COMP 64101
Reasoning and Learning under Uncertainty

Omar Rivasplata

Department of Computer Science, University of Manchester
Manchester Centre for AI Fundamentals

Lecture 3 - Part 3

Topics: Message passing algorithms

I know it's [Tuesday]. It's a good day for math!

–Inspired by Max Mintz, UPenn.

- Intro (§ 9.1)
- Belief propagation on chains (§ 9.2)
- Belief propagation on trees (§ 9.3)
- More goodies on this (§ 9.4 – 9.7)

Nota bene: Section numbers refer to Murphy (2023).

9.1 Introduction

Message passing algorithms

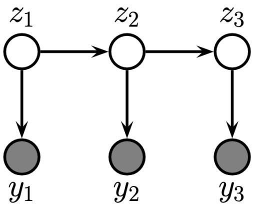
- For PGMs with a *sparse* graph structure.
- Levegage the CI properties encoded by the graph to perform efficient posterior inference: computing marginals, mode(s), sampling etc.
- Based on the principle of **dynamic programming** (DP):
  - Find global solution by finding local solutions to sub-problems.
  - Combine local optima to get global optima.
- Implementation idea:
  - Maintain probability distributions (beliefs) on the value of each node.
  - Update these beliefs given evidence from some part of the graph.
  - Pass beliefs on a node (or clique) to neighbouring nodes (or cliques).
- Note: messages to be passed are nothing else than the beliefs.
- These algorithms are a.k.a. **belief propagation** (BP) algorithms.

To do: Look up and learn more about the principle of DP.

9.2 Belief propagation on chains

- Easy start: Graph structure is a 1D chain.
- The idea of a Markov chain could be helpful to some extent, but the graph structure might not look exactly like a classical Markov chain.
- Keep in mind that there are hidden variables and observed variables.
- Further simplifications:
  - Directed PGMs.
  - Hidden variables are discrete.
- Methods extend to continuous latent variables, Undirected PGMs.

- Variables:
  - $\mathbf{z}_t$ are hidden (a.k.a. latent),
  - $\mathbf{y}_t$ are observations or outputs.
- HMMs are latent variable sequence models.

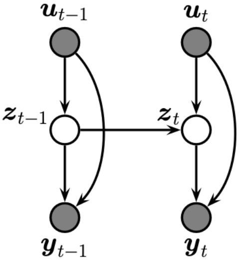


- CI properties given by the chain-structured graph (see figure).
- Joint distribution:

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = \left[p(\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t | \mathbf{z}_{t-1}) \right] \left[\prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{z}_t) \right]$$

Compare: General setting of State Space Models (SSMs)

- Variables:
  - $\mathbf{z}_t$ are hidden (a.k.a. latent),
  - $\mathbf{y}_t$ are observations or outputs,
  - $\mathbf{u}_t$ are optional inputs.
- "Latent variable sequence models."



- CI properties given by the chain-structured graph (see figure).
- Joint distribution:

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T} | \mathbf{u}_{1:T}) = \left[p(\mathbf{z}_1 | \mathbf{u}_1) \prod_{t=2}^{T} p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{u}_t) \right] \left[\prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{z}_t, \mathbf{u}_t) \right]$$

9.2.1.1 Example: casino HMM

- **To do:** Read this example (Murphy (2023), pp. 402 - 403).

9.2.1.2 Posterior inference

- Posterior inference on the hidden states.
- Goal: Compute $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ for $t = 1, \ldots, T$.
- Note: Computation given on whole stream of observations $\mathbf{y}_{1:T}$
- This is an instance of smoothing!
- Forwards Filtering Backwards Smoothing (FFBS).

9.2.1.2 Posterior inference (cont'd)

- Case of discrete hidden variables:

$$
\begin{aligned}
p(\mathbf{z}_t = j|\mathbf{y}_{1:T}) &= p(\mathbf{z}_t = j|\mathbf{y}_{1:t}\mathbf{y}_{t+1:T}) \\
&= p(\mathbf{z}_t = j, \mathbf{y}_{t+1:T}|\mathbf{y}_{1:t})/p(\mathbf{y}_{t+1:T}|\mathbf{y}_{1:t}) \\
&\propto p(\mathbf{z}_t = j, \mathbf{y}_{t+1:T}|\mathbf{y}_{1:t}) \\
&= p(\mathbf{z}_t = j|\mathbf{y}_{1:t})p(\mathbf{y}_{t+1:T}|\mathbf{z}_t = j, \mathbf{y}_{1:t}) \\
&= p(\mathbf{z}_t = j|\mathbf{y}_{1:t})p(\mathbf{y}_{t+1:T}|\mathbf{z}_t = j)
\end{aligned}
$$

- Idea: Compute the factors separately, then combine.
  - Forwards pass: Compute $p(\mathbf{z}_t = j|\mathbf{y}_{1:t})$.
  - Backwards pass: Compute $p(\mathbf{y}_{t+1:T}|\mathbf{z}_t = j)$.

- **Food for thought:** What if continuous hidden variables?

More on BP on chains

- 9.2.2 The forwards algorithm
  - **To do:** Read about this (Murphy (2023), pp. 403 - 404).
- 9.2.3 The forwards-backwards algorithm
  - **To do:** Read about this (Murphy (2023), pp. 404 - 407).
    - 9.2.3.1 Backwards recursion
    - 9.2.3.2 Example
    - 9.2.3.3 Two-slice smoothed marginals
    - 9.2.3.4 Numerically stable implementation
- 9.2.4 Forwards filtering backwards smoothing
  - **To do:** Read about this (Murphy (2023), pp. 407 - 408).
- 9.2.5 Time and space complexity
  - **To do:** Read about this (Murphy (2023), pp. 408 - 409).

More on BP on chains (cont'd)

- 9.2.6 The Viterbi algorithm
  - **To do:** Read about this (Murphy (2023), pp. 409 - 412).
    - 9.2.6.1 Forwards pass
    - 9.2.6.2 Backwards pass
    - 9.2.6.3 Example
    - 9.2.6.4 Time and space complexity
    - 9.2.6.5 N-best list
- 9.2.7 Forwards filtering backwards sampling
  - **To do:** Read about this (Murphy (2023), p. 412).

9.3 Belief propagation on trees

- Trees are special kinds of graph structures.
- Message passing algorithms, extended to trees.
- Build on ideas from:
  - Algorithms for HMMs (§ 9.2.3)
  - Algorithms for Kalman smoothing (§ 8.2.3)

Goodies on BP on trees

- 9.3.1 Directed vs undirected trees
  - **To do:** Read about this (Murphy (2023), pp. 412 - 414).
- 9.3.2 Sum-product algorithm
  - **To do:** Read about this (Murphy (2023), pp. 414 - 415).
- 9.3.3 Max-product algorithm
  - **To do:** Read about this (Murphy (2023), pp. 415 - 417).
    - 9.3.3.1 Connection between MMM and MAP
    - 9.3.3.2 Connection between MPM and MAP
    - 9.3.3.3 Connection between MPE and MAP

More goodies on BP on trees

- 9.4 Loopy belief propagation
  - **To do:** Read about this (Murphy (2023), pp. 417 - 428).
- 9.5 The variable elimination (VE) algorithm
  - **To do:** Read about this (Murphy (2023), pp. 428 - 434).
- 9.6 The junction tree algorithm (JTA)
  - **To do:** Read about this (Murphy (2023), p. 434).
- 9.7 Inference as optimization
  - **To do:** Read about this (Murphy (2023), pp. 435 - 437).

References

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.