The Solution to Assignment 6

Problem 1: Use the 2-fold cross-validation to evaluate the Decision Tree Model for trees up to 2 levels deep (that is, the maximum path length from the root to the leaves is 2). Show the two trees, the confusion matrices and the average error rate.

Solution: Since the 2-fold cross-validation is required to evaluate the decision tree model, we need to partition the original dataset into two folds at first. In theory, in order to guarantee the effectiveness of cross-validation, the instances in the original dataset should be randomly selected to form the two folds. However, for simplicity, we manually partition the original dataset, that is, the first seven instances as the first fold and the rest of seven instances as the second one. Then we calculate the model by hand in the following.

(1). Use the first fold for training and the second for testing.

(a).

gain("Outlook") = info([4, 3]) – info([0, 2], [2, 0], [2, 1])
= entropy([4/7, 3/7]) – (2/7*entropy([0, 1]) + 2/7*entropy([1, 0]) + 3/7*entropy([2/3, 1/3]))
= 0.1781

gain("Temperature") = info([4, 3]) – info([1, 2], [1, 0], [2, 1])
= entropy([4/7, 3/7]) – (3/7*entropy([1/3, 2/3]) + 1/7*entropy([1, 0]) + 3/7*entropy([2/3, 1/3]))
= 0.0596

gain("Humidity") = info([4, 3]) – info([2, 2], [2, 1])
= entropy([4/7, 3/7]) – (4/7*entropy([1/2, 1/2]) + 3/7*entropy([2/3, 1/3]))
= 0.0061

gain("Windy") = info([4, 3]) – info([1, 2], [3, 1])
= entropy([4/7, 3/7]) – (3/7*entropy([1/3, 2/3]) + 4/7*entropy([3/4, 1/4]))
= 0.0386

gain_ratio("Outlook") = gain("Outlook") / info([2, 2, 3]) = 0.38
gain_ratio("Temperature") = gain("Temperature") / info([3, 1, 3]) = 0.1367
gain_ratio("Humidity") = gain("Humidity") / info([4, 3]) = 0.0205
gain_ratio("Windy") = gain("Windy") / info([3, 4]) = 0.13

Since "Outlook" has the maximum gain ratio, it should be chosen as the splitting node.

(b).

gain("Temperature") = info([2, 1]) – info([0, 0], [1, 0], [1, 1])
= entropy([2/3, 1/3]) – (1/3*entropy([1, 0]) + 2/3*entropy([1/2, 1/2]))
= 0.0757

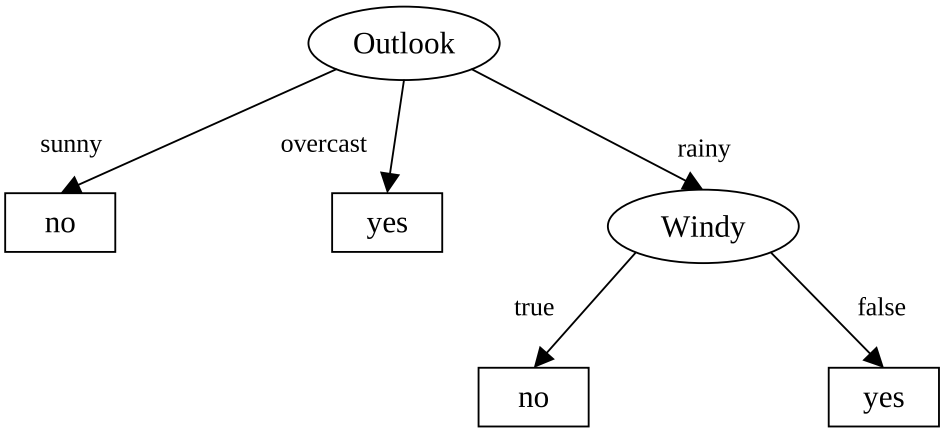
gain("Humidity") = info([2, 1]) – info([1, 0], [1, 1])
= entropy([2/3, 1/3]) – (1/3*entropy([1, 0]) + 2/3*entropy([1/2, 1/2]))
= 0.0757

gain("Windy") = info([2, 1]) – info([0, 1], [2, 0])
= entropy([2/3, 1/3]) – (1/3*entropy([0, 1]) + 2/3*entropy([1, 0]))
= 0.2764

gain_ratio(“Temperature”) = gain(“Temperature”) / info([0, 1, 2]) = 0.274
gain_ratio(“Humidity”) = gain(“Humidity”) / info([1, 2]) = 0.274
gain_ratio(“Windy”) = gain(“Windy”) / info([1, 2]) = 1

Therefore, “Windy” should be chosen as the splitting node.

(c). The decision tree is shown as follows:



(d). Confusion matrix

		predicted	
		yes	no
actual	yes	3	2
	no	0	2

Error rate = 2/7 = 28.57%.

(1). Use the second fold for training and the first for testing.

(a).
gain(“Outlook”) = info([5, 2]) – info([2, 1], [2, 0], [1, 1])
= entropy([5/7, 2/7]) – (3/7*entropy([2/3, 1/3]) + 2/7*entropy([1, 0]) + 2/7*entropy([1/2, 1/2]))
= 0.0553

gain(“Temperature”) = info([5, 2]) – info([1, 0], [3, 2], [1, 0])
= entropy([5/7, 2/7]) – (1/7*entropy([1, 0]) + 5/7*entropy([3/5, 2/5]) + 1/7*entropy([1, 0]))
= 0.0511

gain(“Humidity”) = info([5, 2]) – info([1, 2], [4, 0])
= entropy([5/7, 2/7]) – (3/7*entropy([1/3, 2/3]) + 4/7*entropy([1, 0]))
= 0.1414

gain(“Windy”) = info([5, 2]) – info([2, 1], [3, 1])
= entropy([5/7, 2/7]) – (3/7*entropy([2/3, 1/3]) + 4/7*entropy([3/4, 1/4]))
= 0.0018

gain_ratio(“Outlook”) = gain(“Outlook”) / info([3, 2, 2]) = 0.1181

gain_ratio(“Temperature”) = gain(“Temperature”) / info([1, 5, 1]) = 0.1803
gain_ratio(“Humidity”) = gain(“Humidity”) / info([3, 4]) = 0.4766
gain_ratio(“Windy”) = gain(“Windy”) / info([3, 4]) = 0.0061

Therefore, “Humidity” should be chosen as the splitting node.

(b).
gain(“Outlook”) = info([1, 2]) – info([0, 1], [1, 0], [0, 1])
= entropy([1/3, 2/3]) – (1/3*entropy([0, 1]) + 1/3*entropy([1, 0]) + 1/3*entropy([0, 1]))
= 0.2764

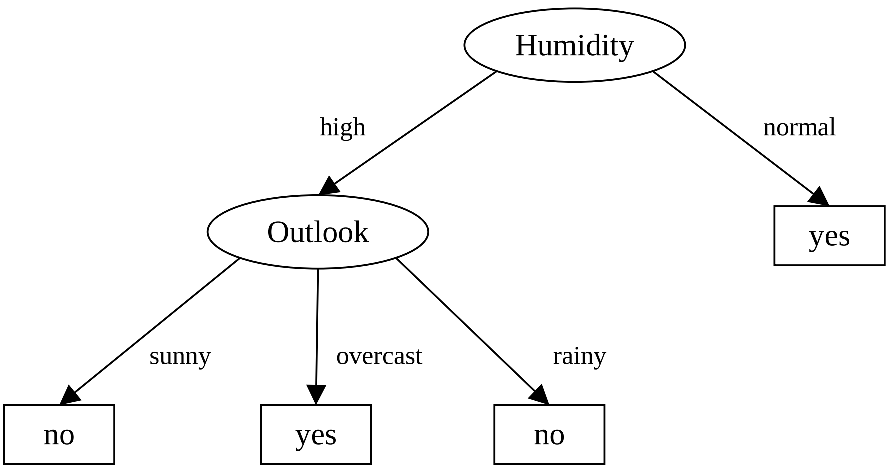
gain(“Temperature”) = info([1, 2]) – info([0, 0], [1, 2], [0, 0])
= 0

gain(“Windy”) = info([1, 2]) – info([1, 1], [0, 1])
= entropy([1/3, 2/3]) – (2/3*entropy([1/2, 1/2]) + 1/3*entropy([0, 1]))
= 0.0757

gain_ratio(“Outlook”) = gain(“Outlook”) / info([1, 1, 1]) = 0.5794
gain_ratio(“Temperature”) = gain(“Temperature”) / info([0, 3, 0]) = 0
gain_ratio(“Windy”) = gain(“Windy”) / info([2, 1]) = 0.274

Therefore, “Outlook” should be chosen as the splitting node.

(c). The decision tree is shown as follows:



(d). Confusion matrix

		predicted	
		yes	no
actual	yes	3	1
	no	1	2

Error rate = 2/7 = 28.57%
The overall average error rate = (2/7 + 2/7)/2 = 28.57%.

(3). Use Weka package.

The decision tree algorithm in Weka is weka. classifiers. j48. J48. The number of folds for cross-validation can be set to be 2 by using the option –x. However, the built-in mechanism is stratified holdout, which partitions the original datset into two folds by randomly sampling, and at the same time, retains the original class distribution, e.g. the ratio of "Yes" to "No". In addition, the resulting decision tree is a pruned one which is one level deep for our dataset. Therefore, the confusion matrix and error rate may be different from the results calculated by hand.

Problem 2: Use the 2-fold cross-validation to evaluate the Naive Bayesian Model for this problem. Use a Laplace-estimator μ=0.01. Show the two conditional probability tables and the confusion matrices. Show the average error rate.
Solution: Similar to problem 1, we manually partition the original dataset into two folds.

(1). Use the first fold for training and the second for testing.

(a). The conditional probability table:

Outlook			Temperature			Humidity			Windy			Play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	0	2	hot	1	2	high	2	2	true	1	2	4	3
overcast	2	0	mild	1	0	normal	2	1	false	3	1		
rainy	2	1	cool	2	1								
sunny	0.0008	0.6656	hot	0.2502	0.6656	high	0.5	0.6661	true	0.2506	0.6661	0.5714	0.4286
overcast	0.4996	0.0011	mild	0.2502	0.0011	normal	0.5	0.3339	false	0.7494	0.3339		
rainy	0.4996	0.3333	cool	0.4996	0.3333								

Note: In order to avoid the "zero frequency" problem, we usually use a Laplace-estimator μ in Naïve Bayesian. We will use "Outlook" and "Humidity" to illustrate how to use μ to compute the conditional probability as follows:

For "Outlook",

P(yes|sunny) = (0+ μ/3)/(4+ μ) = 0.0008

P(yes|overcast) = (2+ μ/3)/(4+ μ) = 0.4996

P(yes|rainy) = (2+ μ/3)/(4+ μ) = 0.4996

For "Humidity",

P(yes|high) = (2+ μ/2)/(4+ μ) = 0.5

P(yes|normal) = (2+ μ/2)/(4+ μ) = 0.5

However, Laplace-estimator μ does not influence the probability P(yes) or P(no).

P(yes) = 4/7 = 0.5714

P(no) = 3/7 = 0.4286

(b). Confusion matrix

		predicted	
		yes	no
actual	yes	3	2
	no	1	1

Error rate = 3/7 = 42.86%

(2). Use the first fold for testing and the second for training.

(a). The conditional probability table:

Outlook			Temperature			Humidity			Windy			Play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	1	hot	1	0	high	1	2	true	2	1	5	2
overcast	2	0	mild	3	2	normal	4	0	false	3	1		
rainy	1	1	cool	1	0								
sunny	0.3999	0.4992	hot	0.2003	0.0017	high	0.2006	0.9975	true	0.4002	0.5	0.7143	0.2857
overcast	0.3999	0.0016	mild	0.5994	0.9966	normal	0.7994	0.0025	false	0.5998	0.5		
rainy	0.2002	0.4992	cool	0.2003	0.0017								

(b). Confusion matrix

		predicted	
		yes	no
actual	yes	3	1
	no	3	0

Error rate 2 = 4/7 = 57.14%

The overall average error rate = (3/7 + 4/7) / 2 = 50%.

(3). Use Weka Package

The Naïve Bayesian algorithm in Weka is weka. classifiers. NaiveBayes. By default, it always add 1 to the number of different values for a particular attribute. In addition, there is not an option which can be used to specify the value of μ. In this case, we need to calculate the result by hand.