10601 Machine Learning HW1

Mingyang Song

Q1.

A robot is tasked with serving meals to a hospital patient. Every time the robot serves a meal, the patient says whether he liked the meal or not. Based on certain characteristics of the food, the robot must determine what type of food the patient will enjoy. There are five food characteristics that the robot is aware of: food temperature (either hot, warm, cool or cold), the state of the food (liquid, solid or mix), if it is greasy (yes or no), its caloric content (low, medium, or high), and whether the food is sweet (yes or no). So far, the robot has observed four meals:

Food Temperature	Food State	Greasy?	Caloric Content	Sweet?	Patient Enjoys Meal?
Warm	Solid	Yes	Medium	Yes	Yes
Warm	Solid	No	Medium	No	Yes
Cold	Mix	Yes	Medium	No	No
Cool	Liquid	Yes	Medium	Yes	No

(1) What is the size of the input space?
The size of input space is 4*3*2*3*2=144.

(2) Design a hypothesis representation for the concept learning problem, based on your common sense and domain knowledge.
Each hypothesis is described by a conjunction of constraints on the attributes *Food Temperature, Food State, Greasy, Caloric Content, Sweet*. The constraints may also be "?" representing any value acceptable or " ϕ " representing no value acceptable or a specific value.
And we use < , , , , > to represent the hypothesis h(x), such as <Warm, ?, Yes, Medium, ϕ >.

(3) Given your representation, what is the size of the hypothesis space H?
The size of hypothesis space: |H|=1+(5*4*3*4*3)=721.

(4) Is H biased? If yes, give an example concept that is not contained in H. If no, explain.
Yes, H is biased. The hypothesis we built includes only conjunctions of contributes. So any disjunctive target concepts are not contained. For example, <Warm or Hot, ?, Yes, Medium, No> which represents that food temperature could be "Warm" or "Hot" to give the c(x).

(5) Given the above set of labeled instances, what is the version space? Show your derivations.
Using Candidate-Elimination

$$S_0 = \langle \phi, \phi, \phi, \phi, \phi \rangle$$
$$S_1 = \langle Warm, Solid, Yes, Medium, Yes \rangle$$
$$S_2 = \langle Warm, Solid, ?, Medium, ? \rangle$$
$$S_3 = \langle Warm, Solid, ?, Medium, ? \rangle$$
$$S_4 = \langle Warm, Solid, ?, Medium, ? \rangle$$

$$\langle Warm, Solid, ?, ?, ? \rangle, \langle Warm, ?, ?, Medium, ? \rangle, \langle ?, Solid, ?, Medium, ? \rangle, \langle Hot, Solid, ?, ?, ? \rangle, \langle Hot, ?, ?, Medium, ? \rangle$$

$$G_4 = \langle Warm, ?, ?, ?, ? \rangle, \langle Hot, ?, ?, ?, ? \rangle, \langle ?, Solid, ?, ?, ? \rangle$$
$$G_3 = \langle Warm, ?, ?, ?, ? \rangle, \langle Cool, ?, ?, ?, ? \rangle, \langle Hot, ?, ?, ?, ? \rangle, \langle ?, Solid, ?, ?, ? \rangle, \langle ?, Liquid, ?, ?, ? \rangle$$
$$G_2 = \langle ?, ?, ?, ?, ? \rangle$$
$$G_1 = \langle ?, ?, ?, ?, ? \rangle$$
$$G_0 = \langle ?, ?, ?, ?, ? \rangle$$

(6) If your version space in the answer to (5) contains only one hypothesis, skip this section. If it contains more than one, what is the next labeled instance you would request to expedite learning? Justify your answer.

I will request <Hot, Solid, Yes, Low, No> to expedite the learning. Actually, Greasy or Sweet do not matter. Any constraints are OK.

Firstly, we need to figure out how many constraints left for each attribute. Food Temperature, we need to clarify two constraints. For Food State, only one. None for Greasy and Sweet. All for Caloric Content.

Secondly, if we want to expedite the learning, we need try to make instance with binary contradict constrains against what we have for now. For example, if we have Yes for Greasy now. We should request No for next instance which will try to remove half of the hypothesis to shrink the version space expeditely. Furthermore, we need create as more as attributes(3 total) in above situation for quicker learning. As a result, we should set "Hot" against "Warm", "Solid", any, "Low" or "High" against "Medium", any for next

Finally, for <Hot, Solid, Yes, Low, No>. If getting "yes", we will get $S_5 = \langle ?, Solid, ?, ?, ? \rangle$ that remove two attributes concerns. If we gets "no", we will get $G_5 = \langle Warm, ?, ?, Medium, ? \rangle, \langle Warm, ?, ?, High, ? \rangle$. This will also remove two concerns and also remove some hypothesis for Caloric Contents difference.

Therefore, no matter what c(x) is, this instance will expedite the learning most.

Q2.

Give the sequence of *S* and *G* boundary sets computed by the CANDIDATE-ELIMINATION algorithm if it is given the sequence of training examples from Table 2.1 in the order: 2,4,3,1 (i.e., first consider example 2, then 4, then 3, and finally 1). Although the final version space will be the same regardless of the sequence of examples (why?), the sets *S* and *G* computed at intermediate stages will, of course, depend on this sequence. Can you come up with ideas for ordering the training examples to minimize the sum of the sizes of these intermediate *S* and *G* sets for the *H* used in the *EnjoySport* example?

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Firstly, the final version space will not change when the sequence of examples change, because as long as the function and algorithm have been decided for the learning model the result won't change as the whole structure has been decided along the functions and algorithms. However, the process of running the algorithm will change according to the order.

To order the training examples to minimize the sum of the sizes of these intermediate, we need to create as many as binary contradict constrains for one attribute to remove half of the version space based on this attribute. And as many as attributes that have contradict constrains for the earlier training example. Finally, we should let training examples with positive concept first. Because to generalize S sets, we usually have one size set. However, any time we specialize G sets, we will more likely increase size.

To sum, we have three strategies:

1.Contradict constrains in the attribute for 2 adjacent examples.

2.As many attributes having contradict constrains as the 2 examples.

3.Put the 2 examples as first as possible.

4.Positive examples go first.

The first two should be 1,4. And which of them goes first does not matter. 2 is following and 3 is last.

1,4,2,3:

$S_0 = \langle \phi, \phi, \phi, \phi, \phi \rangle$
$S_1 = \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$
$S_2 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$
$S_3 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$
$S_4 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$

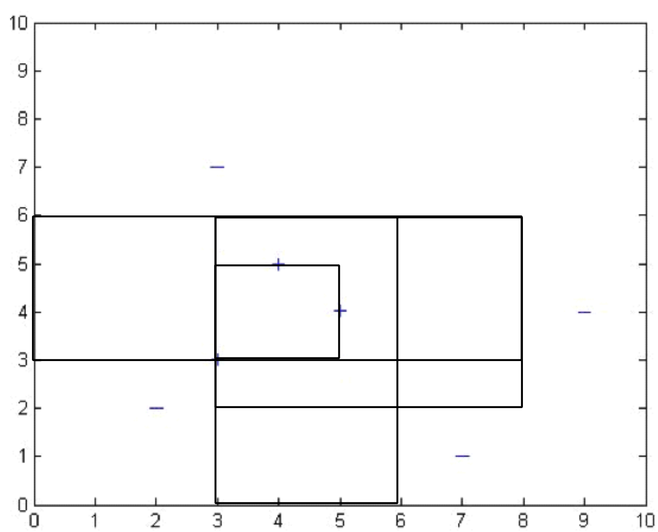
$\langle Sunny, ?, ?, Strong, ?, ? \rangle, \langle Sunny, Warm, ?, ?, ?, ? \rangle, \langle ?, Warm, ?, Strong, ?, ? \rangle$

$G_4 = \langle Sunny, ?, ?, ?, ? \rangle, \langle ?, Warm, ?, ?, ? \rangle$
$G_3 = \langle ?, ?, ?, ?, ? \rangle$
$G_2 = \langle ?, ?, ?, ?, ? \rangle$
$G_1 = \langle ?, ?, ?, ?, ? \rangle$
$G_0 = \langle ?, ?, ?, ?, ? \rangle$

Q3.

Consider the instance space consisting of integer points in the *x*, *y* plane and the set of hypotheses *H* consisting of rectangles. More precisely, hypotheses are of the form $a \le x \le b$, $c \le y \le d$, where *a*, *b*, *c*, and *d* can be any integers.

(1) Consider the version space with respect to the set of positive (+) and negative (-) training examples shown below. What is the *S* boundary of the version space in this case? Write out the hypotheses and draw them in on the diagram.



To consider the three positive examples as Points(3,3), (4,5), (5,4), the S should be rectangle

3≤x≤5, 3≤y≤5.

(2) What is the *G* boundary of this version space? Write out the hypotheses and draw them in.

To consider the four negative examples as Points(2,2), (3,7), (7,1), (9,4), the G should be rectangle

$-\infty$ ≤x≤8, 3≤y≤6;

3≤x≤6, $-\infty$ ≤y≤6;

3≤x≤8, 3≤y≤6

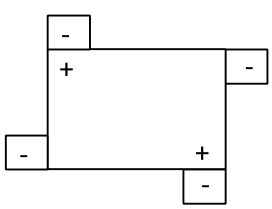
(3) Suppose you can get one more labeled input of your choice. Suggest a new x,y point guaranteed to reduce the size of the version space, regardless of how it is labeled. Suggest one that is not so guaranteed.

(7,5) will guarantee to reduce the size of version space regardless of how it is labeled.

(4,4) will not guarantee because if it is labeled by positive, the version space will not reduce.

(4) Disregard the labeled examples above (namely, re-start with an empty training set). Now, assume you are a teacher, attempting to teach a particular target concept (e.g., 5 ≤ x ≤ 8, 1 ≤ y ≤ 6). What is the smallest number of labeled training examples you need to provide so that the CANDIDATE-ELIMINATION algorithm will perfectly learn the target concept?

6 is the smallest number. Because we need four negative instances to narrow down G boundary with rectangles . And we need two positive instances to anchor the two diagonals. For 5 ≤ x ≤ 8, 1 ≤ y ≤ 6, we have shapes like below:



Q4.

Consider a concept learning problem in which we would like to find the optimal "pleasant temperature" for people. More precisely, the hypothesis space H consists of all hypotheses of the form $a < t < b$, where a and b are any real constants, and t refers to the temperature. For example, the hypothesis "18.5 < t < 25.7" classifies any temperature greater than 18.5℃ but lower than 25.7℃ as a "pleasant temperature", and any other temperature as not "pleasant temperature".

4a. For a given set D of three positive training examples (e.g. {(18.6,+), (20.1,+), (21.5,+)}), can you find a maximally specific hypothesis consistent with D? If yes, list it, otherwise explain why not.

NO. I cannot list it because a specific hypothesis is an interval between “a”,”b” which are real constants. As the given examples, we have only the generalize the specific boundary that is “a<18.6”, “b>21.5”. However, we even cannot confirm the exact number. It’s “21.56”, “21.507623”, or ”18.59”. Any hypothesis with interval “a” and “b” can have a positive instance, so that it’s never a specific hypothesis.

Therefore, we cannot find it.

4b. Suppose you now expand your training set with two additional training examples, one much smaller than all the points in D, and one much larger (e.g. {(1,-), (99,-)}). Answer the same question again with regard to the expanded training set.

NO. Now we have negative examples to specify general boundary. However, we still cannot find the maximally specific hypothesis. For now, we can say that “1<a<18.6”, “21.5<b<99” and “a<t<b” and we do have a general boundary. Because a and b are real constants, we cannot confirm the exact number for maximum specific hypothesis consistent with D. For example, “19<t<29” is a hypothesis, however, “18.6<t<21.5” is more specific, and “18.6001<t<21.50111” is even more specific because of real constants.

Q5.

Consider a learning problem where each instance is described by a conjunction of *n* Boolean attributes $a_1 \ldots a_n$. Thus, a typical instance would be $(a_1 = T) \wedge (a_2 = F) \wedge \ldots \wedge (a_n = T)$.

Now consider a hypothesis space *H* in which each hypothesis is a *disjunction* of constraints over these attributes. For example, a typical hypothesis would be $(a_1 = T) \vee (a_5 = F) \vee (a_7 = T)$.

Assume the target function is in H. Propose an algorithm that accepts a sequence of training examples and outputs a consistent hypothesis. Your algorithm should run in time that is polynomial in n and in the number of training examples.

Firstly, we assume that we have a consistent hypothesis that means the examples do not contain noisy instance that make version space NULL.

H= $(a_1 = T) \vee (a_1 = F) \vee (a_2 = T) \vee (a_2 = F) \vee \ldots \vee (a_n = T) \vee (a_n = F)$

For x in examples D:

    If c(x) is negative:

        For i from 0 to n:

If ai=T: remove ai=T from H

Else: remove ai=F from H

Else: Continue

Return H

Time complexity: To run every example costs m, because the worst case is that all examples are negative. To test every attribute in examples costs n.

Time complexity is mn.