HW2

Jamie Morgenstern *CSE515*

Pick 4 problems you want to solve and only solve those 4 problems.

1. *[2 points]* This is a continuation of hw 1 problem 1. 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 $y = \{y_i\}$ 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 $x = \{x_i\}$, where $x_i \in \{0, 1\}$ 2 is a foreground(1)/background(0) labeling of the image at pixel i. Let us say the probabilistic model for x and y given by their joint distribution can be factored in the form

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

where E is the set of all pairs of adjacent pixels in the same row or column as in 2-dimensional grid. 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 $\phi(x_i, y_i) = \mathbb{P}_{Y_i|X_i}(y_i|x_i)$:

$$\mathbb{P}(y_i|x_i = \alpha) \quad \propto \quad \frac{1}{(2\pi)^{3/2}\sqrt{\det\Lambda_\alpha}} \exp\Big\{ -\frac{1}{2}(y_i - \mu_\alpha)^T \Lambda_\alpha^{-1}(y_i - \mu_\alpha)\Big\} + \epsilon$$

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.

Recall your sketch of an undirected graphical model that represents $\mu(x, y)$.

We will use your computed $\mu_\alpha \in \mathbb{R}^3$ and $\Lambda_\alpha \in \mathbb{R}^{3\times3}$ for each $\alpha \in \{0, 1\}$, that you got 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 from `foreground.bmp` and `background.bmp`. The sample mean of samples $\{y_1, \dots, y_N\}$ is $\bar{y} = \frac{1}{N}\sum_{i=1}^N y_i$ and the sample covariance is $C_y = \frac{1}{N-1}\sum_{i=1}^N (y_i - \bar{y})(y_i - \bar{y})^T$.

(a) We want to run the sum-product algorithm on the graph iteratively to find (approximately) the marginal distribution $\mu(x_i|y)$ at every i. For a joint distribution of the form (1) with pairwise compatibility functions and singleton compatibility functions, the local message update rule for passing the message $\nu_{j\to k}(x_j)$ from x_j to x_k, is represented in terms of the messages from the other neighbors of x_j, the potential functions.

$$\nu_{j\to k}(x_j) \propto \phi(x_j, y_j) \prod_{u\in\partial j\setminus k} \sum_{x_u} \psi(x_j, x_u)\nu_{u\to j}(x_u)$$

Then the final belief on x_j is computed as

$$\nu_j(x_j) \propto \phi(x_j, y_j) \prod_{u \in \partial j} \sum_{x_u} \psi(x_j, x_u) \nu_{u \to j}(x_u)$$

Feel free to use `gridbpsol.m` or `gridbpsol.py` from the website for running the BP algorithm. 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). Since we are working with binary random variables, perhaps it is easier to pass messages in log-likelihood.

After the marginal distributions at the pixels are estimated, visualize their expectation. Where are the beliefs "weak"?

(b) Visualize the expectation after 1, 2, 3, and 4 iterations. Qualitatively, discuss where the loopy belief propagation converges first and where it converges last.

(c) Choose a different value of ϵ, run BP with that different value of $\epsilon \neq 0$, and comment on the result.

(d) Run BP with the following pairwise potential and comment on the result.

$$\psi(x_j, x_k) = \begin{cases} 0.6 & \text{if } x_j = x_k \\ 0.4 & \text{if } x_j \neq x_k \end{cases}$$

2. *[2 points]* **(Belief propagation convergence on a tree)**
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 $\nu^{(0)}_{i \to j}(x_i)$ for all directions $i \to j$ and all states x_i. Messages at step $t \geq 1$ are denoted by $\nu^{(t)}_{i \to j}(x_i)$. 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 $\nu'^{(t)}_{i \to j}(x_i)$ 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 $i \in L'$,

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

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

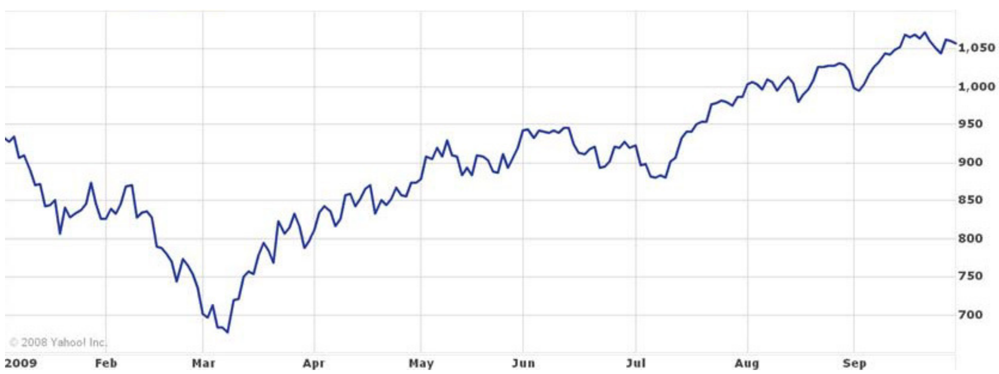
$$\nu'^{(0)}_{i \to j}(x_i) = \nu^{(1)}_{i \to j}(x_i) \ .$$

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

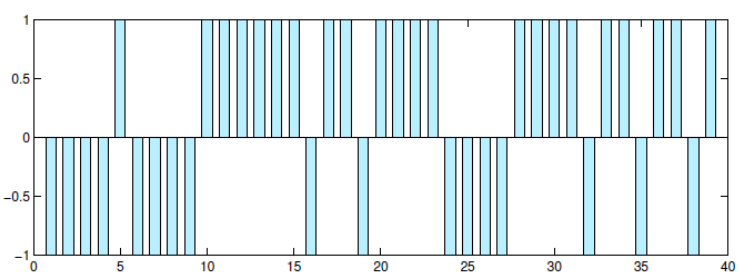
(*b*) Argue that T' has diameter strictly less than $D-1$.

(*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.

3. *[2 points]* **(Hidden Markov models; implementation)**
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` (Matlab file) or `sp500.csv` (csv file) from course website, and load it into MATLAB or whichever programming language you feel comfortable with. The variable `price_move` contains the binary data above. Implement the (sequential) sum-product algorithm and submit the code with the homework solutions.

(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).

4. *[2 points]* **(Application of LDPC codes)**
In this problem we consider using Low-Density Parity Check (LDPC) codes to encode bits to be sent over a noisy channel.

Encoding. LDPC codes are defined by a factor graph model over a bipartite graph $G(V, F, E)$, where V is the set of variable nodes, each representing the bit to be transmitted, and F is a set of factor nodes describing the code and E is a west of edges between a bit-node and a factor node. The total number of variable nodes in the graph define the length of the code (also known as the block length), which we denote by $n \triangleq |V|$.

We consider binary variables $x_i \in \{-1, +1\}$ for $i \in V$, and all codewords that are transmitted satisfy

$$\prod_{i \in \partial a} x_i \quad = \quad +1 \ ,$$

which means that there are even number of -1's in the neighborhood of any factor node.

Channel. We consider a Binary Symmetric Channel, known as BSC(ε), where one bit is transmitted over the channel at each discrete time step, and each transmitted bit is independently flipped with probability ε. Precisely, let $x_i \in \{+1, -1\}$ be a transmitted bit and $y_i \in \{+1, -1\}$ be the received bit (at time i), then

$$\begin{aligned}
\mathbb{P}(y_i = +1 | x_i = +1) &= 1 - \varepsilon \ , \\
\mathbb{P}(y_i = -1 | x_i = +1) &= \varepsilon \ , \\
\mathbb{P}(y_i = -1 | x_i = -1) &= 1 - \varepsilon \ , \\
\mathbb{P}(y_i = +1 | x_i = -1) &= \varepsilon \ .
\end{aligned}$$

The conditional probability distribution over $x_1^n = [x_1, \ldots, x_n]$ given the observed received bits $y_1^n = [y_1, \ldots, y_n]$ is

$$\mu(x_1^n \,|\, y_1^n) \quad = \quad \frac{1}{Z} \prod_{i \in V} \psi_i(x_i, y_i) \prod_{a \in F} \mathbb{I}(\otimes x_{\partial a} = +1) \ ,$$

where $\psi_i(x_i, y_i) = \mathbb{P}(y_i|x_i)$ and $\otimes$ indicates product of binary numbers such that if $x_{\partial a} = \{x_1, x_2, x_3\}$ then $\otimes x_{\partial a} = x_1 \times x_2 \times x_3$ (to be precise we need to take $\psi_i(x_i|y_i) = \mathbb{P}(x_i|y_i)$, but this gives the exactly same conditional distribution as above since any normalization with respect to y_i's are absorbed in the partition function Z). This is naturally a graphical model on a factor graph $G(V, F, E)$ defined by the LDPC code.

(a) Write down the belief propagation updates (also known as the (parallel) sum-product algorithm) for this factor graph model for the messages $\{\nu_{i \to a}^{(t)}(\cdot)\}_{(i,a) \in E}$ and $\{\tilde{\nu}_{a \to i}^{(t)}(\cdot)\}_{(i,a) \in E}$.

(b) What is the computational complexity (how many operations are required in terms of the degrees of the variable and factor nodes) for updating one message $\nu_{i \to a}(\cdot)$ and one message $\tilde{\nu}_{a \to i}(\cdot)$ respectively? Explain how one can improve the computational complexity, to compute the message $\tilde{\nu}_{a \to i}^{(t)}(\cdot)$ exactly in runtime $O(d_a)$, where d_a is the degree of the factor node a.

(c) Now, we consider a different message passing algorithm introduced by Robert Gallager in 1963. The following update rule is a message passing algorithm known as the **Gallager A algorithm**. Similar to the belief propagation for BEC channels we studied in class, this algorithm also sends discrete messages (as opposed to real-valued messages in part (a)). Both $\nu_{i\to a}^{(t)}$'s and $\tilde{\nu}_{a\to i}^{(t)}$'s are binary, i.e. in $\{+1,-1\}$.

$$\begin{aligned} \nu_{i\to a}^{(t+1)} &= \begin{cases} +1 & \text{if } \tilde{\nu}_{b\to i}^{(t)} = +1 \text{ for all } b \in \partial i \setminus a \;, \\ -1 & \text{if } \tilde{\nu}_{b\to i}^{(t)} = -1 \text{ for all } b \in \partial i \setminus a \;\;, \\ y_i & \text{otherwise} \;, \end{cases} \\ \tilde{\nu}_{a\to i}^{(t)} &= \prod_{j \in \partial a \setminus i} \nu_{j\to a}^{(t)} \;. \end{aligned}$$

The interpretation of this update rule is that $\nu_{i\to a}$ messages trust the received bit y_i unless all of the incoming messages disagree with y_i, and $\tilde{\nu}_{a\to i}$ messages make sure that the consistency with respect to $\mathbb{I}(\otimes x_{\partial a})$ is satisfied. In this algorithm, the messages take values in $\{+1,-1\}$ and are the estimated values of x_i's, as opposed to the distribution over those values as in belief propagation.

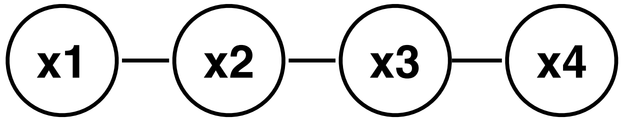
We assume that random (ℓ, r)-regular bipartite graph is used to generate the LDPC code. In the resulting random graph, all variable nodes have degree ℓ and all factor nodes have degree r. Among all such graphs, a random graph is selected uniformly at random.

Define $W^{(t)}$ to be the (empirical) distribution of the messages $\{\nu_{i\to a}^{(t)}\}_{(i,a)\in E}$ and $Z^{(t)}$ to be the (empirical) distribution of the messages $\{\tilde{\nu}_{a\to i}^{(t)}\}_{(i,a)\in E}$. We assume the messages are initialized in such way that $\nu_{i\to a}^{(0)} = y_i$ for all $i \in V$. We also assume, without loss of generality, that all $+1$ messages were sent, i.e. $x_i = +1$ for all i. Then, let $w^{(t)} = \mathbb{P}(W^{(t)} = -1)$ be the probability that a message $\nu_{i\to a}^{(t)}$ is -1 for a randomly chosen edge (i,a), and let $z^{(t)} = \mathbb{P}(Z^{(t)} = -1)$ be the probability that a message $\tilde{\nu}_{a\to i}^{(t)}$ is -1 for a randomly chosen edge (i,a).

Write the **density evolution** *equations for $w^{(t)}$ and $z^{(t)}$, describing how the random distribution of the messages $w^{(t)}$ and $z^{(t)}$ evolve.* [We are looking for a clean answer. Specifically, the number of operations required to compute $z^{(t)}$ from $w^{(t)}$ should be $O(1)$. The same technique that reduced computation in part (b) should be helpful.]

(d) Write the density evolution equation for a single scalar variable $w^{(t)}$, by substituting $z^{(t)}$. This gives a fixed point equation in the form of $w^{(t)} = F(w^{(t-1)})$ for some F. Plot (using your favorite numerical analysis tool) the function $y = F(x)$ and the identity function $y = x$, for $\ell = 3$ and $r = 4$, and for two values of $\varepsilon = 0.05$ and $\varepsilon = 0.1$. Explain the error probability achieved by this algorithm for the (3,4)-code on those two BSC(ε) channels. Use the fixed point equation to calculate (numerically) the probability of error for this (3,4)-code on those two values of ϵ.

5. *[2 points]* **(Max-product algorithm)**



Consider an MRF on a line graph with 4 nodes: $x = [x_1, x_2, x_3, x_4]$. Each node has a ternary alphabet, i.e. $x_i \in \{a, b, c\}$. Suppose that the joint probability for all 81 possible state sequences are *distinct*, i.e. no two realizations have the same probability mass.

Recall that max-marginal is, for example,

$$\tilde{\mu}(x_2) \triangleq \max_{x_{\{1,3,4\}} \in \{a,b,c\}^3} \mu(x_1, x_2, x_3, x_4)$$

We ran the max-product algorithm and recorded the resulting max-marginals for each node $i \in \{1, 2, 3, 4\}$ and each value in the table below.

i	$\tilde{\mu}(x_i = a)$	$\tilde{\mu}(x_i = b)$	$\tilde{\mu}(x_i = c)$
1	0.2447	0.0753	0.0234
2	0.2447	0.0118	0.1199
3	0.2447	0.1199	0.0169
4	0.2447	0.0346	0.0141

In this problem, we want to find the k-th most likely instance $x^{(k)} \in \{a, b, c\}^4$ for $k \in \{1, 2, 3\}$. Here, the k-th most likely instance means a specific joint state $x^{(k)} = [x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)}]$ whose joint probability is the k-th largest among all 81 instances.

(a) Find the most likely instance $x^{(1)}$ and the corresponding probability $\mu(x^{(1)})$ using the information in the given max-marginals in the above table. Explain your answer.

(b) Find the second most likely instance $x^{(2)}$ and the corresponding probability $\mu(x^{(2)})$ using the given max-marginals above. Explain your answer. Is it uniquely determined? Can $x^{(2)}$ be uniquely determined from max-marginals (like the table above) assuming joint probability masses are all distinct, in the general case, for any alphabet size, graph, and number of nodes?

(c) (For this problem ignore the constraints imposed by the structure of the graph.) Given the max-marginals in the above table, list all sequences that can be the third most likely instance $x^{(3)}$. Explain your answer. Can the corresponding probability $\mu(x^{(3)})$ be uniquely determined? For which instances of $x^{(3)}$ in the list you provided, can $\mu(x^{(3)})$ be uniquely determined?

(d) Using the structure of the graph (and the corresponding factorization), and the fact that $\mu(x^{(1)}) > \mu(x)$ for all $x \neq x^{(1)}$, find which instances in the list you provided in the previous step cannot be $x^{(3)}$. Meaning, eliminate as many instances in the previous list by considering the graph structure. Explain your answer.

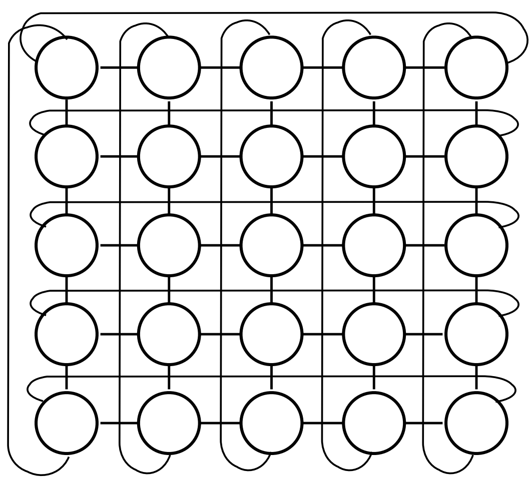
(e) We consider the same (unknown) true distribution as above subproblems. Suppose that instead of computing the *node* max-marginal data above, we ran a different algorithm and computed *edge* max-marginals, for example $\tilde{\mu}(x_2, x_3) \triangleq \max_{x_1, x_4} \mu(x_1, x_2, x_3, x_4)$, for every edge $(i, j) \in E$. For example, we know that $\tilde{\mu}(x_1 = a, x_2 = a) = \tilde{\mu}(x_2 = a, x_3 = a) = \tilde{\mu}(x_3 = a, x_4 = a) = 0.2447$. We also know that $\tilde{\mu}(x_1 = a, x_2 = c) = \tilde{\mu}(x_2 = c, x_3 = b) = \tilde{\mu}(x_3 = b, x_4 = a) = 0.1199$. We know all such pairwise max-marginals. With this edge-max-marginals, can we uniquely determine the third likely instance $x^{(3)}$? Explain your answer. Can $x^{(4)}$ be uniquely determined?

6. *[2 points]* **(Convergence of belief propagation)**
Although belief propagation on graphs with loops is challenging to analyze, we consider a particular example in this problem with highly symmetric structure.

An *Ising model* on a vector of binary variable x, with $x_i \in \{-1, +1\}$ is represented by a set of parameter vector $\theta = [\{\theta_i\}_{i\in[n]}, \{\theta_{ij}\}_{(i,j)\in E}]$ as

$$\mu_\theta(x) = \frac{1}{Z_\theta} \exp\Big\{ \sum_{i\in V} \theta_i x_i + \sum_{(i,j)\in E} \theta_{ij} x_i x_j \Big\}$$

for some given graph $G = (V, E)$. In this problem, we focus on a simple case where $\theta_i = 0$ for all $i \in V$, $\theta_{ij} = \gamma$ for all $(i,j) \in E$ for some positive $\gamma > 0$, and the graph is a toroidal grid, as shown below. Each node is connected to its four neighbors in the grid, and the grid is wrapped around at the boundaries. We will consider general toroidal grid with more than 3×3 nodes.



(*a*) Show, by explicitly analyzing the marginal and also by symmetry, that the single node marginal distributions are uniform for all values of γ, i.e. $\mu(x_i = 1) = \mu(x_i = -1) = 0.5$ for all i. [hint: start with smaller examples]

(*b*) Write down the sum-product algorithm update rules for the messages $\nu^{(t)}_{i\to j}(x_i)$.

(*c*) Write down the sum-product update rules again, but with a change of variables. Let

$$q^{(t)}_{i\to j} = \frac{\nu^{(t)}_{i\to j}(x_1 = -1)}{\nu^{(t)}_{i\to j}(x_1 = +1)}$$

and rewrite the sum-product update rules in terms of $q^{(t)}_{i\to j}$'s.

(*d*) Suppose we initialize the messages $q^{(0)}_{i\to j}$'s with the same value (which might not necessarily be one) such that $q^{(0)}_{i\to j} = q^{(0)} \in \mathbb{R}$ for all $(i,j) \in E$. Then, by the symmetry of the graph and the update rules, all subsequent messages will be the same for all edges, i.e. $q^{(t)}_{i\to j} = q^{(t)}$ for all $(i,j) \in E$. Now, write a single sum-product update rule (that does not depend on the particular edge), by substituting all the messages by $q^{(t)}$ and simplifying the formula you got in the previous step. Define a function $f : \mathbb{R} \to \mathbb{R}$ such that $q^{(t+1)} = f(q^{(t)})$ denote this update rule. Plot the function $y = x$ and $y = f(x)$ for $x \in [0, 2]$ and $y \in [0, 2]$ for two choices of $\gamma \in \{0.3, 0.45\}$. How many fixed points are there in each plot (a fixed point of $q^{(t+1)} = f(q^{(t)})$ is where $y = x$ and $y = f(x)$ cross)?

(e) Notice that we start sum-product algorithm with the usual initialization of $\nu^{(0)}_{i\to j}(x_i = +1) = \nu^{(0)}_{i\to j}(x_i = -1) = 0.5$ (or equivalently $q^{(0)} = 1$), then the messages stay at this uniform distribution and do not change. The reason is that the set of all-uniform messages is a fixed point in the sum-product update equation (or BP equation). This means that all-uniform messages are not changed after sum-product update.

However, depending on the value of γ this fixed point is either stable or unstable. If it is a stable fixed point, the BP will still converge to the same fixed point, even if we start at slightly perturbed initialization. For example let us initialize each message as $\nu^{(0)}_{i\to j}(x_i = +1) = 0.5 + \varepsilon$ and $\nu^{(0)}_{i\to j}(x_i = -1) = 0.5 - \varepsilon$ for some small ε. This is equivalent to initializing $q^{(0)} = \frac{0.5-\varepsilon}{0.5+\varepsilon}$. If the fixed point is stable, sum-product algorithm will still converge to the all-uniform messages. Otherwise, the messages will converge to another fixed point (or diverge). Using the two plots you draw in the previous step, identify whether $q^{(0)} = 1$ is a stable fixed point or not for $\gamma = 0.3$ and also for $\gamma = 0.45$. Explain your answer.

(f) The necessary and sufficient condition for a function f to be stable at a point $q^{(0)}$ is that $f'(q(0)) < 1$. Analytically find the threshold γ^* below which the all-ones initialization is stable, and above which it is not.