Problem Solving with Graphical Models

Rina Dechter
Donald Bren School of Computer Science
University of California, Irvine, USA

Dagshtul, 2011
What is artificial intelligence?
It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Yes, but what is intelligence?
Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?
Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent. We understand some of the mechanisms of intelligence and not others.

**Mechanical Heuristic generation**

**Observation:** People generate heuristics by consulting simplified/relaxed models.

**Context:** Heuristic search (A*) of state-space graph (Nilsson, 1980)

**Context:** Weak methods vs. strong methods

**Domain knowledge:** Heuristic function

\( h(n) \): Heuristic underestimate the best cost from \( n \) to the solution
The Simplified models Paradigm

Pearl 1983 (On the discovery and generation of certain Heuristics, 1983, AI Magazine, 22-23) : “knowledge about easy problems could serve as a heuristic in the solution of difficult problems, i.e., that it should be possible to manipulate the representation of a difficult problem until it is approximated by an easy one, solve the easy problem, and then use the solution to guide the search process in the original problem.”

The implementation of this scheme requires three major steps:
- a) simplification,
- b) solution, and
- c) advice generation.

Simplified = relaxed is appealing because:
1. implies admissibility, monotonicity,
2. explains many human-generated heuristics (15-puzzle, traveling salesperson)

“We must have a simple a-priori criterion for deciding when a problem lends itself to easy solution.”
Systematic relaxation of STRIPS

STRIPS (Stanford Research Institute Problem Solver, Nillson and Fikes 1971)
action representation:

**Move(x,c1,c2)**
Precond list: on(x1,c1), clear(c2), adj(c1,c2)
  Add-list: on(x1,c2), clear(c1)
  Delete-list: on(x1,c1), clear(c2)

**Relaxation (Sacerdoti, 1974):** Remove literals from the precondition-list:
1. clear(c2), adj(c2,c3) \(\rightarrow\) #misplaced tiles
2. Remove clear(c2) \(\rightarrow\) manhattan distance
3. Remove adj(c2,c3) \(\rightarrow\) h3, a new procedure that transfer to the empty location a tile appearing there in the goal

But the main question remained:
"Can a program tell an easy problem from a hard one without actually solving?" (Pearl 1984, Heuristics)
Pearl, 84: Most easy problems we encounter are solved by “greedy” hill-climbing methods without backtracking” and that the features that make them amenable to such methods is their “decomposability”

The question now:
Can we recognize a greedily solved STRIPS problem?”
Whow! Backtrack-free is greedy!
Got absorbed...

**Sufficient condition (Freuder 82):**
1. Trees (width-1) and arc-consistency implies backtrack-free
2. Width=i and (i+1)-consistency implies backtrack-free search

If 3-consistent no deadends

W=2

Arc-consistent
No dead-ends

This moved me to constraint network and ultimately to graphical models.
But: Is it indeed the case that heuristics are generated by simplified Models?
Outline of the talk

- Introduction to graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - How can we predict problem instance hardness?
Outline

- What are graphical models
- Overview of Inference Search and their hybrids

- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)

Questions:
- Representation issues: directed vs undirected
- The role of hidden variables
- Finding good structure
- Representation guided by human representation
- Computation: inspired by human thinking
What are Graphical Models

- A way to represent *global* knowledge, mostly declaratively, using small local pieces of functions/relations. Combined, they give a global view of a world about which we want to reason, namely to answer queries.

- Different types of graphs can capture variable interaction through the local functions.

- Because representation is modular, reasoning can be modular too.
Constraint Networks

Map coloring

Variables: countries (A B C etc.)
Values: colors (red green blue)
Constraints: \( A \neq B, A \neq D, D \neq E, \ldots \)

Global view: all solutions.
Tasks: Is there a solution?, find one, find all, count all
Three dimensional interpretation of 2 dimentional drawing
Huffman-Clowes junction labelings (1975)

Fork: \[ \begin{align*} + & + & \quad + & - & \quad + & - & \quad - & + & \quad - & - \end{align*} \]

Arrow: \[ \begin{align*} + & \quad - & \quad - & \quad + & \quad + & \quad - & \quad + & \quad - \end{align*} \]

Ell: \[ \begin{align*} + & \quad + & \quad - & \quad + & \quad - & \quad + & \quad - & \quad + & \quad - \end{align*} \]

Tee: \[ \begin{align*} + & \quad - & \quad + & \quad - \end{align*} \]
Sudoku - Constraint Satisfaction

Each row, column and major block must be alldifferent

“Well posed” if it has unique solution: 27 constraints
Bayesian Networks
(Pearl, 1988)

Global view: \[ P(S, C, B, X, D) = P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B) \]

Belief Updating, Most probable tuple (MPE)

= find argmax \[ P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B) =? \]
The “alarm” network - 37 variables, 509 parameters (instead of $2^{37}$)
Mixed Probabilistic and Deterministic networks

Alex is **likely**-to-go in bad weather
Chris **rarely**-goes in bad weather
Becky is indifferent but **unpredictable**

If Alex goes, then Becky goes:
If Chris goes, then Alex goes:

**Query:**
*Is it likely that Chris goes to the party if Becky does not but the weather is bad?*

\[ P(C, \neg B \mid w = \text{bad}, A \rightarrow B, C \rightarrow A) \]
Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.
Graphical Models

- A graphical model \((X, D, F)\):
  - \(X = \{X_1, \ldots X_n\}\) variables
  - \(D = \{D_1, \ldots D_n\}\) domains
  - \(F = \{f_1, \ldots f_m\}\) functions

- Operators:
  - combination
  - elimination (projection)

| A | C | F | P(F|A,C) |
|---|---|---|---------|
| 0 | 0 | 0 | 0.14    |
| 0 | 0 | 1 | 0.96    |
| 0 | 1 | 0 | 0.40    |
| 0 | 1 | 1 | 0.60    |
| 1 | 0 | 0 | 0.35    |
| 1 | 0 | 1 | 0.65    |
| 1 | 1 | 0 | 0.72    |
| 1 | 1 | 1 | 0.68    |

Relation:

\[ f_i := (F = A + C) \]

Primal graph (interaction graph)
Complexity of Reasoning Tasks

- Constraint satisfaction
- Counting solutions
- Combinatorial optimization
- Belief updating
- Most probable explanation
- Decision-theoretic planning

Reasoning is computationally hard

Complexity is
Time and space (memory) exponential
Tree-solving is easy

Belief updating
(sum-prod)

CSP – consistency
(projection-join)

MPE (max-prod)

#CSP (sum-prod)

Trees are processed in linear time and memory
Counting

How many people?

SUM operator
CHAIN structure
Maximization

What is the maximum?
What is minimum cost configuration?
Belief Propagation

- Instances of tree message passing algorithm
- Exact for trees
- Linear in the input size

Importance:
- One of the first algorithms for inference in Bayesian networks
- Gives a cognitive dimension to its computations
- Basis for conditioning algorithms for arbitrary Bayesian network
- Basis for Loopy Belief Propagation (approximate algorithms)

(Pearl, 1988)
Transforming into a Tree

- **By Inference (thinking)**
  - Transform into a single, equivalent tree of sub-problems

- **By Conditioning (guessing)**
  - Transform into many tree-like sub-problems.
Inference and Treewidth

Inference algorithm:
Time: \( \exp(\text{tree-width}) \)
Space: \( \exp(\text{tree-width}) \)

Key parameter: \( w^* \)

\( \text{treewidth} = 4 - 1 = 3 \)

\( \text{treewidth} = (\text{maximum cluster size}) - 1 \)
Conditioning and Cycle cutset

Cycle cutset = \{A,B,C\}

A

B

C
Search over the Cutset

Graph Coloring problem

- Inference may require too much memory
- **Condition (guessing)** on some of the variables
Search over the Cutset (cont)

Key parameters:
the cycle-cutset, w-cutset
Complexity: \(\exp(\text{cutset-size})\)

• Inference may require too much memory
• **Condition** on some of the variables

Graph Coloring problem

Key parameters: the cycle-cutset, w-cutset
Complexity: \(\exp(\text{cutset-size})\)
Inference vs Conditioning - Search

**Inference**

exp(treewidth) time/space

**Search**

exp(n) time
Or exp(pseudo-tree O(n) space

Search+inference:
Space: exp(w of sub problem)
Time: exp(w+cutset(w))
Approximation Algorithms

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:
  - **Bounding inference:** Bounding the clusters by i-bound
    - mini-bucket(i) and bounded-i-consistency
    - Belief propagation and constraint propagation
  - **Bounding search:**
    - Sampling
    - Stochastic local search
  - Hybrid of sampling and bounded inference
  - Goal: an anytime scheme
Search vs. Inference

Search (conditioning)

Inference (elimination)

k “sparser” problems

1 “denser” problem
Outline

- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking

Complete
Incomplete
Simulated Annealing
Gradient Descent
Adaptive Consistency
Tree Clustering
Variable Elimination
Mini-bucket(i)
Stochastic Local Search
DFS search
Branch-and-Bound
A*
Bucket Elimination, Variable elimination

Bucket E: \( E \neq D, \ E \neq C \)
Bucket D: \( D \neq A \)
Bucket C: \( C \neq B \)
Bucket B: \( B \neq A \)
Bucket A: contradiction
Bucket Elimination
Adaptive Consistency (Dechter & Pearl, 1987)

Complexity: \(O(n \exp(w^*(d)))\),

\(w^*(d)\) - induced width along ordering \(d\)

we get a greedily solved problem (backtrack-free)
Directional Resolution $\Leftrightarrow$ Bucket-elimination

$|\text{bucket}_i| = O(\exp(w^*))$

DR time and space: $O(n \exp(w^*))$

(Original Davis-Putnam algorithm 1960)
Belief Updating/Probability of evidence
Partition function

\[
P(a|e=0) \propto P(a,e=0) = \\
\sum_{e=0,d,c,b} P(a)P(b|a)P(c|a)P(d|b,a)P(e|b,c) = \\
P(a) \sum_{e=0} \sum_{d} \sum_{c} P(c|a) \sum_{b} P(b|a)P(d|b,a)P(e|b,c)
\]

Variable Elimination

\[h^B(a,d,c,e)\]
Bucket Elimination

Algorithm *elim-bel* (Dechter 1996)

\[ \sum_{b} \prod \]

Elimination operator

bucket B: \( P(b|a) \) \( P(d|b,a) \) \( P(e|b,c) \)

bucket C: \( P(c|a) \) \( h^B(a,d,c,e) \)

bucket D: \( h^C(a,d,e) \)

bucket E: \( e=0 \) \( h^D(a,e) \)

bucket A: \( P(a) \) \( h^E(a) \)

\( P(a|e=0) \)

\( W^* = 4 \) \( \text{Exp}(w^*) \)
Finding

\[ \text{MPE} = \max_{\bar{x}} P(\bar{x}) \]

\[ \sum \text{is replaced by} \max : \]

\[ \text{MPE} = \max_{a,e,d,c,b} P(a)P(c|a)P(b|a)P(d|a,b)P(e|b,c) \]

bucket B: \[ P(b|a) \quad P(d|b,a) \quad P(e|b,c) \]

bucket C: \[ P(c|a) \quad h_B^b(a, d, c, e) \]

bucket D: \[ h_C^c(a, d, e) \]

bucket E: \[ e=0 \quad h_D^e(a, e) \]

bucket A: \[ P(a) \quad h_E^a(a) \quad \text{MPE} \]

\( W^* = 4 \)

"induced width" (max clique size)
Generating the MPE-tuple

1. $a' = \arg\max_a P(a) \cdot h^E(a)$
2. $e' = 0$
3. $d' = \arg\max_d h^C(a', d, e')$
4. $c' = \arg\max_c P(c | a') \times h^B(a', d', c, e')$
5. $b' = \arg\max_b P(b | a') \times P(d' | b, a') \times P(e' | b, c')$

Return $(a', b', c', d', e')$
Cluster Tree Propagation

Join-tree clustering (Spigelhalter et. Al. 1988, Dechter, Pearl 1987)

For each cluster \( P(X|e) \) is computed

- \( A \) \( B \) \( C \)
  - \( p(a), p(b|a), p(c|a,b) \)
  - \( h_{(1,2)}(b,c) = \sum_a p(a) \cdot p(b|a) \cdot p(c|a,b) \)

- \( B \) \( C \) \( D \) \( F \)
  - \( p(d|b), p(f|c,d) \)
  - \( h_{(2,3)}(b,f) = \sum_{c,d} p(d|b) \cdot p(f|c,d) \cdot h_{(1,2)}(b,c) \)
  - \( h_{(3,2)}(b,f) = \sum_e p(e|b,f) \cdot h_{(4,3)}(e,f) \)

- \( B \) \( E \) \( F \)
  - \( p(e|b,f) \)
  - \( h_{(3,4)}(e,f) = \sum_b p(e|b,f) \cdot h_{(2,3)}(b,f) \)
  - \( h_{(4,3)}(e,f) = p(G=g_e|e,f) \)

Time: \( O(\exp(w+1)) \)
Space: \( O(\exp(sep)) \)
Complexity of Elimination

Trees are easy

\[ O(n \exp(w^*(d))) \]

\( w^*(d) \) – the induced width of moral graph along ordering \( d \)

The effect of the ordering:

\[ w^*(d_1) = 4 \]
\[ w^*(d_2) = 2 \]

Spigelhalter et. Al. 1983, Junction tree algorithm (join-tree algorithm)
Outline

- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking
Approximate Inference

- Mini-buckets, mini-clusters, i-consistency
- Belief propagation, constraint propagation, Generalized belief propagation
From Global to Local Consistency

leads to one pass directional bounded inference, or Iterative propagation algorithms.
Mini-Bucket Elimination

\[
\begin{align*}
\max_B \prod & \quad \max_B \prod \\
\text{Bucket B} & \quad P(E|B,C) \quad P(B|A) \quad P(D|A,B) \\
\text{Bucket C} & \quad P(C|A) \quad h^B(C,E) \\
\text{Bucket D} & \quad h^B(A,D) \\
\text{Bucket E} & \quad E = 0 \quad h^C(A,E) \\
\text{Bucket A} & \quad P(A) \quad h^E(A) \quad h^D(A)
\end{align*}
\]

\text{MPE* is an upper bound on MPE --U}
\text{Generating a solution yields a lower bound --L}
MBE(i) (Dechter and Rish 1997)

- **Input:** \( i \) - max number of variables allowed in a mini-bucket
- **Output:** [lower bound (P of a sub-optimal solution), upper bound]

Example: `approx-mpe(3)` versus `elim-mpe`

![Diagram showing mini-bucket example](image)

\[ w^* = 2 \quad \text{vs} \quad w^* = 4 \]
Properties of MBE(i)/mc(I)

- **Complexity:** $O(r \exp(i))$ time and $O(\exp(i))$ space.
- Yields an upper-bound and a lower-bound.

- **Accuracy:** determined by upper/lower (U/L) bound.

- As $i$ increases, both accuracy and complexity increase.

- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search

- Other tasks: similar mini-bucket approximations for: belief updating, MAP and MEU (Dechter and Rish, 1997)
Approximate Inference

- Mini-buckets, mini-clusters
- Belief propagation, constraint propagation, Generalized belief propagation
Iterative, directional algorithms, vs Propagation algorithms

Arcs-consistency

$1 \leq X, Y, Z, T \leq 3$

$X < Y$

$Y = Z$

$T < Z$

$X \leq T$
Arc-consistency

\[ 1 \leq X, Y, Z, T \leq 3 \]
\[ X < Y \]
\[ Y = Z \]
\[ T < Z \]
\[ X \leq T \]

\[ R_X \leftarrow \prod_X R_{XY} \otimes D_Y \]
Iterative (Loopy) Belief Propagation

- Belief propagation is exact for poly-trees
- IBP - applying BP iteratively to cyclic networks

One step:
- Update $\text{BEL}(U_1)$
- $\lambda_{X_1}(u_1)$
- $\pi_{U_2}(x_1)$
- $\lambda_{X_2}(u_1)$
- $\pi_{U_3}(x_1)$

- No guarantees for convergence
- Works well for many coding networks
Collapsing Clusters
Join-Graphs

more accuracy

less complexity
Outline

- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking
Backtracking Search for a Solution
Belief Updating: Searching the Probability Tree

\[ P(a, e = 0) = P(a) \sum_b P(b \mid a) \sum_c P(c \mid a) \sum_b P(d \mid a, b) \sum_{e=0} P(e \mid b, c) \]

Brute-force Complexity: \( O(\exp(n)) \), linear space
Same as counting solutions
Ordering: A B E C D F

Constraint network

Size of search space: $O(exp(n))$
AND/OR Search Space

**Primal graph**

**DFS tree**
AND/ OR vs. OR

AND/OR size: exp(4), OR size exp(6)
Pseudo-Trees
(Freuder 85, Bayardo 95, Bodlaender and Gilbert, 91)

\[ h \leq w \times \log n \]

(a) Graph

(b) DFS tree
    depth=3

(c) pseudo-tree
    depth=2

(d) Chain
    depth=6
## Complexity of AND/OR Tree Search

<table>
<thead>
<tr>
<th></th>
<th>AND/ OR tree</th>
<th>OR tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(n , d^h)$</td>
<td>$O(d^n)$</td>
</tr>
<tr>
<td></td>
<td>$O(n , d^{w* \log n})$</td>
<td></td>
</tr>
</tbody>
</table>

*Freuder & Quinn85*, *(Collin, Dechter & Katz91)*, *(Bayardo & Miranker95)*, *(Darwiche01)*

\[
d = \text{domain size} \\
h = \text{depth of pseudo-tree} \\
n = \text{number of variables} \\
w* = \text{treewidth}
\]
An AND/OR Graph
Complexity of AND/OR Graph Search

<table>
<thead>
<tr>
<th></th>
<th>AND/ OR graph</th>
<th>OR graph</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>O(n (d^{w^*}))</td>
<td>O(n (d^{pw^*}))</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>O(n (d^{w^*}))</td>
<td>O(n (d^{pw^*}))</td>
</tr>
</tbody>
</table>

\[w^* \leq pw^* \leq w^* \log n\]

d = domain size  
n = number of variables  
w* = treewidth  
pw* = pathwidth
All Four Search Spaces

Full OR search tree
126 nodes

Full AND/OR search tree
54 AND nodes

Context minimal OR search graph
28 nodes

Context minimal AND/OR search graph
18 AND nodes
**Theorem**: The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.

\[
\text{max context size} = \text{treewidth}
\]
AND/OR Context Minimal Graph

Variable Elimination

AND/OR Search
The impact of the pseudo-tree

What is a good pseudo-tree? How to find a good one?

W=4, h=8

W=5, h=6
Outline

- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking
The context-minimal graph
Can be "minimized into an AOMDD
By merging and redundancy removal

AOBDD
18 nonterminals
47 arcs

OBDD
27 nonterminals
54 arcs
Constraint Network vs Bayesian Network

**Constraint networks**
- **Is it consistent?**
- **Find solution**
  - NP-complete
- Count solutions
  - #P-complete
- Unminimal const
- Solved by search
- Hard to sample

**Probability networks**
- Always consistent
- Find $t$ s.t $P(t)>0$
  - Easy: backtrack-free
- **Find $P(X|e)$?**
  - #P-complete
- Explicit minimal tables
- Solved by variable elimination
- Easy to sample

represents $\text{sol (A, B, C, D, E, F)}$
represents $P(A, B, C, D, E, F)$
The End

Thank You