Technion - Israel Institute of Technology

Introduction to Artificial Intelligence

Professor Oren Salzman

TA Tal Swisa

Homework 3

Pietro BRACH DEL PREVER 921210282
Yaacov VAKSMAN 316153261

January 27, 2021

Academic Year 2021/2022

Part B

Question 1

The statement is true. Let some node n in the ID3 Algorithm. Without the MinMax normalization, we choose to split by some continues feature f. After performing dynamic discretization we get that the highest information-gain value we can get is by using feature f and some threshold value t_j .

Let us now consider the MinMax normalization. let x be the value of the feature, the normalized value is x_n :

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{x}{x_{max} - x_{min}} - \frac{x_{min}}{x_{max} - x_{min}}$$

The normalization value is a linear function of the original value, and this is a monotonically increasing function. Thus, performing the dynamic discretization and the splitting by every feature will lead to the same results achieved without the normalization (the values by themselves are not important, the important thing is the order: if $x_2 > x_1$ then $x_{n_2} > x_{n_1}$, so if the order of the examples haven't changed then the information gain won't change and we will chose the same feature and relative place to divide the examples by that feature, and thus the same tree will be built and the accuracy will be the same.

Question 2

General Explanation of the plots:

Element	**Meaning**
blue background	goal classifier
green and red points	example nodes labeled 1 and 0 respectively
x markers	test examples we discussing
red lines	goal classifier returns 0
blue lines	goal classifier returns 1

a

With reference to Fig. 1, ID3 algorithm will discard feature v_1 and keep only feature v_2 (achieving the goal classifier) by setting $v_2 > 0$ for class 1, while KNN, given that the two points are equidistant, will return the class of the one with higher v_2 value, which is wrong.

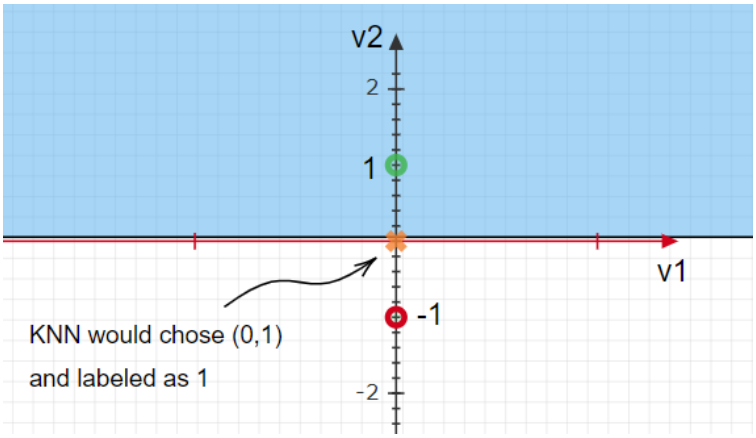

Figure 1: Training set: $D = \{< (0,1), 1 >, < (0,-1), 0 >\}$. Goal classifier $F_{true} = 1$ if $v_2 > 0$. ID3: $v_2 > 0$ (goal classifier). KNN: $x_q = (0,0)$; $d(x_q,(0,1)) = d(x_q,(0,-1))$; $KNN(x_q) = 1$; $\exists x_q : \ KNN(x_q) \neq F_{true}$.

b

With reference to Fig. 2, KNN will always be right, also on the dividing line, consistently with the goal classifier. ID3 will chose feature v_1 and discard feature v_2 because the leaves are reached after one decision node and no more splitting is available while building th tree.

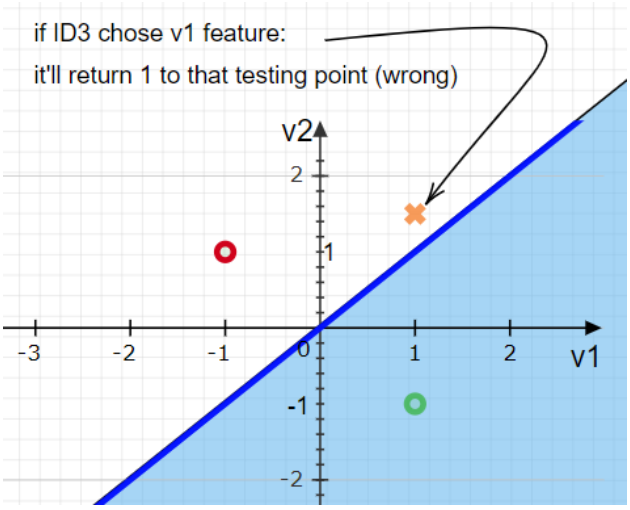

Figure 2: Training set: $D = \{< (1,-1), 1 >, < (-1,1), 0 >\}$. Goal classifier: $F_{true} = \begin{cases} 1 & v1 \geq v2 \\ 0 & v1 < v2 \end{cases}$.
KNN: achieves the goal classifier. ID3: only uses one feature and does not achieve the goal classifier.

c

With reference to Fig. 3, given $K = 1$, ID3 will only use one feature, because it reaches the leaves after defining the first decision node, achieving a bad classifier. KNN will also achieve a bad classifier for points on the line $v_1 = v_2$, as already seen in the answer to question A.

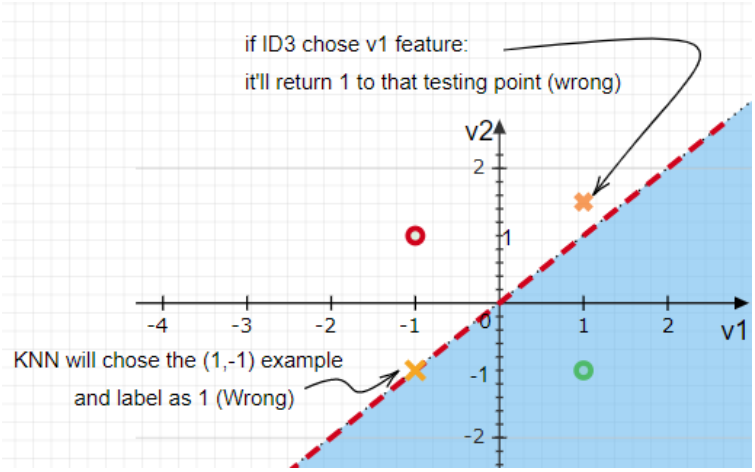

Figure 3: $K = 1$. Training set: $D = \{< (1, -1), 1 >, < (-1, 1), 0 >\}$. Goal classifier: $F_{true} = \begin{cases} 1 & v1 > v2 \\ 0 & v1 \leq v2 \end{cases}$.
KNN: $v_1 = v_2$. ID3: $v1 > 0$ *or* $v2 > 0$.

d

With reference to Fig. 4, given $K = 1$, ID3 will split according to $v_2 \geq 0$ feature only and thus reach the leaves of the decision tree; KNN will compute the distance from the training data, and points on the limiting line will be assigned label 1 because the label of the xample with the higher v_2 value is returned, thus agreeing with the goal classifier.

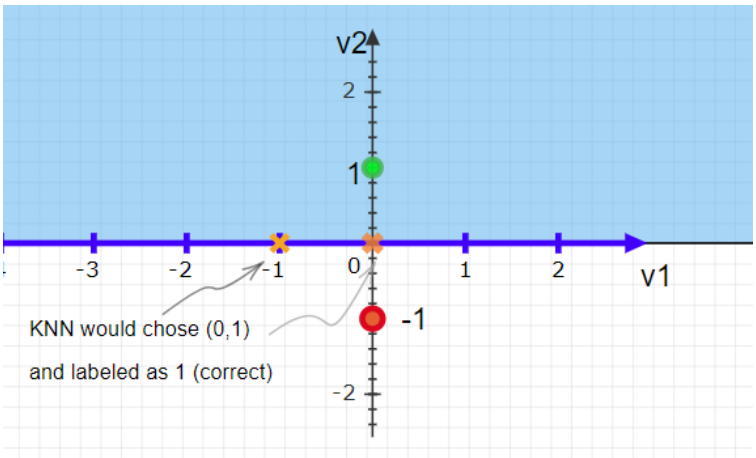

Figure 4: $K = 1$. Training set: $D = \{< (0, 1), 1 >, < (0, -1), 0 >\}$. Goal classifier: $F_{true} = 1 \; if \; v_2 \geq 0$.
KNN: $v_2 \geq 0$. ID3: $v_2 \geq 0$.

Question 3

a Training set accuracy

The number of positive labels ($y = 1$) is 5 and the number of negative labels ($y = 0$) is 5. There is no majority (tie), but according to the definition provided for the majority classifier, we say the majority is $y = 1$. Checking the accuracy of the majority classifier on the example set: $C(x_i) = 1 \ \forall x_i = 1, 2 \ldots 10$. Because half of the examples were labeled as 1 and half as 0, the classifier will be right half of the times.

$$accuracy = \frac{\#(right\ classified\ examples)}{\#examples} = 0.5$$

b 2-fold cross-validation

We run 2-fold cross-validation over the first fold:

$$x_{training} = [1, 2, 3, 4, 5]$$
$$y_{training} = [1, 1, 0, 1, 1]$$

The majority classifier is $C_1(x) \equiv 1$. We now check over the testing fold:

$$x_{testing} = [6, 7, 8, 9, 10]$$
$$y_{testing} = [0, 0, 1, 0, 0]$$

The classifier over the testing points will return: $y_{predict} = [1, 1, 1, 1, 1]$. The error is $\epsilon_1 = 4/5$.

We now run 2-fold cross-validation over the second fold:

$$x_{training} = [6, 7, 8, 9, 10]$$
$$y_{training} = [0, 0, 1, 0, 0]$$

The majority classifier is $C_2(x) \equiv 0$. W now check over the testing fold:

$$x_{testing} = [1, 2, 3, 4, 5]$$
$$y_{testing} = [1, 1, 0, 1, 1]$$

The classifier over the testing points will return: $y_{predict} = [0, 0, 0, 0, 0]$. The error is $\epsilon_2 = 4/5$.

We conclude that the average error is

$$\epsilon_{avg} = \frac{1}{N} \sum \epsilon_i = 4/5$$

and the accuracy of the algorithm is

$$accuracy = 1 - error = 1/5$$

Part C

Question 5

The accuracy achieved in the `basic_experiment` is of 96.46%.

Question 6

a

Pruning is important to avoid over-fitting the decision tree onto the training data-set. Moreover, pruning also decreases the number of nodes in the decision tree and its depth, thus allowing for a shorter runtime both in the fitting phase and in the classification of new examples.

d

The accuracy achieved in the `best_m_test` is of 97.35%.

The pruning slightly improved the performance w.r.t the `basic_experiment` without pruning. We collected the average runtime of each experiment over some runs (in each experiment we include both the training phase and the testing phase) and compared them, as shown il Table 1. However the oscillations in terms of runtime are really large.

Experiment	**Run time**	**Accuracy**
`basic_experiment`	58.55	96.46%
`best_m_test`	55.65	97.35%

Table 1: Runtime and accuracy of the two experiments - with and without pruning - compared.

Part D

Question 7

The K-NN algorithm stores the features of the train data-set and computes the distance of the examples to be classified from the K nearest neighbors. The pros include that it is really easy to implement, and that the data-set can be provided incrementally (the *lazy learning* approach analyzes the data only when it is used for classification). The downsides of the algorithm are that the classification is slow (due to the lazy approach), and that it is sensible to feature selection.

Question 8

a

We have a set of D features and want to choose a subset containing b of them ($1 \leq b \leq D$; the void set is not considered). The number of possible sets by choosing b features from D is:

$$\binom{D}{b} = \frac{D!}{(D-b)!\ b!}$$

The total number of set is obtained by summing for every value of b:

$$|P(s)| = \sum_{i=1}^{D} \binom{D}{i} = 2^D - 1$$

Note that we used a well known formula:

$$\sum_{i=0}^{D} \binom{D}{i} = 2^D$$

$$\sum_{i=1}^{D} \binom{D}{i} = \sum_{i=0}^{D} \binom{D}{i} - \binom{D}{0}$$

b

We have D features and want to choose b of them ($1 \leq b \leq D$). The number of possible sets by choosing b features from D is

$$\binom{D}{b} = \frac{D!}{(D-b)!\ b!}$$