Exercises

Graphical models

1 Prerequisite

For the students taking this course, it is expected that you have the skills and backgrounds to solve the problems in this homework (it is encouraged that you talk to your peers to figure the solutions out). If you have trouble following any of these problems, you need to talk to the instructor.

Basic probability.

1.1 For three discrete random variables x, y, and z with joint probability distribution $p(x, y, z)$, we say x is **conditionally independent** of y given z if and only if $p(x, y|z) = p(x|z)p(y|z)$, where $p(x|z) = \frac{p(x,z)}{p(z)} = \frac{\sum_{y'} p(x,y',z)}{\sum_{x',y'} p(x',y',z)}$ is the conditional probability distribution of x given z, and $p(x, y|z)$ and $p(y|z)$ are defined similarly. We use x–z–y to denote that x and y are independent conditioned on z. We use x–()–y to denote that x is independent of y. Prove each of the following properties.

1. x–()–(y, z) implies x–z–y
2. x–z–(y, w) implies x–z–y
3. x–z–(y, w) and y–z–w implies (x, w)–z–y

1.2 There are two coins, one is a fair coin and the other is biased. The outcome of tossing a fair coin is a head (H) with probability half. The outcome of tossing a biased coin is a head (H) with probability 3/4. We are given a coin at random (equal probability of getting a fair or biased coin), and want to test whether the coin is biased or not based on n coin tosses. We toss the coin 5 times independently and get (H, H, T, T, H). What is the probability that the coin is a biased coin? Then, what is your **maximum a posteriori (MAP) estimate**? Does it depend on the order of the outcome?

Basic linear algebra.

1.3 An $n \times n$ dimensional symmetrix matrix A is **positive definite** if $x^T Ax > 0$ for any vector $x \in \mathbb{R}^n$, where x^T is the transpose of the column vector x. It is **positive semidefinite** if $x^T Ax \geq 0$ for any x. Prove that if A has eigen values which are all positive, then A is positive definite.
[Hint: A symmetric matrix A can be factorized by eigen decomposition as $A = Q\Lambda Q^T$. Q is a unitary matrix such that $QQ^T = Q^T Q = I$, where I is the n-dimensional identity matrix, and Λ is a diagonal matrix with the eigen values λ_i of matrix A in the diagonals. Then, we are left ot show that if Λ is a diagonal matrix with strictly positive entries, then $y^T \Lambda y > 0$, where we changes variables by setting $y = Q^T x$.].

1.4 Find a vector y^* in terms of Q_{ij}'s, h_i's and x that maximizes a **quadratic function**

$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} h_1 & h_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

[Note: the maximizer $y^*(x)$ is a linear function of x.]

2 Definition of graphical models

2.1 **(Exercise 2.5 in Koller/Friedman)**

Let X, Y, Z be three disjoint subsets of random variables. We say X and Y are conditionally independent given Z if and only if

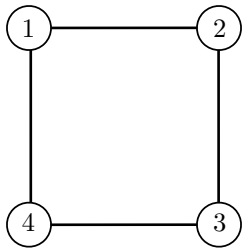
$$\mathbb{P}_{X,Y|Z}(x, y|z) \quad = \quad \mathbb{P}_{X|Z}(x|z)\, \mathbb{P}_{Y|Z}(y|z)\ .$$

Show that X and Y are conditionally independent given Z if and only if the joint distribution for the three subsets of random variables factors in the following form:

$$\mathbb{P}_{X,Y,Z}(x, y, z) \quad = \quad h(x, z)\, g(y, z)\ .$$

2.2 **(Exercise 4.1 in Koller/Friedman)**

In this problem, we will show by example that the distribution of a graphical model need not have a factorization of the form in the Hammersley-Clifford Theorem if the distribution is not strictly positive. In particular, we will consider a distribution on the following simple 4-cycle where each node is a binary



random variable, X_i, for $i \in \{1, 2, 3, 4\}$. Consider a probability distribution that assigns a probability $1/8$ uniformly to each of the following set of values (X_1, X_2, X_3, X_4):

$$\begin{array}{cccc} (0,0,0,0) & (1,0,0,0) & (1,1,0,0) & (1,1,1,0) \\ (0,0,0,1) & (0,0,1,1) & (0,1,1,1) & (1,1,1,1) \end{array}$$

and assigns zero to all other configurations of (X_1, X_2, X_3, X_4).

(a) We first need to show that this distribution is Markov on our graph. To do this, it should not be difficult to see that what we need to show are the following conditions:

- The pair of variables X_1 and X_3 are conditionally independent given (X_2, X_4).
- The pair of variables X_2 and X_4 are conditionally independent given (X_1, X_3).

First, show that if we interchange X_1 and X_4 and interchange X_2 and X_3, we obtain the same distribution, i.e.., $\mathbb{P}(x_1, x_2, x_3, x_4) = \mathbb{P}(x_4, x_3, x_2, x_1)$. This implies that if we can show the first condition, then the other is also true.

(b) Show that whatever pair of values you choose for (X_2, X_4), we then know either X_1 or X_3 with certainty. For example, $(X_2 = 0, X_4 = 0)$ implies that $X_3 = 0$. Since we know either X_1 or X_3 with certainty, then conditioning on the other one of these obviously provides no additional information, trivially proving conditional independence.

(*c*) What we now need to show is that the distribution cannot be factorized in the way stated in the Hammersley-Clifford Theorem. We will do this by contradiction. Noting that the maximal cliques in our graph are just the edges and absorbing the normalization $1/Z$ into any of the pairwise compatibility functions, we know that if our distribution has the factorization implied by the Hammersley-Clifford Theorem, we can write it in the following form:

$$\mathbb{P}(x_1, x_2, x_3, x_4) \quad = \quad \psi_{12}(x_1, x_2)\, \psi_{23}(x_2, x_3)\, \psi_{34}(x_3, x_4)\, \psi_{41}(x_4, x_1)\ .$$

Show that assuming that our distribution has such a factorization leads to a contradiction by examining the values of $\mathbb{P}(0,0,0,0)$, $\mathbb{P}(0,0,1,0)$, $\mathbb{P}(0,0,1,1)$, and $\mathbb{P}(1,1,1,0)$.

2.3 Given a graph $G = (V, E)$, an *independent set* of G is a subset $S \subseteq V$ of the vertices, such that no two vertices in S is connected by an edge in E. Precisely, if $i, j \in S$ then $(i, j) \notin E$. We let $\mathrm{IS}(G)$ denote the set of all independent sets of G, and let $Z(G) = |\mathrm{IS}(G)|$ denote its size, i.e. the total number of independent sets in G. The number of independent sets $Z(G)$ is at least $1 + |V|$, since the empty set and all subsets with single vertex are always independent sets. We are interested in the uniform probability measure over S:

$$\mathbb{P}_{\mathrm{IS}(G)}(S) = \frac{1}{Z(G)} \mathbb{I}(S \in \mathrm{IS}(G))\ ,$$

where $\mathbb{I}(A)$ is an indicator function which is one if event A is true and zero if false.

(*a*) The set S can be naturally encoded by a binary vector $x \in \{0,1\}^{|V|}$ by letting $x_i = 1$ if and only if $i \in S$. Denote by $\mathbb{P}_G(x)$ the probability distribution induced on this vector x according to $\mathbb{P}_{\mathrm{IS}(G)}(S)$. Show that $\mathbb{P}_G(x)$ is a pairwise graphical model on G.
[Hint: A pairwise graphical model on a graph $G = (V, E)$ is defined by a factorization of the form $\mathbb{P}_G(x) = (1/Z) \prod_{(i,j)\in E} \psi_{i,j}(x_i, x_j)$.]

(*b*) Let L_n be the line graph with n vertices, i.e. the graph with vertex set $V(L_n) = \{1, 2, 3, \ldots, n\}$ and edge set $E(L_n) = \{(1,2), (2,3), \ldots, (n-1, n)\}$. Derive a formula for $Z(L_n)$.
[Hint: Write a recursion over n, and solve it using a matrix representation.]

(*c*) With the above definitions, derive a formula for $\mathbb{P}_{L_n}(x_i = 1)$, for each $i \in \{1, \ldots, n\}$. Plot $\mathbb{P}_{L_n}(x_i)$ versus i for $n = 11$. Describe the main features of this plot. Can you give an intuitive explanation?
[Hint: Use the recursion from previous subproblem.]

(*d*) The same probability distribution $\mathbb{P}_{L_n}(x)$ can be also represented with a Bayesian network. For example, $\mathbb{P}_G(x) = \mathbb{P}_{X_1}(x_1)\mathbb{P}_{X_2|X_1}(x_2|x_1)\mathbb{P}_{X_3|X_1,X_2}(x_3|x_1, x_2) \cdots \mathbb{P}_{X_{11}|X_1 \cdots X_{10}}(x_{11}|x_1 \cdots x_{10})$. Using the recursions used in (*b*) and (*c*), write the conditional probability distributions for this Bayesian network.

2.4 We again consider the independent set explained in the previous problem. Now let $G = T_{k,\ell}$ denote the rooted tree with branching factor k and ℓ generations, that is the root has k descendants and each other node has one ancestor and k descendants except for the leaves. The total number of vertices is $(k^{\ell+1} - 1)/(k-1)$, and $T_{k,\ell=0}$ is the graph consisting only of the root. We let ϕ denote the root of $T_{k,\ell}$.

(*a*) Let $Z_\ell = Z(T_{k,\ell})$ denote the total number of independent sets of $G = T_{k,\ell}$. Let $Z_\ell(0)$ be the number of independent sets in $T_{k,\ell}$ such that the root is $x_\phi = 0$, and $Z_\ell(1)$ be the number of independent sets such that $x_\phi = 1$. It is immediate that $Z_0(0) = Z_0(1) = 1$. Derive a recursion expressing $(Z_{\ell+1}(0), Z_{\ell+1}(1))$ as a function of $(Z_\ell(0), Z_\ell(1))$.

(*b*) Using the above recursion, derive a recursion for the probability that the root belongs to a uniformly random independent set. Explicitly, derive a recursion for

$$p_\ell \quad = \quad \mathbb{P}_{T_{k,\ell}}(\{x_\phi = 1\}) \; .$$

(*c*) Program this recursion and plot p_ℓ as a function of $\ell \in \{0, 1, \ldots, 50\}$ for four values of k, e.g. $k \in \{1, 2, 3, 10\}$. Comment on the qualitative behavior of these plots.

(*d*) Prove that, for $k \leq 3$, the recursion converges to a unique value using Banach's fixed point theorem.

3 Markov properties

3.1 **(Intersection lemma)**
In proving that pairwise Markov property implies global Markov property for undirected graphical models, we used the intersection lemma which states that if μ is strictly positive and

$$A\text{–}(C \cup D)\text{–}B, \qquad A\text{–}(B \cup D)\text{–}C,$$

then

$$A\text{–}D\text{–}(B \cup C)\,.$$

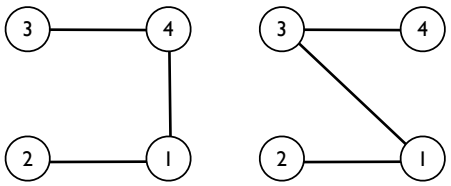
Here A–B–C if and only if $\mu(x_A, x_C|x_B) = \mu(x_A|x_B)\,\mu(x_C|x_B)$. From previous homework, we know that A–$(C \cup D)$–B if and only if $\mu(x_A, x_B, x_C, x_D) = a(x_A, x_C, x_D)\,b(x_B, x_C, x_D)$ for some function $a(\cdot)$ and $b(\cdot)$. Similarly, we have $\mu(x_A, x_B, x_C, x_D) = f(x_A, x_B, x_D)\,g(x_B, x_C, x_D)$.

(*a*) Show $f(x_A, x_B, x_D) = a'(x_A, x_D)\,b'(x_B, x_D)$ for some $a'(\cdot)$ and $b'(\cdot)$ and find one such pair of functions a' and b' in terms of $a(), b(), g()$.

(*b*) Substitute $f(\cdot)$ and prove A–D–$(B \cup C)$.

(*c*) Find a counter example when μ is not strictly positive.

3.2 **(I-map)**
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 $(i,j) \notin E$,

$$x_i \perp x_{V \setminus \{i,j\}} \perp x_j$$

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 $i \in V$,

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

(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.

3.3 **(Markov property)**
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

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

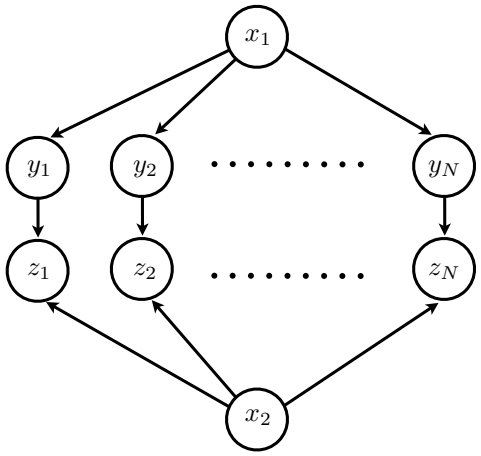
and $\tilde{p}_{Y_i|X_i,T_i}(y_i|x_i, t_i)$, respectively. Express $\phi(x_i, x_{i+1}, t_i)$, $\xi(x_i, t_i)$, and $\tilde{p}_{Y_i|X_i,T_i}(y_i|x_i, t_i)$ in terms of parameters p_1, p_D, p_T, $p_{Y_i|X_i}$, k, N, and M of the original model.

(*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 $O(Nk^2M)$.]

3.4 **(I-map)**
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

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

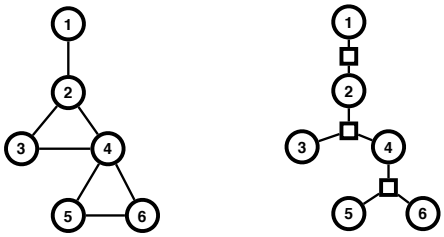
for some function $\eta(x_1, c, z_c)$ that can be evaluated in $O(K)$ operations for fixed (x_1, c, z_c), and some function $\lambda(c, z_1, \ldots, z_N)$ that can be evaluated in $O(N)$ operations for fixed $(c, z_1, \ldots, z_N)$. Express $\eta(x_1, c, z_c)$ in terms of $p_{Y|X_1}$ and $p_{Z|Y,X_2}$,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

3.5 **(Perfect map)**

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.

3.6 **(Converting MRF and FG)**

A straight forward method for converting a Markov Random Field (the general model and not the pairwise model) into a factor graph model is to write the joint distribution as a product of factors, each factor corresponding to a maximal clique in the MRF. Hence, an undirected graph of the MRF can be translated into a factor graph by simply defining a factor node for each maximal clique. Below is an example of such translation. We call this translation a *canonical construction*.



undirected graphical model associated factor graph

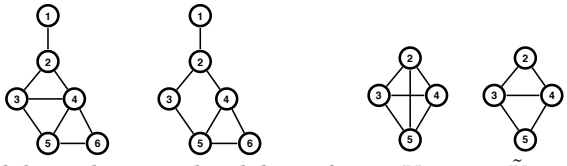
We want to show by constructing an example that the factor graph produced by the canonical construction can have the number of factor nodes exponential in the number of variable nodes. Specifically, we will show that there exists a constant $c > 1$ and an undirected graph with n nodes such that the associated factor graph has at least c^n factor nodes for sufficiently large n.

(*a*) First, consider a *complete* undirected graph with n nodes, where all pairs of nodes are connected by edges. We denote this graph by K_n. How many factors does the factor graph have that is the canonical construction of this K_n?

(*b*) Choose two nodes arbitrarily and erase the edge connecting those two nodes from K_n and call the resulting graph $K_n^{(-1)}$. How many factors does the canonical construction of this $K_n^{(-1)}$ have?

(*c*) There are two choices: choose two new nodes that are different from those chosen in the previous step and erase the edge connecting that pair, or choose one new node and one of the node that was chosen in the previous step and erase the edge connecting those two nodes. In the former case, let's call the resulting undirected graph $K_n^{(-2)}$, and the latter, we call the resulting undirected graph $K_n^{(-1.5)}$. How many factors nodes does the canonical construction associated with $K_n^{(-2)}$ have? How many factors nodes does the canonical construction associated with $K_n^{(-1.5)}$ have?

(*d*) Suppose n is an even number. Explain how to use the above procedures to construct an undirected graph whose canonical construction has number of factors $2^{n/2}$.



chordal graph non-chordal graph K_4 $\tilde{K}_4$

(*e*) Next, given an undirected graph G, we want to check whether the associated canonical construction is a tree or not. A naive attempt is to produce the canonical construction, and check for loops. This approach can be extremely inefficient, since the size of the resulting factor graph (the number of factors plus the number of variables) can be exponential in n as we showed previously.

We will show next that there is a polynomial time algorithm for checking that the factor graph resulting from canonical construction is a tree. First recall that G is *chordal* if any cycle of length 4 or more nodes has a chord, which is an edge joining two nodes that are not adjacent in the cycle. Note that testing for cordiality of a given G can be done in linear time.

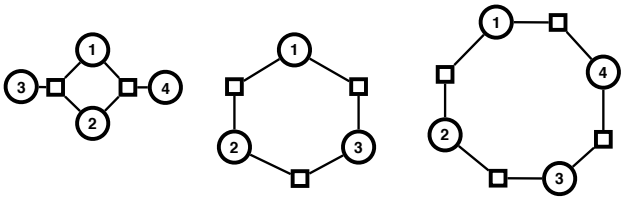
Recall that a clique containing 4 nodes is denoted by K_4. We use $\tilde{K}_4$ to denote the graph generated from deleting one edge from K_4, and we say a graph G contains $\tilde{K}_4$ if it has a subgraph which is $\tilde{K}_4$. Note that testing whether a graph contains $\tilde{K}_4$ can be done in polynomial time (e.g. brute-force search takes time $O(n^4)$).

Prove that if G contains $\tilde{K}_4$ then the resulting canonical construction is not a tree.

(*f*) Prove that if G is not chordal then the resulting canonical construction is not a tree.

(*g*) The above two statements prove that "if G contains $\tilde{K}_4$ or is not chordal, then canonical construction is not a tree". We now show the converse to this statement. Specifically, we will show that "if the canonical construction has at least one loop, then G contains $\tilde{K}_4$ or is not chordal".

First show that a canonical construction can have chordless loops of length 4 or of lengths 8 or larger even number, but not of length 6.



loop of length 4 loop of length 6 loop of length 8

(h) Show that if canonical construction has a (chordless) loop of length 4, then G has $\tilde{K}_4$.

(i) Show that if canonical construction has a (chordless) loop of length 8 or larger, then G is not chordal, i.e. G contains a loop of length larger than or equal to 4 with no chord.

This proves, together with the previous statement, the desired converse. Hence, we can check for trees in resulting factor graphs, by checking that the original graph is chordal and does not contain $\tilde{K}_4$, which can be done in polynomial time.

4 Sum-product algorithm (belief propagation)

4.1 **(Belief propagation)**
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.

4.2 **(Belief propagation)**
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

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

[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 $|i_1 - j_1| = 1$.

(*b*) We give a MATLAB implementation of the code `bpsol.m`, a Python implementation `bpsol.py`, and a iPython note book `bpsol.ipynb` you can download from the course website. Feel free to use whichever you feel comfortable with. Make yourself familiar with it to answer the following questions.

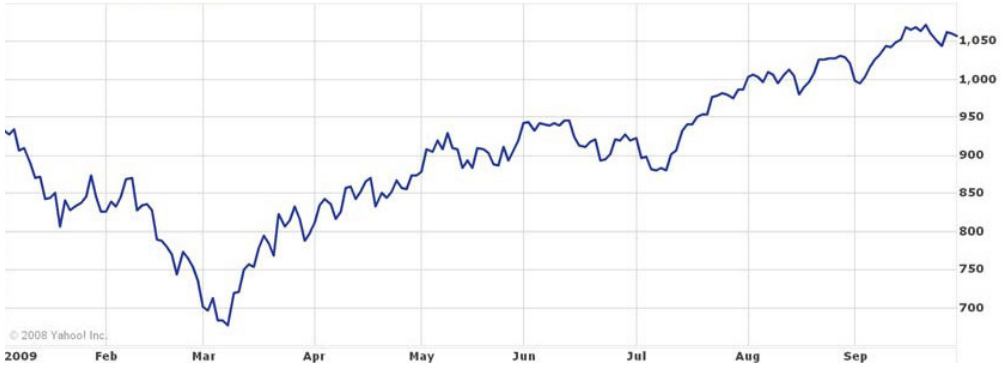
(*c*) Consider the case $\ell = 10$ (and hence $n = 100$ nodes). For each $\beta \in \{0.2, 0.4, \ldots, 2.8, 3.0\}$, generate an instance by drawing θ_i, θ_{ij} uniformly random in $[0, \beta]$. Run the BP iteration and monitor convergence by computing the quantity

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

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.

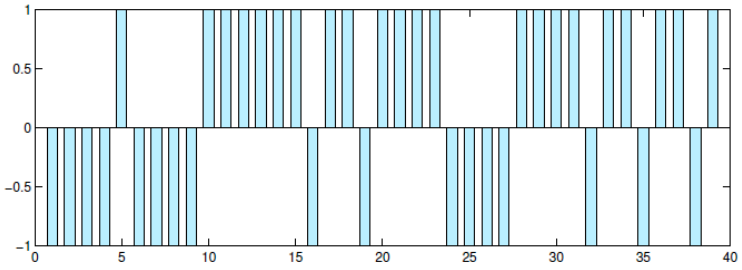
4.3 **(Hidden Markov models; implementation)**
In this problem, you will implement the sum-product algorithm on a line graph and analyze the behavior of S&P 500 index over a period of time. The following figure shows the price of S&P 500 index from January 2, 2009 to September 30, 2009 (`http://finance.yahoo.com`).



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).

4.4 **(Hidden Markov models)**
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

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

Determine the function $f(\cdot)$.

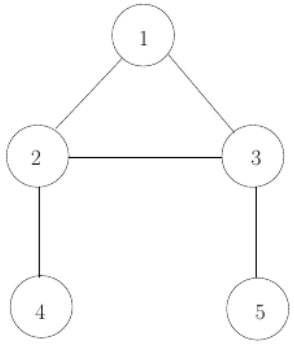
(b) Show that if only the marginal distributions $\mu(x_i|y_1 \ldots, y_n)$, $i \in \{1, \ldots, n\}$ for the model are available, the desired state estimates cannot be determined. In particular, construct two HMMs whose marginals coincide, but whose state estimates differ.
[Hint: it sufficies to consider 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.

4.5 **(Belief propagation)**
Consider the following graphical model.



(*a*) Draw a factor graph representing the graphical model and specify the factor graph message-passing equations. For this particular example, explain why the factor graph message-passing equations can be used to compute the marginals, but the sum-product equations for pairwise MRF cannot be used.

(*b*) Define a new random variable $x_6 = \{x_1, x_2, x_3\}$, i.e., we group variables x_1, x_2, and x_3 into one variable. Draw an undirected graph which captures the relationship between x_4, x_5, and x_6. Explain why you can apply the sum-product algorithm to your new graph to compute the marginals. Compare the belief propagation equations for the new graph with the factor graph message-passing equations you obtained in part (*a*).

(*c*) If we take the approach from part (*b*) to the extreme, we can simply define a random variable $x_7 = \{x_1, x_2, x_3, x_4, x_5\}$, i.e., define a new random variable which groups all five original random variables together. Explain what running the sum-product algorithm on the corresponding one vertex graph means. Assuming that we only care about the marginals for $x_1, x_2, \ldots, x_5$, can you think of a reason why we would prefer the method in part (*b*) to the method in this part, i.e., why it might be preferable to group a smaller number of variables together?

4.6 **(Belief propagation)**
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

$$\mu(x,y) \quad = \quad \frac{1}{Z}\prod_i \phi(x_i,y_i)\prod_{(j,k)\in E}\psi(x_j,x_k) \tag{3}$$

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.

(*a*) Sketch an undirected graphical model that represents $\mu(x,y)$.

(*b*) Compute $\mu_\alpha \in \mathbb{R}^3$ and $\Lambda_\alpha \in \mathbb{R}^{3\times 3}$ for each $\alpha \in \{0,1\}$ 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 $\{y_1,\ldots,y_N\}$ is $\bar{y}=\frac{1}{N}\sum_{i=1}^N y_i$ and the sample covariance is $C_y=\frac{1}{N-1}\sum_{i=1}^N(y_i-\bar{y})(y_i-\bar{y})^T$.

(*c*) 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 (3) 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

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

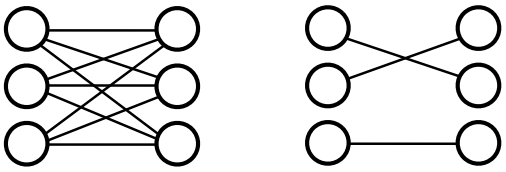
Implement the sum-product algorithm for this problem. There are four directional messages: down, up, left, and right, coming into and out of each x_i (except at the boundaries). Use a parallel update schedule, so all messages at all x_i are updated at once. Run for 30 iterations (or you can state and use some other reasonable termination criterion). Since we are working with binary random variables, perhaps it is easier to pass messages in log-likelihood. Feel free to use `gridbpsol.m` or `gridbpsol.py` from the website for running the BP algorithm.
After the marginal distributions at the pixels are estimated, visualize their expectation. Where are the beliefs "weak"?
Visualize the expectation after 1, 2, 3, and 4 iterations. Qualitatively, discuss where the loopy belief propagation converge first and last.
Run BP with a different value of $\epsilon = 0$ and comment on the result.
Run BP with a different pairwise potential and comment on the result.

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

4.7 **(Applying belief propagation)**

In this problem, we apply inference techniques in graphical models to find *Maximum Weight Matching (MWM)* in a complete bipartite graph. This is one of a few problems where belief propagation converges and is correct on a general graph with loops. Other such examples include Gaussian graphical models studied in class.

Consider an undirected weighted complete bipartite graph $G(X, Y, E)$ where X is a set of n nodes and Y is another set of n nodes: $|X| = |Y| = n$. In a bipartite complete graph all the nodes in X are connected to all the nodes in Y and vice versa, as shown below. Further, each edge in this graph is



a complete bipartite graph a perfect matching

associated with a real valued weight $w_{ij} \in \mathbb{R}$. A *matching* in a graph is a subset of edges such that no edges in this matching share a node. A matching is a *perfect matching* if it matches all the nodes in the graph. Let $\pi = (\pi(1), \ldots, \pi(n))$ be a permutation of n nodes. In a bipartite graph, a permutation π defines a perfect matching $\{(i, \pi(i)\}_{i \in \{1,\ldots,n\}}$. From now on, we use a permutation to represent a matching. A *weight* of a (perfect) matching is defined as $W_\pi = \sum_{i=1}^{n} w_{i,\pi(i)}$. The problem of *maximum weight matching* is to find a matching such that

$$\pi^* \quad = \quad \arg\max_{\pi} W_\pi \ .$$

We want to solve this maximization by introducing a graphical model with probabilty proportional to the weight of a matching:

$$\mu(\pi) \quad = \quad \frac{1}{Z}\, e^{CW_\pi}\, \mathbb{I}(\pi \text{ is a perfect matching}) \ ,$$

for some constant C.

(*a*) The set of matchings can be encoded by a pair of vectors $x \in \{1, \ldots, n\}^n$ and $y \in \{1, \ldots, n\}^n$, where each node takes an integer value from 1 to n. With these, we can represent the joint distribution as a pari-wise graphical model:

$$\mu(x, y) = \frac{1}{Z} \prod_{(i,j) \in \{1,\ldots,n\}^2} \psi_{ij}(x_i, y_j) \prod_{i=1}^{n} e^{w_{i,x_i}} \prod_{i=1}^{n} e^{w_{y_i,i}} \ ,$$

where $\psi_{ij}(x_i, y_j) = \begin{cases} 0 & x_i = j \text{ and } y_j \neq i \ , \\ 0 & x_i \neq j \text{ and } y_j = i \ , \\ 1 & \text{otherwise} \ . \end{cases}$ Show that for the pairwise graphical model defined above, the joint distribution $\mu(x, y)$ is non-zero if and only if $\pi_x = \{(1, x_1), \ldots, (n, x_n)\}$ and $\pi_y = \{(y_1, 1), \ldots, (y_n, n)\}$ both are matchings and $\pi_x = \pi_y$. Further, show that when non-zero, the probability is equal to $\frac{1}{Z} e^{2W_{\pi_x}}$.

(*b*) Let

$$(x^*, y^*) \quad = \quad \arg\max_{x,y} \mu(x, y) \; .$$

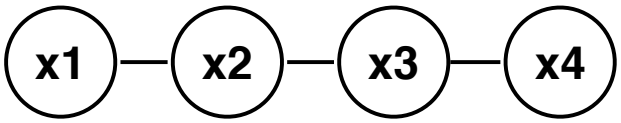
Show that $\pi_{x^*} = \pi_{x^*}$ is the maximum weight matching on the given graph G with weights $\{w_{ij}\}$.

(*c*) Let us denote by l_i the i-th 'left' node in X corresponding to the random variable x_i, and by r_j the j-th 'right' node in Y corresponding to the random variable y_j. We are going to derive *max-product* update rules for this problem. Let $\nu_{l_i \to r_j}(x_i)^{(t)}$ denote the message from a left node l_i to a right node r_j at t-th iteration, which is a vector of size n. Similarly, we let $\nu_{r_j \to l_i}(y_j)^{(t)}$ denote the message from a right node r_j to a leftt node l_i. We initialize all the messages such that

$$\begin{aligned} \nu^{(0)}_{l_i \to r_j}(x_i) &= e^{w_{i,x_i}} \; , \\ \nu^{(0)}_{r_j \to l_i}(y_j) &= e^{w_{y_j,j}} \; . \end{aligned}$$

Write the message update rule for the message $\nu^{(t+1)}_{l_i \to r_j}(x_i)$ and $\nu^{(t+1)}_{r_j \to l_i}(y_j)$ as functions of messages from previous iterations.

4.8 **(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*) 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 $(i, j) \in E$. With this edge-max-marginals, can we uniquely determine the third likely instance $x^{(3)}$? Explain your answer. Can $x^{(4)}$ be uniquely determined?

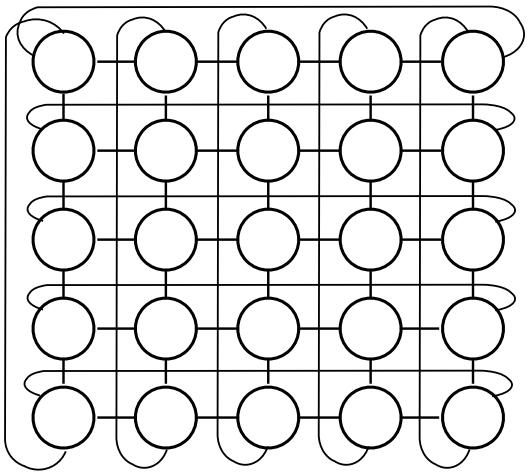
4.9 **(Convergence of belief propagation)**

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

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

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 $\nu_{i\to j}^{(t)}(x_i)$.

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

$$q_{i\to j}^{(t)} \;=\; \frac{\nu_{i\to j}^{(t)}(x_1 = -1)}{\nu_{i\to j}^{(t)}(x_1 = +1)}$$

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

(*d*) Suppose we initialize the messages $q_{i\to j}^{(0)}$'s with the same value (which might not necessarily be one) such that $q_{i\to j}^{(0)} = q^{(0)} \in \mathbb{R}$ 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. $q_{i\to j}^{(t)} = q^{(t)}$ 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 $\nu_{i\to j}^{(0)}(x_i = +1) = \nu_{i\to j}^{(0)}(x_i = -1) = 0.5$ (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 $\nu_{i\to j}^{(0)}(x_i = +1) = 0.5 + \varepsilon$ and $\nu_{i\to j}^{(0)}(x_i = -1) = 0.5 - \varepsilon$ 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.

5 Density evolution

5.1 **(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 $\{+1, -1\}$.

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

$$\tilde{\nu}_{a\to i}^{(t)} = \prod_{j\in\partial a\setminus i} \nu_{j\to a}^{(t)} \ .$$

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 MATLAB to your favorite numerical analysis tool) the function $y = F(x)$ and the identify function $y = x$, for $\ell = 3$ and $r = 4$, and for two values of $\varepsilon = 0.05$ and $\varepsilon = 0.1$ Explain the figure in terms of the error probability of the (3,4)-code on those two BSC(ε)'s.

5.2 **(Application of crowdsourcing; implementation)** From the lecture, we studied a message passing algorithm (developed as a belief propagation for Haldane prior):

- initialize: $y_{j\to i}^{(0)}$'s as independent and identically distributed Gaussian random variable with mean one and variance one (this is one choice of initialization and any reasonable choice works as well)
- update messages:

$$x_{i\to j}^{(\tau+1)} = \sum_{k\in\partial i\setminus j} y_{j\to i}^{(\tau)} A_{ik}$$

$$y_{j\to i}^{(\tau+1)} = \sum_{k\in\partial j\setminus i} x_{k\to j}^{(\tau+1)} A_{kj}$$

- after enough number (e.g. T) of iterations estimate each task label by

$$\hat{t}_i = \text{sign}\left(\sum_{k\in\partial i} y_{k\to i}^{(T)} A_{ik}\right)$$

We will implement this algorithm for the following setting:

- the number of tasks $n = 100$
- the number of workers $m = 100$
- the (average) degree of a task node is ℓ
- the (average) degree of a worker node is also ℓ
- generate random graph as follows: for each task-worker pair (i, j), connect the two nodes with an edge with probability ℓ/m and otherwise do not connect with an edge: for example you can use the following Matlab script to generate such a graph with adjacency matrix E

  ```
  E = zeros(n,m);
  E = ceil( rand(n,m)-1+(l/m) );
  ```

- generate random n task labels i.i.d, such that $t_i = +1$ with probability $1/2$ and -1 with probability $1/2$

  ```
  t = sign( rand(n,1)-0.5 );
  ```

- generate random m worker reliabilities i.i.d., such that p_j is drawn from the uniform distribution over the interval $[a, b]$ for some $0 < a < b < 1$

  ```
  p = a+(b-a)*rand(m,1);
  ```

  Or, equivalently, in Python:

  ```
  import numpy as np
  E = np.random.choice(2, [n,m], p=[1-l/m, l/m])
  t = np.random.choice([-1,1], n)
  p = np.random.uniform(a, b, m)
  ```

We will fix $a = 0.3$ and $b = 0.95$. For each value of $\ell \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$, we will generate 100 instances of {random graph, task labels, worker reliabilities}, and for each instance of the problem, generate the responses of the workers on those tasks assigned to the workers according to the **Dawid-Skene model**, i.e.

$$A_{ij} \quad = \quad \begin{cases} t_i & \text{with probability } p_j \\ -t_i & \text{with probability } 1 - p_j \end{cases}$$

for all $(i, j) \in E$.

For each instance of the problem, use the proposed algorithm to find the estimates $\{\hat{t}_i\}_{i \in n}$, and compute the error probability:

$$P_e(\ell) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(t_i \neq \hat{t}_i)$$

We will compare it to majority voting error rate:

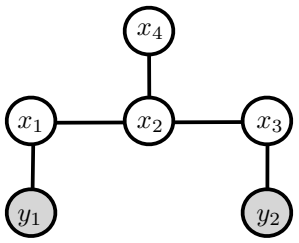
$$P_{\mathrm{MV}}(\ell) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\Big(t_i \neq \mathrm{sign}\big(\sum_{j \in \partial i} A_{ij}\big)\Big)$$

For each value of ℓ plot the $P_e(\ell)$ and $P_{\mathrm{MV}}(\ell)$ averaged over the 100 random instances of the problem, as a function of $\ell \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

6 Gaussian graphical models

6.1 **(Gaussian graphical model and Gaussian BP)**
Let $x \sim \mathcal{N}^{-1}(h_x, J_x)$, and $y = Cx + v$, where $v \sim \mathcal{N}(0, R)$.

1. Find the potential vector $h_{y|x}$ and the information matrix $J_{y|x}$ of $p(y|x)$.
2. Find the potential vector $h_{x,y}$ and the information matrix $J_{x,y}$ of $p(x, y)$.
3. Find the potential vector $h_{x|y}$ and the information matrix $J_{x|y}$ of $p(x|y)$.
4. Consider the following Gaussian graphical model.



Let $y_1 = x_1 + v_1$, $y_2 = x_3 + v_2$, and $R = I$ is the identity matrix. Find C. Represent messages $h_{x_3 \to x_2}$ and $J_{x_3 \to x_2}$ in terms of y_2 and the elements of h_x and J_x. [y_1 and y_2 are measurements, which should be treated as given and deterministically known.]

5. Now assume that we have an additional measurement $y_3 = x_3 + v_3$, where v_3 is a zero-mean Gaussian variable with variance 1 and is independent from all other variables. Find the new C. Represent messages $h_{x_3 \to x_2}$ and $J_{x_3 \to x_2}$ in terms of y_2, y_3 and the elements of h_x and J_x. [again y_2 should be considered as a measurement which is given, and deterministically known.]
6. The BP message from x_3 to x_2 define a Gaussian distribution with mean $m_{x_3 \to x_2} = J^{-1}_{x_3 \to x_2} h_{x_3 \to x_2}$ and variance $\sigma_{x_3 \to x_2} = J^{-1}_{x_3 \to x_2}$. Comment on the difference in the mean and the variance of this message when computed using a single observation y_2 versus when computed using multiple observations (y_2, y_3). Can you guess the mean and variance of the BP message when the number of observations grows to infinity?

6.2 **(Gaussian bp)**
As mentioned in class, Gaussian BP allows to compute the minimum of a quadratic function

$$\widehat{x} = \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \langle x, Qx \rangle + \langle b, x \rangle \right\}. \tag{4}$$

for $Q \in \mathbb{R}^{n \times n}$ positive definite, where $\langle a, b \rangle = a^T b$ indicates the standard inner product of two vectors. In this homework we will consider a case in which Q is not positive definite, but is symmetric and has full rank. in this case we can still define

$$\widehat{x} = -Q^{-1} b\,. \tag{5}$$

which is a stationary point (a saddle point) of the above quadratic function. The BP update equations are exactly the same as for the minimization problem with a positive definite Q. We claim that, when BP converges, it still computes the correct solution $\widehat{x}$.

We consider a specific model. An unknown signal $s_0 \in \mathbb{R}^n$ is observed in Gaussian noise

$$y = As_0 + w_0 \,. \tag{6}$$

Here $y \in \mathbb{R}^m$ is a vector of observations, $A \in \mathbb{R}^{m \times n}$ is a measurement matrix, and $w_0 \in \mathbb{R}^m$ is a vector of Gaussian noise, with i.i.d. entries $w_{0,i} \sim \mathcal{N}(0, \sigma^2)$. We are given y and A, and would like to reconstruct the unknown vector s_0, and hence w_0

A popular method consists in solving the following quadratic programming problem (known as *ridge regression*):

$$\hat{t} = \arg\min_{s \in \mathbb{R}^n} \left\{ \frac{1}{2} \|y - As\|_2^2 + \frac{1}{2} \lambda \|s\|_2^2 \right\}. \tag{7}$$

We will do something equivalent. For $x \in \mathbb{R}^{m+n}$, $x = (z, s)$, $z \in \mathbb{R}^m$, $s \in \mathbb{R}^n$, we define a cost function

$$\mathcal{C}_{A,y}(x = (z, s)) \quad = \quad -\frac{1}{2} \|z\|_2^2 + \frac{1}{2} \lambda \|s\|_2^2 + \langle z, y - As \rangle \,. \tag{8}$$

We will look for the stationary point of $\mathcal{C}_{A,y}$.

(*a*) Show that the cost function $\mathcal{C}_{A,y}(x)$ can be written in the form

$$\mathcal{C}_{A,y}(x) = \frac{1}{2} \langle x, Qx \rangle + \langle b, x \rangle \,. \tag{9}$$

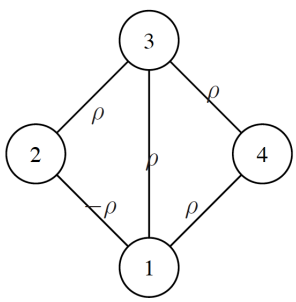
Write explicitly the form of the matrix $Q \in \mathbb{R}^{(m+n) \times (m+n)}$ and the vector $b \in \mathbb{R}^{m+n}$.

(*b*) Let $\widehat{x} = (\widehat{z}, \hat{t})$ be the stationary point of $\mathcal{C}_{A,y}(z, s)$. Assuming it is unique, show that $\hat{t}$ does coincide with the ridge estimator (7).

(*c*) Write the update rule for the BP algorithm (equivalent to the sum-product algorithm) to compute the stationary point $\widehat{x} = (\widehat{z}, \hat{t})$ of $\mathcal{C}_{A,y}(x)$. [hint: use the same ideas from the Gaussian belief propagation for positive definite Q.]

(*d*) Prove the above claim that, if BP converges, then it computes $\widehat{x}$, cf. Eq. (5) even if Q is not positive definite.

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



(a) Confirm (e.g., by checking Sylvester's criterion to see if the determinants of all principal minors are positive) that J is a valid information matrix–i.e., it is positive definite–if $\rho = .39$ or $\rho = .4$. Compute the variances for each of the components (i.e., the diagonal elements of $\Lambda = J^{-1}$)–you can use any software 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 $\rho = .39$ and $\rho = .4$. Describe the behavior that you observe.

(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 two different values of ρ ($\rho \in [0.3, 0.5]$). What do you observe that would explain the result in part (b)?

6.4 **(Gaussian elimination)**

Consider a random vector x made up of two subvecters x_1 and x_2, i.e. $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and is jointly Gaussian vector with potential vector and information matrix denoted by

$$\mathcal{N}^{-1}(h, J) = \mathcal{N}^{-1}\Big(\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \Big) \tag{10}$$

If J_{12} is all zeros, then inverting J is simple and so is finding the marginal distribution of x_1. However, when J_{12} is not all zeros matrix, there is work to be done. In this problem, we will prove that $x_1 \sim \mathcal{N}^{-1}(h_a, J_a)$ with

$$h_a = h_1 - J_{12}J_{22}^{-1}h_2 \ , \quad J_a = J_{11} - J_{12}J_{22}^{-1}J_{21} \ , \tag{11}$$

using *Gaussian elimination*.

(a) We will perform an invertible linear transformation to x such that the components of x_1 remain unchanged but the resulting information matrix is sparse. Consider a linear transformation of the form

$$\begin{bmatrix} x_1 \\ z \end{bmatrix} = \begin{bmatrix} I & 0 \\ A & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ Ax_1 + x_2 \end{bmatrix}$$

for some choice of A. Ideally, we would like to choose A such that x_1 and z are independent, so that the resulting information matrix is sparse. Compute the information matrix of $\begin{bmatrix} x_1 \\ z \end{bmatrix}$ and prove that with the choice of A is $J_{22}^{-1}J_{21}$, x_1 and z are independent.

(b) Use the solution from part (a) to show that the marginal distribution of x_1 is $\mathcal{N}^{-1}(h_a, J_a)$ with h_a and J_a given in (11).

(c) The mean m of x is related to the information form by $Jm = h$. This defines a series of equations, which can be written as two vector equations

$$\begin{aligned} J_{11}m_1 + J_{12}m_2 &= h_1 \ , \text{ and} \\ J_{21}m_1 + J_{22}m_2 &= h_2 \ . \end{aligned}$$

Eliminate m_2 in the above set of equations and show that what is left is precisely the equation of the form

$$J_a m_1 = h_a$$

7 Restricted Boltzmann Machines

7.1 **(Restricted Boltzmann Machines)**
Restricted Boltzmann Machines (RBMs) are a class of Markov networks that have been used in several applications, including image feature extraction, collaborative filtering, and in deep belief networks. An RBM is a bipartite Markov network consisting of a visible (observed) layer and a hidden layer, where each node is a binary random variable. One way to look at an RBM is that it models latent factors that can be learned from input features. For example, suppose we have samples of binary user ratings (like vs. dislike) on 5 movies: Finding Nemo (V_1), Avatar (V_2), Star Trek (V_3), Aladdin (V_4), and Frozen (V_5). We can construct the following RBM:

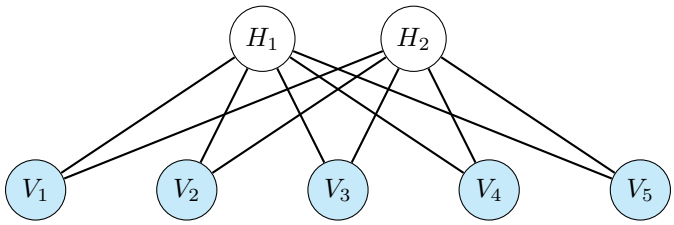


Figure 1: An example RBM with 5 visible units and 2 hidden units.

Here, the bottom layer consists of visible nodes $V_1, \ldots, V_5$ that are random variables representing the binary ratings for the 5 movies, and H_1, H_2 are two hidden units that represent latent factors to be learned during training (e.g., H_1 might be associated with Disney movies, and H_2 could represent the adventure genre). If we are using an RBM for image feature extraction, the visible layer could instead denote binary values associated with each pixel, and the hidden layer would represent the latent features. However, for this problem we will stick with the movie example. In the following questions, let $V = (V_1, \ldots, V_5)$ be a vector of ratings (e.g. the observation $v = (1, 0, 0, 0, 1)$ implies that a user likes only Finding Nemo and Frozen). Similarly, let $H = (H_1, H_2)$ be a vector of latent factors. Note that all the random variables are binary and take on states in $\{0, 1\}$. The joint distribution of a configuration is given by

$$P(V = v, H = h) = \frac{1}{Z} e^{-E(v,h)}, \tag{12}$$

where

$$E(v, h) = -\sum_{ij} w_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j$$

is the energy function, $\{w_{ij}\}$, $\{a_i\}$, $\{b_j\}$ are model parameters, and

$$Z = Z(\{w_{ij}\}, \{a_i\}, \{b_j\}) = \sum_{v,h} e^{-E(v,h)}$$

is the partition function, where the summation runs over all joint assignments to V and H.

(*a*) Using Equation (12), show that $p(H|V)$, the distribution of the hidden units conditioned on all the visible units can be factorized as

$$p(H|V) = \prod_j p(H_j|V),$$

where

$$p(H_j = 1|V = v) = \sigma\left(b_j + \sum_i w_{ij} v_i\right)$$

and $\sigma(s) = \frac{e^s}{1+e^s}$ is the sigmoid function. Note that $p(H_j = 0|V = v) = 1 - p(H_j = 1|V = v)$.

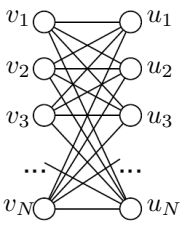
(*b*) Give the factorized form of $p(V|H)$, the distribution of the visible units conditioned on all the hidden units. This should be similar with what's given in part 1, and so you may omit the derivation.

(*c*) Can the marginal distribution over hidden units $p(H)$ be factorized? If so, give the factorization. If not, give the form of $p(H)$ and briefly justify.

(*d*) Based on your answers so far, does the distribution in Equation (12) respect the conditional independences of Figure 1? Explain why or why not. Are there any independences in Figure 1 that are not captured in Equation (12)?

8 Markov Chain Monte Carlo methods

8.1 **(Cheeger's inequality)**
In this problem, we use the Cheeger's inequality from class to upper bound the mixing time of a Markov chain by lower bounding the conductance of the Markov chain. Consider a distribution over matchings in a graph. A *matching* in a graph $G = (V, E)$ is a subsets of edges such that no two edges share a vertex. Here we focus on the special case of a complete bipartite graph G with vertices $v_1, \ldots, v_N$ on the left and $u_1, \ldots, u_N$ on the right, as shown:



In such a graph, a *perfect matching* is a matching which includes N edges. We are interested in sampling from a distribution over perfect matchings. We can denote a perfect matching using the variables $\sigma = [\sigma_{ij}] \in \{0,1\}^{N \times N}$, where $\sigma_{ij} = 1$ is v_i and u_j are matched and $\sigma_{ij} = 0$ otherwise. Observe that σ is a perfect matching if and only if

$$\sum_{k=1}^{N} \sigma_{ik} = 1 \qquad \text{for all } 1 \le i \le N$$
$$\sum_{k=1}^{N} \sigma_{kj} = 1 \qquad \text{for all } 1 \le j \le N$$

A perfect matching σ can also be thought of as a permutation $\sigma : \{1, \ldots, N\} \to \{1, \ldots, N\}$. For example, if $\sigma_{12} = \sigma_{21} = \sigma_{33} = 1$, this would correspond to the permutation $\sigma(1) = 2, \sigma(2) = 1$, and $\sigma(3) = 3$.

Consider the distribution defined by a set of weights on the edges $w_{ij} \ge 0$ for all i and j such that

$$\begin{aligned} \mu(\sigma) &\propto \exp\Big\{ \sum_{i,j} w_{ij}\sigma_{ij} \Big\} \mathbb{I}(\sigma \text{ is a perfect matching}) \\ &= \exp\Big\{ \sum_{i} w_{i\sigma(i)} \Big\} \mathbb{I}(\sigma \text{ is a perfect matching}) \, . \end{aligned}$$

(*a*) First, in this part, consider the uniform distribution over perfect matchings, i.e., $w_{ij} = 0$ for all i, j. Describe a simple procedure to sample σ from this uniform distribution.

(*b*) Now for the weighted distribution, show that for any perfect matching σ,

$$\mu(\sigma) \ge \frac{1}{N! \exp(Nw^*)} \, ,$$

where $w^* = \max_{i,j} w_{ij}$.

(*c*) Consider the Metropolis-Hastings rule defined by: choose $i, i' \in \{1, \ldots, N\}$ uniformly at random. If $i = i'$, do nothing, otherwise with probability

$$R = \min\big\{ 1 \, , \, \exp(w_{i\sigma(i')} + w_{i'\sigma(i)} - w_{i\sigma(i)} - w_{i'\sigma(i')}) \big\}$$

swap $\sigma(i)$ and $\sigma(i')$, i.e. define a new permutation σ' such that $\sigma'(j) = \sigma(j)$ for $j \neq i, i'$ and $\sigma'(i) = \sigma(i')$ and $\sigma'(i') = \sigma(i)$.
Show that, under this Markov chain, for any valid transition $\sigma \to \sigma'$,

$$\begin{aligned} \mathbb{P}_{\sigma,\sigma'} &= \mathbb{P}(\text{ next state is } \sigma' \mid \text{currect state is } \sigma \text{ }) \\ &\geq \frac{1}{N^2 \exp(2w^*)} . \end{aligned}$$

(d) For the conductance of this Markov chain, argue using (b) and (c) that

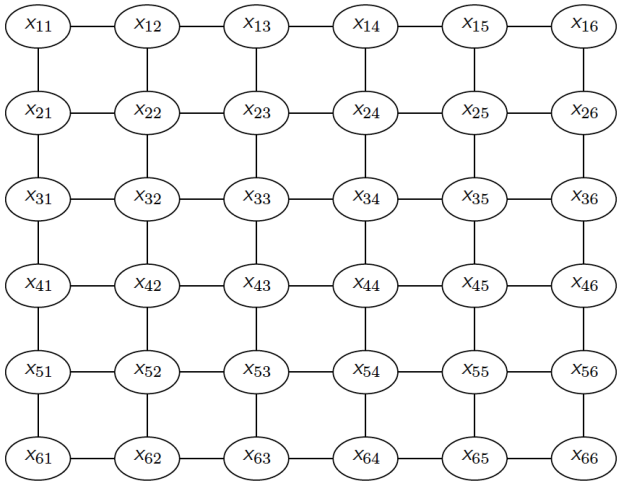
$$\begin{aligned} \Phi &= \min_S \frac{\sum_{\sigma \in S, \sigma' \in S^c} \mu(\sigma) \mathbb{P}_{\sigma,\sigma'}}{\mu(S)\,\mu(S^c)} \\ &\geq \frac{1}{N!\, N^2 \exp((N+2)w^*)} , \end{aligned}$$

where S is a set states (or matchings), S^c is the complement of S, and $\mu(S) = \sum_{\sigma \in S} \mu(\sigma)$.

(e) Using (d), obtain a bound on the mixing time of the Markov chain.

8.2 **(Block Gibbs sampling; implementation)**
In this problem, we develop an efficient algorithm for sampling from a two-dimensional Ising model building on the naive Gibbs sampling. In particular, suppose all variables x_{ij} take values in $\{+1, -1\}$. Using the graph structure G shown below, define the distribution

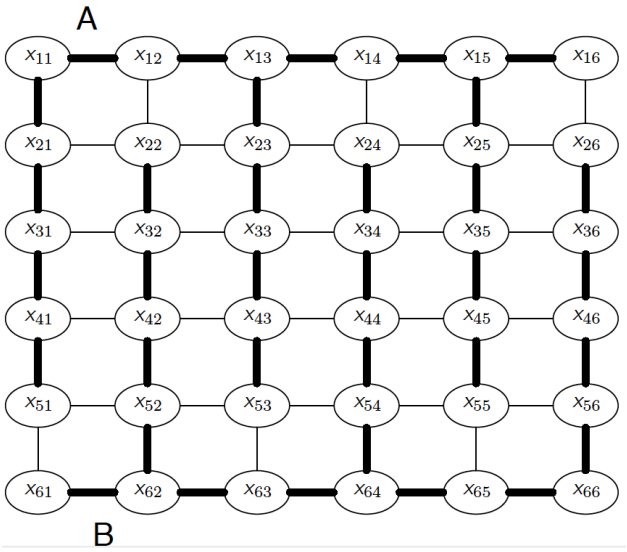
$$\mu_\theta(x) = \frac{1}{Z_\theta} \exp\Big\{ \sum_{(ij,kl) \in E} \theta x_{ij} x_{kl} \Big\} .$$



(a) Derive the update rules for a node-by-node Gibbs sampler for this model. Implement the sampler in Matlab and run it for 3,600,000 iterations on an Ising model of size 60×60 with coupling parameter $\theta = 0.45$. Use uniformly random initialization of $x_{ij} = +1$ with probability 0.5 and $x_{ij} = -1$ otherwise. Show one instance of the state of the variables after every 360,000 iterations. For a 60×60 matrix $x \in \{-1, +1\}^{60 \times 60}$, you can use MATLAB commands `imagesc(x);colormap gray;axis off;` to display the state x.

(*b*) Suppose we are given a tree-structured undirected graphical model T with variables $y = (y_1, \ldots, y_N)$. Give an efficient procedure for sampling from the joint $\mu(y)$.

(*c*) In *block Gibbs sampling*, we partition a graph into r subsets $A_1, \ldots, A_r$. In each iteration, for each A_i, we sample x_{A_i} from the conditional distribution $\mu(x_{A_i}|x_{V\setminus A_i})$. For the Ising model G described above, consider the two comb-shaped subsets A and B shown below. Describe how to use your sampler from part (b) to perform the block Gibbs updates. (For this part, you may assume a black-box implementation of your sampling procedure from part (b).) .



(*d*) We provide an implementation of the block Gibbs sampler from part (*c*) in `comb_gibbs_step.m`, `comb_sum_product.m`, `ising_gibbs_comb.m`. As in part (*a*), we set $\theta = 0.45$ and run the sampler for 1000 iterations updating A and then B at every iteration. Run the block Gibbs sampler in `ising_gibbs_comb.m` and analyze the state of the variables after every 100 iterations. Which of the two samplers appears to mix faster?

You can visually check how much of the initialization is still correlated to current state for different iterations. Qualitatively assess how long it takes for the Markov chain to forget the initial state. This can be used as a proxy for mixing time.

9 Variational methods

9.1 **(Free energy)**
In this problem, we are going to compute free energies of simple graphical models and use BP-like fixed point equations to find the stationary points. We shall consider $G_\ell = (V_\ell, E_\ell)$, an $\ell \times \ell$ two-dimensional torus. This has vertex set $V_\ell = [\ell] \times [\ell]$ and, for any two vertices $i, j \in V_\ell$, $i = (i_1, i_2)$, $j = (j_1, j_2)$, $i_1, i_2, j_1, j_2 \in [\ell]$, we let $(i, j) \in E_\ell$ if and only if either $i_1 = j_1$ and $(i_2 - j_2) \in \{+1, -1\}$ *modulo* ℓ, or $i_2 = j_2$ and $(i_1 - j_1) \in \{+1, -1\}$ *modulo* ℓ.

We consider the *homogeneous* Ising model over $x \in \{+1, -1\}^{V_\ell}$

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

where $\theta_e, \theta_{\rm v}$ are parameters.
[It is rare to encounter such a symmetric model in applications. On the other hand, such toy examples are very useful for developing intuition.]

In the following, fix $\ell = 10$, $\theta_{\rm v} = 0.05$.

(a) Consider the *naive mean field approximation*, and write the naive mean field free energy for

$$\mathbb{F}_{\rm MF}(b) = \mathbb{E}_b[\log \psi_{\rm tot}(x)] - \sum_i \sum_{x_i} b_i(x_i) \log b_i(x_i) \ ,$$

where $b = b_1(\cdot) \times \cdots \times b_n(\cdot)$ and $\psi_{\rm tot}(x) = \prod_{i\in V} \psi_i(x_i) \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j)$.
Assume then the further restriction $b_i(x_i) = b_{\rm v}(x_i)$ for all $i \in V_\ell$ (i.e. the belief is independent of the vertex). Write an expression $\mathbb{F}_{\rm MF}(b_{\rm v})$ as a function of $b_{\rm v} \in \mathbb{R}^2$. This is the objective function to be maximized. Plot the free energy $\mathbb{F}_{\rm MF}(b_{\rm v})$ as a function of a scalar variable $a = (b_{\rm v}(+1) - b_{\rm v}(-1)) \in \mathbb{R}$ for $\theta_e \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. This is equivalent to setting $b_{\rm v}(+1) = (1 + a)/2$ and $b_{\rm v}(-1) = (1 - a)/2$.
Maximize $\mathbb{F}_{\rm MF}(b_{\rm v})$ with respect to $b_{\rm v}$ and plot the optimal value $b^*_{\rm v}(+1)$ and $\mathbb{F}_{\rm MF}(b^*_{\rm v})$ as a function of θ_e.

(b) Repeat the same exercise for the *Bethe free energy*: Write explicitly the Bethe free energy

$$\begin{aligned}\mathbb{F}(b) \quad = \quad & \sum_{(i,j)\in E} \mathbb{E}_{b_{ij}}[\log \psi_{ij}(x_i, x_j)] + \sum_{i\in V} \mathbb{E}_{b_i}[\log \psi_i(x_i)] \\ & - \sum_{(i,j)\in E} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \log b_{ij}(x_i, x_j) - \sum_{i\in V} (1 - \deg(i)) \sum_{x_i} b_i(x_i) \log b_i(x_i) \ .\end{aligned}$$

Assume the further restriction $b_i(x_i) = b_{\rm v}(x_i)$ for all $i \in V_\ell$, $b_{ij}(x_i, x_j) = b_{\rm e}(x_i, x_j)$ (i.e. the belief is independent of the vertex). Write an expression $\mathbb{F}(b_{\rm v}, b_{\rm e})$ as a function of $b_{\rm v}, b_{\rm e}$.
Now, consider $\theta_e = 1.0$, and we want to show that $\mathbb{F}(b_{\rm v}, b_{\rm e})$ has more than one stationary point. The objective function is $\mathbb{F}(b_v, b_e)$, and the constraint is that $\sum_{x_i} b_e(x_i, x_j) = b_v(x_j)$ and $\sum_{x_j} b_e(x_i, x_j) = b_v(x_i)$. The Lagrangian can be written as

$$L(b_v, b_e, \lambda_1, \lambda_2) = \mathbb{F}(b_v, b_e) + \sum_{x_i} \lambda_1(x_i)(\sum_{x_j} b_e(x_i, x_j) - b_v(x_i)) + \sum_{x_j} \lambda_2(x_j)(\sum_{x_i} b_e(x_i, x_j) - b_v(x_j)) \ .$$

The derivative gives

$$\begin{aligned}\frac{\partial L}{\partial b_v(x_i)} &= \frac{\partial \mathbb{F}(b_v, b_e)}{\partial b_v(x_i)} - \lambda_1(x_i) - \lambda_2(x_i) + C \\ \frac{\partial L}{\partial b_e(x_i, x_j)} &= \frac{\partial \mathbb{F}(b_v, b_e)}{\partial b_e(x_i, x_j)} + \lambda_1(x_i) + \lambda_2(x_j) + C' ,\end{aligned}$$

where C and C' are constants (that may differ for each x_i, x_j) that we ignore because we do not care about normalization at this point. Write the explicit derivative of the Lagrangian in terms of ℓ, θ_v, θ_e, $b_v(x_i)$, $b_e(x_i, x_j)$, and Lagrangian multipliers $\lambda_1(x_i)$ and $\lambda_2(x_j)$ which correspond to the constraints $\sum_{x_j} b_e(x_i, x_j) = b_v(x_i)$ and $\sum_{x_i} b_e(x_i, x_j) = b_v(x_j)$.

By symmetry, λ_1 and λ_2 are the same. So we define $\lambda(x_i) = (1/2l^2)\lambda_1(x_i) = (1/2l^2)\lambda_2(x_i)$. Show that $b_v(x_i)$ and $b_e(x_i, x_j)$ at the stationary point satisfy the below equations, by setting the above derivative to zero.

$$\begin{aligned}b_v(x_i) &\propto e^{-(1/3)\theta_v x_i} e^{(4/3)\lambda(x_i)} \\ b_e(x_i, x_j) &\propto e^{\theta_e x_i x_j} e^{(\lambda(x_i)+\lambda(x_j))} ,\end{aligned}$$

By the condition that $\sum_{x_i} b_e(x_i, x_j) = b_v(x_j)$, this gives

$$e^{\theta_e x_i + \lambda(+)} + e^{-\theta_e x_i + \lambda(-)} \propto e^{-(1/3)\theta_v x_i + (1/3)\lambda(x_i)} ,$$

for $x_i \in \{+1, -1\}$. substituting $x_i = +1$ in the above equation, then dividing by the same function evaluated at $x_i = -1$, we get

$$\frac{e^{\theta_e + \lambda(+)} + e^{-\theta_e + \lambda(-)}}{e^{-\theta_e + \lambda(+)} + e^{+\theta_e + \lambda(-)}} = e^{-(2/3)\theta_v + (1/3)(\lambda(+) - \lambda(-))} ,$$

Let $w = (1/2)(\lambda(+) - \lambda(-))$ and change variables to get

$$\frac{e^{\theta_e + w} + e^{-\theta_e - w}}{e^{-\theta_e + w} + e^{+\theta_e - w}} = e^{-(2/3)\theta_v + (2/3)w} ,$$

Using the equality that $\operatorname{atanh}(\tanh(a)\tanh(b)) = (1/2)\log\left(\frac{e^{a+b}+e^{-a-b}}{e^{a-b}+e^{-a+b}}\right)$, show that

$$\tanh(\theta_e)\tanh(w) = \tanh\left(\frac{1}{3}(w - \theta_v)\right) . \tag{13}$$

Plot the left-hand side and the right-hand side of the above equations to finish the proof that there are multiple stationary points of Bethe free energy when $\theta_v = 0.05$ and $\theta_e = 1.0$.

(*c*) We want to maximize $\mathbb{F}(p_1, p_2)$ for each value of $\theta_e \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. Using the above fixed point equations in (13), find all the fixed points of w (numerically and/or approximately). For each fixed point w, find the corresponding value of $b_v(\cdot)$, $b_e(\cdot)$, and $\mathbb{F}(b_v, b_e)$. Plot the optimal (i.e., maximum) value $p_1 = b_v^*(+1)$ and the free energy $\mathbb{F}(p_1^*, p_2^*)$ as a function of θ_e.

9.2 **(Application of minimum cut)**

In this problem, we explore the connections between minimum cut of a graph and pairwise Markov random fields in binary alphabets. Consider a graphical model defined on an undirected graph $G(V, E)$,

$$\mu(x) = \frac{1}{Z}\exp\{-\sum_{i \in V}\phi_i(x_i) - \sum_{(i,j)\in E}\phi_{ij}(x_i, x_j)\} ,$$

for $x = [x_1, \ldots, x_n] \in \{0,1\}^n$. We further assume for now that $\phi_{ij}(0,0) = \phi_{ij}(1,1) = 0$ for all $(i,j) \in E$ (meaning they are zero-diagonal when we consider the functions as 2×2 matrices) such that

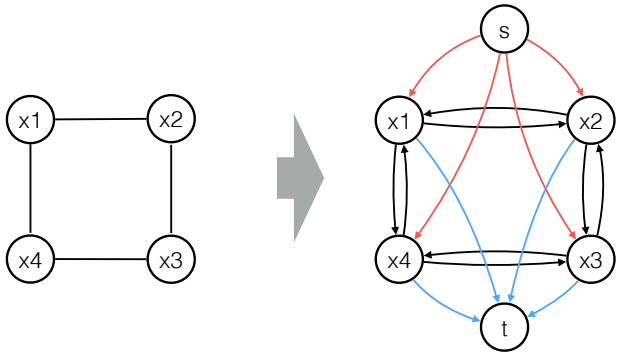
$$\phi_i(\cdot) = \begin{bmatrix} \phi_i(0) \\ \phi_i(1) \end{bmatrix}, \quad \text{and} \quad \phi_{ij}(\cdot,\cdot) = \begin{bmatrix} 0 & \phi_{ij}(0,1) \\ \phi_{ij}(1,0) & 0 \end{bmatrix}.$$

Our goal is to find the maximum likelihood estimate, the one that maximizes the above joint distribution. In order to find the maximizer, we pose this question as a problem of finding the minimum cut of a graph.

Given a pairwise MRF on $G(V,E)$ and the compatibility functions $\phi_{ij}(\cdot,\cdot)$'s, we first create a new **directed** and **weighted** graph as follows.

- Add one node for the source s and one node for the sink t.
- Add an edge from source s to all nodes in V (red edges in the figure below).
- Add an edge from all nodes in V to the sink t (blue edges in the figure below).
- make all edges in E reciprocal (by taking the undirected edge E and making them in to two edges in opposite directions; black edges in the figure below).

An example of a 2×2 grid G, that is transformed is shown below. The colors do not have particular meanings, it is there to help you understand the creation of the new graph. We will find the minimum cut in this transformed graph, after putting appropriate non-negative weights on the edges. A **cut** in a graph is partition of the nodes into two disjoint sets, one containing the source and the other containing the sink. **The value of a cut** is the total weight of the edges that start from a node in the same partition as the source and end in a node in the sink side of the partition, i.e. those that go from the source side of the partition to the other. Note that in the minimum cut, for each node in V, EITHER the edge connecting to the sink will be cut, OR the edge connecting from the source will be cut, but NOT BOTH (since the source and the sink are constrained to be on different sides of the cut). Once we find the minimum cut in this graph, we will assign ZERO to the sink side of the cut and ONE to the source side. This defines a one-to-one mapping between an assignment of binary values in the MRF and a cut in the transformed graph $H(V \cup \{s,t\}, D)$.

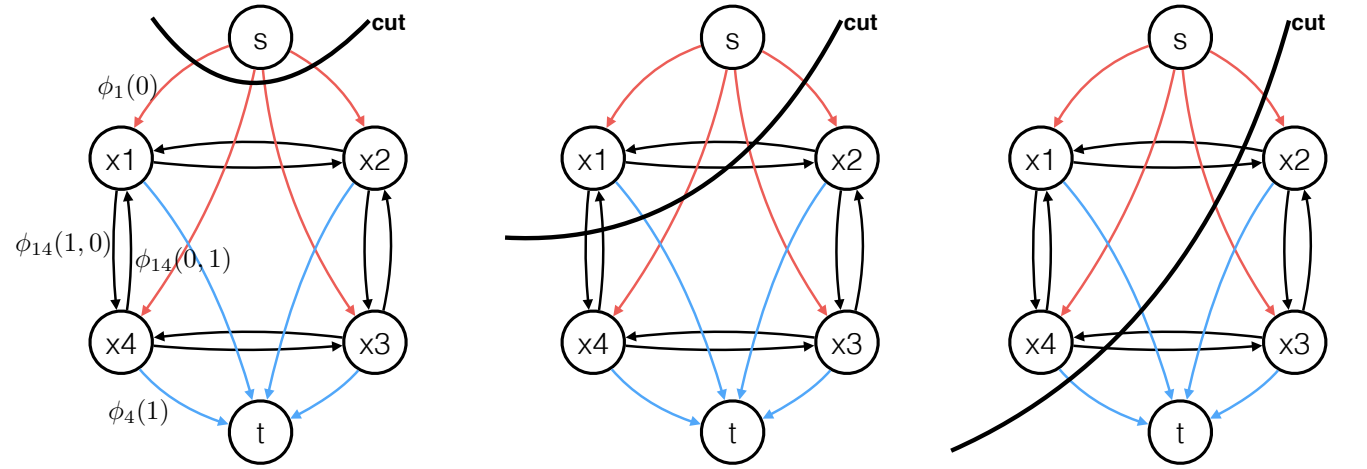


Our goal is to minimize $E(x) \triangleq \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j)$ (which is equivalent as finding the most likely assignment). The following costs on the edges (also called capacities in max-flow min-cut context) ensures that the min-cut of the transformed graph H corresponds to the minimizer of $E(x)$.

- Assign $\phi_i(0)$ to the edge from the source (s,i).
- Assign $\phi_i(1)$ to the edge to the sink (i,t).

– Assign $\phi(1,0)$ to the edge (i,j) and $\phi_{ij}(0,1)$ to the edge (j,i).

An example below shows that this assignment ensures that the value of the cut corresponds to the energy $E(x)$ of the corresponding assignment. In general, cut values are equal to the energy $E(x)$ pf the corresponding assignment x.



assignment: $x = [0,0,0,0]$
cut value: $\phi_1(0) + \phi_2(0) + \phi_3(0) + \phi_4(0)$

$x = [1,0,0,0]$
$\phi_1(1) + \phi_{14}(1,0) + \phi_{12}(1,0)$
$+\phi_2(0) + \phi_3(0) + \phi_4(0)$

$x = [1,0,0,1]$
$\phi_1(1) + \phi_{12}(1,0)$
$+\phi_2(0) + \phi_3(0) + \phi_4(1) + \phi_{34}(01)$

It is known that when the cost on the edges are non-negative, the minimum cut can be found efficiently. Hence, when all $\phi_{ij}(0,0) = \phi_{ij}(1,1) = 0$ and $\phi_i(x_i)$'s, $\phi_{ij}(0,1)$'s and $\phi_{ij}(1,0)$'s are all non-negative, then the costs on the edges are all non-negative and the minimizer of $E(x)$ can be found efficiently by running the off-the-shelf min-cut solvers on H.

(a) Suppose $\phi_1(0) < 0$, and the rest of the compatibility functions are all non-negative, and $\phi_{ij}(0,0) = \phi_{ij}(1,1) = 0$ for all $(i,j) \in E$. Find a new $\phi'_1(x_1)$ such that
   – $\phi'_1(0)$ and $\phi'_1(1)$ are non-negative; and
   – the minimizer of $E'(x) = \phi'_1(x_1) + \sum_{i \in V \setminus \{1\}} \phi_i(x_i) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j)$ is the minimizer of $E(x)$.

   Then, the corresponding transformed graph H with the new costs from $\phi'_1(x_1)$ can be solved for min-cut, since all costs are non-negative.

(b) Now, consider a general case when $\phi_{ij}(0,0)$'s and $\phi_{ij}(1,1)$'s are not necessarily zero. Explain how to assign costs to the directed edges of H (not just for the example given above, but for general $H(V \cup \{s,t\}, D)$ defined from general $G(V,E)$), such that **the value of a cut in this new H is equal to the energy $E(x) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j)$ for the corresponding assignment** x. Note that we do not worry about computational complexity of finding the minimum-cut in this part, and focus in posing the problem as a min-cut problem.
   [hint: consider changing $\phi_i(x_i)$'s and $\phi_{ij}(x_i, x_j)$'s in order to get new $\phi'_{ij}(x_i, x_j)$'s such that the diagonals are zero.]

(c) Suppose $\phi_i(x_i)$'s are all non-negative and $\phi_{ij}(x_i, x_j)$'s are also all non-negative. Assigning costs to the edges of H as per the solution of part (b), it is possible that some edges are assigned negative costs. This is problematic, since min-cut cannot be efficiently solved. However, when all pairwise compatibility functions are **sub-modular**, then the minimizer of $E(x)$ can be found efficiently.

We will prove that this is possible, by constructing a new graph H with non-negative costs under sub-modularity assumption.

A function $f(\cdot)$ over two binary variables is said to be sub-modular if and only if

$$f(0,0) + f(1,1) \quad \leq \quad f(0,1) + f(1,0) \; .$$

Suppose $\phi_i(x_i)$'s are non-negative and $\phi_{ij}(x_i, x_j)$'s are non-negative and sub-modular. Explain how to assign costs to the directed edges of H (not just for the example given above, but for general $H(V \cup \{s,t\}, D)$ defined from general $G(V,E)$), such that

- the value of a cut in this new H is equal to the energy $E(x) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j)$ for the corresponding assignment x; and
- all costs are non-negative.

[hint: consider changing $\phi_i(x_i)$'s and $\phi_{ij}(x_i, x_j)$'s in order to get new $\phi'_{ij}(x_i, x_j)$'s such that the diagonals are zero and the off-diagonals are non-negative.]

10 Learning graphical models