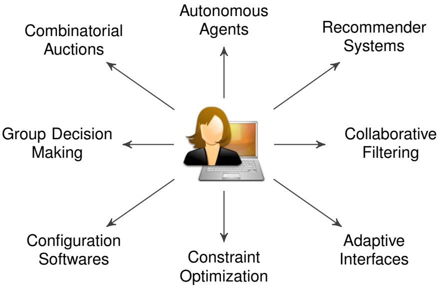
Relational Networks of Conditional Preferences

Frederic Koriche
Université Montpellier II - LIRMM, France

ILP Conference
31st July - 3rd August, 2011

Introduction



Preference Research: developing intelligent agents capable of tailoring their actions and recommendations to the preferences of human users

- Representation: expressing preferences in a compact and transparent form
- Reasoning: answering a broad range of queries
- Learning: predicting and extracting preferences

Related Work

		Representation Model	Reasoning Complexity	Learning Complexity
Directed models	Propositional	CP-nets (Boutilier et. al.,1999)	known	partially known
	Relational	unknown	unknown	unknown
Undirected models	Propositional	GAI-nets (Bacchus and Grove, 1995)	known	partially known
	Relational	GAIR-nets (Brafman, 2008)	unknown	unknown

Related Work

		Representation Model	Reasoning Complexity	Learning Complexity
Directed models	Propositional	CP-nets (Boutilier et. al.,1999)	known	partially known
	Relational	unknown	unknown	unknown
Undirected models	Propositional	GAI-nets (Bacchus and Grove, 1995)	known	partially known
	Relational	GAIR-nets (Brafman, 2008)	unknown	unknown

Challenge: Extending CP-nets to relational domains involving multiple, heterogeneous, and richly interconnected objects

CPR-Nets (Syntax)

Travel
travel id.
child number
adult number
price
duration
stop over

Airline
airline id.
services
incidents
environmental impact

Flight
flight id.
airline
from airport
to airport
day
from time
to time
class
seat

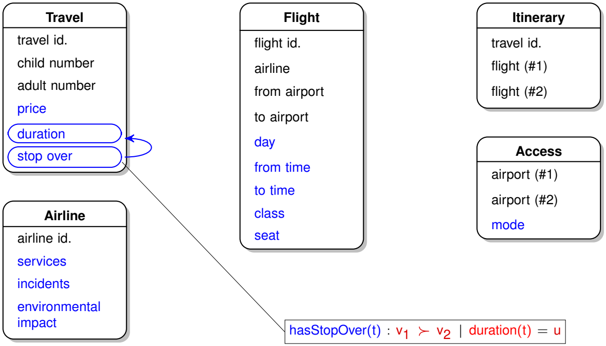
Itinerary
travel id.
flight (#1)
flight (#2)

Access
airport (#1)
airport (#2)
mode

Language:

- Relational schema: attributes, values, references, aggregators
- CP-clause: specifies the dependencies between an attribute and its parents
- CP-table: specifies conditional permutations of values
- CPR-net: assigns a CP-clause and a CP-table to each attribute

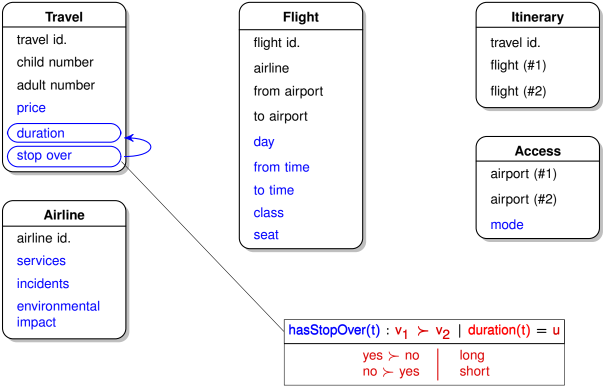
CPR-Nets (Syntax)



Language:

- Relational schema: attributes, values, references, aggregators
- CP-clause: specifies the dependencies between an attribute and its parents
- CP-table: specifies conditional permutations of values
- CPR-net: assigns a CP-clause and a CP-table to each attribute

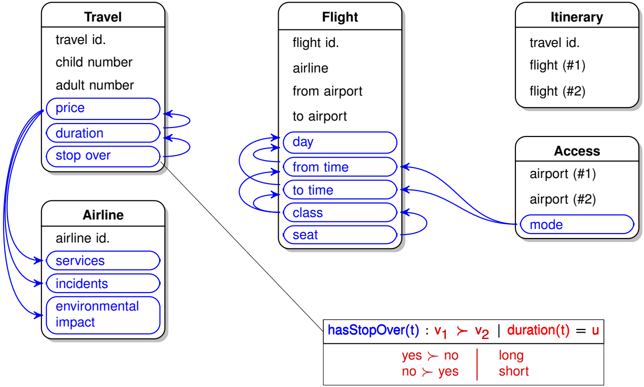
CPR-Nets (Syntax)



Language:

- Relational schema: attributes, values, references, aggregators
- CP-clause: specifies the dependencies between an attribute and its parents
- CP-table: specifies conditional permutations of values
- CPR-net: assigns a CP-clause and a CP-table to each attribute

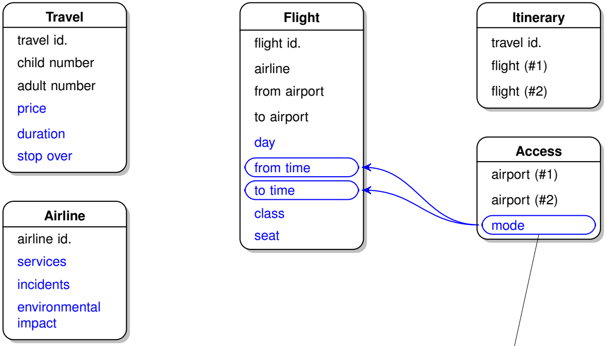
CPR-Nets (Syntax)



Language:

- Relational schema: attributes, values, references, aggregators
- CP-clause: specifies the dependencies between an attribute and its parents
- CP-table: specifies conditional permutations of values
- CPR-net: assigns a CP-clause and a CP-table to each attribute

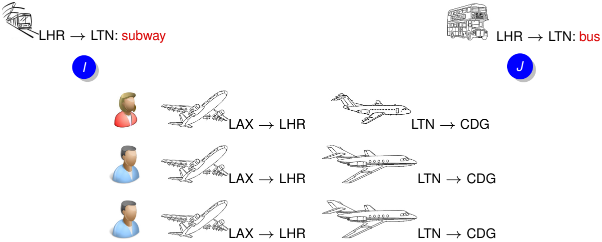
CPR-Nets (Syntax)



$\mathrm{mode}(a_1, a_2) : v_1 \succ v_2 \succ v_3 \mid \mathrm{itinerary}(t, f_1, f_2), \mathrm{toAirport}(f_1, a_1), \mathrm{fromAirport}(f_2, a_2), \mathrm{toTime}(f_1) = u_1, \mathrm{fromTime}(f_2) = u_2$

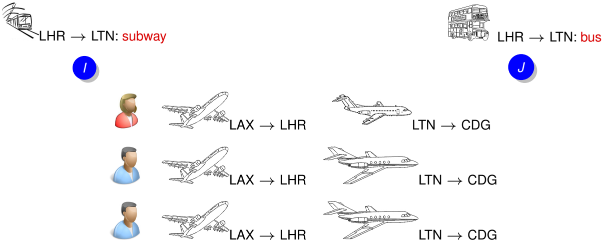
subway $\succ$ taxi $\succ$ bus	am	am
subway $\succ$ taxi $\succ$ bus	pm	pm
taxi $\succ$ bus $\succ$ subway	am	pm
bus $\succ$ taxi $\succ$ subway	pm	am

CPR-Nets (Semantics)



Flip: a pair (I, J) of interpretations that differ in only one ground attribute $a(o)$

CPR-Nets (Semantics)



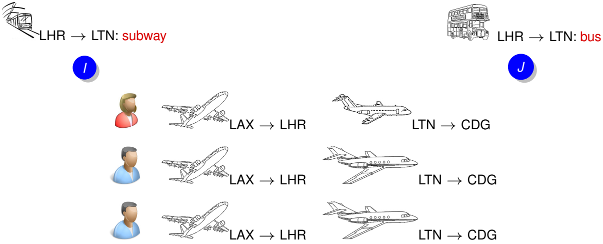
$[C_a(o)]_I$

travel	flight	to Airport	to Time	flight	from Airport	from Time
#t.1	#f.1	LHR	pm	#f.2	LTN	am
#t.2	#f.1	LHR	pm	#f.3	LTN	pm
#t.3	#f.1	LHR	pm	#f.3	LTN	pm

Conditioning:

- $[C_a(o)]_I$ is the set of all tuples v of values of $par(a)$ for which the body of $C_a(o, v)$ is true in I
- $\gamma_{par(a)}[C_a(o)]_I$ is the parent tuple of $a(o)$

CPR-Nets (Semantics)



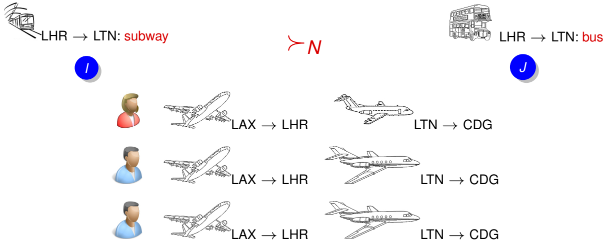
$\gamma_{par(a)}[C_a(o)]_I$

travel	flight	to Airport	to Time	flight	from Airport	from Time
#t.1	#f.1	LHR	pm	#f.2	LTN	am
#t.2	#f.1	LHR	pm	#f.3	LTN	pm
#t.3	#f.1	LHR	pm	#f.3	LTN	pm
			pm			pm

Conditioning:

- $[C_a(o)]_I$ is the set of all tuples v of values of $par(a)$ for which the body of $C_a(o, v)$ is true in I
- $\gamma_{par(a)}[C_a(o)]_I$ is the parent tuple of $a(o)$

CPR-Nets (Semantics)



$\gamma_{par(a)}[C_a(o)]_I$

travel	flight	to Airport	to Time	flight	from Airport	from Time
#t.1	#f.1	LHR	pm	#f.2	LTN	am
#t.2	#f.1	LHR	pm	#f.3	LTN	pm
#t.3	#f.1	LHR	pm	#f.3	LTN	pm
			pm			pm

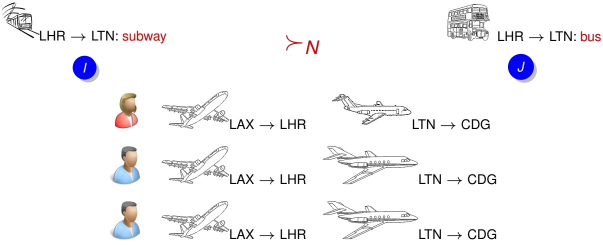
Conditioning:

- $[C_a(o)]_I$ is the set of all tuples v of values of $par(a)$ for which the body of $C_a(o, v)$ is true in I
- $\gamma_{par(a)}[C_a(o)]_I$ is the parent tuple of $a(o)$

Dominance:

- $I \succ_N J$ if the value of $a(o)$ specified by I is preferred to the one specified by J in the entry of $cpt(a)$ indexed by the parent tuple $\gamma_{par(a)}[C_a(o)]_I$

CPR-Nets (Semantics)



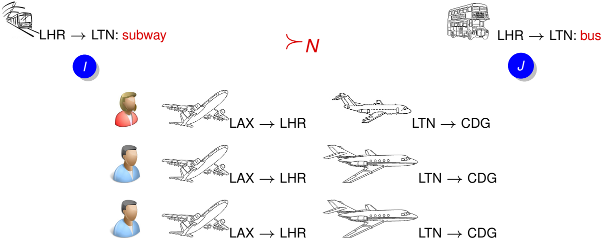
$\gamma_{par(a)}{}^{[C_a(o)]_I}$

travel	flight	to Airport	to Time	flight	from Airport	from Time
#t.1	#f.1	LHR	pm	#f.2	LTN	am
#t.2	#f.1	LHR	pm	#f.3	LTN	pm
#t.3	#f.1	LHR	pm	#f.3	LTN	pm
			pm			pm

Coherence:

- The transitive closure of $\succ_N$ must be a strict partial order

CPR-Nets (Semantics)



$\gamma_{par(a)}[C_a(o)]_I$

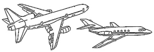
travel	flight	to Airport	to Time	flight	from Airport	from Time
#t.1	#f.1	LHR	pm	#f.2	LTN	am
#t.2	#f.1	LHR	pm	#f.3	LTN	pm
#t.3	#f.1	LHR	pm	#f.3	LTN	pm
			pm			pm

Coherence:

- The transitive closure of $\succ_N$ must be a strict partial order

Theorem 1: *Any acyclic CPR-net is coherent*

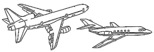
Reasoning



airport: LHR
airport: LTN
mode: *

airline: AA from airport: LAX to airport: LHR day: sun from time: am to time: am class: * seat: *	airline: BA from airport: LTN to airport: CDG day: mon from time: pm to time: pm class: * seat: *

$\longrightarrow$



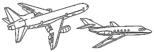
airport: LHR
airport: LTN
mode: bus

airline: AA from airport: LAX to airport: LHR day: sun from time: am to time: am class: 1 seat: win	airline: BA from airport: LTN to airport: CDG day: mon from time: pm to time: pm class: 2 seat: cor

Optimization:

- Input: $\mathcal{X}$ is a space of partial interpretations (allowing the value *)
- Output: $\mathcal{Y}$ is the space of all completions of elements in $\mathcal{X}$
- Problem: Given a CPR-net N and a partial interpretation x, find a completion y of x which is maximally preferred with respect to $\succ_N$

Reasoning

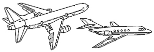


airport: LHR
airport: LTN
mode: *

airline: AA
from airport: LAX
to airport: LHR
day: sun
from time: am
to time: am
class: *
seat: *

airline: BA
from airport: LTN
to airport: CDG
day: mon
from time: pm
to time: pm
class: *
seat: *

$\longrightarrow$



airport: LHR
airport: LTN
mode: bus

airline: AA
from airport: LAX
to airport: LHR
day: sun
from time: am
to time: am
class: 1
seat: win

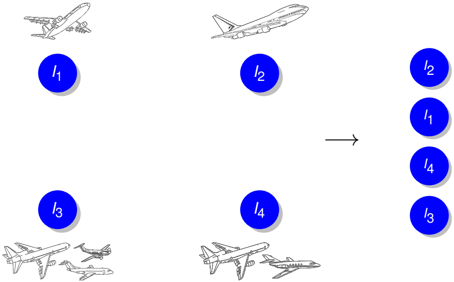
airline: BA
from airport: LTN
to airport: CDG
day: mon
from time: pm
to time: pm
class: 2
seat: cor

Optimization:

- Input: $\mathcal{X}$ is a space of partial interpretations (allowing the value *)
- Output: $\mathcal{Y}$ is the space of all completions of elements in $\mathcal{X}$
- Problem: Given a CPR-net N and a partial interpretation x, find a completion y of x which is maximally preferred with respect to $\succ_N$

Theorem 2: *For acyclic CPR-nets, optimization can be done in polynomial time*

Reasoning



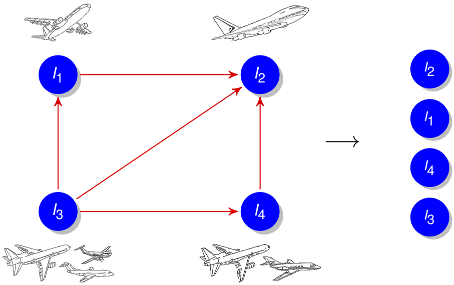
Ranking: an outcome set is a collection of interpretations defined over the same skeleton

- Input: $\mathcal{X}$ is a space of outcome sets of size m
- Output: $\mathcal{Y}$ is the symmetric group of all permutations over m elements
- Problem: Given a CPR-net N and an outcome set x, find a permutation y of x which is consistent with respect to $\succ_N$

Reasoning

$I_1 \gg_N I_3$ if for each ground attribute $a(o)$ with the same parent tuple in I_1 and I_3 the value v_1 is preferred to the value v_3

$\boxed{\succ_N \text{ implies } \gg_N}$

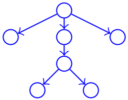


Ranking: an outcome set is a collection of interpretations defined over the same skeleton

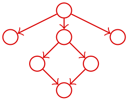
- Input: $\mathcal{X}$ is a space of outcome sets of size m
- Output: $\mathcal{Y}$ is the symmetric group of all permutations over m elements
- Problem: Given a CPR-net N and an outcome set x, find a permutation y of x which is consistent with respect to $\succ_N$

Theorem 3: *For acyclic CPR-nets, ranking can be done in polynomial time*

Learning



Hypothesis N_t

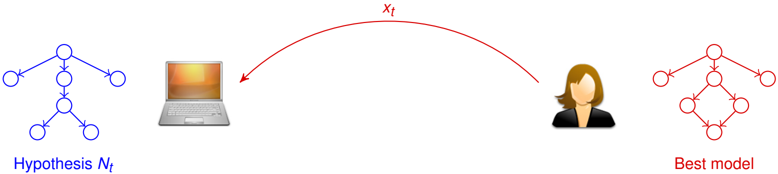


Best model

Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

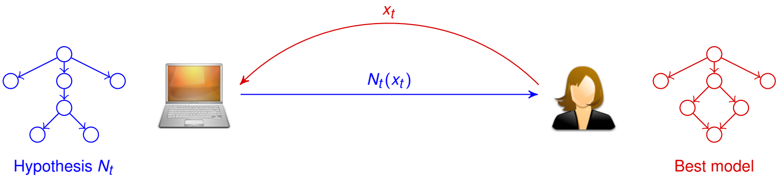
Learning



Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

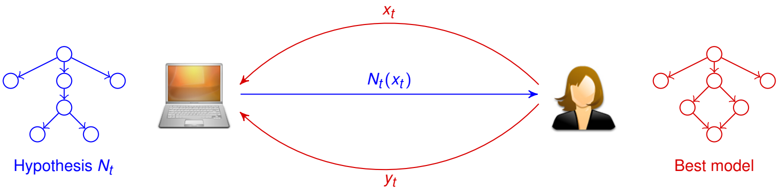
Learning



Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

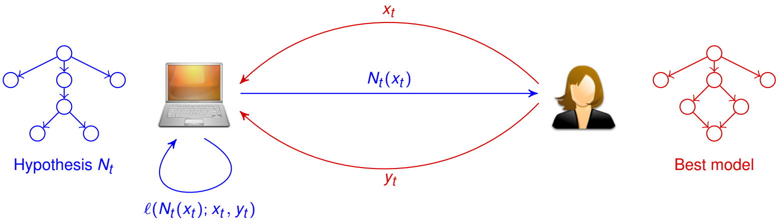
Learning



Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

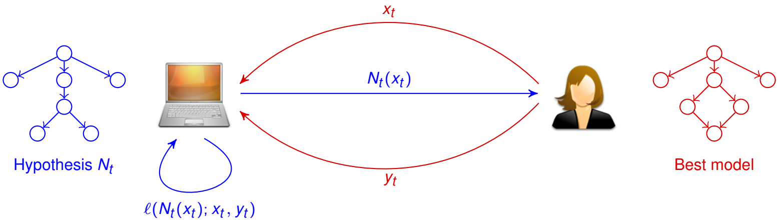
Learning



Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

Learning



Online Learning: the decision maker learns to be competent at a reasoning task by observing instances and feedbacks in a sequential manner. The performance of the algorithm is measured according to a loss function ℓ (bounded by an integer λ)

- Convergence criterion: the regret of the algorithm must be sublinear as a function of the number T of trials
- Complexity criterion: the computational cost of the algorithm must be polynomial in the parameters of the hypothesis class and the reasoning task

Learning

Linear Losses: any CPR-net N is viewed as the set of entries of its CP-table

permutation of the values of a

$$\langle \underbrace{C_a}_{\text{CP-clause for } a},\ \pi_a,\ \underbrace{v_{par(a)}}_{\text{parent tuple of } a} \rangle$$

hasStopOver(t) : $v_1 \succ v_2$	duration(t) = u
yes $\succ$ no	long
no $\succ$ yes	short

hasStopOver(t) : yes $\succ$ no | duration(t) = long

hasStopOver(t) : no $\succ$ yes | duration(t) = short

A loss function ℓ is linear if $\ell(N(x); x, y) = \sum_{e \in N} \ell(e(x); x, y)$

Tree CPR-nets: with constant clause length c and domain size d

Attributes	a
References	r
CP-clauses	$a \cdot ar^c$
Entries	$(a \cdot ar^c) \cdot d! \cdot d$

Learning

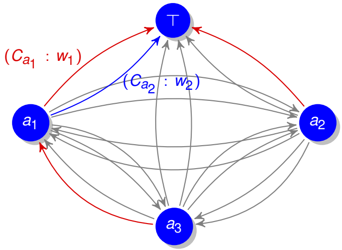
Initialization:

For each $e \in \mathcal{E}_{\text{tree}}$ set $L_1(e) = 0$

Trials: for $t = 1, 2, \ldots$

- Draw N_t according to $\mathbb{P}_t(N) \sim \exp\left[-\sum_{e \in N} L_t(e)\right]$
- Predict on instance (x_t, y_t) with N_t
- For each $e \in \mathcal{E}_{\text{tree}}$ set
  $L_{t+1}(e) = L_t(e) + \eta_t \ell(e(x_t), y_t)$

Expanded Hedge



Weighted dependency graph
of all candidate clauses

Tree CPR-nets:

- The regret of the Expanded Hedge algorithm is

$$\lambda \sqrt{\frac{\ln |\mathcal{N}_{\text{tree}}|}{T}} \text{ where } |\mathcal{N}_{\text{tree}}| \leq (a+1)^{a-1} a^{a^2 r^c} (d!)^d$$

- Using the Matrix-Tree Theorem, the cost of generating a directed random spanning tree at random is polynomial in the number of candidate CP-clauses

Learning

Learning to Optimize: Let $\mathcal{X}$ be a space of partial interpretations, $\mathcal{Y}$ the corresponding space of total interpretations and ℓ_{opt} be the loss function defined as follows:

$$\ell_{opt}(e(x); x, y) = \begin{cases} 1 & \text{if } y \text{ is a suboptimal choice for } e \text{ on } x \\ 0 & \text{otherwise} \end{cases}$$

Theorem 4: *Tree CPR-nets (with constant clause length and domain size) are efficiently learnable from optimization tasks using* ℓ_{opt}

Learning to rank: Let $\mathcal{X}$ be a space of outcome sets of size m, $\mathcal{Y}$ be the space of permutations over m elements, and ℓ_{rank} be the loss function defined as follows:

$$\ell_{\mathrm{rank}}(e(x); x, y) = \begin{cases} 1 & \text{if either } y = (I_1, I_2) \text{ and } I_2 \gg_e I_1 \text{, or } y = (I_2, I_1) \text{ and } I_1 \gg_e I_2 \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5: *Tree CPR-nets (with constant clause length and domain size) are efficiently learnable from ranking tasks using* ℓ_{rank}

Learning

Learning to Optimize: Let $\mathcal{X}$ be a space of partial interpretations, $\mathcal{Y}$ the corresponding space of total interpretations and ℓ_{opt} be the loss function defined as follows:

$$\ell_{opt}(e(x); x, y) = \begin{cases} 1 & \text{if } y \text{ is a suboptimal choice for } e \text{ on } x \\ 0 & \text{otherwise} \end{cases}$$

Theorem 4: *Tree CPR-nets (with constant clause length and domain size) are efficiently learnable from optimization tasks using* ℓ_{opt}

Learning to rank: Let $\mathcal{X}$ be a space of outcome sets of size m, $\mathcal{Y}$ be the space of permutations over m elements, and ℓ_{rank} be the loss function defined as follows:

$$\ell_{\mathrm{rank}}(e(x); x, y) = \begin{cases} 1 & \text{if either } y = (I_1, I_2) \text{ and } I_2 \gg_e I_1 \text{, or } y = (I_2, I_1) \text{ and } I_1 \gg_e I_2 \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5: *Tree CPR-nets (with constant clause length and domain size) are efficiently learnable from ranking tasks using* ℓ_{rank}

Conclusions

Summary: The family of CPR-nets

- Representation: CPR-nets maintain the spirit of CP-nets by representing relational preferences in a compact and a transparent form
- Reasoning: acyclic CPR-nets (of constant in-degree) support tractable inference for both optimization and ranking tasks
- Learning: tree CPR-nets (of constant clause length and domain size) are efficiently online learnable from both optimization and ranking tasks

Ongoing Research:

- Comparing relational preference models: CPR-nets versus GAIR-nets on optimization and ranking problems (flight and movie recommenders)
- Improving the learning algorithm: (Hedge versus Following the Perturbed Leader) and spanning tree generation algorithms (Determinant-based algorithms vs. Markov chains)
- Investigating the issue of cyclic CPR-nets: important applications in social networks