Assignment9 – Learning

Given: June 22 Due: July 2

Problem 9.1 (Relevance-Based Learning) 40 pt

Some people with different attributes go sunbathing, the result class is whether they get sunburned:

#	Hair	Height	Weight	Lotion	Sunburned
1	Blonde	Short	Light	No	Yes
2	Blonde	Short	Average	Yes	No
3	Brown	Short	Average	Yes	No
4	Blonde	Short	Average	No	Yes
5	Blonde	Tall	Heavy	No	Yes
6	Brown	Tall	Heavy	No	No

In order to use relevance-based decision-tree learning, we model these examples in first-order logic.

1. Explain which predicates are needed. For each predicate give the arity and the domain of each argument. Give the formal representation of the description and the classification of Example 1.
2. Explain whether the determinations Height$\succ$ Weight and Weight $\succ$ Height hold or do not hold.
3. Give the minimal consistent determination of the class by a subset of the attributes. Give the formal representation of the rule learned from that determination.

Solution:

1. We use a unary predicate for each boolean attribute/class and a binary predicate for the others. The first argument of all predicates represents the example and has domain $\{1,\dots,6\}$. The domain of the second argument of the binary predicates is the set of possible values. Thus, we have:
   - $Hair(e,h)$ for $h \in \{Blonde, Brown\}$
   - $Height(e,h)$ for $h \in \{Short, Tall\}$
   - $Weight(e,w)$ for $w \in \{Light, Average, Heavy\}$
   - $Lotion(e)$
   - $Sunburned(e)$

   The description of Example 1 is $Hair(1,Blonde) \wedge Height(1,Short) \wedge Weight(1,Light) \wedge \neg Lotion(1)$. Its classification is $Sunburned(1)$.
2. Height $\succ$ Weight does not hold: Examples 1 and 2 agree on Height but not on Weight. Weight $\succ$ Height holds: any pair of examples that agree on Weight also agree on Height.
3. The minimal consistent determination is $Hair \wedge Lotion \succ Sunburned$. The learned rule is $\forall e, e'. \forall c. Hair(e,c) \wedge Hair(e',c) \wedge (Lotion(e) \Leftrightarrow Lotion(e')) \Rightarrow (Sunburned(e) \Leftrightarrow Sunburned(e'))$. Instantiating with $e' = 1$ and $c = Blonde$, that yields in particular $\forall e. Hair(e,Blonde) \wedge \neg Lotion(e) \Rightarrow Sunburned(e)$.

Problem 9.2 (Inductive Learning) 40 pt

Consider the family tree given by the following relations:

couple	children
A, B	E, F
C, D	G, H
F, G	I, J

Assume we already know the predicate $\texttt{par}(x, y)$ for x being a parent of y.

Our goal is to learn the predicate $\texttt{gp}(x, y)$ for x being a grandparent of y. That means to find a formula D such that $\forall x, y.\texttt{gp}(x, y) \Leftrightarrow D(x, y)$.

We do not know D, but we have the following examples for gp:

person-pair	grandparent
A, I	yes
B, I	yes
A, J	yes
A, E	no
A, F	no
F, A	no
A, A	no
C, J	yes
D, H	no
I, A	no

1. Give the intended formula D_1, i.e., the correct definition of grandparent.
2. Give a formula D_2 that covers exactly the positive examples.
3. Explain the pros and cons of learning the formula D as D_1 vs. D_2.
4. We want to learn algorithmically the formula $D(x, y) = \exists u_1, \dots, u_l.L_1 \wedge \dots \wedge L_k$ where each L_i is a literal of the form $P(x, y, u_1, \dots, u_l)$ or $\neg P(x, y, u_1, \dots, u_l)$ for some predicate symbol P including the equality predicate, i.e., $P \in \{\texttt{par}, \texttt{gp}, =\}$. We do so by building the set $\{L_1, \dots, L_k\}$ of literals gradually (with the understanding that each L_i can have any free variables in addition to x and y, which we will collect at the end as the u_i).
   (a) If we start with the empty set of literals, give all useful choices that we can make for the first literal.
   (b) For each choice L that is non-recursive (i.e., P is not the target predicate gp), positive (i.e., the literal is not negated), not an equality (i.e., P is not the equality predicate $=$), and does not introduce a new variable (i.e., only uses x and y), which examples are falsely classified?

Solution:

1. $D_1(x, y) = \exists u.\texttt{par}(x, u) \wedge \texttt{par}(u, y)$
2. $D_2(x, y) = (x = A \wedge y = I) \vee (x = B \wedge y = I) \vee$ and so on for all positive examples.
3. pros of D_1:

- D_2 captures only the provided examples. If that list is incomplete, it is likely the wrong formula. Even if the list is complete, D would have to be changed every time the list of examples changes. D_1 captures all future examples correctly.
- D_1 has size $O(1)$, whereas D_2 has size $O(n)$ where n is the number of examples.

pros of D_2:

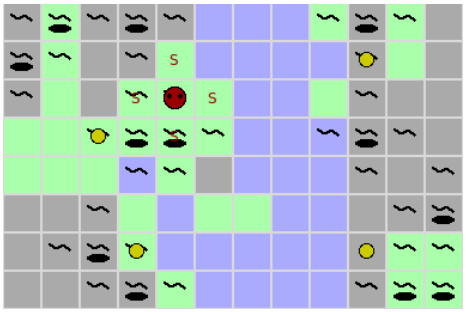
- D_2 can be easily read off the list of examples in $O(n)$ time.
- A nice formula for D_1 might not even exist or might be very difficult to find or might be impossible to find from the given examples.

4. (a) There are 16 choices each for $P \in \{\texttt{par}, \texttt{gp}\}$: $P(x,x)$, $P(y,y)$ $P(x,y)$ or $P(y,x)$ (no new variables), $P(x,u)$, $P(u,x)$, $P(y,u)$, $P(u,y)$ (1 new variable); as well as the negated versions. (It is useless to add $P(u,v)$ or $P(u,u)$ where all variables are new.)
   Additionally, we can choose $x = y$ and $\neg x = y$. (It is useless to add an (in)equality with a new variable or where both variables are the same.)
   (b) There are 4 such choices:
   - $\texttt{par}(x,x)$: no false-positives
   - $\texttt{par}(y,y)$: no false-positives
   - $\texttt{par}(x,y)$: false-positives are (A,E), (A,F), and (D,H)
   - $\texttt{par}(y,x)$: false-positives are (F,A)

   Additionally, for all 4 choices, all positive examples are false-negatives.

Problem 9.3 (Competition: FAULumpus) 200 pt

Implement an agent that explores the FAULumpus world – a variant of the Wumpus World localized to Erlangen and the FAU. The FAULumpus world is on a 12×8 grid. You can explore any square that is adjacent to a square you have already explored. For example, if you have explored squares $(0,0)$, $(0,1)$, $(1,0)$, you could next explore the square $(0,2)$. On some of the squares you can find sights. Your goal is to find as many sights as possible without dying. If you find n sights and then stop the exploration, you get n^2 points. But if your agent dies, you get 0 points. Your agent dies if it steps onto a square with the Wumpus or if it steps onto a square that has a pit. As a warning sign, there is a smell or a breeze on the neighbouring squares.



You can compete with your agent on our server. Your agent's evaluation will be the maximum average score of $1,000$ consecutive games.

Since we will not run your agents ourselves, you may use any libraries and programming languages you want. Further details about APIs and protocols will be published in the forum.

On StudOn, please **submit by September 30**:

- your agent's name and password
- the code for your agent
- a short description of how your agent works ($\approxeq$ 1 page)

Your grade will be based on how well your agent performed on the server. Currently, we are planning for something along these lines:

- You get 50 points, if the average score is above 1
- You get 90 points, if the average score is above 2
- You get 100 points, if the average score is above 3
- Additionally, the 10 best agents will receive up to 100 bonus points on top of that, based on the placement
- You will get fewer points if the description is missing or unintelligible

Important: This is the second time that we run a competition in this manner. Therefore, it is possible that we may have to change the setup or evaluation/grading criteria a little if things go wrong. In particular, we do not know how stable the server will be and there may be down times etc. For the same reason, please avoid any excessive load on the server.