Intelligent Probabilistic Graphical Models

Olivier Croissant

1 General Principle

Probabilistic graphical models are the natural evolution of 1980s expert systems where we sought to give inference mechanisms and their associated certainty coefficients a solid, completely mathematical foundation – that is, a probabilistic one.

The problem is that inference mechanisms now rely on likelihood optimizations that are very computationally expensive when not infeasible.

We are therefore forced to perform these optimizations on a small number of data and with small Bayesian networks.

It then seems natural to try to design systems that start from existing solutions and attempt to improve these solutions without starting from scratch each time a new network is tried.

The idea is even to automatically search for incremental structural modifications to improve system efficiency without questioning what has been acquired.

Among these structural evolutions, enriching the information considered by the system holds a prominent place.

2 EM Algorithm

The E-step of the EM algorithm is meant to handle the learning of the hidden variable chain by computing the Expectation of transitions. This comes from the fact that if we maximize the log-likelihood assuming known emission probabilities (thus the associated parameters), we are reduced to maximizing quantities like $\sum_{i=1}^{n} a_i \log(\delta_i)$. This indeed gives us that the optimal chain parameters are obtained by computing transition expectations. Hence the name given to this E-step.

3 Basic Mechanisms

Theorem 1

We want to maximize $\varphi(\delta_i|_{i=1,n}) = \sum_{i=1}^{n} a_i \log(\delta_i)$ under the constraint:

$$\sum_{i=1}^{n} \delta_i = E$$

Then the result is:

We introduce Lagrange multipliers: maximize $\sum_{i=1}^{n} a_i \log(\delta_i) - \lambda \left(\sum_{i=1}^{n} \delta_i - E\right)$

We derive: $\frac{a_i}{\delta_i} - \lambda = 0$ for all i

Substituting back into the constraint we find: $\delta_i = \frac{a_i E}{\sum_{i=1}^{n} a_i}$

Which is indeed E times a normalized probability.

3.1 Connection with Legendre Transform

3.1.1 1) Starting from equilibrium

The constraint is equivalent to: Let $E = \sum_{i=1}^{n} \delta_i$

We apply the Legendre transform to φ with respect to variable E:

We must therefore minimize $E\lambda - \varphi(\delta_i|_{i=1,n})$

The minimum is given by $\lambda = \frac{\partial \varphi(\delta_i|_{i=1,n})}{\partial E}$ which at the optimum defines the inverse transformation

$$\varphi(E) = \sum_{i=1}^{n} a_i \log(a_i E) - \sum_{i=1}^{n} a_i \log\left(\sum_{i=1}^{n} a_i\right)$$

Or

$$\varphi(E) = \left(\sum_{i=1}^{n} a_i\right) \log(E) + \sum_{i=1}^{n} a_i \log(a_i) - \left(\sum_{i=1}^{n} a_i\right) \log\left(\sum_{i=1}^{n} a_i\right)$$

Thus $\lambda = \frac{\sum_{i=1}^{n} a_i}{E}$

Therefore the free energy is written:

$$F(\lambda) = \sum_{i=1}^{n} a_i - \varphi\left(\frac{\sum_{i=1}^{n} a_i}{\lambda}\right)$$

Which simplifies to $F(\lambda) = (\sum_{i=1}^{n} a_i) \log \lambda + \sum_{i=1}^{n} a_i (1 - \log(a_i))$

Which equals $E\lambda - \varphi = \sum_{i=1}^{n} a_i (1 - \log(a_i))$ when $\lambda = 1$

3.1.2 2) General case

If we write the variables δ_i as: $\delta_i = \frac{E}{n} + \omega_i$ for $i = 2, n$ and $\delta_1 = \frac{E}{n} - \sum_{i=2}^{n} \omega_i$

We can then write $\varphi(E, \omega_i|_{i=2,n}) = a_1 \log\left(\frac{E}{n} - \sum_{i=2}^{n} \omega_i\right) + \sum_{i=2}^{n} a_i \log\left(\frac{E}{n} + \omega_i\right)$

We must therefore minimize $E\lambda - \varphi(E, \omega_i|_{i=2,n})$ The minimum is given by $\lambda(E, \omega_i|_{i=2,n}) = \frac{\partial \varphi(\delta_i|_{i=1,n})}{\partial E} = \frac{a_1}{E - n\sum_{i=2}^{n} \omega_i} + \sum_{i=2}^{n} \frac{a_i}{E + n\omega_i}$

Which implicitly defines $E(\lambda, \omega_i|_{i=2,n})$

The free energy is written $F(\lambda, \omega_i|_{i=2,n}) = E(\lambda, \omega_i|_{i=2,n})\lambda - \varphi(E(\lambda, \omega_i|_{i=2,n}), \omega_i|_{i=2,n})$

When we minimize this free energy with respect to $\omega_i|_{i=2,n}$ we obtain: $\frac{\partial F(\lambda)}{\partial \omega_i} = \frac{\partial E(\lambda, \omega_i|_{i=2,n})}{\partial \omega_i} \left(\lambda - \frac{\partial \varphi(E(\lambda, \omega_i|_{i=2,n}), \omega_i|_{i=2,n})}{\partial E}\right) - \frac{\partial \varphi(E(\lambda, \omega_i|_{i=2,n}), \omega_i|_{i=2,n})}{\partial \omega_i}$

Which can be re-expressed as: $\frac{\partial F(\lambda)}{\partial \omega_i} = \frac{\partial E(\lambda, \omega_i|_{i=2,n})}{\partial \omega_i} \left(\lambda - \frac{a_1}{E - n\sum_{j=2}^{n} \omega_j} - \sum_{j=2}^{n} \frac{a_j}{E + n\omega_j}\right) + \frac{a_1}{\frac{E}{n} - \sum_{j=2}^{n} \omega_j} - \frac{a_i}{\frac{E}{n} + \omega_i}$

At equilibrium: $\frac{\partial F(\lambda)}{\partial \omega_i} = 0$ We deduce a value for the derivative: $\frac{\partial E(\lambda,\omega_i|_{i=2,n})}{\partial \omega_i} = \frac{-\frac{a_1}{\frac{E}{n}-\sum_{j=2}^{n}\omega_j}+\frac{a_i}{\frac{E}{n}+\omega_i}}{\lambda-\frac{a_1}{E-n\sum_{j=2}^{n}\omega_j}-\sum_{j=2}^{n}\frac{a_j}{E+n\omega_j}}$

These equations coupled with: $\lambda(E,\omega_i|_{i=2,n}) = \frac{\partial \varphi(\delta_i|_{i=1,n})}{\partial E} = \frac{a_1}{E-n\sum_{i=2}^{n}\omega_i} + \sum_{i=2}^{n}\frac{a_i}{E+n\omega_i}$

Determine $\omega_i(\lambda)$ at equilibrium.

At equilibrium the denominator becomes zero when substituting the value of $\lambda(E,\omega_i|_{i=2,n})$.

Therefore the numerator must also be zero.

This implies that $\frac{-a_1}{E/n-\sum_{j=2}^{n}\omega_j} + \frac{a_i}{E/n+\omega_i} = 0$

Or $\frac{a_1}{E/n-\sum_{j=2}^{n}\omega_j} = \frac{a_i}{E/n+\omega_i}$

Or further $(\omega_i + E/n) = \frac{a_1}{a_i}(E/n - \sum_{j=2}^{n}\omega_j)$

Or again $\delta_i = \frac{a_1}{a_i}\delta_1$

Or $\delta_i = E/n$

Which is indeed the case at equilibrium.

Theorem 2

We want to maximize $\sum_{i,j,k} v_{i,j,k}\log(y_{i,j})$ under the constraint $\sum_j y_{i,j} = 1$ for all i

We write the Lagrangian $\sum_{i,j,k} v_{i,j,k}\log(y_{i,j}) - \sum_i \lambda_i \left(\sum_j y_{i,j} - 1\right)$

Stationarity with respect to $y_{i,j}$ implies that:

$$\sum_t \frac{v_{i,j,t}}{y_{i,j}} - \lambda_i = 0$$

for all i

Thus we derive the value $y_{i,j} = \frac{\sum_t v_{i,j,t}}{\lambda_i}$

Substituting into the constraint, we derive the value of λ_i:

$$\lambda_i = \sum_{j,k} v_{i,j,k}$$

And therefore the final value of $y_{i,j}$ is:

$$y_{i,j} = \frac{\sum_t v_{i,j,t}}{\sum_{j,k} v_{i,j,k}}$$

Theorem 3

We want to maximize $\sum_{i,j,t} v_{i,j,t}\alpha(i,j)\log(\gamma_{i,j})$ where $\alpha(i,j) = 0$ or 1

In this case the same formula leads to: $\gamma_{i,j} = \frac{\sum_t v_{i,j,t}\alpha(i,j)}{\sum_{j,t} v_{i,j,t}\alpha(i,j)}$

4 Expression of a Correlated HHMM within an HMM Framework

For an HMM, the CDLL (Complete Data Log-Likelihood) is: $\log\left(\delta_{c_1}\prod_t \gamma_{c_{t-1},c_t}\prod_t p_{c_t}(x_t)\right)$

Which re-expresses as $\log(\delta_{c_1})+\sum_t\log(\gamma_{c_{t-1},c_t})+\sum_t\log(p_{c_t}(x_t))$

And when we introduce: $u_j(t)=1$ when $c_t=j$ And $v_{j,k}(t)=1$ when $c_{t-1}=j$ and $c_t=k$

This gives CDLL =

$$\sum_{j=1}^{m}u_j(1)\log(\delta_j)+\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{t=2}^{T}v_{j,k}(t)\log(\gamma_{j,k})+\sum_{j=1}^{m}\sum_{t=1}^{T}u_j(t)\log(p_j(x_t))$$

The case of a doubly independent HMM is of course: CDLL =

$$\log(\delta_{c_1})+\log(\delta_{d_1})+\sum_t\left(\log(\gamma_{c_{t-1},c_t})+\log(\gamma_{d_{t-1},d_t})\right)+\sum_t\left(\log(p_{c_t}(x_t))+\log(p_{d_t}(y_t))\right)$$

If we add a third hidden variable representing correlation without observation: CDLL =

$$\text{CDLL}=\sum_{c_t,d_t,e_t}\log(\delta_{c_1})+\log(\delta_{d_1})+\log(\delta_{e_1})+\sum_t\left(\log(\gamma_{c_{t-1},c_t})+\log(\gamma_{d_{t-1},d_t})+\log(\gamma_{e_{t-1},e_t})\right)+\sum_t\left(\log(p_{c_t}(x_t))+\right.$$

Adding the correlation dependency changes the graph to:

$$\text{CDLL}=\sum_{c_t,d_t,e_t}\log(\delta_{c_t})+\log(\delta_{d_t})+\log(\delta_{e_t})$$

$$+\sum_t\left(\log(\gamma_{c_{t-1},e_t,c_t})+\log(\gamma_{d_{t-1},e_t,d_t})+\log(\gamma_{e_{t-1},e_t})\right)$$

$$+\sum_t\left(\log(p_{c_t}(x_t))+\log(p_{d_t}(y_t))\right)$$

This is equivalent to considering products of the type: $\log\left(\prod_t\gamma_{c_{t-1},e_t,c_t}\gamma_{d_{t-1},e_t,d_t}\gamma_{e_{t-1},e_t}p_{c_t}(x_t)p_{d_t}(y_t)\right)$

Which takes the recursive form:

$$\text{CDLL}=\log\left(\prod_{c_t,d_t,e_t}p_{c_t}(x_t)p_{d_t}(y_t)\prod_{c_{t-1},d_{t-1},e_{t-1}}\cdots\gamma_{c_{t-1},e_t,c_t}\gamma_{d_{t-1},e_t,d_t}\gamma_{e_{t-1},e_t}p_{c_{t-1}}(x_{t-1})p_{d_{t-1}}(y_{t-1})\prod_{c_{t-2},d_{t-2},e_{t-}}\right.$$

Therefore, performing a product over triplets c_t,d_t,e_t is equivalent to performing a simple product over a tensor space.

This is therefore a matrix product over the tensor spaces $c_{t-1}\otimes d_{t-1}\otimes e_{t-1}$, $c_{t-2}\otimes d_{t-2}\otimes e_{t-2}$, etc., followed by a contraction over the last indices: c_t,d_t,e_t.

Which re-expresses as:

$$\text{CDLL} = \sum_{j=1}^{m} u_j^f(1)\log(\delta_j^f) + \sum_{j=1}^{m} u_j^d(1)\log(\delta_j^d) + \sum_{j=1}^{m} u_j^e(1)\log(\delta_j^e)$$

$$+ \sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{t=2}^{T} v_{j,k,l}^f(t)\log(\gamma_{j,k,l}^f)$$

$$+ \sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{t=2}^{T} v_{j,k,l}^d(t)\log(\gamma_{j,k,l}^d) + \sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{t=2}^{T} v_{j,k}^e(t)\log(\gamma_{j,k}^e)$$

$$+ \sum_{j=1}^{m}\sum_{t=1}^{T} u_j^f(t)\log(p_j^f(x_t)) + \sum_{j=1}^{m}\sum_{t=1}^{T} u_j^d(t)\log(p_j^d(y_t))$$

Where the sufficient statistics are given by:

$$v_{j,k,l}^{x_1} = \text{prob}[x_1(t) = j : x_1(t-1) = k, e(t) = l]$$

$$v_{j,k,l}^{x_2} = \text{prob}[x_2(t) = j : x_2(t-1) = k, e(t) = l]$$

$$v_{j,k}^{e} = \text{prob}[e(t) = j : e(t-1) = k]$$

$$u_j^{x1} = \text{prob}[x1(t) = j]$$

$$u_j^{x2} = \text{prob}[x2(t) = j]$$

$$u_j^{e} = \text{prob}[e(t) = j]$$

We compute the following quantities:

$$v_{j,k,l,m,n,p}^T = \text{prob}[e(t) = j \,\&\, x_1(t) = k \,\&\, x_2(t) = l : e(t-1) = m \,\&\, x_1(t-1) = n \,\&\, x_2(t-1) = p]$$

thus $v_{j,k}^e = \sum_{k=1}^m \sum_{l=1}^m \sum_{n=1}^m \sum_{p=2}^T v_{j,k,l,m,n,p}^T$ a quantity obtained by quadruple marginalization of the double-slice marginal statistic over the triple tensor product.

Similarly for other statistics.

When we independently vary $\gamma_{j,k,l}^e, \gamma_{j,k,l}^d$ and $\gamma_{j,k}^e$, we obviously obtain the following stationary problems:

We maximize $\sum_{j=1}^m \sum_{k=1}^m \sum_{l=1}^m \sum_{t=2}^T v_{j,k,l}^{x1}(t)\log(\gamma_{j,k,l}^{x1})$

Under the constraint $\sum_{j=1}^m \gamma_{j,k,l}^{x1} = 1$ which comes from the fact that conditionally, the transition matrix is a probability. This gives us the stationary solution:

$$\gamma_{j,k,l}^{x1} = \frac{\sum_t v_{j,k,l}^{x1}(t)}{\sum_t v_{j,k,l}^{x1}(t)} \quad \text{and analogously for other quantities:}$$

$$\gamma_{j,k,l}^{x2} = \frac{\sum_t v_{j,k,l}^{x2}(t)}{\sum_t v_{j,k,l}^{x2}(t)} \quad \text{and } \gamma_{j,k}^{e} = \frac{\sum_t v_{j,k}^{e}(t)}{\sum_t v_{j,k}^{e}(t)}$$

4.1 Generalization

CDLL=

$$\sum_{i,l,c_{i,l}} \left(\log(\delta_{c_{i,l}})\right) + \sum_{i,l,c_{i,l}} \left(\log(\gamma_{c_{i-l},c_{i,l}})\right) + \sum_{i,l,c_{i,l}} \left(\log(p_{c_{i,l}}(x_{i,l}))\right)$$

Where $[c_{i,l}]$ means $c_{1,1}, c_{1,2}, c_{1,3}, \ldots$

$\gamma_{c_{t-1},c_{t,j}}$ therefore means $\gamma_{c_{t-1,1},c_{t-1,2},c_{t-1,3},\ldots,c_{t,1},c_{t,2},c_{t,3},\ldots}$

But these are identically zero if there is no node with $c_{t-1,1}, c_{t-1,2}, c_{t-1,3}, \ldots, c_{t,1}$ as input;

Similarly, only one of the j is active in $c_{t,j}$

The number of i can be arbitrary, and in particular $p_{c_{t,i}}(x_{t,i})$ can be identically zero, meaning it is a higher-order (abstract) Markov chain.

This is equivalent to considering products of the type: $\log\left(\prod_{t,k,l,p,q>p} \gamma_{c_{t-1,k},c_{t,l}} Y_{d_{t,p},d_{t,q}} \prod_i p_{d_{t,i}}(x_{t,i})\right)$

Which takes the recursive form:

$$\log\left(\prod_{c_t,d_t,e_t} p_{c_t}(x_t) p_{d_t}(y_t) \prod_{c_{t-1,1},\ldots,c_{t-1,n},d_{t,1},\ldots,d_{t,n}} \gamma_{c_{t-1,1},\ldots,c_{t,n}} Y_{d_{t,1},\ldots,d_{t,n}} p_{c_{t-1,1}}(x_{t-1}) \prod_{c_{t-2},d_{t-2},e_{t-2}} \cdots\right)$$

That is, we compute date-slice by date-slice.

4.2 HHMM with 3 Levels of Hidden Nodes

$$\text{CDLL} = \sum_{c_t,d_t,e_t} \log(\delta_{c_t}) + \log(\delta_{d_t}) + \log(\delta_{e_t})$$

$$+ \sum_t \left(\log(\gamma_{c_{t-1},c_t}) + \log(\gamma_{d_{t-1},c_t,d_t}) + \log(\gamma_{e_{t-1},d_t,e_t})\right) + \sum_t \log\left(p_{e_t}(x_t)\right)$$

$$\text{CDLL} = \log\left(\prod_{c_t,d_t,e_t} p_{e_t}(x_t) \prod_{c_{t-1},d_{t-1},e_{t-1}} \cdots \gamma_{c_{t-1},c_t} \gamma_{d_{t-1},c_t,d_t} \gamma_{e_{t-1},d_t,e_t} p_{e_{t-1}}(x_{t-1}) \prod_{c_{t-2},d_{t-2},e_{t-2}} \cdots\right)$$

Therefore, performing a product over triplets c_t, d_t, e_t is equivalent to performing a simple product over a tensor space.

This is therefore a matrix product over the tensor spaces $c_{t-1} \otimes d_{t-1} \otimes e_{t-1}$, $c_{t-2} \otimes d_{t-2} \otimes e_{t-2}$, etc., followed by a contraction over the last indices: c_t, d_t, e_t

Which re-expresses as:

$$\text{LL} = \sum_{j=1}^{m} u_j^c(1) \log(\delta_j^c) + \sum_{j=1}^{m} u_j^d(1) \log(\delta_j^d) + \sum_{j=1}^{m} u_j^e(1) \log(\delta_j^e) + \sum_{j=1}^{m} \sum_{k=1}^{m} v_{j,k}^c(t) \log(\gamma_{j,k}^c)$$

$$+\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{t=2}^{T} v^{d}_{j,k,l}(t)\log(\gamma^{d}_{j,k,l}) + \sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{t=2}^{T} v^{e}_{j,k,l}(t)\log(\gamma^{e}_{j,k,l})$$

$$+\sum_{j=1}^{m}\sum_{t=1}^{T} u^{c}_{j}(t)\log(p^{c}_{j}(x_t))$$

Where the sufficient statistics are given by:

$$v^{e}_{j,k,l} = \text{prob}[e(t) = j : e(t-1) = k, d(t) = l]$$

$$v^{d}_{j,k,l} = \text{prob}[d(t) = j : d(t-1) = k, c(t) = l]$$

$$v^{e}_{j,k} = \text{prob}[c(t) = j : c(t-1) = k]$$

$$u^{e}_{j} = \text{prob}[e(t) = j]$$

$$u^{d}_{j} = \text{prob}[d(t) = j]$$

$$u^{c}_{j} = \text{prob}[c(t) = j]$$

We compute the following quantities:

$$v^{T}_{j,k,l,m,n,p} = \text{prob}[e(t) = j \;\&\; d(t) = k \;\&\; c(t) = l : e(t-1) = m \;\&\; d(t-1) = n \;\&\; c(t-1) = p]$$

thus

$$v^{e}_{j,m,k} = \sum_{l=1}^{m}\sum_{n=1}^{m}\sum_{p=2}^{m} v^{T}_{j,k,l,m,n,p}$$

$$v^{d}_{j,m,k} = \sum_{l=1}^{m}\sum_{n=1}^{m}\sum_{p=2}^{m} v^{T}_{j,k,l,m,n,p}$$

$$v^{c}_{j,k} = \sum_{l=1}^{m}\sum_{n=1}^{m}\sum_{m=1}^{m}\sum_{p=2}^{m} v^{T}_{j,k,l,m,n,p}$$

a quantity obtained by quadruple marginalization of the double-slice marginal statistic over the triple tensor product.

When we independently vary $\gamma^{e}_{j,k,l}$, $\gamma^{d}_{j,k,l}$ and $\gamma^{c}_{j,k}$, we obviously obtain the following stationary problems:

We maximize $\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{t=2}^{T} v^{e}_{j,k,l}(t)\log(\gamma^{e}_{j,k,l})$

Under the constraint $\sum_{j=1}^{n}\gamma^{e}_{j,k,l} = 1$ which comes from the fact that conditionally, the transition matrix is a probability. This gives us the stationary solution:

$$\gamma^{e}_{j,k,l} = \frac{\sum_t v^{e}_{j,k,l}(t)}{\sum_t v^{e}_{j,k,l}(t)} \quad \text{and analogously for other quantities:}$$

$$\gamma^{d}_{j,k,l} = \frac{\sum_t v^{d}_{j,k,l}(t)}{\sum_t v^{d}_{j,k,l}(t)} \quad \text{and } \gamma^{c}_{j,k} = \frac{\sum_t v^{c}_{j,k}(t)}{\sum_t v^{c}_{j,k}(t)}$$

5 HHMM with Infinite Levels of Hidden Nodes and Activation

Variable n_t

Represents the number of reset levels. Value between 0 and infinity. Distribution:

$$\sum_{k=0}^{\infty} \text{Prob}[n_t = k] = 1$$

And $i \to \text{Prob}[n_t = i]$ decreasing to implement the notion of increasing abstraction in the HHMM.

A priori $n_t = n_t(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, \infty)$

Thus $\text{Prob}[n_t = n | c_{0,t-1} = j_{0,t-1}, c_{1,t-1} = j_{1,t-1}, \ldots, \infty] = b_n(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, \infty)$

With

$$\sum_{n=0}^{\infty} b_n(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, \infty) = 1$$

Level initialization:

$$a_{n,j_{n,t}}(j_{n+1,t}) = \text{Prob}[c_{n,t} = j_{n,t}, c_{n+1,t} = j_{n+1,t}, n_t > n]$$

$$\sum_{i=1}^{N_{\max}[n]} a_{n,i}(j_{n+1,t}) = 1$$

Regular transition:

$$a_{0,t}(j_{1,t}, j_{0,t-1}) = \text{Prob}[c_{0,t} = i, c_{0,t-1} = j_{0,t-1}, c_{1,t} = j_{1,t}, n_t = 0]$$

$$\sum_{i=1}^{N_{\max}[n]} a_{0,i}(j_{1,t}, j_{0,t-1}) = 1$$

Reconstruction of the flat transition matrix (associated with the equivalent flat HMM) where red conditions are synchronous, and blue conditions are from the previous date:

$$\text{Prob}[c_{0,t} = j_{0,t}, c_{1,t} = j_{1,t}, \ldots, c_{n,t} = j_{n,t}, c_{0,t-1} = j_{0,t-1}, c_{1,t-1} = j_{1,t-1}, \ldots, c_{n,t-1} = j_{n,t-1}] =$$

$$\text{Prob}[n_t = 0]\text{Prob}[c_{0,t} = j_{0,t}, c_{0,t-1} = j_{0,t-1}, c_{1,t} = j_{1,t} | n_t = 0]$$
$$+ \sum_{n=1}^{\infty} \text{Prob}[n_t = n]\text{Prob}[c_{n,t} = j_{n,t}, c_{n+1,t} = j_{n+1,t} | n_t > n]$$

Thus if n is the cutoff or maximum value for the levels

$$\text{Prob}[c_{0,t} = j_{0,t}, c_{1,t} = j_{1,t}, \ldots, c_{n,t} = j_{n,t}, c_{0,t-1} = j_{0,t-1}, c_{1,t-1} = j_{1,t-1}, \ldots, c_{n,t-1} = j_{n,t-1}] =$$

$$A_{j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t-1}, j_{0,t}, j_{1,t}, j_{2,t}, \ldots, j_{n,t}} =$$

$$b_0(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t}) a_{0,j_{0,t}}(j_{1,t}, j_{0,t-1}) + \sum_{n=1}^{\infty} b_n(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, \infty) a_{n,j_{n,t}}(j_{n+1,t})$$

Having identified the states and the transition matrix,

We of course have the initial probability of the hidden network

$$\text{Prob}[c_{0,0} = j_{0,0}, c_{1,0} = j_{1,0}, \ldots, c_{n,0} = j_{n,0}] = \delta_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}}$$

$$\sum_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}} \delta_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}} = 1$$

The distribution of the observed variable is represented by the distribution

$$p_{t, j_{0,t}}(x, \theta)$$

We can then write the likelihood: CDLL=

$$\sum_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}} \left(\log(\delta_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}})\right)$$

$$+ \sum_{t, j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t-1}, j_{0,t}, j_{1,t}, j_{2,t}, \ldots, j_{n,t}} \left(\log(A_{j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t-1}, j_{0,t}, j_{1,t}, j_{2,t}, \ldots, j_{n,t}})\right)$$

$$+ \sum_{t, j_{0,t}} \left(\log(p_{t, j_{0,t}}(x_{t, j_{0,t}}, \theta))\right)$$

We substitute the transition matrix into this formula to examine simplifications and refactorings:

$$\sum_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}} \left(\log(\delta_{j_{0,0}, j_{1,0}, \ldots, j_{n,0}})\right)$$

$$+ \sum_{t, j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t-1}, j_{0,t}, j_{1,t}, j_{2,t}, \ldots, j_{n,t}} \left(\log\left(b_0(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, j_{n,t}) a_{0,j_{0,t}}(j_{1,t}, j_{0,t-1})\right)\right)$$

$$+ \sum_{n=1}^{\infty} b_n(j_{0,t-1}, j_{1,t-1}, j_{2,t-1}, \ldots, \infty) a_{n,j_{n,t}}(j_{n+1,t}) \Bigg) + \sum_{t, j_{0,t}} \left(\log\left(p_{t, j_{0,t}}(x_{t, j_{0,t}}, \theta)\right)\right)$$

6 Modern Perspective: From HHMMs to Contemporary AI

6.1 Historical Context of HHMMs

Hierarchical Hidden Markov Models (HHMMs) represented a significant conceptual advance in sequential modeling during the early 2000s. They introduced structured abstraction into time series analysis through multi-level latent variables. At the time, this provided an elegant probabilistic formalism for capturing temporal dependencies at multiple scales, especially useful for tasks such as speech recognition, behavior modeling, and time series prediction.

Their key strengths included:

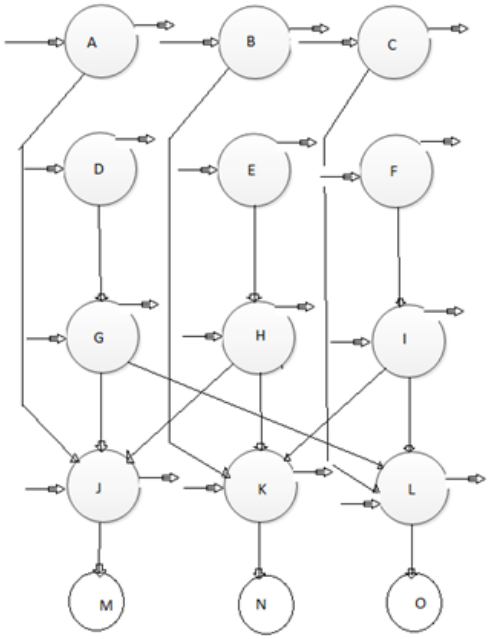

Figure 1: A Bayesian network example

- **Explicit hierarchy:** Modeling different abstraction levels through state hierarchy
- **Interpretability:** Clear probabilistic semantics and explainable state transitions
- **Context modeling:** Ability to incorporate contextual information through hidden state relationships

However, HHMMs faced practical limitations:

- **Computational complexity:** Exponential growth in parameters with hierarchy depth
- **Handcrafted structure:** Required manual design of state hierarchies
- **Discrete state limitations:** Struggled with continuous, high-dimensional data

6.2 Modern Analogues: Neural Sequence Models

Today, the landscape of sequence modeling has been dramatically reshaped by the advent of deep learning, particularly Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and more recently, Transformers.

In this document, we reflect on the core ideas of HHMMs and propose how they can be reinterpreted or enhanced in light of modern architectures, particularly in the context of time series prediction with contextual understanding.

Today's state-of-the-art approaches have largely addressed these limitations through deep learning architectures:

Feature	**HHMM (2013)**	**Modern Approaches**
Hierarchy modeling	Explicit state hierarchy	Learned hierarchical representations
Parameter learning	EM algorithm	Gradient-based optimization
Context handling	Fixed transition rules	Attention mechanisms
State representation	Discrete	Continuous embeddings
Long-range dependencies	Limited	Gating mechanisms (LSTM/GRU)

Table 1: Evolution of time series modeling approaches

7 Key Contributions of HHMMs

- **Multi-level Abstraction**: Each level of the hierarchy encodes a different timescale, enabling abstraction across time.
- **Explicit Temporal Structure**: HHMMs provide a formal mechanism for modeling variable-duration states and nested transitions.
- **Modular Inference**: The model allows modular inference procedures using message passing and expectation maximization (EM).

8 Limitations in the Contemporary Context

While HHMMs were powerful, they also suffered from several limitations that have become more apparent in light of deep learning:

- **Intractable Inference**: For deep hierarchies or large state spaces, EM inference becomes computationally demanding.
- **Rigid Parametric Forms**: Emission and transition distributions are often chosen from simple families (e.g., Gaussians, Multinomials), limiting representational capacity.
- **Data Requirements**: Learning hierarchical structures requires substantial annotated data or strong priors.

8.1 Key Modern Connections

8.2 Connection to Recurrent Neural Networks

Modern RNNs and LSTMs can be viewed as non-linear, continuous generalizations of HMMs. In particular:

- The hidden state in an RNN corresponds to the latent state in an HMM.
- Transition dynamics are learned via gradient descent rather than EM.
- Temporal dependencies can be captured over long ranges using LSTM gates, mimicking hierarchical abstraction.

8.3 Transformers and Attention Mechanisms

Transformers discard sequential recurrence altogether and instead use attention to learn dependencies:

- Self-attention layers enable dynamic contextual weighting over past inputs.
- Positional encodings restore sequence information, which HHMMs encode structurally.
- Transformers can learn multi-scale temporal patterns through stacked attention heads and layers, indirectly modeling abstraction.

8.4 Neural Probabilistic Models

There is a growing movement to blend probabilistic graphical models with deep learning:

- **Variational Autoencoders (VAEs)** with sequential structure (e.g., Variational RNNs) revisit ideas of latent dynamics.
- **Structured State-Space Models (SSMs)** now incorporate neural networks to model transition and observation functions.
- **Neural Hierarchical Models** learn latent abstraction levels similar to HHMMs but with amortized inference.

9 Opportunities for Synthesis

A modern re-examination of HHMMs could include:

- **Hybrid Models**: Embedding HHMM-like structures within neural networks (e.g., with discrete latent variables using Gumbel-Softmax or REINFORCE).
- **Neural Parameterization**: Parameterizing transitions and emissions via deep networks.
- **Hierarchical Attention**: Structuring Transformers to learn multi-level time abstractions explicitly.
- **Contextual Time Series Prediction**: Combining structured temporal priors with learned context encoders.

9.0.1 Recurrent Neural Networks (RNNs) as Continuous Analogues

The HHMM's chain structure finds its continuous counterpart in RNNs:

- Hidden states become continuous vector representations
- State transitions are parameterized by neural networks
- LSTM/GRU gates address vanishing gradient problems

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$$

9.0.2 Transformers and Attention Mechanisms

The context-handling capabilities of HHMMs are dramatically enhanced by attention:

- **Self-attention:** Dynamically focuses on relevant time steps
- **Multi-head attention:** Captures different contextual relationships
- **Positional encoding:** Preserves temporal ordering information

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

9.0.3 Probabilistic Deep Learning

Modern approaches combine probabilistic rigor with deep learning flexibility:

- **Deep State-Space Models:** Combine neural networks with state-space formulations
- **Neural Processes:** Model stochastic processes with neural networks
- **Transformer-based Forecasting:** Models like Temporal Fusion Transformers

9.1 Current Research Directions

Building on HHMM concepts, today's frontier includes:

1. **Hierarchical Attention:** Explicitly modeling multiple time scales (e.g., hourly, daily, weekly patterns)
2. **Neural Differential Equations:** Continuous-time modeling of dynamic systems
3. **Transfer Learning:** Pretraining on large datasets, fine-tuning for specific domains

4. **Uncertainty Quantification:** Bayesian deep learning for reliable predictions

5. **Multimodal Integration:** Combining time series with text, image, and graph data

9.2 Conclusion

While HHMMs provided a principled foundation for hierarchical time series modeling, modern approaches have overcome their limitations through:

- Continuous representation learning instead of discrete states
- Automatic feature extraction instead of handcrafted structures
- Scalable attention mechanisms instead of fixed transition rules
- End-to-end optimization instead of EM algorithms

Bridging HHMMs with deep learning may yield powerful new tools for structured time series analysis under uncertainty.