Exercises for Artificial Intelligence II

Sean B. Holden, 2013-16

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:

$\neg\texttt{At}(B, H), \neg\texttt{At}(L, B)$

$\texttt{Move}(B, H)$

$\texttt{At}(B, H)$

$\texttt{At}(B, H), \texttt{At}(L, B), \texttt{At}(C, B)$

$\texttt{Move}(C, W)$

$\texttt{At}(C, W)$

$\neg\texttt{At}(L, B)$

$\texttt{Move}(L, B)$

$\texttt{At}(L, B)$

$\neg\texttt{At}(C, B)$

$\texttt{Move}(C, B)$

$\texttt{At}(C, B)$

$\texttt{At}(L, B)$

$\texttt{Move}(L, D)$

$\neg\texttt{At}(L, B)$

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

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

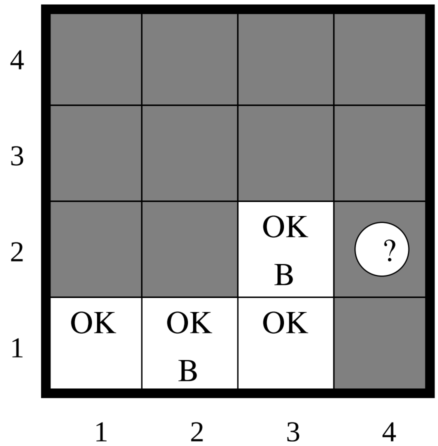
3. This question revisits the Wumpus World, but now our hero, having learned some probability by attending *Artificial Intelligence II*, will use probabilistic reasoning instead of situation calculus.

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 $\texttt{Pit}_{i,j}$ be a Boolean random variable (RV) denoting the presence of a pit at row i, column j. So for all i, j

$$\Pr(\texttt{Pit}_{i,j} = \top) = 0.3 \quad (1)$$

$$\Pr(\texttt{Pit}_{i,j} = \bot) = 0.7 \quad (2)$$

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



$\texttt{Pit}_{1,1} = \bot$

$\texttt{Pit}_{1,2} = \bot$

$\texttt{Pit}_{1,3} = \bot$

$\texttt{Pit}_{2,3} = \bot$

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

$$\texttt{Breeze}_{1,2} = \texttt{Breeze}_{2,3} = \top \quad (3)$$

$$\texttt{Breeze}_{1,1} = \texttt{Breeze}_{1,3} = \bot \quad (4)$$

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 $\texttt{Breeze}_{1,2}, \texttt{Breeze}_{2,3}, \texttt{Breeze}_{1,1}, \texttt{Breeze}_{1,3}$ and $\texttt{Pit}_{i,j}$ for all the i, j. You need to calculate

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

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.

4 Making decisions

1. Prove the result mentioned on slide 161:

$$\texttt{VPI}_E(E', E'') = \texttt{VPI}_E(E') + \texttt{VPI}_{E,E'}(E'').$$

2. Evil Robot is teaching himself surgery. He believes that there are two treatments, t_1 and t_2 suitable for his first patient, each having three possible outcomes: `cure`, `death` and `amputation`. These have utilities of 100, -1000 and -250 respectively. Evil robot thinks that t_1 has probabilities 0.8, 0.1 and 0.1 respectively for the three outcomes and treatment t_2 has probabilities 0.75, 0.05 and 0.2. Compute the expected utility of each treatment.

   Evil Robot has been studying hard, and has learned that an unpleasant test T is available that might help him choose the better treatment. The test has a cost to the patient of -50, while the cost of not performing it is -2. (Evil robot will nonetheless conduct some slightly unpleasant tests.) He estimates that the probability of the test being positive is 0.7. He also thinks that, armed with a positive test he can give t_1 outcome probabilities of 0.9, 0.01 and 0.09 respectively, and t_2 outcome probabilities of 0.85, 0.02 and 0.13. If test T is negative then the outcome probabilities are unchanged. (He does the other tests just for fun.)

   In the interest of the patient, should Evil Robot use test T?

3. Exam question: 2007, paper 8, question 9.

4. Exam question: 2011, paper 8, question 8.

5. Exam question: 2013, paper 8, question 2.

5 HMMs

1. Derive the equation

$$b_{t+1:T} = \mathbf{S}\mathbf{E}_{t+1} b_{t+2:T}$$

   for the backward message in a hidden Markov model (lecture slide 208).

2. Explain why the backward message update should be initialized with the vector $(1, \ldots, 1)$.

3. Establish how the prior $\Pr(S_0)$ should be included in the derivation of the Viterbi algorithm. (This is mentioned on slide 192, but no detail is given.)

4. A hidden Markov model has transition matrix $S_{ij} = \Pr(S_{t+1} = s_j | S_t = s_i)$ where

$$\mathbf{S} = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.1 & 0.6 & 0.3 \\ 0.8 & 0.1 & 0.1 \end{pmatrix}.$$

   In any state we observe one of the symbols $\triangle$, $\triangledown$, $\bigcirc$, $\square$ with the following probabilities:

	$\triangle$	$\triangledown$	$\bigcirc$	$\square$
s_1	0.7	0.1	0.1	0.1
s_2	0.3	0.2	0.4	0.1
s_3	0.4	0.2	0.2	0.2

   Prior probabilites for the states are $\Pr(s_1) = 0.3$, $\Pr(s_2) = 0.3$ and $\Pr(s_3) = 0.4$. We observe the sequence of symbols

$$\bigcirc \bigcirc \square \triangle \triangle \square \triangledown \square.$$

   Use the Viterbi algorithm to infer the most probable sequence of states generating this sequence.

5. Exam question: 2005, paper 9, question 8.

6. Exam question: 2008, paper 9, question 5.

7. Exam question: 2010, paper 7, question 4.

8. Exam question: 2013, paper 7, question 2.

6 Bayesian learning

1. Derive the *weight decay* training algorithm

$$\mathbf{w}_{\text{MAP}} = \underset{\mathbf{w}}{\text{argmin}} \frac{\alpha}{2}||\mathbf{w}||^2 + \frac{\beta}{2}\sum_{i=1}^{m}(y_i - f(\mathbf{w}; \mathbf{x}_i))^2$$

   given on slide 268.

2. Use the standard Gaussian integral to derive the final equation for Bayesian regression

$$p(Y|\mathbf{y}, \mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(y - f(\mathbf{w}_{\text{MAP}}; \mathbf{x}))^2}{2\sigma_y^2}\right)$$

where

$$\sigma_y^2 = \frac{1}{\beta} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}$$

given on slide 284.

3. This question asks you to produce a version of the graph on slide 286, but using the Metropolis algorithm instead of the solution obtained by approximating the integral. Any programming language is fine, although Matlab is probably the most straightforward.

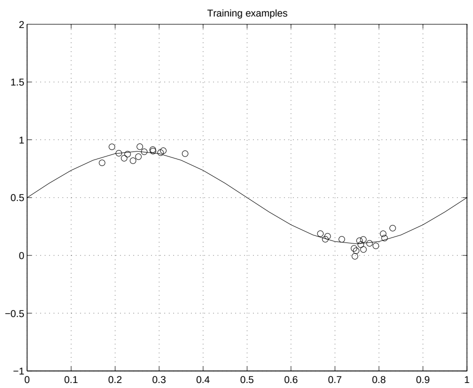
   The data is simple artificial data for a one-input regression problem. Use the target function

$$f(x) = \frac{1}{2} + 0.4 \sin 2\pi x$$

   and generate 30 examples clustered around $x = 0.25$ and $x = 0.75$. Then label these examples

$$y(x) = f(x) + n$$

   where n is Gaussian noise of standard deviation 0.05. Plot the data as follows:



   Let $\mathbf{w} \in \mathbb{R}^W$ be the vector of all the weights in a network. Your supervised learner should be based on a prior density

$$p(\mathbf{w}) = \left(\frac{2\pi}{\alpha}\right)^{-W/2} \exp\left(-\frac{\alpha}{2}||\mathbf{w}||^2\right)$$

   on the weights. A value of $\alpha = 1$ is reasonable. The likelihood used should be

$$p(\mathbf{y}|\mathbf{w}) = \left(\frac{2\pi}{\beta}\right)^{-m/2} \exp\left(-\frac{\beta}{2}\sum_{i=1}^{m}(y(x_i) - h(\mathbf{w}; x_i))^2\right)$$

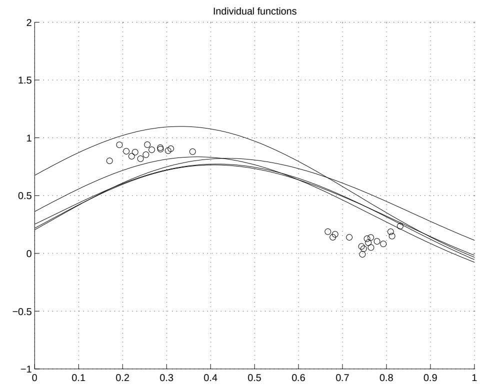
   where m is the number of examples and $h(\mathbf{w}; x)$ is the function computed by the neural network with weights $\mathbf{w}$. A value of $\beta = 1/(0.05)^2$ is appropriate. Note that we are assuming that hyperparameters α and β are known, and the prior and likelihood used are the same as those used in the lectures.

   Complete the following steps:

(a) Write a function `simpleNetwork` function implementing a multilayer perceptron with a single hidden layer, a basic feedforward structure as illustrated in the AI I lectures, and a single output node. The network should use sigmoid activation functions for the hidden units and a linear activation function for its output. You should use a network having 4 hidden units.

(b) Starting with a weight vector chosen at random, use the Metropolis algorithm to sample the posterior distribution $p(\mathbf{w}|\mathbf{y})$. You should generate a sequence of 100 weight vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{100}$.

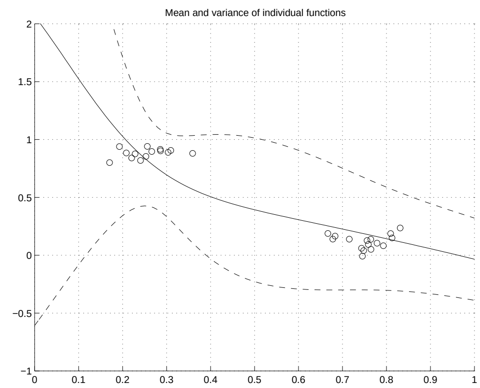
(c) Plot the function $h(\mathbf{w}_i; x)$ computed by the neural network for a few of the weight vectors obtained.



(d) Discard the first 50 weight vectors generated. Using the remainder, calculate the mean and variance of the corresponding functions using

$$\text{mean}(x) = \frac{1}{50} \sum_{i=51}^{100} h(\mathbf{w}_i; x)$$

and a similar expression for the variance. Plot the mean function along with error bars provided by the variance.



4. Can you incorporate hyperparameter estimation into your solution to the previous problem? If so, do the results make sense?

5. Explain how the Gibbs algorithm might be applied to the Bayesian network developed earlier for the *roof-climber alarm* problem.

6. Slide 314 uses the following estimate for the variance of a random variable:

$$\sigma^2 \simeq \hat{\sigma}^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n}(X_i - \hat{X}_n)^2\right].$$

   Show that this estimate is unbiased; that is,

$$\mathbb{E}\left[\hat{\sigma}^2\right] = \sigma^2.$$

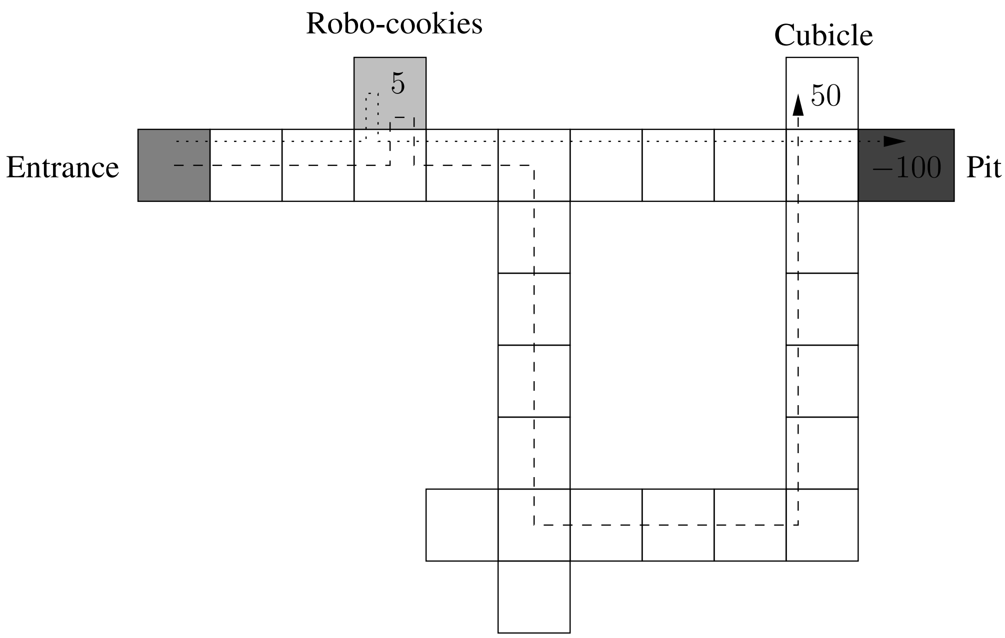
7. Show that if a random variable has zero mean then dividing it by its standard deviation σ results in a new random variable having variance 1.

8. Verify the expression in point 4 on slide 317.

9. Exam question: 2010, paper 8, question 2.

7 Reinforcement learning

1. Evil Robot's Dungeon of Darkness was constructed in such a way that the Pit of Endless Disgruntlement is very close to the Cubicle of Inventive Punishment. Evil Robot likes going to the Cubicle of Inventive Punishment as he gets to be nasty to a human. He does not however like falling into the Pit of Endless Disgruntlement, because that makes him very disgruntled. Between these locations and the entrance is a table with robo-cookies.



   Unfortunately the part of his memory related to navigating the Dungeon has accidentally been wiped, so he can't find his way to punish the human responsible. He is running a simulation of the Q-learning algorithm to re-learn it. His actions are to move left, right, up or down one square. Assuming all Q values are initialised to 0, explain how the Q-learning algorithm operates if the dotted route is followed once, then the dashed route is followed, then the dotted route is followed again. Where no reward is indicated assume the reward is 0.

2. Exam question: 2007, paper 9, question 9.

3. Exam question: 2012, paper 7, question 2.