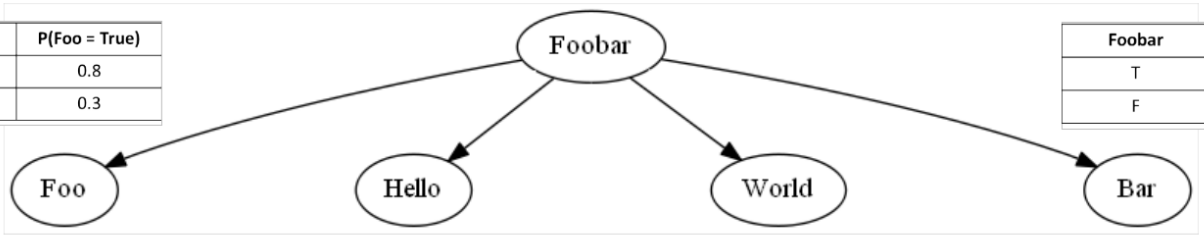
ECE469 – Pset 2

Jonathan Lam

November 23, 2020

1. *Bayesian networks, maximum likelihood estimates, and naïve Bayes*

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 the Bayesian network the one shown above, where the (conditional) probabilities are already determined using maximum likelihood estimates.

Define $\mathbf{E} \coloneqq \{\text{Foo}, \text{Hello} = 2, \text{World} = \text{C}, \overline{\text{Bar}}\}$.

(a) *Compute* $\mathrm{P}(\mathbf{E} \mid \text{Foobar})$.

With the naïve Bayes assumption we get the property that a node is conditionally independent of all its non-descendants given its parents. Thus, Foo, Hello, World, and Bar are all conditionally independent of each other given Foobar (and Foobar is not conditionally independent of any of the other nodes given any other node). If nodes A and B are conditionally independent of each other given C, then $\mathrm{P}(\mathrm{A} \wedge \mathrm{B} \mid \mathrm{C}) = \mathrm{P}(\mathrm{A} \mid \mathrm{C})\,\mathrm{P}(\mathrm{B} \mid \mathrm{C})$. This result extends to the case of multiple elements being mutually conditionally independent (via induction, not shown here). Thus:

$$\begin{aligned}\mathrm{P}(\mathbf{E} \mid \text{Foobar}) &= \mathrm{P}\left(\text{Foo} \wedge (\text{Hello} = 2) \wedge (\text{World} = \mathrm{C}) \wedge \overline{\text{Bar}} \mid \text{Foobar}\right) \\ &= \mathrm{P}(\text{Foo}|\text{Foobar})\,\mathrm{P}(\text{Hello} = 2|\text{Foobar})\,\mathrm{P}(\text{World} = \mathrm{C}|\text{Foobar})\,\mathrm{P}(\overline{\text{Bar}}|\text{Foobar}) \\ &= (0.8)(0.5)(0.1)(0.4) = 0.016\end{aligned}$$

(b) *Compute* $\mathrm{P}\left(\mathbf{E} \mid \overline{\text{Foobar}}\right)$.

This is the same as the previous question, except that Foobar is replaced with $\overline{\text{Foobar}}$.

$$\mathrm{P}(\mathbf{E} \mid \overline{\text{Foobar}}) = (0.3)(0.2)(0.5)(0.8) = 0.024$$

(c) *Compute* $\mathrm{P}(\text{Foobar} \mid \mathbf{E})$ *and* $\mathrm{P}\left(\overline{\text{Foobar}} \mid \mathbf{E}\right)$.

This is finding a posterior probability using Bayes' rule.

$$\begin{aligned}\mathrm{P}(\text{Foobar} \mid \mathbf{E}) &= \frac{\mathrm{P}(\text{Foobar} \wedge \mathbf{E})}{\mathrm{P}(\mathbf{E})} \\ &= \frac{\mathrm{P}(\mathbf{E} \mid \text{Foobar})\,\mathrm{P}(\text{Foobar})}{\mathrm{P}(\mathbf{E} \wedge \text{Foobar}) + \mathrm{P}(\mathbf{E} \wedge \overline{\text{Foobar}})} \\ &= \frac{\mathrm{P}(\mathbf{E} \mid \text{Foobar})\,\mathrm{P}(\text{Foobar})}{\mathrm{P}(\mathbf{E} \mid \text{Foobar})\,\mathrm{P}(\text{Foobar}) + \mathrm{P}(\mathbf{E} \mid \overline{\text{Foobar}})\,\mathrm{P}(\overline{\text{Foobar}})} \\ &= \frac{(0.016)(0.2)}{(0.016)(0.2) + (0.024)(0.8))} = \frac{0.0032}{0.0224} = \frac{1}{7} \\ \mathrm{P}(\overline{\text{Foobar}} \mid \mathbf{E}) &= [\mathrm{P}(\text{Foobar} \mid \mathbf{E})]^C = 1 - \frac{1}{7} = \frac{6}{7}\end{aligned}$$

2. *Machine learning concepts*

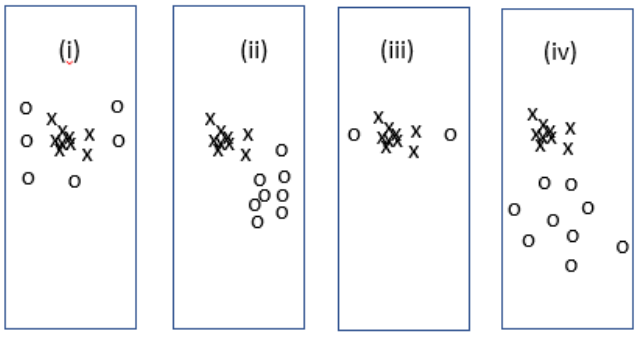
   (a) *Related to machine learning, explain the concept of feature selection.*
   Feature selection involves reducing the dimensionality of the input space as a form of simplifying the model (whether to improve generalization and/or to reduce computation count), and features are usually chosen by evaluating against a validation set.

   (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 – splits (using the algorithm discussed in class) attempt to maximize the amount of information (in the information-theoretic sense) in the split. It is plausible that splitting on one variable provides the most information at one point, and then after conditioning (splitting) on one or more other variable(s), splitting on this variable again gives the most information again.

   (c) *Briefly explain the conclusion of the No Free Lunch theorem.*
   No machine learning representation (e.g., decision trees, neural networks, Bayesian networks, etc.) can efficiently model all possible values of the hypothesis space (all possible mappings from inputs to outputs). In other words, we create efficient models by making assumptions about the form of the learning problem, and if trying to model all possible learning problems (hypotheses), and if all hypotheses were equally likely, random guessing is as good as you can get.

   (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 KNN is essentially remembering all of the points (trivial); inference using a KNN involves finding the K nearest neighbors (less trivial, have to comb through the points in some way). This isn't great; usually we want a fast inference, because this is how the model will be used (most other models have complex models that take a lot of computation to train but are straightforward to apply).

   (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.)*



   A (single-level) perceptron is limited in that it can only represent linearly-separable classes, while a neural net with at least one hidden layer (or multi-level perceptron) can represent non-linearly separable classes. Thus, (ii) and (iv) are representable with a perceptron, all of them are representable with a neural network with hidden layers.