4 Exercise Session 4

Exercises on probabilistic methods and reinforcement learning.

4.1 Naive Bayes

Classify `<single,light,one>` with Naive Bayes given the following examples:

```
Healthy: * <single,dark,one>
         * <single,light,two>
         * <double,light,one>

Virulent: * <single,dark,two>
          * <double,dark,one>
          * <double,light,two>
```

Solution

According to Naive Bayes, the most likely hypothesis is:

$$\text{argmax}_H P(single|H)P(light|H)P(one|H)P(H)$$

$P(single|Healthy) = 2/3$, $P(double|Healthy) = 1/3$,
$P(dark|Healthy) = 1/3$, $P(light|Healthy) = 2/3$,
$P(one|Healthy) = 2/3$, $P(two|Healthy) = 1/3$
$P(Healthy) = 1/2$

$P(single|Virulent) = 1/3$, $P(double|Virulent) = 2/3$,
$P(dark|Virulent) = 2/3$, $P(light|Virulent) = 1/3$,
$P(one|Virulent) = 1/3$, $P(two|Virulent) = 2/3$
$P(Virulent) = 1/2$

$$\begin{array}{rcl} \text{Likelihood(Healthy)} & = & P(\text{single}|\text{Healthy})P(\text{light}|\text{Healthy})P(\text{one}|\text{Healthy})P(\text{Healthy}) \\ & = & 2/3 \cdot 2/3 \cdot 2/3 \cdot 1/2 = 4/27 \\ \text{Likelihood(Virulent)} & = & 1/3 \cdot 1/3 \cdot 1/3 \cdot 1/2 = 1/54 \end{array}$$

⇒ conclude class Healthy

4.2 Text Classification

Classify the sentence “Cats eat mice and dogs bury bones.” given the following examples:

- Class A: “The cat crabs the curls off the stairs.”, “The cat curled under the stairs.”
- Class B: “It’s raining cats and dogs.”

and given Voc = {Cat, Crab, Curl, Stairs, Rain, Dog}.

Solution

There is a number of subtle differences between applying Naive Bayes for classifying attribute-value data as in 4.1 and for text classification. The procedure is as follows:

1. Remove words not in the vocabulary.

   - Class A "cat crabs curls stairs", "cat curled stairs."
   - Class B: "raining cats dogs."

2. Discard grammatical information, i.e. for each word, use the same grammatical form as in the vocabulary.
   - Class A: "Cat Crab Curl Stairs", "Cat Curl Stairs."
   - Class B: "Rain Cat Dog."

3. Compute probabilities of classes using the number of documents: $P(\text{Class A}) = \frac{2}{3}$, $P(\text{Class B}) = \frac{1}{3}$

4. To compute $P(\text{word occurs}|\text{class})$, count how often this word occurs among all the words occurring in this class and perform the Laplacian correction based on the size of vocabulary (Hint: you are effectively concatenating all documents of a particular class and counting word occurrences in the resulting long document):
   For example, the vocabulary size $|Voc| = 6$; Class A has 7 word occurrences in total, "Cat Crab Curl Stairs Cat Curl Stairs.", and "Cat" occurs 2 times. Hence, $P(\text{Cat}|A) = \frac{2+\mathbf{1}}{7+\mathbf{6}} = \frac{3}{13}$ (Laplacian correction is in bold).

   In standard Naive Bayes, $P(\textit{word occurs}|\textit{class})$ *would be computed as the proportion of examples of the* class *where a particular word occurs. For example, the uncorrected* $P(\text{Cat}|A)$ *would be* $2/2 = 1$ *instead of* $2/7$.

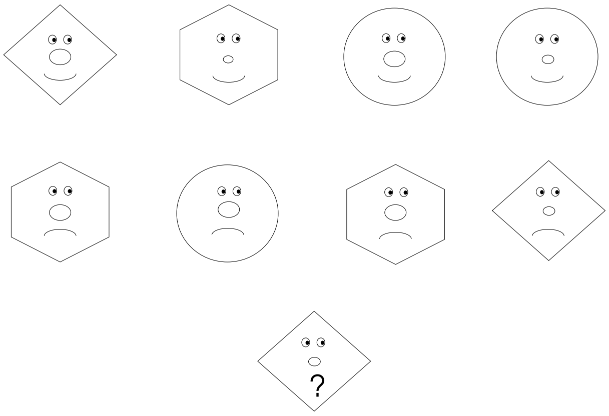
5. To compute the prediction for a sentence, first perform the transformations ("Cats eat mice and dogs bury bones." ⇒ "Cat Dog"); then only consider words that occur in this sentence when computing the likelihoods:
   - $L(\text{Class A}|\text{Cat Dog}) = P(A) \cdot P(Cat|A) \cdot P(Dog|A) = \frac{2}{3} \cdot \frac{2+1}{7+6} \cdot \frac{0+1}{7+6} = \frac{2}{3} \cdot \frac{3}{13} \cdot \frac{1}{13} = 0.012$
   - $L(\text{Class B}|\text{Cat Dog}) = P(B) \cdot P(Cat|B) \cdot P(Dog|B) = \frac{1}{3} \cdot \frac{1+1}{3+6} \cdot \frac{1+1}{3+6} = \frac{1}{3} \cdot \frac{2}{9} \cdot \frac{2}{9} = 0.016$

   $L(B) > L(A) \Rightarrow$ the sentence "Cats eat mice and dogs bury bones." belongs to Class B.

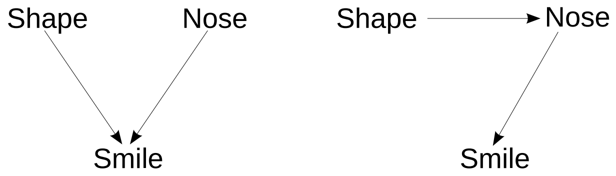
   In standard Naive Bayes, we would consider all attributes, i.e. every word in the vocabulary whether or not it occurs in the sentence. Then $L(A|Cat\ Dog) = P(A) \cdot P(Cat|A) \cdot P(Dog|A) \cdot \mathbf{P}(\neg\mathbf{Crab}|\mathbf{A}) \cdot \mathbf{P}(\neg\mathbf{Curl}|\mathbf{A}) \cdot \mathbf{P}(\neg\mathbf{Stairs}|\mathbf{A}) \cdot \mathbf{P}(\neg\mathbf{Rain}|\mathbf{A})$. *However, in practice, for sufficiently large vocabularies and document collections, a probability of a word not occurring in a document of a class is typically very high (in other words, most words do not occur in most documents); hence these additional terms do not change the relative ranking of class likelihoods. Very common words, such as "and", are counterexamples, however they are not indicative of class membership either and typically are not included in vocabularies.*

To sum up, instead of treating each word as a binary attribute and each document as a training example, Naive Bayes for text classification uses several domain-specific simplifications to make computations easier.

4.3 Bayesian Networks



1. (Optional) Predict wether the last face is smiling or not using Naive Bayes.
2. Calculate all the probability tables for each of the Bayesian networks shown in the figure below according to the given examples-faces. Predict the class of the last face according to each of the networks.



3. Draw the Bayesian network that is equivalent to Naive Bayes.

Solution

1. Using Naive Bayes:

   smile $P(\text{smile}) \cdot P(\text{small }|\text{smile}) \cdot P(\Diamond\ |\text{smile}) = 1/2 \cdot 1/2 \cdot 1/4 = 1/16$

   unhappy $P(\text{unhappy}) \cdot P(\text{small }|\text{unhappy}) \cdot P(\Diamond\ |\text{unhappy}) = 1/2 \cdot 1/4 \cdot 1/4 = 1/32$

   $\Rightarrow$ conclude class smile

2. The conditional probability table for the first network:

	◊, small	⬡, small	◯, small	◊, big	⬡, big	◯, big
smile	0	1	1	1	0	1/2
unhappy	1	0	0	0	1	1/2

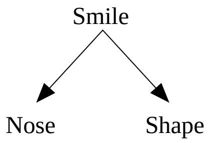
   The conditional probability tables for the second network:

	◊	⬡	◯
small	1/2	1/3	1/3
big	1/2	2/3	2/3

	small	big
smile	2/3	2/5
unhappy	1/3	3/5

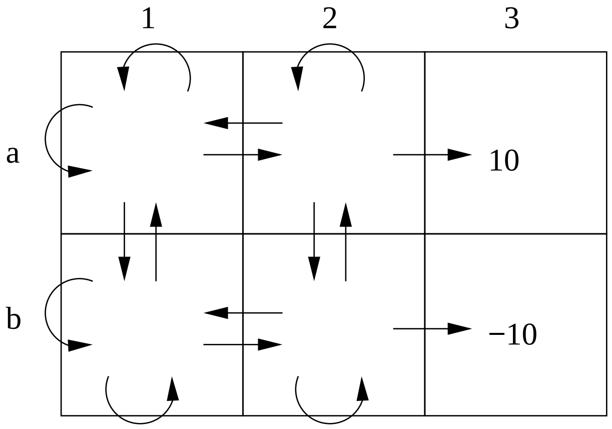
   The first network predicts class unhappy because $P(\text{unhappy}|\ \Diamond, \text{small}) > P(\text{smile}|\ \Diamond, \text{small})$. The second network predicts class smile because $P(\text{smile}|\ \text{small}) > P(\text{unhappy}|\ \text{small})$.

3. The equivalent Bayesian network:



4.4 Reinforcement Learning in a Nondeterministic Environment

Consider the following environment:



Entering the terminal states $a3$ and $b3$ gives a reward of 10 and a penalty of -10 respectively.

Write out the Bellman equations for the utility V^* of states for the optimal policy

1. when actions are deterministic
2. when actions are executed correctly in 70% of the cases, and a random other action is performed in other cases (bumping into wall results in staying in same state) using a discount factor $\gamma = 0.9$.

Solution

One optimal policy[1] for this environment is $b1 \rightarrow a1 \rightarrow a2 \rightarrow a3$ and $b2 \rightarrow a2 \rightarrow a3$. If we know the optimal policy then we can find the utility V^* of each state by solving a set of linear equations:

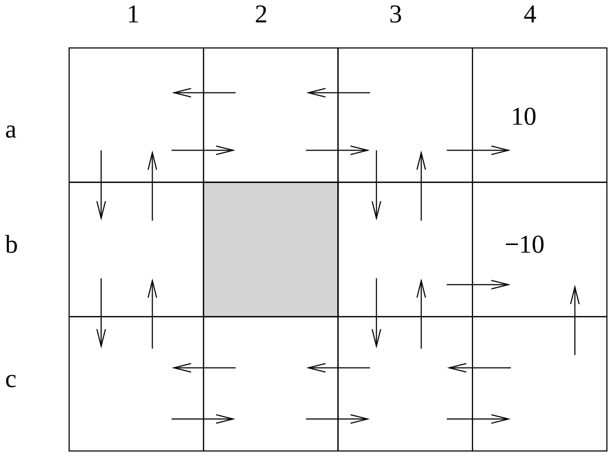
$$V^*(s) \quad = \quad \sum_{s' \in \mathcal{S}} P_\delta(s, \pi(s), s') \left[r(s') + \gamma V^*(s') \right]$$

1. In the deterministic case this becomes:
   $V^*(a1) = \gamma \cdot V^*(a2)$
   $V^*(a2) = 10$
   $V^*(b1) = \gamma \cdot V^*(a1)$
   $V^*(b2) = \gamma \cdot V^*(a2)$
2. For the stochastic actions the equations become:
   $V^*(a1) = \gamma \cdot (0.7V^*(a2) + 0.1V^*(b1) + 0.2V^*(a1))$
   $V^*(a2) = 0.7 \cdot 10 + \gamma \cdot (0.1V^*(b2) + 0.1V^*(a1) + 0.1V^*(a2))$
   $V^*(b1) = \gamma \cdot (0.7V^*(a1) + 0.1V^*(b2) + 0.2V^*(b1))$
   $V^*(b2) = 0.1 \cdot (-10) + \gamma \cdot (0.7V^*(a2) + 0.1V^*(b1) + 0.1V^*(b2))$

Notice that $V^*(a3) = V^*(b3) = 0$.

4.5 Q-Learning

Simulate Q-learning for a robot walking around in the following environment (b2 is a wall, entering b4 gives a penalty of -10, entering a4 gives a reward of 10).



Indicate Q-values after the following episodes, assuming the Q-values are initialized to 0 and using the "back-propagated" Q update rule (i.e. after getting in a goal state, updating the Q values in reverse order from goal to start, $\gamma = 0.9$).

1) c2,c1,b1,a1,a2,a3,a4

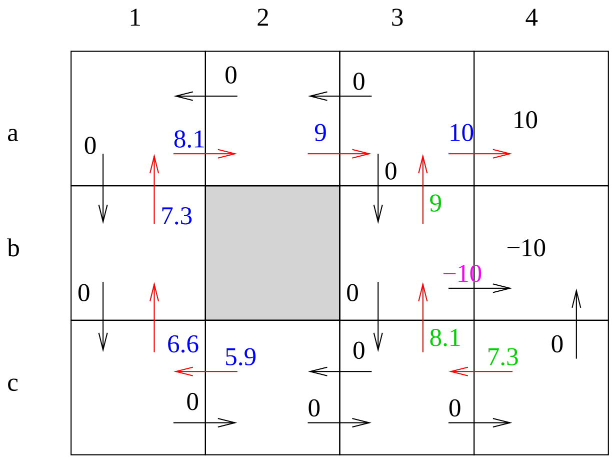
2) a1,a2,a3,b3,b4

[1] This is in fact the unique optimal policy for the case of nondeterministic actions.

3) c4,c3,b3,a3,a4

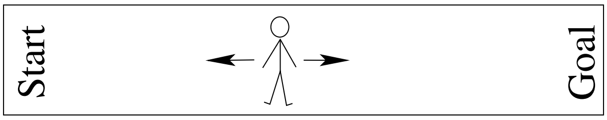
Assume the robot will now use the policy of always performing the action having the greatest Q value. Indicate this policy on the drawing. Is it optimal?

Solution



This policy is not optimal, since Q-learning has not yet converged given just these three episodes. Adding an episode c2, c3,b3, a3, a4 would result a Q-value of 7.3 to 'go right' in c2, and this will result in a different policy.

4.6 Reinforcement Learning with Generalization



Consider the reinforcement learning problem shown above. The hallway is 10 meters long. The agent starts at the leftmost end of the hallway and receives a positive reward when it reaches the rightmost end of the hallway.

The position of the agent (i.e. its state) is described by a real number $\in [0 : 10]$ indicating its distance (in meters) from the leftmost end of the hallway (position 0). This kind of representation is known as a continuous state-space and it does not allow a straightforward use of a table-based representation of the Q-function.

Suppose the agent uses a step-size 1 and uses a discount factor $\gamma = 0.9$. Using TD(1), we have accurately estimated Q^*-values for 4 state-action pairs.

Example	State	Action	Q-Value
1	2	$\leftarrow$	4.30
2	9	$\rightarrow$	10
3	5	$\rightarrow$	6.56
4	5	$\leftarrow$	5.90

1. Use linear regression to generalize the estimated values to all possible positions in the hallway. (Hint: Use a separate function for each of the two available actions.)
2. Using these functions, compute (and interpret) the preferred action in both state "8" and state "1".

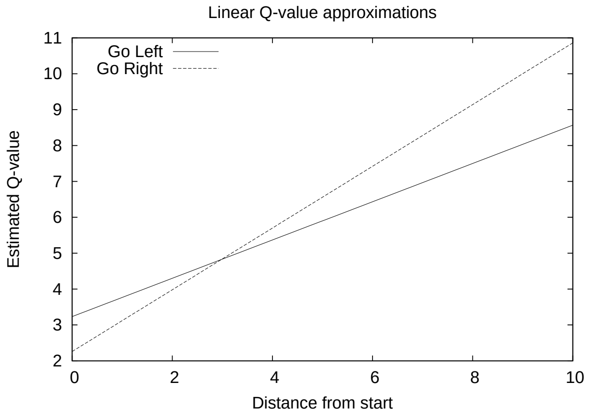
(Hint: For those who don't remember anything about high school math, you can also simply draw the functions.)

Solution

1. The equation for a straight line through two points (x_1, y_1) and (x_2, y_2) is

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

   This gives:
   For ←: $q = 4.3 + \frac{1.6}{3}(x - 2) = 0.533x + 3.233$
   For →: $q = 6.56 + \frac{3.44}{4}(x - 5) = 0.86x + 2.26$
   The resulting functions are shown below



2. Using the equations above
   $\hat{Q}(8, \leftarrow) = 7.497$ and $\hat{Q}(8, \rightarrow) = 9.14$ so the chosen action is →.
   $\hat{Q}(1, \leftarrow) = 3.766$ and $\hat{Q}(1, \rightarrow) = 3.12$ so the chosen action is ←, which is not optimal.

   This shows that the inductive bias of linear regression, i.e. the target function can be approximated by a straight line, is not suited to approximate a Q-function which, by definition, shows an exponential decay. More care is needed to choose an appropriate regression format for Q-learning.

4.7 Using SAMIAM: Homework

SAMIAM is a tool to easily design and query Bayesian networks using a graphical interface. You can freely download it at http://reasoning.cs.ucla.edu/samiam/. You can use it to gain more insight in the workings of Bayesian networks by reproducing within SAMIAM the networks of exercises 4.1 to 4.3.

1. Draw the networks of those three exercises, enter the probability tables, and see what results you obtain when making predictions.

2. Play around by modifying the probability tables (pay attention to the restrictions applying to probabilities and probability tables). How do the changes your make affect the predictions of your networks? Do all changes have the same impacts for the two networks of exercise 4.3.2?