Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

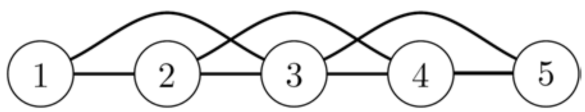
6.438 Algorithms For Inference
Fall 2014

Problem Set 6

Issued: Tuesday, October 28, 2014 **Due:** Thursday, November 13, 2014

Suggested Reading: Lecture notes 12–16

Problem 6.1
Consider a *second-order* Gauss-Markov process represented by the undirected graph below.



Associated with each node $i \in \{1, \ldots, 5\}$ in the graph is a random variable $\mathbf{x}_i$, and the joint distribution for these variables is of the form

$$p_{\mathbf{x}_1,\ldots,\mathbf{x}_5}(\mathbf{x}_1, \ldots, \mathbf{x}_5) \propto \prod_{i\in\mathcal{V}} \psi_i(\mathbf{x}_i) \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges, and where

$$\psi_i(\mathbf{x}_i) = \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ii}\mathbf{x}_i\right\}, \qquad \psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ij}\mathbf{x}_j\right\}, \tag{2}$$

for some given $\mathbf{J}_{ii}$'s and $\mathbf{J}_{ij}$'s.

(a) Draw a junction tree for the graphical model above and specify the separator sets between adjacent cliques in your junction tree.

(b) Specify a clique potential for each clique in your junction tree in terms of the $\mathbf{J}_{ii}$'s and $\mathbf{J}_{ij}$'s of Eq. (2) subject to the following constraints: the potential $\psi_V(\mathbf{x}_V)$ for the clique V that includes node 1 should include as many terms as possible, and the potential $\psi_Z(\mathbf{x}_Z)$ for the clique Z that includes node 5 must include as few terms as possible. Remember that the product of all clique potentials must equal the joint distribution Eq. (1).

(c) Pick a neighboring clique W of V (the clique that includes node 1) and denote it as W, and let S denote the associated separator set. The marginal probabilities of cliques and separator sets are to be computed using Hugin updates. If the algorithm is started at clique V, the update equations for the pass from V to W, i.e.,

$$\phi_S^*(\mathbf{x}_S) = \int \psi_V(\mathbf{x}_V)\, d\mathbf{x}_{V\setminus S} \qquad \text{and} \qquad \psi_W^*(\mathbf{x}_W) = \frac{\phi_S^*(\mathbf{x}_S)}{\phi_S(\mathbf{x}_S)}\, \psi_W(\mathbf{x}_W)$$

with $\phi_S(\mathbf{x}_S) \equiv 1$, can be expressed in the form

$$\phi_S^*(\mathbf{x}_S) = \exp\left\{-\frac{1}{2}\mathbf{x}_S^{\mathrm{T}}\mathbf{J}_S^*\mathbf{x}_S\right\} \qquad \text{and} \qquad \psi_W^*(\mathbf{x}_W) = \exp\left\{-\frac{1}{2}\mathbf{x}_W^{\mathrm{T}}\mathbf{J}_W^*\mathbf{x}_W\right\}.$$

Express $\mathbf{J}_S^*$ and $\mathbf{J}_W^*$ in terms of the $\mathbf{J}_{ii}$'s and $\mathbf{J}_{ij}$'s.

Hint: You may find useful the identities

$$\mathcal{N}^{-1}(\mathbf{x}; \mathbf{h}_{\mathrm{a}}, \mathbf{J}_{\mathrm{a}})\mathcal{N}^{-1}(\mathbf{x}; \mathbf{h}_{\mathrm{b}}, \mathbf{J}_{\mathrm{b}}) \propto \mathcal{N}^{-1}(\mathbf{x}; \mathbf{h}_{\mathrm{a}} + \mathbf{h}_{\mathrm{b}}, \mathbf{J}_{\mathrm{a}} + \mathbf{J}_{\mathrm{b}})$$

and

$$\int \mathcal{N}^{-1}\left(\begin{bmatrix}\mathbf{x}_{\mathrm{a}} \\ \mathbf{x}_{\mathrm{b}}\end{bmatrix}; \begin{bmatrix}\mathbf{h}_{\mathrm{a}} \\ \mathbf{h}_{\mathrm{b}}\end{bmatrix}, \begin{bmatrix}\mathbf{J}_{\mathrm{aa}} & \mathbf{J}_{\mathrm{ab}} \\ \mathbf{J}_{\mathrm{ba}} & \mathbf{J}_{\mathrm{bb}}\end{bmatrix}\right) d\mathbf{x}_{\mathrm{b}} \\ \propto \mathcal{N}^{-1}(\mathbf{x}_{\mathrm{a}}; \mathbf{h}_{\mathrm{a}} - \mathbf{J}_{\mathrm{ab}}\mathbf{J}_{\mathrm{bb}}^{-1}\mathbf{h}_{\mathrm{b}}, \mathbf{J}_{\mathrm{aa}} - \mathbf{J}_{\mathrm{ab}}\mathbf{J}_{\mathrm{bb}}^{-1}\mathbf{J}_{\mathrm{ba}}),$$

when

$$\mathcal{N}^{-1}(\mathbf{x}; \mathbf{h}, \mathbf{J}) \propto \exp\left\{\mathbf{h}^{\mathrm{T}}\mathbf{x} - \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{J}\mathbf{x}\right\}.$$

(d) Suppose that we have completed all Hugin updates and have the resulting adjusted potentials at all (maximal) cliques and separator sets in the junction tree. Now we observe $\mathsf{x}_3 = \mathbf{x}_3$. At which cliques C can we compute $p_{\mathsf{x}_i|\mathsf{x}_3}(\mathbf{x}_i|\mathbf{x}_3)$, where $i \in C$, from the results of the preceding Hugin algorithm, without any further Hugin updates or other subsequent communication between the (maximal) cliques?

Does your answer change if, instead of observering x_3, we observe $\mathsf{x}_5 = \mathbf{x}_5$ and seek marginals conditioned on this observation? Explain.

Problem 6.2

For each of the graphs in Figure 1, construct a junction tree by first triangulating and then applying a greedy algorithm. You should write each step of the greedy algorithm.

Problem 6.3

As discussed in lecture, loopy belief propagation on general (loopy) graphs is neither guaranteed to converge nor, when it does converge, guaranteed to yield correct marginals. In this problem, we'll consider an example of such phenomena on a highly symmetric graphical model.

An *Ising model* on a vector of binary variables x, with each $x_i \in \{+1, -1\}$, with parameter vector θ is described by

$$p(\mathsf{x}; \theta) \propto \exp\left\{\sum_{i \in V}\theta_i x_i + \sum_{(i,j)\in E}\theta_{ij}x_i x_j\right\}$$

for a graph (V, E).

For this problem, we'll consider an Ising model with $\forall i \in V\ \theta_i = 0$ and $\forall (i,j) \in E\ \theta_{ij} = \gamma > 0$ on a toroidal grid graph, as shown in Figure 2.

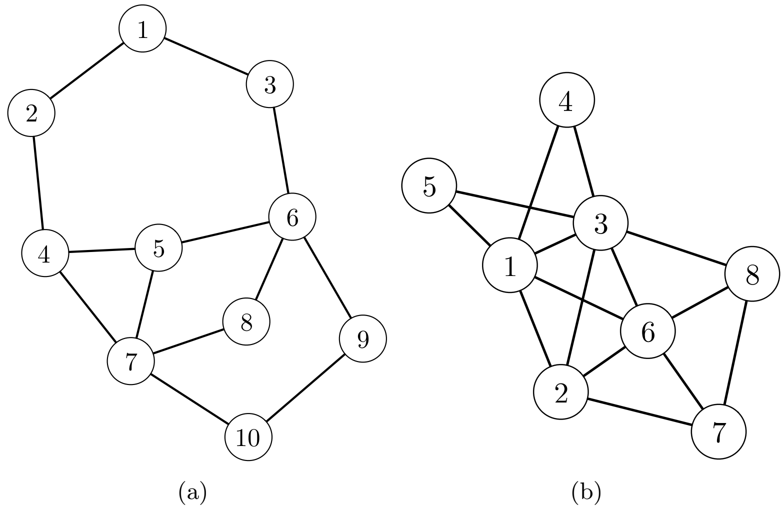


Figure 1

(a) Show that the single node marginal distributions are uniform for all values of γ (i.e. $\Pr[x_i = +1] = \Pr[x_i = -1] = \frac{1}{2}$).

(b) Implement the loopy belief propagation (sum-product) algorithm for the model.

Note that because of the symmetry introduced to make the problem easy to analyze, our usual initialization procedure (setting all the messages to be identically 1) will actually start the algorithm at a fixed point for any γ. (You can check that the all-ones messages are always fixed points of the update formula for any γ.) However, such a fixed point may not be interesting because it may be *unstable*, meaning that if our messages weren't initialized to be exactly uniform but instead initialized within a small neighborhood of the uniform messages, then the message updates would trend away from the uniform fixed point. Unstable fixed points are uninteresting because loopy BP would not find these fixed points unless initialized exactly at them, which for general models (without so many symmetries) would not happen for any initialization procedure. **Instead of initializing the messages to be identically** 1**, set their values to be near 1 but perturbed by uniformly random noise on** $[-0.01\ 0.01]$**.** Random initialization is another standard initialization procedure, and allows us to break the symmetry in this problem and thus only examine stable fixed points of the loopy BP algorithm. You can check that random message initialization would also provide correct results for BP on trees using synchronous parallel updates, but note that the initialization of loopy BP messages will affect which fixed point is reached (if any).

Run your implementation for various values of $\gamma \in (0, 1)$. Find the smallest γ (up to **two** decimal places) for which the algorithm converges to pseudo-marginals that are *not* uniform. Provide a plot of the error in the marginals as a function of γ.

(c) Show that there exists a threshold $\gamma^* > 0$ such that for any $\gamma \in (0, \gamma^*)$, if loopy BP converges on our model, it will yield the correct (uniform) marginals. *Hint: since*

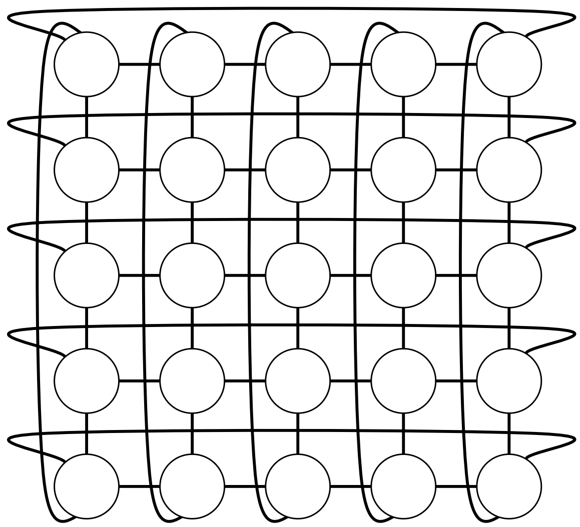


Figure 2

all nodes are symmetric, it suffices to consider a fixed-point of the message update formula at any node.

Problem 6.4

In this exercise, you will construct an undirected graphical model for the problem of segmenting foreground and background in an image, and use loopy belief propagation to solve it.

Load the image `flower.bmp` into MATLAB® using `imread`. (The command `imshow` may also come in handy.) Partial labeling of the foreground and background pixels are given in the mask images `foreground.bmp` and `background.bmp`, respectively. In each mask, the white pixels indicate positions of representative samples of foreground or background pixels in the image.

Let $\mathbf{y} = \{y_i\}$ be an observed color image, so each y_i is a 3-vector (of RGB values between 0 and 1) representing the pixel indexed by i. Let $\mathbf{x} = \{x_i\}$, where $x_i \in \{0, 1\}$ is a foreground(1)/background(0) labeling of the image at pixel i. Let us say the probabilistic model for $\mathbf{x}$ and $\mathbf{y}$ given by their joint distribution can be factored in the form

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_i \psi(x_i, y_i) \prod_{(j,k)\in E} \psi(x_j, x_k)$$

where E is the set of all pairs of adjacent pixels in the same row or column.

Suppose that we choose

$$\psi(x_j, x_k) = \begin{cases} 0.9, & \text{if } x_j = x_k \\ 0.1, & \text{if } x_j \neq x_k \end{cases}$$

This encourages neighboring pixels to have the same label — a reasonable assumption.

Suppose further that we use a simple model for the conditional distribution $p(y_i|x_i)$:

$$p(y_i|x_i = \alpha) \propto \frac{1}{(2\pi)^{3/2}(\det \mathbf{\Lambda}_\alpha)^{1/2}} \exp\left[-\frac{1}{2}(y_i - \mu_\alpha)^T \mathbf{\Lambda}_\alpha^{-1}(y_i - \mu_\alpha)\right] + \epsilon$$

for $y_i \in [0, 1]^3$. That is, the distribution of color pixel values over the same type of image region is a modified Gaussian distribution, where ϵ accounts for outliers. Set $\epsilon = 0.01$ in this problem.

(a) Sketch and label the undirected graph with respect to which the distribution $p(\mathbf{x}, \mathbf{y})$ is Markov. What are the potential functions $\psi(x_i, y_i)$?

(b) Compute $\mu_\alpha, \mathbf{\Lambda}_\alpha$ from the labeled masks by finding the sample mean and covariance of the RGB values of those pixels for which the label $x_i = \alpha$ is known.

(c) We want to run the sum-product algorithm on the graph iteratively to find (approximately) the marginal distribution $p(x_i|\mathbf{y})$ at every i.

Write the local message update rule for passing the message $m_{j\to k}(x_k)$ from x_j to x_k, in terms of the messages from the other neighbors of x_j, the potential functions, and some arbitrary scaling constant.

Then write the final belief update rule on x_i, that is, the marginal computation in terms of the messages from all neighbors of x_i, the potential functions, and some arbitrary scaling constant.

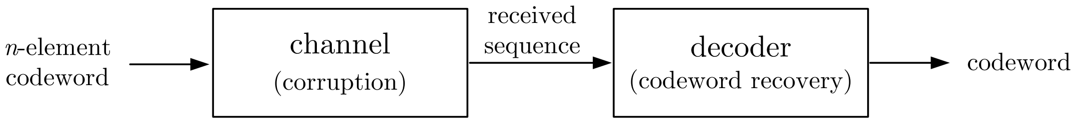
(d) Implement the sum-product algorithm for this problem. There are four directional messages: down, up, left, and right, coming into and out of each x_i (except at the boundaries). Use a parallel update schedule, so all messages at all x_i are updated at once. Run for 30 iterations (or you can state and use some other reasonable termination criterion). You should renormalize the messages after each iteration to avoid overflow/underflow problems.

After the marginal distributions at the pixels are estimated, visualize their expectation (that would be the BLS estimates of x_i based on $\mathbf{y}$ if the inference were exact). Where are the beliefs "weak"? Where did the loopy belief propagation converge first and last?

(e) **(Practice)** Convert the graph in this problem to a factor graph and write the belief propagation messages passed between variable nodes and factor nodes.

Problem 6.5

Consider a very simple digital communication system. One of a finite number of equally-likely, length-n binary sequences (referred to as codewords) is sent by a transmitter. The receiver obtains a corrupted version of the sent codeword. Specifically, some subset of the elements in the codeword are erased. Moreover, the erasures are independent, and identically distributed within a codeword, and independent of the codeword. A decoder attempts to reconstruct the sent codeword from the un-erased elements and their locations.



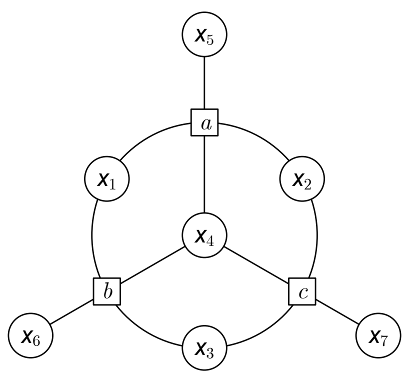
In this problem, $n = 7$ and the codewords $(\mathsf{x}_1, \mathsf{x}_2, ..., \mathsf{x}_7)$ have distribution

$$p_{\mathsf{x}_1,\mathsf{x}_2,...,\mathsf{x}_7}(x_1, x_2, ..., x_7) = \frac{1}{Z}\,\delta(x_1 \oplus x_2 \oplus x_4 \oplus x_5)\,\delta(x_1 \oplus x_3 \oplus x_4 \oplus x_6)\,\delta(x_2 \oplus x_3 \oplus x_4 \oplus x_7), \quad (3)$$

where the x_i are each 0 or 1, $\oplus$ is addition modulo-2 (i.e., XOR), and where $\delta(x) = 1$ when $x = 0$ and is 0 otherwise.

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise.} \end{cases}$$

The (minimal) factor graph expressing the relationship between the 7 codeword symbols is as follows:



(a) Draw the undirected graph with the minimum number of edges that expresses the relationship between the 7 codeword symbols.

(b) Determine the partition function Z in (3).

(c) If the received sequence is $(0, ?, ?, 1, 0, ?, 0)$ where "?" denotes an erased symbol, determine the codeword that was sent. Give an example of a received sequence with 3 erasures from which the sent codeword cannot be determined with certainty.

One method of decoding is to apply loopy belief propagation (LBP), i.e., the sum-product algorithm in which all nodes send messages at all times. At each iteration of the algorithm, factor nodes process all incoming messages to produce outgoing messages to variable nodes, then, in turn, the variable nodes use these messages to produce new messages for the factor nodes.

Recall that the sum-product belief propagation message-passing takes the form:

$$\mu_{s,i}(x_i) \propto \sum_{x_{\mathcal{N}(s)\setminus i}} \left(f_s(x_{\mathcal{N}(s)}) \prod_{j \in \mathcal{N}(s)\setminus i} \nu_{j,s}(x_j) \right)$$
$$\nu_{i,s}(x_i) \propto \prod_{t \in \mathcal{N}(i)\setminus s} \mu_{t,i}(x_i)$$

where $\mu_{s,i}(\cdot)$ and $\nu_{i,s}(\cdot)$ are the factor-to-variable and variable-to-factor messages, respectively, and where $\mathcal{N}(\cdot)$ denotes the neighbors of its node argument.

(d) Draw the factor graph that LBP will operate on. Include any additional variable and/or factor nodes that are required, and indicate any modifications to factor nodes that are required.

(e) Show that the (vector) messages in this application of LBP, when normalized so that their largest entry is 1, take on one of only three possible values.

(f) Apply the resulting LBP algorithm to the same observed sequence $(0, ?, ?, 1, 0, ?, 0)$ given in part (c). Express the operation of the algorithm as follows: start by describing the initial messages; then, with each iteration, specify any factor-to-variable and variable-to-factor messages that have changed from the previous iteration, and specify the new values of these messages.

Remark: While this can be done by hand, coding this up to handle any length-7 sequence of 1's, 0's, and ?'s could be a useful exercise.

Problem 6.6

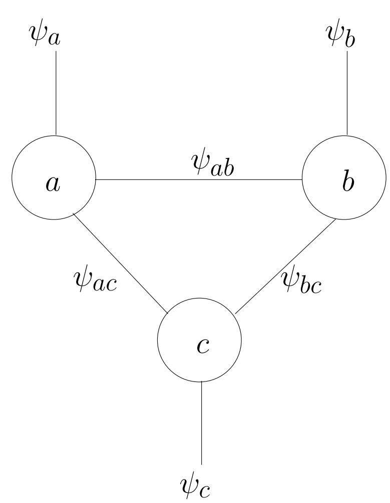


Figure 3

(a) Consider a three-node loopy graphical model depicted in Figure 3, where the compatibility functions for this model are given by the following:

$$\psi_a(0) = \psi_a(1) = \psi_b(0) = \psi_b(1) = \psi_c(0) = \psi_c(1) = 1 \tag{4}$$

$$\psi_{ab}(0,0) = \psi_{ab}(1,1) = 10 \quad , \quad \psi_{ab}(1,0) = \psi_{ab}(0,1) = 1 \tag{5}$$

$$\psi_{ac}(0,0) = \psi_{ac}(1,1) = 10 \quad , \quad \psi_{ac}(1,0) = \psi_{ac}(0,1) = 1 \tag{6}$$

$$\psi_{bc}(0,0) = \psi_{bc}(1,1) = 10 \quad , \quad \psi_{bc}(1,0) = \psi_{bc}(0,1) = 1 \tag{7}$$

(i) Compute the max-marginals at each of the three nodes. You should see that there is a non-unique maximum for each of these.

(ii) What are the configurations of values at the three nodes of this graph that are most probable? For this simple graph, show that you can still do something similar to what we did in Problem 5.1 using edge-max-marginals to help us compute one of these maximizing configurations.

(b) In general when a loopy graph has non-unique optimizing configurations, the node- and edge-max-marginals aren't enough to allow computation of a maximum probability configuration. In particular, show that the triangle graph above has the following property:

There does not exist any (deterministic) algorithm which given as input the triangle graph and the node- and edge-max-marginals, can determine a configuration[1] with maximum probability.

For simplicity, restrict yourself to finite alphabets in this problem, i.e., each node in your graph should only take on values from a finite set, which without of loss generality can be denoted by $0, 1, \ldots, M$.

Note that the algorithm does not get access to the full probability distribution, since otherwise there is a trivial (but potentially inefficient) algorithm to solve the problem - compute the probability for every possible configuration, and then output a configuration with maximum probability. Also, note that when the graph is a tree, we saw that the max-product algorithm can solve this problem. Specifically, given the node- and edge-max-marginals, we saw in lecture how to break ties and compute a maximum probability configuration. Thus, your construction shows yet another reason why trees are easier to deal with than graphs with cycles.

Problem 6.7 (Practice)
Consider the graph, shown in Figure 4, with a single loop, where $y_1, ..., y_4$ are observed. Let

[1] There can be multiple maximizing configurations. In this case, the algorithm's output is considered correct as long as it outputs *some* maximizing configuration.

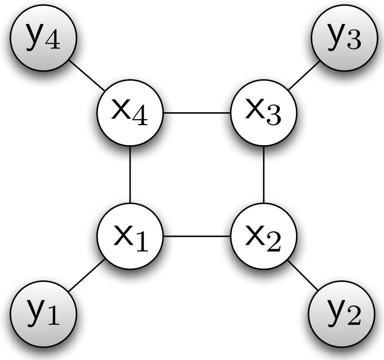


Figure 4

$x_1, ..., x_4$ be discrete random variables defined over a finite alphabet $\{1, 2, ..., M\}$ for some finite M, so we can represent edge potentials $\psi_{x_i x_j}(x_i, x_j)$ by matrices

$$\mathbf{A}_{ij} = \begin{bmatrix} \psi_{x_i x_j}(1,1) & \dots & \psi_{x_i x_j}(1,M) \\ \vdots & \ddots & \vdots \\ \psi_{x_i x_j}(M,1) & \dots & \psi_{x_i x_j}(M,M) \end{bmatrix}$$

i.e., $\mathbf{A}_{ij}[u,v] = \psi_{x_i x_j}(x_i = u, x_j = v)$. Similarly, node potentials $\psi_{x_i}(x_i)$ can be represented by diagonal matrices

$$\mathbf{B}_i = \begin{bmatrix} \psi_{x_i}(1) & & 0 \\ & \ddots & \\ 0 & & \psi_{x_i}(M) \end{bmatrix}$$

Consider running loopy belief propagation on this graph using a parallel schedule, where each node computes the usual sum-product messages to be passed to its neighbors, as if the graph were acyclic. Let a column vector $\mathbf{m}_{ij}^{(k)}$ of length M denote the message that x_i sends to x_j on the k-th iteration. Moreover, we will use similar vector notation to denote the marginal probabilities of interest. For example, $\mathbf{p}_{x_1|\mathbf{y}}$ will denote an M-vector whose j-th entry is $p_{x_1|\mathbf{y}}(j|\mathbf{y})$.

(a) Express the messages $\mathbf{m}_{12}^{(1)}$ and $\mathbf{m}_{41}^{(4)}$ in terms of the matrices $\mathbf{A}_{ij}$, $\mathbf{B}_i$ and the initial messages $\mathbf{m}_{ij}^{(0)}$.

(b) Let $\mathbf{b}_{x_i}^{(k)}$ denote, in M-vector notation, the estimate of the marginal for x_i produced by this instance of loopy belief propagation that x_i can compute after k iterations, in the usual way. Specifically, at node x_1, after $4n$ iterations, we have

$$\mathbf{b}_{x_1}^{(4n)} \propto \mathbf{W}(\mathbf{m}_{41}^{(4n)} \odot \mathbf{m}_{21}^{(4n)}) \tag{8}$$

where $\odot$ denotes component-wise multiplication of vectors.

Find $\mathbf{m}_{41}^{(4n)}$, $\mathbf{m}_{21}^{(4n)}$, and the matrix $\mathbf{W}$ in terms of the matrices $\mathbf{A}_{ij}$, $\mathbf{B}_i$, and the initial messages $\mathbf{m}_{ij}^{(0)}$.

(c) The *exact* marginal $\mathbf{p}_{x_1|\mathbf{y}}$ depends on the $\mathbf{A}_{ij}$ and $\mathbf{B}_i$ only through the matrix product $\mathbf{A}_{14}\mathbf{B}_4\mathbf{A}_{43}\mathbf{B}_3\mathbf{A}_{32}\mathbf{B}_2\mathbf{A}_{21}\mathbf{B}_1$. Express $\mathbf{p}_{x_1|\mathbf{y}}$ in this form.

In the following, assume that all potential functions and initial messages are strictly positive.

(d) It is a well known fact that if a matrix $\mathbf{P}$ and a vector $\mathbf{z}$ both have strictly positive entries, then

$$\lim_{k\to\infty}\left(\frac{\mathbf{P}}{\lambda_1\{\mathbf{P}\}}\right)^k \mathbf{z} \propto \mathbf{v}_1\{\mathbf{P}\}$$

where $\mathbf{v}_1\{\mathbf{P}\}$ denotes the positive eigenvector of $\mathbf{P}$ associated with its unique largest eigenvalue $\lambda_1\{\mathbf{P}\}$.

Using this fact, it follows that the estimated marginal $\mathbf{b}_{x_1}^{(k)}$ at node x_1 converges to a steady state

$$\mathbf{b}_{x_1}^{(\infty)} \propto \mathbf{v}_1\{\mathbf{C}\} \odot \mathbf{v}_1\{\mathbf{C}^{\mathrm{T}}\}$$

Express $\mathbf{C}$ in terms of the matrices $\mathbf{A}_{ij}$ and $\mathbf{B}_i$.

Hint: $\mathbf{A}_{ij} = \mathbf{A}_{ji}^{\mathrm{T}}$.

(e) Suppose $x_1, ..., x_4$ are binary variables, then the maximum marginal assignment based on the steady state estimate of loopy belief propagation is correct. Show this for x_1, that is, if $\mathbf{p}_{x_1|\mathbf{y}}$ is the true marginal and $\mathbf{b}_{x_1}^{(\infty)}$ is the steady state belief, show that

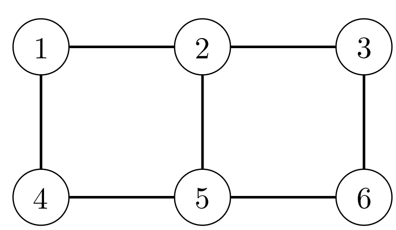
$$\arg\max_j p_{x_1|\mathbf{y}}(j|\mathbf{y}) = \arg\max_j b_{x_1}^{(\infty)}(j) \quad , \quad j \in \{1,2\}$$

where $b_{x_1}^{(\infty)}(j)$ is the j-th entry of $\mathbf{b}_{x_1}^{(\infty)}$.

Hint: A 2×2 matrix $\mathbf{P}$ with positive entries can be expressed in the form $\mathbf{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{\Lambda}$ is diagonal.

Problem 6.8

Consider the undirected graphical model depicted below.



(a) Treating node 2 as the root node, draw the computation tree corresponding to the first 4 iterations of loopy belief propagation.

Associated with each node $i \in \{1, \ldots, 6\}$ in the graphical model is a binary random variable $\mathsf{x}_i \in \{0, 1\}$, and the joint distribution for these variables is of the form

$$p_{\mathsf{x}_1,\mathsf{x}_2,\mathsf{x}_3,\mathsf{x}_4,\mathsf{x}_5,\mathsf{x}_6}(x_1, x_2, x_3, x_4, x_5, x_6) \propto \psi_2(x_2)\, \psi_5(x_5) \prod_{(i,j)\in\mathcal{E}} \delta(x_i, x_j),$$

where $\mathcal{E}$ is the set of edges in the graph, and, for some $0 < \gamma < 1$,

$$\delta(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}, \qquad \psi_2(x_2) = \begin{cases} 1 - \gamma & x_2 = 0 \\ \gamma & x_2 = 1 \end{cases}, \qquad \psi_5(x_5) = \begin{cases} \gamma & x_5 = 0 \\ 1 - \gamma & x_5 = 1 \end{cases}.$$

(b) Determine the sequence of marginal probabilities of x_2 obtained from loopy belief propagation, starting with its initialization, then after each of the first four iterations. Assume parallel message updates, i.e., that a new message is computed along each edge at each iteration of the algorithm.

(c) Consider the messages generated by loopy belief propagation for this graph, with the messages normalized so that the message from i to j satisfies $\sum_{x_j} m_{ij}(x_j) = 1$ for all $(i, j) \in \mathcal{E}$. Suppose

$$m_{23}(x_3) = \begin{cases} \alpha & x_3 = 0 \\ 1 - \alpha & x_3 = 1 \end{cases} \qquad \text{and} \qquad m_{21}(x_1) = \begin{cases} \beta & x_1 = 0 \\ 1 - \beta & x_1 = 1 \end{cases}.$$

Determine all values of α and β such that there are messages in the graph that both (i) satisfy the belief propagation message update equations everywhere and (ii) give the correct marginal probability at each node.

For messages satisfying (i) and (ii), with $m_{23}(x_3)$ and $m_{21}(x_1)$ as above, specify $m_{12}(x_2)$, $m_{52}(x_2)$, and $m_{32}(x_2)$, expressing these messages in terms of α and β.

Problem 6.9 (Practice)

Consider the Gaussian graphical model depicted below. More precisely, if we let $\mathbf{x}$ denote the 4-dimensional vector of variables at the 4 nodes (ordered according to the node numbering given), then $\mathbf{x} \sim \mathcal{N}^{-1}(\mathbf{h}, \mathbf{J})$ where $\mathbf{J}$ has diagonal values all equal[2] to 1 and non-zero off-diagonal entries as indicated in the figure (e.g., $\mathbf{J}_{12} = -\rho$).

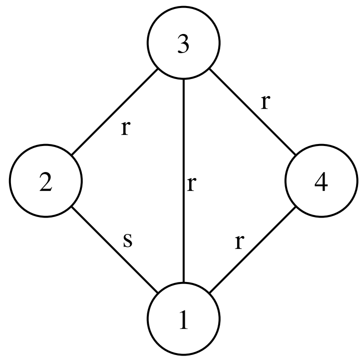
(a) Confirm (e.g., by checking Sylvester's criterion) that $\mathbf{J}$ is a valid information matrix—i.e., it is positive definite—if $\rho = 0.39$ *or* 0.4.

Compute the variances for each of the components (i.e., the diagonal elements of $\mathbf{\Lambda} = \mathbf{J}^{-1}$)—you can use MATLAB to do this if you'd like.

(b) We now want to examine Loopy BP for this model, focusing on the recursions for the information matrix parameters.

Write out these recursions in detail for this model. Implement these recursions and try for $\rho = 0.39$ and $\rho = 0.4$. Describe the behavior that you observe.

[2] We can always achieve this by scaling the components of $\mathbf{x}$.



(c) Construct the computation tree for this model. Note that the effective "**J**"-parameters for this model are copies of the corresponding ones for the original model (so that every time the edge (1,2) appears in the computation tree, the corresponding **J**-component is $-\rho$. Use MATLAB to check the positive-definiteness of these implied models on computation trees for different depths and for our two different values of ρ. What do you observe that would explain the result in part (b)?

6.438 Algorithms for Inference
Fall 2014