Homework 4

- *Homework deadline: 10:30am on Nov 8*

1. **Hidden Markov Model [40 pts]**. In class we defined a forward looking variable $\alpha_{t+1}(i) = P(O_1, \cdots, O_{t+1} \wedge q_{t+1} = s_i)$. We also defined a backward looking variable $\beta_t(i) = P(O_{t+1}, \cdots, O_n | q_t = s_i)$.

   Solution: by *Geoff Hollinger*

   (a) [**15 pts**] Use the above two definitions, show how to get the following equation.

$$P(q_t = s_i | O_1, \cdots, O_n) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)}$$

   .

$$P(q_t = s_i | O_1, \ldots, O_n) = \frac{P(q_t = s_i, O_1, \ldots, O_n)}{P(O_1, \ldots, O_n)}$$

   by definition of conditional probability

$$numerator = P(O_{t+1}, \ldots, O_n | O_1, \ldots, O_t, q_t = s_i) P(O_1, \ldots, O_t, q_t = s_i)$$

   by chain rule

$$numerator = P(O_{t+1}, \ldots, O_n | q_t = s_i) P(O_1, \ldots, O_t, q_t = s_i)$$

   by Markov property of HMM's

$$numerator = \alpha_t(i)\beta_t(i)$$

$$denominator = \sum_j P(O_1, \ldots, O_n, q_t = s_j)$$

   by marginalizing over all s_j

$$denominator = \sum_j P(O_{t+1}, \ldots, O_n | O_1, \ldots, O_t, q_t = s_j) P(O_1, \ldots, O_t, q_t = s_j)$$

   by chain rule

$$denominator = \sum_j P(O_{t+1}, \ldots, O_n | q_t = s_j) P(O_1, \ldots, O_t, q_t = s_j)$$

   by Markov property of HMM's

$$denominator = \sum_j \alpha_t(j)\beta_t(j)$$

Thus:

$$P(q_t = s_i|O_1, \ldots, O_n) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)}$$

(b) [**25 pts**] Use the above two definitions and (a), show how to get the following equation.

$$P(q_t = s_i, q_{t+1} = s_j|O_1, \cdots, O_n) = \frac{\alpha_t(i)P(q_{t+1} = s_j|q_t = s_i)P(O_{t+1}|q_{t+1} = s_j)\beta_{t+1}(j)}{\sum_j \alpha_t(j)\beta_t(j)}$$

$$P(q_t = s_i, q_{t+1} = s_j|O_1, \ldots, O_n) = \frac{P(q_t = s_i, q_{t+1} = s_j, O_1, \ldots, O_n)}{P(O_1, \ldots, O_n)}$$

by definition of conditional probability

$$denominator = \sum_k \alpha_t(k)\beta_t(k)$$

by same steps as in previous section

$$numerator = P(O_{t+1}|q_t = s_j, q_{t+1} = s_j, O_1, \ldots, O_t, O_{t+2}, \ldots, O_n)$$
$$P(q_t = s_j, q_{t+1} = s_j, O_1, \ldots, O_t, O_{t+2}, \ldots, O_n)$$

by chain rule

$$numerator = P(O_{t+1}|q_{t+1} = s_j)P(q_t = s_j, q_{t+1} = s_j, O_1, \ldots, O_t, O_{t+2}, \ldots, O_n)$$

by Markov property of HMM's

$$numerator = P(O_{t+1}|q_{t+1} = s_j)P(O_{t+2}, \ldots, O_n|q_{t+1} = s_j, q_t = s_i, O_1, \ldots, O_t)$$
$$P(q_{t+1} = s_j, q_t = s_i, O_1, \ldots, O_t)$$

by chain rule

$$numerator = P(O_{t+1}|q_{t+1} = s_j)P(O_{t+2}, \ldots, O_n|q_{t+1} = s_j)P(q_{t+1} = s_j, q_t = s_i, O_1, \ldots, O_t)$$

by Markov property of HMM's

$$numerator = P(O_{t+1}|q_{t+1} = s_j)\beta_{t+1}(j)P(q_{t+1} = s_j|q_t = s_i, O_1, \ldots, O_t)P(q_t = s_i, O_1, \ldots, O_t)$$

by chain rule

$$numerator = P(O_{t+1}|q_{t+1} = s_j)\beta_{t+1}(j)P(q_{t+1} = s_j|q_t = s_i)P(q_t = s_i, O_1, \ldots, O_t)$$

by Markov property of HMM's

$$numerator = \alpha_t(i)\beta_{t+1}(j)P(O_{t+1}|q_{t+1} = s_j)P(q_{t+1} = s_j|q_t = s_i)$$

Thus:

$$P(q_t = s_i, q_{t+1} = s_j|O_1, \ldots, O_n) = \frac{\alpha_t(i)\beta_{t+1}(j)P(O_{t+1}|q_{t+1} = s_j)P(q_{t+1} = s_j|q_t = s_i)}{\sum_k \alpha_t(k)\beta_t(k)}$$

2. **Decision Trees and Neural Nets [35 pts]**. In this problem we compare decision trees and neural nets on two functions, namely the *majority function* and the *parity function*. Both functions are logic functions from n inputs to 1 output. Majority function returns TRUE if more than half of its inputs are TRUE. Parity function returns TRUE if and only if an odd number of inputs are TRUE.

   Solution: partially by *Ian Fette*

   (a) [**12 pts**] Can a perceptron be used to solve

   - the majority function?
     Yes. For n inputs x_i, let positive inputs return +1 and negative inputs return -1. Let $w_i, \ldots, w_n = 1$. Let $w_o = 0$. Define the perceptron boundary as:

$$y = \sum w_i x_i + w_0$$

$$y = \sum_{i=1}^{n} x_i$$

     For this boundary, negative values return false, positive values return true, and zero returns false. This solves the majority function.
   - the parity function?
     No. The decision boundary of parity is not linearly separable, which can not be solved by perceptrons.

   For each of these two functions, if you say yes present the perceptron that solves it, and if you say no explain why.

   (b) [**12 pts**] Can a decision tree be used to solve

   - the majority function?
     Yes.
   - the parity function?
     Yes.

   For each of these two functions, if you say yes explain how to construct a decision tree that solves it, and if you say no explain why.

   Decision trees can be used to solve both the majority function and the parity function. A decision tree can be constructed using the following method for the majority function:

   Construct a full binary tree. The node at the root is "input 1", the nodes at the next level represent splits on "input 2", and so forth to the last level of internal nodes, which represent a split on "input n". The leafs of the tree are either "true" or "false", and are determined as follows. Each leaf node is reached by traversing a number of "true" splits and a number of "false" splits. If the number of "true" splits is greater than the number of "false" splits, the assignment to the leaf node should be true, else the assignment to the leaf node should be false.

   To implement the parity function:

   Construct a full binary tree in the same manner as for the majority function. Assign the following values to the leaf nodes: If the tree walk required to get to the leaf traverses an odd number of "true" splits, the assignment to the leaf shall be "true", else its value shall be false.

(c) [**11 pts**] For each of the functions in (a) for which you said no, can it be solved using a two layered neural network (one input layer and one hidden layer)? If so, present the neural nets that solves the problem(s); if no, how many layers would you need to solve it (assume there are n inputs)?

N-parity function can be solved using a two layered neural network. The network has n input units, n binary threshold units in the middle layer, and a single binary threshold output unit. The figure below shows a two layered neural nets that solves the problem ($http://www.mth.kcl.ac.uk/ \sim iwilde/notes/nn/nnnotespdf.pdf$).

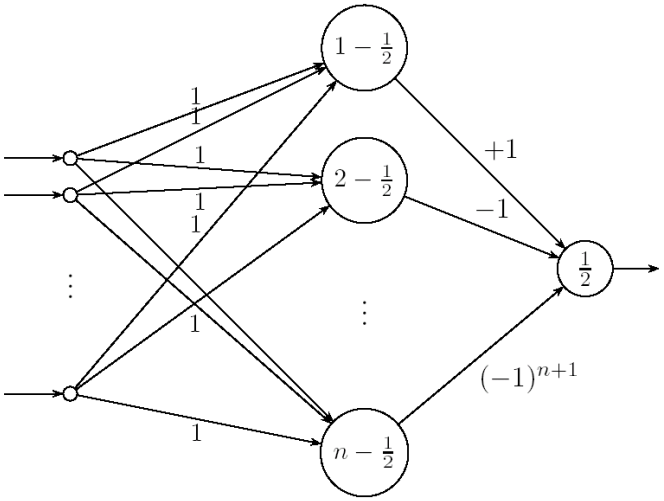

Figure 4.12: A 3-layer network implementing the n-parity function. All weights between the first and second layers are equal to 1, whereas those between the middle and output unit alternate between the values ± 1. The threshold values of the units are as indicated.

3. **Decision Trees [25 pts]**. Denote by n and p the set of negative and positive samples at a specific internal node in a decision tree. Show that if an attribute k divides the set of samples into p_0 and n_0 (for $k = 0$), and p_1 and n_1 (for $k = 1$), then the information gain from using attribute k at this node is greater or equal to 0. Hint: you may want to use the following version of Jensen's inequality:

$$\sum_{i=1}^{v} \alpha_i \log x_i \leq \log(\sum_{i=1}^{v} \alpha_i x_i)$$

where $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$.

Solution 1: : by *Ian Fette*

We offer the following proof that information gain is always nonnegative. (The exact number of positive and negative examples, n_0, n_1, p_0, p_1 are actually not important for this proof.)

Assume that the classes we are trying to distinguish between are represented by X and that the attribute we are splitting on is K. Then let us denote $P(X, K)$ to be the joint PDF of

X and K. We can obtain the marginal density $P(X)$ by summing over values of K, and vice versa. (i.e. $P(X) = \sum_K P(X, K)$ and $P(K) = \sum_X P(X, K)$.)

Our proof is as follows:

$$IG(X, K) = H(X) - H(X|K) \tag{1}$$

$$IG(X, K) = \sum_X -P(X) \log_2 P(X) - \sum_K P(K) \sum_X (-P(X|K) \log_2 P(X|K)) \tag{2}$$

$$-IG(X, K) = \sum_X P(X) \log_2 P(X) - \sum_K P(K) \sum_X (P(X|K) \log_2 P(X|K)) \tag{3}$$

$$-IG(X, K) = \sum_X \sum_K P(X, K) \log_2 P(X) - \sum_K P(K) \sum_X (P(X|K) \log_2 P(X|K)) \tag{4}$$

$$-IG(X, K) = \sum_X \sum_K P(X, K) \log_2 P(X) - \sum_K \sum_X (P(K)P(X|K) \log_2 P(X|K)) \tag{5}$$

$$-IG(X, K) = \sum_X \sum_K P(X, K) \log_2 P(X) - \sum_K \sum_X (P(X, K) \log_2 P(X|K)) \tag{6}$$

$$-IG(X, K) = \sum_X \sum_K P(X, K)(\log_2 P(X) - \log_2 P(X|K)) \tag{7}$$

$$-IG(X, K) = \sum_X \sum_K P(X, K) \left(\log_2 \left(\frac{P(X)}{P(X|K)} \right) \right) \tag{8}$$

$$-IG(X, K) = \sum_X \sum_K P(X|K)P(K) \left(\log_2 \left(\frac{P(X)}{P(X|K)} \right) \right) \tag{9}$$

$$-IG(X, K) = \sum_K P(K) \sum_X P(X|K) \left(\log_2 \left(\frac{P(X)}{P(X|K)} \right) \right) \tag{10}$$

$$-IG(X, K) \leq \sum_K P(K) \left(\log_2 \left(\sum_X \frac{P(X|K)P(X)}{P(X|K)} \right) \right) \tag{11}$$

$$-IG(X, K) \leq \log_2 \left(\sum_K \sum_X \frac{P(K)P(X|K)P(X)}{P(X|K)} \right) \tag{12}$$

$$-IG(X, K) \leq \log_2 \left(\sum_K \sum_X P(K)P(X) \right) \tag{13}$$

$$-IG(X, K) \leq \log_2 \left(\sum_K P(K) \sum_X P(X) \right) \tag{14}$$

$$-IG(X, K) \leq \log_2 \left(\sum_K P(K) \right) \tag{15}$$

$$-IG(X, K) \leq \log_2(1) \tag{16}$$

$$-IG(X, K) \leq 0 \tag{17}$$

$$IG(X, K) \geq 0 \tag{18}$$

In this proof, lines 11 and 12 are both applications of Jensen's inequality. On line 11, $\sum_X P(X|K) = 1$, and by definition each probability is nonnegative. The same argument applies for the application of Jensen's inequality on line 12.

Solution 2:

Lemma: $f(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is a concave function where $x \in (0,1)$.

Proof:

$$f'(x) = -\log_2 x + \log_2(1-x)$$

$$f''(x) = -\frac{1}{ln2} \cdot \frac{1}{x(1-x)}$$

Since $x \in (0,1)$, we have $f''(x) < 0$.

Known from concave function's property that if f is twice continuously differentiable function on R. Then f is concave if and only if $f'' \leq 0$. So we have that $f(x)$ is concave. Q.E.D.

Information gain from using attribute k at this node is:

$IG = I(\frac{p}{p+n}, \frac{n}{p+n}) - \sum_{i=0}^{1} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i})$

We want to show that $IG \geq 0$.

$$\sum_{i=0}^{1} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}) \tag{19}$$

$$= \frac{p_0+n_0}{p+n} I(\frac{p_0}{p_0+n_0}, \frac{n_0}{p_0+n_0}) + \frac{p_1+n_1}{p+n} I(\frac{p_1}{p_1+n_1}, \frac{n_1}{p_1+n_1}) \tag{20}$$

$$= \frac{p_0+n_0}{p+n} f(\frac{p_0}{p_0+n_0}) + \frac{p_1+n_1}{p+n} f(\frac{p_1}{p_1+n_1}) \tag{21}$$

$$\leq f(\frac{p_0+n_0}{p+n} \cdot \frac{p_0}{p_0+n_0} + \frac{p_1+n_1}{p+n} \cdot \frac{p_1}{p_1+n_1}) \tag{22}$$

$$= f(\frac{p_0+p_1}{p+n}) \tag{23}$$

$$= f(\frac{p}{p+n}) \tag{24}$$

$$= I(\frac{p}{p+n}, \frac{n}{p+n}) \tag{25}$$

Line (22) uses the following form of Jensen's Inequality:

$$\sum_x p(x)f(x) \leq f(\sum_x p(x)x)$$

where $\sum_x p(x) = 1, p(x) \geq 0, f(x)$ is concave.

So that $IG = I(\frac{p}{p+n}, \frac{n}{p+n}) - \sum_{i=0}^{1} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}) \geq 0$. Finally we have shown that the information gain is non-negative.