Solution Sheet 10

Total points: 30

Homework Exercises

H10.1 Growing a Decision Tree by Hand [9 pts.]

In this exercise, you will predict the risk class of the stocks of companies in the German industry. For this, you will step-by-step grow a decision tree by hand. You are provided with data with the following attributes:

Attribute	Values	Description
sector	A, F, T	industry of the company: automotive, finance, technology
age	o, y	age of the company: old (founded before 1980), young (founded in 1980 or after)
employees	s, b	number of employees: small ($\leq 100,000$) or big ($> 100,000$)
risk (target)	H, L	risk class of the stock: H high risk, L low risk

The analyses should be done based on the following dataset:

ID	Sector	Age	Employees	Risk
1	F	y	b	H
2	T	y	b	L
3	A	y	s	H
4	F	o	b	L
5	T	y	b	H
6	F	y	b	H
7	A	o	b	L
8	A	o	s	L

(a) Calculate the information gain for each of the attributes *sector*, *employees* and *age* independently. Which attribute is the most important?

(b) Assume that the data is split into the classes of the most important attribute from part (a). Now, find the second most important feature. Hint: calculate the information gain with respect to one and then the other remaining attribute in the two splits from (a). Then, calculate the overall information gain for both cases to find the better attribute.

(c) Based on your results, construct a level-2 decision tree. Are all data instances in the terminal leaves of the same risk class?

Solution:

Recall that *information gain* is a measure for the decrease of disorder achieved by a data split on some attribute. To quantify the disorder, we use the *entropy* H:

$$H(S) \equiv \text{entropy}(S) = -\sum_{i=1}^{N} p_i \log_2(p_i).$$

Here, the total set S is partitioned into N subsets $S_1, \ldots, S_N$ with relative sizes $p_1, \ldots, p_N$, i.e. $p_1 = |S_1|/|S|, \ldots, p_N = |S_N|/|S|$. Usually, the partitioning is based on a target characteristic in the data (in this case, the risk class).

In the case that the target is binary, the entropy can be expressed in a simpler way:

$$H(n_+, n_-) = -\frac{n_+}{n}\log_2(\frac{n_+}{n}) - \frac{n_-}{n}\log_2(\frac{n_-}{n}).$$

We introduced $_+$, n_-, and n for the numbers of positive, negative, and total numbers of samples.

Information gain with respect to a feature A is then

$$\text{gain}(S, A) = H(S) - \sum_{j=1}^{N_A} \frac{|S_{A,j}|}{|S|} H(S_{A,j})$$

where $S_{A,1}, \ldots, S_{A,N_A}$ are the subsets of the data induced by the partitioning on the attribute A with N_A different values, and $H(S_{A,j})$ are the entropies with respect to the target characteristics in the subsets. By the summation, we build an average of the entropies in those subsets where the relative subset sizes serve as the weights. This exercise helps us to understand this better.

(a) ✓✓✓✓ First, we have to determine $H(S)$, the *baseline entropy* in the total data set. The data size is $|S| = 8$, comprising $|S_1| = 4$ low risk and $|S_2| = 4$ high risk samples. Thus,

$$H(S) = -\frac{4}{8}\log_2(\frac{4}{8}) - \frac{4}{8}\log_2(\frac{4}{8}) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = -\log_2(\frac{1}{2}) = 1.$$

This is in accordance with the statement from the lecture: a split into two equal-sized subsets refers to the entropy 1, $H(0.5, 0.5) = 1$.

Now we have to calculate the *average entropy* for splitting on each of the three attributes. When splitting by *sector*, we have all instances from the automotive sector $|S_{\text{sector},1}| = 3$, all from the financial sector $|S_{\text{sector},2}| = 3$, and all from the technology sector $|S_{\text{sector},3}| = 2$. The proportions of low and high-risk samples, respectively, in these subsets are 2/3 and 1/3 for class A, 1/3 and 2/3 for class F, and 1/2 and 1/2 for class T. This leads to the entropies in the subsets

$$H(S_{\text{sector},1}) = -\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3}) \approx 0.918,$$
$$H(S_{\text{sector},2}) = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) \approx 0.918,$$
$$H(S_{\text{sector},3}) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1.$$

With this, the average entropy is

$$\sum_{j=1}^{3} \frac{|S_{\text{sector},j}|}{|S|} H(S_{\text{sector},j}) \approx \frac{3}{8}\cdot 0.918 + \frac{3}{8}\cdot 0.918 + \frac{2}{8}\cdot 1 \approx 0.939.$$

Hence, information gain achieved by a split on *sector* is

$$\text{gain}(S, \text{sector}) = 1 - 0.939 \approx 0.061.$$

The table below brings this in a compact form.

Subset	Sector	$\lvert S_{\text{sector},j}\rvert$	risk	proportion	entropy
$S_{\text{sector},1}$	A	3	L	2/3	0.918
			H	1/3	
$S_{\text{sector},2}$	F	3	L	1/3	0.918
			H	2/3	
$S_{\text{sector},3}$	T	2	L	1/2	1
			H	1/2	

Proceeding with the split based on *employees*, we see that there are $|S_{\text{employees},1}| = 2$ instances with a small number of employees, of which 1/2 are with low and 1/2 with high risk. The remaining companies with big number of employees $|S_{\text{employees},2}| = 6$ are by 3/6 of low and by 3/6 of high risk. This brings the entropies

$$H(S_{\text{employees},1}) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1,$$

$$H(S_{\text{employees},2}) = -\frac{3}{6}\log_2(\frac{3}{6}) - \frac{3}{6}\log_2(\frac{3}{6}) = 1.$$

Thus, the information gain by splitting on the number of employees is

$$\text{gain}(S, \text{employees}) = 1 - \frac{1}{2}\cdot 1 - \frac{1}{2}\cdot 1 = 1 - 1 = 0,$$

i.e. there is *no* gain in information. This is also intuitive because the proportion of high-risk companies is the same in the two found sets $S_{\text{employees},1}$ and $S_{\text{employees},2}$ (and hence also the same as in the total data). In the one split, we know as much about the risk as we do in the other. If somebody told us a company's number of employees without any further information, we cannot conclude that the company has an over- or underproportional risk.

Subset	Employees	$\lvert S_{\text{employees},j}\rvert$	risk	proportion	entropy
$S_{\text{employees},1}$	s	2	L	1/2	1
			H	1/2	
$S_{\text{employees},2}$	b	6	L	3/6	1
			H	3/6	

The last attribute is *age*. Here, we have $|S_{\text{age},1}| = 3$ well-tried old companies in the data set, all of low risk and none of high risk. The $|S_{\text{age},2}| = 5$ younger companies are by 1/5 of low and by 4/5 of high risk. Again, we can calculate the entropies

$$H(S_{\text{age},1}) = -\frac{3}{3}\log_2(\frac{3}{3}) = 0,$$

$$H(S_{\text{age},2}) = -\frac{1}{5}\log_2(\frac{1}{5}) - \frac{4}{5}\log_2(\frac{4}{5}) \approx 0.722.$$

An entropy of 0 means that there is *no* disorder or *perfect* order in the set. This is the case if a split contains only instances of the same characteristic, i.e., if it is pure instead of being mixed: $H(1, 0) = 0$ (in the binary case). Note also that we have only one summand in $H(S_{\text{age},1})$ – the set of high-risk is empty and hence does not have to be considered. It follows for the information gain with respect to *age*:

$$\text{gain}(S, \text{age}) \approx 1 - \frac{3}{8}\cdot 0 - \frac{5}{8}\cdot 0.722 \approx 1 - 0.451 = 0.549.$$

Subset	Age	$\lvert S_{\text{age},j}\rvert$	risk	proportion	entropy
$S_{\text{age},1}$	p	3	L	3/3	0
			H	(0/3)	
$S_{\text{age},2}$	b	5	L	1/5	0.722
			H	4/5	

In summary, we gain the highest amount of information by the split based on *age*. This is the most important attribute, and we will use it for the first branch of the tree.

(b) ✓✓✓ After splitting on *age*, we have a branch of old companies that all have low risk, referring to entropy 0. This branch is already pure and hence does not have to be split further – we already safely know about the risk in this case (safely on the level of the data).

So, we will restrict the consideration to the subset of the 5 young companies, denoted as $\tilde{S}$. These are by 1/5 low-risk and by 4/5 high-risk companies. Hence, the baseline entropy in this subset is

$$H(\tilde{S}) = H(1,4) = -\frac{1}{5}\log_2(\frac{1}{5}) - \frac{4}{5}\log_2(\frac{4}{5}) \approx 0.722$$

which we need for the next gain in information. It is important that we do not use a disorder that refers to all data points here (even if we already subtracted the information gain by the first split) since we are in the subset. In a more general situation, we could also need a second splitting of the *age*=o branch, which then also should be treated separately. Here, it could happen that we find different second most important attributes in the two subbranches, e.g., to split by *employees* after *age*=o but by *sector* after *age*=y.

Now, if we consider *sector* first, we come to the table

Subset	Sector	$\lvert\tilde{S}_{\text{sector},j}\rvert$	risk	proportion	entropy
$\tilde{S}_{\text{sector},1}$	A	1	L	0/1	0
			H	1/1	
$\tilde{S}_{\text{sector},2}$	F	2	L	0/2	0
			H	2/2	
$\tilde{S}_{\text{sector},3}$	T	2	L	1/2	1
			H	1/2	

where we marked the sets by "tilde" ˜ to indicate that we restrict the data to the young companies. Similar to part (a), the average entropy is

$$\sum_{j=1}^{3} \frac{|\tilde{S}_{\text{sector},j}|}{|\tilde{S}|} H(\tilde{S}_{\text{sector},j}) \approx \frac{1}{5}\cdot 0 + \frac{2}{5}\cdot 0 + \frac{2}{5}\cdot 1 = 0.400.$$

Hence, information gain achieved by a further split on *sector* is

$$\text{gain}(\tilde{S}, \text{sector}) \approx 0.722 - 0.400 \approx 0.322.$$

With respect to *employees*, the young companies expand to

Subset	Employees	$\lvert\tilde{S}_{\text{employees},j}\rvert$	risk	proportion	entropy
$\tilde{S}_{\text{employees},1}$	s	1	L	0/1	0
			H	1/1	
$\tilde{S}_{\text{employees},2}$	b	4	L	1/4	0.811
			H	3/4	

.

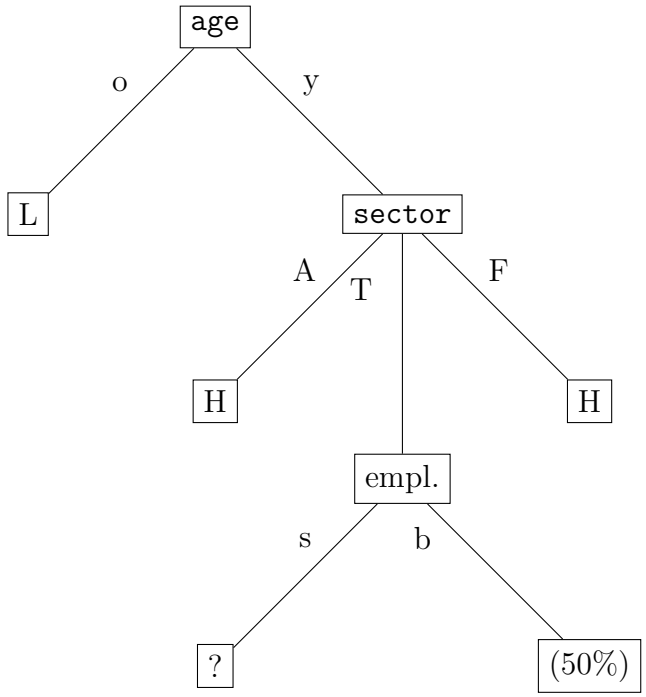
This yields

$$\text{gain}(\tilde{S}, \text{employees}) \approx 0.722 - \frac{1}{5} \cdot 0 - \frac{4}{5} \cdot 0.811 = 0.073.$$

Comparing these two possible splits, we see that *sector* achieves a higher gain in information. This is because it leads to 3/5 of the remaining data with perfect order and 2/5 of entropy 1, while *employees* reaches only a proportion of 1/5 with perfect order and 4/5 of entropy 1. Obviously, *sector* is the second most important attribute.

As a general comment, note that we actually do not need the baseline entropies if we were only to decide on the attribute *importance* since we will always subtract from the same baseline at one decision step. Instead of picking the attribute with maximum information gain, we could instead choose the one with minimum average entropy. Here, however, we were explicitly asked for information gain.

(c) ✓✓ The previous results lead to the following decision tree:

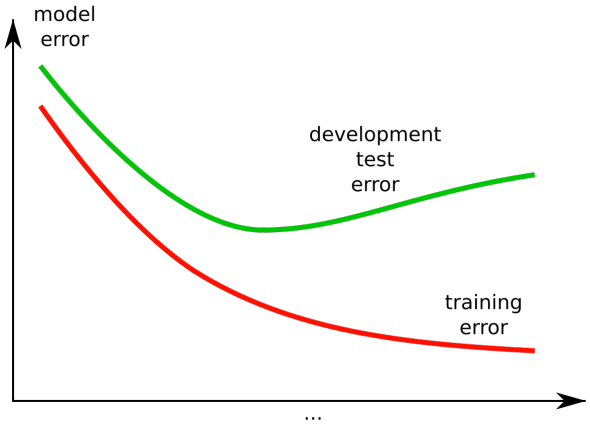


It was explained already before that the branch "old" is pure. The sub-branches "automotive" and "financial" both have entropy 0, i.e., they are pure as well. Thus, we do not have to split those further. For the last split after the path "young" - "technology", we see that there are only companies with big number of employees in the data. These, however, are of mixed low and high risk by an equal proportion, so we cannot safely predict the risk class in that case. Instead, we could decide for 50% risk, or argue for high risk, which is the majority in the "young" sub-branch.

Moreover, we do not have any information on the "young"-"technology"-"small" branch in the data because there are no such samples, so we cannot decide in this case, either. One may argue for 50% risk as in the node "T" one level above for high risk based on the majority in the node "young" two levels above.

Lastly, we never found a use of the attribute *employees* in this exercise. One, therefore, could think of completely skipping *employees* from the data, in favor of a different one.

H10.2 Finding the Optimal Decision Tree [5 pts.]



We build decision trees on data with a high number of attributes. For this, we split the total data $\mathcal{D}$ into a development part $\mathcal{D}_L$ and a final test set $\mathcal{D}_t$. The development part is further split into a training set on which we train the trees, and a development test set to estimate their generalizability. Above, you see how the training error and the test error of the models would typically behave. Such a plot can be used to determine the optimal decision tree (that then should be finally evaluated on the held-out test set).

(a) Explain why we do this three-fold splitting of the data.

(b) Which quantity that characterizes the decision trees is indicated on the x-axis?

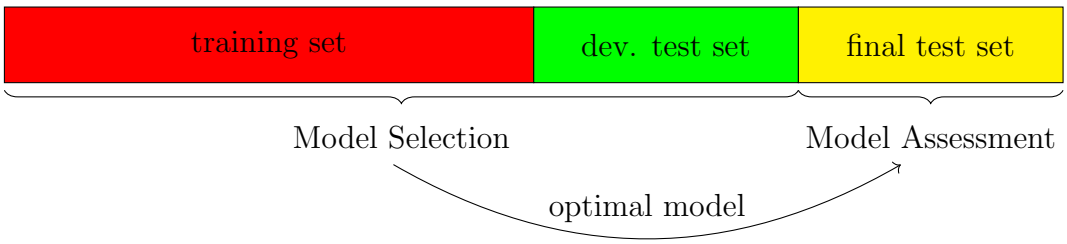
(c) Separate the plot into different regions on the x-axis, based on training and test error behavior. What are the correct terms for the model performance in these regions? Give a brief explanation of the regions.

(d) Where would you locate the optimal decision tree? What does *early stopping* mean in this context?

Solution:

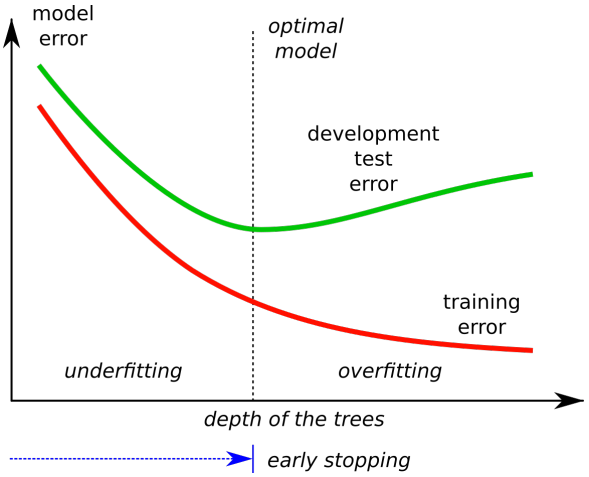
(a) It is important to distinguish between *model selection* and *model assessment*. Model selection (or development) means finding an optimal model among a set of different models (for example, of different complexity). Model assessment means estimating how well the final model will generalize to new data.
So, practically, we keep a final test set apart from the data that will not be touched before the end. The rest of the data is used to select an optimal model. In our scope, this is defined as the model that is neither over- nor under-fit. But to check this, we again need to split the data into a proper training set and a development test set. So we will train several models on the training set and validate them on the development test set to calculate their training and test errors. Having chosen the one with balanced training and test error (more on that later), we finally estimate its generalizability on the final test set. It is very important that the model has not seen this data before, not even for calculating the test errors in the selection, because this would bias the generalizability. Otherwise, the data already had some influence on the model, and we could not argue that it is totally new to the model. ✓
In a nutshell: strictly separate the data used for model selection from that used for model

assessment.



(b) The x-axis shows the *complexity* of the trees, i.e. their size or their *depth*. On the left, we have small trees (e.g., involving just a single split or decision); on the right, we have large trees (of many levels of splits or decisions). ✓

(c) On the left part, both training and test errors are high, which we call *underfitting*. The depth of the tree is not sufficient not predict the more complex behavior in the data.
Increasing the model complexity leads to a decrease in both training and test error. We reach an optimum at the turning point of the (development) test error curve. ✓
Going further to the right leads to *overfitting*. The training error still decreases here, but the (development) test error increases. This is because the high complexity enables the trees to adapt very closely to the training data at the cost of worse performance on the test data. ✓



(d) It should be clear from the previous that the optimal tree is between the regimes of underfitting and overfitting, i.e., at the minimum of the test error curve (the point is given above where it already should have been said). The training error curve typically is monotonically decreasing, basically because adding a (dummy) layer to a simpler tree at least gives the level of performance of the simpler tree.
The optimal tree complexity can be searched algorithmically, starting on the left with a low tree depth, and then going to the right. The training will automatically stop once the (development) test error curve has started to increase (with respect to some threshold). That way, the algorithm avoids wasting time fitting trees, which will be useless due to overfitting anyway. ✓

H10.3 Decision Tree: Spam or Ham? [8 pts.]

Please download a data set of 50 messages that are spam (1) or ham (0). In this task, you should generate an optimal level-two decision tree. If you use Python, you should prefer the package `sklearn`, which includes an implementation of decision trees.

(a) Think of five features that could be useful for the classification.

(b) Generate a 50×5 data-matrix $\mathbf{X}$ where each row has three indicator variables for the features.

(c) Generate the optimal two-level tree using entropy as the splitting criterion.

(d) Make a sketch of its structure.

(e) Test your tree on the test set, which is also uploaded. What accuracy is obtained?

Solution:

Please find an example for a solution uploaded to Moodle.

(a) ✓✓✓ After a look at the messages, you see that they contain a lot of nonsense such that it is hard to directly see good features. A bit of brainstorming and trial-and-error could yield features as follows:

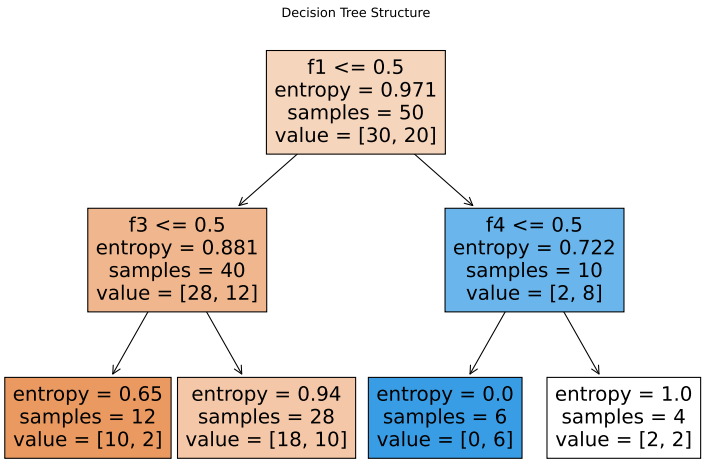
- f1: contains the word *win*
- f2: the length (in characters) of the message is less than 60
- f3: the average word length in the message is less than 4.8
- f4: contains one of "!" and "?"
- f5: contains a word from the non-sense list (e.g. *humdrum*, *nitwit*, etc.)

(b) This is solved by iterating over all messages in the training set and getting the value of the indicator value for each feature.
For example:

$$\texttt{'The win of mind shakes beliefs widely held.'} \quad \mapsto \quad (1,1,1,0,0). \checkmark$$

(c) This can be solved totally within the `sklearn` package. We just need to define a one-hot vector for the target class label and then fit the tree based on entropy and with the entropy as the criterion. ✓✓

(d) Also this is provided in `sklearn`. You should get something like below. ✓



(e) We can apply the method `predict` of the tree object to a data matrix of the test set. This needs to be compared with the true class labels of the test set. The accuracy is calculated by counting the number of matches (true positives and true negatives) and dividing this by the total number of messages in the test set. In our example, this yields

$$\text{accuracy} = 0.6. \checkmark$$

This leaves a lot of space for improvement – or demonstrates that the data is too artificial.

H10.4 Properties of Disorder [8 pts.]

Consider an observation $\boldsymbol{x} \in \{0,1\}^n$ where n_+ of the outcomes are equal to 1 and n_- of the outcomes are equal to 0.

(a) Prove that the entropy satisfies $D(n_+, n_-) = D(m \cdot n_+, m \cdot n_-)$ for any $m \in \mathbb{N}$.

(b) Prove that $D(n_+, n_-) = D(\bar{X}, 1 - \bar{X})$ holds where $\bar{X}$ denotes the sample mean.

Now consider the general case that $\boldsymbol{x}$ refers to a random sample of size n of a Bernoulli experiment X with success rate p.

(c) Write down the entropy $D(X)$ for one instance and explain that this can be viewed as an expected value.

(d) Show that the entropy can be expressed as $n \cdot D(p, 1-p)$.

(e) Let $p = 1 - e^{-\gamma}$ for $\gamma > 0$. Determine D as a function of γ and make a plot of $D(\gamma)$.

(f) Think of a system as in exercise H6.4, but with only two levels. How can we connect this to the results of parts (d) and (e)?

Comment: the first parts refer to a specific outcome of the random sample, the second to the general case. This means that the first parts actually consider the empirical entropy, the second the theoretical entropy.

Solution:

(a) The problem breaks down to expand the fractions by $m \neq 0$:

$$\begin{aligned} D(n_+, n_-) &= -\frac{n_+}{n} \log_2 \frac{n_+}{n} - \frac{n_-}{n} \log_2 \frac{n_-}{n} \\ &= -\frac{n_+ \cdot m}{n \cdot m} \log_2 \frac{n_+ \cdot m}{n \cdot m} - \frac{n_- \cdot m}{n \cdot m} \log_2 \frac{n_- \cdot m}{n \cdot m} \\ &= D(m \cdot n_+, m \cdot n_-). \checkmark \end{aligned}$$

This says that the disorder is invariant to copying the data m-times.

(b) The sample mean of a random sample of a Bernoulli random variable satisfies $\bar{X} = \frac{n_+}{n}$, it is just the proportion of positive outcomes (and the MLE for the success probability). This also implies that $\frac{n_-}{n} = \frac{n-n_+}{n} = 1 - \bar{X}$. So, we can calculate:

$$\begin{aligned} D(n_+, n_-) &= -\frac{n_+}{n} \log_2 \frac{n_+}{n} - \frac{n_-}{n} \log_2 \frac{n_-}{n} \\ &= -\bar{X} \log_2 \bar{X} - (1 - \bar{X}) \log_2 (1 - \bar{X}) \\ &= -\frac{\bar{X}}{1} \log_2 \frac{\bar{X}}{1} - \frac{\bar{X}}{1} \log_2 \frac{\bar{X}}{1} \\ &= D(\bar{X}, 1 - \bar{X}). \checkmark \end{aligned}$$

(c) For a single random variable $X \sim \text{Ber}(p)$, the entropy is given by

$$D(X) = -p \log_2 p - (1-p) \log_2 (1-p). \checkmark$$

By the definition of the expected value of a function

$$\mathbb{E}(g(X)) = \sum_{x \in V_X} g(x) Pr(X = x),$$

we can interpret the entropy as

$$D(X) = \mathbb{E}(-\log_2 Pr(X)) = \mathbb{E}(\frac{1}{\log_2 Pr(X)}). \checkmark$$

(d) By definition, a random sample consists of n independent and identically distributed realizations of a random variable. In this case, we can write for the probability mass function

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{n} f_x(x_i) = Pr(\boldsymbol{x}).$$

With the interpretation as an expected value, the entropy of the joint random variable is

$$D(\boldsymbol{X}) = D((X_1, \ldots, X_n)) = \mathbb{E}(-\log_2 Pr(\boldsymbol{X} =)) = \mathbb{E}(-\log_2 \prod_{i=1}^{n} f_x(x_i))$$

$$= \mathbb{E}(-\sum_{i=1}^{n} \log_2 f_x(x_i)). \checkmark$$

Since all X_i are independent, also the functions $\log_2(X_i)$ are independent. In this case, the expected value of their sum equals the sum of their expected values:

$$\mathbb{E}(-\sum_{i=1}^{n} \log_2 f_x(x_i)) = -\sum_{i=1}^{n} \mathbb{E}(-\log_2 Pr(X_i = x_i)) = -\sum_{i=1}^{n} D(X_i) = n \cdot D(X). \checkmark$$

We just used that the individual entropies are all equal to $D(X)$ as they share the distribution of X.

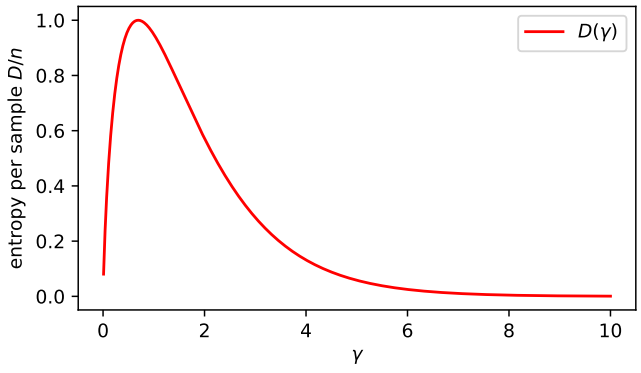
(e) We plug in the given probabilities to get

$$D(X) = nD(p, (1-p)) = -(1-e^{-\gamma}) \log_2(1-e^{-\gamma}) - e^{-\gamma} \log_2 e^{-\gamma}.$$

Using $\log_2 y = \frac{\log(y)}{\log(2)}$, we can change the base:

$$D(X) = \frac{-n}{\log(2)} \left((1-e^{-\gamma}) \log(1-e^{-\gamma}) - \gamma e^{-\gamma}\right).$$

This leads to the plot below, where the scale is "per sample" since we do not know n. The maximum is attained for $p = 1/2$ which is at $\gamma = \log(2)$.



✓

(f) If we identify $\gamma = \lambda t$, this refers to the time evolution of the entropy of a system of water drops falling down from the top to the bottom level by an exponential-decay-like distribution. We start with all n drops on the top level, where $D = 0$, reach the maximum $D = 1$ at $t = \log(2)/\lambda$ where (on expectation) half of the drops have fallen down, and approach $D = 0$ in the limit $t \to \infty$ when all drops are on the bottom level. ✓