Lifted reasoning (for RL and beyond)

Geoff Gordon
ggordon@cs.cmu.edu

Summary of (part of) Martijn's talk

- There are lots of lifted languages
- No general consensus on scope or semantics
- Consequent arguments about "my language can represent this and yours can't"
- Consequent proliferation of incomparable reasoning algorithms (different requirements, capabilities)

Goal

- A (class of) representation language(s) supporting many different reasoning styles
- Consequence: can apply different styles to parts of same problem (and so solve problems that won't fall to any single style)
- Consequence: can represent problems that are impossible to solve efficiently
- Subgoal: don't make it too easy to do so

Efficiency

- Claim: efficiently solving all representable problems is not a fair requirement
- Proposal: want a reasoning step which is efficient and makes non-negligible progress
  - e.g., policy improvement
- If we ran it forever, we'd get optimal answer
  - nontrivial guarantee! (see: Gödel)
- Easy problem class: good answer soon

Learning

- General learning problem is information-theoretically difficult
  - e.g., even learning an uncontrolled HMM can't be done w/ poly data and time (no matter the representation or algorithm)
- But, a general language lets us flexibly specify tractable learning problems
- And, can ask for efficient progress r.t. solution, or solutions to easy problems

Styles: search

- E.g., DPLL (SAT, propositional entailment)
- E.g., Graphplan (propositional STRIPS)
- E.g., RRTs (deterministic satisficing control)
- E.g., Deep Blue (chess)

Styles: optimization

- E.g., MDP (Bellman LP)
- E.g., resource allocation (divisible: LP; indivisible: matching ... general ILP)
- E.g., scheduling (ILP)
- E.g., stochastic program w/ recourse
  - e.g., facility location problem

Style: Bayes rule

- E.g., discrete belief propagation
- E.g., {extended, unscented} Kalman filters
- E.g., exact filtering in HMMs or DBNs
- E.g., particle filters
- E.g., expectation propagation

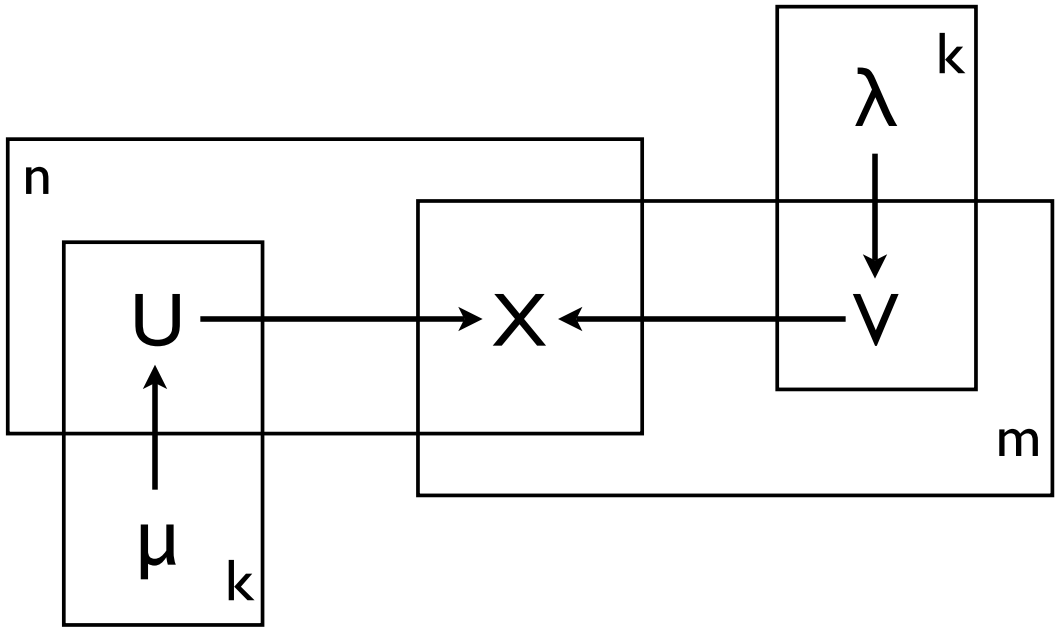
Style: dynamic programming

- E.g., MDPs (value iteration)
- E.g., HMMs (forward-backward, Viterbi)

Style: lifting

- World is composed of objects with properties and relationships
- Reason at level of classes of objects (all people, all relatives of Fred, all advisors of students who are taking my course)
- E.g., FOL
- E.g., natural language

E.g., plate models



this diagram represents a probabilistic, hierarchical principal components analyzer

"Weak" lifting

- Simplest lifted semantics: "for loop" or "macro" to generate structure
  - e.g., plates, MLNs, AMPL
- Can't handle uncertainty about existence or identity of objects: no connection among different-size domains
  - although sometimes decent approx

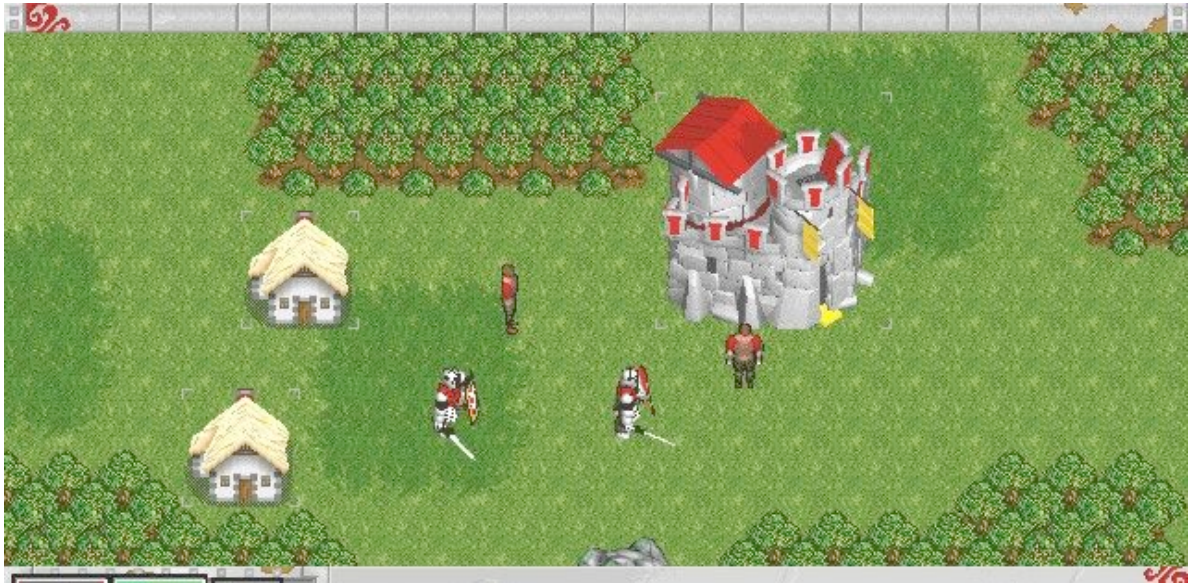
Lifting as glue

- Lifting is one candidate for the "glue" that holds together multiple levels of detail or multiple styles of reasoning
- Expand different parts of model for different tasks
- Expose detail, or expose special structure

Example: robot navigation

- Low-level vision
  - spectral graph cuts to get superpixels, texture operators to get features, MRF to get objects and relations
- Motor control
  - SNOPT + DP + Russ's magic
- Planning room to room (or to Barbados)
  - search (but, not deterministic)

Example: Warcraft



- Guestrin, 2003
- learn policy in simple world w/ few objects
- generalize to bigger world

David's "Space Invaders"

- Low-level vision
- Belief tracking
- Planning

Example: negotiation

- Learn about people: preferences, trustworthiness, prejudices, negotiation styles
- Learn about world: "pies" to be divided, ways to "extend the pie"
- Plan to achieve the best possible agreement

Efficiency, again

- Search == NP
- Optimization == NP-opt
- Bayes rule == #P
- (Search or optimization) + Bayes rule == PSPACE (e.g., POMDP)
- search + lifting == r.e. (!!!) (e.g., FOL)
- all together == ??? (hopefully still r.e.)

Reminder

- The full problem is r.e. == **intractable**
- We are asking for a **general** reasoning algorithm which is **efficient** in special cases, and which makes **progress** in all cases

Our contribution

- Language unifying FOL, MILP
- First-order programs (FOP)
- General* sound & complete reasoning method
  - specializes to DPLL for SAT
  - specializes to branch & cut for MILPs

} known efficient

  - specializes to FOL theorem proving

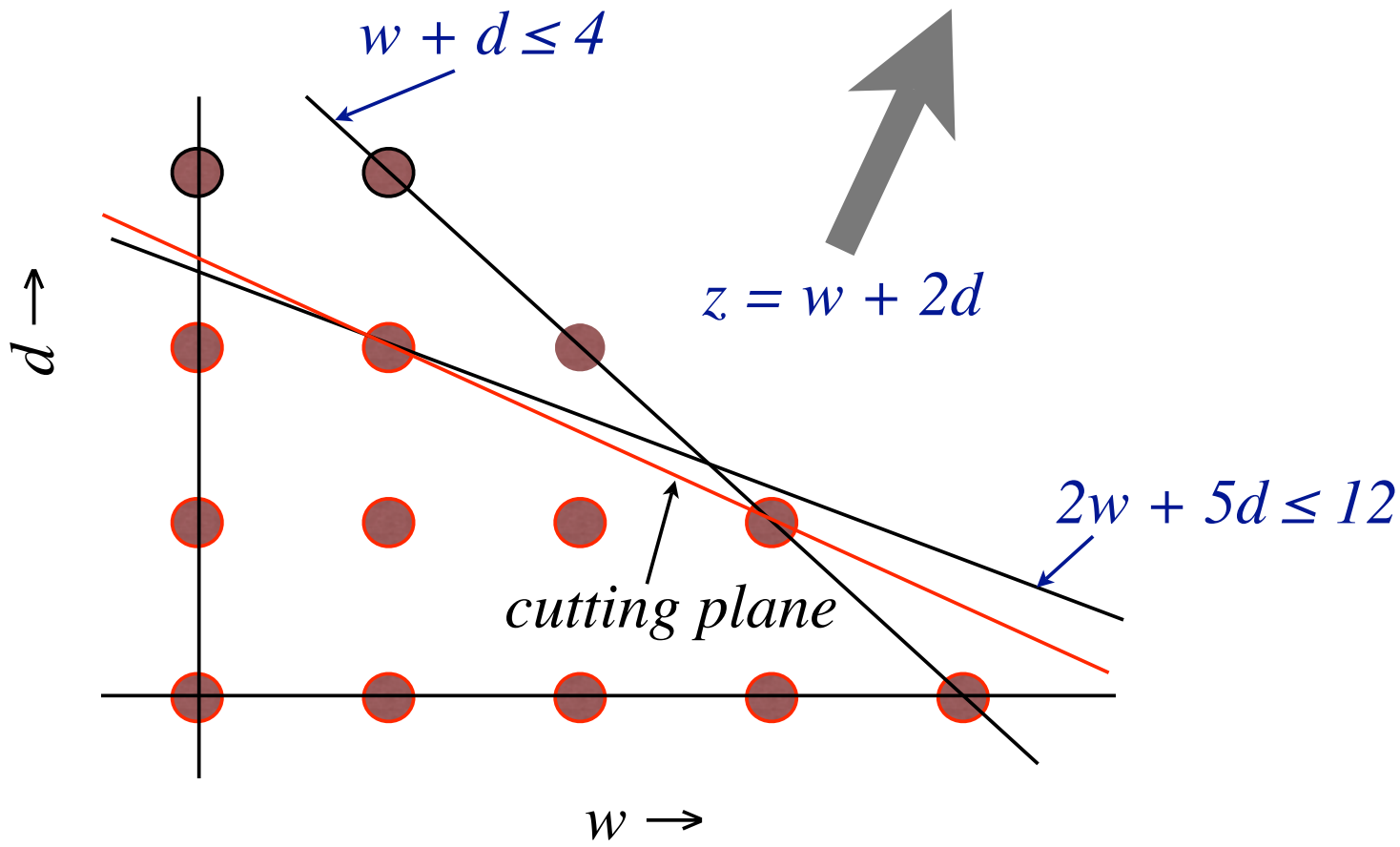
Expressiveness

- Any FOL KB and any MILP can be translated into FOP with no increase in size
- There are (many!) FOP KBs which cannot be translated efficiently into FOL or MILP

Probability

- MILPs already handle some probabilistic reasoning
- Limited add'l extensions for probability
- Others have designed reasoning algorithms which specialize to BP

MILP example



MILP in FOP

$$4 - w - d \quad \wedge$$
$$12 - 2w - 5d$$
$$w \in \{0, 1, 2, 3, 4\}$$
$$d \in \{0, 1, 2, 3, 4\}$$

- Step 1: feasibility
- This FOP sentence has value ≥ 0 iff w+d ≤ 4 and 2w + 5d ≤ 12

MILP in FOP

- Step 2: optimality
- Just binary search on objective value
- Turns objective into another constraint

FOL example

$bird(x) \Rightarrow flies(x)$

$eagle(x) \Rightarrow bird(x)$

$eagle(x) \Rightarrow eagle(father(x))$

$eagle(Stanley)$

? $flies(father(Stanley))$

Eagles in FOP

$$\bigwedge x.\ (\text{flies}(x) - \text{bird}(x)) \quad \wedge$$

$$\bigwedge x.\ (\text{bird}(x) - \text{eagle}(x)) \quad \wedge$$

$$\bigwedge x.\ (\text{eagle}(\text{father}(x)) - \text{eagle}(x)) \quad \wedge$$

$$\text{eagle}(\text{Stanley}) - 1$$

$$\text{flies} \in \{0, 1\}$$

$$\text{bird} \in \{0, 1\}$$

$$\text{eagle} \in \{0, 1\}$$

FOP synopsis

- 2 types: **objects** (John, father(x)); **values** (T, F in FOL; bounded reals or ints in FOP)
- **Functions** map object* → object
- **Predicates** map object* → value

} just like FOL

- **Connectives** min, max (like and, or), +, –
- **Scalar multiplication** (not)
- **Quantifiers** inf, sup over object-valued variables (like forall, exists)

Summary

	First-order logic	FOP
Operators	not¬, and ∧, or ∨	-, min ∧, max ∨, +, scalar *
Quantifiers	for-all ∀, exists ∃	inf Λ, sup V
Values	true, false	numbers*
Functions	objects -> object	
Predicates	objects -> value	
Terms: *objects*	variables, constants, function(t_1,...,t_n)	
Atoms: values	T, F, predicate(t_1,...,t_n)	floats, predicate(t_1,...,t_n)

FOP synopsis

- **Model**: set of objects, table of function values, table of predicate values
  - Stanley, William
  - father(S) = W, father(W) = S
  - eagle(S) = bird(S) = flies(S) = 1
    eagle(W) = bird(W) = flies(W) = 1

FOP synopsis

- **Value** of sentence in model: substitute in for objects, functions, predicates; combine values as expected
- **Value** of sentence: use best (maximizing) model

Inference

- Inference = generating entailed sentences
  - informally: true conclusions
- **Sound**: only get entailed sentences
- **Complete**: eventually get any entailed sentence

Entailment

- S entails S': value(S', M) ≥ value(S, M) for all models M
  - equivalently, value(S–S') ≥ 0
- Generalizes FOL: S' is true whenever S is

Lifted Gomory cuts

- Our sound & complete* inference method

- Put KB in normal form
- Subtract desired conclusion (proof by contradiction)
  - goal is now to test whether value ≥ 0

Lifted Gomory cuts

- Select a subset of clauses
- Bind variables to unify sets of terms
- Translate to a MILP (each syntactically distinct atom becomes a variable)
- Make a Gomory cut
  - if it demonstrates infeasibility, stop
- Translate cut back to FOP, add to KB
- Repeat

Eagles, revisited

- KB is already in normal form
- Select clauses: eagle(S)–1, (bird(x) – eagle(x))
- Bind x = S: bird(S) – eagle(S)
- Translate to MILP: $p \geq 1$, $q - p \geq 0$ (p, q in {0,1})
- Trivial cut: $q \geq 1$
- Translate back: bird(S)–1

A nontrivial cut

$$S = (x \vee y \vee z) \wedge (\neg z \vee y \vee a)$$

$$\begin{aligned} x + y + z &\geq 1 \\ (1 - z) + y + a &\geq 1 \\ x, y, z, a &\in \{0, 1\} \end{aligned}$$

$$x + 2y + a \geq 1$$

Example cont'd

$$x + 2y + a \geq 1$$

$$\frac{1}{2}x + y + \frac{1}{2}a - \frac{1}{2}s = \frac{1}{2}$$

$$x + y + a \geq \frac{1}{2}$$

$$x + y + a \geq 1$$

Summary

- Lifted language and inference method (FOP, lifted Gomory cuts)
- Subsumes FOL, MILP
- Specializes to efficient algorithms; always makes progress
- Current work: better handle "Bayes/DP" reasoning style
- Hope: allow tying together multiple scales, reasoning styles