School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (2021)

Workshop: Week 9

1. For the following dataset:

ID	*Outl*	*Temp*	*Humi*	*Wind*	PLAY
TRAINING INSTANCES					
A	s	h	h	F	N
B	s	h	h	T	N
C	o	h	h	F	Y
D	r	m	h	F	Y
E	r	c	n	F	Y
F	r	c	n	T	N
TEST INSTANCES					
G	o	c	n	T	?
H	s	m	h	F	?

Classify the test instances using the **ID3 Decision Tree** method:

a) Using the **Information Gain** as a splitting criterion

For Information Gain, at each level of the decision tree, we're going to choose the attribute that has the largest difference between the entropy of the class distribution at the parent node, and the average entropy across its daughter nodes (weighted by the fraction of instances at each node);

$$IG(A|R) = H(R) - \sum_{i \in A} P(A = i)H(A = i)$$

In this dataset, we have 6 instances total — 3 Y and 3 N. The entropy at the top level of our tree is $H(R) = -\left[\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right]$

This is a very even distribution. We're going to hope that by branching the tree according to an attribute, that will cause the daughters to have an uneven distribution - which means that we will be able to select a class with more confidence - which means that the entropy will go down.

For example, for the attribute `Outl`, we have three attribute values: `s, o, r`.

- When `Outl=s`, there are 2 instances, which are both N. The entropy of this distribution is $H(O = s) = -\left[0\log_2 0 + \frac{2}{2}\log_2\frac{2}{2}\right] = 0$. Obviously, at this branch, we will choose N with a high degree of confidence.
- When `Outl=o`, there is a single instance, of class Y. The entropy here is going to be 0 as well.
- When `Outl=r`, there are 2 Y instances and 1 N instance. The entropy here is $H(o = r) = -\left[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] \approx 0.9183$

To find the average entropy (the "mean information"), we sum the calculated entropy at each daughter multiplied by the fraction of instances at that daughter: $MI(O) = \frac{2}{6}(0) + \frac{1}{6}(0) + \frac{3}{6}(0.9183) \approx 0.4592$

The overall Information Gain here is $IG(O) = H(R) - MI(O) = 1 - 0.4592 = 0.5408$.

	R	*Outl*			Temp			H		*Wind*		ID					
		s	o	r	h	m	c	h	n	T	F	A	B	C	D	E	F
Y	3	0	1	2	1	1	1	2	1	0	3	0	0	1	1	1	0
N	3	2	0	1	2	0	1	2	1	2	1	1	1	0	0	0	1
Total	6	2	1	3	3	1	2	4	2	2	4	1	1	1	1	1	1
P(Y)	1/2	0	1	2/3	1/3	1	1/2	1/2	1/2	0	3/4	0	0	1	1	1	0
P(N)	1/2	1	0	1/3	2/3	0	1/2	1/2	1/2	1	1/4	1	1	0	0	0	1
H	1	0	0	0.9183	0.9183	0	1	1	1	0	0.8112	0	0	0	0	0	0
MI		0.4592			0.7924			1		0.5408		0					
IG		0.5408			0.2076			0		0.4592		1					
SI		1.459			1.459			0.9183		0.9183		2.585					
GR		0.3707			0.1423			0		0.5001		0.3868					

The table above lists the Mean Information and Information Gain, for each of the 5 attributes.

At this point, `ID` has the best information gain, so hypothetically we would use that to split the root node. At that point, we would be done, because each daughter is purely of a single class — however, we would be left with a completely useless classifier! (Because the IDs of the test instances won't have been observed in the training data.)

Instead, let's take the second-best attribute:`Outl`.

There are now three branches from our root node: for s, for o, and for r. The first two are pure, so we can't improve them anymore. Let's examine the third branch (`Outl=r`):

- Three instances (D, E, and F) have the attribute value r; we've already calculated the entropy here to be 0.9183.
- If we split now according to Temp, we observe that there is a single instance for the value m (of class N, the entropy is clearly 0); there are two instances for the value c, one of class Y and one of class N (so the entropy here is 1). The mean information is $\frac{1}{3}(0) + \frac{2}{3}(1) \approx 0.6667$, and the information gain at this point is 0.9183 – 0.6667 ≈0.2516.
- For `Humi`, we again have a single instance (with value `h`, class Y, H = 0), and two instances (of `n`) split between the two classes (H = 1). The mean information here will also be 0.6667, and the information gain 0.2516.
- For `Wind`, there are two `F` instances, both of class Y (H = 0), and one `T` instance of class N (H = 0). Here, the mean information is 0 and the information gain is 0.9183.
- `ID` would still look like a good attribute to choose, but we'll continue to ignore it.
- All in all, we will choose to branch based on Wind for this daughter.

All of the daughters of `r` are pure now, so our decision tree is complete:

- `Outl=o` U`(Outl=r` ∩ `Wind=F)` → `Y` (so we classify `G` as Y)
- `Outl=s` U`(Outl=r` ∩ `Wind=T)` → `N` (so we classify `H` as N)

b) Using the **Gain Ratio** as a splitting criterion

Gain ratio is similar, except that we're going to weight down (or up!) by the "split information" — the entropy of the distribution of instances across the daughters of a given attribute.

For example, we found that, for the root node, `Outl` has an information gain of 0.5408. There are 2 (out of 6) instances at the s daughter, 1 at the o daughter, and 3 at the r daughter.

The split information for `Outl` is $SI(o) = -\left[\frac{2}{6}\log_2\frac{2}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{3}{6}\log_2\frac{3}{6}\right] \approx 1.459$.

The Gain ratio is consequently $GR(o) = \frac{IG(o)}{SI(o)} \approx \frac{0.5408}{1.459} \approx 0.3707$

The values for split information and gain ratio for each attribute at the root node are shown in the table above. The best attribute (with the greatest gain ratio) at the top level this time is `Wind`.

`Wind` has two branches: T is pure, so we focus on improving `F` (which has 3 Y in- stances (C, D, E), and 1 N instance (A)). The entropy of this daughter is 0.8112.

- For `Outl`, we have a single instance at s (class N, H = 0), a single instance at `o` (class Y, H = 0), and 2 Y instances at `r` (H = 0). The mean information here is clearly 0; the information gain is 0.8112. The split information is $SI(o|(W = F)) = -\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1.5$, so the gain ratio is $GR\big(o\big|(W = F)\big) = \frac{0.8112}{1.5} \approx 0.5408$
- For `Temp`, we have two `h` instances (one Y and one N, so H = 1), a single `m` instance (Y, H = 0), and a single `c` instance (Y, H = 0). The mean information is $\frac{2}{4}(1) + \frac{1}{4}(0) + \frac{1}{4}(0) = 0.5$, so the information gain is 0.8112-0.5 = 0.3112. The distribution of instances here is the same as Outl, so the split information is also 1.5, and the gain ratio is $GR\big(T\big|(W = F)\big) = \frac{0.3112}{1.5} \approx 0.2075$
- For `Humi`, we have 3 `h` instances (2 Y and 1 N, H = 0.9183), and 1 `n` instance (Y, H = 0): the mean information is $\frac{3}{4}$ (0.9183) + $\frac{1}{4}$(0) = 0.6887 and the information gain is 0.8112 – 0.6887 = 0.1225. The split information is $SI(H|(W = F)) = -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] \approx 0.8112$, so the gain ratio is $GR\big(H\big|(W = F)\big) = \frac{0.1225}{0.8112} \approx 0.1387$.
- For `ID`, the mean information is obviously still 0, so the information gain is 0.8112. The split information at this point is $-\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] = 2$, so the gain ratio is approximately 0.4056.

Of our four choices at this point, `Outl` has the best gain ratio. The resulting daughters are all pure, so the decision tree is finished:

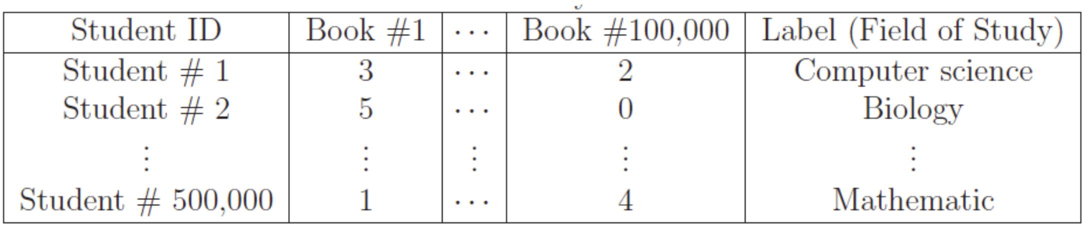
- `Wind=F ∩(Outl=o ∪ Outl=r) → Y`
- `Wind=T ∪(Wind=F ∩ Outl=s) → N` (so we classify G and H as N)

Note that this decision tree is superficially similar to the tree above but gives different classifications because of the order in which the attributes are considered.

Note also that we didn't need to explicitly ignore the `ID` attribute for Gain Ratio (as we needed to do for Information Gain) — the split information pushed down its "goodness" to the point where we didn't want to use it anyway!

2. Imagine you are given a dataset from the university's library, and your job is to build a classification model that classify students based on the list of books that they borrowed.

   The dataset includes the list of books available in the library (columns) and the students who borrowed them (rows), and the ranking for each item (ranking value is between 0–5, 0 if the book was not borrowed and 1–5 indicates the student's interest). The metadata for the books (e.g. titles) are not readily available to us, we just have the book IDs (e.g. Book #i). The dataset also includes the students' field of study (in total there are 10 fields), which can be used for the classification task. Answer the following questions, considering that there are 500,000 students and 100,000 books in this dataset.



Student ID	Book #1	· · ·	Book #100,000	Label (Field of Study)
Student # 1	3	· · ·	2	Computer science
Student # 2	5	· · ·	0	Biology
⋮	⋮	⋮	⋮	⋮
Student # 500,000	1	· · ·	4	Mathematic

(i). Consider the following supervised machine learning methods, and for each one, explain why it would be appropriate or inappropriate to use for this problem:

i. Naïve Bayes

Too many attributes, most probabilities based on numeric values will be essentially random. Naïve Bayes is oversensitive to redundant and/or irrelevant attributes.

ii. k-NN

Too many dimensions, similarities are mostly meaningless. Also due to high dimension of the instances (100,000), calculating the similarity would be computationally heavy.

iii. Decision Tree

Too many attributes. With too many nodes Decision Tree is in danger of overfitting.

(ii). Would "feature selection" be useful here? Explain why, by referring to a single machine learning method.

Feature selection is expected to improve any of the above approaches.

The distances used in K-NN become meaningless in high-dimensional space (intuitively, everything is far away from everything). So, feature selection is very important.

Naive Bayes is slightly more robust to many / irrelevant features, but since it does not have an embedded feature weighting mechanism (unlike e.g., logistic regression) it is expected to improve.

Decision trees are prone to overfitting, and we know that feature selection can help in this case. Decision trees are, in fact, have an *embedded method* for feature selection -- they perform selection as part of the model. For example, you can prune the decision tree ('chop off' branches with low IG) to rid yourself of unpredictive features and get a more robust model.

(iii). Explain how you would evaluate the effectiveness of your system: you should briefly describe an evaluation strategy and an evaluation metric that are suitable for this data. What might be an example of a baseline?

- Evaluation strategy: Cross-validation - partition the data into M (approximately) equal size partitions, train the system M times and the average performance is computed across the M runs.
- Evaluation metric: F1-score or accuracy
- Baseline: zero-R - classify all instances according to the most common class in the training data. OR Random Baseline - randomly assign a class to each test instance