Homework 2 Solution

October 21, 2018

1. (a) Compute the Entropy for the overall collection of training examples.

$$\begin{aligned} \text{Entropy}(t) &= -\,\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t) \\ &= -\tfrac{10}{20}\log_2\tfrac{10}{20} - \tfrac{10}{20}\log_2\tfrac{10}{20} \\ &= 1 \end{aligned}$$

   (b) Compute the Entropy for Movie ID attribute.
   The Movie ID attribute split all the records into 20 separate leaf node, for each node t_i, the Entropy is like:

$$\begin{aligned} \text{Entropy}(t_i) &= -\,\text{p}(0|t_i)\log_2\text{p}(0|t_i) - \text{p}(1|t_i)\log_2\text{p}(1|t_i) \\ &= -\log_2 1 \\ &= 0 \end{aligned}$$

   So the total weighted entropy of leaf nodes is 0.

   (c) Compute the Entropy for the Format attribute.
   The Format attribute split all the records into 2 leaf node, we compute separately.

      i. for "Onine" Node

$$\begin{aligned} \text{Entropy}(t) &= -\,\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t) \\ &= -\tfrac{4}{12}\log_2\tfrac{4}{12} - \tfrac{8}{12}\log_2\tfrac{8}{12} \\ &= 0.91829583405448956 \end{aligned}$$

      ii. for "DVD" Node

$$\begin{aligned} \text{Entropy}(t) &= -\,\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t) \\ &= -\tfrac{6}{8}\log_2\tfrac{6}{8} - \tfrac{2}{8}\log_2\tfrac{2}{8} \\ &= 0.81127812445913283 \end{aligned}$$

   The weighted sum of the entropy is

$$\begin{aligned} \textstyle\sum_{i=1}^{k}\frac{N(t_i)}{N}I(t_i) &= \tfrac{12}{20}\times 0.91829583405448956 + \tfrac{8}{20}\times 0.81127812445913283 \\ &= 0.8754887502163469 \end{aligned}$$

   (d) Compute the Entropy for the Movie Category attribute using multiway split.
   The Movie Category split the records into 3 separate leaf nodes.

      i. for "Entertainment".

$$\begin{aligned} \text{Entropy}(t) &= -\,\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t) \\ &= -\tfrac{1}{4}\log_2\tfrac{1}{4} - \tfrac{3}{4}\log_2\tfrac{3}{4} \\ &= 0.81127812445913283 \end{aligned}$$

      ii. for "Comedy".

$$\begin{aligned} \text{Entropy}(t) &= -\,\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t) \\ &= -\tfrac{1}{8}\log_2\tfrac{1}{8} - \tfrac{7}{8}\log_2\tfrac{7}{8} \\ &= 0.5435644431995964 \end{aligned}$$

iii. for "Documentaries".

$$\begin{aligned}\text{Entropy}(t) &= -\text{p}(0|t)\log_2\text{p}(0|t) - \text{p}(1|t)\log_2\text{p}(1|t)\\ &= -\tfrac{2}{8}\log_2\tfrac{2}{8} - \tfrac{6}{8}\log_2\tfrac{6}{8}\\ &= 0.81127812445913283\end{aligned}$$

The weighted sum of the entropy is

$$\begin{aligned}\textstyle\sum_{i=1}^{k}\frac{N(t_i)}{N}I(t_i) &= \tfrac{4}{20}\times 0.81127812445913283 + \tfrac{8}{20}\times 0.5435644431995964\\ &\quad +\tfrac{8}{20}\times 0.81127812445913283\\ &= 0.7041926519553183\end{aligned}$$

(e) Which of the three attributes has the lowest Entropy?
The Movie ID has the lowest entropy

(f) Which of the three attributes will you use for splitting at the root node? Briefly explain your choice.
Considering the gain, the criterion that can be used to determine the goodness of a split

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k}\frac{N(v_j)}{N}I(v_j)$$

The best split is using the Movie Category attribute.

2. (a) Estimate the generalization error rate of the tree using both the optimistic approach and the pessimistic approach. While computing the error with pessimistic approach, to account for model complexity, use a penalty value of 2 to each leaf node.

i. Optimistic Error Estimate: if we look at the table about training set, we can see there is no error, so the training error rate is 0.

ii. Pessimistic Error Estimate:

$$e_g(T) = \frac{\sum_{i=1}^{6}[e(t_i)+\Omega(t_i)]}{\sum_{i=1}^{6}n(t_i)} = \frac{6\times 2}{15} = 0.8$$

(b) Compute the error rate of the tree on the test set shown in Table 3.
There are 7 errors in 15 records totally, so the error rate on the test data is $\frac{7}{15} = 0.4666667$

(c) Comment on the behavior of training and test set errors with respect to model complex- ity. Comment on the utility of incorporating model complexity in building a predictive model.

i. The model performance on training set quite well, since there is 0 error on training data, but when applied to the test data, its error rate is as high as 46.7%. Considering the model's generalization error(pessimistic approach) 0.8, this over-fitting happens because the model is too complex, which makes it lost the generality.

ii. When building a model for predicting, the complexity of the model is as important as the training set error. We need to find a balance between these two. Comparing the Pessimistic Error Estimate is a good way to select model, since it take both the complexity and the training error into account, help us to avoid building a over-fitting model with much complexity.

3. Given the data sets shown in Figure 2, explain how the decision tree and k-nearest neighbor (k-NN) classifiers would perform on these data sets.

Answer: We will dicuss separately.

(a) For the synthetic data 1, we notice there is a small group of Discriminating Attributes, which have clear patterns relative to the true class label, there are also much more noise attributes, which didn't provide usable information to predict. In such case, **Decision Tree** is more suitable since while building a decision tree, measures for selecting best splitting (like Δ_{info})

will greatly help us to find the split based on Discriminating Attributes avoiding using the noise attributes. In contrast, the **K-Nearest Neighbor** is not suitable. There are much more noise attributes than the discriminating ones, as k-NN will use all the attributes to make a predication, its judgment will be more affected by the noise attributes, which make k-NN perform badly here.

(b) For the synthetic data 2, **K-Nearest Neighbor** is more suitable, since there is a clear spatial pattern in the figure. With a enough small k, the k-NN classifier will give a right prediction. In contrast, it's difficult to apply the **Decision Tree** here, suppose we use attribute x,y to build a tree, it's more likely first we split with $x \leq x_0$ and $x > x_0$, then we split with $y \leq y_0$ and $y > y_0$, the data suitable for such classification is always a rectangle, since now the boundaries between various classification is something like ellipses, $(\frac{x-x_0}{a})^2 + (\frac{y-y_0}{b})^2 = c^2$, the decision tree will perform worse than k-NN here.

4. Answer the following questions. Make sure to provide a brief explanation or an example to illustrate the answer.

   (a) Are the rules mutually exclusive ?

   Answer: No. If a record satisfies "Home Owner = No" and "Marital Status = Single", it will trigger both the second and last rules.

   (b) Is the rule set exhaustive ?

   Answer: No. If there is a record whose attributes satisfies "Home Owner = No, Marry Status = Divorced, Annual Income = High, Currently Employed = Yes", no rule listed will cover this record.

   (c) Is ordering needed for this set of rules ?

   Answer: Yes, this set of rules need ordering, because this set of rules is not mutually exclusive, if a record is covered by several rules, some of them may predict conflicting classes. Considering a record which satisfies "Home Owner = No, Marry Status = Married, Annual Income = High, Currently Employed = No", according to rule 4, the predict is Yes, according to rule 6, the predict is No.

   (d) Do you need a default class for the rule set?

   Answer: Yes, because the set of rules is not a exhaustive one, so there exist records which are not covered by the rules. To cover such records, a default class is needed.

5. Consider the problem of predicting whether a movie is popular given the following attributes: Format (DVD/Online), Movie Category (Comedy/Documentaries), Release Year, Number of world-class stars, Director, Language, Expense of Production and Length. If you had to choose between RIPPER and a k-nearest neighbor classifier, which would you prefer and why? Briefly explain why the other one may not work so well?

   Answer: I prefer the **RIPPER** classifier. Here is why :

   (a) Thinking of the data of movies here, it's a simple data set, with a few attributes, so RIPPER will produce a more descriptive model which are easier to interpret.

   (b) It's true that we really don't know the class label boundaries of the data and usually rule-based classifier usually create rectilinear partitions, but if the rule-based classifier allows multiple rules to be triggered for a given record, then RIPPER can also construct a more complex decision boundary.

   (c) RIPPER is more robust to the presence of noise. As we don't know how much noise existed in the training data, a noise-proof classifier is a first choice.

   (d) Presence of redundant attributes doesn't adversely affect the accuracy of RIPPER, there might exist redundant features here, so a classifier robust with this is preferred.

   (e) Once a model is built by RIPPER, it's extremely fast to calculate the class label of a record.

   Compared with RIPER, k-NN will build arbitrarily shaped decision boundaries but have following concerns

(a) Before applying k-NN, an appropriate proximity must be found, and also data preprocessing need to be taken.

(b) k-NN is susceptible to noise. As we don't know how much noise existed in the training data, it take a risk to apply k-NN here.

(c) k-NN is susceptible to redundant attributes, considering the synthetic data in Question 4.

(d) k-NN is time consuming when used to predict the class label of a record compared to RIPPER.

Conclusion: according to current information of the data, RIPPER will give us a more interpretable and robust model.