Graphical models and Hidden Markov Models

Dr. Francis Colas

7.10.2011

Introduction

Probabilistic reasoning:

- transform uncertainty into probabilities,
- specify the joint probability distribution,
- generic form for the inference;

But:

- complexity of the joint probability distribution,
- need independence or conditional independence.

Example

Let's have:

$$P(X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \mathbf{\Omega}, \mathbf{\Phi}_0, \mathbf{\Phi}_1, \mathbf{\Phi}_2)$$

With 10 cases for each dimension:

$$10^{20} - 1 = 9,999,999,999,999,999,999$$

With recursive application of Bayes' rule:

$$
\begin{aligned}
= \quad & P(X)P(Y|X)P(V_x|X, Y)P(V_z|X, Y, V_x)P(R_y|X, Y, V_x, V_z) \\
\times \quad & P(\Lambda|X, Y, V_x, V_z, R_y)P(\mathbf{T}|X, Y, V_x, V_z, R_y, \Lambda)P(\mathbf{\Omega}|X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}) \\
\times \quad & P(\mathbf{\Phi}_0|X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \mathbf{\Omega})P(\mathbf{\Phi}_1|X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \mathbf{\Omega}, \mathbf{\Phi}_0) \\
\times \quad & P(\mathbf{\Phi}_2|X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \mathbf{\Omega}, \mathbf{\Phi}_0, \mathbf{\Phi}_1)
\end{aligned}
$$

Space complexity:

$$
\begin{aligned}
& (10-1) + (10-1) * 10 + (10-1) * 10^2 + (10-1) * 10^3 + (10-1) * 10^4 \\
+ \quad & (10-1) * 10^5 + (10^3 - 1) * 10^6 + (10^3 - 1) * 10^9 \\
+ \quad & (10^2 - 1) * 10^{12} + (10^4 - 1) * 10^{14} \\
+ \quad & (10^2 - 1) * 10^{18} \\
= \quad & 9,999,999,999,999,999,999
\end{aligned}
$$

Adding conditional independence assumptions

Let's assume:

$$
\begin{aligned}
& P(X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \boldsymbol{\Omega}, \boldsymbol{\Phi}_0, \boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2) \\
= \quad & P(X)P(Y|X)P(V_x)P(V_z)P(R_y) \\
\times \quad & P(\Lambda)P(\mathbf{T}|V_x, V_z, R_y)P(\boldsymbol{\Omega}|V_x, V_z, R_y) \\
\times \quad & P(\boldsymbol{\Phi}_0|\mathbf{T}, \boldsymbol{\Omega})P(\boldsymbol{\Phi}_1|X, Y, \mathbf{T}, \boldsymbol{\Omega}) \\
\times \quad & P(\boldsymbol{\Phi}_2|X, Y, \Lambda, \mathbf{T}, \boldsymbol{\Omega})
\end{aligned}
$$

Space complexity:

$$
\begin{aligned}
& (10-1) + (10-1) * 10 + (10-1) + (10-1) + (10-1) \\
+ \quad & (10-1) + (10^3-1) * 10^3 + (10^3-1) * 10^3 \\
+ \quad & (10^2-1) * 10^6 + (10^4-1) * 10^8 \\
+ \quad & (10^2-1) * 10^9 \\
= \quad & 1,099,000,998,135 \\
\ll \quad & 9,999,999,999,999,999,999
\end{aligned}
$$

Structure

Probabilistic reasoning:

- specification of the joint distribution,
- using independence assumptions,
- structure of the model;

But:

- algebraic formulation,
- need for a graphical representation.

Graphical models

Aim:

- diagrammatic representation of a joint probability distribution,
- represent the dependency structure,
- nodes to represent variables,
- edges to represent dependency;

Different forms:

- Bayesian networks (belief network): directed acyclic graph,
- Markov random fields (Markov network): undirected graph,
- factor graph: undirected bipartite graph,
- chain graph: directed and undirected without directed cycles,
- ...

Why different forms?

Using graphical models:

- which probabilistic model for a given graph?
- which graph for a given probabilistic model?
- are there models that cannot be represented in a graph?

Issue:

- some probabilistic relationships may not be represented by some kinds of graphs,
- different kind of graphs can represent different kind of relationship,
- standard graphical representation don't represent all,
- but still useful.

Bayesian neworks

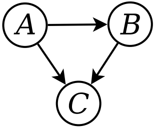
Definition:

- nodes for variables,
- edges for dependencies,
- directed acyclic graph;

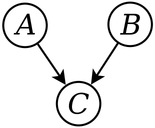
Example:

Joint $P(A, B, C)$

Bayes' rule $P(A)P(B|A)P(C|A, B)$



Cond. ind. $P(A)P(B)P(C|A, B)$



Bayesian network

Relationship between a Bayesian network and probabilities:

$$P(V_1, V_2, \ldots, V_n) = \prod_{i=1}^{n} P(V_i | Pa(V_i)),$$

where $Pa(V_i)$ is the set of parents of V_i.
This implies:

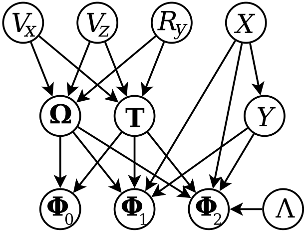
- for the graph:
  - directed edges (to have parents),
  - no directed loop (iterated Bayes' rule);
- for the joint:
  - only one variable on the left.

More complex example

Algebraic formulation:

$$
\begin{aligned}
& P(X, Y, V_x, V_z, R_y, \Lambda, \mathbf{T}, \mathbf{\Omega}, \mathbf{\Phi}_0, \mathbf{\Phi}_1, \mathbf{\Phi}_2) \\
= \quad & P(X)P(Y|X)P(V_x)P(V_z)P(R_y) \\
\times \quad & P(\Lambda)P(\mathbf{T}|V_x, V_z, R_y)P(\mathbf{\Omega}|V_x, V_z, R_y) \\
\times \quad & P(\mathbf{\Phi}_0|\mathbf{T}, \mathbf{\Omega})P(\mathbf{\Phi}_1|X, Y, \mathbf{T}, \mathbf{\Omega}) \\
\times \quad & P(\mathbf{\Phi}_2|X, Y, \Lambda, \mathbf{T}, \mathbf{\Omega})
\end{aligned}
$$

Graphical representation:



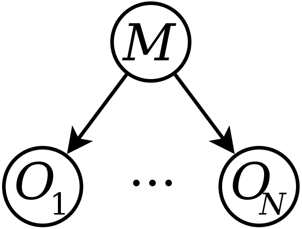
Additional elements

Plate:

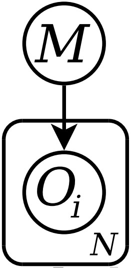
- series of variables with equal dependencies:

For example:

$$P(M, O_1, \cdots, O_N) = P(M) \prod_{i=1}^{N} P(O_i|M)$$



can be drawn:



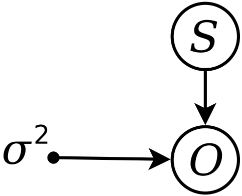
Additional elements

Hyperparameters:

- probability distribution which depends on explicit parameters:

For example:

$$P(S, O|\sigma^2) = P(S)P(O|S, \sigma^2)$$



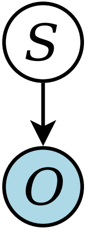
Additional elements

Observed variables:

- signaling which variables are observed:

For example:

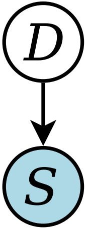
$$P(S|O) \propto P(S)P(O|S)$$



Examples

Back to the doors:

$$P(D|S) \propto P(D)P(S|D)$$

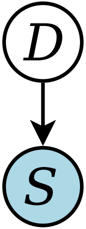


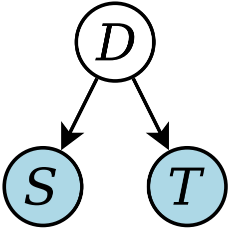
Examples

Back to the doors:

$$P(D|S) \propto P(D)P(S|D)$$

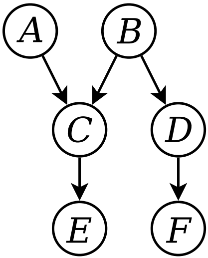
$$P(D|S,T) \propto P(D)P(S|D)P(T|D)$$





Independence in Bayesian networks

$$
\begin{aligned}
& P(A, B, C, D, E) \\
= \quad & P(A)P(B)P(C|A, B) \\
\times \quad & P(D|B)P(E|C)P(F|D)
\end{aligned}
$$



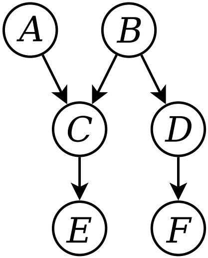
Assumptions:

- $A \perp\!\!\!\perp B$,
- $D \perp\!\!\!\perp A, C \mid B$,
- $E \perp\!\!\!\perp A, B, D \mid C$,
- $F \perp\!\!\!\perp A, B, C, E \mid D$;

Autonomous Systems Lab

Independence in Bayesian networks

$$
\begin{aligned}
& P(A, B, C, D, E) \\
= \quad & P(A)P(B)P(C|A, B) \\
\times \quad & P(D|B)P(E|C)P(F|D)
\end{aligned}
$$



Assumptions:

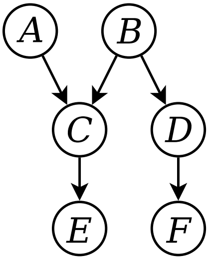
- $A \perp\!\!\!\perp B$,
- $D \perp\!\!\!\perp A, C \mid B$,
- $E \perp\!\!\!\perp A, B, D \mid C$,
- $F \perp\!\!\!\perp A, B, C, E \mid D$;

We have, for example:

- $F \perp\!\!\!\perp B \mid D$,
- $E \perp\!\!\!\perp F \mid B$,
- $A \perp\!\!\!\perp B \mid F$,
- $A \perp\!\!\!\perp D$,
- ...

Independence in Bayesian networks

$$
\begin{aligned}
& P(A, B, C, D, E) \\
= \; & P(A)P(B)P(C|A, B) \\
\times \; & P(D|B)P(E|C)P(F|D)
\end{aligned}
$$



Assumptions:

- $A \perp\!\!\!\perp B$,
- $D \perp\!\!\!\perp A, C \mid B$,
- $E \perp\!\!\!\perp A, B, D \mid C$,
- $F \perp\!\!\!\perp A, B, C, E \mid D$;

But not:

- $A \perp\!\!\!\perp B \mid C$,
- $F \perp\!\!\!\perp B \mid E$,
- $C \perp\!\!\!\perp D \mid E$,
- ...

d-separation

In a graph:

- S_1, S_2, S_3 non intersecting subsets of nodes;
- a path from S_1 to S_2 is blocked by S_3 if it contains a node such that either:
  - the node is in S_3 and is head-to-tail or tail-to-tail,
  - or the node is head-to-head and neither the node or its descendants are in S_3;
- if all paths between S_1 and S_2 are blocked by S_3 then S_1 and S_2 are d-separated by S_3;

For a Bayesian network:

- d-separation $\Leftrightarrow$ conditional independence of associated model.

Not true for arbitrary graphs and models.

Markov random fields

Bayesian networks:

- partial ordering between all variables,
- d-separation to indicate (cond.) independence,
- great in a lot of cases;

What with: pixels of a camera?

- pixels of a camera?
- cells in space?
- ...

Markov random fields

Markov random field:

- undirected graphical model;

d-separation and independence:

- no head-to-head issue,
- a path is blocked by S_3 if it contains a node in S_3,
- Markov blanket: set of neighbors;

Joint probability:

- not using Bayes' rule,
- product of potential functions over cliques.

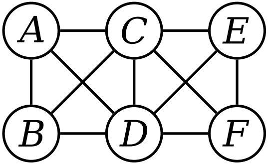
Joint probability distribution

Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.



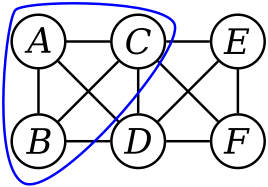
Joint probability distribution

Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

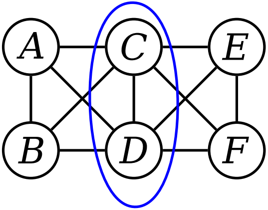


Joint probability distribution

Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.
Clique: set of nodes all connected to each other.



Joint probability distribution

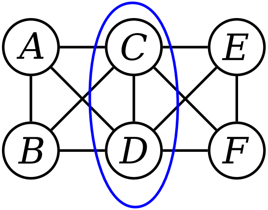
Factorization:

$$P(V_1, V_2, \dots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



Joint probability distribution

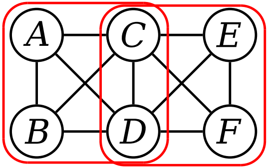
Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



ETH Zürich

Joint probability distribution

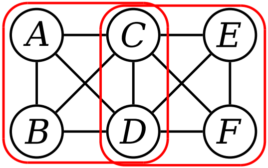
Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



$$P(A, B, C, D, E, F) = \frac{1}{Z} \phi_{ABCD}(A, B, C, D) \phi_{CDEF}(C, D, E, F)$$

Joint probability distribution

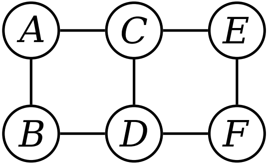
Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



Joint probability distribution

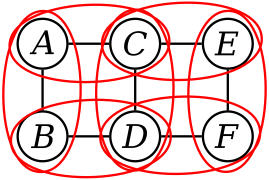
Factorization:

$$P(V_1, V_2, \dots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



Joint probability distribution

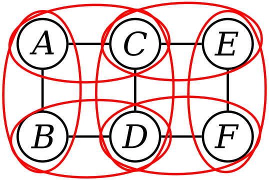
Factorization:

$$P(V_1, V_2, \ldots, V_n) = \frac{\prod_C \phi_C(\mathbf{V}_C)}{\sum_{\mathbf{V}'} \prod_C \phi_C(\mathbf{V}'_C)} = \frac{1}{Z} \prod_C \phi_C(\mathbf{V}_C)$$

where $\mathbf{V}_C$ are the variables in each of the maximal cliques C and ϕ_C the potential function of C.

Clique: set of nodes all connected to each other.

Maximal clique: clique not contained into another clique.



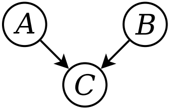
$$P(A, B, C, D, E, F) = \frac{1}{Z} \phi_{AB}(A, B)\phi_{AC}(A, C)\phi_{BD}(B, D)\phi_{CD}(C, D)\phi_{CE}(C, E)\phi_{DF}(D, F)\phi_{EF}(E, F)$$

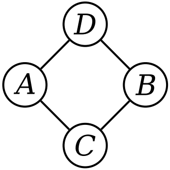
Expressivity

Some models can be perfectly expressed by:

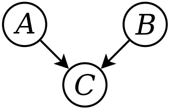
- Bayesian networks,
- Markov random fields,
- both,
- none;

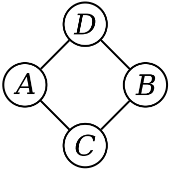
Expressivity





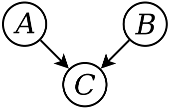
Expressivity



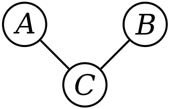


$A \not\perp B \mid C$

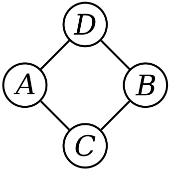
Expressivity



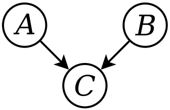
$A \not\perp\!\!\!\perp B \mid C$



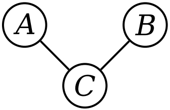
$A \perp\!\!\!\perp B \mid C$



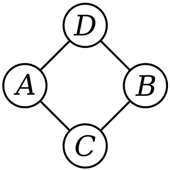
Expressivity



$A \not\perp\!\!\!\perp B \mid C$

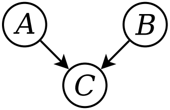


$A \perp\!\!\!\perp B \mid C$

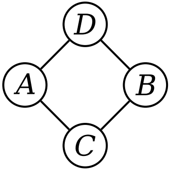


$A \perp\!\!\!\perp B \mid C, D$

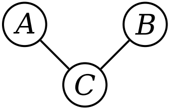
Expressivity



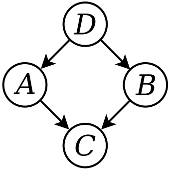
$A \not\perp\!\!\!\perp B \mid C$



$A \perp\!\!\!\perp B \mid C, D$



$A \perp\!\!\!\perp B \mid C$



$A \not\perp\!\!\!\perp B \mid C, D$

Inference

Inference in graphical models:

- similar to algebraic representation:
- summation over free variables,
- exploit independence,
- rearrange sums;

On trees:

- message passing,
- sum-product algorithm,
- belief propagation;

On general graphs:

- junction tree (exact but can be slow),
- loopy belief propagation (approximate).

Summary on graphical models

Graphical models:

- graphical representation of probabilistic models,
- represent dependencies,
- different types,
- same inference problems;

Bayesian networks:

- directed acyclic graphs,
- direct link with Bayes' rule;

Markov random fields:

- undirected graphs,
- factorization using potentials on cliques.

Time

So far:

- probabilistic models,
- graphical representation,
- inference on variables;

What about:

- data series,
- time,
- ...

Dynamic Bayesian networks

Often you need to:

- take change into account,
- have variables whose value change with time,
- specify that relations are similar whichever instant you consider;

Solution:

- one variable per instant:

$$P(S^0, D^0, S^1, D^1, S^2, D^2, \ldots, S^T, D^T)$$

But:

- specify huge joint distribution,
- inference by summing over many variables.

Markov assumption

Reduce dependency using Markov assumption:

- distribution over a state at time t is independent of former timesteps given the state at $t-1$.

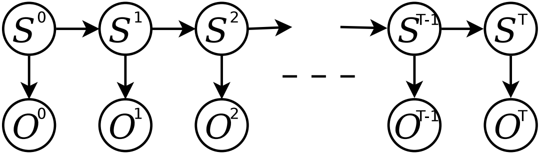
(Markov random fields have a similar property but with their neighbors: the Markov blanket.)

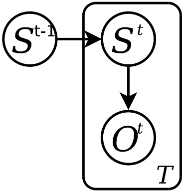
Hidden Markov Model

H.M.M.:

- hidden: state are not observed directly,
- Markov: order-1 Markov assumption,
- discrete variables;

$$P(S^0, O^0, \ldots, S^T, O^t) = P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)$$





Hidden Markov Model

$$P(S^0, O^0, \dots, S^T, O^t) = P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)$$

You need:

- $P(S^0)$: prior π_0,
- $\forall t, P(S^t|S^{t-1})$: transition matrix A^t (constant for homogeneous HMMs: A),
- $\forall t, P(O^t|S^t)$: observation matrix B^t (constant for homogeneous HMMs: B):
- parameters $\theta = (\pi, A, B)$.

Hidden Markov Models

You can:

- distribution over current state based on all observations: $P(S^T|O^0,\ldots,O^T,\theta)$: forward algorithm,
- probability value of a given observation or a series of observations: $P(O^T|\theta)$, $P(O^0,\ldots,O^T|\theta)$: forward algorithm,
- probability distribution over a state given past and future observation (smoothing): $P(S^t|O^0,\ldots,O^T)$: forward-backward algorithm,
- most likely state sequence: $\arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T|O^0,\ldots,O^T,\theta)$: Viterbi algorithm,
- learning parameters θ based on an observation sequence: Baum-Welch algorithm,
- ...

Iterative formulation

Distribution over the last state:

$$P(S^T|O^0,\dots,O^T)$$
$$= \frac{\sum_{S_0,\dots,S^{T-1}} P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)}{\sum_{S_0,\dots,S^{T}} P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)}$$

Huge complexity: $O(N^T T)$ but...

Iterative formulation

Distribution over the last state:

$$
\begin{aligned}
& P(S^T|O^0,\dots,O^T) \\
= \quad & \frac{\sum_{S_0,\dots,S^{T-1}} P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)}{\sum_{S_0,\dots,S^{T}} P(S^0)P(O^0|S^0)\prod_{t=1}^{T} P(S^t|S^{t-1})P(O^t|S^t)}
\end{aligned}
$$

Huge complexity: $O(N^T T)$ but...
Iterative expression:

$$
\begin{aligned}
& P(S^T|O^0,\dots,O^T) \\
\propto \quad & P(O^T|S^T)P(S^T|O^0,\dots,O^{T-1}) \\
\propto \quad & P(O^T|S^T)\sum_{S^{T-1}} P(S^T|S^{T-1})P(S^{T-1}|O^0,\dots,O^{T-1})
\end{aligned}
$$

Same result but only $O(N^2 T)$.

Forward algorithm

Let's define:

$$\alpha(S^t) = P(S^t, O^0, \ldots, O^t)$$

We have:

$$\alpha(S^{t+1}) = P(O^{t+1}|S^{t+1}) \sum_{S^t} P(S^{t+1}|S^t)\alpha(S^t)$$

Forward algorithm

Let's define:

$$\alpha(S^t) = P(S^t, O^0, \ldots, O^t)$$

We have:

$$\alpha(S^{t+1}) = P(O^{t+1}|S^{t+1}) \sum_{S^t} P(S^{t+1}|S^t)\alpha(S^t)$$

And:

$$P(S^t|O^0, \ldots, O^t) \propto \alpha(S^t)$$

Forward algorithm

Let's define:

$$\alpha(S^t) = P(S^t, O^0, \ldots, O^t)$$

We have:

$$\alpha(S^{t+1}) = P(O^{t+1}|S^{t+1}) \sum_{S^t} P(S^{t+1}|S^t)\alpha(S^t)$$

And:

$$P(S^t|O^0, \ldots, O^t) \propto \alpha(S^t)$$

And also:

$$P(O^0, \ldots, O^t) = \sum_{S_t} \alpha(S^t)$$

Forward-backward algorithm

Let's define:

$$\beta(S^t) = P(O^{t+1}, \ldots, O^T | S^t)$$

We have similarly:

$$\beta(S^t) = \sum_{S^{t+1}} P(O^{t+1}|S^{t+1}) P(S^{t+1}|S^t) \beta(S^{t+1})$$

Forward-backward algorithm

Let's define:

$$\beta(S^t) = P(O^{t+1}, \ldots, O^T | S^t)$$

We have similarly:

$$\beta(S^t) = \sum_{S^{t+1}} P(O^{t+1}|S^{t+1}) P(S^{t+1}|S^t) \beta(S^{t+1})$$

Then, smoothing:

$$\begin{aligned} & P(S^t|O^0, \ldots, O^T) \\ \propto \quad & P(O^{t+1}, \ldots, O^T|S^t) P(S^t|O^0, \ldots, O^t) \\ \propto \quad & \beta(S^t)\alpha(S^t) \end{aligned}$$

Viterbi algorithm

Most probable sequence of states given observations:

$$
\begin{aligned}
& \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T \mid O^0,\ldots,O^T,\theta) \\
= \quad & \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T,O^0,\ldots,O^T,\theta)
\end{aligned}
$$

Viterbi algorithm

Most probable sequence of states given observations:

$$
\begin{aligned}
& \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T|O^0,\ldots,O^T,\theta) \\
= & \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T,O^0,\ldots,O^T,\theta)
\end{aligned}
$$

Let:

$$\delta(S^t) = \max_{S^0,\ldots,S^{t-1}} P(S^0,\ldots,S^t,O^0,\ldots,O^t)$$

then:

$$\delta(S^{t+1}) = P(O^{t+1}|S^{t+1}) \max_{S^t} P(S^{t+1}|S^t)\delta(S^t)$$

Viterbi algorithm

Most probable sequence of states given observations:

$$\begin{aligned} & \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T | O^0,\ldots,O^T,\theta) \\ = \quad & \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T, O^0,\ldots,O^T,\theta) \end{aligned}$$

Let:

$$\delta(S^t) = \max_{S^0,\ldots,S^{t-1}} P(S^0,\ldots,S^t, O^0,\ldots,O^t)$$

then:

$$\delta(S^{t+1}) = P(O^{t+1}|S^{t+1}) \max_{S^t} P(S^{t+1}|S^t)\delta(S^t)$$

Same as α but with max instead of $\sum$.

Viterbi algorithm

Most probable sequence of states given observations:

$$
\begin{aligned}
& \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T|O^0,\ldots,O^T,\theta) \\
= \quad & \arg\max_{S^0,\ldots,S^T} P(S^0,\ldots,S^T,O^0,\ldots,O^T,\theta)
\end{aligned}
$$

Let:

$$\delta(S^t) = \max_{S^0,\ldots,S^{t-1}} P(S^0,\ldots,S^t,O^0,\ldots,O^t)$$

then:

$$\delta(S^{t+1}) = P(O^{t+1}|S^{t+1}) \max_{S^t} P(S^{t+1}|S^t)\delta(S^t)$$

For the states:

$$\psi(S^t) = \arg\max_{S^{t-1}} P(S^t|S^{t-1})\delta(S^{t-1})$$

that allows backtracking.

Parameter estimation

Previous algorithms require parameters $\theta = (\pi, A, B)$. Where:

- π prior probability over the state: $P(S^0)$,
- A transition matrix: $P(S^{t+1}|S^t)$,
- B observation matrix: $P(O^t|S^t)$;

Can we get parameters from a sequence of observations?

$$\arg\max_{\theta} P(O^0, \dots, O^T|\theta)$$

- not directly (no closed form solution),
- iterative "hill climbing" approximation,
- Baum-Welch algorithm.

Baum-Welch algorithm

Basic idea:

- take some parameters θ^{old},
- compute the distribution over state sequences,
- compute new parameters θ based on this distribution,
- loop taking the new parameters;

Baum-Welch algorithm

Basic idea:

- take some parameters θ^{old},
- compute the distribution over state sequences,
- compute new parameters θ based on this distribution,
- loop taking the new parameters;

Each time: $P(O^0, \dots, O^T|\theta) > P(O^0, \dots, O^T|\theta^{old})$.

Baum-Welch algorithm

Basic idea:

- take some parameters θ^{old},
- compute the distribution over state sequences,
- compute new parameters θ based on this distribution,
- loop taking the new parameters;

More details:

- take some parameters θ^{old},
- (E) compute:

$$Q(\theta, \theta^{old}) = \sum_{S^0,\ldots,S^T} \log\left((P(S^0,\ldots,S^T,O^0,\ldots,O^T|\theta)\right) P(S^0,\ldots,S^T|O^0,\ldots,O^T,\theta^{old})$$

- (M) optimize $Q(\theta, \theta^{old})$ to get the new θ,
- loop.

Summary on H.M.M.

Aims:

- time series,
- discrete variables,
- several uses:
  - probability of an observation, a sequence of observations,
  - probability of a state after several observations,
  - smoothing (state in the middle of observations),
  - most likely sequence,
  - most likely parameters;

Algorithms:

- forward: iterative inference in Bayesian filters,
- forward backward: similar to message passing in chains or trees,
- Viterbi: max-product,
- Baum-Welch: specific case of Expectation-Maximization (class 11).

Summary

Graphical models:

- graphical representation of dependencies,
- Bayesian networks (directed acyclic graphs): follow Bayes' rule, difficult independence,
- Markov random fields (undirected graphs): easy independence, potential functions instead of (cond.) probability distributions;

H.M.M.:

- time series,
- discrete variables,
- inference algorithms: simpler versions than on general models.