440/540 Artificial Intelligence HW7

INSTRUCTIONS

- **Due: Wednesday, 12 April 2023 at 23:59 PM PDT** Remember that you have 5 slip days to use at your discretion BUT you can use no more than 1 per homework.
- **Format:** Submit the answer sheet containing your answers in PDF. You can provide either typed or handwritten answers to this homework. Handwritten answers must be legible and scanned into a PDF.
- **Note:** **Please DO NOT FORGET to include your name and WSU ID in your submission.**
- **How to submit:** Submit a PDF containing your answers on Canvas
- **Policy:** See the course website for homework policies and Academic Integrity.

Last Name	Ghosh Dastidar
First Name	Priyanka
WSU ID	11804812

Q1. [20 pts] Entropy

Given the following sets of 2-class values (T/F), calculate the entropy on each of the below set. Note: Use the entropy definition in the lecture slides which uses $\log_2$ (base 2) instead of the log operator with natural base (base *e*).

(a) [5 pts] {*T,T,T,T*}

Answer:

To calculate the entropy of a set, we can use the formula:

entropy = - p_true * log2(p_true) - p_false * log2(p_false)

where p_true is the proportion of true values in the set and p_false is the proportion of false values in the set.

For the set {T, T, T, T}, there are only true values, so the proportion of true values is 1 and the proportion of false values is 0.

entropy = - 1 * log2(1) - 0 * log2(0)

= - 1 * 0 - 0 * undefined

= 0

Therefore, the entropy of this set is 0.

(b) [5 pts] {*T,T,T,F*}

Answer:

To calculate the entropy of a set, we can use the formula:

entropy = - p_true * log2(p_true) - p_false * log2(p_false)

where p_true is the proportion of true values in the set and p_false is the proportion of false values in the set.

For the set {T, T, T, F}, the proportion of true values is 3/4 and the proportion of false values is 1/4.

entropy = - (3/4) * log2(3/4) - (1/4) * log2(1/4)

= - 0.75 * (-0.415) - 0.25 * (-2)

= 0.311 + 0.5

= 0.811

Therefore, the entropy of this set is 0.811

(c) [5 pts] {*T,T,F,F*}

Answer:

To calculate the entropy of a set, we can use the formula:

entropy = - p_true * log2(p_true) - p_false * log2(p_false)

where p_true is the proportion of true values in the set and p_false is the proportion of false values in the set.

For the set {T, T, F, F}, the proportion of true values is 2/4 and the proportion of false values is 2/4.

entropy = - (2/4) * log2(2/4) - (2/4) * log2(2/4)

= - 0.5 * (-1) - 0.5 * (-1)

= 0.5 + 0.5

= 1

Therefore, the entropy of this set is 1.

(d) [5 pts] {*T,F,F,F*}

Answer:

To calculate the entropy of a set, we can use the formula:

entropy = - p_true * log2(p_true) - p_false * log2(p_false)

where p_true is the proportion of true values in the set and p_false is the proportion of false values in the set.

For the set {T, F, F, F}, the proportion of true values is 1/4 and the proportion of false values is 3/4.

entropy = - (1/4) * log2(1/4) - (3/4) * log2(3/4)

= - 0.25 * (-2) - 0.75 * (-0.415)

= 0.5 + 0.311

= 0.811

Therefore, the entropy of this set is 0.811

Q2. [40 pts] Decision Tree

Given the following training dataset about exotic dishes, we want to predict whether or not a dish is *Appealing* based on the input attributes *Temperature*, *Taste* and *Size*.

ID	Temperature	Taste	Size	Appealing
1	Hot	Salty	Small	No
2	Cold	Sweet	Large	No
3	Cold	Sweet	Large	No
4	Cold	Sour	Small	Yes
5	Hot	Sour	Small	Yes
6	Hot	Salty	Large	No
7	Hot	Sour	Large	Yes
8	Cold	Sweet	Small	Yes
9	Cold	Sweet	Small	Yes
10	Hot	Salty	Large	No

(a) [10 pts] What is the information gain **Gain**(*Taste*) at the root node of the decision tree?

Answer:

To calculate the information gain of an attribute, we need to first calculate the entropy of the target variable (Appealing) and then calculate the weighted sum of entropies of the target variable after splitting on the attribute.
Entropy of target variable:

- Total number of samples = 10
- Number of positive samples (Appealing = Yes) = 5
- Number of negative samples (Appealing = No) = 5
- Probability of positive samples = 5/10 = 0.5
- Probability of negative samples = 5/10 = 0.5

Entropy(Appealing) = -0.5log2(0.5) - 0.5log2(0.5) = 1
To calculate the entropy of the target variable after splitting on the Taste attribute, we need to calculate the entropy for each value of Taste (Salty, Sweet, Sour).
Entropy(Appealing|Taste=Salty):

- Total number of samples with Taste=Salty = 3
- Number of positive samples (Appealing = Yes) = 0
- Number of negative samples (Appealing = No) = 3
- Probability of positive samples = 0/3 = 0
- Probability of negative samples = 3/3 = 1

Entropy(Appealing|Taste=Salty) = -0log2(0) - 1log2(1) = 0
Entropy(Appealing|Taste=Sweet):

- Total number of samples with Taste=Sweet = 4
- Number of positive samples (Appealing = Yes) = 2
- Number of negative samples (Appealing = No) = 2
- Probability of positive samples = 2/4 = 0.5
- Probability of negative samples = 2/4 = 0.5

Entropy(Appealing|Taste=Sweet) = -0.5log2(0.5) - 0.5log2(0.5) = 1
Entropy(Appealing|Taste=Sour):

- Total number of samples with Taste=Sour = 3
- Number of positive samples (Appealing = Yes) = 3
- Number of negative samples (Appealing = No) = 0
- Probability of positive samples = 3/3 = 1
- Probability of negative samples = 0

Entropy(Appealing|Taste=Sour) = -1log2(1) - 0log2(0) = 0
Weighted sum of entropies after splitting on Taste:

- Total number of samples = 10
- Number of samples with Taste=Salty = 3
- Number of samples with Taste=Sweet = 4
- Number of samples with Taste=Sour = 3

Entropy(Appealing|Taste) = (3/10)*0 + (4/10)*1 + (3/10)*0 = 0.4
Information gain of Taste:

- Gain(Taste) = Entropy(Appealing) - Entropy(Appealing|Taste) = 1-0.4 = 0.6

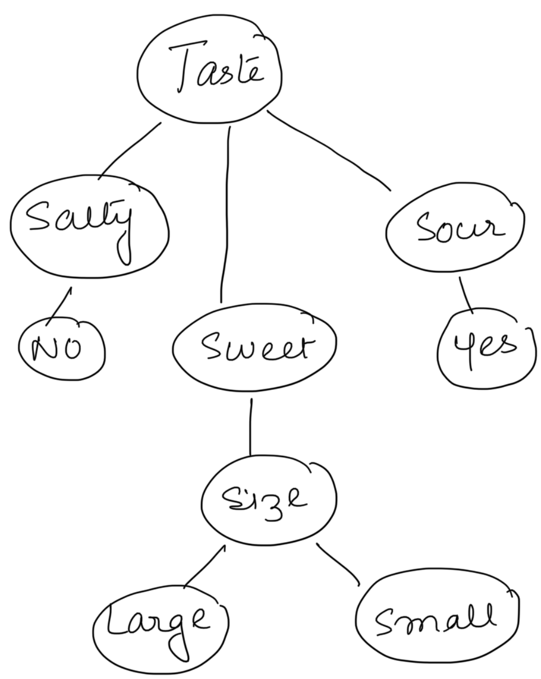
Therefore, the information gain Gain(Taste) at the root node of the decision tree is 0.6.

(b) [10 pts] Suppose we build a decision tree with *Taste* as the attribute to split at the root. How many children does the root have? Which of them requires further splitting and which attribute to use next? Draw this tree.

Answer :

Building the decision tree with Taste as the attribute to split at the root, the root will have three children corresponding to the three possible values of the Taste attribute: Salty, Sweet, and Sour.

The Sweet branch has four dishes (IDs 2, 3, 8, 9). Two of these dishes (IDs 8, 9) are Appealing, and the other two are non-Appealing. Since there are two possible target values in this branch, we need to split this branch further.The resulting decision tree is:



(c) [10 pts] Use the decision tree to predict the class value for the two records given by

ID	Temperature	Taste	Size
11	Hot	Salty	Small
12	Cold	Sweet	Large

Answer :

To predict the class value for the two new records, we can use the decision tree as follows:

For record 11:

The root node splits on Taste, and Taste=Salty for record 11, so we follow the left child to the Salty node.

The Salty node splits on Size, and Size=Small for record 11, so we follow the right child to the leaf node that predicts No. Therefore, the predicted class for record 11 is No.

For record 12:

The root node splits on Taste, and Taste=Sweet for record 12, so we follow the middle child to the Sweet node.

The Sweet node splits on Size, and Size=Large for record 12, so we follow the left child to the leaf node that predicts No. Therefore, the predicted class for record 12 is No.

Therefore, the predicted class values for the two new records are No.

(d) [10 pts] Explain why the decision tree would never choose the same attribute twice along a single path in a decision tree. Note: A single path is a path that starts from the root node and ends at a leaf node.

Answer:

In a decision tree, each internal node represents a split on an attribute, and the edges from the node correspond to the different possible values of that attribute. When constructing a decision tree, the algorithm chooses the attribute that provides the highest information gain (or the lowest Gini index, depending on the algorithm) to split the data at each node.

When a split is made on an attribute at an internal node, the decision tree separates the training data into two or more subsets based on the values of that attribute. Since each internal node represents a different subset of the data, it would be redundant to split on the same attribute again along the same path. This is because splitting on the same attribute again would lead to the same subsets of data and would not provide any additional information gain.

Therefore, the decision tree would never choose the same attribute twice along a single path in a decision tree because it would be redundant and not provide any additional information gain.

Q3. [30 pts] Naive Bayes Classifier

The loan department of a bank has the following past loan processing records, each of which contains an applicant's income, credit history, debt and the final approval decision. These records can serve as training examples to build a decision-making software for a loan advisory system.

ID	Income	Credit History	Debt	Decision
1	0-5K	Bad	Low	Reject
2	0-5K	Good	Low	Approve
3	0-5K	Unknown	High	Reject
4	0-5K	Unknown	Low	Approve
5	0-5K	Unknown	Low	Approve
6	0-5K	Unknown	Low	Reject
7	5-10K	Bad	High	Reject
8	5-10K	Good	High	Approve
9	5-10K	Unknown	High	Approve
10	Over 10K	Unknown	Low	Approve
11	Over 10K	Bad	Low	Reject
12	Over 10K	Good	Low	Approve

We will build a Naive Bayes classifier in this question.

Recall that Naive Bayes inference (Lecture 9, slides 49-50) is based on computing *P*(Cause**|**Effect_1,Effect_2,Effect_3) while assuming that:

(1) Effect_1,Effect_2 and Effect_3 are conditionally independent given Cause; and

(2) *P*(Effect_i **|** Cause) is given for each effect Effect_i as well as *P*(Cause).

To apply Naive Bayes inference here, we can set Cause = Decision, Effect_1 = Income, Effect_2 = Credit History and Effect_3 = Debt. However, we are not explicitly provided *P*(Effect_i **|** Cause) and *P*(Cause).

So we need to estimate these probabilities from the data.

(a) [5 pts] Estimate *P*(Cause = Approve) = **(no. of approved cases / no. of cases)** and *P*(Cause = Reject) = **(no. of rejected cases / no. of cases)** from the data.

Answer :

From the data, we can see that there are a total of 12 loan processing records. Among these, there are 7 records where the decision was to Approve and 5 records where the decision was to Reject.

Thus, P(Cause=Approve) = 7/12

Similarly, P(Cause=Reject) = 5/12

(b) [5 pts] Estimate *P*(Income = *u* | Decision = Approve) = **[no. of approved cases where (**Income = *u***) / no. of approved cases]**; and *P*(Income = *u* | Decision = Reject) = **[no. of rejected cases where (**Income = *u***) / no. of rejected cases]**. Do that for each value of $u \in$ {0–5K, 5–10K, Over10K}.

Answer :

P(Income=0-5K**|**Decision=Approve) = 3/7

P(Income=5-10K|Decision=Approve) = 2/7

P(Income=Over10K|Decision=Approve) = 2/7

P(Income=0-5K|Decision=Reject) = 3/5

P(Income=5-10K|Decision=Reject) = 1/5

P(Income=Over10K|Decision=Reject) = 1/5

(c) [5 pts] Estimate *P*(Credit History = *u* | Decision = Approve) = **[no. of approved cases where (**Credit History = *u***) / no. of approved cases]**; and *P*(Credit History = *u* | Decision = Reject) = **[no. of rejected cases where (**Credit History = *u***) / no. of rejected cases]**. Do that for each value of *u* ∈ {Bad, Good, Unknown}.

Answer:

P(Credit History=Bad|Decision=Approve) = 0/7 = 0

P(Credit History=Good|Decision=Approve) = 3/7

P(Credit History=Unknown|Decision=Approve) = 4/7

P(Credit History=Bad|Decision=Reject) = 3/5

P(Credit History=Good|Decision=Reject) = 0/5 (since there are no rejected cases with Credit History=Good) =0

P(Credit History=Unknown|Decision=Reject) = 2/5

(d) [5 pts] Estimate *P*(Debt = *u* | Decision = Approve) = **[no. of approved cases where (**Debt = *u***) / no. of approved cases]**; and *P*(Debt = *u* | Decision = Reject) = **[no. of rejected cases where (**Debt = *u***) / no. of rejected cases]**. Do that for each value of *u* ∈ {Low, High}.

Answer :

P(Debt=Low|Decision=Approve) = 5/7

P(Debt=High|Decision=Approve) = 2/7

P(Debt=Low|Decision=Reject) = 3/5

P(Debt=High|Decision=Reject) = 2/5

(e) [10 pts] What is the Naive Bayes decision for an applicant who has 4K annual income, a good credit and a high amount of debt?

Answer:

To make a decision using Naive Bayes, we need to calculate the probability of each class (approve or reject) given the values of the applicant's income, credit history, and debt. Then, we choose the class with the highest probability as our decision.

Calculating the conditional probabilities for each value of income, credit history, and debt, given the class (approve or reject):

P(Income=4K|Decision=Approve) = 3/7

P(Income=4K|Decision=Reject) = 3/5

P(Credit History=Good|Decision=Approve) = 3/7

P(Credit History=Good|Decision=Reject) = 0

P(Debt=High|Decision=Approve) = 2/7

P(Debt=High|Decision=Reject) = 2/5

Calculating the prior probabilities for each class:

P(Decision=Approve) = 7/12

P(Decision=Reject) = 5/12

Using Bayes' rule to calculate the posterior probabilities for each class:

P(Decision=Approve|Income=4K, Credit History=Good, Debt=High) α P(Income=4K|Decision=Approve) * P(Credit History=Good|Decision=Approve) * P(Debt=High|Decision=Approve) * P(Decision=Approve)

= 3/7 * 3/7 * 2/7 * 7/12

= 0.0306

P(Decision=Reject|Income=4K, Credit History=Good, Debt=High) α P(Income=4K|Decision=Reject) * P(Credit History=Good|Decision=Reject) * P(Debt=High|Decision=Reject) * P(Decision=Reject)

= 3/5 * 0 * 2/5 * 5/12

= 0

Since both posterior probabilities of Approve is greater with the value of 0.306, Naive Bayes decision for an applicant who has 4K annual income, a good credit and a high amount of debt is to approve.

Q4. [10 pts] Perceptron

Consider two perceptron units $\theta_1(\mathbf{x}) = w_1 \cdot \tanh(\mathbf{a}^\top\mathbf{x} + b) + h_1$ and $\theta_2(\mathbf{x}) = w_2 \cdot \mathrm{sigmoid}(2\mathbf{a}^\top\mathbf{x} + 2b) + h_2$ where $\mathrm{sigmoid}(z) = 1/(1 + e^{-z})$ and $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$.

(a) [5 pts] Show that $\tanh(z) = 2\mathrm{sigmoid}(2z) - 1$.

Answer:

To show that tanh(z) = 2sigmoid(2z) - 1, we can start by expressing sigmoid(2z) in terms of tanh(z) as follows:

sigmoid(2z) = 1/(1 + e^(-2z)) [Definition of sigmoid function]

= (1 + e^(-z - z))/(1 + e^(-z - z)) [Multiplying numerator and denominator by e^(z)]

= (e^z + e^(-z))/(e^z + e^(-z) + e^(-z - z)) [Expanding the numerator]

= (e^z + e^(-z))/(e^z + e^(-z) + e^(-2z)) [Simplifying the denominator]

Now, we can substitute this expression for sigmoid(2z) into the right-hand side of the equation we want to prove:

2sigmoid(2z) - 1 = 2((e^z + e^(-z))/(e^z + e^(-z) + e^(-2z))) - 1

= (2e^z + 2e^(-z))/(e^z + e^(-z) + e^(-2z)) - 1

= (2e^z + 2e^(-z) - e^z - e^(-z) - e^(-2z))/(e^z + e^(-z) + e^(-2z))

= (e^z - e^(-z))/(e^z + e^(-z) + e^(-2z)) [Simplifying the numerator]

= (e^(2z) - 1)/(e^(2z) + 1 + 1) [Multiplying numerator and denominator by e^(2z)]

= (e^(2z) - 1)/(e^(2z) + 2)

Now, we can show that this expression is equal to tanh(z) by expressing tanh(z) in a similar way:

tanh(z) = (e^z - e^(-z))/(e^z + e^(-z))

= (e^(2z) - 1)/(e^(2z) + 1) [Multiplying numerator and denominator by e^(2z)]

Since (e^(2z) - 1)/(e^(2z) + 2) = (e^(2z) - 1)/(e^(2z) + 1) · (e^(2z) + 1)/(e^(2z) + 2) = tanh(z), we have shown that tanh(z) = 2sigmoid(2z) - 1, as required.

(b) [5 pts] Given $w_1 = 2$ and $h_1 = 5$, find the values of w_2 and h_2 such that for any input $\mathbf{x}$, the outputs produced by θ_1 and θ_2 are always the same (regardless of how we set $\mathbf{a}$ and b).

Answer:

To find the values of w2 and h2 such that the outputs of θ1 and θ2 are always the same, we need to equate the expressions for θ1(x) and θ2(x) and solve for w2 and h2.

$\theta 1(x) = w1 \cdot \tanh(a^\top x + b) + h1$

$\theta 2(x) = w2 \cdot \text{sigmoid}(2a^\top x + 2b) + h2$

We can start by equating the activation functions, tanh and sigmoid:

$w1 \cdot \tanh(a^\top x + b) + h1 = w2 \cdot \text{sigmoid}(2a^\top x + 2b) + h2$

Now, we can substitute w1 and h1 with their given values:

$2 \cdot \tanh(a^\top x + b) + 5 = w2 \cdot \text{sigmoid}(2a^\top x + 2b) + h2$

Next, we need to eliminate the dependence on x, which means that the expression on the right-hand side must be a constant. We can achieve this by setting a = 0 and b = 0, which gives:

$2 \cdot \tanh(0) + 5 = w2 \cdot \text{sigmoid}(0) + h2$

Simplifying this expression, we get:

$2 + 5 = w2/2 + h2$

$w2/2 + h2 = 7$

Now, we need one more equation to solve for both w2 and h2. We can achieve this by setting x = 1, which gives:

$2 \cdot \tanh(a^\top + b) + 5 = w2 \cdot \text{sigmoid}(2a^\top + 2b) + h2$

Substituting a = 0 and b = 0, we get:

$2 \cdot \tanh(0) + 5 = w2 \cdot \text{sigmoid}(0) + h2$

$2 + 5 = w2/2 + h2$

$w2/2 + h2 = 7$

Now, we have two equations in two unknowns, w2 and h2:

$w2/2 + h2 = 7$

$w2 + 5 = w2/2 + h2$

Solving this system of equations, we get:

$w2 = 4$

$h2 = 3$

Therefore, if we set w2 = 4 and h2 = 3, then for any input x, the outputs produced by θ1 and θ2 will always be the same.