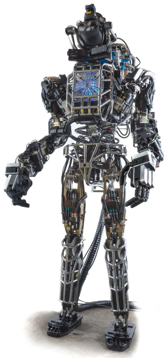
Inferring the Structure of Probabilistic Graphical Models for Efficient Natural Language Understanding

Istvan Chung Oron Propp
Mentor: Dr. Thomas Howard[1]

[1]Robust Robotics Group
MIT CSAIL

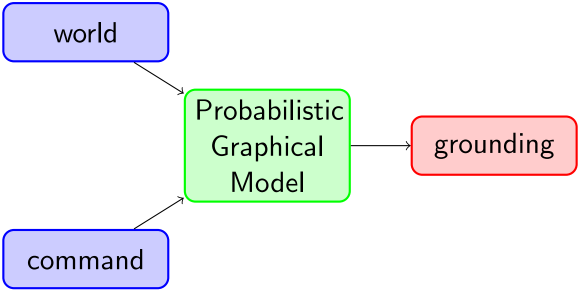
Fourth Annual MIT PRIMES Conference
18 May 2014

Introduction

- Existing interfaces for controlling robots are specialized and difficult to use
- It would be much easier to control robots using natural language commands
- Existing natural language interfaces do not scale well with the complexity of the environment

Probabilistic Graphical Models



Example

{`WorldObject(0, ‘robot’), WorldObject(1, ‘crate’), WorldObject(2, ‘box’)`} + “approach the box” →
`Constraint(WorldObject(0), WorldObject(2), ‘near’)`

Grammar

- It doesn't make sense to view the input as a monolithic block of text
- It is more meaningful to understand the input with its grammatical structure
- A grammar is used to assign meaning to the words

$$
\begin{array}{lcl}
\text{VP} & \rightarrow & \text{VB NP} \\
\text{VP} & \rightarrow & \text{VB NP PP} \\
\text{VP} & \rightarrow & \text{VB PP} \\
\text{NP} & \rightarrow & \text{DT NN} \\
\text{NP} & \rightarrow & \text{NP PP} \\
\text{PP} & \rightarrow & \text{IN NP} \\
\text{VB} & \rightarrow & \text{``approach'', ``land'', ``fly''} \\
\text{DT} & \rightarrow & \text{``a'', ``the''} \\
\text{NN} & \rightarrow & \text{``box'', ``chair'', ``table''} \\
\text{IN} & \rightarrow & \text{``near'', ``far'', ``to''}
\end{array}
$$

Parse Tree

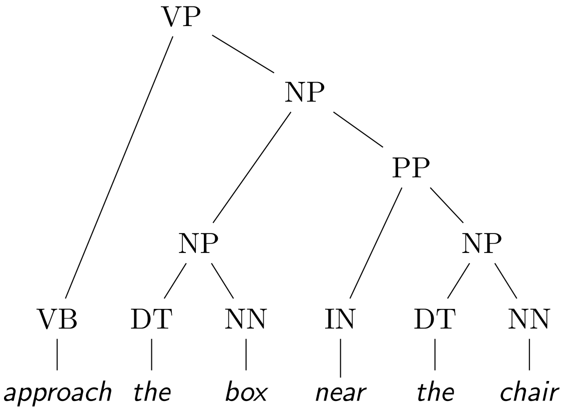


Figure: Parse tree for “approach the box near the chair”

Parse Ambiguity

- Some sentences are ambiguous

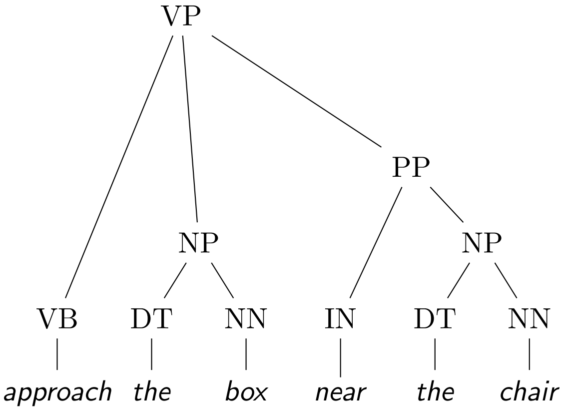


Figure: Alternate parse tree for "approach the box near the chair"

CYK Chart Parser

- The CYK Parsing algorithm [4, 5, 6] accomplishes this task in $O(n^3)$ time.
- All possible parses of an ambiguous sentence are returned

VP					
	NP/X0				
			PP		
	NP			NP	
VB	DT	NN	IN	DT	NN
approach	the	box	near	the	chair

Generalized Grounding Graph

- Comprised of "factors" which relate groundings, correspondences, and phrases, and are represented by log-linear models
- Grounding each phrase depends on the groundings of the child phrases

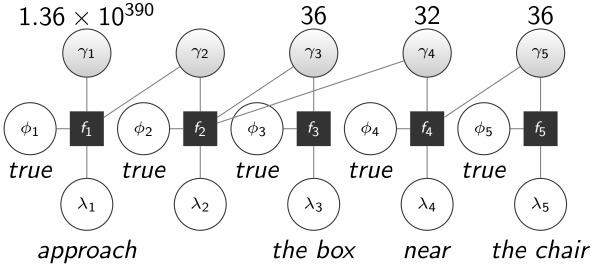


Figure: Generalized grounding graph for "approach the box near the chair"

[2] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, *Approaching the Symbol Grounding Problem with Probabilistic Graphical Models*. 2013.

Log-linear Model

- Log-linear models [1] are used to assign a score to a grounding given some input.
- This is done using a set of features
- Features evaluate aspects of the input and grounding

Scoring function

$$p(c \mid x, y; \mathbf{v}) = \frac{\exp(\mathbf{v} \cdot \mathbf{f}(x, y, c))}{\sum_{c' \in \mathcal{C}} \exp(\mathbf{v} \cdot \mathbf{f}(x, y, c'))}$$

Where x is the input, y is the grounding, c is a correspondence variable, $\mathbf{f}$ is the array of features, and $\mathbf{v}$ is the array of feature weights.

[1] M. Collins, *Log-Linear Models*.
`http://www.cs.columbia.edu/~mcollins/loglinear.pdf`

Log-linear Model – Training

- Feature weights **v** are trained according to data from a corpus of examples.
- The aim of training is to maximize the objective function:

Objective function and gradient

$$L'(\mathbf{v}) = \sum_i \log p(c_i \mid x_i, y_i; \mathbf{v}) - \frac{\lambda}{2} \sum_k {v_k}^2$$

$$(\nabla L')(\mathbf{v})_k = \sum_i f_k(x_i, y_i, c_i) - \sum_i \sum_{c \in \mathcal{C}} p(c \mid x_i, y_i; \mathbf{v}) f_k(x_i, y_i, c) - \lambda v_k$$

- The LBFGS optimization method [7] efficiently maximizes L' while consuming little space.

[7] Byrd, R. H., Lu, P., Nocedal, J., Zhu, C. *A Limited Memory Algorithm for Bound Constrained Optimization*. 1995.

The Problem

- Number of possible individual groundings is $O(n^2)$ in the number of objects
- Adding in sets of groundings makes it $2^{O(n^2)}$

The Problem

With 17 objects and 8 relations, the number of sets of constraints is

$$2^{8\times(17+8\times17)^2} = \boxed{3.08 \times 10^{56374}}$$

Partitioning Grounding Spaces

- In many situations, most groundings are irrelevant
- Partition the grounding space to eliminate irrelevant objects from consideration

Rules

- Aim of rules is to partition grounding spaces to only include pertinent groundings

Example

World: `WorldObject(0, ‘robot’)`, `WorldObject(1, ‘crate’)`, `WorldObject(2, ‘box’)`
“approach the box” $\rightarrow$ {`Rule(‘box’)`, `Rule(‘robot’)`}

- Effectiveness of rules increases with complexity of environment and grounding spaces

Hierarchical Grounding Graph

- Run inference on space of rules
- Apply result to grounding spaces in grounding graph model
- Run inference in graphical model on partitioned grounding spaces for efficient grounding

Hierarchical Grounding Graph

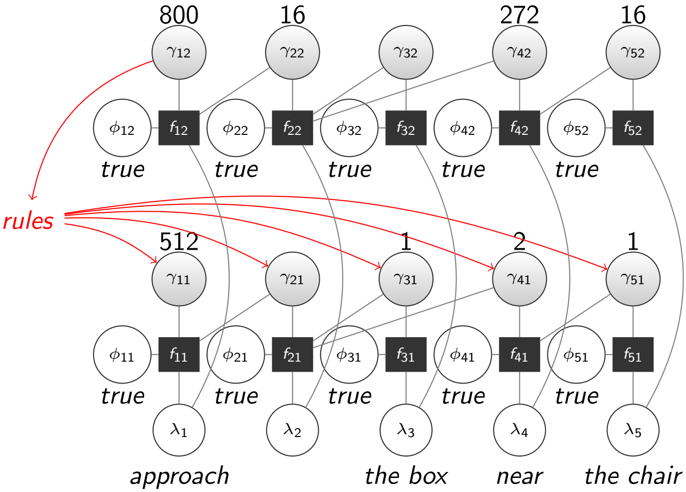
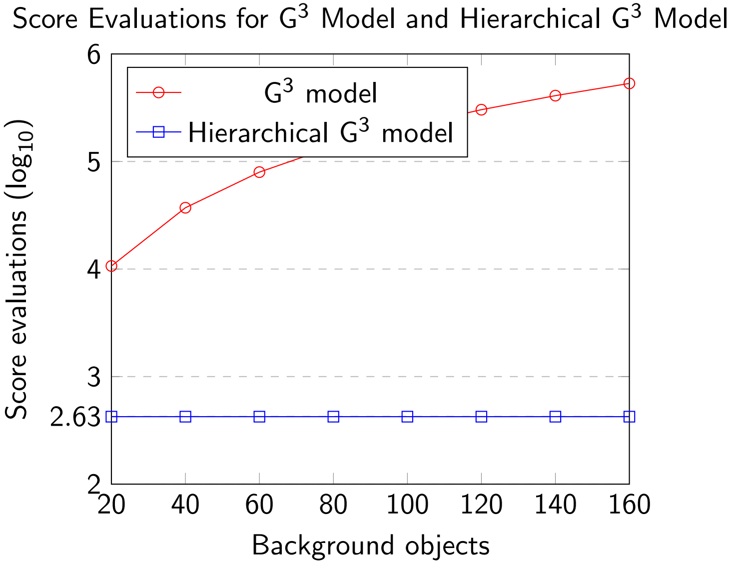
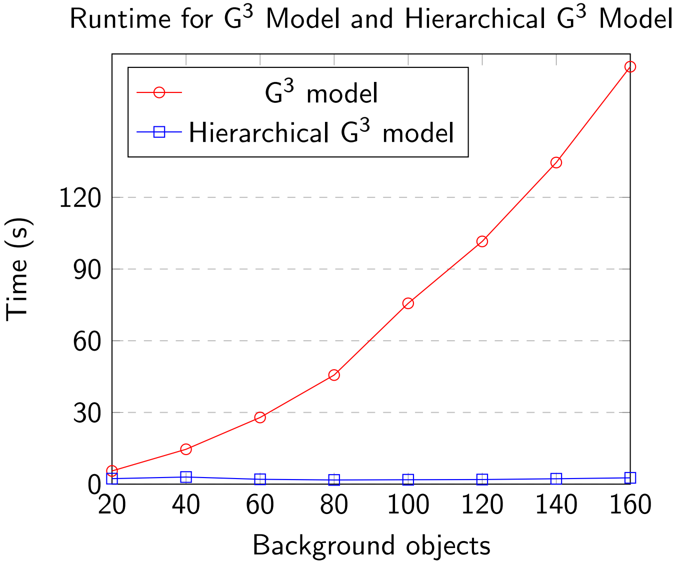


Figure: Hierarchical Grounding Graph for “approach the box near the chair”

Score Evaluations



Run-time



Runtime for G3 Model and Hierarchical G3 Model
G3 model
Hierarchical G3 model
Time (s)
0
30
60
90
120
20
40
60
80
100
120
140
160
Background objects

Holodeck Experiment

Future Work

- Expand space of rules to handle region and constraint types
- Implement spatial features with regards to physical world model
- Improve optimization routine (current runtime is impractical)
- Test on Distributed Correspondence Graph model [3]
- Handle parse ambiguity
- Support more sophisticated sentence structures
- Rigorous testing in more complex environments
- Compute bounds on the efficiency of the algorithm

[3] T.M. Howard, S. Tellex, and N. Roy, *A Natural Language Planner Interface for Mobile Manipulators*, to appear in the Proceedings of the 2014 International Conference on Robotics and Automation. 2014.

Acknowledgements

Thank you to

- MIT PRIMES
- Dr. Thomas Howard
- Professor Nicholas Roy
- Dr. Marec Doniec
- Our parents

Bibliography I

[1] M. Collins, *Log-Linear Models*. http://www.cs.columbia.edu/~mcollins/loglinear.pdf

[2] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, *Approaching the Symbol Grounding Problem with Probabilistic Graphical Models*. 2013.

[3] T.M. Howard, S. Tellex, and N. Roy, *A Natural Language Planner Interface for Mobile Manipulators*, to appear in the Proceedings of the 2014 International Conference on Robotics and Automation. 2014.

[4] J. Cocke, and J. Schwartz, *Programming languages and their compilers: Preliminary notes*. 1970.

[5] D. Younger, *Recognition and parsing of context-free languages in time* n^3. 1967.

[6] T. Kasami, *An efficient recognition and syntax-analysis algorithm for context-free languages*. 1965.

Bibliography II

[7] Byrd, R. H., Lu, P., Nocedal, J., Zhu, C. *A Limited Memory Algorithm for Bound Constrained Optimization*. 1995.