HW3

11.05.2018

Unsupervised learning

2

$$m^{(1)} = \begin{bmatrix} 1 \\ 2 \\ 6 \\ 8 \\ 2 \end{bmatrix} \quad m^{(2)} = \begin{bmatrix} \frac{1}{3} \\ 80 \\ 8 \\ 3 \end{bmatrix}$$

First of all we need to know how to transfer from non-Euclidean space to Euclidian. The possible solution is to represent features in a different way: For example:

$$m_0 = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}$$

We, after analysing every element could understand the variety of possibilities for each feature. Assume that we discovered that there are 3 different types for feature „x", 2 different types for feature „y", 4 different types for feature „z" in $m_0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$.

x of $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

y = $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

z = $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

y of $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

That will allow us to represent the data points in the Euclidean space.

This attitude is also named as one hot encoding.

So we receive $m_0 = [0,1,0,0,1,1,0,0,0]$. The same rule will stay for other elements.

The idea behind is to represent the features, which ~~are~~ were represented as just different, to features that their difference could be measured.

a) As we have $n=5$ from the task considering suggested attitude we initialize 5 centroids randomly assigning features from "available range. Then we move towards the normal k means implementation and clusterize by lowest distance to centroid by euclidean distance in a 5-dimentional space

b) $$J(c,\mu) = \sum_{i=1}^{m} \|x^{(i)} - \mu_c^{(i)}\|^2$$

~~So it would look like this~~

Considering the above attitude having for example centroid [0,0,0,1, 001, 01, 1] and one of the elements [0001, 010, 10, 1] applying euclidean distance: $\sqrt{0^2+0^2+0^2+0^2+0^2+1^2+1^2+5^2+1^2+0} =$ $= \underline{4}$. So, basically it will have the same algorithm as ~~the~~ K-means The convergence is reached if the difference between previous iteration and current that is less than a certain epsilon.

c)

c) Consider a portion of data (mini-batch) pozmipy S. let us initialise $p = 0.5$
So in case such porsion will appear to have a half of the elements of S to have another common feature. And that one was not the one by which it was clusterised only.
After each iteration I suggest ~~edit~~ to initialise counters for every x, y, z mentioned above. Considering Hoeffding Inequality the Probability of a difference of $\mathcal{90}$ (a random variable expessed as a sum of independent R.V. of S) and Expectation of accuracy such feature $\mathcal{90}$ will be > than p. assigned. this could allow us to change the feature type over which the cluster should be initialized. Formally this Probability should be less than a certain amount ($2\exp(-2\gamma^2 m)$ from a formula).

Ending up