CS221 Final Exam

CS 221
December 10, 2012

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

SUNet ID (e.g., *psl*): ______________________

Read all of the following information before starting the exam:

- Show all work, clearly and in order, if you want to get full credit. We reserve the right to take off points if we cannot see how you arrived at your answer (even if your final answer is correct).
- Please keep your written answers brief; be clear and to the point. We will take points off for rambling and for incorrect or irrelevant statements.
- This test has 8 problems and is worth 76 points. It is your responsibility to make sure that you have all of the pages!
- Be sure to read all the problems, as some may be more difficult than others, and they are not ordered by difficulty.
- Good luck!

1. **Touring Gendarra** (*12 points*)

You have been employed by a tour company to schedule four-day itineraries for Gendarra, a country that has recently opened up to tourism after decades of isolation. There are three tour groups, and each tour group will visit some number of sites each of the four days. Below is the list of nine sites that each tour group can visit:

City Points of Interest	**Country Points of Interest**
Gendarran Castle East Wing *(ECastle)*	Base of Mt. Arah *(BaseArah)*
Gendarran Castle West Wing *(WCastle)*	Mt. Arah Summit *(SummitArah)*
Gendarran Castle North Wing *(NCastle)*	Village of Gendarra *(Village)*
Gendarran Castle South Wing *(SCastle)*	
Market of Gendarra *(Market)*	
Church of Gendarra *(Church)*	

There are several constraints on the itineraries (the first two are imposed by the government, and the last three are imposed by the tourists):

1. On any given day, each group can visit at most one of the four sites in Gendarran Castle.
2. On any given day, each group can only visit city sites or country sites but not both.
3. Over the course of the four days, $Group1$ must visit at least one site from the city and at least one site from the country.
4. Over the course of the four days, $Group2$ must visit all sites in the castle.
5. Over the course of the four days, $Group3$ must visit at least one site per day.

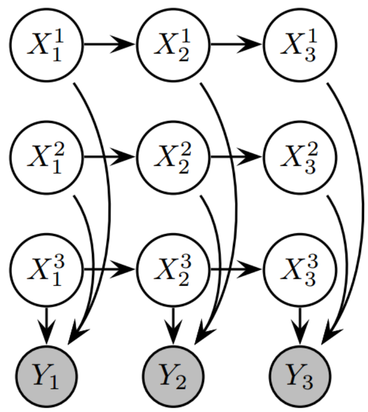
a. (*5 pts*) We will now formulate this problem as a CSP. For each group g, site s, and day d, define a variable $Visit(g, s, d)$. For each of the five constraints above, define the *smallest* number of factors capturing these constraints. How many factors are there, and what is the arity of each factor?

b. (*1 pt*) How many connected components does the factor graph corresponding to the CSP in part (b) have?

c. (*1 pt*) Being low on funds, you want to satisfy the constraints but also minimize the number of total sites visited by the three groups over the four days. Add a factor to the CSP (making it a weighted CSP) so that the maximum weight assignment meets this objective.

2. Factorial HMMs (*9 points*)

A factorial HMM is an example of a Bayesian network, shown below for $T = 3$ time steps:



The idea is that at each time slice, we have a state represented by $k = 3$ components; for example, at time 1, the components are $X_1 = (X_1^1, X_1^2, X_1^3)$. Each component i transitions over time independently according to $p_1(X_t^i \mid X_{t-1}^i)$. The components participate together to generate the corresponding observation according to $p_2(Y_t \mid X_t^1, \ldots, X_t^k)$.

a. (*1 pt*) Assume that $Y_1, \ldots, Y_T$ are unobserved. If we eliminate these variables, what is the tree-width of the resulting factor graph as a function of k and T?

b. (*1 pt*) Assume that $Y_1, \ldots, Y_T$ are observed (we're conditioning on these variables). What is the tree-width of the resulting factor graph as a function of k and T?

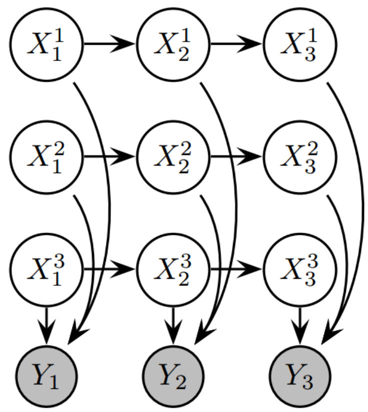
c. (*1 pt*) Suppose we condition on all the variables except for $X_1^1, \ldots, X_T^1$. What is the tree-width of the resulting factor graph as a function of k and T?

d. (*2 pts*) Express the conditional distribution $\mathbb{P}(X_3^1, X_3^2 \mid Y_1, X_2^1, \ldots, X_2^k)$ in terms of p_1 and p_2 (you must simplify the expression if possible).

e. (*2 pts*) If instead of using a single shared transition distribution p_1, we change the model and use a separate transition distribution $p_{1,j}$ for each component $j = 1, \ldots, K$. The parameters of the new model are now $\theta' = (p_{1,1}, \ldots, p_{1,k}, p_2)$. Let L be the likelihood obtained by the maximum likelihood estimate of θ in the old model. and let L' be the likelihood obtained by the maximum likelihood estimate of θ' in the new model. Prove that $L' \geq L$.

3. Modeling productivity (*10 points*)

In Gates, there is an office occupied by three students. For each time step t, let $X_t^i = 1$ if student i wants to do work and 0 if not. At each point, student i will have a probability α of changing his/her desire to do work (1 to 0 or 0 to 1). Let Y_t denote whether or not work gets done at time t. If all three students want to do work ($X_t^i = 1$ for all i), then $Y_t = 1$ with probability β. Otherwise, no work gets done ($Y_t = 0$). In the first time step, all students want to do work ($X_1^i = 1$ for all i).



a. (*2 pts*) What is the probability that work gets done at time step 2 (compute $\mathbb{P}(Y_2) = 1$)?

b. (*1 pt*) Compute $\mathbb{P}(X_2^i = 1 \mid Y_2 = 1)$ for each student i.

c. (*2 pts*) For the rest of the problem, we will assume we observe $Y_1, Y_2, Y_3, \ldots$. In this factor graph, write the new factors created by eliminating $X_1 = (X_1^1, X_1^2, X_1^3)$.

4. **Learning an Evaluation Function** (*7 points*)

In this problem, you will learn an evaluation function for a Chess-playing agent. We assume that the evaluation function is a weighted combination of features $\phi(s) \in \mathbb{R}^d$:

$$Eval_{\mathbf{w}}(s) = \mathbf{w} \cdot \phi(s).$$

As an initial guess, you set all the weights to 1 (denote the resulting evaluation function $Eval_1(s)$). Now you'd like to improve the evaluation function by learning the weights automatically. To get training data, you obtain a set of game states S where it's the agent's turn. For each state $s \in S$, you run minimax from that state to depth d and use $Eval_1(s)$ to evaluate the resulting state. This minimax recursion in the end produces a value $V(s)$ for each state $s \in S$. Next, you set $\mathbf{w}$ to minimize the squared loss:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{s \in S} \frac{1}{2}(Eval_{\mathbf{w}}(s) - V(s))^2.$$

The final agent will perform minimax up to depth d and use the evaluation function $Eval_{\mathbf{w}^*}(s)$.

a. (*1 pt*) Suppose that $\mathbf{w}^*$ achieves zero loss on S. For each $s \in S$, does the final agent perform the same action as an algorithm that just runs minimax up to some depth using the initial evaluation function $Eval_1$? If so, to what depth? If not, provide a brief explanation.

b. (*1 pt*) Suppose that $\mathbf{w}^*$ achieves zero loss on S. For each $s \notin S$, does the final agent perform the same action as an algorithm that just runs minimax up to some depth using the initial evaluation function $Eval_1$? If so, to what depth? If not, provide a brief explanation.

c. (*1 pt*) Suppose your features $\phi(x)$ are as follows: the number of pieces the agent has ($\phi_1(x)$), the number of pieces the opponent has ($\phi_2(x)$), and the number of pieces the agent has minus the number of pieces the opponent has ($\phi_3(x)$). Suppose $\mathbf{w}^* = (3, -2, 4)$. Can you replace the features $\phi(x)$ with two features and set weights appropriately without changing the evaluation function $Eval_{\mathbf{w}^*}(s)$? If so, what are the features and weights?

d. (*2 pts*) Suppose we have $S = \{s_1, s_2\}$ and $V(s_1) = 2$ and $V(s_2) = 6$. Prove that for any feature vector $\phi(x)$, the minimum loss obtained by $\mathbf{w}^*$ is at most 20 (the upper bound on the minimum loss is 20).

e. (*2 pts*) If we have a single feature which is always one ($\phi(x) = (1)$), give a closed-form expression for $\mathbf{w}^*$ as a function of $V(s)$ over $s \in S$.

5. K-Means and EM (*14 points*) You have been given the task of clustering a set of objects as either **large** or **small**. We represent each object x as a vector of two features $\phi(x)$ based on the bounding box of x: $\phi_1(x)$ is the bounding box's width; and $\phi_2(x)$ is the bounding box's height.

a. (*4 pts*) Run K-means on the following set of objects. The dimensions of each object below (width$\times$height) is given.

	Object	**Dimensions**
(a)	Ant	1×1
(b)	Box	50×100
(c)	Cougar	150×130
(d)	Dining Room	300×200
(e)	Elephant	240×300

Your initial cluster centers are:

$$\begin{aligned} \mu_1 &= (0, 0) \\ \mu_2 &= (300, 300) \end{aligned}$$

A skeleton of the K-means algorithm is given below; fill in the 10 blanks for running 3 iterations of the algorithm. Use euclidean distance. Your answer for the E steps should be a list of objects assigned to each cluster (e.g., *(a), (b), (c), (d)*); your answer for the M steps should be a new cluster center (e.g., $(42.0, 7.0)$).

Iteration	**Task**	**Value**
1.E	Assignment $z_i = 1$:	(i) ____________________
1.E	Assignment $z_i = 2$:	(ii) ____________________
1.M	New μ_1:	(iii) ____________________
1.M	New μ_2:	(iv) ____________________
2.E	Assignment $z_i = 1$:	(v) ____________________
2.E	Assignment $z_i = 2$:	(vi) ____________________
2.M	New μ_1:	(vii) ____________________
2.M	New μ_2:	(viii) ____________________
3.E	Assignment $z_i = 1$:	(ix) ____________________
3.E	Assignment $z_i = 2$:	(x) ____________________
3.M	New μ_1:	(xi) ____________________
3.M	New μ_2:	(xii) ____________________

b. (*3 pts*) Recall that K-means clusters points based on reconstruction loss, defined by **sum** of the squared distance:

$$\min_z \min_\mu \sum_{i=1}^n \|\mu_{z_i} - \phi(x_i)\|^2.$$

In this problem, we will construct a weighted CSP corresponding to this K-means objective function. Your variables should be the assignments $z_1, \ldots, z_n$ of the n data points and the cluster centers $\mu_1, \ldots, \mu_K$.

First, clearly define the domain of each variable and the factors (be precise). The resulting weighted CSP should have the property that the maximum weight assignment to the weighted CSP should correspond to the K-means solution. Draw the corresponding factor graph for $K = 2$ clusters and $n = 4$ data points.

c. (*2 pts*) In the E-step, we *condition* on the cluster centers $\mu = (\mu_1, \ldots, \mu_K)$. Draw the resulting factor graph (the nodes in the graph should be $z_1, \ldots, z_4$). What is the tree-width? What does the graph structure reveal about computing the maximum weight assignment efficiently?

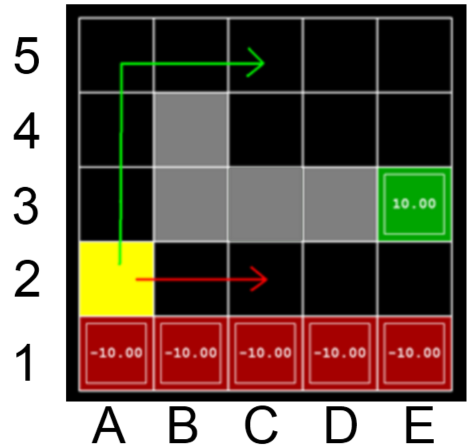
d. (*2 pts*) In the M-step, we are given the assignments z and optimize with respect to the cluster centers μ. This operation is akin to eliminating μ. Recall that this produces a weighted CSP over variables $z_1, \dots, z_n$. Write an expression for the weight of an assignment to $z_1, \dots, z_n$. What is the tree-width? Draw the resulting factor graph (the nodes in the graph should be $z_1, \dots, z_4$).

e. (*3 pts*) Suppose we want certain points to end up in the same cluster. To do this, we are given sets $S_1, \dots, S_m$ which are disjoint subsets of $\{1, \dots, n\}$. Clearly write down the factors needed to enforce that all the points in the same S_i must be assigned to the same cluster.

For $S_1 = \{1, 2\}$ and $S_2 = \{3, 4\}$, draw the corresponding modification of the factor graph in (d).

Finally, write down the formal expression for the new E-step. Hint: you will need to update z_is for all $i \in S_j$ at the same time.

6. Pac-Man's Lair (*11 points*) During a hike with some of your fellow CS221 students, you accidentally stumble upon the cavernous lair of a giant Pac-Man, who is trying to eat you in retribution for the dastardly algorithms you put him through this quarter. The light grey squares are impassable terrain. *Both the cliffs (get reward* -10 *for entering) and the cave exit (get reward* $+10$ *for entering) are terminal states.* This means the game is over when you reach a -10 or +10 state (you die or escape).



You jump in a mining cart to escape, with the actions LEFT, RIGHT, UP, DOWN. Unfortunately the steering system has not been used in a while, and thus it will not always take you in the intended direction. An action will take you (i) in the direction commanded with $1 - 2\tau$ probability, (ii) in one of the two adjacent directions with τ probability each, and (iii) in the opposite direction with probability 0. *If the result of an action would take you into a wall or off the grid, you end up in your same location.* There is a constant "cost of living" C incurred on every action you take.

a. (*1 pt*) Your mining cart will choose its path obtained from the optimal policy from running value iteration. What is the value of square E4 after the first iteration of value iteration with discount $\gamma = 0.8$, noise parameter $\tau = 0.25$, and cost of living $C = -1$?

b. (*1 pt*) What is the value of square E4 after two iterations of value iteration, with the same parameters as above? Note: you can just write down an arithmetic expression without simplifying it.

c. (*3 pts*) This mining cart is no ordinary cart—it does not have a steering wheel! Instead, there are three dials, to control the parameters γ, τ, C, which are used by value iteration to determine the optimal path. The possible paths are the upper path or the lower path in the figure. Suppose the current setting of γ, τ, C yields the lower path by the -10 cliff.

For the following parts, please explain your answer in one or two sentences. Answers without correct justification will not receive credit.

- Would we increase or decrease γ to make value iteration produce the longer path?
- Would we increase or decrease C to make value iteration produce the longer path?
- Would we increase or decrease τ to make value iteration produce the longer path?

Note: each part involves changing only one dial, and assume it is possible to produce the longer path in all three cases.

d. (*3 pts*) You realize that during the execution of a policy, Pac-Man can actually interfere and cause you to choose a random action every other action. In other words, you control the first action, Pac-Man generates the second action uniformly at random, etc. Write a recurrence equation for the optimal value $V(s)$ in this case (recall $\text{Reward}(s, a)$ is the reward function and $T(s, a, s')$ specifies the transition distribution).

e. (*3 pts*) You hear the mining cart's motor begin to wind down because it is running low on fuel. The mining cart can only hold 5 gallons of fuel at a time, but luckily you have a spare tank of 15 gallons of fuel on board. Each of the original four actions (LEFT, RIGHT, UP, DOWN) consume one gallon of fuel and are only available if the main tank is not empty. There is a new action REFUEL, which moves as much fuel from the spare tank to the main tank as possible, either filling up the main empty or emptying the spare tank.

There is also a RESET action. If you try to move or refuel more than two times in row without performing an intervening RESET action, the cart will overheat and you will perish in flames.

Re-define the **states** and **transitions** so that the optimal policy can be still computed by using value iteration under these new constraints. Be precise.