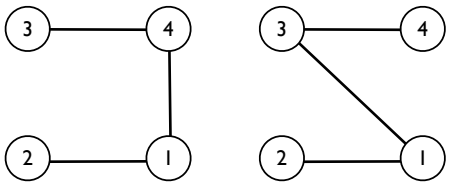
IE 598 Inference in Graphical Models

Homework 2

Problem 2.1

In this problem, we will show that when the distribution $\mu(x)$ is not strictly positive (i.e. $\mu(x) = 0$ for some x), then the I-map for this distribution is not unique. Consider a distribution of 4 binary random variables x_1, x_2, x_3, and x_4 such that $\mu(x_1 = x_2 = x_3 = x_4 = 1) = 0.5$ and $\mu(x_1 = x_2 = x_3 = x_4 = 0) = 0.5$. The following two undirected graphical models are both minimal I-maps for this distribution, hence it is not unique.



(*a*) Prove that the two undirected graphical models above are minimal I-maps for the distribution $\mu(x)$. You need to show that both graphs are I-maps for the given distribution $\mu(x)$ and that removing any edge results in introducing independencies that are not implied by the distribution $\mu(x)$.

(*b*) Now, we show that starting with a complete graph and eliminating edges that are pairwise conditionally independent does not always give you an I-map (minimal or not). Start with a complete graph K_4. For each pair of nodes, eliminate the edge between this pair if they are conditionally independent given the rest of the nodes in the graph. Continue this procedure for all pairs of nodes and examine the resulting graph. Is this an I-map of the distribution $\mu(x_1, x_2, x_3, x_4)$?

Recall from class, that a distribution over x is (globally) Markov with respect to $G = (V, E)$ if, for any disjoint subsets of nodes A, B, C such that B separates A from C, x_A–x_B–x_C is satisfied. Recall two other notions of Markovity. A distribution is pairwise Markov with respect to G if, for any two nodes i and j not directly linked by an edge in G, the corresponding variables x_i and x_j are independent conditioned on all of the remaining variables, i.e. for all $(i, j) \notin E$,

$$x_i – x_{V\setminus\{i,j\}} – x_j$$

A distribution is locally Markov with respect to G if any node i, when conditioned on the variables on the neighbors of i, is independent of the remaining variables, i.e. for all $i \in V$,

$$x_i – x_{\partial i} – x_{V\setminus\{i,\partial i\}}$$

(*c*) Using the example of distribution on 4 random variables as a counter example, prove that a distribution is pairwise Markov w.r.t. G does not always imply that it is locally Markov w.r.t. the same graph G. (However, if the distribution is positive, pairwise Markovity implies local and global Markovity.)

(*d*) Using the definitions of Markov properties, prove that if a distribution is globally Markov with respect to G, then it is locally Markov with respect to G.

(*e*) (Optional) Using the definitions of Markov properties, prove that if a distribution is locally Markov with respect to G, then it is pairwise Markov with respect to G.

Solution 2.1

(*a*) The (set of) independencies implies by the first graph G_1 is x_3–x_4–$\{x_1, x_2\}$, x_3–$\{x_4, x_1\}$–x_2, x_4–x_1–x_2. Since this is included in $\mathcal{I}(\mu)$, G_1 is an I-map of μ.

The (set of) independencies implies by the second graph G_2 is x_4–x_3–$\{x_1, x_2\}$, x_4–$\{x_1, x_3\}$–x_2, x_3–x_1–x_2. Since this is included in $\mathcal{I}(\mu)$, G_1 is an I-map of μ.

Further, if we remove an edge from either G_1 or G_2, the graphs are separated into two disjoint. This implies that there are two sets of nodes which are independent, which is not true for the given μ. Hence, bothe graphs are minimal I-maps of μ.

(*b*) The resulting graph is a graph of 4 nodes with no edges. This graph implies, for instance, that x_1 and x_2 are independent, which is not true for the given μ. Hence, this is not an I-map.

(*c*) The given distribution $\mu(x)$ is pairwise Markov with respect to the graph in (*b*). However, it is not globally Markov w.r.t. that graph. This proves that pairwise Markovity does not imply global Markovity.

(*d*) Using the definition of global Markovity, let $A = i$ and $B = \partial i$. Then it follows from the definition of ∂i that B separates A from the rest of the graph. Hence, x_i–$x_{\partial i}$–$x_{V\setminus\{i,\partial i\}}$. Since this is true for any choice of i, this proves that mu is also locally Markov w.r.t. G.

(*e*) First, note that x_i–$x_{\partial i}$–$x_{V\setminus\{i\}\setminus\partial i}$ implies x_i–$x_{\partial i}$–$x_{V\setminus\{i,j\}\setminus\partial i}$. Then, by local Markovity,

$$\begin{aligned}\mu(x_i, x_j|x_{V\setminus\{i,j\}}) &= \mu(x_j|x_{V\setminus\{i,j\}})\mu(x_i|x_{V\setminus\{i\}}) \\ &= \mu(x_j|x_{V\setminus\{i,j\}})\mu(x_i|x_{\partial i}) \\ &= \mu(x_j|x_{V\setminus\{i,j\}})\mu(x_i|x_{V\setminus\{i,j\}})\end{aligned}$$

this implies that local Markovity implies pairwise Markovity.

Problem 2.2

Consider a stochastic process that transitions among a finite set of states $s_1, \ldots, s_k$ over time steps $i = 1, \ldots, N$. The random variables $X_1, \ldots, X_N$ representing the state of the system at each time step are generated as follows:

- Sample the initial state $X_1 = s$ from an initial distribution p_1, and set $i := 1$.
- Repeat the following:
  - Sample a duration d from a duration distribution p_D over the integers $\{1, \ldots, M\}$, where M is the maximum duration.
  - Remain in the current state s for the next d time steps, i.e., set
$$X_i := X_{i+1} := \ldots := X_{i+d-1} := s$$
  - Sample a successor state s' from a transition distribution $p_T(\cdot|s)$ over the other states $s' \neq s$ (so there are no sef-transitions).
  - Assign $i := i + d$ and $s := s'$.

This process continues indefinitely, but we only observe the first N time steps. You need not worry about the end of the sequence to do any of the problems. As an example calculation with this model, the probability of the sample state sequence $s_1, s_1, s_1, s_2, s_3, s_3$ is

$$p_1(s_1)p_D(3)p_T(s_2|s_1)p_D(1)p_T(s_3|s_2) \sum_{2 \geq d \leq M} p_D(d) \; .$$

Finally, we do not directly observe the X_i's, but instead observe emissions y_i at each step sampled from a distribution $p_{Y_i|X_i}(y_i|x_i)$.

(*a*) For this part only, suppose $M = 2$, and $p_D(d) = \begin{cases} 0.6 & \text{for } d = 1 \\ 0.4 & \text{for } d = 2 \end{cases}$, and each X_i takes on a value from an alphabet $\{a, b\}$. Draw a minimal directed I-map for the first five time steps using the variables $(X_1, \ldots, X_5, Y_1, \ldots, Y_5)$. Explain why none of the edges can be removed.
[Note: you do not need to solve part (*a*) in order to solve part (*b*) and (*c*).]

(*b*) This process can be converted to an HMM using an *augmented state representation*. In particular, the states of this HMM will correspond to pairs (x, t), where x is a state in the original system, and t represents the time elapsed in that state. For instance, the state sequence $s_1, s_1, s_1, s_2, s_3, s_3$ would be represented as $(s_1, 1), (s_1, 2), (s_1, 3), (s_2, 1), (s_3, 1), (s_3, 2)$. the transition and emission distribution for the HMM take the forms

$$\tilde{p}_{X_{i+1}, T_{i+1}|X_i, T_i}(x_{i+1}, t_{i+1}|x_i, t_i) = \begin{cases} \phi(x_i, x_{i+1}, t_i) & \text{if } t_{i+1} = 1 \text{ and } x_{i+1} \neq x_i \\ \xi(x_i, t_i) & \text{if } t_{i+1} = t_i + 1 \text{ and } x_{i+1} = x_i \\ 0 & \text{otherwise} \end{cases}$$

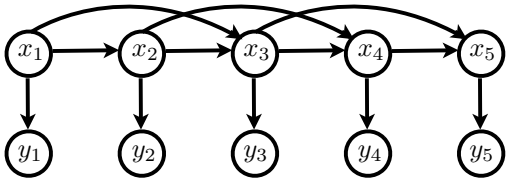
and $\tilde{p}_{Y_i|X_i,T_i}(y_i|x_i, t_i)$, respectively. Express $\phi(x_i, x_{i+1}, t_i)$, $\xi(x_i, t_i)$, and $\tilde{p}_{Y_i|X_i,T_i}(y_i|x_i, t_i)$ in terms of parameters p_1, p_D, p_T, $p_{Y_i|X_i}$, k, N, and M of the original model.

(*c*) We wish to compute the marginal probability for the final state X_N given the observations $Y_1, \ldots, Y_N$. If we naively apply the sum-product algorithm to the construction in part (*b*), the computational complexity is $O(Nk^2M^2)$. Show that by exploiting additional structure in the model, it is possible to reduce the complexity to $O(N(k^2+kM))$. In particular, give the corresponding rules for computing the forward messages $\nu_{i+1 \to i+2}(x_{i+1}, t_{i+1})$ from the previous message $\nu_{i \to i+1}(x_i, t_i)$. Do not worry about

the beginning or the end of the sequence and restrict your attention to $2 \leq i \leq N-1$.
[Hint: substitute your solution from part (b) into the standard update rule for HMM messages and simplify as much as possible.]
[Note: If you cannot fully solve this part of the problem, you can receive substantial partial credit by constructing an algorithm with complexity $O(Nk^2M)$.]

Solution 2.2

(a) Here is the minimal directed I-map:



Consider the factorization

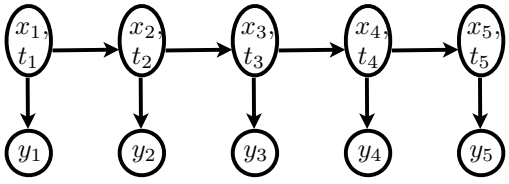
$$p_{X_1,Y_2,\ldots,X_N,Y_N} = p_{X_1}\, p_{Y_1|X_1}\, p_{X_2|X_1,Y_1}\, p_{Y_2|X_1,Y_1,X_2} \cdots p_{Y_N|X_1,Y_1,\ldots,Y_{N-1},X_N}$$

The distribution $p_{X_i|X_1,\ldots,X_{i-1},Y_1,\ldots,Y_{i-1}}$ depends only on $\{X_{i-M},\ldots,X_{i-1}\}$ through the transition distributions and duration distributions. Similar to the HMMs, the distribution $p_{Y_i|X_1,\ldots,X_i,Y_1,\ldots,Y_{i-1}}$ depends only on X_i. This yields the DAG above.

The arrows $X_i \to Y_i$ are necessary because the observations depend on the states. As for the arrows between states, the following table for $p_{X_i|X_{i-1},X_{i-2}}$ shows that neither the one-step nor the two-step edges can be removed for $i \geq 3$.

x_{i-2}	x_{i-1}	$p_{X_1\|X_{i-2},X_{i-1}}(b\|x_{i-1},x_{i-1})$
a	a	1
a	b	0.6
b	a	0.4
b	b	0

(b) The HMM with augmented state representation is
First note that the probability of switching after m-th consecutive time step in a given state is



$$S(m) \quad = \quad \mathbb{P}(d = m | d \geq m) = \frac{p_D(m)}{\sum_{i=1}^{m} p_D(i)}$$

Then,

$$\tilde{p}_{X_{i+1},T_{i+1}|X_i,T_i}(x_{i+1},t_{i+1}|x_i,t_i) = \begin{cases} p_T(x_{i+1}|x_i)S(t_i) & \text{if } t_{i+1}=1 \text{ and } x_{i+1} \neq x_i \\ \mathbb{P}(D \geq t_i+1 | D \geq t_i) & \text{if } t_{i+1} = t_i+1 \text{ and } x_{i+1} = x_i \\ 0 & \text{otherwise} \end{cases}$$

This gives

$$\begin{aligned}
\phi(x_i, x_{i+1}, t_i) &= \mathbb{P}(D = t_i \mid D \geq t_i) p_T(x_{i+1}|x_i) = \frac{p_D(t_i)}{\sum_{d \geq t_i} p_D(d)} p_T(x_{i+1}|x_i) \\
\xi(x_i, t_i) &= \mathbb{P}(D \geq t_i + 1 \mid D \geq t_i) = \frac{\sum_{d \geq t_i + 1} p_D(d)}{\sum_{d \geq t_i} p_D(d)} \\
\tilde{p}_{Y_i|X_i,T_i}(y_i|x_i, t_i) &= p_{Y_i|X_i}(y_i|x_i)
\end{aligned}$$

(*c*) For HMMs, the forward BP update rule is

$$\nu_{i+1 \to i+2}(x_{i+1}, t_{i+1}) = \tilde{p}_{Y_{i+1}|X_{i+1},T_{i+1}}(y_{i+1}|x_{i+1}, t_{i+1}) \sum_{x_i} \sum_{t_i} \tilde{p}_{X_{i+1},T_{i+1}|X_i,T_i}(x_{i+1}, t_{i+1}|x_i, t_i) \nu_{i \to i+1}(x_i, t_i)$$

Substituting the formula in part (*b*), we get

$$\nu_{i+1 \to i+2}(x_{i+1}, t_{i+1}) = \tilde{p}_{Y_{i+1}|X_{i+1}}(y_{i+1}|x_{i+1}) \sum_{x_i} \sum_{t_i} \left\{ \mathbb{I}_{(t_{i+1}=1)} \mathbb{I}_{(x_{i+1} \neq x_i)} \phi(x_i, x_{i+1}, t_i) + \mathbb{I}_{(t_{i+1}=t_i+1)} \mathbb{I}_{(x_{i+1}=x_i)} \xi(x_i, t_i) \right\} \nu_{i \to i+1}(x_i, t_i)$$

To sum out the terms involving $\xi(x_i, t_i)$, it requires $O(kM)$ operations, since we need to consider all combinations of $x_{i+1} \in \{1, \ldots, k\}$ and $t_{i+1} \in \{2, \ldots, M\}$ and for each combination we need to evaluate $\xi(x_{i+1}, t_{i+1} - 1)\nu_{i \to i+1}(x_{i+1}, t_{i+1} - 1)$ once.

If we naively sum out the terms involving $\phi(x_i, x_{i+1}, t_i)$, it requires $O(k^2 M)$ operations, since we can fix $t_{i+1} = 1$ and consider all $x_{i+1} \in \{1, \ldots k\}$ and sum over all $x_i \neq x_{i+1}$ and $t_i \in \{1, \ldots, M\}$.

Further speedup can be achieved by the factorization of $\phi(x_i, x_{i+1}, t_i) = a(t_i)b(x_i, x_{i+1})$. Then the first term becomes

$$\begin{aligned}
&\tilde{p}_{Y_{i+1}|X_{i+1}}(y_{i+1}|x_{i+1}) \sum_{x_i} \sum_{t_i} \mathbb{I}_{(t_{i+1}=1)} \mathbb{I}_{(x_{i+1} \neq x_i)} \phi(x_i, x_{i+1}, t_i) \nu_{i \to i+1}(x_i, t_i) \\
&= \tilde{p}_{Y_{i+1}|X_{i+1}}(y_{i+1}|x_{i+1}) \mathbb{I}_{(t_{i+1}=1)} \sum_{x_i \neq x_{i+1}} \sum_{t_i} a(t_i) b(x_i, x_{i+1}) \nu_{i \to i+1}(x_i, t_i) \\
&= \tilde{p}_{Y_{i+1}|X_{i+1}}(y_{i+1}|x_{i+1}) \mathbb{I}_{(t_{i+1}=1)} \sum_{x_i \neq x_{i+1}} b(x_i, x_{i+1}) \sum_{t_i} a(t_i) \nu_{i \to i+1}(x_i, t_i)
\end{aligned}$$

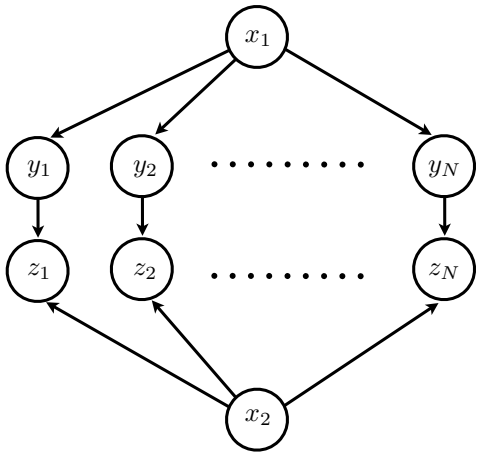
The inner summation does not depend on x_{i+1} and can be computed in $O(kM)$ operations. Then, the outer summation requires $O(k)$ operations and must be computed for each x_{i+1}, requiring total $O(k^2)$ operations. Hence, the total running time for the forward update is $O(N(k^2 + kM))$.

Problem 2.3

Consider random variables X_1, X_2, $Y_1, \ldots, Y_N$, $Z_1, \ldots, Z_N$ distributed according to

$$p_{X_1,X_2,Y,Z}(x_1, x_2, y, z) = p_{X_1}(x_1)p_{X_2}(x_2)\prod_{i=1}^{N}\left[p_{Y|X_1}(y_i|x_1)p_{Z|Y,X_2}(z_i|y_i, x_2)\right],$$

where $X_1, Y_1, \ldots, Y_N, Z_1, \ldots, Z_N$ take on values in $\{1, 2, \ldots, K\}$ and X_2 instead takes on a value in $\{1, 2, \ldots, N\}$. A minimal directed I-map for the distribution is as follows:



Assume throughout this problem that the complexity of table lookups for p_{X_1}, p_{X_2}, $p_{Y|X_1}$, and $p_{Z|Y,X_2}$ are $O(1)$.

(a) A Bayesian network represented by a directed acyclic graph can be turned into a Markov random field by *moralization*. The moralized counterpart of a directed acyclic graph is formed by connecting all pairs of nodes that have a common child, and then making all edges in the graph undirected. Draw the moral graph over random variables $X_1, X_2, Y_1, \ldots, Y_N$ conditioned on $Z_1, \ldots, Z_N$. In other words, find an undirected I-map for the distribution of random variables $X_1, X_2, Y_1, \ldots, Y_N$ conditioned on $Z_1, \ldots, Z_N$.

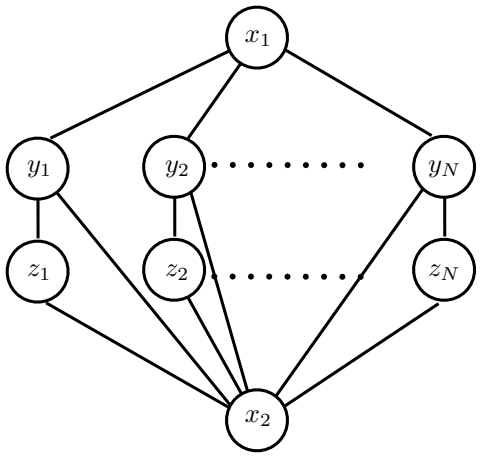
Provide a good elimination ordering for computing marginals of $X_1, X_2, Y_1, \ldots, Y_N$ conditioned on $Z_1, \ldots, Z_N$. For your elimination ordering, determine α and β such that complexity of computing $p_{X_1|Z_1,\ldots,Z_N}$ using the associated elimination algorithm is $O(N^\alpha K^\beta)$.

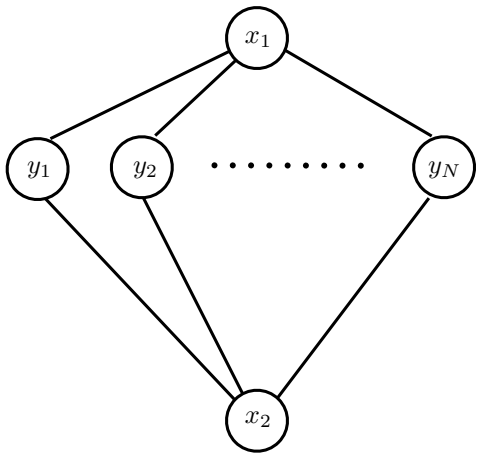
(b) For the remainder of this problem, suppose that we also have the following *context-dependent* conditional independencies: Y_i is conditionally independent of Z_i given $X_2 = c$ for all $i \neq c$. For fixed $z_1, \ldots, z_N$, x_1, and c, show that

$$p_{Z_1,\ldots,Z_N|X_1,X_2}(z_1 \ldots, z_N|x_1, c) = \eta(x_1, c, z_c)\lambda(c, z_1, \ldots, z_{c-1}, z_{c+1}, \ldots, z_N)$$

for some function $\eta(x_1, c, z_c)$ that can be evaluated in $O(K)$ operations for fixed (x_1, c, z_c), and some function $\lambda(c, z_1, \ldots, z_N)$ that can be evaluated in $O(N)$ operations for fixed $(c, z_1, \ldots, z_N)$. Express $\eta(x_1, c, z_c)$ in terms of $p_{Y|X_1}$ and $p_{Z|Y,X_2}$, and $\lambda(c, z_1, \ldots, z_{c-1}, z_{c+1}, \ldots, z_N)$ in terms of $p_{Z|X_2}$.

Solution 2.3

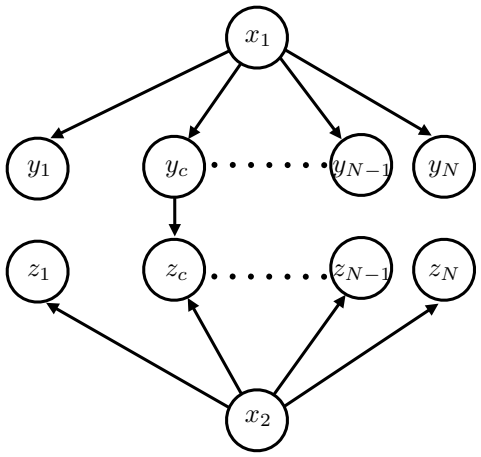




(a) The moral graph of random variables $(x_1, x_2, y_1, \ldots, y_N, z_1, \ldots, z_N)$ is
Conditioning on (a specific realization of) $z_1, \ldots, z_N$, we get the following undirected graph.
Now, consider an elimination ordering $I = (Y_1, \ldots, Y_N, X_2, X_1)$. For this elimination ordering, computing the marginal $p_{X_1|Z_1,\ldots,Z_N}$ requires summing out variables in the elimination order. Summing out Y_i requires $O(NK^2)$ operations, since for each Y_i we need to sum over K values of Y_i and we need to repeat this for all instances of $X_1 \in \{1, \ldots, K\}$ and $X_2 \in \{1, \ldots, N\}$. Further, we need to repeat this for each $\{Y_i\}_{i\in\{1,2,\ldots,N\}}$. Hence, the overall computational complexity for the elimination ordering is $O(N^2K^2)$,i.e., $\alpha = 2$, $\beta = 2$.

(b) The I-map for random variables $((x_1, x_2, y_1, \ldots, y_N, z_1, \ldots, z_N))$ given $x_2 = c$ is
By the context-dependent conditional independencies, we have $p_{Z|Y,X_2}(z_i|y_i, c) = p_{Z|X_2}(z_i|c)$ for $i \neq c$. Then,

$$p_{Y,Z_1,\ldots,Z_N|X_1,X_2}(y_c, z_1, \ldots, z_N|x_1, x_2 = c) \quad = \quad p_{Y|X_1}(y_c|x_1)p_{Z|Y,X_2}(z_c|y_c, c) \prod_{i=1,i\neq c}^{N} p_{Z|X_2}(z_i|c)$$



Summing over y_c to marginalize, we get

$$p_{Z_1,\ldots,Z_N|X_1,X_2}(z_1,\ldots,z_N|x_1,x_2=c) = \underbrace{\Big(\sum_{y_c} p_{Y|X_1}(y_c|x_1)p_{Z|Y,X_2}(z_c|y_c,c)\Big)}_{\equiv\eta(x_1,c,z_c)} \underbrace{\prod_{i=1,i\neq c}^{N} p_{Z|X_2}(z_i|c)}_{\equiv\lambda(c,z_1,\ldots,z_{c-1},z_{c+1},\ldots,z_N)}$$

Evaluating $\eta(\cdot)$ requires $O(K)$ summations and evaluating $\lambda(\cdot)$ requires $O(N)$ multiplications.

(c)

$$p_{X_1,Z_1\ldots,Z_N}(x_1,z_1,\ldots,z_N) = p_{X_1}(x_1)\sum_{c=1}^{N} p_{X_2}(x_2)\eta(x_1,c,z_c)\lambda(c,z_1,\ldots,z_{c-1},z_{c+1},\ldots,z_N)$$

To evaluate $p_{X_1|Z_1\ldots,Z_N}(x_1|z_1,\ldots,z_N)$, we compute $p_{X_1,Z_1\ldots,Z_N}(x_1,z_1,\ldots,z_N)$ for a given values of z_i's and normalize it. Hence the complexity is the same as the complexity of evaluating $p_{X_1,Z_1\ldots,Z_N}(x_1,z_1,\ldots,z_N)$ for a specific values of z_i's and all K values of x_1. For a given values of $x_1,z_1,\ldots,z_N$, it requires $O(N(N+K))$ operations to evaluate the function. $O(N)$ for evaluating $\lambda(\cdot)$ and $O(K)$ for evaluating $\eta(\cdots)$. The extra $O(N)$ comes from the summation over N possible values of c.

Problem 2.4

The graph G is a perfect undirected map for some strictly positive distribution $\mu(x)$ over a set of random variables $x=(x_1,\ldots,x_n)$, each of which takes values in a discrete set $\mathcal{X}$. Choose some variable x_i and let x_A denote the rest of the variables in the model, i.e., $\{x_i,x_A\}=\{x_1,\ldots,x_N\}$. Construct the graph G' from G by removing the node x_i and all its edges. Let some value $c\in\mathcal{X}$ be given. Show that G' is not necessarily a perfect map for the conditional distribution $\mathbb{P}_{x_A|x_i}(\cdot|c)$ by giving a counterexample.

Solution 2.4

Consider a distribution $\mu(x_1,x_2,x_3)=\frac{1}{Z}e^{-x_1x_2x_3}$ for binary random variables $x_1,x_2,x_3\in\{0,1\}$. There is no independence and a perfect map of this distribution is a complete undirected graph with thee nodes. However, if we condition on $x_1=0$ ($c=0$ in this example), then the conditional distribution is $\mu(x_2,x_3|x_1=0)=\frac{1}{Z}$, which means that x_2 and x_3 are independent. For this conditional distribution, a perfect map will be two isolated nodes with no edge, but the induced graph from the original complete graph (by removing

node x_1 and corresponding edges) is two nodes connected by an edge. Hence, G' is not necessarily a perfect map for the conditional distribution.

Problem 2.5

Consider the (parallel) sum-product algorithm on an undirected tree $T = (V, E)$ with compatibility functions ψ_{ij} such that $\mu(x) = \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j)$. Consider any initialization of messages, which is denoted by $\nu_{i\to j}^{(0)}(x_i)$ for all directions $i \to j$ and all states x_i. Messages at step $t \geq 1$ are denoted by $\nu_{i\to j}^{(t)}(x_i)$. In this problem, we will prove by induction that the sum-product algorithm, with the parallel schedule, converges in at most diamater of the graph iterations. (Diameter of the graph is the length of the longest path.)

(*a*) For $D = 1$, the result is immediate. Consider a graph of diameter D. At each time step the message that each of the leaf nodes sends out to its neighbors is constant because it does not depend on messages from any other nodes. Construct a new undirected graphical model $T' = (V', E')$ by stripping each of the leaf nodes from the original graph T. Let $\psi'_{ij}(x_i, x_j)$ be the compatibility functions for the new graphical model, and $\nu_{i\to j}^{\prime(t)}(x_i)$ be the messages of (parallel) sum-product algorithm on the new graphical model. Let L be the set of leaves in T and L' be the set of nodes that is adjacent to a node in L. For the new graphical model, we add, for all $i \in L'$,

$$\psi'_i(x_i) = \psi_i(x_i) \prod_{k\in\partial i\cap L} \sum_{x_k} \nu_{k\to i}^{(0)}(x_k)\psi_{ki}(x_k, x_i)$$

where $\psi_i(x_i) = 1$ if $\psi_i(x_i)$ is not defined for the original graph G and for all other edges we keep the original compatibility functions

$$\psi'_{ij}(x_i, x_j) = \psi_{ij}(x_i, x_j) \; .$$

Also we initialize the messages as

$$\nu_{i\to j}^{\prime(0)}(x_i) = \nu_{i\to j}^{(1)}(x_i) \; .$$

Show that $\nu_{i\to j}^{\prime(t)}(x_i) = \nu_{i\to j}^{(t+1)}(x_i)$ for all $(i, j) \in E'$ and all $t \geq 0$.

(*b*) Argue that T' has diameter strictly less than $D - 1$.

(*c*) By the induction assumption that the parallel sum-product algorithm converges to a fixed point after at most d time steps when the diameter is $d \leq D - 1$, the sum-product algorithm on T' converges after at most $D - 2$ time steps. Show that if we add back the leaf nodes into T' and run (parallel) sum-product algorithm for one more time step, then all messages will have converged to a fixed point.

Solution 2.5

(*a*) At each time step the message that each of the leaf nodes sends out to its neighbor is constant because it does not depend on messages from any other nodes. Hence, $\nu_{i\to j}^{(t)}(x_i) = \nu_{i\to j}^{(0)}(x_i)$ if $i \in L$. Then it follows from our construction of the new graphical model on T' that $\nu_{i\to j}^{\prime(t)}(x_i) = \nu_{i\to j}^{(t+1)}(x_i)$ for all $(i, j) \in E'$ and all $t \geq 0$.

(*b*) The longest path in a tree is between two leaf nodes, so any path of length D will be length $D - 2$ in T'. We also claim that for a path of length $D - 1$, at least one of the start and end nodes must be a leaf. Otherwise, we could extend (if we cannot extend the path, we must form a cycle which is impossible) the path in both directions to get a path of length $D + 1$, a contradiction. Hence, T' has diameter strictly less than $D - 1$.

(*c*) By the induction hypothesis, we know that the messages in T' have converged after $D-2$ steps. Hence $\nu'^{(D-2)}_{i\to j} = \nu'^{*}_{i\to j}$. But by construction from part (*a*), we know that the messages in T' are the same as the corresponding ones in T except for a 1 timestep difference, hence $\nu^{(D-1)}_{i\to j} = \nu'^{(D-2)}_{i\to j} = \nu'^{*}_{i\to j}$ where neither i nor j is a leaf in T (i.e. they have converged). Therefore, after one more iteration, all the messages from i to j, where j is a leaf in T, can be computed. Each of these messages are again also fixed points. Thus, after D iterations each message will have converged to the fixed point.

Problem 2.6

For $\ell \in \mathbb{N}$, let $G_\ell = (V_\ell, E_\ell)$ be an $\ell \times \ell$ two-dimensional grid[1]. We consider an Ising model on G_ℓ with parameters $\theta = \{\theta_{ij},\, \theta_i :\ (i,j) \in E_\ell,\, i \in V_\ell\}$. This is the probability distribution over $x \in \{+1,-1\}^{V_\ell}$

$$\mu(x) = \frac{1}{Z_G}\, \exp\Big\{ \sum_{(i,j)\in E_\ell} \theta_{ij}\, x_i x_j + \sum_{i\in V_\ell} \theta_i x_i \Big\} \tag{1}$$

(*a*) Write the belief propagation (BP) update equations for this model. Also write the update equation for the log-likelihood ratio

$$L^{(t)}_{i\to j} \;\;=\;\; \frac{1}{2}\log\Big(\frac{\nu^{(t)}_{i\to j}(+1)}{\nu^{(t)}_{i\to j}(-1)}\Big)$$

(*b*) Write a program that implements these update. You are requested to return a printout of the code (Matlab, C, C++, Java, ..., are accepted). Feel free to download and start from the skeleton in `bp.m` from the course website. In this case, you are only required to print out the parts of the code that you added.

(*c*) Consider the case $\ell = 10$ (and hence $n = 100$ nodes). For each $\beta \in \{0.2, 0.4, \ldots, 2.8, 3.0\}$, generate an instance by drawing θ_i, θ_{ij} uniformly random in $[0, \beta]$. Run the BP iteration and monitor convergence by computing the quantity

$$\Delta(t) \equiv \frac{1}{|\vec{E}_\ell|} \sum_{(i,j)\in\vec{E}_\ell} \big|\nu^{(t+1)}_{i\to j}(+1) - \nu^{(t)}_{i\to j}(+1)\big|\,. \tag{2}$$

Here $\vec{E}_\ell$ denotes the set of directed edges in G_ℓ, in particular $|\vec{E}_\ell| = 2\,|E_\ell|$.

Plot $\Delta(t=15)$ and $\Delta(t=25)$ versus β, for the random instances generated with $\beta \in \{0.2, 0.4, \ldots, 2.8, 3.0\}$. Comment on the results.

(*d*) Repeat the calculation at the precious point, with now θ_i, θ_{ij} uniformly random in $[-\beta, +\beta]$, with $\beta \in \{0.2, 0.4, \ldots, 2.8, 3.0\}$. Comment on the results.

Solution 2.6

(*a*) The belief propagation update equations for the grid are, up to normalization, as below:

$$\nu^{(t+1)}_{i\to j}(x_i) \;\;\propto\;\; e^{\theta_i x_i} \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k} e^{\theta_{ik} x_i x_k} \nu^{(t)}_{k\to i}(x_k) \Big\}$$

[1]Namely $V_\ell = [\ell[\times[\ell]$ and, for any two vertices $i, j \in V_\ell$, $i = (i_1, i_2)$, $j = (j_1, j_2)$, $i_1, i_2, j_1, j_2 \in [\ell]$, $(i,j) \in E_\ell$ if and only if $i_1 = j_1$ and $|i_2 - j_2| = 1$, or $i_2 = j_2$ and $|i_1 - j_1| = 1$.

Since the variables take two values only, it is somewhat easier to use the log-likelihood ratio instead of the beliefs as messages:

$$L_{i\to j}^{(t)} \quad = \quad \frac{1}{2}\log\Big(\frac{\nu_{i\to j}^{(t)}(+1)}{\nu_{i\to j}^{(t)}(-1)}\Big)$$

The update equations then become:

$$\begin{aligned} L_{i\to j}^{(t+1)} &= \frac{1}{2}\log\Big(\frac{e^{\theta_i}\prod_{k\in\partial i\setminus j}\sum_{x_k} e^{\theta_{ik}x_k}\nu_{k\to i}^{(t)}(x_k)}{e^{-\theta_i}\prod_{k\in\partial i\setminus j}\sum_{x_k} e^{-\theta_{ik}x_k}\nu_{k\to i}^{(t)}(x_k)}\Big) \\ &= \theta_i + \sum_{k\in\partial i\setminus j}\frac{1}{2}\log\Big(\frac{\sum_{x_k} e^{\theta_{ik}x_k}\nu_{k\to i}^{(t)}(x_k)}{\sum_{x_k} e^{-\theta_{ik}x_k}\nu_{k\to i}^{(t)}(x_k)}\Big) \\ &= \theta_i + \sum_{k\in\partial i\setminus j}\frac{1}{2}\log\Big(\frac{e^{(\theta_{ik}+L_{k\to i}^{(t)})}+e^{-(\theta_{ik}+L_{k\to i}^{(t)})}}{e^{(\theta_{ik}-L_{k\to i}^{(t)})}+e^{-(\theta_{ik}-L_{k\to i}^{(t)})}}\Big) \end{aligned}$$

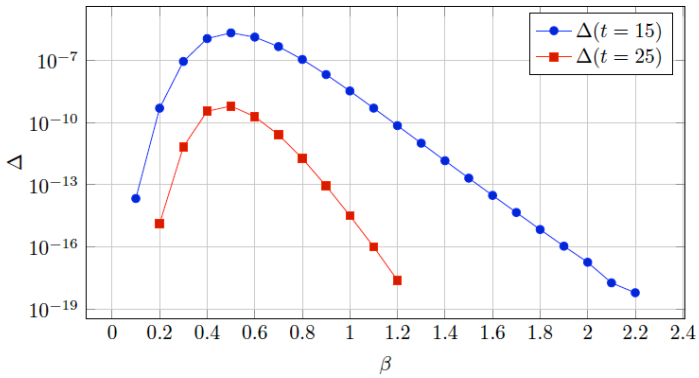
Since $\frac{1}{2}\log(z) = \operatorname{arctanh}(\frac{z-1}{z+1})$, the above simplifies to:

$$L_{i\to j}^{(t)} \quad = \quad \theta_i + \sum_{k\in\partial i\setminus j}\operatorname{arctanh}\Big\{\tanh(\theta_{ik})\tanh(L_{k\to i}^{(t)})\Big\}$$

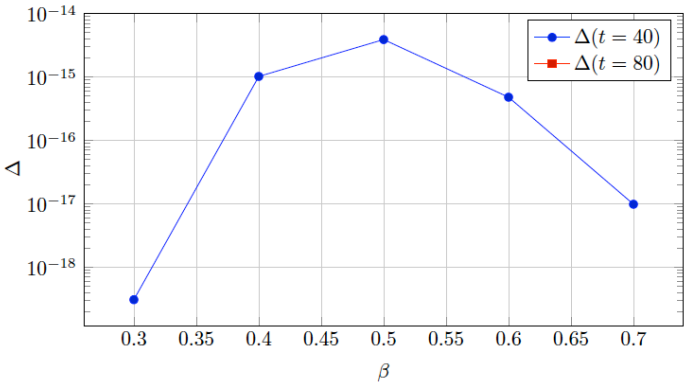
(b) A solution is given as a separate .m file.

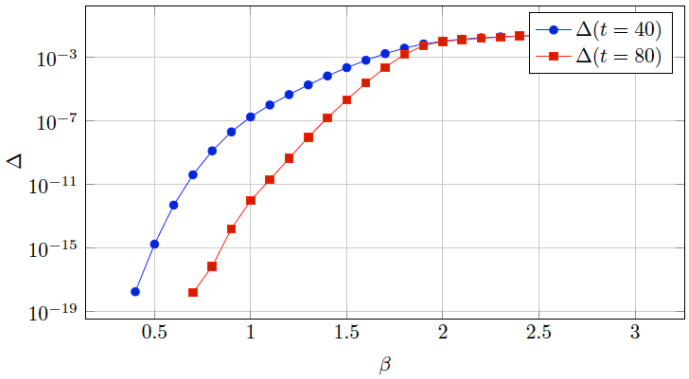
(c) The algorithm converges to a fixed point with high accuracy for the iteration numbers mentioned. Lower iteration numbers yield a slightly better picture.

At low and high values of β, the algorithm converges quickly. For low values this is expected since the interactions along edges are weak and the marginals are approximately the normalized node potentials (corresponding to the graph with no edges). For intermediate values of beta the convergence is slower. For reference, the plot for larger iterations is as below:



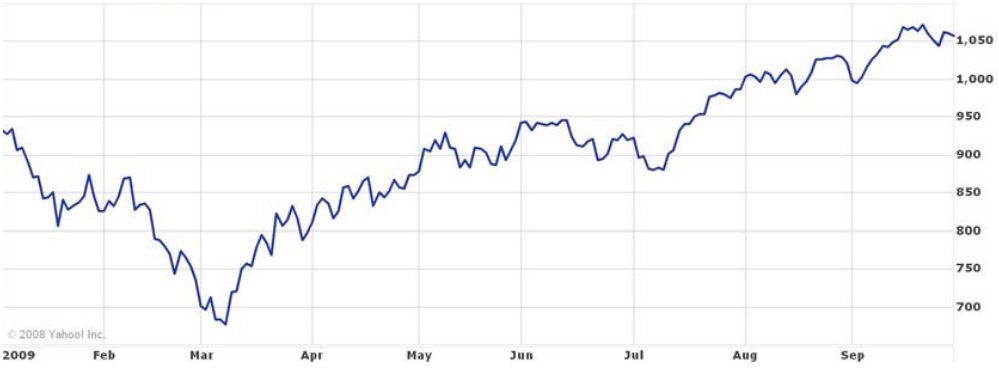
(d) The algorithm does not converge, except at very low values of β. For these values, the measure is approximately independent over the vertices of the graph.





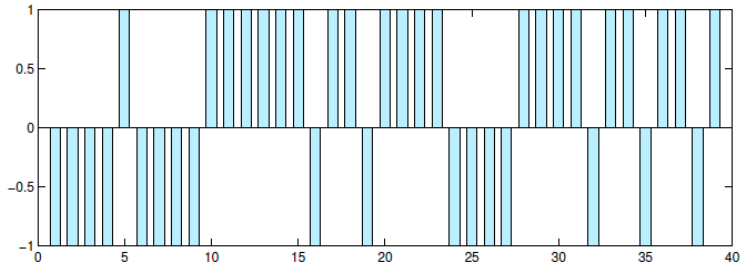
Problem 2.7

In this problem, you will implement the sum-product algorithm on a line graph and analyze the behavior of S&P 500 index over a period of time. The following figure shows the price of S&P 500 index from January 2, 2009 to September 30, 2009 (`http://finance.yahoo.com`).



For each week, we measure the price movement relative to the previous week and denote it using a binary variable (+1 indicates up and 1 indicates down). The price movements from week 1 (the week of January 5) to week 39 (the week of September 28) are plotted below:

Consider a hidden Markov model in which x_t denotes the economic state (good or bad) of week t and



y_t denotes the price movement (up or down) of the S&P 500 index. We assume that $x_{t+1} = x_t$ with probability 0.8, and $\mathbb{P}_{Y_t|X_t}(y_t = +1|x_t = \text{'good'}) = \mathbb{P}_{Y_t|X_t}(y_t = -1|x_t = \text{'bad'}) = q$. In addition, assume that $\mathbb{P}_{X_1}(x_1 = \text{'bad'}) = 0.8$. Download the file `sp500.mat` from course website, and load it into MATLAB. The variable `price_move` contains the binary data above. Implement the (sequential) sum-product algorithm and submit a hardcopy of the code (you dont need to include the code for loading data, generating figures, etc.).

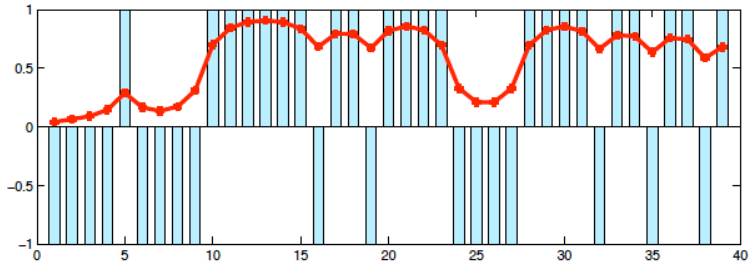
(a) Assume that $q = 0.7$. Plot $\mathbb{P}_{X_t|Y}(x_t = \text{'good'}|y)$ for $t = 1, 2, \ldots, 39$. What is the probability that the economy is in a good state in the week of September 28, 2009 (week 39)?

(b) Repeat (a) for $q = 0.9$. Compare the results of (a) and (b).

Solution 2.7

The code will be provided in a separate .m file.

(a) $\mathbb{P}_{X_{39}|Y}(x_{39} = \text{'good'}|y) = 0.6830$.

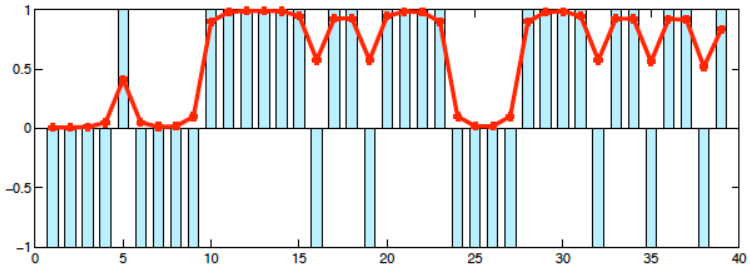


(b) $\mathbb{P}_{X_{39}|Y}(x_{39} = \text{'good'}|y) = 0.8379$.

When q is smaller, the S&P 500 price movement is a less reliable indicator of the economic state, so $\mathbb{P}_{X_t|Y}$ relies more on transition probabilities than emission probabilities. Since $\mathbb{P}_{X_{t+1}|Y}$ is higher when $X_{t+1} = X_t$, $\mathbb{P}_{X_t|Y}$ becomes smoother.

Problem 2.8

Consider a hidden Markov model (HMM) with binary states $x_i \in \{0, 1\}$ for $i \in \{1, \ldots, n\}$ and observations y_i's. For simplicity, let us assume that the model is homogeneous, i.e., $\psi_{i,i+1}(x_i, x_{i+1}) = \psi(x_i, x_{i+1})$ and $\phi_i(x_i, y_i) = \phi(x_i, y_i)$. Given the observations y_i's we are interested in state estimates $\hat{x}_i(y_1, \cdots, y_n)$ that maximizes the probability that at least one of those state estimates $\hat{x}_i$ is correct.



(*a*) The desired state estimates can be expressed in the form

$$(\hat{x}_1, \ldots, \hat{x}_n) \in \arg\min \mathbb{P}(X_1 = f(\hat{x}_1) \wedge \cdots \wedge X_n = f(\hat{x}_n) | y_1, \ldots, y_n) \, .$$

Determine the function $f(\cdot)$.

(*b*) Show that if only the marginal distributions $\mu(x_i|y_1 \ldots, y_n)$, $i \in \{1, \ldots, n\}$ for the model are available, the desired state estimates cannot be determined. In particular, construct two HMMs whose marginals coincide, but whose state estimates differ.
[Hint: it sufficies to consider a model with $n = 2$, and in which the observations are independent of the states thus can be ignored. Accordingly, express your answer in the form of two compatibility functions $\psi(x_1, x_2)$ and $\psi'(x_1, x_2)$.]

(*c*) Construct an example of an HMM in which our desired estimates are not the same as the MAP estimates obtained from running the max-product algorithm on our model. The same hint in part (b) applies, so again give your answer in the form of a compatibility function $\psi(x_1, x_2)$.

(*d*) Let's assume that you are given two pieces of code (e.g., matlab scripts).

The first routine implements the sum-product algorithm, taking as input the potential functions that describe a homogeneous HMM, and an associated list of n observations. It produces as output the list of marginal distributions for each associated n states conditioned on the full set of n observations, for the specified HMM.

The second routine implements the max-product algorithm, taking the same inputs as sum-product algorithm, but producing as output the max-marginals for each associated n states conditioned on the full set of n observations, for the specified HMM.

Describe how to use one or both of these routines to compute the desired estimates $\hat{x}_i(y_1, \ldots, y_n)$ for $i \in \{1, \ldots, n\}$ for our model of interest, assuming that the potentials are strictly positive. You are free to use these routines with any input values you like (whether or not related to the model of interest), and you can further process the outputs of these routines to compute the desired state estimates. However, in such further processing, you are not allowed to (re)use the model's potential functions or observations.

Solution 2.8

(*a*) $f(x) = 1 - x$.

Since any fixed estimate sequence either has at least one of the state estimates correct or has all of them wrong (i.e. those events partition the sample space), maximizing the probability of at least one

state estimate being correct is equivalent to minimizing the probability that all state estimates are wrong. Because of the binary state values, the probability that all state estimates are wrong is exactly the probability that the true state sequence was the state-wise opposite of the estimate sequence, i.e. where $\mathbb{P}(X_1 = f(\hat{x}_1) \wedge \cdots \wedge X_n = f(\hat{x}_n)|y_1, \ldots, y_n)$ $f(\cdot)$ is the bit-flipping function.

(*b*) Consider two compatibility functions $\psi(x_1, x_2) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \psi'(x_1, x_2) = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$. For both cases, the marginals are uniform. However, the variables under ψ are always in a $(0, 0)$ or $(1, 1)$ configuration but could be in either one with equal probability, which means one should choose estimates of $(0, 1)$ or $(1, 0)$ to cover both possibilities and get at least one estimate right with probability 1. For ψ', the situation is reversed: the most likely states are $(0, 1)$ or $(1, 0)$, and so to get at least one right with probability 1 we would want to choose $(1, 1)$ or $(0, 0)$. Therefore marginal information will not suffice, since it would not allow us to discriminate between ψ and ψ'.

(*c*) Either of our examples from part (*b*) would work here as well. For $\psi(x_1, x_2)$, the MAP estimates are clearly $(0, 0)$ and $(1, 1)$, but to choose one of those as our estimates for this problem would not be optimal because an estimate of $(0, 1)$ or $(1, 0)$ means the all-wrong probability is 0, while the all-wrong probability for either $(1, 1)$ or $(0, 0)$ is 1. Thus MAP estimates are not what we are after.

(*d*) Define new potentials $\psi'(x_i, x_{i+1}) = \frac{1}{\psi(x_i, x_{i+1})}$ and $\phi'(x_i, y_i) = \frac{1}{\phi(x_i, y_i)}$ and run max-product on a new HMM defined with these new potentials. We take the output MAP estimate and flip every bit, and return the flipped sequence as our estimate.

As hinted in part (*a*), we can express our estimates easily in terms of the least likely sequence in the original HMM, and the key insight is that the max-product algorithm can give us the least likely state sequence if we give it inverted potentials, i.e. we invert the edge weights on the trellis on which the max-product algorithm operates.