1 Introduction

Algorithms

- Don’t ask complexity
- Don’t ask correctness
- Ask them if they are useful
- Achieve communication at Shannon limit

Examples

- Belief Propagation in Bayesian Networks
- Viterbi Algorithm (PDAs)
- Kalman Filter (Airplanes)
- Turbo Algorithm - first to meet Shannon limit

Outline

- Problem Statement
- Graphical Models
  - Bayesian Networks
  - Factor Graphs
  - Pairwise Markov Random Fields
- Equivalence of Graphical Models

Reference: Yedidia, Freeman, Weiss. “Understanding Belief Propagation and it’s generalization”.

2 Problem Statement

- Discrete system, state vector $\vec{x}$, $x_i \in Dom$
- Weight of each state (configuration), $weight(\vec{x}) \in \mathbb{R}_0^+$
- Represent the weight concisely ⇒ Graphical Models
  - Graph (directed or undirected)
    $V \sim$ state variables
    $E \sim$ (in)dependencies
Weights are “distributed” among the nodes and edges

**Special Case**

Configurations ($\vec{x}$) $\sim$ possible worlds in some problem space

$\text{weight}(\vec{x}) = P[\vec{X} = \vec{x}]$, where $P$ is the probability of some event occurring.

- Joint Probability $P[\vec{X} = \vec{x}]$
- Marginal Probability $P[X_i = x_i] = \sum_{\vec{x}_{-i}} P[\vec{X} = \vec{x}]$, where $\vec{x}_{-i} \triangleq (x_j)_{j \neq i}$

**Questions (Tasks/Queries)**

- Compute the marginal probabilities
- Compute the most likely configuration $\arg \max_{\vec{x}} \text{weight}(\vec{x})$

**Inference** is the mechanism to answer queries ($\#P$-complete problems)

### 3 Bayesian Networks

- Directed Acyclic Graph (DAG)
- $V$ = state variables
- Independence Property: $X_i$ is independent of Non-descendants($X_i$) given Parents($X_i$)

![Bayesian Network Diagram](image)

\[
P(s, h, l, b) = P(b|s, h, l) P(s, h, l)
= P(b|h, l) P(h, l|s) P(s)
= P(b|h, l) P(h|s) P(l|s) P(s)
\]

$\Rightarrow P[\vec{X} = \vec{x}] = \prod_i P[X_i|\text{Par}(X_i)]$

- In Bayesian Networks, we may compute the probability of a given node given some evidence $P[X_i = x_i|\text{evidence}]$

### 4 Factor Graphs

- Undirected bipartite graph
- Two kinds of nodes:
  1. Variables $\bigcirc x_i \in \text{Dom}$
  2. Factors $\square f_\alpha(\vec{x}_\alpha) \in \mathbb{R}_0^{+}$, where $\vec{x}_\alpha$ is a subvector of $\vec{x}$

- $\text{weight}(\vec{x}) \triangleq \prod_\alpha f_\alpha(\vec{x}_\alpha)$
SAT: \((x \lor y) \land (\neg x \lor z)\)

Let \(\alpha = (x \lor y)\) and \(\beta = (\neg x \lor z)\)

\[
f_\alpha = \begin{cases} 
1 & \text{iff } (x \lor y) \\
0 & \text{otherwise}
\end{cases}
\]

\[
f_\beta = \begin{cases} 
1 & \text{iff } (\neg x \lor z) \\
0 & \text{otherwise}
\end{cases}
\]

• Probabilistic Interpretation

\[
P[\vec{X} = \vec{x}] \propto \text{weight}(\vec{x})
\]

\[= \frac{1}{z} \text{weight}(\vec{x}), \text{ where } z = \sum_{\vec{x}} \text{weight}(\vec{x})
\]