CS221 Midterm

CS221
November 19, 2013

Name: $\underbrace{\hspace{8cm}}_{\text{by writing my name I agree to abide by the honor code}}$

SUNet ID: ______________________________

Read all of the following information before starting the exam:

- This test has 3 problems and is worth 150 points total. It is your responsibility to make sure that you have all of the pages.
- Keep your answers precise and concise. Show all work, clearly and in order, or else points will be deducted, even if your final answer is correct.
- Don't spend too much time on one problem. Read through all the problems carefully and do the easy ones first. Try to understand the problems intuitively; it really helps to draw a picture.
- Good luck!

1. Life on a line (*50 points*)

In this problem, we will look at classification and clustering in one dimension. You might find it helpful to draw the points.

a. (*10 points*)

Recall that in binary classification, the goal is to learn a weight vector $\mathbf{w} \in \mathbb{R}^d$ in order to predict the output label $y \in \{-1, +1\}$ given input x (which is mapped to a feature vector $\phi(x) \in \mathbb{R}^d$).

Define the slightly modified hinge loss:

$$\text{Loss}(x, y, \mathbf{w}) = \max\{2 - \mathbf{w} \cdot \phi(x)y, 0\}. \tag{1}$$

Consider the following training set of (x, y) pairs:

$$\mathcal{D}_{\text{train}} = \{(-4, +1), (1, -1), (0, +1)\}. \tag{2}$$

Suppose the features are

$$\phi(x) = [1, x]. \tag{3}$$

Recall that the stochastic gradient algorithm starts with $\mathbf{w} = [0, 0]$ and loops through each example (x, y) and performs an update:

$$\mathbf{w} \leftarrow \mathbf{w} - \nabla_{\mathbf{w}} \text{Loss}(x, y, \mathbf{w}). \tag{4}$$

Compute the weight vector $\mathbf{w}$ after updating on example 1, example 2, and example 3 (fill out the table below):

	x	$\phi(x)$	y	weights $\mathbf{w}$
Initialization	n/a	n/a	n/a	$[0, 0]$
After example 1	-4		$+1$	
After example 2	1		-1	
After example 3	0		$+1$	

For what values of x does the resulting classifier output a $+1$ label (those x such that $\mathbf{w} \cdot \phi(x) > 0$)?

____ (5)

Solution

- Initial weights: $\mathbf{w} = [0, 0]$
- After example 1, $\phi(x) = [1, -4], y = +1$ (margin 0): $\mathbf{w} = [1, -4]$ (update)
- After example 2, $\phi(x) = [1, 1], y = -1$ (margin -3): $\mathbf{w} = [1, -4]$ (no update)
- After example 3, $\phi(x) = [1, 0], y = +1$ (margin 1): $\mathbf{w} = [2, -4]$ (update)

The resulting classifier will predict $+1$ for x such that $2 - 4x > 0$, which is $x < \frac{1}{2}$.

b. (*10 points*) Consider the following loss function:

$$\text{Loss}(x, y, \mathbf{w}) = \frac{1}{2} \max\{2 - \mathbf{w} \cdot \phi(x) y, 0\}^2. \tag{6}$$

Compute its gradient $\nabla_{\mathbf{w}} \text{Loss}(\phi(x), y, \mathbf{w})$. In one sentence, compare how this loss function differs from the hinge loss above in its treatment of misclassified examples.

Solution Using the chain rule, we get that the gradient is:

$$\nabla_{\mathbf{w}} \text{Loss}(x, y, \mathbf{w}) = \begin{cases} -(2 - \mathbf{w} \cdot \phi(x) y) \phi(x) y & \text{if } 2 - \mathbf{w} \cdot \phi(x) y \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This loss function would penalize misclassifications a lot more (quadratically rather than linearly).

c. (*10 points*)

Consider the following two training datasets of (x, y) pairs:

- $\mathcal{D}_1 = \{(-1, +1), (0, -1), (1, +1)\}$.
- $\mathcal{D}_2 = \{(-1, -1), (0, +1), (1, -1)\}$.

Observe that neither dataset is linearly separable if we use $\phi(x) = x$, so let's fix that.

Define a two-dimensional feature function $\phi(x)$ such that:

- There exists a weight vector $\mathbf{w}_1$ that classifies $\mathcal{D}_1$ perfectly (meaning that $\mathbf{w}_1 \cdot \phi(x) > 0$ if x is labeled $+1$ and $\mathbf{w}_1 \cdot \phi(x) < 0$ if x is labeled -1); and
- There exists a weight vector $\mathbf{w}_2$ that classifies $\mathcal{D}_2$ perfectly.

Note that the weight vectors can be different for the two datasets, but the features $\phi(x)$ must be the same.

Solution One option is $\phi(x) = [1, x^2]$, and using $\mathbf{w}_1 = [-1, 2]$ and $\mathbf{w}_2 = [1, -2]$. There are many options that work, so long as -1 and 1 are separated from 0.

d. (*10 points*)

Consider doing ordinary K-means clustering with $K = 2$ clusters on the following set of 3 one-dimensional points:

$$\{-2, 0, 10\}. \tag{8}$$

Recall that K-means can get stuck in local optima. Describe the precise conditions on the initialization $\mu_1 \in \mathbb{R}$ and $\mu_2 \in \mathbb{R}$ such that running K-means will yield the global optimum of the objective function. Notes:

- Assume that $\mu_1 < \mu_2$.
- Assume that if in step 1 of K-means, no points are assigned to some cluster j, then in step 2, that centroid μ_j is set to ∞.
- Hint: try running K-means from various initializations μ_1, μ_2 to get some intuition; for example, if we initialize $\mu_1 = 1$ and $\mu_2 = 9$, then we converge to $\mu_1 = -1$ and $\mu_2 = 10$.

Solution The objective is minimized for $\mu_1 = -1$ and $\mu_2 = 10$. First, note that if all three points end up in one cluster, K-means definitely fails to recover the global optimum. Therefore, -2 must be assigned to the first cluster, and 10 must be assigned to the second cluster. 0 can be assigned to either: If 0 is assigned to cluster 1, then we're done. If it is assigned to cluster 2, then we have $\mu_1 = -2, \mu_2 = 5$; in the next iteration, 0 will be assigned to cluster 1 since its closer. Therefore, the condition on the initialization written formally is $|-2-\mu_1| < |-2-\mu_2|$ and $|10-\mu_1| > |10-\mu_2|$.

e. (*10 points*)

Suppose we have the following labeled dataset of (x, y) pairs:

$$\{(-11,+1),(-10,+1),(-9,+1),(-1,-1),(0,-1),(1,-1),(9,+1),(10,+1),(11,+1)\} \quad (9)$$

Suppose you fully optimize the K-means objective on the inputs x (finding the global optimum) with $K = 3$, producing centroids $\mu_1, \mu_2, \mu_3 \in \mathbb{R}$. Define the following three features, each indicating whether a point x is closest to a particular centroid: $\phi(x) = [\phi_1(x), \phi_2(x), \phi_3(x)]$, where

$$\phi_j(x) = [|\mu_j - x| < |\mu_{j'} - x| \text{ for all } j \neq j']. \quad (10)$$

Using this feature function, run the Perceptron algorithm on the dataset. What is the weight vector you get and what is the training error? What would happen if you had just used the feature function $\phi(x) = x$ instead? Compare the two approaches in two sentences.

Solution Running K-means yields three clusters at $\mu_1 = -10$, $\mu_2 = 0$, and $\mu_3 = 10$. This maps the first three points into $[1, 0, 0]$, the second three points into $[0, 1, 0]$, and the last three points into $[0, 0, 1]$. Running Perceptron then yields $\mathbf{w} = [1, -1, 1]$, which classifies the training set perfectly. On the other hand, with $\phi(x) = x$, the data are not separable, so the Perceptron algorithm does not converge. This example shows how K-means can be used to derive non-linear feature mappings.

2. Star-crossed Lovers (*50 points*)

In 16th century England, there were a set of $N+1$ cities $C = \{0, 1, 2, \dots, N\}$. Connecting these cities were a set of bidirectional roads R: $(i, j) \in R$ means that there is a road between city i and city j. Assume there is at most one road between any pair of cities, and that all the cities are connected. If a road exists between i and j, then it takes $T(i, j)$ hours to go from i to j.

Romeo lives in city 0 and wants to travel along the roads to meet Juliet, who lives in city N. They want to meet.

a. (*10 points*)

It is decided that Juliet will stay at city N and Romeo will travel from city 0 to city N to meet his love. He can hardly wait to see her, so help him find the fastest way to get there! Let's formulate Romeo's task as a search problem. Let the *state* $s \in C$ be the city that Romeo is currently in. Complete the rest of the search problem by specifying the actions Actions(s), action costs Cost(s, a) for taking action a, successor state Succ(s, a), start state s_{start}, and goal test IsGoal(s). Use the notation that we established above: C, R, T.

- Actions(s) = ____

- Cost(s, a) = ____

- Succ(s, a) = ____

- s_{start} = ____

- IsGoal(s) = ____

Solution

- Each state s is the current city that Romeo is currently in.
- Actions(s) $= \{t : (s, t) \in R\}$ is the set of cities which are connected to city s.
- Cost(s, t) $= T_{sa}$: the time it takes to go from city s to city t.
- Succ(s, t) $= t$: end up in city t.
- $s_{\text{start}} = 0$.
- IsGoal(s) $= \mathbb{I}[s = N]$ is whether Romeo has reached Juliet in city N.

b. (*10 points*)

Fast-forward 400 years and now our star-crossed lovers now have iPhones to coordinate their actions. To reduce the commute time, they will both travel at the same time, Romeo from city 0 and Juliet from city N.

To reduce confusion, they will reconnect after each traveling a road. For example, if Romeo travels from city 3 to city 5 in 10 hours at the same time that Juliet travels from city 9 to city 7 in 8 hours, then Juliet will wait 2 hours. Once they reconnect, they will both traverse the next road (neither is allowed to remain in the same city). Furthermore, they must meet in the end in a city, not in the middle of a road. Assume it is always possible for them to meet in a city.

Help them find the best plan for meeting in the least amount of time by formulating the task as a (single-agent) search problem. Fill out the rest of the specification:

- Each state is a pair $s = (r, j)$ where $r \in C$ and $j \in C$ are the cities Romeo and Juliet are currently in, respectively.

- Actions$((r, j))$ = ____

- Cost$((r, j), a)$ = ____

- Succ$((r, j), a)$ = ____

- $s_{\text{start}} = (0, N)$

- IsGoal$((r, j)) = \mathbb{I}[r = j]$ (whether the two are in the same city).

Solution

- Each state $s = (r, j)$ is the pair of cities that Romeo and Juliet are currently in, respectively.
- Actions$((r, j)) = \{(r', j') : (r, r') \in R, (j, j') \in R\}$ corresponds to both traveling to a connected city
- Cost$((r, j), (r', j')) = \max(T(r, r'), T(j, j'))$ is the maximum over the two times.
- Succ$((r, j), (r', j')) = (r', j')$: just go to the desired city

c. (*10 points*)

Assume that Romeo and Juliet have done their CS221 homework and used Uniform Cost Search to compute $M(i, k)$, the minimum time it takes one person to travel from city i to city k for all pairs of cities $i, k \in C$.

Recall that an A* heuristic $h(s)$ is consistent if

$$h(s) \leq \mathrm{Cost}(s, a) + h(\mathrm{Succ}(s, a)). \tag{11}$$

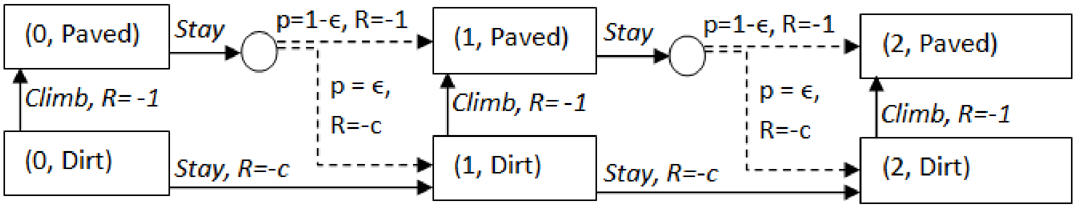
Give a consistent A* heuristic for the search problem in (b). Your heuristic should take $O(N)$ time to compute, assuming that looking up $M(i, k)$ takes $O(1)$ time. In one sentence, explain why it is consistent. Hint: think of constructing a heuristic based on solving a relaxed search problem.

$$h((r, j)) = \underline{\hspace{20em}} \tag{12}$$

Solution Consider the relaxed search problem of giving Romeo and Juliet the option to not wait for each other at every city, but still allowing the waiting at meeting point. Then if Romeo and Juliet are in (r, j), then traveling to some city c in this fashion takes $\max(M(r, c), M(j, c))$. We just need to minimize over all possible cities c:

$$h((r, j)) = \min_{c \in C} \max\{M(r, c), M(j, c)\}. \tag{13}$$

d. (*10 points*)



In the end, things got awfully complicated, so the two decided on something dirt simple: Romeo will simply travel from city 0 to city 1 to city 2 to city 3, etc. until he reaches city N. He sets off on his journey using his navigation app. Unfortunately, his iPhone has an outdated OS, so the app keeps sending him down a dirt road.

We can model the situation using the Markov Decision Process (MDP) in the figure above. There are $2(N+1)$ states (boxes in the figure): (i, Paved) and (i, Dirt) for city $i = 0, 1, \ldots, N$. The start state is $(0, \text{Dirt})$ and the terminal state is (N, Paved). The actions, transitions, and rewards are defined as follows:

- If Romeo is in state (i, Paved), his only possible action is to **stay** on the paved road. With probability $1-\epsilon$, it's smooth sailing and it takes 1 hour (reward -1) to reach state $(i+1, \text{Paved})$. With probability ϵ, the iPhone will act up and send him to $(i+1, \text{Dirt})$, costing him c hours (reward $-c$).

- If Romeo is in state (i, Dirt), he has two possible actions: (i) **stay** on the dirt road, which takes c hours (reward $-c$) to reach state $(i+1, \text{Dirt})$ deterministically, or (ii) **climb** up to the paved road, which takes 1 hour (reward -1) to reach state (i, Paved) deterministically.

Romeo is entertaining the following two policies:

- Policy π_1: Always try to take the paved road. Formally, $\pi_1((i, \text{Dirt})) = \text{Climb}$ and $\pi_1((i, \text{Paved})) = \text{Stay}$.

- Policy π_2: Stay on the dirt road until the end and then climb to the paved road. Formally, $\pi_2((i, \text{Dirt})) = \text{Stay}$ for $i < N$, and $\pi_2((N, \text{Dirt})) = \text{Climb}$. Note that π_2 does not need to be defined for states (i, Paved), since we will never end up there.

Write the **policy evaluation** recurrence for each policy and compute the expected utilities for each policy: $V_{\pi_1}((0, \text{Dirt}))$ and $V_{\pi_2}((0, \text{Dirt}))$?

When should Romeo choose policy 1 over policy 2? Express your answer in terms of an inequality involving only ϵ, c.

Use the next page for your solution.

2d. (write your solution here)

Solution First, let us compute the value of policy 1. Since we will always climb and climbing is a deterministic action, we only need to consider the value of the states (i, Paved). Let A_i denote the value of policy 1 from the paved road in city i. The base case is $A_N = 0$ (at the destination). The recursive case is:

$$\begin{aligned} A_i &= (1-\epsilon)(-1 + A_{i+1}) + \epsilon(-c - 1 + A_{i+1}) && (14) \\ &= -1 - \epsilon c + A_{i+1}. && (15) \end{aligned}$$

Applying the recurrence N times, we have $A_1 = -(1+\epsilon c)N$. Including the initial cost for climbing, we get that the final $A = -1 - (1+\epsilon c)N$.

Now let us compute the value of policy 2. Since policy 2 just involves moving deterministically along the dirt road n time steps and then climbing, we get $B = -1 - cN$.

Policy 1 is better than policy 2 when $A > B$. This happens when $-(1+\epsilon c) > -c$, or more simply $\boxed{\epsilon < \frac{c-1}{c}}$.

e. (*10 points*)

After finally meeting up, Romeo (R) and Juliet (J) decide to try to catch a goose (G) to keep as a pet. Eventually, they chase it into a 3 × 3 hedge maze show below. Now they play the following turn-based game:

1. The Goose moves either Down or Right.
2. Romeo moves either Up or Right.
3. Juliet moves either Left or Down.

G	o	J
	WALL	
R		

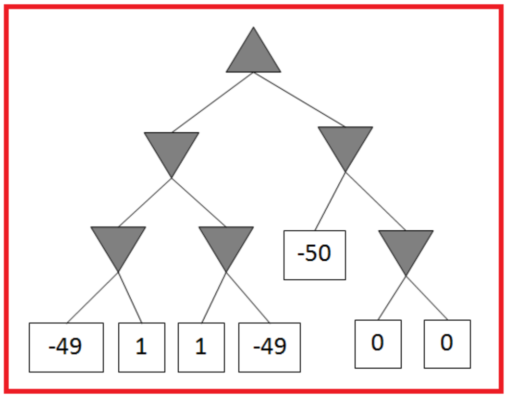
Participants: Goose (G), Romeo (R), Juliet (J), bread (o)

If the Goose enters the square with bread, it gets a reward 1. If either Romeo or Juliet enters the same square as the Goose, they catch it and the Goose gets a reward of −50. The game ends when either the Goose has been caught or everyone has moved once. Note that it is possible for the Goose to get both rewards.

Construct a depth one minimax tree for the above situation, with the Goose as the maximizer and Juliet and Romeo as the minimizers. Use up-triangles Δ for max nodes, down-triangles ∇ for min nodes, and square nodes for the leaves. Label each node with its minimax value.

What is the minimax value of the game if Romeo defects and becomes a maximizer?

Solution Here is the minimax tree:



The value of the game is −49 (the goose might as well go for the bread before it gets caught). If Romeo defects, then the value of the game is 0 (the Goose moves towards Romeo).

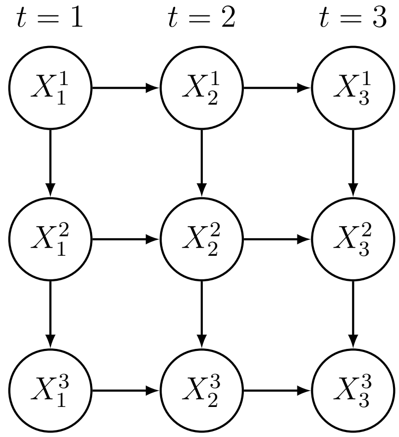
3. Social Bayesian networks (*50 points*)

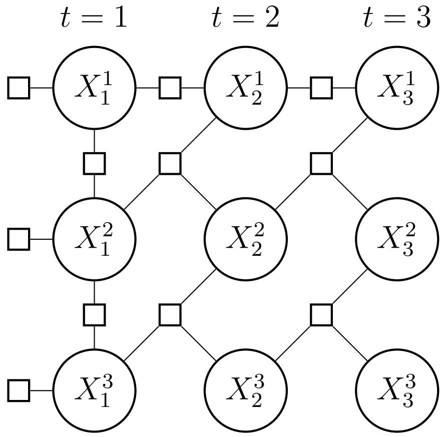
How do people's interests change over time? Suppose we are modeling are n people (numbered $1, \dots, n$) for T weeks (numbered $1, \dots, T$). Let I be a fixed set of interests (e.g., $I = \{\text{(S)cience}, \text{(T)echnology}\}$). Define a random variable $X_t^i \in I$ to represent the interest of person i in week t.

We define the following generative model: In week t, a person i chooses her interest X_t^i as follows:

- With probability α,
  - For $t = 1$ (i.e. the 1st week), she chooses an interest uniformly from I.
  - For $t > 1$, she keeps her interest from the previous week (setting $X_t^i = X_{t-1}^i$).
- With probability $1 - \alpha$,
  - If $i = 1$ (i.e. the 1st person), she chooses an interest uniformly at random from I.
  - If $i > 1$, she switches to person $i - 1$'s interest of that week (set $X_t^i = X_t^{i-1}$).

The diagram on the left is the Bayesian network for $n = 3$ and $T = 3$. Its corresponding factor graph is on the right (recall we have exactly one potential for each random variable).





a. (*10 points*) **Conditional probabilities:**

Consider modeling $n = 3$ people for $T = 3$ weeks with interests $I = \{$(S)cience, (T)echnology$\}$.

Fill out the conditional probability table in terms of the probability α for $p(x^1_{t+1} \mid x^1_t)$ (person 1 in week t) and $p(x^2_{t+1} \mid x^1_{t+1}, x^2_t)$ (person 2 in week $t + 1$).

x^1_t	x^1_{t+1}	$p(x^1_{t+1} \mid x^1_t)$
S	S	
S	T	
T	S	
T	T	

x^1_{t+1}	x^2_t	x^2_{t+1}	$p(x^2_{t+1} \mid x^1_{t+1}, x^2_t)$
S	S	S	
S	S	T	
S	T	S	
S	T	T	
T	S	S	
T	S	T	
T	T	S	
T	T	T	

Solution Following the definition of α, the following two tables should be obtained.

x^1_t	x^1_{t+1}	$p(x^1_{t+1} \mid x^1_t)$
S	S	$(1 + \alpha)/2$
S	T	$(1 - \alpha)/2$
T	S	$(1 - \alpha)/2$
T	T	$(1 + \alpha)/2$

x^1_{t+1}	x^2_t	x^2_{t+1}	$p(x^2_{t+1} \mid x^1_{t+1}, x^2_t)$
S	S	S	1
S	S	T	0
S	T	S	$1 - \alpha$
S	T	T	α
T	S	S	α
T	S	T	$1 - \alpha$
T	T	S	0
T	T	T	1

b. (*10 points*) **CSP:** Suppose there are three interests

$$I = \{\text{(S)cience, (T)echnology, (P)olitics}\}. \tag{16}$$

We have obtained more information about the interests of each of the $n = 3$ people and $T = 3$ weeks. This reduces the domain of some of the random variables:

t	1			2			3		
Person 1	S	T	P	S			S	T	P
Person 2	S			S	T	P	S	T	P
Person 3	S			S	T	P			P

Run AC-3 to remove as many inconsistent domain values as possible. You might find it useful to look at the factor graph. Cross out the values that are removed, and write one sentence justifying their removal.

Solution First, apply arc consistency down the second week, using the fact that if both the previous person and the previous week have the same interest, then the current person in the current week must also get the same interest. Second, apply arc consistency up the third week, using the fact that if the previous week and the current week are different, then the interest must have come from the previous person (of the current week).

t	1			2			3		
Person 1	S	T	P	S			~~S~~	~~T~~	P
Person 2	S			S	~~T~~	~~P~~	~~S~~	~~T~~	P
Person 3	S			S	~~T~~	~~P~~			P

c. (*10 points*) **Treewidth:** Consider the factor graph for n people and T weeks (see the figure on the first page of this problem for $n = T = 3$). Condition on all variables except those in week 1.

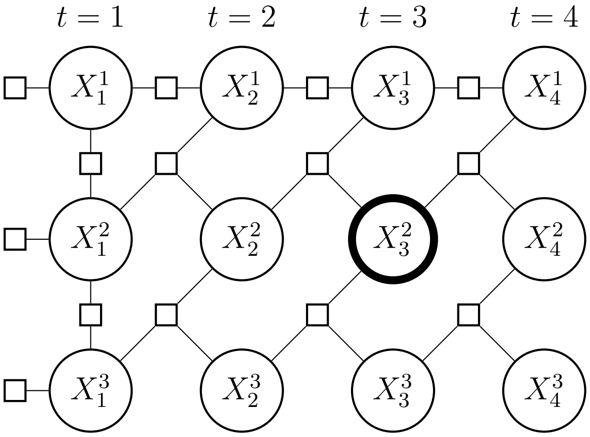
What's the treewidth of the resulting factor graph? In one sentence, justify your answer (e.g., describe your variable elimination strategy that yields this answer).

Partial credit will be awarded for answering the $n = T = 3$ case; full credit will be awarded for arbitrary n and T.

Solution The treewidth is 1 regardless the value of n or T since the resulting factor graph is a chain; eliminating the variables from the leaves up always creates a potential with arity 1.

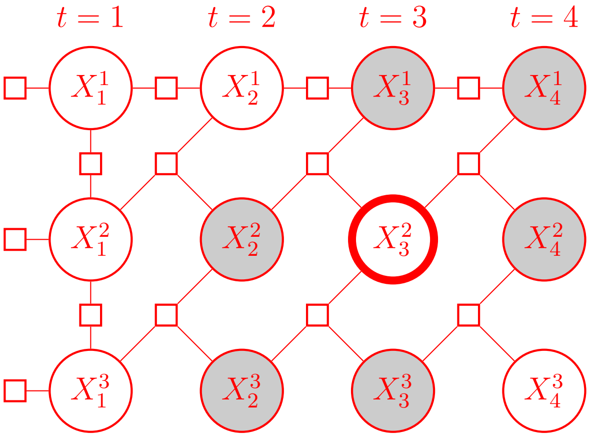
d. (*10 points*) **Gibbs sampling:** Suppose we model $n = 3$ people for $T = 4$ weeks. Recall that performing Gibbs sampling of a variable just requires the values of its Markov blanket.

Shade the Markov blanket of X_3^2 (person 2 in week 3) in the diagram below and write down the Gibbs sampling update for X_3^2 as a function of potentials $p(x_t^i \mid x_t^{i-1}, x_{t-1}^i)$ and $p(x_t^1 \mid x_{t-1}^1)$ (for various values of i and t) in the space provided. Ignore the normalization constant.



$p(x_3^2 \mid$ ____________________________ $) \propto$ ____

Solution The Gibbs update depends on the Markov blanket of X_3^2, which is shown below



Remember to include X_4^1 and X_2^3. The update is therefore just the product of all the factors:

$$p(x_3^2 \mid x_2^2, x_4^2, x_3^1, x_3^3, x_4^1, x_2^3) \propto p(x_3^2 \mid x_3^1, x_2^2) p(x_4^2 \mid x_4^1, x_3^2) p(x_3^3 \mid x_3^2, x_2^3)$$

e. (*10 points*) **Learning:** Suppose we observed the interest of $n = 2$ people over $T = 5$ weeks:

t	1	2	3	4	5
Person 1	S	T	T	S	T
Person 2	S	S	T	T	T

Previously, we had only one parameter α to model the complex dynamics of people, which is clearly insufficient. For this question, let us assume that our parameters are the full conditional probability tables based on our observations for each person.

Fill in the following tables with the parameter estimates using Laplace smoothing with $\lambda = 1$.

x_t^1	x_{t+1}^1	$p(x_{t+1}^1 \mid x_t^1)$
S	S	
S	T	
T	S	
T	T	

x_{t+1}^1	x_t^2	x_{t+1}^2	$p(x_{t+1}^2 \mid x_{t+1}^1, x_t^2)$
S	S	S	
S	S	T	
S	T	S	
S	T	T	
T	S	S	
T	S	T	
T	T	S	
T	T	T	

Solution

x_t^1	x_{t+1}^1	$p(x_{t+1}^1 \mid x_t^1)$
S	S	1/4
S	T	3/4
T	S	1/2
T	T	1/2

x_{t+1}^1	x_t^2	x_{t+1}^2	$p(x_{t+1}^2 \mid x_{t+1}^1, x_t^2)$
S	S	S	1/2
S	S	T	1/2
S	T	S	1/3
S	T	T	2/3
T	S	S	1/2
T	S	T	1/2
T	T	S	1/3
T	T	T	2/3