Name:

Birth Date:

Matriculation Number:

Exam
Artificial Intelligence 2

Feb. 13., 2019

	To be used for grading, do not write here													
prob.	1.1	1.2	1.3	2.1	2.2	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2	Sum	grade
total	4	3	7	12	12	4	3	8	4	8	8	5	6	84	
reached															

Exam Grade: Bonus Points: Final Grade:

The “solutions” to the exam/assignment problems in this document are supplied to give students a starting point for answering questions. While we are striving for helpful “solutions”, they can be incomplete and can even contain errors.

If you find “solutions” you do not understand or you find incorrect, discuss this on the course forum and/or with your TA and/notify the instructors.

In any case, grading student’s answers is not a process of simply “comparing with the reference solution”, therefore errors in the “solutions” are not a problem in this case.

In the course Artificial Intelligence I/II we award 5 bonus points for the first student who reports a factual error (please report spelling/formatting errors as well) in an assignment or old exam and 10 bonus points for an alternative solution (formatted in LaTeX) that is usefully different from the existing ones.

1 Bayesian Reasoning

Problem 1.1 (Bayesian Rules) 4 pt

Name four of the basic rules in Bayesian inference and explain each with a short sentence and formula. 4 min

Solution:

1. Bayes rule (compute $P(A|B)$ from $P(B|A)$,
2. Normalization (Fixing evidence e, updating the probabilities of all other events using a normalization constant α),
3. Marginalization ($P(A) = \sum_y P(A, y)$),
4. Chain rule ($P(A_1, \ldots, A_n) = P(A_n|A_{n-1}, \ldots, A_1) \cdot P(A_{n-1}|A_{n-2}, \ldots, A_1) \cdot \ldots$)
5. Product rule ($P(A, B) = P(A|B)P(B)$)
6. Conditional Independence (Not really bayesian inference, but rather bayesian networks, but we'll be lenient)

Problem 1.2 (Conditional Independence) 3 pt

Define *conditional independence.* 3 min

Solution: Two events A, B are conditionally independent given C, if $P(A \wedge B|C) = P(A|C)P(B|C)$.

Problem 1.3 (Medical Bayesian Network 2) 7 pt 7 min

Both smoking and living in a city with high air pollution can cause lung cancer, which can be indicated by a patient coughing up blood. We consider the following random variables for a given patient:

- *Smoke*: The patient is a smoker.
- *Smog*: The patient lives in a polluted city.
- *Blood*: The patient is coughing up blood.
- *LC*: The patient has lung cancer.

1. Draw the corresponding Bayesian network for the above data using the algorithm presented in the lecture, assuming the variable order *Smoke*, *Smog*, *Blood*, *LC*. Explain rigorously(!) the exact criterion for whether to insert an arrow between two nodes.
2. Which arrows are causal and which are diagnostic? Which order of variables would be better suited for constructing the network?
3. How do we compute the probability the patient is a smoker, given that they have lung cancer? State the query variables, hidden variables and evidence and write down the equation for the probability we are interested in.

Solution:

2 Decision Theory

Problem 2.1 (Markov Decision Procedures) 12 pt 12 min

1. How do Markov decision procedures differ from (simple) decision networks?
2. How does the value iteration algorithm work? (Give an actual equation and explain its role in the algorithm)
3. What is the disadvantage of value iteration that is "fixed" by policy iteration?
4. How can we reduce *partially observable Markov decision procedures* to normal MDPs?

Solution:

1. In Markov decision procedures, the probabilistic model is a Markov Process (i.e. random variables are indexed over time, Markov Properties)

2. We assing a random utility to each state and update them using the Bellman equation:

$$U(s) = R(s) + \gamma \cdot \max_a \left(\sum_{s'} U(s') \cdot T(s, a, s') \right)$$

   Once this iteration has converged, we can compute the "best" action for each state by considering the expected utilities of all possible actions.

3. The policy resulting from value iteration can be stable long before the individual utilities have converged to their precise values.

4. By introducing belief states representing the probability distribution over the physical state space (i.e. the belief state space has one dimension for each physical state).

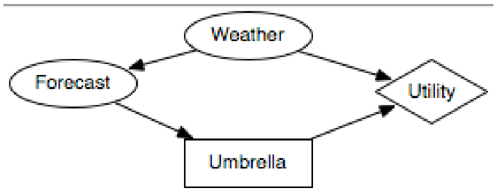
Problem 2.2 (Decision Network) 12 pt

12 min

You try to decide on whether to take an umbrella to Uni. Obviously, it's useful to do so if it rains when you go back home, but it's annoying to carry around if it doesn't even rain. You look at the weather forecast, which hast three possible values: sunny, cloudy and rainy.

1. Draw the decision network for bringing/leaving an umbrella depending on whether it does or doesn't rain later.
2. Explain *formally* how to compute whether or not to take an umbrella, assuming you know $P(rain = b|forecast = x)$ for all $b \in \texttt{Bool}, x \in \{\texttt{sunny}, \texttt{cloudy}, \texttt{rainy}\}$.

Solution:



Let $U_{\pm r,\pm u}$ be the base utilities of having an/no umbrella when it rains/doesn't rain. Assume the forecast says x, then compute:

$$U(umb) = P(rain = \top|forecast = x)U_{+r,+u} + P(rain = \bot|forecast = x)U_{-r,+u}$$
$$U(\neg umb) = P(rain = \top|forecast = x)U_{+r,-u} + P(rain = \bot|forecast = x)U_{-r,-u}$$

If the former is greater than the latter you should take an umbrella.

3 Markov Models

Problem 3.1 (Bellman Equation) 4 pt

State the Bellman Equation and explain every symbol in the equation and what the equation is used for and how. 4 min

Solution:

$$U(s) = R(s) + \gamma \cdot \max_{a \in A(s)} \left(\sum_{s'} P(s'|s,a) \cdot U(s') \right)$$

The meaning of the components is as follows:

- $U(s)$: the utility of the state s (long-term, global)
- $R(s)$: the reward at state s (short-term, local)
- $A(s)$: the set of actions available in state s
- $\max_{a \in A(s)}$: take the maximum over all available actions in state s
- $P(s'|s,a)$: the probability that taking action a in state s yields state s'
- $U(s')$: the utility in successor state s'
- $(\sum_{s'} P(s'|s,a) \cdot U(s'))$: the expected utility of action a by summing over all possible successor states

The equation is used to compute the utility of every state. The algorithm uses the equation as an iteration operator that computes new values for every $U(s)$ by evaluating the right hand side for the current values of U. If this leads to a fixpoint, a solution for the utilities has been found.

Problem 3.2 (Stationary) 3 pt

Define what it means for a Markov model to be *stationary*, and why we are interested in stationarity. 3 min

Solution: A Markov process is called stationary if the transition model is independent of time, i.e. $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ is the same for all t.

We like stationary Markov processs, since they are finite.

Problem 3.3 (Sleeping Patterns Predictions) 8 pt 8 min

Your roommate tends to keep you up by blasting music whenever they are awake. Notably, they tend to sleep a lot less when they are stressed (binary variable St), but since you don't talk to each other you never know when they are. You only observe whether they sleep a lot (Sl) or little ($\neg Sl$). Stress seems to come in phases and last for a couple of days, so if they are stressed at day t, they will more likely be stressed at day $t+1$ as well (and analogously for $\neg St$).

1. Model this situation as a Markov Model and explain what the *prediction*, *filtering* and *smoothing* algorithms compute in this scenario.

2. Give the underlying equations for the first two of these algorithms and explain what each variable in the equation represents.

Solution:

1. **Prediction** Given the amount of sleep up to time t_0, compute the probability of them being stressed at time $t_1 > t_0$

   Filtering Given the of sleep up to now, compute the probability of them being stressed right now

   Smoothing Given the amount of sleep up to t_0, compute the probability that they were stressed at an earlier point $t_1 < t_0$.

2. $$P(St_{t+1}|Sl_{1:t+1}) = \alpha P(Sl_{t+1}|St_{t+1}) \cdot \sum_{st_t} P(St_{t+1}|st_t, Sl_{1:t}) \cdot P(st_t|Sl_{1:t})$$

---

4 Learning

Problem 4.1 (Overfitting) 4 pt

4 min

Explain what *overfitting* means and why we want to avoid it.

Solution: Overfitting is a modeling error that occurs when the chosen hypothesis is too closely fit to a sample set of data points. It picks an overly complex hypothesis that also explains idiosyncrasies and errors in the data. A simpler hypothesis that fits the data less exactly is often a better match for the underlying mechanisms.

Problem 4.2 (Home Decisions) 8 pt

Eight people go sunbathing. Some of them got a sunburn, others didn't: 8 min

Name	Hair	Height	Weight	Lotion	Result
Sarah	Blonde	Average	Light	No	Sunburned
Dana	Blonde	Tall	Average	Yes	None
Alex	Brown	Short	Average	Yes	None
Annie	Blonde	Short	Average	No	Sunburned
Julie	Blonde	Average	Light	No	None
Pete	Brown	Tall	Heavy	No	None
John	Brown	Average	Heavy	No	None
Ruth	Blonde	Average	Light	No	None

Explain how the information-theoretic decision tree learning algorithm would proceed on this table (up to two iterations). Explicitly state how to compute the information gain (and what that means).

Note that you do not need to compute any actual values; if it is helpful for your explanation, you may guess any values you might want to use.

Note that *Name* is only an index, not a (meaningful) attribute!

Solution:

$$E_0 := I(\langle\frac{2}{8},\frac{6}{8}\rangle) = -\frac{2}{8}\log_2(\frac{2}{8}) - \frac{6}{8}\log_2(\frac{6}{8}) \approx 0.81$$

$$\texttt{Gain(Hair)} = E_0 - \underbrace{\frac{5}{8}I(\langle\frac{2}{5},\frac{3}{5}\rangle)}_{\texttt{Blonde}} - \underbrace{\frac{3}{8}I(\langle 0,1\rangle)}_{\texttt{Brown}} \qquad \approx 0.20$$

$$\texttt{Gain(Height)} = E_0 - \underbrace{\frac{4}{8}I(\langle\frac{1}{4},\frac{3}{4}\rangle)}_{\texttt{Average}} - \underbrace{\frac{2}{8}I(\langle 0,1\rangle)}_{\texttt{Tall}} - \underbrace{\frac{2}{8}I(\langle\frac{1}{2},\frac{1}{2}\rangle)}_{\texttt{Short}} \qquad \approx 0.16$$

$$\texttt{Gain(Weight)} = E_0 - \underbrace{\frac{3}{8}I(\langle\frac{1}{3},\frac{2}{3}\rangle)}_{\texttt{Average}} - \underbrace{\frac{3}{8}I(\langle\frac{1}{3},\frac{2}{3}\rangle)}_{\texttt{Light}} - \underbrace{\frac{2}{8}I(\langle 0,1\rangle)}_{\texttt{Heavy}} \qquad \approx 0.12$$

$$\texttt{Gain(Lotion)} = E_0 - \underbrace{\frac{2}{8}I(\langle 0,1\rangle)}_{\texttt{Yes}} - \underbrace{\frac{6}{8}I(\langle\frac{2}{6},\frac{4}{6}\rangle)}_{\texttt{No}} \qquad \approx 0.12$$

`Hair` has the highest information gain, so we split here. All table entries with `Brown` have result `None`, so we continue with `Hair` = `Blonde`:

$$E_1 := I(\langle\frac{2}{5},\frac{3}{5}\rangle) \approx 0.97$$

$$\texttt{Gain}(\texttt{Height}) = E_1 - \underbrace{\frac{3}{5}I(\langle\frac{1}{3},\frac{2}{3}\rangle)}_{\texttt{Average}} - \underbrace{\frac{1}{5}I(\langle 0,1\rangle)}_{\texttt{Tall}} - \underbrace{\frac{1}{5}I(\langle 1,0\rangle)}_{\texttt{Short}} \qquad \approx 0.42$$

$$\texttt{Gain}(\texttt{Weight}) = E_1 - \underbrace{\frac{2}{5}I(\langle\frac{1}{2},\frac{1}{2}\rangle)}_{\texttt{Average}} - \underbrace{\frac{3}{5}I(\langle\frac{1}{3},\frac{2}{3}\rangle)}_{\texttt{Light}} - \underbrace{0}_{\texttt{Heavy}} \qquad \approx 0.02$$

$$\texttt{Gain}(\texttt{Lotion}) = E_1 - \underbrace{\frac{1}{5}I(\langle 0,1\rangle)}_{\texttt{Yes}} - \underbrace{\frac{4}{5}I(\langle\frac{2}{4},\frac{2}{4}\rangle)}_{\texttt{No}} \qquad \approx 0.17$$

`Height` has the highest information gain, so we proceed here. All short blondes are sunburned, all tall blondes are not, hence we only need consider `Average`...

Problem 4.3 (Backpropagation) 8 pt

Explain what *Backpropagation* means in the context of Neural Networks, when and why we need it, and how to do it using an example. 8 min

Solution: A possible answer:

Backpropagation is an algorithm for training feedforward neural networks for supervised learning. It computes the gradient of the loss function with respect to the weights of the network for a single input–output example.

5 Communication with Natural Language

Problem 5.1 (Ambiguity) 5 pt

5 min

1. Explain the concept of ambiguity of natural languages.
2. Give two examples of different kinds of ambiguity and explain the readings.

Solution:

1. Ambiguity is the phenomenon that in natural languages a single utterance can have multiple readings.
2. Here are some examples
   - *bank* can be a financial institution or a geographical feature.
   - In *I saw her duck* the word *duck* can be a verb or a noun.
   - *Time flies like an arrow* could be about the preferences of special insects (*time flies*) or about the fact that time passes quickly – e.g. in an exam.
   - In *Peter saw the man with binoculars*, it could be Peter who is using binoculars, or it could be that Peter saw *the man* who had *binoculars*.

Problem 5.2 (Language Identification) 6 pt

You are given an English, a German, a Spanisch, and a French text corpus of considerable size, and you want to build a language identification algorithm A for the EU administration. Concretely A takes a string as input and classifies it into one of the four languages $\ell^* \in \{English, German, Spanish, French\}$. The prior probability distribution for the strings being English/German/Spanisch/French, is $\langle 0.4, 0.2, 0.15, 0.15\rangle$. 6 min

How would you proceed to build algorithm A? Specify the general steps and give/derive the formula for computing ℓ given a string $\mathbf{c}_{1:N}$.

Solution:

1. Build a trigram language model $\mathbf{P}(\mathrm{c}_i \mid \mathbf{c}_{i-2:i-1}, \ell)$ for each candidate language ℓ by counting trigrams in a ℓ-corpus.

2. Apply Bayes' rule and the Markov property to get the most likely language:

$$\begin{aligned}
\ell^* &= \operatorname*{argmax}_{\ell}(P(\ell \mid \mathbf{c}_{1:N})) \\
&= \operatorname*{argmax}_{\ell}(P(\ell) \cdot P(\mathbf{c}_{1:N} \mid \ell)) \\
&= \operatorname*{argmax}_{\ell}(P(\ell) \cdot \prod_{i=1}^{N} P(\mathrm{c}_i \mid \mathbf{c}_{i-2:i-1}, \ell))
\end{aligned}$$