so that no two connected vertices have the same colour.



Simple Example

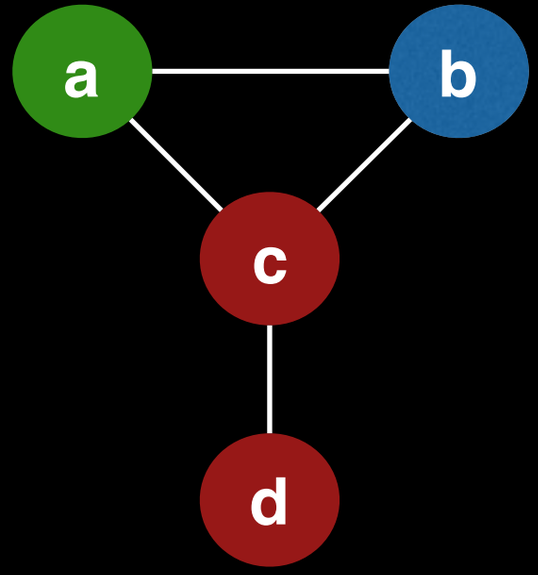
- We want to colour the following graph using 3 colours so that no two connected vertices have the same colour.



ok

Simple Example

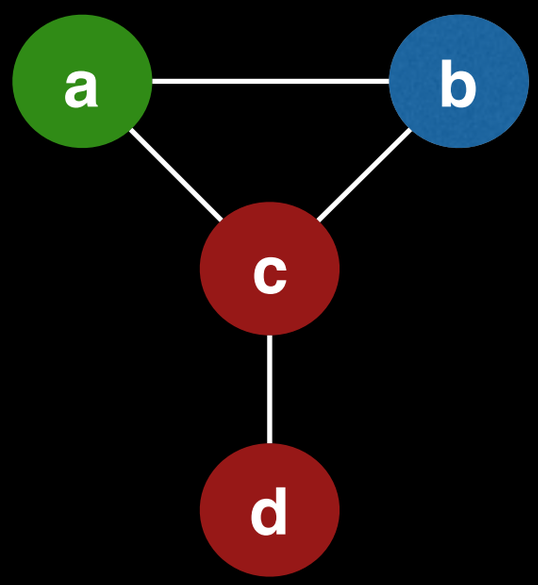
- We want to colour the following graph using 3 colours so that no two connected vertices have the same colour.



wrong

Simple Example

- We want to colour the following graph using 3 colours so that no two connected vertices have the same colour.



wrong

- But we are lazy and we just colour randomly. **How likely is that we find the right answer?**

Computing the Probability

$$P_{\text{correct}} = \frac{\#\text{Correct colourings}}{\#\text{All colourings}}$$

Computing the Probability

$$P_{\text{correct}} = \frac{\#\text{Correct colourings}}{\#\text{All colourings}}$$

We can compute the numerator and also denominator using model counting *(although here we wouldn't have to - it is simple)*.

And We Use Model Counting…

- But, first, what is model counting?
- Propositional model counting (#SAT) is the problem of computing the number of distinct truth assignments (“models”) to variables for which the formula evaluates to true.

Ex: Do you remember how many “models” the formula

$$(a \vee b) \wedge c$$

had?

And We Use Model Counting…

- But, first, what is model counting?
- Propositional model counting (#SAT) is the problem of computing the number of distinct truth assignments (“models”) to variables for which the formula evaluates to true.

Ex: Do you remember how many “models” the formula

$$(a \vee b) \wedge c$$

had?

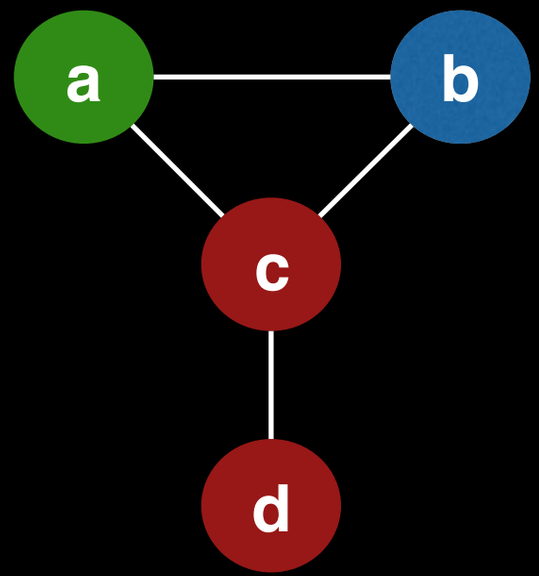
Answer: ***3*** *(a=1,b=0,c=1), (a=0,b=1,c=1), (a=1,b=1,c=1).*

Note: #CNF and #DNF

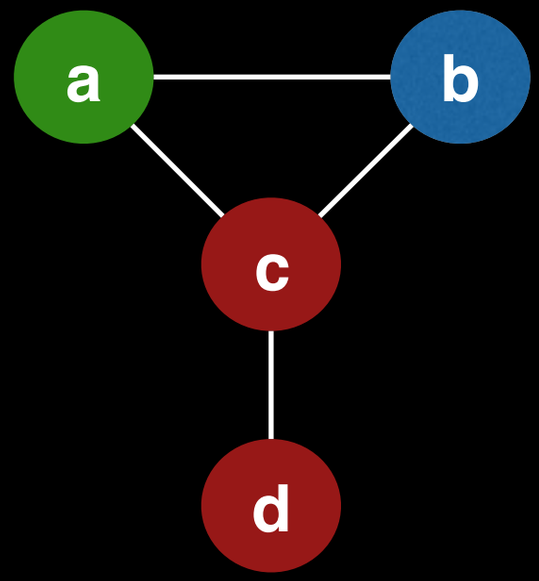
- Unlike for the decision problem, #DNF is hard to compute (#P-complete).
- However, it is still “easier” than #CNF in the sense that there is a FPRAS for #DNF.

Back to Our Problem…

- SAT encoding of the colouring problem:
  - Variables: aRed, aGreen, aBlue, bRed, …



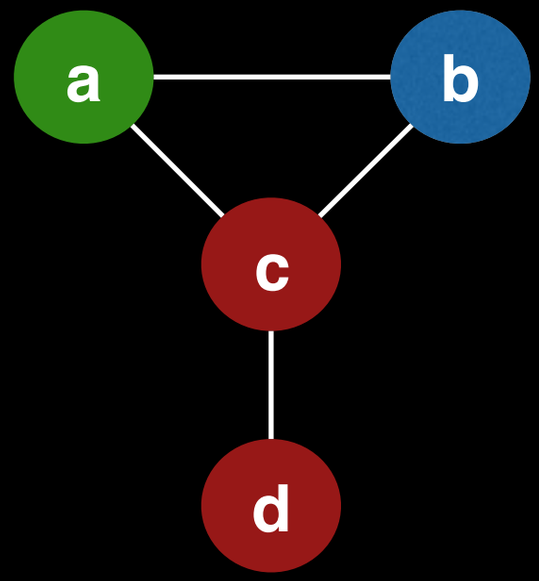
Back to Our Problem…



- SAT encoding of the colouring problem:
  - Variables: aRed, aGreen, aBlue, bRed, …
  - Every node has at least one colour:

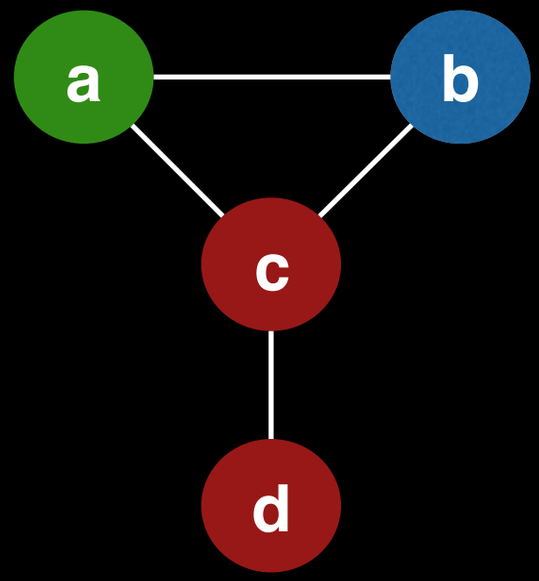
(aRed ∨ *aGreen* ∨ *aBlue)* ∧ (bRed ∨ *bGreen* ∨ *bBlue)* ∧ …

Back to Our Problem…



- SAT encoding of the colouring problem:
  - Variables: aRed, aGreen, aBlue, bRed, …
  - Every node has at least one colour:
    (aRed ∨ *aGreen* ∨ *aBlue)* ∧ (bRed ∨ *bGreen* ∨ *bBlue)* ∧ …
  - *Every node has at most one colour:*
    (¬ aRed ∨ ¬ aGreen) ∧ *(¬ aRed ∨ ¬ aBlue)* ∧ *(¬ aGreen ∨ aBlue)* ∧ …

Back to Our Problem…



- SAT encoding of the colouring problem:
  - Variables: aRed, aGreen, aBlue, bRed, …
  - Every node has at least one colour:
    (aRed ∨ *aGreen* ∨ *aBlue)* ∧ (bRed ∨ *bGreen* ∨ *bBlue)* ∧ …
  - *Every node has at most one colour:*
    (¬ aRed ∨ *¬ aGreen)* ∧ *(¬ aRed* ∨ *¬ aBlue)* ∧ *(¬ aGreen* ∨ *aBlue)* ∧ …
  - *Connected nodes have different colours:*
    (¬ aRed ∨ *¬* b*Red)* ∧ *(¬ aGreen* ∨ *¬* b*Green)* ∧ *(¬ aBlue* ∨ *¬* b*Blue)* ∧ …

Now We Can Compute the Probability…

$$P_{\text{correct}} = \frac{\#\text{Correct colourings}}{\#\text{All colourings}} = \frac{12}{81} \approx 0.15$$

- Computed as model count without the condition:
  "*Connected nodes have different colours*":
  (¬ aRed ∨ ¬ bRed) ∧ (¬ aGreen ∨ ¬ bGreen) ∧ (¬ aBlue ∨ ¬ bBlue) ∧ …

Weighted Model Counting

- For each propositional variable x, we define weights *w(x)* and *w(¬x)* (some authors require *w(x)=1-w(¬x)*)
- We then define

$$WMC(\Phi) = \sum_{v \in \mathrm{Mod}(\Phi)} \prod_{x:v(x)=1} w(x) \cdot \prod_{x:v(x)=0} w(\neg x)$$

WMC Can Do A Lot!

- The example with colouring was only about uniform distribution (+ combinatorial constraints).
- WMC can do more!
- **Using WMC, we can do:**
  - Marginal and conditional inference in Markov and Bayesian networks.
  - Marginal and conditional inference **in probabilistic (logic) programs!**

Some Existing (W)MC Systems

- Chakraborty et al.: ApproxMC2 (uses SAT solver as an oracle)
- Sang et al,: Cachet (exploits connected components caching)
- Darwiche: SDD (compilation-based approach)
- ...