CS781 Quiz 2 (Autumn 2025)

Max marks: 20 Duration: 60 mins Roll No.

- *You are required to answer each question only in the space provided with each question.*
- *Only material written within the allotted answering space for each question will be graded.*
- *The spaces allotted for answering questions should give a rough indication of the relative lengths of correct answers to the questions.*
- *Please attach all your rough sheets.*
- *The exam is open book and notes. However, you are not allowed to search on the internet or consult others over the internet for your answers.*
- *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.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***

1. You are given the following perfectly classifiable dataset, where f_1, f_2, f_3, f_4 are Boolean features and ℓ is a classification label.

f_1	f_2	f_3	f_4	ℓ	
0	1	1	0	A	— r_1
1	0	1	1	B	— r_2
0	1	0	0	A	— r_3
0	1	1	1	B	— r_4
1	0	0	1	C	— r_5
1	1	1	1	C	— r_6

r_i refers to data instance (or row) i

(a) *[2 × 3 = 6 marks]* We wish to construct optimal decision sets for explaining the above dataset using a reduction to weighted MaxSAT. Using the technique studied in class, construct an instance of weighted MaxSAT for any two of three class labels A, B and C. Thus, there should be two weighted MaxSAT instances as part of your solution.

Clearly identify what each variable in your weighted MaxSAT instances correspond to. Use the sum of count of rules and count of literals as the measure of the size of a decision set.

Following the paper studied in class, for label A, the hard clauses correspond to:

$(\neg P_1 \vee \neg n_1) \wedge (\neg P_2 \vee \neg n_2) \wedge (\neg P_3 \vee \neg n_3) \wedge (\neg P_4 \vee \neg n_4)$

$\wedge$

for r_2 ----- $(n_1 \vee P_2 \vee n_3 \vee n_4) \wedge$

for r_4 ----- $(P_1 \vee n_2 \vee n_3 \vee n_4) \wedge$

for r_5 ----- $(n_1 \vee P_2 \vee P_3 \vee n_4) \wedge$

for r_6 ----- $(n_1 \vee n_2 \vee n_3 \vee n_4) \wedge$

for r_1 ----- $[t_1 \leftrightarrow \neg(P_1 \vee n_2 \vee n_3 \vee P_4)] \wedge$

for r_3 ----- $[t_3 \leftrightarrow \neg(P_1 \vee n_2 \vee P_3 \vee P_4)]$

$\wedge \; (t_1 \vee t_3)$

Soft clauses : $\neg p_1, \neg n_1, \neg p_2, \neg n_2, \neg p_3, \neg n_3, \neg p_4, \neg n_4$

(weight 1 for each soft clause.)

Interpretation of variables :

p_i, n_i : dual-rail encoding of feature f_i

t_i : data instance r_i is covered

The above encoding doesn't necessarily minimize count of literals (think why?) To minimize # literals, we can add the foll. to the hard clauses:

$$(u_1 \leftrightarrow (\neg p_1 \wedge \neg n_1)) \wedge (u_2 \leftrightarrow (\neg p_2 \wedge \neg n_2)) \wedge$$
$$(u_3 \leftrightarrow (\neg p_3 \wedge \neg n_3)) \wedge (u_4 \leftrightarrow (\neg p_4 \wedge \neg n_4))$$

and use the following for soft clauses :

Soft clauses : $u_1, u_2, u_3, u_4, t_1, t_3$

with weight 1 for each soft clause

For label B, following similar reasoning, we get hard clauses corresponding to:

$(\neg p_1 \vee \neg n_1) \wedge (\neg p_2 \vee \neg n_2) \wedge (\neg p_3 \vee \neg n_3) \wedge (\neg p_4 \vee \neg n_4)$ for r_2

$\wedge$

for r_1 ----- $(p_1 \vee n_2 \vee n_3 \vee p_4) \wedge$ $[t_2 \leftrightarrow \neg(n_1 \vee p_2 \vee n_3 \vee n_4)] \wedge$

for r_3 ---- $(p_1 \vee n_2 \vee p_3 \vee p_4) \wedge$ $[t_4 \leftrightarrow \neg(p_1 \vee n_2 \vee n_3 \vee n_4)]$

for r_5 ---- $(n_1 \vee p_2 \vee p_3 \vee n_4) \wedge$ $\wedge (t_2 \vee t_4) \wedge$ for r_4

for r_6 ----- $(n_1 \vee n_2 \vee n_3 \vee n_4) \wedge$

$$(u_1 \leftrightarrow (\neg p_1 \wedge \neg n_1)) \wedge (u_2 \leftrightarrow (\neg p_2 \wedge \neg n_2)) \wedge$$
$$(u_3 \leftrightarrow (\neg p_3 \wedge \neg n_3)) \wedge (u_4 \leftrightarrow (\neg p_4 \wedge \neg n_4))$$

Soft clauses : $u_1, u_2, u_3, u_4, t_2, t_4$ with each having wt. 1.

Interpretation of u_i : feature f_i is unused in the decision rule.

For label c, the reasoning is similar, and you should be able to write out the MaxSAT instance.

(b) *[2 × 2 = 4 marks]* Find a solution to each of the above weighted MaxSAT instances constructed by you, and give the corresponding decision sets for the two labels chosen by you.

For label A:

MaxSAT Solution: $n_4 = 1,\ p_4 = n_1 = p_1 = n_2 = p_2 = n_3 = p_3 = 0$

$t_1 = 1,\ t_3 = 1,\ u_1 = u_2 = u_3 = 1,\ u_4 = 0$

Corresponding decision set has only one rule:

$\boxed{\text{If } \neg f_4 \text{ then } A}$

This rule covers both rows r_1 and r_3. Hence this is the entire decision set for A.

For label B:

MaxSAT solution:

$n_1 = p_1 = p_2 = n_3 = n_4 = p_4 = 0,\ n_2 = p_3 = 1$

$t_4 = 0, t_2 = 1,\ u_2 = u_3 = 0$

$u_1 = u_4 = 1$

Corresponding decision rule:

$\boxed{\text{If } \neg f_2 \wedge f_3 \text{ then } B}$

The above solu. doesn't cover r_4, so this is just one rule in the decision set.

To cover r_4, we need to re-solve the MaxSAT problem, looking for a different solution.

We now have the following solution:

$p_1 = n_2 = p_2 = n_3 = p_3 = n_4 = 0,\ n_1 = p_4 = 1$

$t_4 = 1,\ t_2 = 0,\ u_1 = u_4 = 0$

$u_2 = u_3 = 1$

Corresponding decision rule:

$\boxed{\text{If } \neg f_1 \wedge f_4 \text{ then } B}$

For label C:

Solution of MaxSAT problem (other solutions also possible)

$$n_1 = p_1 = p_2 = p_3 = n_4 = p_4 = 0$$

$$n_2 = n_3 = 1$$

$$t_5 = 1, t_6 = 0, u_1 = u_4 = 1, u_2 = u_3 = 0$$

Corresponding decision rule:

$$\boxed{\text{If } \neg f_2 \wedge \neg f_3 \text{ then } C}$$

This rule only covers r_5. To cover r_6, we need another solution for the MaxSAT problem. One such solution:

$$n_1 = n_2 = n_3 = p_3 = n_4 = p_4 = 0$$

$$p_1 = p_2 = 1$$

$$t_5 = 0, t_6 = 1, u_1 = u_2 = 0, u_3 = u_4 = 1$$

Corresponding decision rule:

$$\boxed{\text{If } f_1 \wedge f_2 \text{ then } C}$$

(c) *[5 marks]* Recall that in a decision set, the “if-then” rules are *unordered*. We now wish to obtain an optimal decision list, which is an *ordered* list of “if-then-else” rules.

An example decision list for explaining a (hypothetical) dataset with Boolean features `IsOld` and `IsFemale` and labels `likely`, `unlikely`, `certain` could be the following:

if (`IsOld` ∧ `IsFemale`) **then** `likely`
else if (`IsFemale`) **then** `unlikely`
else if (¬ `IsOld`) **then** `likely`
else `certain`

Thus, each condition in a decision list is a conjunction of feature values, and the corresponding decision is one of the allowed labels. The size of a decision list (much like we measure the size of a decision set) is the sum of count of rules (excluding the **else**) and the count of literals used in the various **if** and **else if** conditions. For example, the size of the above decision list is 3 (rules) +4 (2 literals in rule 1, 1 literal each in rules 2 and 3), i.e. 7.

Find a minimum sized decision list for the dataset given at the beginning of this problem. Give justification for why you think your solution is a minimum sized decision list.

Consider the dataset reproduced below.

f_1	f_2	f_3	f_4	ℓ	
0	1	1	0	A	--- r_1
1	0	1	1	B	
0	1	0	0	A	-- r_3
0	1	1	1	B	
1	0	0	1	C	
1	1	1	1	C	

Clearly, we have

If $\neg f_4$ then A

Note that once we have used the above rule, column f_4 is no longer useful (all rows other than r_1 & r_3 have same value = 1 in column f_4). Therefore, for subsequent rules, we can ignore col. f_4 & rows r_1, r_3.

This gives the following table:

f_1	f_2	f_3	f_4	ℓ	
~~0~~	~~1~~	~~1~~	~~0~~	~~A~~	
1	0	1	~~1~~	B	
~~0~~	~~1~~	~~0~~	~~0~~	~~A~~	
0	1	1	~~1~~	B	---- r_4
1	0	0	~~1~~	C	
1	1	1	~~1~~	C	

For this table, we can use:

else if $\neg f_1$ then B

Now deleting column f_1 and row r_4 (by same logic as above), we get the foll. table.

f_1	f_2	f_3	f_4	ℓ
~~0~~	~~1~~	~~1~~	~~0~~	~~A~~
~~1~~	0	1	~~1~~	B
~~0~~	~~1~~	~~0~~	~~0~~	~~A~~
~~0~~	~~1~~	~~1~~	~~1~~	~~B~~
~~1~~	0	0	~~1~~	C
~~1~~	1	1	~~1~~	C

For this, we can use.

else if $\neg f_2 \wedge f_1$ then B

else C

Size of decision list = 3 (rules) + 4 (lit.) = 7

Is 7 the smallest size for a decision list?

Note that with 3 labels, the best we can do is:

```
if ($lit_1$) then $label_1$
else if ($lit_2$) then $label_2$
else $label_3$
```

This has size 4
(2 rules + 2 lit.)

For this to work for our dataset, we must have $lit_1 = \neg f4$, $label_1 = A$ (follows from decision set solution in prev. part).

However, after this, we can't have a single lit_2 for $label_2$ (follows from decision sets for B and C).

$\therefore$ Min. size $\geq$ 2(rules) + 3 (lit) = 5.

We also know from the decision set solutions for B and C that a single rule doesn't suffice to cover all rows with label B or all rows with label C.

With an additional rule and with two literals in one of the rules, we have

```
if ($\neg f4$) then A
else if ($lit_1 \wedge lit_2$) then $label_2$
  else if ($lit_3$) then $label_3$
  else $label_4$.
```

This already has size 7 (3 rules + 4 literals)

2. *[10 marks]* This question concerns shielding in the context of reinforcement learning. Consider the abstraction automaton (of the MDP modeling the environment) shown in Fig. 1(a), and a supposed pre-emptive shield designed by a student, shown in Fig. 1(b). Assume that the actions of the agent come from $\{a_1, a_2, a_3\}$ and the observations of MDP states comes from $\{l_1, l_2, l_3\}$. The safety automaton is given in Fig. 1(c). All states shown in Fig. 1(c) are "safe" states; "unsafe" states are not shown for clarity. Similarly, all states shown in Fig. 1(a) are the "non-paradise" states; "paradise" states are not shown for clarity.

You are required to determine if the pre-emptive shield serves the purpose of ensuring safety in this example. A "Yes"/"No" answer will fetch no marks. You must give complete justification for your answer.

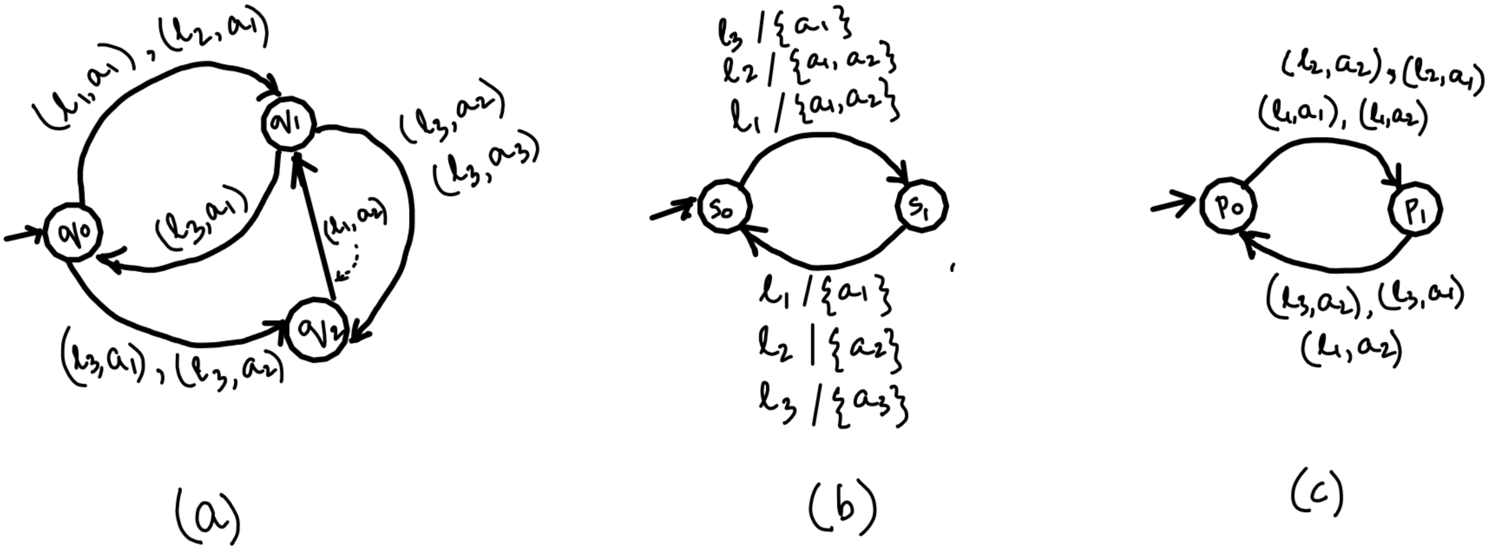
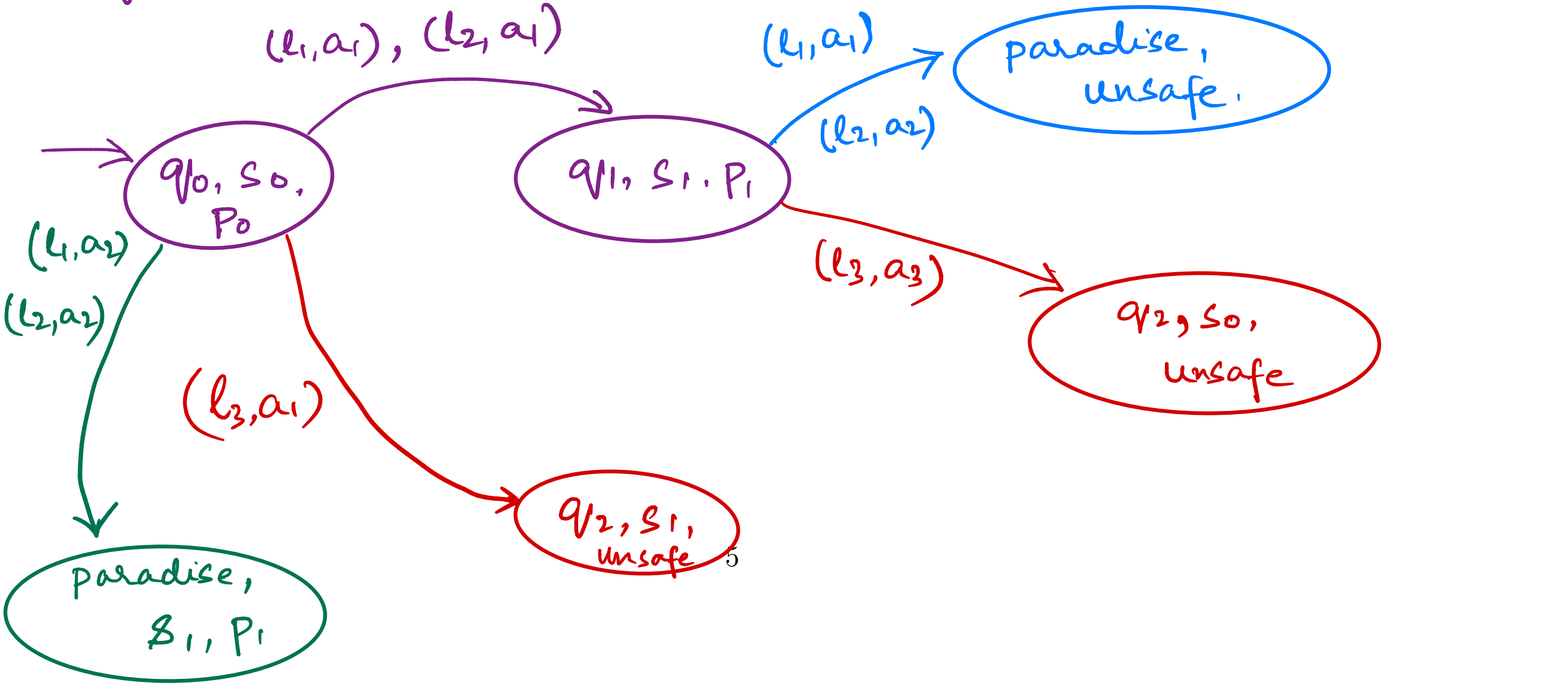


Figure 1: (a) Abstraction automaton, (b) Pre-emptive "shield", (c) Safety automaton

Taking the product of (a), (b), (c), i.e. env. + agent constrained by preemptive shield running in parallel with safety automaton:



Since there are paths in this product transition system from the initial state (q_0, s_0, p_0) to (non-paradise, s_i, unsafe) states, the shield doesn't ensure safety.

E.g., (l_3, a_1) or $(l_1, a_1), (l_3, a_3)$ takes us to unsafe states.