Introduction to Information Theory and Machine Learning
(Prof. Permuter Haim, Mr. Oron Sabag and Mr. Ron Shoham)

Final Exam - Moed B
Total time for the exam: 3 hours!

Important: For **True / False** questions, copy the statement to your notebook and write clearly true or false. You should prove the statement if true, and provide counterexample otherwise.

1) **Duality bound for memoryless channel (34 Points)**: In this question, you will prove a simple upper bound on the capacity of a memoryless channel. The channel is given by $P_{Y|X}$ and the capacity is denoted by C. We will also need a new distribution on channel outputs $Q_Y(\cdot)$ which is an arbitrary distribution. Finally, when we write $I_P(X;Y)$ this means that the mutual information is computed with respect to $P_{X,Y}$.
   a) (4 points) Write the capacity C of a memoryless channel, $P_{Y|X}$ in terms of a divergence (do not use mutual information).
   Solution:
   $$C = \max_{P_X} \; I(X;Y) = \max_{P_X} \; D(P_{X,Y}||P_X P_Y)$$
   b) (6 points) Complete $\leq, =, \geq$ between the following expressions (prove you answer):
   $$I_P(X;Y) \quad \text{Vs.} \quad \sum_{x\in\mathcal{X}} \left[P_X(x) D\left(P_{Y|X=x}||Q_Y\right)\right] - D(P_Y||Q_Y).$$
   Solution: These are equal.
   $$\begin{aligned}
   &\sum_{x\in\mathcal{X}} \left[P_X(x) D\left(P_{Y|X=x}||Q_Y\right)\right] - D(P_Y||Q_Y) \\
   =&\sum_{x\in\mathcal{X}} \left[P_X(x) \sum_{y\in\mathcal{Y}} P_{Y|X}(y|x) \log \frac{P_{Y|X}(y|x)}{Q_Y(y)}\right] - \sum_{y\in\mathcal{Y}} \left[P_Y(y) \log \frac{P_Y(y)}{Q_Y(y)}\right] \\
   =&\sum_{x,y\in\mathcal{X}\mathcal{Y}} \left[P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{Q_Y(y)P_X(x)}\right] - \sum_{y\in\mathcal{Y}} \left[P_Y(y) \log \frac{P_Y(y)}{Q_Y(y)}\right] \\
   =&\sum_{x,y\in\mathcal{X}\mathcal{Y}} \left[P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{Q_Y(y)P_X(x)}\right] - \sum_{x,y\in\mathcal{X}\mathcal{Y}} \left[P_{X,Y}(x,y) \log \frac{P_Y(y)}{Q_Y(y)}\right] \\
   =&\sum_{x,y\in\mathcal{X}\mathcal{Y}} \left[P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{Q_Y(y)P_X(x)} \frac{Q_Y(y)}{P_Y(y)}\right] \\
   =&\sum_{x,y\in\mathcal{X}\mathcal{Y}} \left[P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{P_Y(y)P_X(x)}\right] \\
   =&I_P(X;Y).
   \end{aligned}$$
   c) (5 points) Prove the duality bound (justify each step):
   $$C \leq \max_{x\in\mathcal{X}} D\left(P_{Y|X=x}||Q_Y\right).$$
   Solution:
   $$\begin{aligned}
   C &= \max_{P_X} I_P(X;Y) \\
   &= \max_{P_X} \sum_{x\in\mathcal{X}} \left[P_X(x) D\left(P_{Y|X=x}||Q_Y\right)\right] - D(P_Y||Q_Y) \\
   &\overset{(a)}{\leq} \max_{P_X} \sum_{x\in\mathcal{X}} \left[P_X(x) D\left(P_{Y|X=x}||Q_Y\right)\right] \\
   &\overset{(b)}{=} \max_{x\in\mathcal{X}} D\left(P_{Y|X=x}||Q_Y\right).
   \end{aligned}$$
   Where (a) follows the non-negativity of KullbackLeibler divergence
   and (b) is true since $\max_{P_X}$ yields $P_X = \mathbb{1}_{\{X=x_m\}}$, where $x_m \in \mathcal{X}$ is maximizing the KL divergence.
   d) (6 points) Find sufficient and necessary conditions for the tightness of the duality bound.
   Solution: For the inequality marked (a) we need $D(P_Y||Q_Y) = 0$, which will happens *iff* $Q_Y = P_Y$. The tightness of duality bound depends on P_Y and Q_Y and its *not* affected by $\max_{P_X}$.
   e) (7 points) We now define $P_{Y|X}$ to be a binary symmetric channel (BSC) with transition probability α. Compute the duality bound when $Q_Y \sim \text{Bernoulli}(0.25)$ and $Q_Y \sim \text{Bernoulli}(0.5)$. Your answers should be simple and without a

maximum.
Solution:

$$DB = \max_{x\in\mathcal{X}} D\left(P_{Y|X=x}||Q_Y\right) = \max_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \left[P_{Y|X=x} \log \frac{P_{Y|X=x}}{Q_Y}\right]$$

For $Q_Y \sim \text{Bernoulli}(0.25)$
$\underline{x=0}$

$$\begin{aligned} DB &= (1-\alpha)\log\left(\frac{1-\alpha}{0.25}\right) + \alpha\log\left(\frac{\alpha}{0.75}\right) \\ &= \log\left(4(1-\alpha)\right) + \alpha\log\left(\frac{\frac{\alpha}{0.75}}{\frac{1-\alpha}{0.25}}\right) \\ &= 2 + \log(1-\alpha) + \alpha\log\left(\frac{\alpha}{3(1-\alpha)}\right) \end{aligned}$$

$\underline{x=1}$

$$\begin{aligned} DB &= \alpha\log\left(\frac{\alpha}{0.25}\right) + (1-\alpha)\log\left(\frac{1-\alpha}{0.75}\right) \\ &= \log\left(\frac{4(1-\alpha)}{3}\right) + \alpha\log\left(\frac{\frac{\alpha}{0.25}}{\frac{1-\alpha}{0.75}}\right) \\ &= 2 + \log\left(\frac{1-\alpha}{3}\right) + \alpha\log\left(\frac{3\alpha}{1-\alpha}\right) \end{aligned}$$

So $x=0$ or 1 depends on α, if $\alpha \geq 0.5$ we take $x=1$, otherwise $x=1$.
For $Q_Y \sim \text{Bernoulli}(0.5)$, results will be the same for $x=0$ and $x=1$

$$\begin{aligned} DB &= (1-\alpha)\log\left(\frac{1-\alpha}{0.5}\right) + \alpha\log\left(\frac{\alpha}{0.5}\right) \\ &= \log\left(2(1-\alpha)\right) + \alpha\log\left(\frac{\alpha}{1-\alpha}\right) \\ &= 1 + \log\left(1-\alpha\right) + \alpha\log\left(\frac{\alpha}{1-\alpha}\right) \end{aligned}$$

f) (6 points) Are the two upper bounds from the previous question equal the capacity of the BSC? prove your answers.
Solution: For $Q_Y \sim \text{Bernoulli}(0.5)$ we have that the upper bound satisfies:

$$\begin{aligned} C_{BSC} &= 1 - H_b(\alpha) \\ &= 1 + \alpha\log\alpha + (1-\alpha)\log(1-\alpha) \\ &= 1 + \log(1-\alpha) + \alpha\log\left(\frac{\alpha}{1-\alpha}\right) \end{aligned}$$

For $Q_Y \sim \text{Bernoulli}(0.25)$ and $\alpha \leq 0.5$ we have

$$\begin{aligned} DB &= 2 + \log(1-\alpha) + \alpha\log\left(\frac{\alpha}{3(1-\alpha)}\right) \\ &= C_{BSC} + 1 - \alpha\log(3) \\ &\geq C_{BSC}. \end{aligned}$$

For $\alpha \geq 0.5$ we result with

$$\begin{aligned} DB &= C_{BSC} + 1 - (1-\alpha)\log(3) \\ &\geq C_{BSC} \end{aligned}$$

The upper bound that you proved is called the duality upper bound. As seen above, there are conditions for the tightness of the bound, and wise choices of the test distribution Q_Y may give rise to good bounds on the capacity.

2) **Compression using machine learning (30 Points)**
A source sequence $x_1, x_2, ..., x_n$ is given where the cardinality of the alphabet of x_i is 4, namely, $|\mathcal{X}| = 4$. You observe a noisy version of the sequence, $y_1, y_2, ..., y_n$ where $Y_i = X_i + Z_i$, and Z_i has a Gaussian distribution with zero mean and some variance. You do not know the variance of the noise Z_i nor the explicit alphabet of X, but, you do know that the noise is with high probability lower than the minimal difference between the values of X.

a) (5 points) What would you expect the histogram of y^n to be. Draw it.
Solution: Without loss of generality $x_1 \leq x_2 \leq x_3 \leq x_4$, and heights of each x_i is also arbitrary. Note that there are

infinite number of options for this histogram.

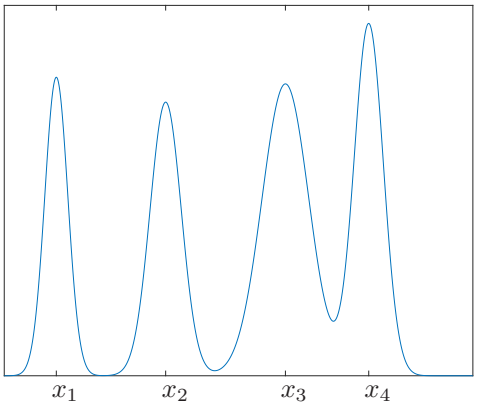


Fig. 1: An example of a histogram of y^n

b) (5 points) Given the sequence y^n, suggest an ML algorithm that estimates x^n. (provide a pseudo code).
Solution: Since it is given that the noise's variance is much smaller than the minimal difference between the values of X, we need a ML algorithm that can clean the noise and find all the alphabet. Most simple way is Kmeans algorithm shown in class, with Y as input and $\{\mu_j\}_{j=1}^4$ as output
Kmeans Algorithm(Y)

---

i) Initialize centroids $\mu_1 ... \mu_4 \in \mathbb{R}$
ii) Repeat until $\mu_j^{(t-1)} \approx \mu_j^{(t)}$:
A) for every i, set

$$c_i := \arg\min_j \|y_i - \mu_j\|^2$$

B) for each j, set

$$\mu_j^{(t)} := \frac{\sum_{i=1}^m \mathbb{1}\{c_i = j\} y_i}{\sum_{i=1}^m \mathbb{1}\{c_i = j\}}$$

As the algorithm stops, $\{\mu_j\}_{j=1}^4$ will represent x_i. Another option is to set a complete GMM, for that we need in addition to Kmeans an EM algorithm. If you chose GMM you may initialize it with random samples instead of Kmeans.
EM algorithm for GMM

---

i) E-step: for each i, j

$$w(j,i) := \frac{\phi(j) P(y_i | g_i = j; \mu_j, \Sigma_j)}{\sum_{l=1}^k \phi(l) P(y_i | g_i = l; \mu_l, \Sigma_l)}$$

ii) M-step: for each j

$$\phi(j) := \frac{1}{m} \sum_{i=1}^m w(j,i)$$
$$\mu_j := \frac{\sum_{i=1}^m w(j,i) y_i}{\sum_{i=1}^m w(j,i)}$$
$$\Sigma_j := \frac{\sum_{i=1}^m w(j,i)(y_i - \mu_j)(y_i - \mu_j)^T}{\sum_{i=1}^m w(j,i)}$$

Where g_i are the hidden variables(in the lecture they are marked as z_i). Iterating EM algorithm until $\theta^{(t)} \approx \theta^{(t-1)}$, this means that $\{\mu_j\}_{j=1}^4$ will be the requested $\{x_i\}_{i=1}^4$.

c) (5 points) In what category the ML algorithm that you suggested in 2b (previous sub question) is: supervised learning or non-supervised learning.
Solution: Unsupervised learning, classes are unknown and training data had no labels.

d) (5 points) Now, you have the following system that is given in Fig. 2. A new sequence y^l arrives to the encoder and it has a similar distribution as the sequence y^n from 2b. The encoder first estimates x_i from y_i using the inference of the

ML algorithm that you have build and trained in 2b and then compress it using variable length coding. Suggest how to build variable length codes using the sequence y^n from 2b. Suggest at least two variable-length codes.

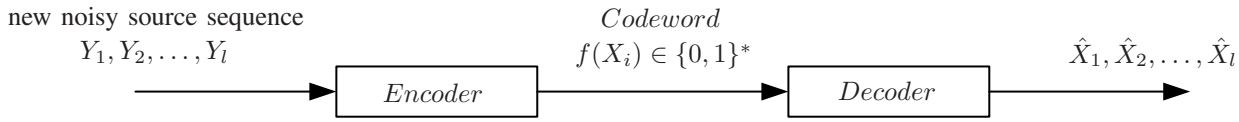


Fig. 2: ML and Source coding problem. The encoder in the figure converts a value y_i into x_i using the ML algorithm you suggested in 2b and then into sequence of bits $\{0,1\}$ of variable length denoted as $\{0,1\}^{l(x)}$. The goal of the decoder is to reconstruct the original signal X_i.

Solution: If EM has been used, then the following orderings are made according to G, which is the probability for each $\hat{x}_i$. If only Kmeans is used, we calculate empirical probabilities, i.e.

$$P_{\hat{x}_i} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{\{c_k=i\}}.$$

where c_i is calculated for each sample similar to the Kmean algorithm. Now there are some options, the obvious suggestion is using Huffman code.

Huffman algorithm

---

i) Create a leaf node for each symbol and add it to a priority queue in decending order, with respect to the probablities.
ii) While there is more than one node in the queue:

A) Remove the last two nodes from the queue and add 0 or 1 ,respectively , to any code already assigned to them.
B) Create a new internal node with these two nodes as children and with probability equal to the sum of the two nodes' probabilities, and add it to the queue.

iii) The remaining node is the root(and should have probability of 1).

Another suggestion is Shannon-Fano code.

Shannon-Fano algorithm

---

i) Set a desending list of symbols with respect to their probabilities.
ii) Divide the list into two, such that the sum of prob. on each side is as close to equality as possible.
iii) The left part is assigned the digit 0, and the right part is assigned the digit 1.
iv) Recursively repeat the last 2 steps (ii and iii) on each of the part, until all parts have only one symbol.

---

e) (5 points) Are the variable codes you suggested optimal, and if yes in what sense.
Solution: Huffman code is optimal in mean sense, it will produces prefix codes that always achieve the lowest expected code word length. Shannon-Fano can reaches lowest expected code word but not always, hence not optimal.

f) (5 points) Repeat all the previous sub-question where x_i, y_i and z_i are two-dimensional vectors. i.e.

$$x_i = (x_i^{(1)}, x_i^{(2)})$$
$$z_i = (z_i^{(1)}, z_i^{(2)})$$
$$Z_i \sim \mathcal{N}(0, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}).$$

*In the two dimensional part you draw a representative contour instead of sketching histogram.
Solution:

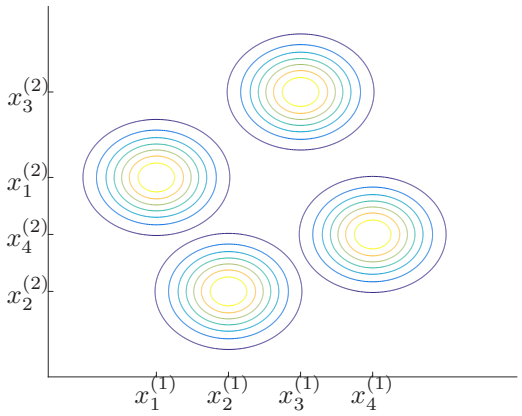


Fig. 3: Arbitrary histogram of y^n

Kmeans_Algorithm(x_1, x_2, x_3, x_4)

i) Initialize centroids $\mu_1 ... \mu_4 \in \mathbb{R}^2$

ii) Repeat until convergence:

A) for every i, set

$$c_i := \arg\min_j \|(x_i^{(1)} - \mu_j^{(1)})^2 + (x_i^{(2)} - \mu_j^{(2)})^2\|^2 \tag{1}$$

B) for each j, set

$$\mu_j^{(1)} := \frac{\sum_{i=1}^m \mathbb{1}\{c_i = j\} x_i^{(1)}}{\sum_{i=1}^m \mathbb{1}\{c_i = j\}} \tag{2}$$

$$\mu_j^{(2)} := \frac{\sum_{i=1}^m \mathbb{1}\{c_i = j\} x_i^{(2)}}{\sum_{i=1}^m \mathbb{1}\{c_i = j\}} \tag{3}$$

EM algorithm works with the same equations. Suggested codes are the same, the only change is the code-book which will expands due to adding another dimension.

3) **True/False (28 Points)**:

a) **Information Theory:** Given is a joint distribution $P_{X,Y}$ and a deterministic function $f : \mathcal{X} \to \mathcal{X}$ that satisfy

$$H(Y|f(X)) \leq H(Y|X).$$

On each of the next statements write True/False.

i) (4 points) There exists a Markov chain $Y - f(X) - X$.
Solution: True For deterministic function $f(X)$ we know that $Y - X - f(X)$ is a Markov chain. Thus $I(Y; f(X) \leq I(Y; X)$. Additionally,

$$\begin{aligned} I(Y;X) &= H(Y) - H(Y|X) \\ &\leq H(Y) - H(Y|f(X)) \\ &= I(Y; f(X)) \end{aligned}$$

While the inequality follows from the given inequality. From these two inequalities, we can conclude that $I(Y;X) = I(Y; f(X))$. Thus, $I(Y; X|f(X)) = 0$. Hence $Y - f(X) - X$ is a Markov chain.

ii) (4 points) The function $f(\cdot)$ is an injective (one to one) function.
Solution: False Choose X and Y to be independent, and $f()$ to be any non-injective mapping. We can see that the inequality

$$H(Y|f(X)) \leq H(Y|X)$$

holds but $|f(\mathcal{X})| = 1 < |\mathcal{X}|$. Thus $f(\cdot)$ is not an injective function.

iii) (4 points) If $f(X) \sim \text{Unif}(1, \ldots, \mathcal{X})$, then $X \sim \text{Unif}(1, \ldots, \mathcal{X})$.
Solution: True For the given distribution we can conclude that $\forall x \in \mathcal{X} : P(f(x)) > 0$. Thus $f(X)$ is injective function of X and that means that $X \sim f(X) \sim \text{Unif}(1, \ldots, |\mathcal{X}|)$.
An alternative proof:
$H(f(X)) \leq H(X) \leq \log |\mathcal{X}|$. A sandwich argument concludes that $H(X) = \log |X|$.

b) **Machine learning:**

i) (4 points)**True/False:** The log-likelihood of the data will *always* increase through successive iterations of the expectation maximization algorithm.
Solution: false We can see in the lecture note Eq. 13-17 that EM step only promise that the log-likelihood won't decrease in a successive iteration.

ii) (4 points) **True/False:** In distribution tree, as defined in Chow-Liu algorithm, a node can have more than one 'father'.
Solution: False The distribution tree as defined in Chow-Liu algorithm must have only one father. We can see that from the assumption that the *pmf* must have only pairs ($P(u|z), P(w|z), P(v|y)$ etc.).

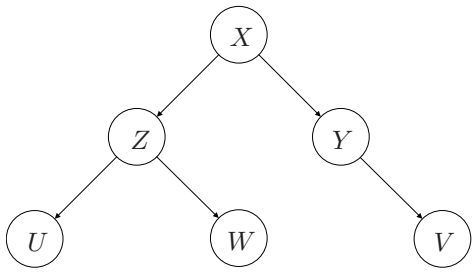


Fig. 4: A tree with distributions of the structure $P_t(x, y, z, u, v, w) = P(x)P(y|x)P(z|x)P(u|z)P(w|z)P(v|y)$

Another way to see that is from the proof of the Chow Liu algorithm (the constructed tree will be the best approximation among all first-order dependency trees) - the way we build a tree is with respect to the mutual information between 2 nodes only, 'father' and 'son'.

iii) (4 points) We wish to generate classifier which classify between K classes. In order to do so we train a Neural Net with softmax output layer (with k output neurons). Let us note the output vector as $\hat{Q}_\theta(x)$. We use the cross-entropy cost function to measure the distance between the output distribution and the real labels. **True/False:** By the law-of-large-numbers, the cost is equal to the KL-divergence between the distribution $\hat{Q}_\theta(x)$ and the distribution of the real labels.
Solution: False Proof from the lecture: Consider the following cost

$$C_n(\theta) = -\sum_{i=1}^{n} \log \hat{P}(y^{(i)}|x^{(i)}, \theta)$$

By the law of large number this cost convergence to

$$\begin{aligned}\lim_{n\to\infty} C_n(\theta) &= -E[(Y|X,\theta)] \\ &= -\sum_{x,y} p(y,x) \log p(y|x,\theta) \\ &= -\sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x,\theta) \\ &= -\sum_{x} p(x) H\left(p(y|x), p(y|x,\theta)\right)\end{aligned}$$

Where last equality follows from the fact that for a given x,
$-\sum_y p(y|x) \log p\left(y|x,\theta\right)$ is the cross entropy $H\left(p(y|x), p(y|x,\theta)\right)$.

iv) (4 points) **True/False:** Decision Tree which was built by ID3 algorithm guarantees 0% training error.
Solution: False $ID3$ algorithm has a stopping condition that states **if** $A \in \varnothing$ **then** mark new node as leaf and label it as the majority of labels in S. It means that there are times that the algorithm failed to split the data completely after using all features, and needs to make a decision.

4) **Tree Distribution (23 Points)**: You wish to generate a model to predict if a mushroom is poisonous or not. You have some empirical data:

Example	Is heavy	Is smelly	Is spotted	Is smooth	Is poisonous
A	0	0	0	0	0
B	0	0	1	0	0
C	1	1	0	1	0
D	1	0	0	1	1
E	0	1	1	0	1
F	0	0	1	1	1
G	0	0	0	1	1
H	1	1	0	0	1

a) (9 points) Calculate the empirical mutual information between all couples of features (including *Is poisonous*).

b) (7 points) Build tree distribution for the data according to the maximum-likelihood criteria. You have a constraint that the node of 'Is poisonous' must be the main root (head) of the tree.

c) (7 points) Use the tree you built to determine by the maximum-likelihood criteria whether U,V,W are poisonous or not. If it happens to be that there is a tie, you define it as poisonous.

Example	Is heavy	Is smelly	Is spotted	Is smooth	Is poisonous
U	0	1	1	1	?
V	0	1	0	1	?
W	1	1	0	0	?

Note:

$$h_b(\frac{1}{8}) = 0.5436, \quad h_b(\frac{1}{4}) = 0.8113, \quad h_b(\frac{3}{8}) = 0.9544, \quad h_b(\frac{1}{7}) = 0.5917, \quad h_b(\frac{2}{7}) = 0.8631,$$
$$h_b(\frac{3}{7}) = 0.9852, \quad h_b(\frac{1}{6}) = 0.6500, \quad h_b(\frac{1}{3}) = 0.9183, \quad h_b(\frac{1}{5}) = 0.7219, \quad h_b(\frac{2}{5}) = 0.9710.$$

solution:

a)

$$I(X;Y) = H(X) - H(X|Y) = H(X) - P(Y=0)H(X|Y=0) - P(Y=1)H(X|Y=1)$$

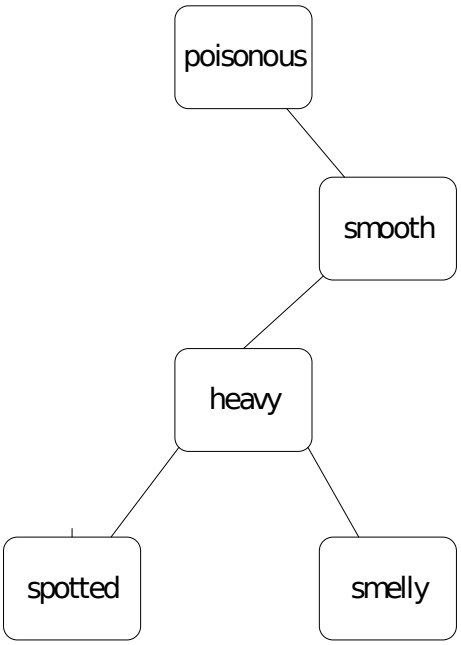
X_1=heavy.
X_2=smelly.
X_3=spotted.
X_4=smooth.
X_5=poisonous.
Therefore,

$$I(X_1;X_2) = h_b(\frac{3}{8}) - \frac{5}{8}h_b(\frac{1}{5}) - \frac{3}{8}h_b(\frac{2}{3}) = 0.1588$$
$$I(X_1;X_3) = h_b(\frac{3}{8}) - \frac{5}{8}h_b(\frac{3}{5}) - \frac{3}{8}h_b(0) = 0.3475$$
$$I(X_1;X_4) = h_b(\frac{3}{8}) - \frac{1}{2}h_b(\frac{1}{4}) - \frac{1}{2}h_b(\frac{1}{2}) = 0.0487$$
$$I(X_1;X_5) = h_b(\frac{3}{8}) - \frac{3}{8}h_b(\frac{1}{3}) - \frac{5}{8}h_b(\frac{2}{5}) = 0.00316$$
$$I(X_2;X_3) = h_b(\frac{3}{8}) - \frac{5}{8}h_b(\frac{2}{5}) - \frac{3}{8}h_b(\frac{1}{3}) = 0.00316$$
$$I(X_2;X_4) = h_b(\frac{3}{8}) - \frac{4}{8}h_b(\frac{1}{2}) - \frac{4}{8}h_b(\frac{1}{4}) = 0.0487$$
$$I(X_2;X_5) = h_b(\frac{3}{8}) - \frac{3}{8}h_b(\frac{1}{3}) - \frac{5}{8}h_b(\frac{2}{5}) = 0.00316$$
$$I(X_3;X_4) = h_b(\frac{3}{8}) - \frac{4}{8}h_b(\frac{1}{2}) - \frac{4}{8}h_b(\frac{1}{4}) = 0.0487$$
$$I(X_3;X_5) = h_b(\frac{3}{8}) - \frac{3}{8}h_b(\frac{1}{3}) - \frac{5}{8}h_b(\frac{3}{5}) = 0.00316$$
$$I(X_4;X_5) = h_b(\frac{1}{2}) - \frac{3}{8}h_b(\frac{1}{3}) - \frac{5}{8}h_b(\frac{3}{5}) = 0.0487$$

b) By the greedy algorithm that chow and liu proposed which doesn't promise an optimal solution (although that in our case it achieves the optimum), when the node "poisonous" must be the main root,we get
the largest mutual info is "heavy" with "spotted" (0.3475), therefore "heavy" is connected with "spotted".
The next largest I is between "heavy" and "smelly" (0.1588), therefore "heavy" is connected with "smelly".
The next largest I that have left is (0.0487) between "heavy" and "smooth" and between "smooth" and "poisonous",

therefore ”heavy” is connected with ”smooth” and ”smooth” is connected with ”poisonous”.

$$I(X_4;X_5) = h_b(\frac{1}{2}) - \frac{3}{8}h_b(\frac{1}{3}) - \frac{5}{8}h_b(\frac{3}{5}) = 0.0487$$
$$I(X_1;X_4) = h_b(\frac{3}{8}) - \frac{1}{2}h_b(\frac{1}{4}) - \frac{1}{2}h_b(\frac{1}{2}) = 0.0487$$
$$I(X_1;X_2) = h_b(\frac{3}{8}) - \frac{5}{8}h_b(\frac{1}{5}) - \frac{3}{8}h_b(\frac{2}{3}) = 0.1588$$
$$I(X_1;X_3) = h_b(\frac{3}{8}) - \frac{5}{8}h_b(\frac{3}{5}) - \frac{3}{8}h_b(0) = 0.3475$$



c) By using the tree to determine whether U,V,W are poisonous or not, the decision is taken by ”poisonous” son ,”smooth”, by the following function, when

$$\hat{X}_5 = \text{"}poisonous\text{"}; X_4 = \text{"}smooth\text{"}$$

$$\begin{aligned}\hat{X}_5 &= \text{argmax}_{x_5\in\{0,1\}} p(X_5 = x_5|X_4 = x_4, X_3 = x_3, X_2 = x_2, X_1 = x_1)\\ &= \text{argmax}_{x_5\in\{0,1\}} p(X_5 = x_5|X_4 = x_4)\end{aligned}$$

The last step is due to markov chain.
the probability is now calculated empirically over the train set.
we can see that U,V are poisonous because $smooth = 1$

$$P(poisonous = 1|smooth = 1) = \frac{3}{4} > \frac{1}{4} = P(poisonous = 0|smooth = 1)$$

and with W there is a tie, i.e.

$$P(poisonous = 1|smooth = 0) = \frac{1}{2} = P(poisonous = 0|smooth = 0)$$

, therefore W is ”poisonous” as well.