Algorithms for reasoning with graphical models

Class 1

Rina Dechter

• **Graphical models:** The constraint network, Probabilistic networks, cost networks and mixed networks. queries: consistency, counting, optimization and likelihood queries.

• **Inference:** Bucket elimination for deterministic networks (Adaptive-consistency, and the Davis-Putnam algorithms.) The induced-width
  - Inference: Bucket-elimination for Bayesian and Markov networks queries (mpe, map, marginal and probability of evidence)
  - **Graph properties:** induced-width, tree-width, chordal graphs, hypertrees, join-trees.
  - Inference: Tree-decomposition algorithms (join-tree propagation and junction-trees)

• **Approximation by bounded Inference:** (weighted Mini-bucket, belief/constraint-propagation, constraint propagation, generalized belief propagation, variational methods)

• **Search for csps:** Backtracking; pruning search by constraint propagation, backjumping and learning.

• **Search:** AND/OR search Spaces for likelihood, optimization queries (Probability of evidence, Partition function, MAP and MPE queries, AND/OR branch and bound).

• **Approximation by sampling:** Gibbs sampling, Importance sampling, cutset-sampling, SampleSearch and AND/OR sampling, Stochastic Local Search.

• **Hybrid of search Inference:** cutset-conditioning and cutset-sampling
Outline

- **Graphical models:** The constraint network, Probabilistic networks, cost networks and mixed networks. Graphical representations and queries: consistency, counting, optimization and likelihood queries.

- **Constraints inference:** Bucket elimination for deterministic networks (Adaptive-consistency, and the Davis-Putnam algorithms.) The induced-width.

- Inference: Bucket-elimination for Bayesian and Markov networks queries (mpe, map, marginal and probability of evidence)

- Graph properties: induced-width, tree-width, chordal graphs, hypertrees, join-trees.

- Inference: Tree-decomposition algorithms (join-tree propagation and junction-trees algorithm, Cluster tree-elimination.)

- Approximation by bounded Inference: (Mini-bucket, belief-propagation, constraint propagation, generalized belief propagation)

- Search: Backtracking search algorithms; pruning search by constraint propagation, backjumping and learning.
Course Requirements/Textbook

- Homeworks: There will be 5-6 problem sets, graded 70% of the final grades.

- A term project: paper presentation, a programming project.

- Books:
  - "Reasoning with probabilistic and deterministic graphical models", R. Dechter, Claypool, 2013
    https://www.morganclaypool.com/doi/abs/10.2200/S00529ED1V01Y201308AIM023
Outline of classes

• Part 1: Introduction and Inference

• Part 2: Search

• Part 3: Variational Methods and Monte-Carlo Sampling
RoadMap: Introduction and Inference

- Basics of graphical models
  - Queries
  - Examples, applications, and tasks
  - Algorithms overview

- Inference algorithms, exact
  - Bucket elimination for trees
  - Bucket elimination
  - Jointree clustering
  - Elimination orders

- Approximate elimination
  - Decomposition bounds
  - Mini-bucket & weighted mini-bucket
  - Belief propagation

- Summary and Part 2
RoadMap: Introduction and Inference

• Basics of graphical models
  – Queries
  – Examples, applications, and tasks
  – Algorithms overview

• Inference algorithms, exact
  – Bucket elimination for trees
  – Bucket elimination
  – Jointree clustering
  – Elimination orders

• Approximate elimination
  – Decomposition bounds
  – Mini-bucket & weighted mini-bucket
  – Belief propagation

• Summary and Class 2
Probabilistic Graphical models

• Describe structure in large problems
  – Large complex system $F(X)$
  – Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  – Complexity emerges through interdependence
Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence

- Examples & Tasks
  - Maximization (MAP): compute the most probable configuration

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha) \quad f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$

![Phenylalanine][1]

[1]: Yanover & Weiss 2002
Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence

- Examples & Tasks
  - Summation & marginalization
    $p(x_i) = \frac{1}{Z} \sum_{x \neq x_i} \prod_{\alpha} f_\alpha(x_\alpha)$ and $Z = \sum_x \prod_{\alpha} f_\alpha(x_\alpha)$

```
Observation y
```

```
Marginals \ p( x_i \mid y )
```

```
Observation y
```

```
Marginals \ p( x_i \mid y )
```

* e.g., [Plath et al. 2009]
Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence

- Examples & Tasks
  - Mixed inference (marginal MAP, MEU, ...)
  \[ f(x^*_M) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_\alpha) \]

Influence diagrams & optimal decision-making

(the “oil wildcatter” problem)

e.g., [Raiffa 1968; Shachter 1986]
In more details...
Constraint Networks

Example: map coloring

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

Example: party problem

- If Alex goes, then Becky goes: $A \rightarrow B$
- If Chris goes, then Alex goes: $C \rightarrow A$
- Question:
  Is it possible that Chris goes to the party but Becky does not?

Is the propositional theory

$$\varphi = \{A \rightarrow B, C \rightarrow A, \neg B, C\}$$
satisfiable?
Bayesian Networks (Pearl 1988)

\[ P(S, C, B, X, D) = P(S) P(C|S) P(B|S) P(X|C,S) P(D|C,B) \]

- Posterior marginals, probability of evidence, MPE

- \[ P(D = 0) = \sum_{S,L,B,X} P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B) \]

MAP(P)= \[ max_{S,L,B,X} P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B) \]

\[ \text{BN} = (G, \Theta) \]

Combination: Product
Marginalization: sum/max

\begin{array}{c|c|c|c}
C & B & P(D|C,B) \\
\hline
0 & 0 & 0.1 & 0.9 \\
0 & 1 & 0.7 & 0.3 \\
1 & 0 & 0.8 & 0.2 \\
1 & 1 & 0.9 & 0.1 \\
\end{array}
Probabilistic reasoning (directed)

Party example: the weather effect

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

Questions:
- **Given bad weather, which group of individuals is most likely to show up at the party?**
- **What is the probability that Chris goes to the party but Becky does not?**

\[
P(W, A, C, B) = P(B | W) \cdot P(C | W) \cdot P(A | W) \cdot P(W) 
\]

\[
P(A, C, B | W=bad) = 0.9 \cdot 0.1 \cdot 0.5 
\]
The “alarm” network: 37 variables, 509 parameters (rather than $2^{37} = 10^{11}$ !)
Alex is-likely-to-go in bad weather
Chris rarely-goes in bad weather
Becky is indifferent but unpredictable

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

\[ P(C, \neg B \mid w = bad, A \rightarrow B, C \rightarrow A) \]
Graphical models (cost networks)

A graphical model consists of:

\[ X = \{X_1, \ldots, X_n\} \quad \text{-- variables} \]
\[ D = \{D_1, \ldots, D_n\} \quad \text{-- domains (we’ll assume discrete)} \]
\[ F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \quad \text{-- functions or “factors”} \]

and a combination operator

The combination operator defines an overall function from the individual factors, e.g., “+”:

\[ F(A, B, C) = f_{AB}(A, B) + f_{BC}(B, C) \]

Notation:

Discrete \( X_i \) values called states

Tuple or configuration: states taken by a set of variables

Scope of \( f \): set of variables that are arguments to a factor \( f \)

often index factors by their scope, e.g., \( f_{\alpha}(X_{\alpha}), \quad X_{\alpha} \subseteq X \)
Graphical models (cost networks)

A graphical model consists of:

- $X = \{X_1, \ldots, X_n\}$ -- variables
- $D = \{D_1, \ldots, D_n\}$ -- domains (we’ll assume discrete)
- $F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or “factors”

and a combination operator

$$F(A, B, C) = f_{AB}(A, B) + f_{BC}(B, C)$$

For discrete variables, think of functions as “tables” (though we might represent them more efficiently)

$$F(A = 0, B = 1, C = 1)$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>f(A,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>f(B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

A $\in \{0, 1\}$

$B \in \{0, 1\}$

$C \in \{0, 1\}$

$f_{AB}(A, B), \quad f_{BC}(B, C)$
Graph Visualization: Primal Graph

A graphical model consists of:

\[ X = \{X_1, \ldots, X_n\} \quad \text{-- variables} \]
\[ D = \{D_1, \ldots, D_n\} \quad \text{-- domains} \]
\[ F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \quad \text{-- functions or “factors”} \]

and a combination operator

Primal graph:
variables → nodes
factors → cliques

\[
F(A, B, C, D, F, G) = f_1(A, B, D) + f_2(D, F, G) \\
+ f_3(B, C, F) + f_4(A, C)
\]
Example: Constraint networks

\[ X_i \in \{ \text{red, green, blue} \} \]

\[ f_{ij}(X_i, X_j) = (X_i \neq X_j) \text{ for adjacent regions } i,j \]

Overall function is “and” of individual constraints:
\[ F(X) = f_{01}(X_0, X_1) \land f_{12}(X_1, X_2) \land f_{02}(X_0, X_2) \land \ldots \]

“Tabular” form:
\[ f_{ij}(X_i, X_j) = \begin{cases} 1.0 & X_i \neq X_j \\ 0.0 & X_i = X_j \end{cases} \]

\[ F(X) = \prod_{i,j} f_{ij}(X_i, X_j) = \begin{cases} 1.0 & \text{all valid} \\ 0.0 & \text{any invalid} \end{cases} \]

Tasks: “max”: is there a solution?
“sum”: how many solutions?
Markov logic, Markov networks

[Richardson & Domingos 2005]

1.5 \( \forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x) \)

1.1 \( \forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \iff \text{Smokes}(y)) \)

Two constants: Anna (A) and Bob (B)

\[
\begin{array}{c|c|c|c}
S_A & C_A & f(S_A, C_A) \\
0 & 0 & \exp(1.5) \\
0 & 1 & \exp(1.5) \\
1 & 0 & 1.0 \\
1 & 1 & \exp(1.5) \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
F_{AB} & S_A & S_B & f(.) \\
0 & 0 & 0 & \exp(1.1) \\
0 & 0 & 1 & \exp(1.1) \\
0 & 1 & 0 & \exp(1.1) \\
0 & 1 & 1 & \exp(1.1) \\
1 & 0 & 0 & \exp(1.1) \\
1 & 0 & 1 & 1.0 \\
1 & 1 & 0 & 1.0 \\
1 & 1 & 1 & \exp(1.1) \\
\end{array}
\]
A graphical model consists of:

\[ X = \{X_1, \ldots, X_n\} \quad \text{-- variables} \]
\[ D = \{D_1, \ldots, D_n\} \quad \text{-- domains} \]
\[ F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \quad \text{-- functions or “factors”} \]

and a combination operator

Primal graph:
variables \rightarrow nodes
factors \rightarrow cliques

\[ F(A, B, C, D, F, G) = f_1(A, B, D) + f_2(D, F, G) \]
\[ + f_3(B, C, F) + f_4(A, C) \]

Dual graph:
factor scopes \rightarrow nodes
edges \rightarrow intersections (separators)
Graphical visualization

“Factor” graph: explicitly indicate the scope of each factor
variables → circles
factors → squares

\[
F(A, B, C, D, F, G) = f_1(A, B, D) + f_2(D, F, G) \\
+ f_3(B, C, F) + f_4(A, C)
\]

Useful for disambiguating factorization:

\[
f_1(A, B, C, D) \quad \text{vs.} \quad f_1(A, B) + f_2(A, C) + \ldots
\]

\[
O(d^4) \quad \text{vs.} \quad \text{pairwise: } O(d^2)
\]
Graphical models

A **graphical model** consists of:

- \( X = \{X_1, \ldots, X_n\} \) -- variables
- \( D = \{D_1, \ldots, D_n\} \) -- domains
- \( F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \) -- functions or “factors”

**Operators:**

- combination operator  
  (sum, product, join, ...)

- elimination operator  
  (projection, sum, max, min, ...)

**Types of queries:**

- Marginal: \( Z = \sum_x \prod_{\alpha} f_\alpha(x_\alpha) \)
- MPE / MAP: \( f(x^*) = \max_x \prod_{\alpha} f_\alpha(x_\alpha) \)
- Marginal MAP: \( f(x^*_M) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_\alpha) \)

- All these tasks are NP-hard
- exploit problem structure
- identify special cases
- approximate
Definition 2.1.2 (graphical model) A graphical model $\mathcal{M}$ is a 4-tuple, $\mathcal{M} = \langle X, D, F, \otimes \rangle$, where:

1. $X = \{X_1, \ldots, X_n\}$ is a finite set of variables;

2. $D = \{D_1, \ldots, D_n\}$ is the set of their respective finite domains of values;

3. $F = \{f_1, \ldots, f_r\}$ is a set of positive real-valued discrete functions, defined scopes of variables $S_i \subseteq X$,

4. $\otimes$ is a combination operator (e.g., $\otimes \in \{\prod, \sum, \wedge\}$ (product, sum, join)).

The graphical model represents a global function whose scope is $X$ which is the combination of all its functions: $\otimes_{i=1}^r f_i$.

Definition 2.1.3 (a reasoning problem) A reasoning problem over a graphical model $\mathcal{M} = \langle X, D, F, \otimes \rangle$ and a subset of variable $Y \subseteq X$ is defined by a marginalization operator $\downarrow_Y$. If $S$ is the scope of function $f$ then $\downarrow_Y f \in \{\max_{S-Y} f, \min_{S-Y} f, \pi_Y f, \sum_{S-Y} f\}$ is a marginalization operator. The reasoning problem $\mathcal{P}(\mathcal{M}, \downarrow_Y, Z)$ is the task of computing the function $\mathcal{P}_\mathcal{M}(Z) = \downarrow_Z \otimes_{i=1}^r f_i$, where $r$ is the number of functions in $F$. 
Summary of graphical models types

• Constraint networks
• Cost networks
• Bayesian network
• Markov networks
• Mixed probability and constraint network
• Influence diagrams
Constraint Networks

Map coloring

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

Queries: Find one solution, all solutions, counting

Combination = join
Marginalization = projection
Example of a Cost Network

(a) Cost functions

(b) Constraint graph

Figure 2.3: A cost network.

Definition 2.3.2 (WCSP) A Weighted Constraint Satisfaction Problem (WCSP) is a graphical model \( \langle X, D, F, \sum \rangle \) where each of the functions \( f_i \in F \) assigns "0" (no penalty) to allowed tuples and a positive integer penalty cost to the forbidden tuples. Namely, \( f_i : D_{X_{i_1}} \times \ldots \times D_{X_{i_t}} \rightarrow \mathbb{N} \), where \( S_i = \{X_{i_1}, \ldots, X_{i_t}\} \) is the scope of the function.
A Bayesian Network

Combination: product
Marginalization: sum or min/max
Figure 2.6: (a) An example $3 \times 3$ square Grid Markov network (ising model) and (b) An example potential $H_6(D, E)$

The network represents a global joint distribution over the variables $X$ given by:

$$P(x) = \frac{1}{Z} \prod_{i=1}^{m} H_i(x), \quad Z = \sum_{x \in X} \prod_{i=1}^{m} H_i(x)$$
Example domains for graphical models

• Natural Language processing
  – Information extraction, semantic parsing, translation, topic models, ...

• Computer vision
  – Object recognition, scene analysis, segmentation, tracking, ...

• Computational biology
  – Pedigree analysis, protein folding and binding, sequence matching, ...

• Networks
  – Webpage link analysis, social networks, communications, citations, ....

• Robotics
  – Planning & decision making
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)
RoadMap: Introduction and Inference

• Basics of graphical models
  – Queries
  – Examples, applications, and tasks
  – Algorithms overview

• Inference algorithms, exact
  – Bucket elimination for trees
  – Bucket elimination
  – Jointree clustering
  – Elimination orders

• Approximate elimination
  – Decomposition bounds
  – Minibucket & weighted mini-bucket
  – Belief propagation

• Summary and Class 2
### Types of queries

<table>
<thead>
<tr>
<th>Inference Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Inference</td>
<td>[ f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha) ]</td>
</tr>
<tr>
<td>Sum-Inference</td>
<td>[ Z = \sum_x \prod_\alpha f_\alpha(x_\alpha) ]</td>
</tr>
<tr>
<td>Mixed-Inference</td>
<td>[ f(x_{*M}) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha) ]</td>
</tr>
</tbody>
</table>

- **NP-hard**: exponentially many terms
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate! Slower & more accurate
Tree-solving is easy

Belief updating (sum-prod)

\[ P(X) \]
\[ m_{xy}(X) \rightarrow P(Y|X) \]
\[ m_{yx}(X) \rightarrow P(X) \]
\[ m_{xz}(X) \rightarrow P(Z|X) \]
\[ m_{zx}(X) \rightarrow P(X) \]

\[ m_{yz}(Y) \rightarrow P(Y|X) \]
\[ m_{zy}(Y) \rightarrow P(X) \]
\[ m_{zr}(Z) \rightarrow P(Z|X) \]
\[ m_{rz}(Z) \rightarrow P(X) \]

MPE (max-prod)

CSP – consistency (projection-join)

Trees are processed in linear time and memory

#CSP (sum-prod)
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: $\exp(\text{tree-width})$

$\text{treewidth} = 4 - 1 = 3$
$\text{treewidth} = (\text{maximum cluster size}) - 1$
Conditioning and Cycle cutset

Cycle cutset = \{A,B,C\}
Search over the Cutset

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

Graph Coloring problem

- A=yellow
- A=green
- B=red
- B=blue
- B=green
- B=yellow

*class1 276-2018*
Bird's-eye View of Exact Algorithms

Inference

exp(w*) time/space

Search

Exp(w*) time
O(w*) space

Search+inference:
Space: exp(q)
Time: exp(q+c(q))
q: user controlled
Bird's-eye View of Exact Algorithms

Inference

\[ \text{exp}(w^*) \text{ time/space} \]

Search

\[ \text{Exp}(w^*) \text{ time } O(w^*) \text{ space} \]

Search+inference:

Space: \( \text{exp}(q) \)

Time: \( \text{exp}(q+c(q)) \)

q: user controlled
Bird's-eye View of Approximate Algorithms

Inference

Bounded Inference

Search

Sampling

Search + inference:

Sampling + bounded inference

Context minimal AND/OR search graph
18 AND nodes