EECS 349 (Machine Learning) Homework 1

Xinyi Chen

Problem 1

A) There are more than one H_D, given D, X, L.

Because in H_D, a hypothesis(for example hypothesis A) can be replaced by a different hypothesis A' which is indistinguishable from A with given D, but distinguishable from A with given X, then H_D becomes H_D', which is a new hypotheses set but H_D' is also a largest set of distinguishable hypotheses with given D.

B) Size of $H_D = |L|^{|D|}$

Because in hypotheses, every sample can correspond to |L| different labels, so there are at most $|L|^{|D|}$ combinations of D and labels, which means there are at most $|L|^{|D|}$ distinguishable hypotheses.

C) T $= \frac{|L|^{|D|}}{10^9} = 1.27 \times 10^{21}$ seconds $=$ 40196.9 billion years

This is not a reasonable time to wait. Because there are $|L|^{|D|}$ distinguishable hypotheses, in the worst case, all of them have to be tested to see if it is indistinguishable to *c*.

D) $\text{P(D)} = \frac{1}{|L|^{|D|}}$ $\quad$ $\text{P(X} \cap \text{ D)} = \frac{1}{|L|^{|X|}}$

$\text{P(X|D)} = \frac{P(X \cap D)}{P(D)} = \frac{|L|^{|D|}}{|L|^{|X|}} = \frac{1}{2^{100}}$

There are $|L|^{|D|}$ distinguishable hypotheses given D in total, only one of them is indistinguishable from *c*, so $\text{P(D)} = \frac{1}{|L|^{|D|}}$. There's only one hypothesis is indistinguishable from *c* in X and it must indistinguishable from *c* in D , so $\text{P(X} \cap \text{ D)} = \frac{1}{|L|^{|X|}}$. And $\text{P(X|D)} = \frac{P(X \cap D)}{P(D)}$.

Problem 2

A)

$$\text{Gain(S, A)} \equiv \text{Entropy(S)} - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$\text{Entropy(S)} = -\frac{3}{5} log_2\left(\frac{3}{5}\right) - \frac{2}{5} log_2\left(\frac{2}{5}\right) = 0.9709$$

$$\begin{aligned}\text{Gain(S, Country)} &= \text{Entropy(S)} - \frac{4}{5} Entropy(C_{japan}) - \frac{1}{5} Entropy(C_{USA}) \\ &= \text{Entropy(S)} - \frac{4}{5}0.811 - \frac{1}{5}0 \\ &= 0.322\end{aligned}$$

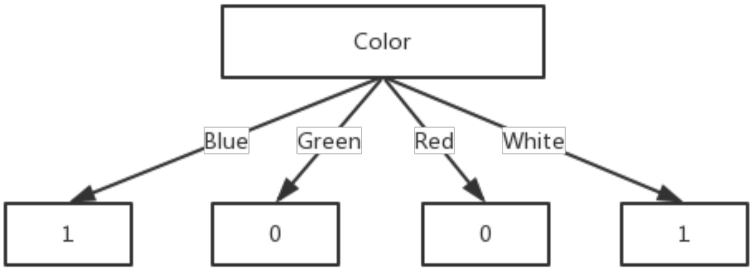
$$\begin{aligned}\mathrm{Gain}(\mathrm{S},\mathrm{Manufacturer})\\ &= \mathrm{Entropy}(\mathrm{S}) - \frac{2}{5}Entropy(M_{Honda}) - \frac{2}{5}Entropy\left(M_{Toyota}\right)\\ &- \frac{1}{5}Entropy\left(M_{Chrysler}\right)\\ &= \mathrm{Entropy}(\mathrm{S}) - \tfrac{2}{5}0 - \tfrac{2}{5}1 - \tfrac{2}{5}0\\ &= 0.571\end{aligned}$$

$$\begin{aligned}\mathrm{Gain}(\mathrm{S},\mathrm{Color}) &= \mathrm{Entropy}(\mathrm{S}) - \frac{2}{5}Entropy(C_{Blue}) - \frac{1}{5}Entropy(C_{Green})\\ &- \frac{1}{5}Entropy(C_{Red}) - \frac{1}{5}Entropy(C_{White})\\ &= \mathrm{Entropy}(\mathrm{S}) - \tfrac{2}{5}0 - \tfrac{1}{5}0 - \tfrac{1}{5}0 - \tfrac{1}{5}0\\ &= 0.9709\end{aligned}$$

$$\begin{aligned}\mathrm{Gain}(\mathrm{S},\mathrm{Decade})\\ &= \mathrm{Entropy}(\mathrm{S}) - \frac{3}{5}Entropy(D_{1980}) - \frac{1}{5}Entropy(D_{1970})\\ &- \frac{1}{5}Entropy(C_{1990})\\ &= \mathrm{Entropy}(\mathrm{S}) - \tfrac{3}{5}0.9183 - \tfrac{1}{5}0 - \tfrac{1}{5}0\\ &= 0.42\end{aligned}$$

$$\begin{aligned}\mathrm{Gain}(\mathrm{S},\mathrm{Type}) &= \mathrm{Entropy}(\mathrm{S}) - \frac{4}{5}Entropy\left(T_{Economy}\right) - \frac{1}{5}Entropy\left(T_{Sports}\right)\\ &= \mathrm{Entropy}(\mathrm{S}) - \tfrac{4}{5}0.8113 - \tfrac{1}{5}0\\ &= 0.322\end{aligned}$$

Because $\mathrm{Gain}(\mathrm{S},\mathrm{Color})$ is the biggest, so the root of decision tree is Color.



B)

Logical function: $f(\mathrm{x}) = \begin{cases} 1 \ \ x = blue \\ 0 \ x = green \\ 0 \ x = red \\ 1 \ x = white \end{cases}$

This function categorizes data above well, percent of cases correctly = 100%.

I feel this doesn't capture the concept “Japanese Economy Car”. The decision tree categorizes cars by color, which has nothing to do with “Japanese Economy Car”.

I think this result is caused by small training set, there are only 5 examples in training set, accidentally data can be well categorized by color. To capture the concept “Japanese Economy Car”, we need a much bigger training set.

Problem 3

A) I would pick a split point by using following approach.

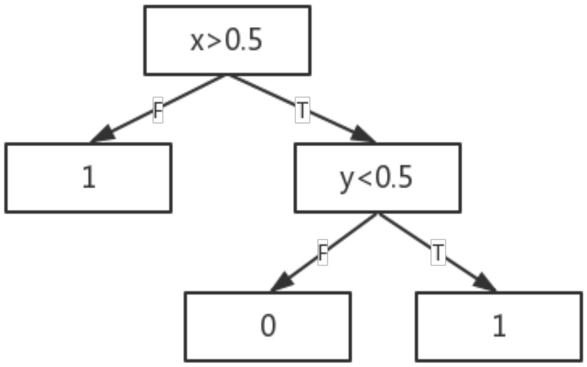
If a attribute has N real values, for example [1.2, 4.5, 6.7,… Nth value], between every two values there’s a possible split point, so there are N-1 possible split points in total.

Calculate the information gain of each possible split point, choose the one that has the biggest information gain value to be the split point.

The strength of my approach: It can pick the split point with highest information gain value.

The weakness: It has to calculate N-1 times, if N is very big, it can be time-consuming.

B) The first one cannot be represented as a decision tree, because there’s no specific split point for x and y value. The second one can be represented as a following decision tree:



Problem 4

A) Statistical measures of the performance of a binary classification test. Positive = Identified, Negative = Rejected [1] so:

True Positive = correctly identified [1] e.g. correct password accepted

False Positive = incorrectly identified [1] e.g. wrong password accepted

True Negative = correctly rejected [1] e.g. wrong password rejected

False Negative = incorrectly rejected [1] e.g. correct password rejected

References:

[1]. Wikipedia: Sensitivity and specificity

B) Precision: The fraction of retrieved instances that are relevant. [1]

$$\text{Precision} = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{retrieved\ instances\}|}$$ [1]

Recall: The fraction of relevant instances that are retrieved. [1]

$$\text{Recall} = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{relevant\ instances\}|}$$ [1]

e.g. A search engine returns 50 results, only 30 of them are relevant to keyword while failing to return 30 additional relevant results. The precision is 30/50 = 0.6, recall = 30/60 =0.5.

References:

[1]. Wikipedia: Precision and Recall

C) F1 score: A measure of a test's accuracy in statistical analysis of binary classification, it can be interpreted as a weighted average of the precision and recall, its best value at 1 and worst at 0. [1]

$$\text{F1 Score} = \frac{precision \cdot recall}{precision + recall}$$ [1]

References:

[1]. Wikipedia: F1 Score

D) Confusion Matrix: In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix [1], is a specific table layout that allows visualization of the performance of an algorithm (typically a supervised learning one). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa) [2]

e.g. A system has been trained to distinguish three basic human actions, walking, running and jumping. Confusion matrix:

		Predicted		
		walking	running	jumping
Actual Class	walking	6	3	0
	running	1	5	2
	jumping	0	3	4

In this confusion matrix, of 9 actual walking cases, the system predicted 6 of them are walking, 3 are running and 0 is jumping. In this matrix we can see that the system performance is good when distinguish between walking and jumping, but has trouble with distinguish among other types of actions. It's easy to visually inspect the table for errors, as they will be represented by values outside the diagonal. [3]

References:

[1]. Stehman, Stephen V. (1997). "Selecting and interpreting measures of thematic classification accuracy". Remote Sensing of Environment. 62 (1): 77–89. doi:10.1016/S0034-4257(97)00083-7

[2]. Powers, David M W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). Journal of Machine Learning Technologies. 2 (1): 37–63.

[3]. Wikipedia: Confusion matrix