CS 475 Machine Learning: Homework 6
Graphical Models
Analytical Questions

Due: Dec 7, 2020, 11:59 pm US/Eastern

50 Points Total Version 1.1 (Updated Nov 30)

Wenkai Luo (wluo14), Yuetong Liu (yliu390)

Instructions

We have provided this LaTeX document for turning in this homework. We give you one or more boxes to answer each question. The question to answer for each box will be noted in the title of the box.

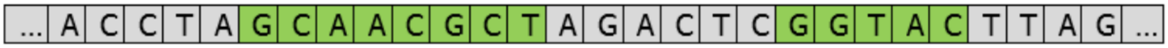
Other than your name, do not type anything outside the boxes. Leave the rest of the document unchanged.

Do not change any formatting in this document, or we may be unable to grade your work. This includes, but is not limited to, the height of textboxes, font sizes, and the spacing of text and tables. Additionally, do not add text outside of the answer boxes. Entering your answers are the only changes allowed.

We strongly recommend you review your answers in the generated PDF to ensure they appear correct. We will grade what appears in the answer boxes in the submitted PDF, NOT the original latex file.

1) Expectation Maximization: Hidden Markov models to find genes

DNA carries genetic information and is made of four bases A, C, G and T. Broadly, regions of the genome are either coding consisting of genes that encode the primary structure of proteins or non-coding. While the genome is made of roughly three million bases, protein coding genes only make up approximately 1% of the genome. Biologists are interested in determining which regions of the genome correspond to protein coding genes since mutations in these regions can change the structure and hence function of proteins leading to pathological states.



... A C C T A G C A A C G C T A G A C T C G G T A C T T A G ...

Figure 1: Example of DNA sub-sequence with genes shown in green

In general, the observed frequency of different bases is different between the coding and the non-coding regions of the genome. For instance, C and G are found more often in protein coding regions. These statistical patterns can be exploited to identify regions corresponding to genes using Hidden Markov models. These models take as input the observed sequence of bases and includes latent binary hidden variables that encode whether a particular region is protein coding or not. When formulated in this manner, the observations and the latent variable are discrete and can only occupy finite states.

Experimentally, different sequencing methods are employed to "read" the genome. One such method is nanopore sequencing which uses a protein nanopore set in an electrically resistant polymer membrane. An ionic current is passed through the pore by setting a voltage across this membrane, and as different bases pass through the pore, they result in different values of the current measured (Fig 2). Here, the observations are not discrete, rather the observed current values can be modeled using a Gaussian mixture model with four components corresponding to each base.

We would like to identify whether a particular region of the genome is protein coding or not based on observing the pore residual current. Further, we would like to obtain the most probable sequence of bases given a certain current observation. The model we will use for this is called tied-mixture HMM.

For a DNA sequence of length T, z_t determines the region type of the t-th base and x_t is the electrical signal for the t-th base. Finally, the value of the t-th base is given by y_t which is called the mixture variable of the model. We specify our model as below

$$p(z_1 = 1) = a_1 \tag{1}$$

$$p(z_t = j | z_{t-1} = i) = a_{ij} \qquad i = 1, 2 \qquad j = 1, 2 \tag{2}$$

$$p(y_t = j | z_t = 1) = b_{ij} \qquad i = 1, 2 \qquad j = 1, 2, 3, 4 \tag{3}$$

$$p(x_t | y_t) = \prod_{i=1}^{4} \mathcal{N}(\mu_i, \sigma_i)^{\mathbb{I}(y_t = i)} \tag{4}$$

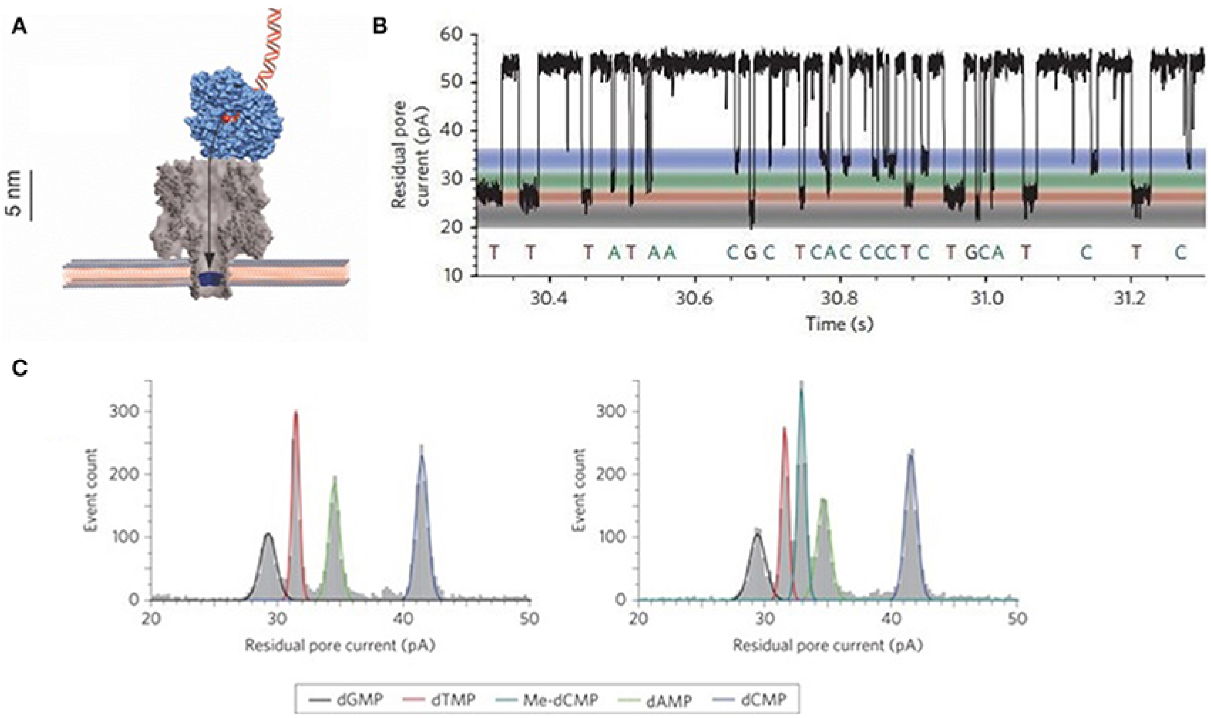


Figure 2: (A) Schematic representation of nanopore sequencing, (B) Observed current as the strand is read, (C) The distribution of residual pore current for different bases

The parameters in our model are therefore

a_1 : The initial probability of $z_1 = 1$

$a_{i,j}$: The transition probability of z_t to z_{t+1}

$b_{i,j}$: The probability of a particular base given that a region is protein coding

μ_i : Mean residual current for a given base i

σ_i : Standard deviation of residual current for a given base i

We are given S sequences $\{x_1^s, \ldots, x_T^s\}_{s=1}^S$. Therefore the complete data likelihood is given by,

$$p(\mathcal{D}) = \prod_{s=1}^{S} p(x_1^s|y_1^s, \mu, \sigma)p(y_1^s|z_1^s, \mathbf{b})p(z_1^s|a_1) \prod_{t=2}^{T} p(x_t^s|y_t^s, \mu, \sigma)p(y_t^s|z_t^s, \mathbf{b})p(z_t^s|z_{t-1}^s, \mathbf{a}) \tag{5}$$

$$\begin{aligned}\log\text{ p}(\mathcal{D}) = \sum_{s=1}^{S} \Bigg[&\log\text{ p}(x_1^s|y_1^s, \mu, \sigma) + \log\text{ p}(y_1^s|z_1^s, \mathbf{b}) + \log\text{ p}(z_1^s|a_1) \\ &+ \sum_{t=2}^{T} \log\text{ p}(x_t^s|y_t^s, \mu, \sigma) + \log\text{ p}(y_t^s|z_t^s, \mathbf{b}) \\ &+ \log\text{ p}(z_t^s|z_{t-1}^s, \mathbf{a}) \Bigg] \end{aligned} \tag{6}$$

(a) In the E-step, we compute the posterior of the latent variables given the data, and set the $q(.)$ distribution to the posterior. Using Bayes rule compute $p(\mathbf{y}, \mathbf{z}|\mathbf{x}, \Theta^{old})$. Hint: $\mathbf{x} \perp\!\!\!\perp \mathbf{z}|\mathbf{y}$

Propose an algorithm to compute the posterior exactly (How will you compute the marginal in the denominator?)

$p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old}) = \frac{p(\mathbf{x},\mathbf{z}|\mathbf{y},\Theta^{old})\ p(\mathbf{y}|\Theta^{old})}{p(\mathbf{x}|\Theta^{old})} = \frac{p(\mathbf{x}|\mathbf{y},\Theta^{old})\ p(\mathbf{y}|\mathbf{z},\Theta^{old})\ p(\mathbf{z}|\Theta^{old})}{p(\mathbf{x}|\Theta^{old})}$
$p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old}) = \frac{p(x_1|y_1,\mu,\sigma)p(y_1|z_1,\mathbf{b})p(z_1|a_1)\prod_{t=2}^{T} p(x_t|y_t,\mu,\sigma)p(y_t|z_t,\mathbf{b})p(z_t|z_{t-1},\mathbf{a})}{p(\mathbf{x}|\Theta^{old})}$
We set $p_s(D) = p(x_1|y_1,\mu,\sigma)p(y_1|z_1,\mathbf{b})p(z_1|a_1)\prod_{t=2}^{T} p(x_t|y_t,\mu,\sigma)p(y_t|z_t,\mathbf{b})p(z_t|z_{t-1},\mathbf{a})$, and $p(\mathbf{x}|\Theta^{old}) = \sum_{\mathbf{y},\mathbf{z}} p(\mathbf{x},\mathbf{y},\mathbf{z}|\Theta^{old}) = \sum_{\mathbf{y},\mathbf{z}} p_s(D)$, then we can rewrite $p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})$ as the following:

$$p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old}) = \frac{p_s(D)}{\sum_{\mathbf{y},\mathbf{z}} p_s(D)}$$

$$p_s(D) = p(x_1|y_1,\mu,\sigma)p(y_1|z_1,\mathbf{b})p(z_1|a_1)\prod_{t=2}^{T} p(x_t|y_t,\mu,\sigma)p(y_t|z_t,\mathbf{b})p(z_t|z_{t-1},\mathbf{a})$$

(b) In the M-step, we would like to maximize the $Q(\Theta \mid \Theta^{old})$ which is the expectation of the complete data log likelihood with respect to the posterior of the latent variables. Therefore,

$$Q(\Theta \mid \Theta^{old}) = \mathrm{E}[\log\ \mathrm{p}(\mathbf{x},\mathbf{y},\mathbf{z}|\Theta)|\mathbf{x},\Theta^{old}] \tag{7}$$

$$= \sum_{\mathcal{Y},\mathcal{Z}} p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})\log\ \mathrm{p}(\mathbf{x},\mathbf{y},\mathbf{z}|\Theta) \tag{8}$$

Assuming you have access $p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})$ from part (a), write out the expression for $Q(\theta \mid \theta^{\text{old}})$ in terms of the components of θ: $a_1, a_{i,j}, b_{i,j}, \mu_i$, and σ_i.

Assuming we have access $p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})$, thus
$Q(\Theta \mid \Theta^{old}) = \sum_{\mathcal{Y},\mathcal{Z}} p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})\log\ \mathrm{p}(\mathbf{x},\mathbf{y},\mathbf{z}|\Theta) = \sum_{\mathcal{Y},\mathcal{Z}} p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old})\log\ \mathrm{p}_s(D)$
$\log\ \mathrm{p}_s(D) = \log\ \mathrm{p}(x_1|y_1,\mu,\sigma) + \log\ \mathrm{p}(y_1|z_1,\mathbf{b}) + \log\ \mathrm{p}(z_1|a_1) + \sum_{t=2}^{T} [\log\ \mathrm{p}(x_t|y_t,\mu,\sigma) + \log\ \mathrm{p}(y_t|z_t,\mathbf{b}) + \log\ \mathrm{p}(z_t|z_{t-1},\mathbf{a})]$
$\log\ \mathrm{p}_s(D) = \sum_{t=1}^{T} [\log\ \mathrm{p}(x_t|y_t,\mu,\sigma) + \log\ \mathrm{p}(y_t|z_t,\mathbf{b})] + \sum_{t=2}^{T} \log\ \mathrm{p}(z_t|z_{t-1},\mathbf{a}) + \log\ \mathrm{p}(z_1|a_1)$
$\log\ \mathrm{p}_s(D) = \log\ \mathrm{p}(z_1|a_1) + \sum_{t=1}^{T} \left[\log\ \mathrm{a}_{z_t\ \ z_{t-1}} + \log\ \mathrm{p}(y_t|z_t,\mathbf{b}) + \log\ \mathrm{p}(z_t|z_{t-1},\mathbf{a})\right]$
$\log\ \mathrm{p}_s(D) = \sum_{t=1}^{T} \left[\log\ (a_{z_t z_{t-1}} + \log\ (\prod_{i=1}^{4} \mathcal{N}(\mu_i,\sigma_i)^{\mathbb{I}(y_t=i)})\right] + \sum_{t=2}^{T} \log\ \mathrm{b}_{z_t y_t} + \log\ (a_1^{\mathbb{I}(y_1=1)} \cdot (1-a_1)^{\mathbb{I}(y_1=2)})$
$Q(\Theta \mid \Theta^{old}) = \sum_{\mathcal{Y},\mathcal{Z}} p(\mathbf{y},\mathbf{z}|\mathbf{x},\Theta^{old}) \sum_{t=1}^{T} \left[\log\ (a_{z_t z_{t-1}} + \log\ (\prod_{i=1}^{4} \mathcal{N}(\mu_i,\sigma_i)^{\mathbb{I}(y_t=i)})\right] + \sum_{t=2}^{T} \log\ \mathrm{b}_{z_t y_t} + \log\ (a_1^{\mathbb{I}(y_1=1)} \cdot (1-a_1)^{\mathbb{I}(y_1=2)})$

(c) Finally, we can compute the gradients of the $Q(\Theta \mid \Theta^{old})$ w.r.t our parameters to derive the updates. Derive an expression for μ^{n+1} in terms of the data and the parameters from the previous iteration Θ^n.

Suppose you trained your HMM using EM on a dataset and obtained the following transition and emission probabilities.

$p(z_1 = \mathbf{P}) = 0.5$

	Next		
Current	P	N	End
Start	0.5	0.5	0
P	0.7	0.2	0.1
N	0.1	0.8	0.1

	Base			
State	A	C	G	T
P	0.2	0.3	0.3	0.2
N	0.3	0.2	0.2	0.3

We will assume that we are provided with the nucleotide sequence and are required to predict the most probable path that generated the sequence. An example of an input sequence is **AGCTTAACG**. The latent variables $\mathbf{z}$ encode whether a particular base belongs to a protein coding region or not and $\mathbf{z} \in \{\mathbf{P}, \mathbf{N}\}$. The graphical model corresponding to this setup is shown below.

(d) We observe that the path P (states of latent variables z) given by $\{\mathbf{N\ N\ P\ P\ P\ P}\}$ produces the sequence S given by $\{\mathbf{G\ C\ T\ G\ G\ C}\}$. What is the probability that the HMM described above produced sequence S by the path P?

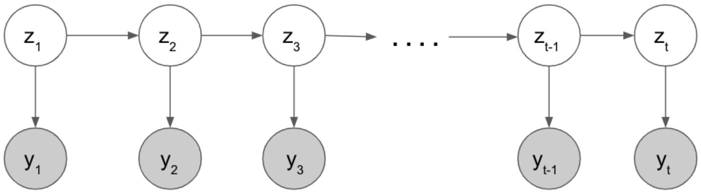


Figure 3: z_i refers to the latent state and y_i is the observed base at the i^{th} position

To generate such sequence {**N N P P P P**}, the probability of such sequence can be computed as following:
$P\{N,N,P,P,P,P\} = P(z_6 = P|z_5 = P)\ P(z_5 = P|z_4 = P)\ P(z_4 = P|z_3 = P)\ P(z_3 = P|z_2 = N)\ P(z_2 = N|z_1 = N)\ P(z_1 = N)$
$P\{N,N,P,P,P,P\} = 0.7 * 0.7 * 0.7 * 0.1 * 0.8 * 0.5 = 0.01372$

Based on the state sequence {**N N P P P P**}, we can compute the probability of S sequence as following:
$P\{G,C,T,G,G,C\} = P(y_1 = G|z_1 = N)\ P(y_2 = C|z_2 = N)\ P(y_3 = T|z_3 = P)\ P(y_4 = G|z_4 = P)\ P(y_5 = G|z_5 = P)\ P(y_6 = C|z_3 = P)$
$P\{G,C,T,G,G,C,\} = 0.2 * 0.2 * 0.2 * 0.3 * 0.3 * 0.3 = 2.16 * 10^{-4}$

Thus we can compute the total probability of sequence as following:
$P(\mathbf{S,P}) = 0.01372 * 2.16 * 10^{-4} = 2.96352 * 10^{-6}$

(e) We observe the sequence {**G C T A A C**} and we would like to determine the most likely path that produced this sequence and the probability associated with this path. To do this we can use max-product or in this case because we're working with a HMM, Viterbi decoding. Viterbi decoding relies on the following recursion by noting that the probability of the most probable path for the t^{th} observation to be $i \in \{A,C,G,T\}$, given that it was in state $k \in \{P,N\}$ depends on the most probable path for the $(t-1)^{th}$ observation for base $j \in \{A,C,G,T\}$.

$$\begin{aligned}\text{Probability of the most probable path} &= p_k(t = i)\\ &= e_k(i)\max_s[p_s(t-1 = j) \cdot p_{sk}]\end{aligned}$$

Where $e_k(i)$ is the probability that base i is observed when the latent state is $k \in \{P,N\}$. Specify the most probable path and the associated probability below. Show your work. Hint: Refer to the ice cream and weather example by Jason Eisner.

For $y_1 = G$, $p_P(y_1 = G) = 0.3$ and $p_N(y_1 = G) = 0.2$. Thus, $z_1 = P$.
For $y_2 = C$, $p_P(y_2 = C) = 0.3 * \max(0.3 * 0.7, 0.2 * 0.1) = 0.063$ and $p_N(y_2 = C) = 0.2 * \max(0.3 * 0.2, 0.2 * 0.8) = 0.032$. Thus $z_2 = P$.
For $y_3 = T$, $p_P(y_3 = T) = 0.2 * \max(0.063 * 0.7, 0.032 * 0.1) = 0.00882$ and $p_N(y_3 = T) = 0.3 * \max(0.063 * 0.2, 0.032 * 0.8) = 0.00768$. Thus $z_3 = P$.

For $y_4 = A$, $p_P(y_4 = A) = 0.2 * \max(0.00882 * 0.7, 0.00768 * 0.1) = 1.2348 * 10^{-3}$ and $p_N(y_4 = A) = 0.3 * \max(0.00882 * 0.2, 0.00768 * 0.8) = 1.8432 * 10^{-3}$. Thus $z_4 = N$.

For $y_5 = A$, $p_P(y_5 = A) = 0.2 * \max(0.0012348 * 0.7, 0.0018432 * 0.1) = 1.72872 * 10^{-4}$ and $p_N(y_5 = A) = 0.3 * \max(0.0012348 * 0.2, 0.0018432 * 0.8) = 4.42368 * 10^{-4}$. Thus $z_5 = N$.

For $y_6 = C$, $p_P(y_6 = C) = 0.3 * \max(0.000172872 * 0.7, 0.000442368 * 0.1) = 3.630312 * 10^{-5}$ and $p_N(y_5 = C) = 0.2 * \max(0.000172872 * 0.2, 0.000442368 * 0.8) = 7.077888 * 10^{-5}$. Thus $z_6 = N$.

Based on the Viterbi Decoding, the path is **P, P, P, N, N, N**

2) Belief propagation in Factor graphs

Recall, that we discussed for an undirected graphical model, the factorization of the joint probability distribution specified by a particular graph is not unique. One such factorization relies on the identification of maximal cliques. For a graph $G = V, E$, a complete subgraph G'' satisfies the condition $G'' = \{V' \subseteq, E' \subseteq E\}$ such that the nodes in V' are fully interconnected. A (maximal) clique is a complete subgraph such that any $V'' \supset V'$ is not complete. A sub-clique is not necessarily a maximal clique. The maximal clique factorization can be used to convert an undirected graph to a factor graph.

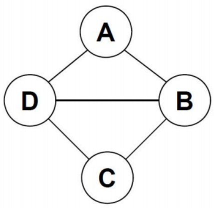


Figure 4: The maximal cliques in this case are given by $\{A, B, D\}$ and $\{B, C, D\}$

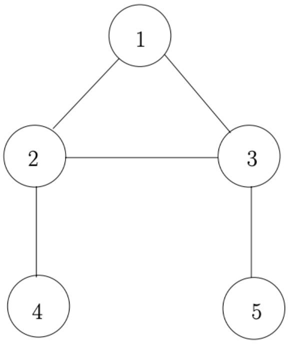
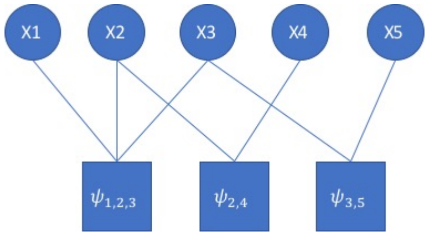


Figure 5: Undirected graph with 5 random variables

(a) Consider the undirected graph shown in Fig 5. Identify the maximal cliques and write a factor graph that represents the joint distribution of the product of the factors given by maximal cliques. In this example, why can we use factor graph message passing equations to compute the marginals but not sum product? Note: You can draw the graph using your favorite text/ slide editor and include as a .jpg (or whatever file format is easiest).



Maximal Clique: $\{x_1, x_2, x_3\}, \{x_2, x_4\}, \{x_3, x_5\}$
$P(x_1, x_2, x_3, x_4, x_5) = \frac{1}{Z}\psi_{1,2,3}(x_1, x_2, x_3)\psi_{2,4}(x_2, x_4)\psi_{3,5}(x_3, x_5)$

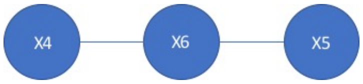
If there is a loop in the graph, the factor inside the loop wouldn't propagate the messages, noticed that sum-product algorithm the factor to propagate the messages until it receives messages from all neighbors, which is impossible in the the case with loop, because the the neighbors might also be waiting for the message from that specific factor. Thus the sum-product algorithm cannot be applied in this case.

(b) For the factor graph that you obtained in part (a) write out all the messages that would be calculated by the sum product algorithm. Recall that in order for a factor or a node to pass messages, it should have received messages from all neighbors but one. Therefore begin with the message passed by the leaf nodes. Use $\psi_{i,j}(x_i, x_j)$ to denote the factors that are connected to nodes i, j and $\nu_{i\to f}(x_i)$ to denote a message passed from node i to factor f. Each variable is discrete with K classes. Write the messages in terms of factors and previous messages. If you write your messages as the full distribution over the K classes, you may also need to include K in your equations.

Assume x_1 is the root, then we will have x_4 and x_5 as leaves.
From leaves to root:
$\nu_{x_4\to\psi_{2,4}}(x_4) = 1 \qquad \nu_{x_5\to\psi_{3,5}}(x_5) = 1$
$\mu_{\psi_{2,4}\to x_2}(x_2) = \sum_{x_4} \psi_{2,4}(x_2, x_4)\ \nu_{x_4\to\psi_{2,4}}(x_4)$
$\mu_{\psi_{3,5}\to x_3}(x_3) = \sum_{x_5} \psi_{3,5}(x_3, x_5)\ \nu_{x_5\to\psi_{3,5}}(x_5)$
$\nu_{x_2\to\psi_{1,2,3}}(x_2) = \mu_{\psi_{2,4}\to x_2}(x_2) \qquad \nu_{x_3\to\psi_{1,2,3}}(x_3) = \mu_{\psi_{3,5}\to x_3}(x_3)$
$\mu_{\psi_{1,2,3}\to x_1}(x_1) = \sum_{x_2,x_3} \psi_{1,2,3}(x_1, x_2, x_3)\ \nu_{x_2\to\psi_{1,2,3}}(x_2)\ \nu_{x_3\to\psi_{1,2,3}}(x_3)$

From root to leaves:
$\nu_{x_1\to\psi_{1,2,3}}(x_1) = 1$
$\mu_{\psi_{1,2,3}\to x_2}(x_2) = \sum_{x_1,x_3} \psi_{1,2,3}(x_1, x_2, x_3)\ \nu_{x_1\to\psi_{1,2,3}}(x_1)\ \nu_{x_3\to\psi_{1,2,3}}(x_3)$
$\mu_{\psi_{1,2,3}\to x_3}(x_3) = \sum_{x_1,x_2} \psi_{1,2,3}(x_1, x_2, x_3)\ \nu_{x_1\to\psi_{1,2,3}}(x_1)\ \nu_{x_2\to\psi_{1,2,3}}(x_2)$
$\nu_{x_2\to\psi_{2,4}}(x_2) = \mu_{\psi_{1,2,3}\to x_2}(x_2)$
$\nu_{x_3\to\psi_{3,5}}(x_3) = \mu_{\psi_{1,2,3}\to x_3}(x_3)$
$\mu_{\psi_{2,4}\to x_4}(x_4) = \sum_{x_2} \psi_{2,4}(x_2, x_4)\ \nu_{x_2\to\psi_{2,4}}(x_2)$
$\mu_{\psi_{3,5}\to x_5}(x_5) = \sum_{x_3} \psi_{3,5}(x_3, x_5)\ \nu_{x_3\to\psi_{3,5}}(x_3)$

(c) Construct a new random variable given by $x_6 = \{x_1, x_2, x_3\}$, we group the three random variables x_1, x_2, x_3 into one. Now draw an undirected graph that captures the relationship between x_6, x_4 and x_5. Explain why the sum-product algorithm can be used to compute marginals now. Note: You can draw the graph using your favorite text/ slide editor and include as a .jpg (or whatever file format is easiest).



Now that we don't have a loop in our graph, which mean we won't have the waiting issues during the message propagation. Additionally, this graph is the tree-like graph, which is suitable for sum-product algorithm. Thus we can use sum-product algorithm in this graph.

(d) Write the sum product algorithm for graph you obtained in part (c). Note: $\psi_{5,6}(x_5, x_6) = \psi_{5,6}(x_1, x_2, x_3, x_6)$. Compare the message passing equations that you obtained using sum product to the message passing you derived for the factor graph in (b).

Assume x_4 is the root, then we will have x_5 as leave.
From leaves to root:
$\nu_{x_5\to\psi_{5,6}}(x_5) = 1$
$\mu_{\psi_{5,6}\to x_6}(x_6) = \sum_{x_5} \psi_{5,6}(x_5, x_6)\ \nu_{x_5\to\psi_{5,6}}(x_5) = \sum_{x_5} \psi_{5,6}(x_5, x_1, x_2, x_3)\ \nu_{x_5\to\psi_{5,6}}(x_5)$
$\nu_{x_6\to\psi_{4,6}}(x_6) = \mu_{\psi_{5,6}\to x_6}(x_6)$
$\mu_{\psi_{4,6}\to x_4}(x_4) = \sum_{x_6} \psi_{4,6}(x_4, x_6)\ \nu_{x_6\to\psi_{4,6}}(x_6) = \sum_{x_1,x_2,x_3} \psi_{4,6}(x_4, x_1, x_2, x_3)\ \nu_{x_6\to\psi_{4,6}}(x_6)$

From root to leaves:
$\nu_{x_4\to\psi_{4,6}}(x_4) = 1$
$\mu_{\psi_{4,6}\to x_6}(x_6) = \sum_{x_4} \psi_{4,6}(x_4, x_6)\ \nu_{x_4\to\psi_{4,6}}(x_4) = \sum_{x_4} \psi_{4,6}(x_4, x_1, x_2, x_3)\ \nu_{x_4\to\psi_{4,6}}(x_4)$
$\nu_{x_6\to\psi_{4,6}}(x_6) = \mu_{\psi_{4,6}\to x_6}(x_6)$
$\mu_{\psi_{5,6}\to x_5}(x_5) = \sum_{x_6} \psi_{5,6}(x_5, x_6)\ \nu_{x_6\to\psi_{5,6}}(x_6) = \sum_{x_1,x_2,x_3} \psi_{5,6}(x_5, x_1, x_2, x_3)\ \nu_{x_6\to\psi_{5,6}}(x_6)$

(e) Taking the approach in (c) to the extreme, we can group all the variables in the graph to create a new random variable $x_7 = \{x_1, x_2, x_3, x_4, x_5\}$. Assuming that we only care about the marginals of x_1, x_2, x_3, x_4 and x_5, why would we prefer the method proposed in part (c) to grouping all the variables, i.e. why might it be preferable to group a smaller number of variables together?

If we create a new random variable $x_7 = \{x_1, x_2, x_3, x_4, x_5\}$, we consider the whole graph is a node, we cannot use sum-product algorithm anymore. Thus we need to use somehow the brute force method to compute the probability of specific node, which is expensive and inefficient. By using the brute force method, it will take $O(K^5)$ operations to compute the marginal of $\{x_1, x_2, x_3, x_4, x_5\}$. Thus it better to group a smaller number variables together.

3) Graphical modeling and inference

Consider the biological model which is represented by the following directed graphical model. G_i refers to the genotype of an individual. $G_i = 1$ if the individual has a healthy copy of the gene and $G_i = 2$ if the copy is unhealthy. G_1 is the genotype of the parent, while G_2 and G_3 correspond to the genotypes of the children. $X_i \in \mathbb{R}$ corresponds to the phenotype of interest, in this case BMI. Healthy individuals have $BMI \leq 25$, while a $BMI \geq 30$ is considered unhealthy. (We will make a simplifying assumption that it's not unhealthy to have a BMI value that is too low, although it is). We define the conditional probability distributions as follows.

$$p(G_1) = [0.5, 0.5] \tag{9}$$

$$p(G_2|G_1) = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix} \tag{10}$$

$$p(G_3|G_1) = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix} \tag{11}$$

$$p(X_i|G_i = 1) = \mathcal{N}(\mathbf{X}_i|\mu = 21, \sigma^2 = 3) \tag{12}$$

$$p(X_i|G_i = 2) = \mathcal{N}(\mathbf{X}_i|\mu = 24, \sigma^2 = 3) \tag{13}$$

The meaning of the matrix for $p(G_2 = 1|G_1 = 1) = 0.8$ and $p(G_2 = 1|G_1 = 2) = 0.2$ etc.

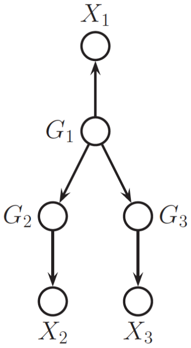


Figure 6: Directed graphical model explaining the relationship between the genotypes and phenotypic dependence on genotype

(a) Suppose we observe $X_2 = 21$, and X_1 is unobserved, what is the posterior belief on G_1, i.e. $p(G_1|X_2 = 21)$?

$p(G_1|X_2 = 21) = \frac{p(X_2=21|G_1)\ p(G_1)}{p(X_2=21)} = \frac{\sum_{G_2} p(X_2=21|G_2)\ p(G_2|G_1)\ p(G_1)}{\sum_{G_2} p(X_2=21|G_2)\ p(G_2)}$

From $p(G_2) = \sum_{G_1} p(G_2|G_1)\ p(G_1)$, we can obtain that $p(G_2 = 1) = p(G_2 = 2) = 0.5$

$$p(G_1 = 1|X_2 = 21) = \frac{\sum_{G_2} p(X_2 = 21|G_2)\ p(G_2|G_1 = 1)}{\sum_{G_2} p(X_2 = 21|G_2)} \approx 0.691$$

$$p(G_1 = 2|X_2 = 21) = 1 - p(G_1 = 1|X_2 = 21) = 0.309$$

(b) Now you observe that $X_2 = 21$ and $X_3 = 21$. What is $p(G_1|X_2, X_3)$? Explain how your answer is different from what you obtained in part (a)?

From the fact that X_2 and X_3 are conditional independent, which implies that $p(X_2 = 21, X_3 = 21|G_1) = p(X_2 = 21|G_1)\ p(X_3 = 21|G_1)$ thus we can rewrite $p(G_1|X_2, X_3)$ as following form:
$p(G_1|X_2 = 21, X_3 = 21) = \frac{p(X_2=21,X_3=21|G_1)\ p(G_1)}{p(X_2=21,X_3=21)} = \frac{p(X_2=21|G_1)\ p(X_3=21|G_1)\ p(G_1)}{\sum_{G_1} p(X_2=21|G_1)\ p(X_3=21|G_1)\ p(G_1)}$
$p(G_1 = 1|X_2 = 21, X_3 = 21) = \frac{p(X_2=21|G_1=1)\ p(X_3=21|G_1=1)\ p(G_1=1)}{\sum_{G_1} p(X_2=21|G_1)\ p(X_3=21|G_1)\ p(G_1)}$
Notice that $\frac{p(X_2=21|G_1=2)}{p(X_2=21|G_1=1)} = \frac{p(G_1=2|X_2=21)}{p(G_1=1|X_2=21)} \approx 0.448$ and $p(G_3|G_1) = p(G_2|G_1)$, thus,
$p(G_1 = 1|X_2 = 21, X_3 = 21) = \frac{1}{1+\frac{p(X_2=21|G_1=2)\ p(X_3=21|G_1=2)}{p(X_2=21|G_1=1)\ p(X_3=21|G_1=1)}} \approx 0.833$
$p(G_1 = 2|X_2 = 21, X_3 = 21) = 1 - p(G_1 = 1|X_2 = 21, X_3 = 21) = 0.167$
We can see that the probability of $G_1 = 1$ is larger, which implies that we are more confident with both children are healthy. With the face that (b) case is a sub-case of (c) case, the (b) case also take case $X_2 = 21$ and $X_3 \neq 21$ into account, which will decrease the probability of $G_1 = 1$.

(c) Now you observe that $X_2 = 21$ and $X_3 = 24$. What is $p(G_1|X_2, X_3)$? Compare to the answers you obtained in (a) and (b), are you more or less certain about the genotype of the parent when the children exhibit different phenotypes?

From above analysis, we can also obtain that $(G_1 = 1|X_3 = 24) = 0.309$ and $(G_1 = 2|X_3 = 24) = 0.691$, thus we can also obtain that:
$p(G_1 = 1|X_2 = 21, X_3 = 24) = 0.5$
$p(G_1 = 2|X_2 = 21, X_3 = 24) = 1 - p(G_1 = 1|X_2 = 21, X_3 = 24) = 0.5$
when the children exhibit different phenotypes, we are less certain about the genotype of the parent.