(Exact) Lifted inference

David Poole

Department of Computer Science,
University of British Columbia

Work with: http://minervaintelligence.com, https://treatment.com/

February 25, 2020

1. Representations: Problog & MLNs
2. Representation Issues
3. **(Exact) Lifted Inference**
4. Lifted Approximate Inference and Optimization
5. Learning
6. Applications

(Exact) Lifted Inference

Outline

1. (Exact) Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning

Why Exact Inference?

Why do we care about exact inference?

- Gold standard

Why Exact Inference?

Why do we care about exact inference?

- Gold standard
- Size of problems amenable to exact inference is growing

Why Exact Inference?

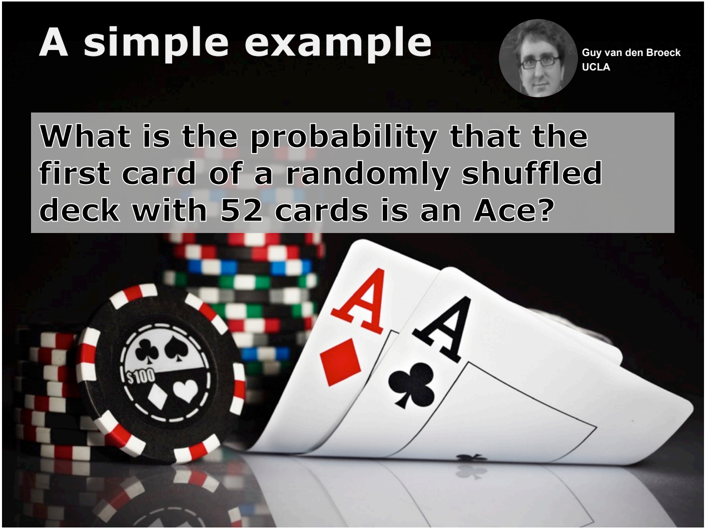
Why do we care about exact inference?

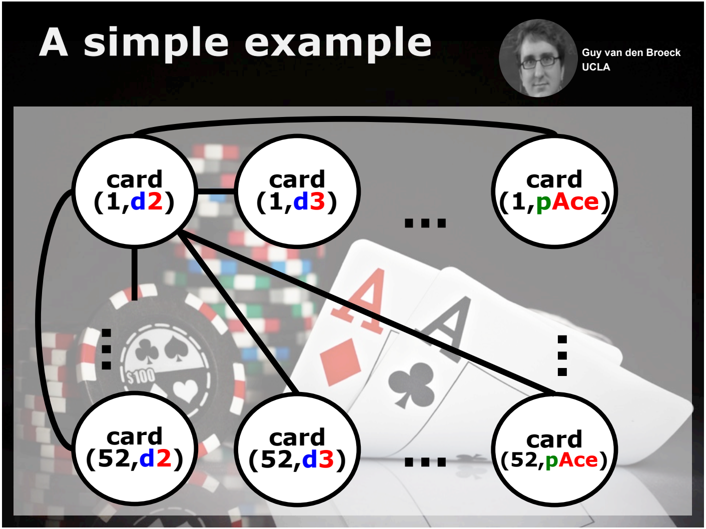
- Gold standard
- Size of problems amenable to exact inference is growing
- Learning for inference

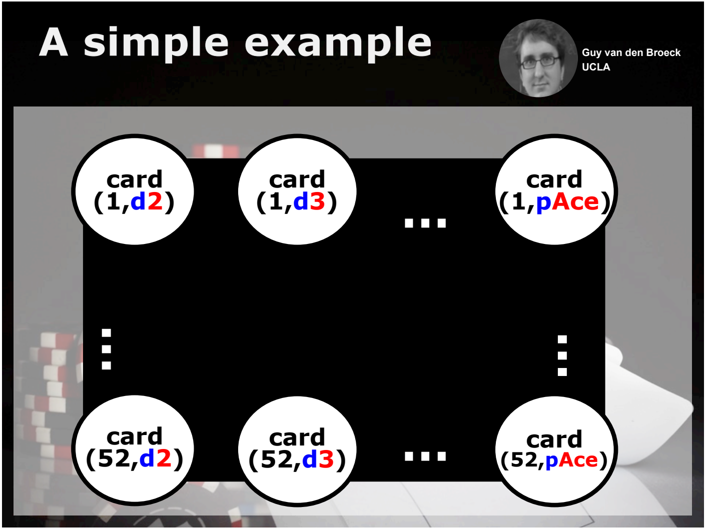
Why Exact Inference?

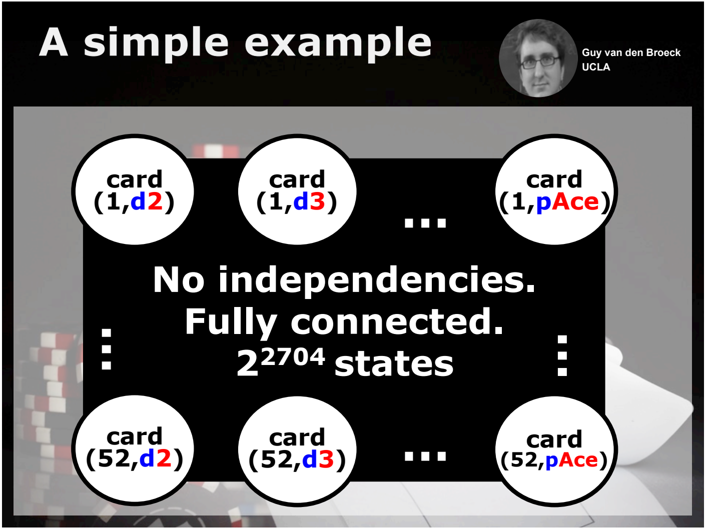
Why do we care about exact inference?

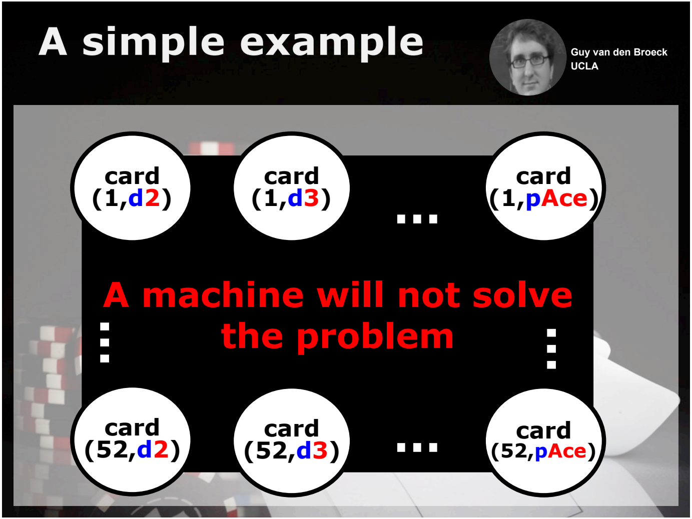
- Gold standard
- Size of problems amenable to exact inference is growing
- Learning for inference
- Basis for efficient approximate inference:
  - Rao-Blackwellization
  - Variational Methods

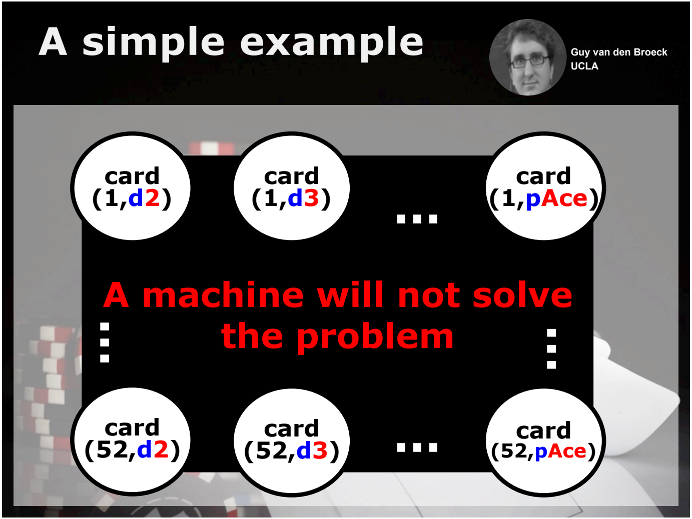

A simple example
Guy van den Broeck
UCLA
What is the probability that the first card of a randomly shuffled deck with 52 cards is an Ace?
$100


A simple example
Guy van den Broeck
UCLA
card
(1,d2)
card
(1,d3)
...
card
(1,pAce)
card
(52,d2)
card
(52,d3)
...
card
(52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
card (52,d2)
card (52,d3)
...
card (52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
No independencies.
Fully connected.
2^2704 states
card (52,d2)
card (52,d3)
...
card (52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
A machine will not solve the problem
card (52,d2)
card (52,d3)
...
card (52,pAce)



. . . unless it can represent and exploit symmetry.

Example: lifted inference

- Consider determining the guilt of someone fitting the description of a person who committed a crime.
- Suppose the probability of someone at random matching the description is one in a million.

Example: lifted inference

- Consider determining the guilt of someone fitting the description of a person who committed a crime.
- Suppose the probability of someone at random matching the description is one in a million.
- The probability this person committed the crime depends on how many people there are:
    - If there were a thousand other people,

Example: lifted inference

- Consider determining the guilt of someone fitting the description of a person who committed a crime.
- Suppose the probability of someone at random matching the description is one in a million.
- The probability this person committed the crime depends on how many people there are:
  - If there were a thousand other people, it is very unlikely there was someone else who committed the crime.
  - If there was a population of 10 million,

Example: lifted inference

- Consider determining the guilt of someone fitting the description of a person who committed a crime.
- Suppose the probability of someone at random matching the description is one in a million.
- The probability this person committed the crime depends on how many people there are:
  - If there were a thousand other people, it is very unlikely there was someone else who committed the crime.
  - If there was a population of 10 million, then we would expect that there would be 10 people who fit the description, and so the probability that this suspect was guilty would be around 10%.

Example: lifted inference

- Consider determining the guilt of someone fitting the description of a person who committed a crime.
- Suppose the probability of someone at random matching the description is one in a million.
- The probability this person committed the crime depends on how many people there are:
  - If there were a thousand other people, it is very unlikely there was someone else who committed the crime.
  - If there was a population of 10 million, then we would expect that there would be 10 people who fit the description, and so the probability that this suspect was guilty would be around 10%.
- We don't need to reason about all of the other individuals separately, but can count over them.

Example: lifted inference

- Suppose someone is giving a presentation, and three people out of 100 people in the audience asked a question (so 97 people were observed to not ask a question).

Example: lifted inference

- Suppose someone is giving a presentation, and three people out of 100 people in the audience asked a question (so 97 people were observed to not ask a question).
- A reasonable model about the eloquence of the speaker might depend on the questions asked
- each of the people who didn't ask a question, their silence might depend on the questions asked, but not on the questions not asked.
- Rather than reasoning separately about each person who was observed to not ask a question, it is reasonable to just count over them.

Example: lifted inference

- The spread of a malaria (or other diseases) may depend on the number of people and the number of mosquitoes.
- Individual mosquitoes are important in such a model, but we don't want to model each mosquito separately.

Lifted Inference

- Idea: treat those individuals about which you have the same information as a block; just count them.

Lifted Inference

- Idea: treat those individuals about which you have the same information as a block; just count them.
- Use the ideas from lifted theorem proving - no need to ground.

Lifted Inference

- Idea: treat those individuals about which you have the same information as a block; just count them.
- Use the ideas from lifted theorem proving - no need to ground.
- Potential to be exponentially faster in the number of non-differentialed individuals.

Lifted Inference

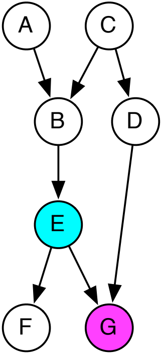
- Idea: treat those individuals about which you have the same information as a block; just count them.
- Use the ideas from lifted theorem proving - no need to ground.
- Potential to be exponentially faster in the number of non-differentialed individuals.
- Relies on knowing the number of individuals (the population size).

Outline

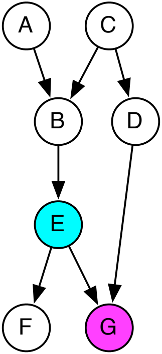
1. (Exact) Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning

Inference via factorization in graphical models

$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$



Inference via factorization in graphical models



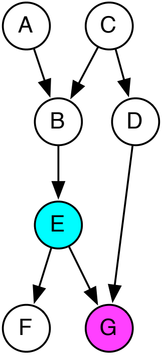
$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

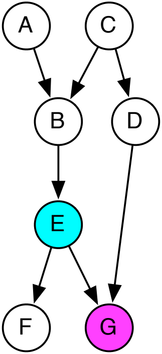
$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

$$\left(\sum_D P(D \mid C)P(g \mid ED) \right) \Bigg)$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

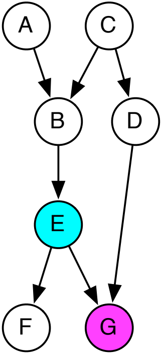
$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

$$\left(\sum_A P(A)P(B \mid AC)\right)$$

$$\left(\sum_D P(D \mid C)P(g \mid ED)\right)\Bigg)\Bigg)$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

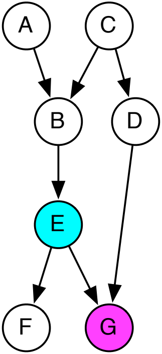
$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

$$\sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC) \right) \left(\sum_D P(D \mid C)P(g \mid ED) \right) \right)$$

Inference via factorization in graphical models



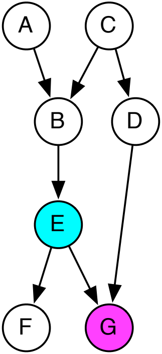
$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$= \sum_B P(E \mid B) \sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC) \right) \left(\sum_D P(D \mid C)P(g \mid ED) \right) \right)$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$= \left(\sum_F P(F \mid E)\right) \sum_B P(E \mid B) \sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC)\right) \left(\sum_D P(D \mid C)P(g \mid ED)\right)\right)$$

Recursive Conditioning

- Computes sum (partition function) from outside in

Input:

- Context - assignment of values to variables
- Set of factors

Output: value of summing out other variables (partition function)

- Evaluate a factor as soon as all its variables are assigned
- Cache values already computed
- Recognize disconnected components
- Recursively branch on a variable

Recursive Conditioning

procedure $rc(Con : \text{context}, \quad Fs : \text{set of factors})$:
- if $\exists v$ such that $\langle\langle Con, Fs\rangle, v\rangle \in cache$
  - return v
- else if $vars(Con) \not\subseteq vars(Fs)$
  - return $rc(\{X = v \in Con : X \in vars(Fs)\}, Fs)$
- else if $\exists F \in Fs$ such that $vars(F) \subseteq vars(Con)$
  - return $eval(F, Con) \times rc(Con, Fs \setminus \{F\})$
- else if $Fs = Fs_1 \uplus Fs_2$ where $vars(Fs_1) \cap vars(Fs_2) \subseteq vars(Con)$
  - return $rc(Con, Fs_1) \times rc(Con, Fs_2)$
- else select variable $X \in vars(Fs)$
  - $sum \leftarrow 0$
  - for each $v \in domain(X)$
    - $sum \leftarrow sum + rc(Con \cup \{X = v\}, Fs)$
  - $cache \leftarrow cache \cup \{\langle\langle Con, Fs\rangle, sum\rangle\}$
  - return sum

Recursive Conditioning

procedure $rc(Con : \text{context}, \quad Fs : \text{set of factors})$:
  if $\exists v$ such that $\langle\langle Con, Fs\rangle, v\rangle \in cache$
    return v
  else if $vars(Con) \not\subseteq vars(Fs)$
    return $rc(\{X = v \in Con : X \in vars(Fs)\}, Fs)$
  else if $\exists F \in Fs$ such that $vars(F) \subseteq vars(Con)$
    return $eval(F, Con) \times rc(Con, Fs \setminus \{F\})$ Evaluate
  else if $Fs = Fs_1 \uplus Fs_2$ where $vars(Fs_1) \cap vars(Fs_2) \subseteq vars(Con)$
    return $rc(Con, Fs_1) \times rc(Con, Fs_2)$
  else select variable $X \in vars(Fs)$ Branch
    $sum \leftarrow 0$
    for each $v \in domain(X)$
      $sum \leftarrow sum + rc(Con \cup \{X = v\}, Fs)$
    $cache \leftarrow cache \cup \{\langle\langle Con, Fs\rangle, sum\rangle\}$
    return sum

Recursive Conditioning

procedure $rc(Con : \text{context}, \quad Fs : \text{set of factors})$:

- if $\exists v$ such that $\langle\langle Con, Fs\rangle, v\rangle \in cache$ — Recall
  - return v
- else if $vars(Con) \not\subseteq vars(Fs)$ — Forget
  - return $rc(\{X = v \in Con : X \in vars(Fs)\}, Fs)$
- else if $\exists F \in Fs$ such that $vars(F) \subseteq vars(Con)$
  - return $eval(F, Con) \times rc(Con, Fs \setminus \{F\})$
- else if $Fs = Fs_1 \uplus Fs_2$ where $vars(Fs_1) \cap vars(Fs_2) \subseteq vars(Con)$
  - return $rc(Con, Fs_1) \times rc(Con, Fs_2)$
- else select variable $X \in vars(Fs)$
  - $sum \leftarrow 0$
  - for each $v \in domain(X)$
    - $sum \leftarrow sum + rc(Con \cup \{X = v\}, Fs)$
  - $cache \leftarrow cache \cup \{\langle\langle Con, Fs\rangle, sum\rangle\}$ — Remember
  - return sum

Recursive Conditioning

procedure $rc(Con : \text{context}, \quad Fs : \text{set of factors})$:
  if $\exists v$ such that $\langle\langle Con, Fs\rangle, v\rangle \in cache$
    return v
  else if $vars(Con) \not\subseteq vars(Fs)$
    return $rc(\{X = v \in Con : X \in vars(Fs)\}, Fs)$
  else if $\exists F \in Fs$ such that $vars(F) \subseteq vars(Con)$
    return $eval(F, Con) \times rc(Con, Fs \setminus \{F\})$
  else if $Fs = Fs_1 \uplus Fs_2$ where $vars(Fs_1) \cap vars(Fs_2) \subseteq vars(Con)$
    return $rc(Con, Fs_1) \times rc(Con, Fs_2)$ Disconnected
  else select variable $X \in vars(Fs)$
    $sum \leftarrow 0$
    for each $v \in domain(X)$
      $sum \leftarrow sum + rc(Con \cup \{X = v\}, Fs)$
    $cache \leftarrow cache \cup \{\langle\langle Con, Fs\rangle, sum\rangle\}$
    return sum

Variable Elimination and Recursive Conditioning

- Variable elimination is the dynamic programming variant of recursive conditioning.
- Recursive Conditioning is the search variant of variable elimination
- They do the same additions and multiplications.
- Complexity $O(nr^t)$, for n variables, range size r, and treewidth t.

Outline

Weighted Formula

A Weighted formula is a pair $\langle F, v\rangle$ where

- F a formula on parametrized random variables
- v number

Example:

$\langle X \neq Y \land likes(X,Y) \land rich(Y), 0.001\rangle$
$\langle likes(X,X) \land rich(X), 0.7\rangle$

...

Lifted Recursive Conditioning

LiftedRC(Context, WeightedFormulas)

- *Context* is a set of assignments to random variables and counts to assignments of instances of relations. e.g.:

$$\begin{aligned}\{\neg a,\ & \#_X f(X) \land g(X) = 7, \\ & \#_X f(X) \land \neg g(X) = 5, \\ & \#_X \neg f(X) \land g(X) = 18, \\ & \#_X \neg f(X) \land \neg g(X) = 0\}\end{aligned}$$

- *WeightedFormulas* is a set of weighted formulae, e.g.,

$$\begin{aligned}\{ & \langle \neg a \land \neg f(X) \land g(X), 0.1 \rangle, \\ & \langle a \land \neg f(X) \land g(X), 0.2 \rangle, \\ & \langle f(X) \land g(Y), 0.3 \rangle, \\ & \langle f(X) \land h(X), 0.4 \rangle \}\end{aligned}$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \land g(X) = 7,$$
$$\#_X f(X) \land \neg g(X) = 5,$$
$$\#_X \neg f(X) \land g(X) = 18,$$
$$\#_X \neg f(X) \land \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \land \neg f(X) \land g(X), 0.1 \rangle,$$
$$\langle a \land \neg f(X) \land g(X), 0.2 \rangle,$$
$$\langle f(X) \land g(Y), 0.3 \rangle,$$
$$\langle f(X) \land h(X), 0.4 \rangle \}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7, \\ \#_X f(X) \wedge \neg g(X) = 5, \\ \#_X \neg f(X) \wedge g(X) = 18, \\ \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1\rangle, \\ \langle a \wedge \neg f(X) \wedge g(X), 0.2\rangle, \\ \langle f(X) \wedge g(Y), 0.3\rangle, \\ \langle f(X) \wedge h(X), 0.4\rangle\}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

$$0.1^{18} *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1\rangle,$$
$$\langle a \wedge \neg f(X) \wedge g(X), 0.2\rangle,$$
$$\langle f(X) \wedge g(Y), 0.3\rangle,$$
$$\langle f(X) \wedge h(X), 0.4\rangle\}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

$$0.1^{18} * 1 *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7, \; \#_X f(X) \wedge \neg g(X) = 5, \; \#_X \neg f(X) \wedge g(X) = 18, \; \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle, \; \langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle, \; \langle f(X) \wedge g(Y), 0.3 \rangle, \; \langle f(X) \wedge h(X), 0.4 \rangle \}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

$$0.1^{18} * 1 * 0.3^{12*}$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7, \quad \#_X f(X) \wedge \neg g(X) = 5, \quad \#_X \neg f(X) \wedge g(X) = 18, \quad \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle, \quad \langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle, \quad \langle f(X) \wedge g(Y), 0.3 \rangle, \quad \langle f(X) \wedge h(X), 0.4 \rangle \}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

$$0.1^{18} * 1 * 0.3^{12*25} *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7, \quad \#_X f(X) \wedge \neg g(X) = 5, \quad \#_X \neg f(X) \wedge g(X) = 18, \quad \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle, \quad \langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle, \quad \langle f(X) \wedge g(Y), 0.3 \rangle, \quad \langle f(X) \wedge h(X), 0.4 \rangle \}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

$$0.1^{18} * 1 * 0.3^{12*25} * LiftedRC(Context, \{\langle f(X) \wedge h(X), 0.4 \rangle\})$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \land g(X) = 7,\\ \#_X f(X) \land \neg g(X) = 5,\\ \#_X \neg f(X) \land g(X) = 18,\\ \#_X \neg f(X) \land \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \land h(X), 0.4\rangle, \ldots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \land g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7, \#_X f(X) \wedge \neg g(X) = 5, \#_X \neg f(X) \wedge g(X) = 18, \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \dots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \#_X f(X) \wedge g(X) \wedge h(X) = i, \#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i, \#_X f(X) \wedge \neg g(X) = 5, \dots\}, WeightedFormulas)$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \dots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \#_X f(X) \wedge g(X) \wedge h(X) = i,$$
$$\#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i,$$
$$\#_X f(X) \wedge \neg g(X) = 5, \dots\},$$
$$WeightedFormulas)$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \dots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \#_X f(X) \wedge g(X) \wedge h(X) = i,$$
$$\#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i,$$
$$\#_X f(X) \wedge \neg g(X) = 5, \dots\},$$
$$WeightedFormulas)$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7,\ \#_X f(X) \wedge \neg g(X) = 5,\ \#_X \neg f(X) \wedge g(X) = 18,\ \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

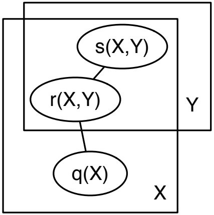
WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \ldots\}$

Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:

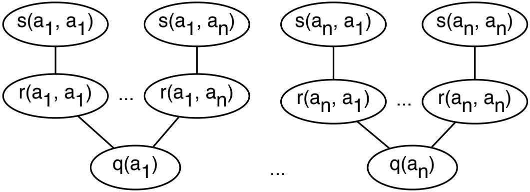
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \#_X f(X) \wedge g(X) \wedge h(X) = i,\ \#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i,\ \#_X f(X) \wedge \neg g(X) = 5, \ldots\},\ WeightedFormulas)$$

Recognizing Disconnectedness



Relational Model

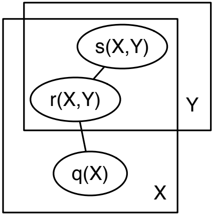


Grounding

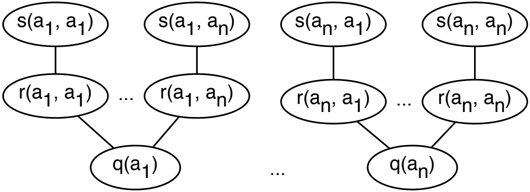
Weighted formulae *WeightedFormulas*:

$$\{\langle\{s(X,Y)\wedge r(X,Y)\}, t_1\rangle$$
$$\langle\{q(X)\wedge r(X,Y)\}, t_2\rangle\}$$

Recognizing Disconnectedness



Relational Model



Grounding

Weighted formulae *WeightedFormulas*:

$$\{\langle\{s(X,Y)\wedge r(X,Y)\},t_1\rangle$$
$$\langle\{q(X)\wedge r(X,Y)\},t_2\rangle\}$$

$$LiftedRC(Context, WeightedFormulas)$$
$$= LiftedRC(Context, WeightedFormulas\{X/c\})^n$$

...now we only have unary predicates

Observations and Queries

- Observations become the initial context.
  Observations can be ground or lifted.
-
  $$P(q|obs) = \frac{LiftedRC(q \wedge obs, WFs)}{LiftedRC(q \wedge obs, WFs) + LiftedRC(\neg q \wedge obs, WFs)}$$

  calls can share the cache
- "How many?" queries are also allowed

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.
- Otherwise, ground one individual from population, recurse. Sometimes this domain recursion is linear, but is typically exponential (as is grounding the population).

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.
- Otherwise, ground one individual from population, recurse. Sometimes this domain recursion is linear, but is typically exponential (as is grounding the population).

Always exponentially faster than grounding everything.

What we can and cannot lift

We can lift a model that consists just of

$$\langle \{f(X) \land g(Z)\}, \alpha_4 \rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

or just of

$$\langle\{f(X, Z) \wedge g(Y, Z)\}, \alpha_2\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

or just of

$$\langle\{f(X, Z) \wedge g(Y, Z)\}, \alpha_2\rangle$$

or just of

$$\langle\{f(X, Z) \wedge g(Y, Z) \wedge h(Y)\}, \alpha_3\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

or just of

$$\langle\{f(X,Z) \wedge g(Y,Z)\}, \alpha_2\rangle$$

or just of

$$\langle\{f(X,Z) \wedge g(Y,Z) \wedge h(Y)\}, \alpha_3\rangle$$

We cannot lift (still exponential) a model that consists just of:

$$\langle\{f(X,Z) \wedge g(Y,Z) \wedge h(Y,W)\}, \alpha_3\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle \{ f(X) \wedge g(Z) \}, \alpha_4 \rangle$$

or just of

$$\langle \{ f(X,Z) \wedge g(Y,Z) \}, \alpha_2 \rangle$$

or just of

$$\langle \{ f(X,Z) \wedge g(Y,Z) \wedge h(Y) \}, \alpha_3 \rangle$$

We cannot lift (still exponential) a model that consists just of:

$$\langle \{ f(X,Z) \wedge g(Y,Z) \wedge h(Y,W) \}, \alpha_3 \rangle$$

or

$$\langle \{ f(X,Z) \wedge g(Y,Z) \wedge h(Y,X) \}, \alpha_3 \rangle$$

Compilation

- The computation reduces to products and sums
- The structure can be determined at compile time
- Orders of magnitude faster than lifted recursive conditioning
- Often abstracted as weighted model counting (WMC)

Take Home

- Lifted inference exploits symmetries ("for all")
- Instead of considering which individuals a predicate is true for, count how many individuals it is true for, and determine appropriate probabilities.
- Always exponentially better in the number of undifferentiated individuals than grounding everything.
- Open problem: finding a dichotomy of those problems we know we can lift and those we know it is impossible to lift.

What is now required is to give the greatest possible development to mathematical logic, to allow to the full the importance of relations, and then to found upon this secure basis a new philosophical logic, which may hope to borrow some of the exactitude and certainty of its mathematical foundation. If this can be successfully accomplished, there is every reason to hope that the near future will be as great an epoch in pure philosophy as the immediate past has been in the principles of mathematics. Great triumphs inspire great hopes; and pure thought may achieve, within our generation, such results as will place our time, in this respect, on a level with the greatest age of Greece.

– Bertrand Russell 1917