End Semester Examination

Name: Mohnish Devaraj

Reg No: 39110636

Roll No: 195115398

Date: 12 May 2022

Subject Name: Machine Learning

Subject code: SCSA1601

No. of pages: 6

3604572376800?

(32) Explain Association rule mining using the Apriori Algorithm?

Ans:

Frequent patterns are patterns that appear frequently in a data set.

- a set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent itemset.
- a subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a sequential pattern.
- a substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with items or subsequences.

Let us take an example,

TID	List of items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Suppor Count = 2

Minimum Confidence = 60%

Itemset	sup-coond
I1	6
I2	7
I3	6
I4	2
I5	2

Creating a table containing support count of each item present in dataset. After, combining the Itemset, support count having less than 2, will be removed from the itemset.

→

Itemset	sup-count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

This is the Itemset. After continuing the same step, we finally get.

Itemset	Sup. Count
I1, I2, I3	2
I1, I2, I5	2

Lets, now find the all the strong association rule using support count and confidence

Confidence $(A \Rightarrow B) = P(B/A) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$

$\{I1, I2, I5\}$

$\{I1, I2\} \Rightarrow I5$, Confidence = 2/4 = 50%.

$\{I1, I5\} \Rightarrow I2$, Confidence = 2/2 = 100%.

$\{I2, I5\} \Rightarrow I1$, Confidence = 2/2 = 100%.

$I1 \Rightarrow \{I2, I5\}$, Confidence = 2/6 = 33%.

$I2 \Rightarrow \{I1, I5\}$, Confidence = 2/7 = 29%.

$I5 \Rightarrow \{I1, I2\}$, Confidence = 2/2 = 100%.

$\{I1, I2, I3\}$

$\{I1, I2\} \Rightarrow I3$, Confidence = 2/4 = 50%.

$\{I2, I3\} \Rightarrow I1$, Confidence = 2/4 = 50%.

$\{I1, I3\} \Rightarrow I2$, Confidence = 2/4 = 50%.

$I3 \Rightarrow \{I1, I2\}$, Confidence = 2/5 = 40%.

$I1 \Rightarrow \{I2, I3\}$, Confidence = 2/6 = 33.33%.

$I2 \Rightarrow \{I1, I3\}$, Confidence = 2/7 = 28%.

As the given threshold or minimum confidence is 60%, no rule can be considered as strong association rules for the problem.

(34) Explain the perceptron learning algorithm with an example.

Ans.

A Perceptron is a neural network unit that does certain computations to detect features or business ~~intt~~ intelligence in the input data.

1. Initialise weights and threshold

Define $w_i(t_i)(0 \le i \le n)$, to be the weight from input i at time t, and θ to be the threshold value in the output node. Set w_0 to be $-\theta$, the bias and x_0 to be always 1.

Set $w_i(0)$ to small random values, thus initialising all the weights and the threshold.

2. Present Input and desired output

Present input $x_0, x_1, x_2, \ldots x_n$ and desired output $d(t)$

3. Calculate actual output

$$y(t) = f_h\left[\sum_{i=0}^{n} w_i(t)\, x_i(t)\right]$$

4. Adapt weights.

if correct $\quad w_i(t+1) = w_i(t)$

if output 0, should be 1 (class A) $w_i(t+1) = w_i(t) + x_i(t)$

if output 1, should be 0 (class B) $w_i(t+1) = w_i(t) - x_i(t)$

Various modification have been suggested to this basic algorithm. the first to introduce a multiplicative factor of less than one into the weight adoption term. This has the effect of slowing down the change in the weights.

Another algorithm was suggested by Widrow and Hoff. They proposed a learning rule known as the Widrow-Hoff delta rule, which calculates the difference between the weighted sum and the required output, called the error. Weight adjustment is then carried out in proportion to that error.

$$\Delta = d(t) - y(t)$$

where $d(t)$ is the desired response of the system, and $y(t)$ is the actual response.

- If the desired output is 1 and the actual output is 0, $\Delta = +1$ and so the weights are increased.

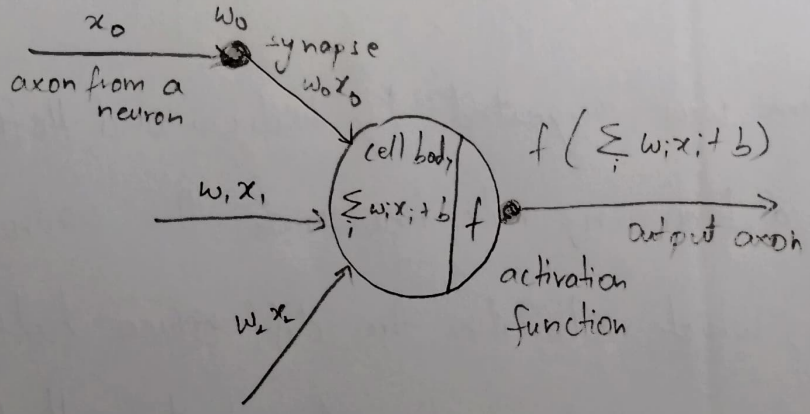
Conversely, if the desired output is 0 and the actual output is 1, $\Delta = -1$ and so the weight will be decreased.

Perceptron Learning algorithm is classified into two categories.

-> Single Layer Perceptron

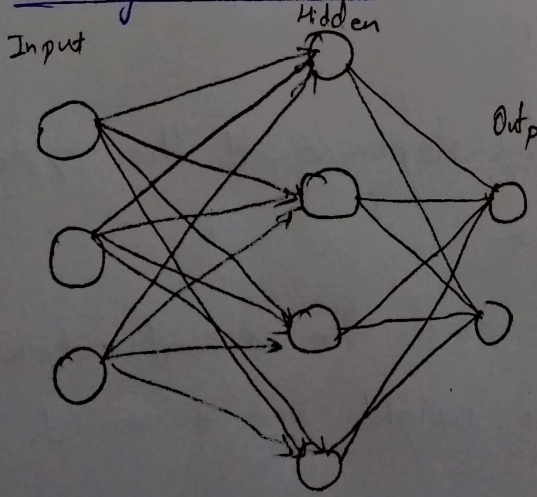
-> Multi Layer Perceptron

-> Singlelayer Perceptron



It performs a weigted sum of its input, compare this to some internal threshold level, and turns on only if this level is exceeded.

-> Multilayer Perceptron



It is a fully connected class feed forward artificial neural network.