Shikhar Mohan

EECS 349 – HW1

**Problem 1:**

**d1 = <J H B 80 E> +**

S = <J H B 80 E>

G = <? ? ? ? ? >

**d2 = <J T G 70 S> -**

S = <J H B 80 E>

G = { < ? H ? ? ? > , < ?, ? ? B ?> < ? ? ? 80 ?> <? ? ? ? E> }

**d3 =< J T B 90 E> +**

S = <J ? B ? E>

G ={ < ? ? ? B ? > <? ? ? ? E> }

**d4 = <U C R 80 E> -**

S = <J ? B ? E>

G = {<? ? ? B ?>, <J,?,?, E>}

**d5 = <J H W 80 E>+**

S = <J ? ? ? E>

G = <J ? ? ? E>

**Problem 2:**

A.) Let X and Y be two users of Facebook. Each user has a unique user ID this can be represented as a feature of the user profile X = {x1} and Y = {y1}. Then the distance metric, d(X,Y), is equal to the size of the set of features not common to both users.

d(X,Y) = | (X union Y) \ (X intersection Y) |

Math proof:

d(X,Y) = 0 => X = Y => x1 = y1 => x = y

Reflexivity: x=y => X = Y => d(X,Y) = 0.

Symmetry: d(x,y) = | (X union Y) \ (X intersection Y) | = | (Y union X) \ (Y intersection X) |

Non-negativity: d(X,Y) = | (X union Y) \ (X intersection Y) | => d(x,y) >= 0 because size of a set cannot be negative.

Triangle inequality can be proved with properties of set addition and cardinality.

| (X union Y) \ (X intersection Y) | + | (Y union Z) \ (Y intersection Z) | = |X union Z|

=> (X union Z) \ (X intersection Z) (X union Y union Z) \ (X intersection Y intersection Z)

B.) Expected range of values for height (1ft – 7ft) ; weight (3kg – 150kg); hairs on head (1,000 – 1,000,000). It doesn’t make sense to treat these values equally because height and weight don’t naturally correlate to a person’s age. There needs to be weights added to each element indicating it’s importance when clustering people by age. A suitable metric would include 60% weight on hairs, 25% weight on height and 15% weight on weight. This is suitable because hair loss is somewhat strongly correlated to seniority, so it should be heavily weighted, and this same logic follows with height and weight and their respective correlations with a person’s age.

C. ) This metric needs to have a weighting function to favor longest subsequences when matching two strands of DNA. Simply using distance as measured by the number of “edit operations” is not sufficient in the context of DNA matching where the subset of labels is only limited to 4 letters but the sample size can be very long, so the probability that a matching subsequence exists is much higher than a single edit operation taking place in order to create an optimal match.

**Problem 3:**

A.) DT is governed by |L|^|T|. DT is the largest set of distinguishable hypotheses given an example set T therefore, the size of DT is related to the size of the set T. Now, given that each T can be assigned a label from L, the potential size of DT is |L|^|T|.

B.) P = 1/|DT| = 1/ 2100- assuming that the likeliness of the labeling is the same, in order find a hypothesis that is indistinguishable from the target function, the size of our entire space is determined by the unique values that the set T can assume with L labels. All we need is one example from an example space with size 2100, which results in the probability being 1/2100.

C.) Specifically it is P = 2100/2200 = 1/2100. Since we are adding another 100 new examples still with 2 possible labels, there is now an additional 2100 possible ways to find a distinguishable set (i.e. proven wrong). And we have already found an indistinguishable hypothesis using T, so we can disregard those examples when calculating the probability of finding an indistinguishable set from X is 2100/2200 = 1/2100.

**Problem 4:**

A.)

Choose attribute with largest information gain:

IG(S) = E(S) – W(v1)\*E(v1) - W(v2)\*E(v2)

E(S) = -0.6\*log2\*(0.6) = 0.971

Level 1

origin 0.64902,

manufacturer 0.4,

decade 0.550977,

type 0.811278

color: 0.971

Color is the most optimal attribute to choose and no need to go further because each option has no children in the decision tree

Parent: root attribute:Color trueChild: Blue, White falseChild: Red, Green Information Gain: 0.971

Parent: Color attribute:Blue trueChild: leaf falseChild: leaf Information Gain: 0

Parent: Blue +

Parent: Color attribute:White trueChild: leaf falseChild: leaf Information Gain: 0

Parent: White +

Parent: Color attribute:Red trueChild: leaf falseChild: leaf Information Gain: 0

Parent: Red –

Parent: Color attribute:Green trueChild: leaf falseChild: leaf Information Gain: 0

Parent: Green –

B.) f (s) = (colorBlue) || (colorWhite)

This function perfectly categorizes the data used to build the tree. This doesn’t capture the concept of the Japanese Economy Car because the car has many other attributes besides just color. The fact that the Color attribute had no entropy and resulted in the most information gain, caused this result where the positive classification is solely determined by whether the car is Blue or White.

**Problem 5:**

A.)

=== Classifier model (full training set) ===

Id3

IsRich = true

| GoodLetters = true

| | GoodGrades = true : true

| | GoodGrades = false

| | | GoodSAT = true : true

| | | GoodSAT = false : false

| GoodLetters = false

| | GoodGrades = true

| | | SchoolActivities = true : true

| | | SchoolActivities = false : false

| | GoodGrades = false : false

IsRich = false

| HasScholarship = true

| | GoodSAT = true

| | | GoodLetters = true : true

| | | GoodLetters = false : false

| | GoodSAT = false : false

| HasScholarship = false : false

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 60 96.7742 %

Incorrectly Classified Instances 2 3.2258 %

Kappa statistic 0.9355

Mean absolute error 0.0323

Root mean squared error 0.1796

Relative absolute error 6.448 %

Root relative squared error 35.8995 %

Total Number of Instances 62

=== Confusion Matrix ===

a b <-- classified as

31 0 | a = true

2 29 | b = false

The confusion matrix shows a representation of how good the training set is. In order to have a good training set, you need large numbers on the main diagonal.

B.)

=== Classifier model (full training set) ===

J48 pruned tree

------------------

IsRich = true

| GoodLetters = true : true (25.0/1.0)

| GoodLetters = false

| | GoodGrades = true

| | | SchoolActivities = true : true (3.0)

| | | SchoolActivities = false : false (4.0)

| | GoodGrades = false : false (6.0)

IsRich = false

| HasScholarship = true

| | GoodSAT = true : true (5.0/1.0)

| | GoodSAT = false : false (6.0)

| HasScholarship = false : false (13.0)

Number of Leaves : 7

Size of the tree : 13

Correctly Classified Instances 59 95.1613 %

Incorrectly Classified Instances 3 4.8387 %

Kappa statistic 0.9032

Mean absolute error 0.0715

Root mean squared error 0.2174

Relative absolute error 14.2982 %

Root relative squared error 43.46 %

Total Number of Instances 62

=== Confusion Matrix ===

a b <-- classified as

30 1 | a = true

2 29 | b = false

This run was less accurate than 5A. Pruning chooses the most probable option in the training set, which doesn’t necessarily mean that it will be equally accurate for the testing set, as it could have a different probability and distribution of an option.

C.) Yes, there is a difference in performance on these data sets – the decision tree was much more accurate for IvyLeague compared to MajorityRule, given by the confusion matrix. The structure of the concepts is much stronger in IvyLeague compared to MajorityRule – the concepts have a stronger relationship to the classification for each example. It’s hard to represent influence of decisions between people simply by using the concepts used in MajorityRule. IvyLeague concepts were much more comprehensive.

D.) No, because of the inherent structure of the concepts and probabilities associated with attributes and classifications within MajorityRule removes any large positive effect that Reduced Error Pruning might have on MajorityRule.

**Problem 6**

See program attached.