Toward Inductive Logic Programming for Collaborative Problem Solving

Jian Huang & Adrian R. Pearce

NICTA Victoria Research Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne

International Conference on Intelligent Agent Technology
December 20, 2006

Motivation

Learning in multi-agent systems:

- Deploying single learners in multi-agent environments;
- Increasing advocates on interactive learning;

- Learning in isolation makes sharing knowledge a problem;
- Learning through interaction overcomes the problem of knowledge being distributed but interaction is costly.

Our work focus, specifically, on ILP as the approach to learning.

Contribution

Our work on collaborative ILP (C-ILP) contributes in:

- Defining the problem of C-ILP;
- Integrating ILP and epistemic reasoning;
- Learn through interaction;
- Learning by collaboration;

- Achieved with low communication effort.

THE UNIVERSITY OF MELBOURNE

Outline

Inductive Logic Programming

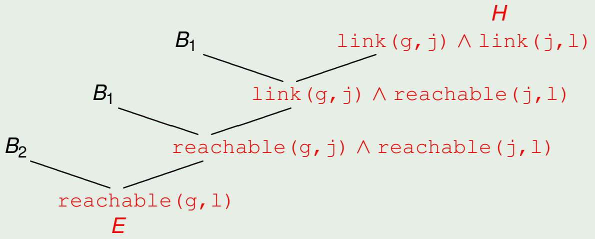
An Illustrative Example

Example

B_1 : `link(A,B)` $\longrightarrow$ `reachable(A,B)`

B_2 : `reachable(A,B)` $\wedge$ `reachable(B,C)` $\longrightarrow$ `reachable(A,C)`

E : `reachable(g,l)`

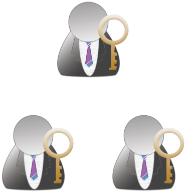


H : `link(g,j)` $\wedge$ `link(j,l)`

Collaborative ILP

Extends ILP [Muggleton, 1999] allowing collaboration:

- $\mathbb{B} = \bigcup_{i \in \mathbb{A}} B_i$: total background knowledge
- $\mathbb{E}$: total training examples
- H: hypothesis
- C-ILP: construct H such that $\mathbb{B} \wedge H \models \mathbb{E}$



Collaborative ILP

Formally,

Definition (Collaborative ILP)

1. Necessity: $\mathbb{B} \not\models \mathbb{E}^+$
2. Sufficiency: $\mathbb{B} \wedge H \models \mathbb{E}^+$
3. Weak Consistency: $\mathbb{B} \wedge H \not\models \Box$
4. Strong Consistency: $\mathbb{B} \wedge H \wedge \mathbb{E}^- \not\models \Box$, and
5. $\neg\exists i \in \mathbb{A}$ such that $\mathbb{B}_i \wedge H \models \mathbb{E}^+$

Outline

Inductive Agent Constructs

Each agent consists of:

1. Background Set $\mathbb{B} = \{b_1, \cdots, b_n\}$
2. Example Set $\mathbb{E} = E^+ \cup E^-$, e.g. $e_1^+ = sort([2,1,3],[1,2,3])$
3. Capability Set $\mathbb{C} = C^+ \cup C^-$, e.g. $c_1^+ = A_i sort$
4. Knowledge Base $\mathbb{K} = K^B \cup K^E \cup K^C$, e.g. $k_1 = K_i(A_i sort)$

Integrates ILP and reasoning, through communicating training examples [Huang & Pearce, 2006].

Outline

1. Overview
2. Defining Collaborative ILP
3. C-ILP Model
4. Distributed Path Planning using C-ILP
5. Experiment Results

Distributed path planning using C-ILP

Problem setting

Each agent has the following background knowledge:

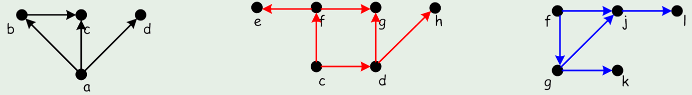
Example (background knowledge)

B_1 : `link(A,B)` $\longrightarrow$ `reachable(A,B)`

B_2 : `reachable(A,B)` $\wedge$ `reachable(B,C)` $\longrightarrow$ `reachable(A,C)`

Plus, each agent knows the links it has traveled, e.g.:

Example (background knowledge)

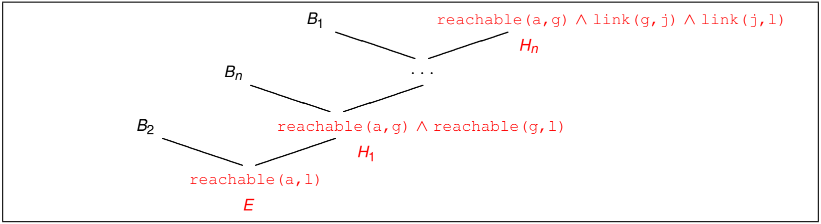


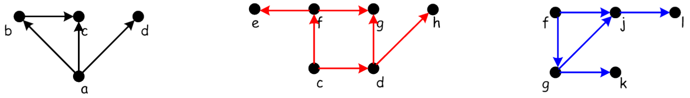
Car A Car B Car C

Distributed path planning using C-ILP

Inducing a path

A path can be induced:



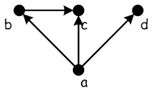


Car A Car B Car C

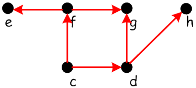
Distributed path planning using C-ILP

Collaboration

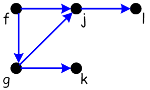
$$
\begin{array}{llrcl}
1 & \text{A}: & E & = & reachable(a, l) \\
2 & \text{A ASKS C}: & E & = & reachable(a, l) \\
3 & \text{C INDUCES}: & H & = & reachable(a, g) \wedge link(g, j) \wedge link(j, l) \\
4 & \text{C REPLIES}: & H & = & reachable(a, g) \\
5 & \text{A DEDUCES}: & K_1 & = & K_a(K_c(reachable(g, l))) \\
 & & K_2 & = & K_a(\exists i K_i(reachable(a, g)) \rightarrow K_a(reachable(a, l))) \\
6 & \text{A}: & E & = & reachable(a, g)
\end{array}
$$



Car A



Car B

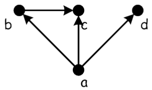


Car C

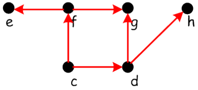
Distributed path planning using C-ILP

Collaboration

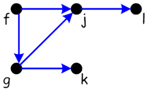
6	A :	$E = reachable(a, g)$
7	A ASKS B :	$E = reachable(a, g)$
8	B INDUCES:	$H = reachable(a, c) \wedge link(c, d) \wedge link(d, g)$
9	B REPLIES:	$H = reachable(a, c)$
10	A DEDUCES:	$K_3 = K_a(K_b(reachable(c, g)))$
		$K_4 = K_a(\exists i K_i(reachable(a, c)) \rightarrow K_a(reachable(a, g)))$
11	A :	$E = reachable(a, c)$
12	A INDUCES:	$H = link(a, c)$



Car A

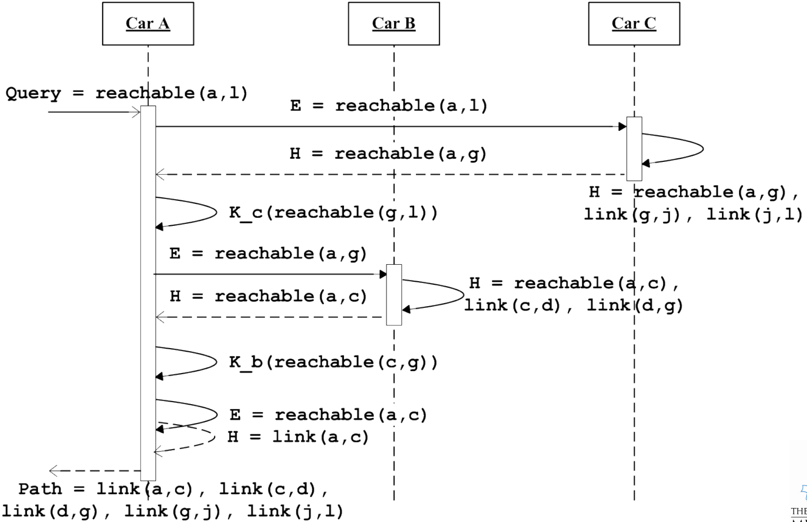


Car B



Car C

Distributed path planning using C-ILP
Collaboration



Outline

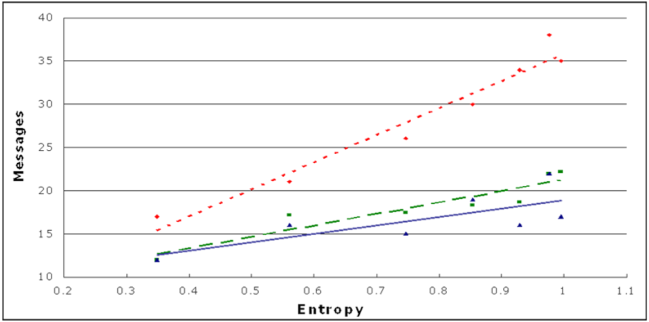
THE UNIVERSITY OF MELBOURNE

Experiments

Experiments to investigate communication cost:

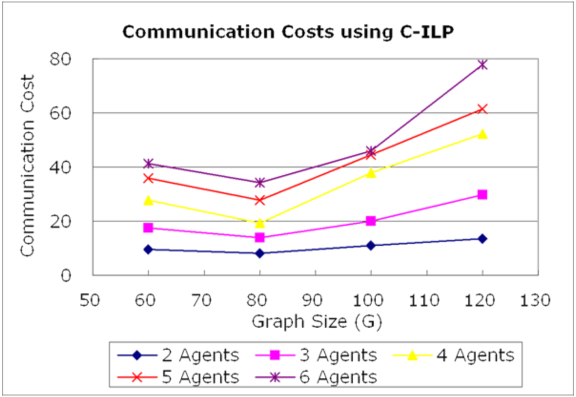
- compare C-ILP approach against a centralized approach
- different numbers of agents, A, from 2 to 6
- varying graph sizes, G, from 60 to 120
- 100 trials for every value of A and G

Results



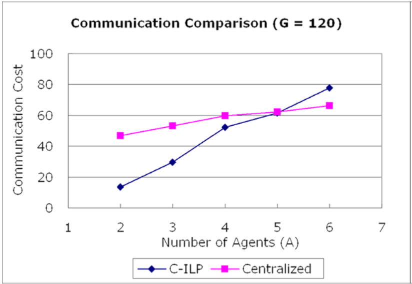
1. Communication increases as knowledge is more evenly distributed.
2. Communication increase much slower with C-ILP approach.

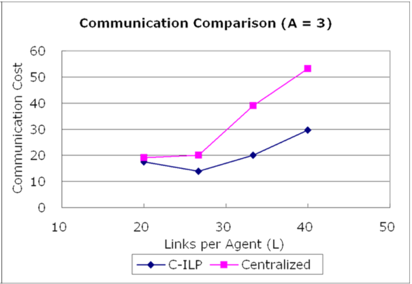
Results



3. Communication increases as number of agents increases.
4. Communication increases as graph size increases.

Results





5. Communication lower with C-ILP in general, except when each agent knows only very few links.
6. Communication lower with C-ILP when each agent knows 30 or more links.

Conclusion

1. Define Collaborative ILP (C-ILP)
2. Describe C-ILP model which:
   - integrate ILP and epistemic reasoning;
   - allows interaction and collaboration among agents.
3. Applications in:
   - learning missing program fragments;
   - distributed path planning.
4. Results show promise for:
   - overcoming the problem of knowledge being distributed;
   - avoiding central control;
   - saving communication cost.