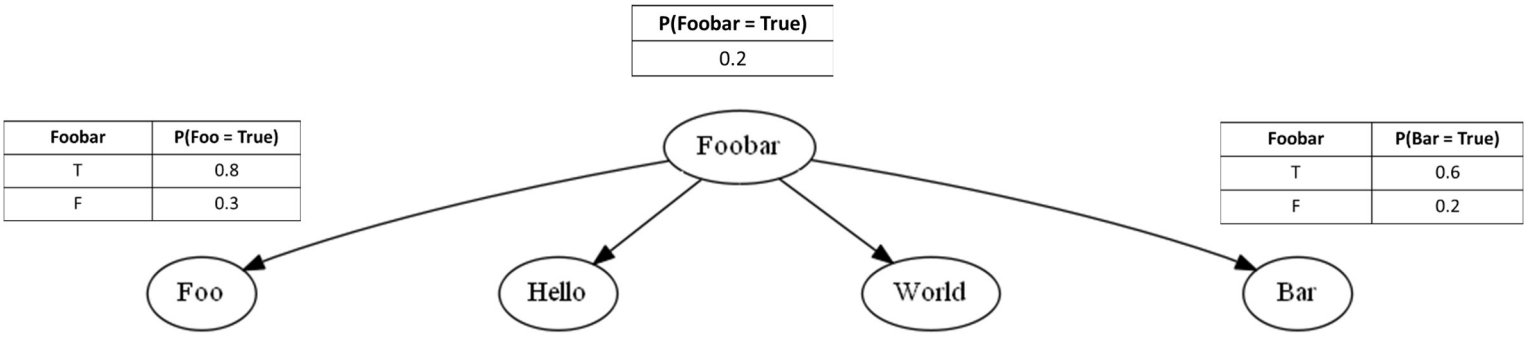
ECE 469: Artificial Intelligence
Fall 2020
Problem Set #2

1) Bayesian Networks, Maximum Likelihood Estimates, and Naïve Bayes

You are developing a Naïve Bayes system to predict whether or not documents are members of a class called *Foobar*, whatever that means. Predictions will be based on the values of four features, called *Foo*, *Hello*, *World*, and *Bar*, which are assumed to be conditionally independent given *Foobar*. *Foobar* is a Boolean class (i.e., it is either True or False); *Foo* and *bar* are Boolean features; and *Hello* and *World* are discrete features (the domain of *Hello* is {1, 2, 3, 4} and the domain of *World* is {A, B, C}).

A maximum likelihood approach has already been used to estimate the probabilities of values for each feature given the value of *Foobar*, as well as the prior probability of *Foobar*. The learned probability estimates are indicated in the conditional probability tables shown alongside the Bayesian network below. As is typical, to minimize the number of indicated parameters, the conditional probability tables leave out one possible value of each random variable, since they can be determined from the given information.



P(Foobar = True)
0.2

Foobar	P(Foo = True)
T	0.8
F	0.3

Foobar	P(Bar = True)
T	0.6
F	0.2

Foobar	P(Hello = 1)	P(Hello = 2)	P(Hello = 3)
T	0.3	0.5	0.1
F	0.5	0.2	0.2

Foobar	P(World = A)	P(World = B)
T	0.2	0.7
F	0.1	0.4

Assume that the system is used to predict the category (*foobar* or $\overline{foobar}$) of a new example, **e**, with the following feature values: **e** = [*Foo* = True, *Hello* = 2, *World* = C, *Bar* = False].

(a) What is the computed probability of seeing these feature values, assuming that *Foobar* is True? In other words, compute P(**e** | *foobar*). Show your work.

$$\mathrm{P}(\mathbf{e} \mid foobar) = \mathrm{P}(\mathrm{Foo} = \mathrm{True} \mid \mathrm{foobar}) * \mathrm{P}(\mathrm{Hello} = 2 \mid \mathrm{foobar}) * \mathrm{P}(\mathrm{World} = \mathrm{C} \mid \mathrm{foobar}) * \mathrm{P}(\mathrm{Bar} = \mathrm{False} \mid \mathrm{foobar})$$

$$= 0.8 * 0.5 * 0.1 * 0.4$$
$$= 0.016$$

ECE 469: Artificial Intelligence
Fall 2020
Problem Set #2

(b) What is the computed probability of seeing these feature values, assuming that *Foobar* is False? In other words, compute P(**e** | $\overline{foobar}$)? Show your work.

$$\mathrm{P}(\mathbf{e} \mid \overline{foobar}) = \mathrm{P}(\mathrm{Foo} = \mathrm{True} \mid \overline{foobar}) * \mathrm{P}(\mathrm{Hello} = 2 \mid \overline{foobar}) * \mathrm{P}(\mathrm{World} = \mathrm{C} \mid \overline{foobar}) * \mathrm{P}(\mathrm{Bar} = \mathrm{False} \mid \overline{foobar})$$

$$= 0.3 * 0.2 * 0.5 * 0.8$$
$$= 0.024$$

(c) What are the computed probabilities of *Foobar* being True and False, given the feature values of **e**? In other words, compute P(*foobar* | **e**) and P($\overline{foobar}$ | **e**)? Show your work and simplify your final answers.

$$< \mathrm{P}(foobar \mid \mathbf{e}), \mathrm{P}(\overline{foobar} \mid \mathbf{e}) >= \alpha < \mathrm{P}(\mathbf{e} \mid foobar) * \mathrm{P}(foobar), \mathrm{P}(\mathbf{e} \mid \overline{foobar}) * \mathrm{P}(\overline{foobar}) >$$

$$= \alpha < [0.016 * 0.2], [0.024 * 0.8] >$$

$$= \alpha < 0.0032, 0.0192 >$$

$$=< \frac{1}{7}, \frac{6}{7} >$$

$$\mathrm{P}(foobar \mid \mathbf{e}) = \frac{1}{7}$$

$$\mathrm{P}(\overline{foobar} \mid \mathbf{e}) = \frac{6}{7}$$

ECE 469: Artificial Intelligence
Fall 2020
Problem Set #2

2) Machine Learning Concepts

Briefly answer each of the following questions related to machine learning (answers should contain one or at most two sentences each).

(a) Related to machine learning, explain the concept of feature selection.

Feature selection selects the features that are the most relevant to a problem. This is used to mitigate the 'curse of dimensionality', simplifying the model because it does not need to handle many irrelevant (or not very useful) features. In addition, those irrelevant features can potentially lead to overfitting.

(b) Related to decision trees, would it ever make sense for the same feature / attribute to be tested twice along a single path from a root to a leaf?

Yes. It may be beneficial to split multiple times on the same feature. For example, a decision tree may split on different ranges on a continuous variable. It is plausible that splitting on the same feature more than once will maximize the amount of information.

(c) Briefly explain the conclusion of the No Free Lunch theorem.

The No Free Lunch theorem says that no machine learning method will be good for every problem. There will always be a trade-off in viability. Some methods will be better suited for certain problems and poorly suited for other problems. Efficient models usually have built-in assumptions, which do not apply to every problem.

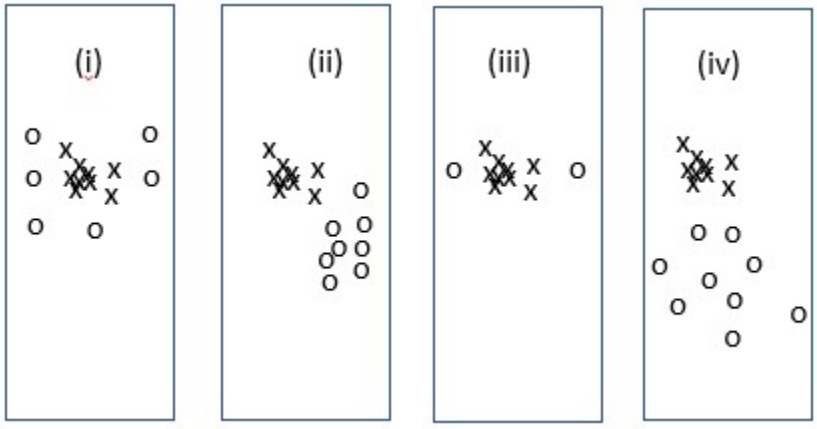
(d) Briefly explain why it typically takes longer to apply a k-nearest neighbor system than to train it. Is this a good thing or a bad thing?

Training a k-nearest neighbor system is as simple as storing the information about the examples. However, applying that system requires finding the k-nearest, which requires a search. Depending on the data structure used, it may take longer to search than to store the information. For example, a sequential table has a linear-time search, while a binary tree has a logarithmic-time search, while a hash table has a constant-time search.

This is usually a bad thing. In general, it is usually preferable to have a faster inference, rather than a faster training. This is because a model is usually used for inference. For example, when the goal is to repeatedly use the system for inference, the longer inference time will be a downside. This is the case for most machine learning models.

There are some situations where faster training is preferred over faster inference, but these situations are uncommon or contrived. For example, researchers may want to train many different models and evaluate the models once to compare results. In this case, it would be more beneficial to have a model with a faster training time, because the benefits of a faster inference time only show after repeatedly testing on many examples.

(e) Consider figures (i) through (iv) below. In each of them, the x's represent positive examples of some class, and the o's represent negative examples of the same class. Each example is fully represented as a two-dimensional numeric feature vector. Which of these classification tasks can theoretically be represented by a perceptron? Which of them can theoretically be represented by a neural network with one or more hidden layers? (Specify all that apply.)



The following assumes 'perceptron' refers to a single-layer perceptron.

i. Neural network only.
ii. Perceptron and neural network.
iii. Neural network only.
iv. Perceptron and neural network.

(i) and (iii) are not linearly separable, so they cannot be represented by a perceptron.