National University of Singapore
School of Computing
CS3244 Machine Learning

Tutorial 7: Bayesian Inference

Issue: April 7, 2022 Due: April 11, 2022

Important Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *SUBMIT YOUR SOLUTIONS in PDF format to the 'TUTORIAL 6 SUBMISSION' folder under Files in LumiNUS by the DUE DATE specified above. Late submissions will NOT be entertained.*
- *Indicate your NAME, STUDENT NUMBER, and TUTORIAL GROUP in your submitted solution.*
- *YOUR SOLUTION TO QUESTION* **BL 11** *will be GRADED for this tutorial.*
- *You may discuss the content of the questions with your classmates. But everyone should work out and write up ALL the solutions by yourself.*

BL 11 In the solution to question TM 6.1 in Tutorial 5, I have shown you the use of the "incremental" version of Bayes' rule. One may wonder how this 'incremental" version can be derived from the original Bayes' Theorem. In this question, you are asked to derive the "incremental" version of Bayes' rule.

Specifically, let $h \in H$. Using the Bayes' Theorem, we know that

$$P(h|D_1, D_2) = \frac{P(D_1, D_2|h)P(h)}{P(D_1, D_2)} .$$

By assuming the conditional independence of data D_1 and data D_2 given hypothesis h, give a step-by-step derivation of the following "incremental" version of Bayes' rule:

$$P(h|D_1, D_2) = \frac{P(D_2|h)P(h|D_1)}{\sum_{h \in H} P(D_2|h)P(h|D_1)}$$

and state any result/assumption that you have used in each step.

Solution.

$$\begin{aligned} P(h|D_1, D_2) &= \frac{P(D_1, D_2|h)P(h)}{P(D_1, D_2)} \\ &= \frac{P(D_2|h)P(D_1|h)P(h)}{P(D_2|D_1)P(D_1)} \\ &= \frac{P(D_2|h)P(h|D_1)P(D_1)}{\sum_{h\in H} P(D_2, h|D_1)P(D_1)} \\ &= \frac{P(D_2|h)P(h|D_1)}{\sum_{h\in H} P(D_2|h, D_1)P(h|D_1)} \\ &= \frac{P(D_2|h)P(h|D_1)}{\sum_{h\in H} P(D_2|h)P(h|D_1)} \end{aligned}$$

The second equality is due to conditional independence assumption in the numerator and product rule in the denominator. The third equality is due to product rule or Bayes' Theorem: $P(D_1, h) = P(D_1|h)P(h) = P(h|D_1)P(D_1)$ in the numerator and marginalization in the denominator. The fourth equality follows from product rule in the denominator. The last equality is due to conditional independence assumption in the denominator.

TM 6.5 Consider the Minimum Description Length principle applied to the hypothesis space H consisting of conjunctions of up to n Boolean attributes (e.g., $Sunny \wedge Warm$). Assume each hypothesis is encoded simply by listing the attributes present in the hypothesis where the number of bits needed to encode any one of the n Boolean attributes is $1+\log_2 n$ (i.e., $\log_2 n$ bits to indicate which of the n Boolean attributes is present in the hypothesis and 1 bit to indicate its corresponding Boolean value). Suppose that the encoding of an example given the hypothesis uses zero bits if the example is consistent with the hypothesis and uses $1 + \log_2 m$ bits otherwise (i.e., 1 bit to indicate the correct classification and $\log_2 m$ bits to indicate which of the m examples was misclassified).

(a) Write down the expression for the quantity to be minimized according to the Minimum Description Length principle.

(b) Is it possible to construct a set of training data such that a consistent hypothesis exists, but MDL chooses an inconsistent hypothesis? If so, give such a set of training data. If not, explain why not.

(c) Give probability distributions for $P(h)$ and $P(D|h)$ such that the above MDL algorithm outputs MAP hypotheses.

Solution.

(a) $x_h(1 + \log_2 n) + y_h(1 + \log_2 m)$ where x_h is the number of Boolean attributes present in hypothesis h and y_h is the number of misclassified examples incurred by hypothesis h.

(b) Consider the following $-$ve training example as the training data:

$$\langle Sunny, Warm, High, Strong\rangle, - \ .$$

- Any hypothesis that is consistent with the $-$ve training example has to specify the value of at least one attribute. For example, the hypothesis $\langle Rainy, ?, ?, ?\rangle$ specifies the value of one attribute and misclassifies no example. So, its description length is $1(1+\log_2 4)+0(1+\log_2 1)=3$.
- The maximally general hypothesis $\langle ?, ?, ?, ?\rangle$, which is inconsistent with the $-$ve training example, specifies no attribute and misclassifies this single example. So, its description length is $0(1+\log_2 4)+1(1+\log_2 1)=1$ and it is thus selected by MDL.

(c) We assume no '$\emptyset$' symbol in the hypothesis representation.

The MDL algorithm selects the hypothesis that minimizes $x_h(1+\log_2 n)+y_h(1+\log_2 m)$ while the MAP hypothesis minimizes $-\log_2 P(h)-\log_2 P(D|h)$ (page 18 of the "Bayesian Inference" lecture slides).

$$\begin{aligned}
\arg\min_{h\in H}\ x_h(1+\log_2 n)+y_h(1+\log_2 m) &= \arg\min_{h\in H} -\log_2 (1/(2n))^{x_h}(1/(2m))^{y_h} \\
&= \arg\min_{h\in H} -\log_2 \frac{(1/(2n))^{x_h}}{(1+1/n)^n}\frac{(1/(2m))^{y_h}}{(1+1/(2m))^m} \\
&= \arg\min_{h\in H} -\log_2 \frac{(1/(2n))^{x_h}}{(1+1/n)^n} - \log_2 \frac{(1/(2m))^{y_h}}{(1+1/(2m))^m} \ .
\end{aligned}$$

To determine the probability distributions for $P(h)$ and $P(D|h)$ such that the MDL algorithm outputs a MAP hypothesis, simply let $P(h)=\dfrac{(1/(2n))^{x_h}}{(1+1/n)^n}$ and $P(D|h)=\dfrac{(1/(2m))^{y_h}}{(1+1/(2m))^m}$.
Note that

$$\begin{aligned}
\sum_{h\in H} P(h) &= \frac{1}{(1+1/n)^n}\sum_{h\in H}(1/(2n))^{x_h} \\
&= \frac{1}{(1+1/n)^n}\sum_{x_h=0}^{n} C(n,x_h)2^{x_h}(1/(2n))^{x_h} \\
&= \frac{1}{(1+1/n)^n}(1+1/n)^n \\
&= 1
\end{aligned}$$

where the third equality is due to Binomial Theorem.

Similarly,

$$\begin{aligned}\sum_D P(D|h) &= \frac{1}{(1+1/(2m))^m}\sum_D (1/(2m))^{y_h} \\ &= \frac{1}{(1+1/(2m))^m}\sum_{y_h=0}^{m} C(m, y_h)(1/(2m))^{y_h} \\ &= \frac{1}{(1+1/(2m))^m}(1+1/(2m))^m \\ &= 1\end{aligned}$$

where the third equality is due to Binomial Theorem.

BL 12a (Final Exam AY2020/21) Fig. 1a and Fig. 1b below show perceptron units A and B. Fig. 1c below shows a network C of perceptron units with two hidden layers of two units each, while Fig. 1d below shows a network F of perceptron units with a hidden layer of two units. They are based on the following structure:

- Perceptron units A and B have one (Boolean) output unit each for producing the output o_A and the output o_B, respectively. Similarly, networks C and F of perceptron units have one (Boolean) output unit each for producing the output o_C and the output o_F, respectively.
- There should be four input units (i.e., one input unit for each of the four (Boolean) input attributes x_1, x_2, x_3, x_4).
- A Boolean is **–1 if false**, and **1** if true.
- The activation function of every (non-input) unit is a **–1 to 1 step function** (refer to page 6 of the "Neural Networks" lecture slides), including that of the output unit.
- The weights $\mathbf{w}_A$ (i.e., hypothesis) of perceptron unit A and the weights $\mathbf{w}_B$ (i.e., hypothesis) of perceptron unit B are indicated in Fig. 1a and Fig. 1b, respectively. The weights $\mathbf{w}_C$ (i.e., hypothesis) of network C of perceptron units and the weights $\mathbf{w}_F$ (i.e., hypothesis) of network F of perceptron units are indicated in Fig. 1c and Fig. 1d, respectively.
- A bias input is of value 1 and is not considered a hidden unit.

One of the four perceptron networks in Fig. 1 has been used to generate a dataset of 4 Boolean input attributes x_1, x_2, x_3, x_4 and a Boolean target output t_d with the following 3 noise-free training examples of the form $d = \langle(x_1, x_2, x_3, x_4), t_d\rangle$:

$$D = \{d_1 = \langle(-1,-1,-1,1), 1\rangle, d_2 = \langle(1,1,-1,-1), 1\rangle, d_3 = \langle(1,1,1,1), -1\rangle\}.$$

Suppose that the **prior beliefs** of hypotheses/weights $\mathbf{w}_A$, $\mathbf{w}_B$, $\mathbf{w}_C$, and $\mathbf{w}_F$ are equal and they sum to 1.

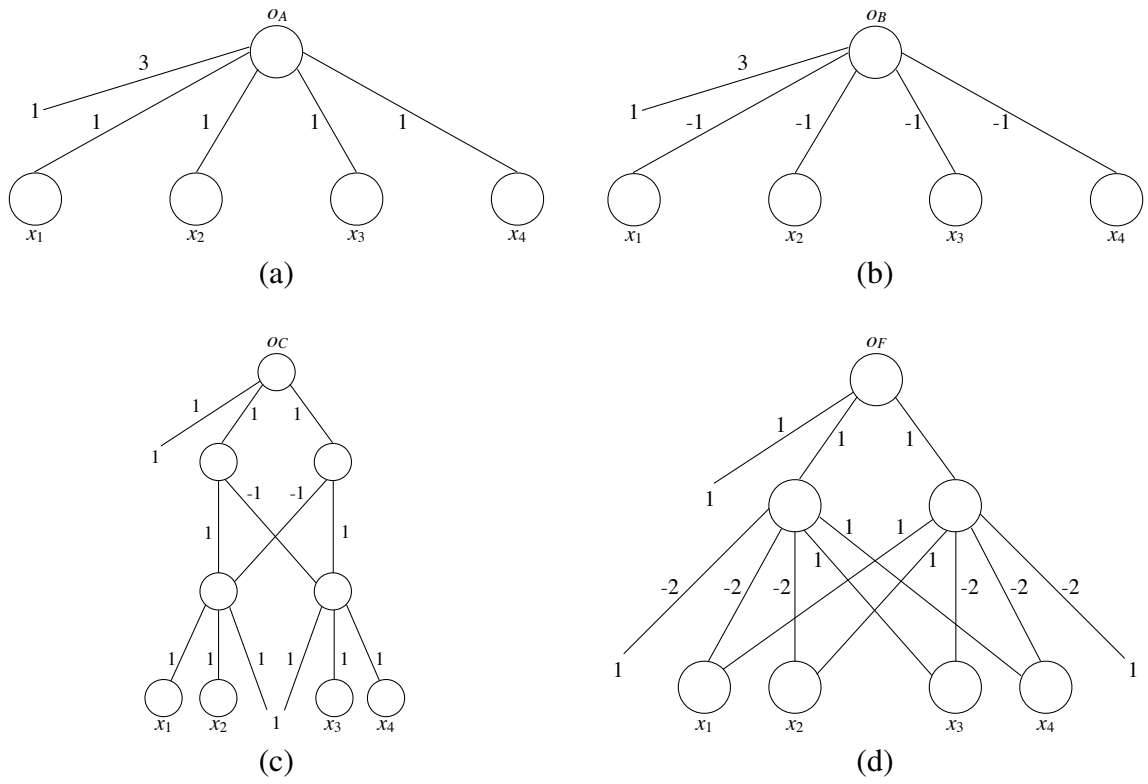

Figure 1: Perceptron networks: (a) perceptron unit A, (b) perceptron unit B, (c) network C of perceptron units, and (d) network F of perceptron units.

Using Bayes' Theorem, calculate the posterior beliefs $P(\mathbf{w}_A|D)$, $P(\mathbf{w}_B|D)$, $P(\mathbf{w}_C|D)$, and $P(\mathbf{w}_F|D)$. Show the steps of your derivation. **No marks will be awarded for not doing so.**

We assume that the input instances $\mathbf{x}_d = (x_1, x_2, x_3, x_4)$ for $d \in D$ are fixed. Therefore, in deriving an expression for $P(D|\mathbf{w}_A)$, $P(D|\mathbf{w}_B)$, $P(D|\mathbf{w}_C)$, or $P(D|\mathbf{w}_F)$, we only need to consider the probability of observing the target outputs t_d for $d \in D$ for these fixed input instances $\mathbf{x}_d$ for $d \in D$.

Furthermore, we assume that the training examples are conditionally independent given the hypothesis/weights of any perceptron network in Fig. 1.

Solution.

1. Note that A implements the OR gate while B implements the NAND gate.

2. Both C (tutorial 4 question BL 8c) and F (see truth table below) implement (x_1 OR x_2) XOR (x_3 OR x_4).

x_1	x_2	x_3	x_4	(x_1 OR x_2) XOR (x_3 OR x_4)
-1	-1	-1	1	1
-1	-1	1	-1	1
-1	-1	1	1	1
-1	1	-1	-1	1
1	-1	-1	-1	1
1	1	-1	-1	1
-1	1	-1	1	-1
-1	1	1	-1	-1
-1	1	1	1	-1
1	-1	-1	1	-1
1	-1	1	-1	-1
1	-1	1	1	-1
1	1	-1	1	-1
1	1	1	-1	-1
1	1	1	1	-1
-1	-1	-1	-1	-1

$$
\begin{aligned}
P(\mathbf{w}_A|D) = &\ \frac{\prod_{i=1}^{3} P(t_{d_i}|\mathbf{w}_A, \mathbf{x}_{d_i})P(\mathbf{w}_A)}{P(D)} \\
= &\ 0 \quad \text{since } P(t_{d_3}|\mathbf{w}_A, \mathbf{x}_{d_3}) = 0 \\
P(\mathbf{w}_F|D) = P(\mathbf{w}_C|D) = P(\mathbf{w}_B|D) = &\ \frac{\prod_{i=1}^{3} P(t_{d_i}|\mathbf{w}_B, \mathbf{x}_{d_i})P(\mathbf{w}_B)}{P(D)} \\
= &\ \frac{0.25}{P(D)} \\
P(D) = &\ 0.75 \\
P(\mathbf{w}_F|D) = P(\mathbf{w}_C|D) = P(\mathbf{w}_B|D) = &\ 1/3\ .
\end{aligned}
$$

Using the posterior beliefs calculated above, compute the **Bayes-optimal classification** for the new input instance $\mathbf{x}_{d_4} = (-1, -1, -1, -1)$. Show the steps of your derivation. **No marks will be awarded for not doing so.**

Solution.

$$
\begin{aligned}
P(t_{d_4} = -1|D, \mathbf{x}_{d_4}) = &\ P(t_{d_4} = -1|\mathbf{w}_B, \mathbf{x}_{d_4})P(\mathbf{w}_B|D) + P(t_{d_4} = -1|\mathbf{w}_C, \mathbf{x}_{d_4})P(\mathbf{w}_C|D) \\
&\ + P(t_{d_4} = -1|\mathbf{w}_F, \mathbf{x}_{d_4})P(\mathbf{w}_F|D) \\
= &\ (1/3)(0 + 1 + 1) \\
= &\ 2/3 \\
P(t_{d_4} = 1|D, \mathbf{x}_{d_4}) = &\ 1/3
\end{aligned}
$$

So, the Bayes-optimal classification for the new input instance $\mathbf{x}_{d_4} = (-1, -1, -1, -1)$ is $t_{d_4} = -1$.