Exercises for Artificial Intelligence II

Sean B. Holden, 2013-14

1 Introduction

1. Evaluate the integral
$$\int_{-\infty}^{\infty} \exp(-x^2)\, dx.$$

2. Evaluate the integral
$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\left(\mathbf{x}^T \mathbf{\Sigma} \mathbf{x} + \mathbf{x}^T \boldsymbol{\alpha} + \beta\right)\right) dx_1 \cdots dx_n$$
where $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ is a real, symmetric $n \times n$ matrix, $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a real vector, $\beta \in \mathbb{R}$ and
$$\mathbf{x}^T = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_n \end{array}\right] \in \mathbb{R}^n.$$

2 Planning

1. An undergraduate, eager to meet some new friends, has turned up at the term's Big Party, only to find that it is in the home of her arch-rival, who has turned her away. She notices in the driveway a large box and a ladder, and hatches a plan to gatecrash by getting in through a second-floor window. Party on!

   Here is the planning problem. She needs to move the box to the house, the ladder onto the box, then climb onto the box herself and at that point she can climb the ladder to the window.

   Using the abbreviations

   - B - `Box`
   - L - `Ladder`
   - H - `House`
   - C - `Ms Compsci`
   - W - `Window`

   the start state is $\neg\texttt{At}(B, H)$, $\neg\texttt{At}(L, B)$, $\neg\texttt{At}(C, W)$ and $\neg\texttt{At}(C, B)$. The goal is $\texttt{At}(C, W)$. The available actions are:

$\neg\texttt{At}(B,H), \neg\texttt{At}(L,B)$	$\texttt{At}(B,H), \texttt{At}(L,B), \texttt{At}(C,B)$
$\boxed{\texttt{Move}(B,H)}$	$\boxed{\texttt{Move}(C,W)}$
$\texttt{At}(B,H)$	$\texttt{At}(C,W)$

$\neg\texttt{At}(L,B)$	$\neg\texttt{At}(C,B)$	$\texttt{At}(L,B)$
$\boxed{\texttt{Move}(L,B)}$	$\boxed{\texttt{Move}(C,B)}$	$\boxed{\texttt{Move}(L,D)}$
$\texttt{At}(L,B)$	$\texttt{At}(C,B)$	$\neg\texttt{At}(L,B)$

Construct the planning graph for this problem (you should probably start by finding a nice big piece of paper) and use the Graphplan algorithm to obtain a plan.

If you are feeling keen, implement the algorithm for constructing the planning graph and use it to check your answer.

2. Beginning with the domains

$$D_1 = \{\texttt{climber}\}$$
$$D_2 = \{\texttt{home}, \texttt{jokeShop}, \texttt{hardwareStore}, \texttt{spire}\}$$
$$D_3 = \{\texttt{rope}, \texttt{gorilla}, \texttt{firstAidKit}\}$$

and adding whatever actions, relations and so on you feel are appropriate, explain how the problem of purchasing and attaching a gorilla to a famous spire can encoded as a constraint satisfaction problem (CSP).

If you are feeling keen, find a CSP solver and use it to find a plan. The course text book has a code archive including various CSP solvers at:

`http://aima.cs.berkeley.edu/code.html`

The following is an example of how to set up and solve a very simple CSP.

```
import java.io.*;
import java.util.*;
import aima.core.search.csp.*;

public class simpleCSP {
    public static void main(String[] args) {

        Variable v1 = new Variable("v1");
        Variable v2 = new Variable("v2");
        Variable v3 = new Variable("v3");

        List<String> domain1 = new LinkedList<String>();
        domain1.add("red");
        domain1.add("green");
```

```
        domain1.add("blue");

        Domain d1 = new Domain(domain1);

        List<Variable> vars = new ArrayList<Variable>();
        vars.add(v1);
        vars.add(v2);
        vars.add(v3);

        CSP csp = new CSP(vars);

        csp.setDomain(v1,d1);
        csp.setDomain(v2, d1);
        csp.setDomain(v3, d1);

        Constraint c1 = new NotEqualConstraint(v1,v2);
        Constraint c2 = new NotEqualConstraint(v1,v3);
        Constraint c3 = new NotEqualConstraint(v2,v3);
        csp.addConstraint(c1);
        csp.addConstraint(c2);
        csp.addConstraint(c3);

        ImprovedBacktrackingStrategy solver =
            new ImprovedBacktrackingStrategy();
        Assignment solution = new Assignment();
        solution = solver.solve(csp);

        System.out.println(solution);
    }
}
```

3. Exam question: 2008, paper 7, question 6.

4. Exam question: 2009, paper 7, question 4.

5. Exam question: 2011, paper 7, question 2.

6. Exam question: 2012, paper 8, question 2.

3 Uncertainty

1. Prove that conditional independence, defined in the lectures notes as

$$\Pr(A, B|C) = \Pr(A|C)\Pr(B|C)$$

can equivalently be defined as

$$\Pr(A|B, C) = \Pr(A|C).$$

2. Derive, from first principles, the general form of Bayes rule

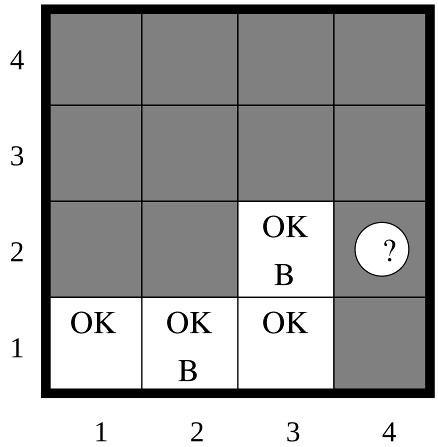
$$\Pr(A|B, C) = \frac{\Pr(B|A, C)\Pr(A|C)}{\Pr(B|C)}.$$

3. This question revisits the Wumpus World, but now our hero, having learned some probability by attending *Artificial Intelligence II*, will use probabilistic reasoning instead of situation calculus.

   Our hero, through careful consideration of the available knowledge on Wumpus caves, has established that each square contains a pit with prior probability 0.3, and pits are independent of one-another. Let $\mathtt{Pit}_{i,j}$ be a Boolean random variable (RV) denoting the presence of a pit at row i, column j. So for all i, j

$$\Pr(\mathtt{Pit}_{i,j} = \top) = 0.3 \tag{1}$$
$$\Pr(\mathtt{Pit}_{i,j} = \bot) = 0.7 \tag{2}$$

   In addition, after some careful exploration of the current cave, our hero has discovered the following.



$$\mathtt{Pit}_{1,1} = \bot$$
$$\mathtt{Pit}_{1,2} = \bot$$
$$\mathtt{Pit}_{1,3} = \bot$$
$$\mathtt{Pit}_{2,3} = \bot$$

   B denotes squares where a breeze is perceived. Let $\mathtt{Breeze}_{i,j}$ be a Boolean RV denoting the presence of a breeze at i, j

$$\mathtt{Breeze}_{1,2} = \mathtt{Breeze}_{2,3} = \top \tag{3}$$
$$\mathtt{Breeze}_{1,1} = \mathtt{Breeze}_{1,3} = \bot \tag{4}$$

   He is considering whether to explore the square at $2, 4$. He will do so if the probability that it contains a pit is less than 0.4. Should he?

   Hint: The RVs involved are $\mathtt{Breeze}_{1,2}, \mathtt{Breeze}_{2,3}, \mathtt{Breeze}_{1,1}, \mathtt{Breeze}_{1,3}$ and $\mathtt{Pit}_{i,j}$ for all the i, j. You need to calculate

$$\Pr(\mathtt{Pit}_{2,4} | \text{all the evidence you have so far})$$

4. Continuing with the running example of the roof-climber alarm...

   The porter in lodge 1 has left and been replaced by a somewhat more relaxed sort of chap, who doesn't really care about roof-climbers and therefore acts according to the probabilities

$$\Pr(l1|a) = 0.3 \qquad \Pr(\neg l1|a) = 0.7$$
$$\Pr(l1|\neg a) = 0.001 \quad \Pr(\neg l1|\neg a) = 0.999$$

   Your intrepid roof-climbing buddy is on the roof. What is the probability that lodge 1 will report him? Use the variable elimination algorithm to obtain the relevant probability. Do you learn anything interesting about the variable $L2$ in the process?

5. In the lecture notes, an example was given for which we would expect Pr$(A \rightarrow B)$ to be (relatively) much larger than Pr$(B|A)$. Suggest a situation where the converse would be true.

6. Later in the course it is shown that in constructing a two-class classifier (such as a multilayer perceptron) the optimal approach involves computing Pr$(\texttt{class}|\texttt{features})$. Suggest an approach to performing this calculation in practice. (Hint: apply Bayes' theorem and estimate some probabilities.) What problems might this present in practice, and what assumption(s) might you introduce to overcome them?

7. In designing a Bayesian network you wish to include a node representing the value reported by a sensor. The quantity being sensed is real-valued, and if the sensor is working correctly it provides a value close to the correct value, but with some noise present. The correct value is provided by its first parent. A second parent is a boolean random variable that indicates whether the sensor is faulty. When faulty, the sensor flips between providing the correct value, although with increased noise, and a known, fixed incorrect value, again with some added noise. Suggest a conditional distribution that could be used for this node.

8. Exam question: 2005, paper 8, question 2.

9. Exam question: 2006, paper 8, question 9.

10. Exam question: 2009, paper 8, question 1.