clp(pfd(Y)) : Constraints for Probabilistic Reasoning in Logic Programming

Nicos Angelopoulos
nicos@cs.york.ac.uk

http://www.cs.york.ac.uk/˜nicos

Department of Computer Science

University of York

probabilistic finite domains

For discrete graphical models: extend the idea of finite domains to admit distributions.

from

$$X \ in \ \{a, b\} \qquad \text{(i.e. } X = a \quad \text{or} \quad X = b)$$

to

$$[p(X = a) + p(X = b)] = 1$$

clp(pfd(Y)) framework

For finite domain variable $V\ in\ \{v_1, \ldots, v_n\}$

and specific probabilistic inference algorithm Y, clp(pfd(Y)) computes

$$\psi_{\mathcal{S}}(V) = \{(v_1, \pi_1), (v_2, \pi_2), \ldots, (v_n, \pi_n)\}$$

clp(pfd(Y)) framework

$\mathcal{E}_i$ the probabilistic variables in E,
e vector, one element from each variable
$P_{\mathcal{S}}(\mathcal{E}_i = e_i) = \pi_i$
E/e predicate E, variables replaced by e.

$$P_{\mathcal{S}}(E) = P(E \mid \mathcal{P} \cup \mathcal{S}) = \sum_{\substack{\forall e \\ \mathcal{P}\cup\mathcal{S}\vdash E/e}} P_{\mathcal{S}}(E/e)$$

$$= \sum_{\substack{\forall e \\ \mathcal{P}\cup\mathcal{S}\vdash E/e}} \prod_i P_{\mathcal{S}}(\mathcal{E}_i = e_i)$$

Graphical Models integration

Execution, assembles the graphical model in the store according to program and query.

Existing algorithms can be used for probabilistic inference on the model present in the store.

Similarities in constraint propagation and probability propagation algorithms suggest interleaving algorithms maybe possible.

clp(pfd(Y)) example

For example, for program $\mathcal{P}_1$:

lucky(iv, hd). lucky(v, hd). lucky(vi, hd).

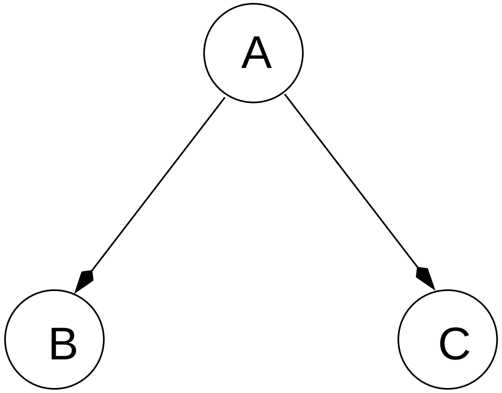
store $\mathcal{S}_1$ with variables D and C, with

$$\psi_{\mathcal{S}_1}(D) = \{(i, 1/6), (ii, 1/6), (iii, 1/6), (iv, 1/6), (v, 1/6), (vi, 1/6)\}$$

$$\psi_{\mathcal{S}_1}(C) = \{(hd, 1/2), (tl, 1/2)\}.$$

The probability of a lucky combination is $P_{\mathcal{S}_1}(lucky(D, C)) = 1/4$.

clp(pfd(bn)) example



	A = y	A = n
B = y	0.80	0.10
B = n	0.20	0.90

	A = y	A = n
C = y	0.60	0.90
C = n	0.40	0.10

clp(pfd(bn)) program

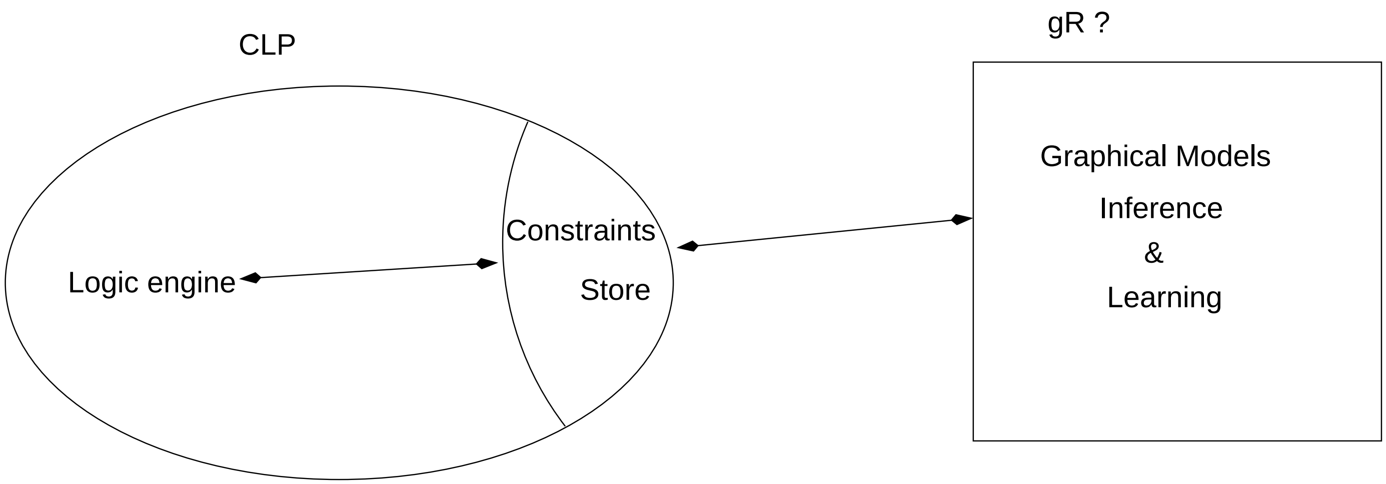
```
example_bn( A, B, C ) :-
        cpt(A,[],[y,n]),
        cpt(B,[A],[(y,y,0.8),(y,n,0.2),
                   (n,y,0.1),(n,n,0.9)]),
        cpt(C,[A],[(y,y,0.6),(y,n,0.4),
                   (n,y,0.9),(n,n,0.1)]).
```

clp(pfd(bn)) query

```
?- example_bn(X,Y,Z),
   evidence(X,[(y,0.8),(n,0.2)],
   Zy is p(Z = y).
```

Zy = 0.66

interactions



clp(pfd(c))

Probabilistic variables are declared with

$$V \sim \phi_V(Fd, Args)$$

Probability ascribing function ϕ_V and finite domain Fd are kept separately.

Variable example

$$Heat \sim finite\_geometric([l, m, h], [2])$$

finite geometric distribution with deterioration factor is 2.
In the absence of other information

$$\psi_\emptyset(Heat) = \{(l, 4/7)(m, 2/7), (h, 1/7)\}$$

clp(pfd(c)) conditionals

Conditional C

$$D_1 : \pi_1 \oplus \ldots \oplus D_m : \pi_m \mid Q$$

Each D_i is a predicate and all should share a single probabilistic variable V. Q is a predicate not containing V, and

$$0 \leq \pi_i \leq 1, \sum_i \pi_i = 1$$

V's distribution is altered as a result of C being added to the store.

clp(pfd(c)) subspaces

C partitions the space to weighted subspaces within which different events hold. Inference uses these partitions and the application of functions to compute updated probability distributions for the conditioned variables.

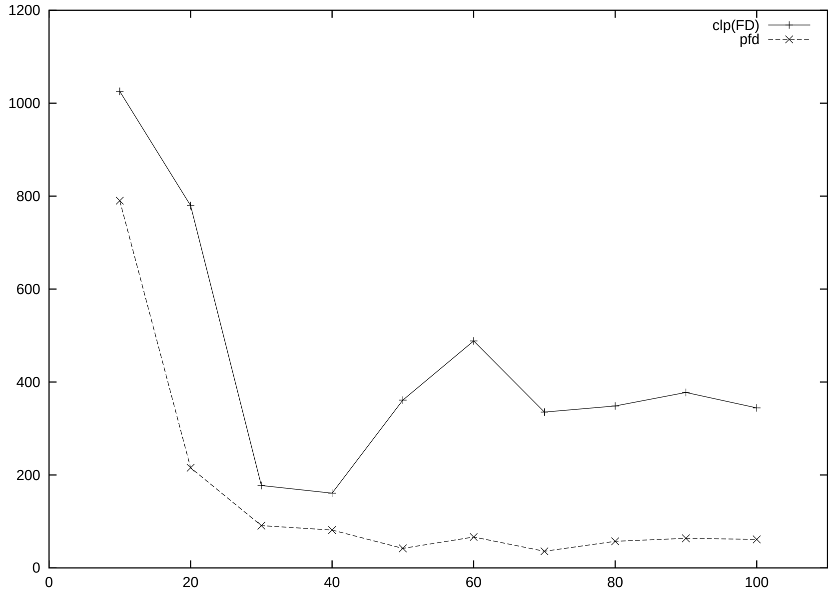
clp(pfd(Y)) Caesar's encodings

To illustrate benefits from the additional information in clp(pfd(Y)) when compared to clp(fd) we juxtapose performances of respective programs for a simple Caesar encoding scheme. The two programs are identical bar: (i) distribution over domains in clp(pfd(c)) based on the formula

$$\frac{\mid freq(E_i) - freq(D_i) \mid}{\sum_k \mid freq(E_i) - freq(D_k) \mid}$$

and (ii) labelling in clp(pfd(c)) uses a best-first algorithm.

clp(pfd(Y)) vs. clp(fd) time comparison



Run on SICStus 3.8

http://www.cs.york.ac.uk/˜nicos/sware/pfds

http://www.doc.ic.ac.uk/˜nicos/sware/pfds