CS781 Practice Problem Set 2 (Autumn 2024)

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***

1. A binary decision tree on a set $\mathcal{F}$ of features is a tree-structured ML model, in which every non-leaf node is labeled by a predicate on features in $\mathcal{F}$, and every leaf node is labeled by a decision/outcome. The edges from a non-leaf node n to its children are labeled by 1 or 0, corresponding to the result of evaluating the predicate labeling n. Given an input, i.e. valuation of all features in $\mathcal{F}$, the predicition of a binary decision tree is the label of the leaf reached by following the unique path from the root determined by evaluating the predicates labeling the nodes. For a given set $\mathcal{D}$ of labeled data, an *optimal binary decision tree* is the smallest binary decision tree (in terms of count of non-leaf nodes) that predicts the correct decision for all (inputs of) data points in $\mathcal{D}$.

   Consider the following data for which we wish to construct an optimal binary decision tree. The table below lists values of three binary features f_1, f_2 and f_3 and the prediction l for the corresponding point in the feature space.

f_1	f_2	f_3	l
0	0	0	1
1	1	1	0
1	1	0	1
0	1	1	0

   We wish to design an optimal binary decision tree for this data with a single decision node labeled by one of the features, and two leaf nodes labeled by 0 and 1.

   Construct an MILP formula using as few variables as you can, such that a satisfying assignment of the formula gives an optimal decision tree with a single decision node, as desired. You must clearly indicate the meaning of all variables used in your formula.

2. Recall the abduction based algorithm studied in class for finding explanations. We wish to apply this algorithm to a complete binary decision tree with two layers of decision nodes that is used to classify the data shown in the table in the previous question. The root node of the tree is labeled f_1 and all children of the root are labeled f_3. Our goal in this question is to find an abduction based explanation of row 3 of the table shown in the previous question (i.e. $f_1 f_2 f_3 = 110$ and $l = 1$).

   Find the abductive explanation obtained by applying the algorithm studied in class to the above binary decision tree for row 3. Show all steps clearly.

3. In this question, we will try to construct optimal *binary decision diagrams*, which are like binary decision trees (i.e. every non-leaf node is labeled by a feature, and has a left child and a right child), except that the final model can be a rooted DAG and does not necessarily have to be a tree. Figure 1 shows an example binary decision tree and a binary decision diagram for the same data set (shown in sub-question (b)).

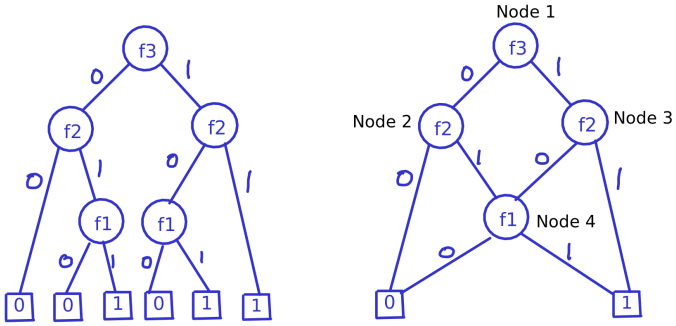

Figure 1: Decision tree and decision diagram

(a) Show that for every $n > 0$, there exists a data set over n binary features $f_1, f_2, \ldots f_n$ and a single binary decision, such that the smallest binary decision tree for the data set has $\Omega(2^n)$ decision (or internal nodes), while the smallest binary decision diagram for the same data has $O(n)$ decision nodes.

This shows the advantage of binary decision diagrams over binary decision trees.

(b) Consider the data shown in the following table, where f_1, f_2, f_3 are binary features and d is a binary decision.

Data point	f_1	f_2	f_3	d
1	0	0	1	0
2	0	1	0	0
3	1	0	1	1
4	1	1	0	1
5	1	0	0	0
6	0	1	1	1

We wish to find an optimal decision diagram with 4 decision nodes (and two leaves labeled 0 and 1) for this data set by solving a system of constraints. Assume that

- Nodes of the decision diagram are numbered from 1 through 4, with 1 being the root.
- All non-leaf children of a node are numbered higher than the node itself.
- For every non-leaf node, if the feature labeling the node has value 0 for a given data point, then we move to the left child of the node en-route to a leaf to find the decision for the data point. Otherwise, if the feature value is 1, we move to the right child of the corresponding node.
- A node i is *truthful* for data point j iff starting from node i and reading the values of features in data point j, we can reach a leaf node labeled by the decision corresponding to data point j.

We use the following variables to construct a system of constraints such that finding a satisfying assignment of the constraints yields a decision diagram for the given data set.

Variable	True iff ...
$a_{i,j}$	Node i is labeled by feature f_j, $1 \leq i \leq 4$, $1 \leq j \leq 3$
$l_{i,j}$	Node j is the left child of node i, $1 \leq i \leq 3$, $i < j \leq 4$
$r_{i,j}$	Node j is the right child of node i, $1 \leq i \leq 3$, $i < j \leq 4$
$l0_i$	Leaf labeled 0 is the left child of node i, $1 \leq i \leq 4$
$l1_i$	Leaf labeled 1 is the left child of node i, $1 \leq i \leq 4$
$r0_i$	Leaf labeled 0 is the right child of node i, $1 \leq i \leq 4$
$r1_i$	Leaf labeled 1 is the right child of node i, $1 \leq i \leq 4$
$t_{i,j}$	Node i is truthful for data point j, $1 \leq i \leq 4$, $1 \leq j \leq 6$

(a) For the decision diagram shown in Fig. 1, list the values of all variables described above.

(b) Write propositional formulas in terms of the above variables to encode each of the following constraints:

i. Every non-leaf node i has two distinct children.

ii. Every non-root and non-leaf node i has at least one parent.

iii. For every non-leaf node i and for every data point j, the node is truthful for data point j iff the child of i reached on the value of feature labeling node i is also truthful for data point j.

iv. For every distinct non-leaf nodes i and j labeled by the same feature, there is at least one data point on which either node i is truthful but node j is not, or vice versa.

(c) What additional constraints would you add to those obtained above to ensure that a satisfying assignment necessarily gives an *optimal (i.e. smallest number of nodes)* binary decision diagram for a given data set. Given reasons in support of your answer. Answers without reasons will fetch no marks.

4. Recall the technique we learnt in class to find abductive explanations for classification decisions taken by a neural network. For a network N modeled by a first order formula φ_N, let $\mathbf{X}_0$ denote the vector of inputs, $\mathbf{X}_i$ denote the vector of outputs of neurons of the i^{th} layer (not being the output layer), and let $\mathbf{Y}$ denote the vector of outputs. Let us denote the abductive explanation finding algorithm as Abd. Algorithm Abd takes as inputs the following:

- A formula $\varphi_N(\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{Y})$ encoding the relation between inputs $\mathbf{X}_0$ and outputs $\mathbf{Y}$ of the network (along with the outputs of all intermediate neurons)
- An assignment σ of the inputs $\mathbf{X}_0$
- An assignment ρ of the outputs $\mathbf{Y}$ such that the network N on feeding the input σ outputs ρ. Technically, this can be found by evaluating N on σ, but we'll take this as an input to the algorithm because we wish to use this algorithm for some other purpose as explained below.

Algorithm Abd outputs a (subset-) minimal sub-assignment $\widetilde{\sigma}$ of the inputs $\mathbf{X}_0$ that suffices to generate the same output ρ when fed into the network N.

In this question, we wish to use algorithm Abd to design another algorithm that finds minimal adversarial examples for a given input σ. Our goal therefore is to find another image σ' that differs (subset-)minimally from σ such that the output of the network N when fed σ' as input is different from ρ.

Indicate clearly how you will use Abd to design an algorithm Adv for the above purpose. You must clearly describe your algorithm as a pseudocode or flow-chart, and provide justification of why it finds minimally changed adversarial counterexamples.

The algorithm Adv must take as inputs the formula φ_N, σ and an output ρ' (that needn't be the same as the output of N on being fed σ as input) , and it must output an image σ' that is (subset-)minimally different from σ such that the output of N on being fed σ' as input is different from ρ.

Hint: You can use two copies of φ_N (modeling two copies of the network N), one of which is fed σ as input and the other is fed σ' as input. You can also create a new output that evaluates to 1 if the output of N with σ as input matches the output of N with σ' as input, and evaluates to 0 otherwise.

5. In a reinforcement learning setting, suppose the environment is modeled by a Markov Decision Process (MDP) $\mathcal{M} = (S, s_0, A, P, R)$, where the set of states $S = \{s_0, s_1, s_2\}$, the initial state is s_0, the set of agent actions is $A = \{a, b\}$, the probabilistic transition function P is given by the following table, and the reward function R is unknown for now.

	s_0	s_1	s_2
s_0	$P_a = 0.0$, $P_b = 0.0$	$P_a = 0.5$, $P_b = 1.0$	$P_a = 0.5$, $P_b = 0.0$
s_1	$P_a = 0.3$, $P_b = 0.5$	$P_a = 0.0$, $P_b = 0.3$	$P_a = 0.7$, $P_b = 0.2$
s_2	$P_a = 0.2$, $P_b = 0.0$	$P_a = 0.0$, $P_b = 1.0$	$P_a = 0.8$, $P_b = 0.0$

The probabilistic transition function is read off the above table as follows: The entry "$P_a = 0.3$, $P_b = 0.5$" in row s_1 and column s_0 says that when the MDP is in state s_1, if it sees an agent action a, it transitions to state s_0 with probability 0.3. Similarly, if it sees an agent action b when in state s_1, then it transitions to state s_0 with probability 0.5. The interpretation of other entries in the table are similar.

Now suppose we use an MDP observer function (as described in the paper studied in class) $f : \{s_0, s_1, s_2\} \to \{x, y\}$ given by $f(s_0) = f(s_2) = x$ and $f(s_1) = y$.

(a) Construct a non-deterministic safety automaton $\varphi^{\mathcal{M}}$ with three safe states and one unsafe state (representing spurious or unexpected behaviour of the environment, as discussed in class) to abstract the behaviour of the above MDP. Your automaton $\varphi^{\mathcal{M}}$ should have $\{(a, x), (a, y), (b, x), (b, y)\}$, i.e. $A \times f(S)$, as its input alphabet. Furthemore, for every infinite trace of states $q_0 q_1, \ldots$ of the MDP where $q_0 = s_0$ and where the corresponding action sequence is $a_0 a_1, \ldots$, every run of $\varphi^{\mathcal{M}}$ on the infinite word $(a_0, f(q_0))$, $(a_1, f(q_1))$, $\ldots$ must always stay within the safe states of $\varphi^{\mathcal{M}}$. Additionally, for every sequence of states and actions $q_0 a_0 q_1 a_1 \ldots$ that *does not* correspond to a trace of the MDP $\mathcal{M}$, the corresponding run of $\varphi^{\mathcal{M}}$ on the infinite word $(a_0, f(q_0))$, $(a_1, f(q_1))$, $\ldots$ must necessarily pass through the unsafe state.

(b) Consider a safety specification that requires that no two consecutive observations from the MDP must be identical. Recall, the observer function f has $\{x, y\}$ in its range. Moreover $f(s_0) = x$. Therefore, the safety specification effectively requires that the observed sequence of states of the environment should be $xyxyxy\ldots$.

Indicate whether it is possible to design a shield in this case to ensure the above safety specification. If your answer is in the negative, give reasons for the same. Otherwise, give a shield along with justification for why it ensures that the above safety property is never violated.

[Hint: It is possible to solve this problem without going through all the steps of the game construction studied in class.]

6. Recall our study of Pareto-optimal interpretations of black-box ML models. We had used an accuracy score c and an explainability score e for each interpretation. Now suppose there is a third score, say f, associated with each interpretation, and we wish to find interpretations with Pareto-optimal (c, e, f) values.

(a) Show that an interpretation with largest value of $c + e + f$ has a Pareto-optimal (c, e, f) value. *[Hint: The argument is similar to that used in the (c, e) case.]*

(b) Consider the algorithm `FindAllPO` shown below. Here, `FindAllPO(S, f_l, f_u)` takes a set `S` (possibly empty) of already computed Pareto-optimal triples and tries to add new triples (c, e, f) to S such that (c, e, f) is Pareto-optimal in the region corresponding to $f_l < f < f_u$. Assume that `FindOnePO(f_l, f_u)` returns one Pareto-optimal triple $(c^\star, e^\star, f^\star)$ such that $f_l < f^\star < f_u$.

```
FindAllPO(S, f_l, f_u) {
  (c*, e*, f*) := FindOnePO(f_l, f_u);
  if no such (c*, e*, f*) exists or (c*, e*, f*) dominated by a point in S, return S;
  S := S union {(c*, e*, f*)};
  S := FindAllPO(S, f_l, f*);
  S := FindAllPO(S, f*, f_u);
  return S;
}
```

Assume initially, `FindAllPO` is called with $S = \emptyset$, f_l slightly less than the smallest possible value of f and f_u slightly more than thte largest possible values of f.

Show that the above algorithm may fail to compute all Pareto-optimal triples (c, e, f). You must give justification/counterexample to support your answer.

7. Recall the paper on abduction-based minimal explanation finding for white-box neural networks. Suppose we apply it to the network shown in Fig. 2, with inputs $(x_1, x_2) = \{(1, 1), (2, 2), (1, 2), (2, 1)\}$. **Assume that nodes in the hidden layer have ReLU activation functions, while nodes in the input and output layers don't have ReLU.** Indicate the explanation obtained for $x_5 = 0$ using this algorithm. You must clearly show all your steps.

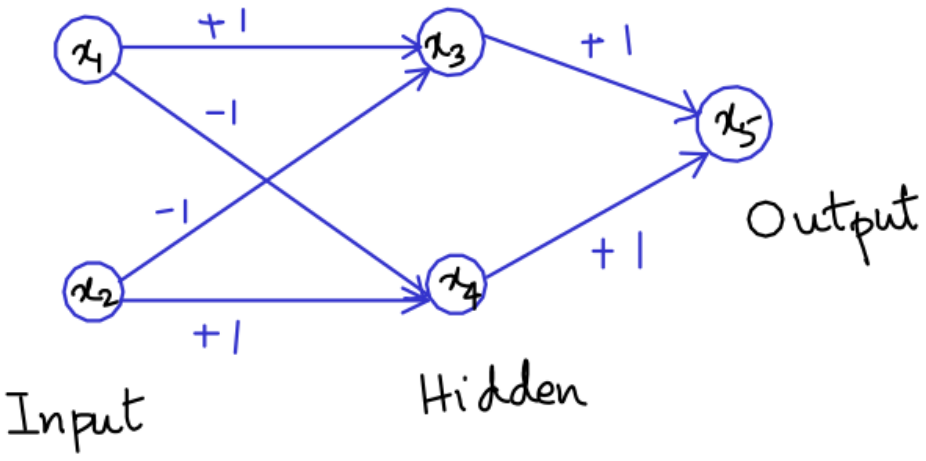

Figure 2: Simple neural network

8. In our study of Pareto-optimal (PO) interpretations of black-box ML models, we used two quantitative scores, Δ_c and Δ_e, for correctness and explainability of an intepretation. Suppose there is a third X-

factor that also affects the desirability of an interpretation, but is incomparable with correctness and explanability. Let $\Delta_X(E)$ denote this X-factor score of a candidate interpretation E.

A student wants to re-use the SYNPLICATE algorithm studied in class to find Pareto-optimal interpretations of black-box ML models as follows.

```
1. Find all PO points using Synplicate, where Delta_x and
   Delta_c are combined as (Delta_x + Delta_c) to give a
   single "augmented correctness" score.
2. For each PO point (x+c, e) thus obtained,
    a. Add (Delta_e = e) to the constraint system
    b. Find all PO points using Delta_c as correctness
       measure and Delta_x as explanability measure.
```

Explain with clear justification whether the student can find all 3-dimensional Pareto points using the above scheme. If you think the answer is in the negative, you must give a (counter-)example that shows where the student's scheme goes wrong. If you think the answer is in the positive, you must give clear arguments why the student's scheme can find all 3-dimensional PO points.

9. Consider the following grid of 0-1 valued inputs of a deep neural network.

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$

In the following, we specify a concrete input by providing only the subscripts of input variables in the above grid that evaluate to 1. Thus, the concrete input represented by $\{(1,2),(3,1),(0,2)\}$ is the one where $x_{1,2} = x_{3,1} = x_{0,2} = 1$ and all other $x_{i,j}$'s in the grid above are 0.

The deep neural network has come up with a binary classification of inputs as given in the following table. Note that not the classification of all concrete inputs is not known. In other words, there are some inputs, for which we don't know what the classification result is.

Sr.No.	Concrete input	Classification
1	$\{(0,1),(1,2),(2,3)\}$	1
2	$\{(0,1),(2,1),(3,1)\}$	0
3	$\{(0,0),(1,1),(2,2),(3,3)\}$	1
4	$\{(0,3),(1,2),(2,1),(2,2),(3,0)\}$	1
5	$\{(0,3),(2,3),(1,1),(3,2),(3,3)\}$	0
6	$\{(0,0),(1,2),(1,1),(3,3)\}$	0
7.	$\{(1,1),(2,2),(3,3)\}$	1
8.	$\{(0,0),(1,1),(2,2),(3,3),(0,2),(1,3)\}$	0

We wish to find an abduction-based explanation of the classification result in row 3 (i.e. Sr. No. 3) of the above table.

(a) Using the hitting set technique studied in class, find two minimal explanations of the form $x_{i,j} \wedge \neg x_{i',j'} \wedge \cdots$. In other words, your explanation must list the minimal set of literals at various grid-points whose conjunction is responsible for the classification result.

(b) *[5 marks]* Augment the classification in the table given above minimally, i.e. add as few new classification rows as possible, such that the minimal explanation of row 3 has sixteen literals.

10. This question concerns the synthesis of shields for safe reinforcement learning. Consider a safety automaton as specified below. All transitions that are not indicated below are assumed to go to the safety violating state.

Present state	(env state, agent action)	Possible set of next states
s_0	(x_0, y_0)	$\{s_0, s_1, s_2\}$
s_0	(x_0, y_1)	$\{s_1\}$
s_0	(x_1, y_0)	$\{s_0, s_1\}$
s_0	(x_1, y_1)	$\{s_1, s_2\}$
s_1	$(x_0, *)$	$\{s_0\}$
s_1	(x_1, y_0)	$\{s_0, s_1\}$
s_2	(x_1, y_1)	$\{s_0, s_2\}$

The non-deterministic automaton abstracting the MDP representing the environment is given by the following table. All transitions that are not indicated below are assumed to go to an abstract state that is not of interest to us. Once the environment goes to such a state, it is assumed to stay forever in that state.

Present state	(env state, agent action)	Possible set of next states
e_0	(x_0, y_0)	$\{e_1, e_2\}$
e_0	(x_0, y_1)	$\{e_0, e_1\}$
e_0	(x_1, y_1)	$\{e_2\}$
e_0	(x_1, y_0)	$\{e_1\}$
e_1	(x_0, y_1)	$\{e_0, e_1\}$
e_1	(x_1, y_1)	$\{e_2\}$
e_2	(x_0, y_1)	$\{e_0, e_2\}$
e_2	(x_0, y_0)	$\{e_0, e_2\}$

You are required to modify the above abstraction of the MDP minimally, i.e. change as few rows as possible to the above table, such that it is possible to synthesize a safety shield.