COMP 652: Assignment 3

Due on Tuesday, March 31 2015

Presented to Dr. Doina Precup

Geoffrey Stanley
Student ID: 260645907

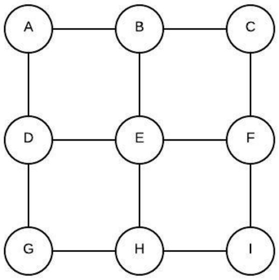
Question 1

A)

In spin glass models the maximum likelihood is computed as the sum of the products of the maximal cliques energy of each node.

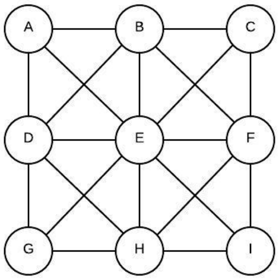
$$\log L(\psi|D) = \sum_{i=1}^{N} \log \frac{1}{Z} \prod_{C} \psi_C(x_C) \tag{1}$$

In a 4-neighbor spin glass model the maximal cliques were the edges between each pixel. As illustrated below:



$$P(E) = \psi(x, Y = \{B, E\})\psi(x, Y = \{D, E\})\psi(x, Y = \{E, F\})\psi(x, Y = \{E, H\}) \tag{2}$$

In an 8-neighbor spin glass model the maximal cliques become clusters of 4 nodes. As such, the energy of a node will be computed as the product of 4 clusters of 4:



$$P(E) = \psi(x, Y = \{A, B, D, E\})\psi(x, Y = \{B, C, E, F\})\psi(x, Y = \{D, E, G, H\})\psi(x, Y = \{E, F, H, I\}) \tag{3}$$

The energy computation can remain the same however.

$$\psi(x, Y) = \alpha x + \sum_{i} \beta_i y_i x \tag{4}$$

Where i ranges from 0 to 7 for the 8 neighbors of a particular node.

B)

The advantages and disadvantages would be related to a trade off between model precision and computation time.

Given a 4 neighbor model computation time will be smaller given 20 fewer edge energies to be computed. However, this is at the expense of model accuracy. With the 8 neighbor model more data will be used to infer the value of a pixel as well as being more flixible due to using information coming from 4 new angles.

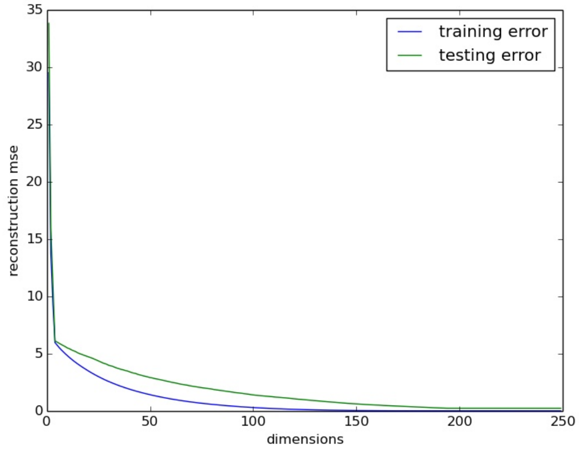
This should improve the models accuracy. In particular, when working with an image that contains circular shapes or curved lines.

C)

A gibbs sampling algorithm in this situation, where evidence is injected along the left most edge, would require two passes. The first, would be a forward pass using only two of the four neighbor pixels. In the case of the first column, it will use the injected pixel to the left as well as the last infered pixel. Subsequent columns would use the two previous infered pixels.

The second would be a backward pass and would use all four pixels. It would start from the bottom and works it's way back to the initially infered pixel. This would ensure that all pixels were estimated given the probability distribution of it's four neighbors.

Question 2



As dimensions are reduced from 250 the reconstruction error is initially quite small but becomes more important as dimensions approach 0. The shoulder of the reconstruction error line is at a dimension of approximately 25.

From this, we can conclude that given this input space we can use PCA to reduce dimensionality of the data to 25 in order to make computation more manageable without losing very much granularity in the feature data.

Question 3

A)

As with standard Hidden Markov Models, Coupled Hidden Markov models will have three categories of parameters. These are the initial probabilities, the transition probabilites and the emission probabilities. Given the system depicted in Figure 1 of assignment 3:

Initial Probabilities:

$$P(s0) \tag{5}$$

$$P(u0) \tag{6}$$

Transition Probabilities:

$$P(s_i|s_{i-1}, u_{i-1}) \tag{7}$$

$$P(u_i|u_{i-1}, s_{i-1}) \tag{8}$$

Emission Probabilities:

$$P(y_i|s_i) \tag{9}$$

$$P(z_i|u_i) \tag{10}$$

B)

In order to compute the joint probability of a sequence of observations a forward algorithm will need to be derived.

$$\begin{aligned}
\alpha_t(s_t, u_t) &= P(s_t, u_t, y_{0:T}, x_{0:T}) \\
&= \sum_{t=1}^{T} p(s_t, s_{t-1}, u_t, u_{t-1}, y_{1:T}, x_{1:T}) \\
&= \sum_{t=1}^{T} p(y_t|s_t)p(x_t|u_t)p(s_t|s_{t-1}, u_{t-1})p(u_t|s_{t-1}, u_{t-1})p(s_{t-1}, u_{t-1}, y_{1:T-1}, x_{1:T-1}) \\
&= \sum_{t=1}^{T} p(y_t|s_t)p(x_t|u_t)p(s_t|s_{t-1}, u_{t-1})p(u_t|s_{t-1}, u_{t-1})\alpha_{t-1}(y_{t-1}, x_{t-1})
\end{aligned} \tag{11}$$

Because the equation listed above does not contain the initial probability we need to compute it seperately:

$$\alpha_0(s_0, u_0) = p(y_0, z_0, s_0, u_0) = p(s_0)p(y_0|s_0)p(u_0)p(z_0|u_0) \tag{12}$$

Summing the equations above we obtain the joint probability:

$$p(y_0, z_0, y_1, z_1, ..., y_T, z_T) = \sum_{i=0}^{T} \alpha_i(s_i, u_i) \tag{13}$$

C)

The forward-backward algorithm is equal to the product between the forward and the backward algorithm. In the previous question I derived the forward algorithm. So, what is remaining is to derive the backward algorithm:

$$\begin{aligned}
\beta_t(s_t, u_t) &= p(y_{t+1:n}|s_t, u_t)p(z_{t+1:n}|s_t, u_t) \\
&= \sum_{k=0}^{T-1} p(y_{t+1:n}, z_{t+1:n}, s_{t+1}, u_{t+1}|s_t, u_t) \\
&= \sum_{k=0}^{T-1} p(y_{t+2:n}|s_{t+1}, u_{t+1})p(z_{t+2:n}|s_{t+1}, u_{t+1})p(y_{t+1}|s_{t+1})p(z_{t+1}|u_{t+1})p(s_{t+1}|s_t, u_t)p(u_{t+1}|s_t, u_t) \\
&= \sum_{k=0}^{T-1} \beta_{t+1}(s_{t+1}, u_{t+1})p(y_{t+1}|s_{t+1})p(z_{t+1}|u_{t+1})p(s_{t+1}|s_t, u_t)p(u_{t+1}|s_t, u_t)
\end{aligned} \tag{14}$$

The probability at time T will be equal to 1 :

$$\beta_T(s_T, u_T) = 1 \tag{15}$$

Summing the equations above we obtain te backward algorithm:

$$p(y_{t+1:n}|s_t, u_t)p(z_{t+1:n}|s_t, u_t) = \sum_{i=0}^{T} \beta_i(s_i, u_i) \tag{16}$$

We can now derive the forward-backward algorithm as the product between the forward and the backward algorithm:

$$p(s_t, u_t|y_{0:T}, x_{0:T}) = \left(\sum_{i=0}^{T} \alpha_i(s_i, u_i)\right)\left(\sum_{i=0}^{T} \beta_i(s_i, u_i)\right) \tag{17}$$

D)

The transition probabilities as well as the inference algorithms used will be conditional on whether $t \mod k = 0$. More precisely, for transition probabilities:

$$\begin{cases} P(s_i|s_{i-1}, u_{i-1}) & \text{if } i \mod k = 0 \\ P(s_i|s_{i-1}) & \text{otherwise} \end{cases} \tag{18}$$

$$\begin{cases} P(u_i|u_{i-1}, s_{i-1}) & \text{if } i \mod k = 0 \\ P(u_i|u_{i-1}) & \text{otherwise} \end{cases} \tag{19}$$

And the computation for the inference algorithms will be similar. For the forward algorithm:

$$\begin{cases} \sum_{t=1}^{T} p(y_t|s_t)p(x_t|u_t)p(s_t|s_{t-1}, u_{t-1})p(u_t|s_{t-1}, u_{t-1})\alpha_{t-1}(y_{t-1}, x_{t-1}) & \text{if } t \mod k = 0 \\ \sum_{t=1}^{T} p(y_t|s_t)p(x_t|u_t)p(s_t|s_{t-1})p(u_t|u_{t-1})\alpha_{t-1}(y_{t-1}, x_{t-1}) & \text{otherwise} \end{cases} \tag{20}$$

And for the backward algorithm:

$$\begin{cases} \sum_{t=0}^{T-1} \beta_{t+1}(s_{t+1}, u_{t+1})p(y_{t+1}|s_{t+1})p(z_{t+1}|u_{t+1})p(s_{t+1}|s_t, u_t)p(u_{t+1}|s_t, u_t) & \text{if } i \mod k = 0 \\ \sum_{t=0}^{T-1} \beta_{t+1}(s_{t+1}, u_{t+1})p(y_{t+1}|s_{t+1})p(z_{t+1}|u_{t+1})p(s_{t+1}|s_t)p(u_{t+1}|u_t) & \text{otherwise} \end{cases} \tag{21}$$

E)

The object of the learning algorithm should be to find the K whose joint probability is the greatest. As such, for each K in the the range 0 to n the joint probability will have to be calculated. Once this is done, the K whose joint probability is the greatest should be used as the most accurate model.