CPSC 422

Practice Final Exam

April 2012

This practice exam is meant to give some example questions. You can expect questions that show your knowledge of the details (e.g., asynchronous value iteration, Q-learning, particle filtering), as well as testing your understanding about the bigger picture. Note that this practice exam emphasizes the last half of the course. The final exam will cover *all* of the course. You should also look at the practice midterm and the actual midterm (and their solutions). There will be similar questions to the midterm (particularly the questions students did not do as well at).

You may bring into the exam a letter-sized sheet of paper with anything written on it. You may not use calculators, phones, smart watches, robotic assistants or other electronic aids.

Some important points (that students often forget):

- Read and answer the question. You will not get marks for writing things (whether they are true or not) that are not relevant to the question.
- Use proper English in full sentences. You will not get marks if we cannot work out what you are saying.
- If a question asks about a particular instance of a problem, make sure your answer refers to that instance. If there are variables whose value can be inferred from the question, use that value. When we ask for a arithmetic expression we mean one without variables (an expression with variables is an algebraic expression). Writing a general formula that you may have copied from the sheet you can bring in, is not worth any marks. (The questions are usually asking to apply that formula to a particular case, to make sure you understand it).

The course covered the following sections of the textbook:

- Agents: Sections 1.3, 1.5, 2.1, 2.2, 2.3

- Probability and time: Section 6.5
- Probabilistic inference: Section 6.4
- Learning: Sections 7.1-7.4, 7.8, 11.1,
- Decision making: Chapter 9 and Sections 10.4, 11.3
- Individuals and relations: Section 13.1-13.3, 14.2, 14.3. [Note that you will not be expected to memorize any of OWL; we will give any constructs you need. We did not cover ICL in 14.3.]

If you can do the questions below, you should have no trouble with the final exam.

1. For a decision-tree learner what is a myopically-optimal split?
   Why do we not always return the smallest decision tree that correctly classifies the training data?

2. Suppose we have a stationary hidden Markov model with 10 states where, at every time, there is a Boolean observation. What probabilities need to be specified to model this HMM? How many numbers need to be specified?

3. For a robot in a complex environment, which of the following methods would be suitable to implement robot localization, the problem of determining the location of a robot given the history of sensor observations? For each unsuitable method, specify why it would not be suitable. For each suitable method, specify what needs to be stored to implement the method, and what each of the stored items represents.
   (a) variable elimination
   (b) rejection sampling
   (c) importance sampling
   (d) particle filtering

4. A travel site has a Prolog database that represents information about hotels and feedback from users that uses the relations:

$$hotel(Id, Name, City, Province, Country, Address)$$

$$reported\_clean(Hotel, RoomNumber, Cleanliness, day(Year, Month, Day))$$

Show how the following facts can be represented using triple notation, using vocabulary that make sense:

```
hotel(h345,"The Beach Hotel",victoria,bc,
       canada,"300 Beach St").
reported_clean(h345,127,clean,day(2012,01,25)).
```

Is it reasonable to represent the hotel name and address as strings? Explain.

5. Consider the following ontology (written in the OWL functional syntax) about hotels and hotel rooms. In this ontology, a hotel has rentable units, each of which is a suite, a standard room or a room.

```
Declaration(Class(:Hotel))
Declaration(Class(:Room))
Declaration(Class(:BathRoom))
SubClassOf(:BathRoom :Room)

Declaration(Class(:RentableHotelUnit))
Declaration(Class(:RoomOnly))
SubClassOf(:RoomOnly :RentableHotelUnit)
SubClassOf(:RoomOnly :Room)
Declaration(Class(:StandardRoom))
EquivalentClasses(:StandardRoom
        ObjectIntersectionOf(
             :RentableHotelUnit
             ObjectSomeValuesFrom(:ContainsRoom :BathRoom)
             ObjectExactCardinality(2 :ContainsRoom :Room)))
Declaration(Class(:Suite))
EquivalentClasses(:Suite
        ObjectIntersectionOf(
             :RentableHotelUnit
             ObjectSomeValuesFrom(:ContainsRoom :BathRoom)
             ObjectMinCardinality(3 :ContainsRoom :Room)))

Declaration(ObjectProperty(:ContainsRoom))
InverseFunctionalObjectProperty(:ContainsRoom)
ObjectPropertyDomain(:ContainsRoom :RentableHotelUnit)
ObjectPropertyRange(:ContainsRoom :Room)

Declaration(ObjectProperty(:HasForRent))
InverseFunctionalObjectProperty(:HasForRent)
ObjectPropertyDomain(:HasForRent :Hotel)
ObjectPropertyRange(:HasForRent :RentableHotelUnit)
```

```
Declaration(Class(:XX))
EquivalentClasses(:XX
        ObjectIntersectionOf(
             ObjectSomeValuesFrom(:HasForRent :Suite)
             ObjectMaxCardinality(0 :HasForRent :RoomOnly)
             :Hotel))
```

(a) Explain in English what a standard room is.

(b) Is a standard room a room? Should it always be? Should it never be? Explain.

(c) Can a standard room contain two bathrooms? How could the ontology be modified to disallow this?

(d) Describe in English the class *XX*.

6. Suppose we have parmetrized random variables $buys(Person, Item)$, $young(Person)$ and $cool(Item)$ where there at 3000 people and 200 items. Suppose $young(Person)$ and $cool(Item)$ are parents of $buys(Person, Item)$.

   (a) Draw this in plate notation.

   (b) How many random variables are in the grounding of this model?

   (c) Draw the grounding belief network assuming the population of *Person* is $\{sam, chris\}$ and the population of *Item* is $\{ipod, mortgage, spinach\}$.

   (d) What could be observed to make $cool(ipod)$ and $cool(mortgage)$ probabilistically dependent on each other given the observations?

7. One of the goals of this course was to introduce you to a wide field of methods in AI and to allow you to think about solving problems using agents. This question asks you to think critically about how to solve a complex problem.

   Election Prediction: Large corporations don't like unpredictability, so they want a better way to predict the outcome of upcoming elections so they can plan appropriately for future governments. Your company has been asked about the feasibility of building a tool for predicting the results of the upcoming Provincial election in Canada. You will have access to data about the outcome of previous elections, demographic data from the census about the people in the riding at the time of the election and polling data if it was available. The goal is to predict the probability of which party will win each riding in the province if an election were held today. A rival company has proposed solving it by combining hidden Markov models and relational probabilistic models.

(a) Explain how this problem fits into the abstraction of an agent as studied in this course.

(b) Explain how the rival company's solution may work, and explain why they may have chosen the technologies they proposed.

(c) What is the most challenging part of solving this problem? What would you recommend as a way to solve this? Justify any recommendation made.

8. A pure-strategy Nash equilibrium in a game is a Nash equilibrium where no agent acts stochastically. Which of the following games in normal form have a pure strategy Nash equilibrium? For those that do, specify a Nash equilibrium. For those that do not, explain how you know there is no pure-strategy Nash equilibrium.

(a)

		Player 2	
		a2	b2
Player 1	a1	10,10	110,0
	b1	0,110	100,100

(b)

		Player 2	
		a2	b2
Player 1	a1	10,10	11,20
	b1	0,11	20,1

(c)

		Player 2	
		a2	b2
Player 1	a1	10,20	5,10
	b1	7,11	20,12