Factored Probabilistic Belief Tracking

Blai Bonet[1] and Hector Geffner[2]

[1]Universidad Simón Bolívar, Caracas, Venezuela
[2]ICREA & Universitat Pompeu Fabra, Barcelona, Spain

IJCAI. New York, USA. July 2016.

Motivation

Partially Observable MDPs (POMDPs) can be described **compactly**

Key question is how to use the compact representation for:

1. Keeping track of beliefs (distribution over states)
2. Action selection for achieving goals

This work is about 1, but efficient tracking is required as well when monitoring partially observable stochastic systems

Basic, Flat Algorithm for Probabilistic Belief Tracking

Task: Given initial belief b_0, transitions $P(s'|s,a)$ and sensing $P(o|s,a)$, compute posterior $P(s_{t+1}|o_t,a_t,\ldots,o_0,a_0,b_0)$

Basic algorithm: Use plain Bayes updating $b_{t+1} = b_a^o$ for $b = b_t$:

$$\begin{aligned} b_a^o(s) &\propto P(o|s,a) \times b_a(s) \\ b_a(s) &= \textstyle\sum_{s'} P(s|s',a)\, b(s') \end{aligned}$$

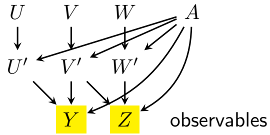
Complexity: Linear in # of states (single update) that is **exponential** in number of variables (task is **untractable** for compact POMDPs)

Challenge: **Exploit structure** to scale up better when not worst case

Structure of Actions and Sensing: Dynamic Bayesian Network

As usual, we assume transition and sensing probabilities given by **2-layer dynamic bayesian network (2-DBN):**

- state variables at times t and $t+1$
- single action variable at time t
- observation variables at time $t+1$



Posterior at time t corresponds to marginal over state variables at time t over **unfolded 2-DBN**

Main obstacle: Even if 2-DBN is sparse, all state variables interact so **treewidth** of unfolded DBN becomes **unbounded in worst case**

Approximate Inference for DBNs

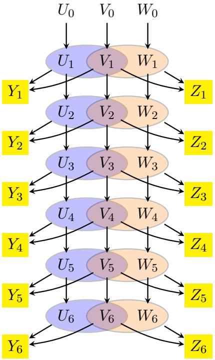
- Sampling: (Rao-Blackwellized) particle filtering
  - Sample selected variables to make inference tractable
- Decomposition: Boyen-Koller (BK), Factored Frontier (FF), etc.
  - Joint distribution **approximated** at each time step as product of **marginals** over clusters (BK) or variables (FF)

Our contribution:

- **Principled** and **general formulation** where:
  - Joint at each time step maintained **exactly** as product of **non-disjoint** and **non-arbitrary** factors, under general **decomposability conditions**
  - **Sampling** (if necessary) done to make these conditions true

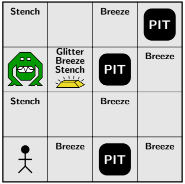
Beam Tracking (B & G, JAIR 2014)

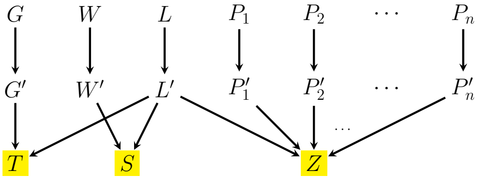
- 2-DBN gives groups of state vars called **beams**:
  - for each observable variable Z, a beam B that contains:
    - □ **parents** of Z in 2-DBN
    - □ parents of such parents in 2-DBN **recursively**
- Beams thus determined by 2-DBN and non-arbitrary or disjoint (usually)
- **Causal width** defined as size of largest beam



Beam tracking is belief tracking algorithm for **logical POMDPs** **exponential in causal width**; here we formulate **probabilistic version**

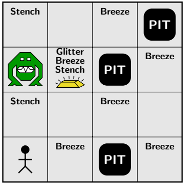
Example: Basic Model for Wumpus (Causal Width = $n+1$)

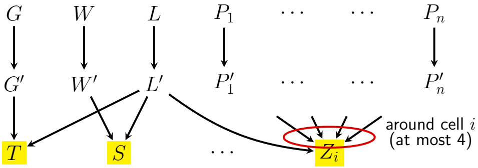




- $n+3$ vars: G (gold), W (wumpus), L (agent), P_1 (pit@1), ..., P_n (pit@n)
- 3 obs vars: T (glitter), S (stench) and Z (breeze)
- 3 beams: $B_0 = \{G, L\}$, $B_1 = \{W, L\}$ and $B_2 = \{L, P_1, P_2, \ldots, P_n\}$
- **Causal width is** $n+1$ (n is number of cells)

Example: Better Model for Wumpus (Causal Width = 5)





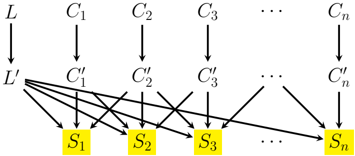
- $n+3$ vars: G (gold), W (wumpus), L (agent), P_1 (pit@1), $\ldots$, P_n (pit@n)
- $n+2$ obs vars: T (glitter), S (stench), Z_1 (breeze@1), $\ldots$, Z_n (breeze@n)
- $n+2$ beams: $B_0 = \{G, L\}$, $B_1 = \{W, L\}$, $B_{1+i} = \text{parents}(Z_i)$

$$P(Z_i|\text{parents}(Z_i)) \;=\; \begin{cases} 1/2 & L \neq i \\ \text{“model”} & L = i \end{cases}$$

$$P(\bar{Z}|L, \bar{P}) \;=\; \textstyle\prod_{i=1,\ldots,n} P(Z_i|\text{parents}(Z_i))$$

- **Causal width is** 5 (bounded, independent of number of cells n)

Example: 1-Line-3 SLAM (Causal Width = 4)



- $n+1$ state vars: L (agent), C_1 (cell@1), ..., C_n (cell@n)
- n obs vars: S_1 (sensed@1), ..., S_n (sensed@n)
- n beams: $B_1 = \{L, C_1, C_2\}$, $B_2 = \{L, C_1, C_2, C_3\} \ldots B_n = \{L, C_{n-1}, C_n\}$
- **Causal width is** 4 (bounded, independent of number of cells n)
- **Unlike Wumpus:** agent moves **stochastically** and its **location isn't known or observable** (initially at leftmost cell)
- **Unlike Color SLAM:** observation at cell i depends on colors of cell i and **surrounding cells**

Decomposable Models: Definition + Theoretical Results

- A state variable is **external** if it appears in more than one beam
- A state variable X is **backward deterministic (BD)** if, for all time steps t, its value x_t at time t is **determined** by:
  - Its value x_{t+1} at time $t+1$
  - The action at time t
  - The history of actions/observation up to time $t-1$
  - The prior b_0
- A model is **decomposable** if all external variables are BD

Theorem

If model is decomposable, the joint at time t **factorizes** *as product of* **factors**, *one for each beam, where each factor is* **independently updated**. *All factors updated in time/space* **exponential in causal width**

Examples of Decomposable Models

- **Wumpus** is **decomposable**
  - Only external variable is agent's location that is backward deterministic (It is BD since initial location is known and actions are deterministic)
  - Causal width is 5

- **1-Line-3 SLAM** is **non-decomposable**
  - Agent's location is external and non-BD because location isn't known or observable, and actions are stochastic
  - Causal width is 4

- **Minesweeper** is **decomposable**
  - All variables are **static** and thus backward deterministic
  - Causal width is 9

Decomposable Models and Factored Beliefs

Joint in decomposable models can be **tracked exactly** in polytime when causal width is bounded (because of **polysize** factors)

Doesn't imply that marginals over joint can be answered in polytime

Complexity of queries depend on the **treewidth** associated with the beam structure:

- E.g. if beam structure is **"tree"**, marginals can be computed in polytime (for bounded causal width) at every time step
- Otherwise, **belief propagation** can be used to approximate marginals

Sampling: Making Non-Decomposable Models Decomposable

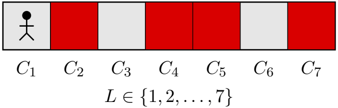
Non-decomposable models tackled by **sampling** non-BD external vars

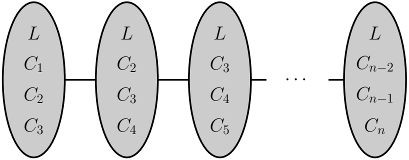
Such variables become **BD given their sampled history**

Sampling done for making the model **decomposable**, not for making it **tractable** as in Rao-Blackwellized PFs

This form of sampling generalizes idea in SLAM algorithms where cells (or landmarks) are independent given observations and (sampled) history of agent's location

Example: 1-Line-3 SLAM (Causal Width = 4)





Decomposition of beam structure at any time point (treewidth 3)

- **Sample** agent's location to make model **decomposable**
- Cell colors **not independent** of each other given sampled agent's location, but factorization has **treewidth of 3**
- Exact marginals can be computed in polytime (e.g. using **join-tree algorithm**) given sampled history of agent's location

Technical Details in Paper

Belief expressed as **product of factors** (one factor per beam):

$$Bel^h(x) \;=\; Bel^h(X_t = x) \;=\; \prod_j B_j^h(x_j)$$

where x_j is valuation over beam B_j, and $B_j(\cdot)$ is factor for B_j

Each factor B_j is **tracked independently**. For history $h' = \langle h, a, o \rangle$:

$$B_j^{h'}(y_j', z_j') \;\propto\; q_j(o_j | y_j', z_j', a) \sum_{y_j'} tr_j(x_j' | x_j, z_j^*, a)\, B_j^h(y_j, z_j^*)$$

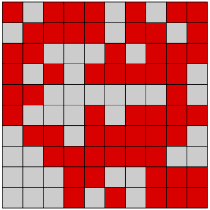
where Y_j/Z_j are internal/external vars in B_j, q_j and tr_j are sensor and transitions in 2-DBN, and $z_j^* = \mathcal{R}_a(z_j' | h)$ is the **regression** of the value z_j' for Z_j given last action a and history h (as Z is BD)

Experiments in Paper

- **1-Line-3 SLAM:** sizes with 64 and 512 cells, different algorithms for computing marginals (JT, BP, AC)

- **Minesweeper:** sizes 6×6, 8×8, 16×16 and 30×16, different algorithms for computing marginals

- **Minemapping:**
  - Agent moves **stochastically** in grid 6×6 or 10×10
  - **Noisy sensing** is integer in $\{0, 1, \ldots, 9\}$ telling how many cells of the 9 cells around are red
  - Causal width is 9
  - **Non-decomposable** so sampling of agent's location
  - Factorization has **unbounded treewidth**



10×10

See results and analyses in paper!

Probabilistic Belief Tracking: Summary

- **General formulation** and algorithm determined by structure
- Joint maintained in factored form in polytime when **causal width** is bounded and external variables are **backward deterministic (BD)**
- If bounded causal width and **beam structure** has bounded treewidth, marginals computed exactly in polytime; else approximated by belief propagation
- Non-BD vars appearing in more than one beam are **sampled**
- Sampling done for making such variables BD, not for making inference tractable
- Need to speed up computation of marginal further to make scheme **sufficiently practical**

Differences with Boyen-Koller and Factored Frontier

Boyen-Koller:

- Joint decomposed as product of marginals over **clusters** of variables
- Progression of decomposition requires **exact inference**
- Clusters are not required to be **causally closed**
- Variables appearing in more than one cluster not required to be BD

Factored frontier like BK but:

- Joint decomposed as product of marginals over **variables**
- Efficient progression of decomposition

Our probabilistic beam tracking:

- Beams (clusters) and sampling (if necessary) determined by 2-DBN and BD
- Progression of beams exponential in causal width
- Computation of marginals required for **query answering** (intractable if exact)
- Exact algorithm (if BD) or (statistically) consistent as #particles increase

Challenges Ahead

- Tracking of beam factors **across time** exponential in causal width, but linear in time and number of samples (when sampling needed)
  - This doesn't appear to be a problem, as causal width is usually bounded and small
  - Bottleneck is **computation of marginals** from factors at time t
  - Approximation by belief propagation not always good or fast
  - Need faster and scalable approximate inference algorithms for computing marginals over factor models
- Address problems with large or unbounded causal width