83%

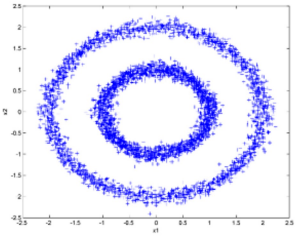
Final Exam - Duc Trinh

Tuesday, May 12, 2020 10:34 PM

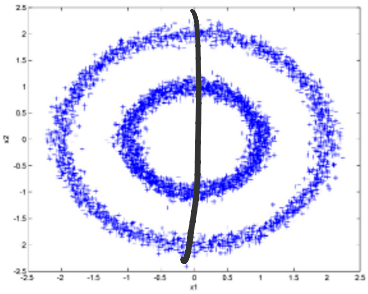
1. Perform clustering on the two-dimensional data illustrated below.
   a. Explain how the following four algorithms would split the data into **two clusters**: ***K*-means**, **hierarchical clustering** (separately **single linkage** and **complete linkage**), **DBSCAN**. Draw the clusters and give brief explanations to your answers!
   b. Let us assume that the goal is to find the two annular natural clusters. Give an $R^2 \rightarrow R^2$ **coordinate transformation** that assists the bad-performing clustering algorithms to find the natural clusters. Plot the coordinate system transformation!

(20%)

16%

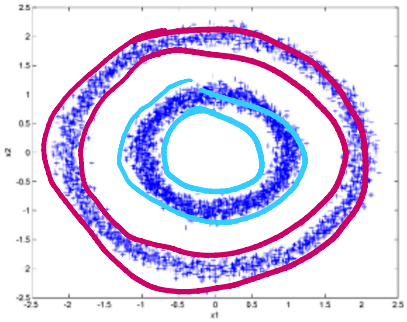


a.



← k-mean

k-means would probably split the data in half (vertically or horizontally). k-means will always form spherical clusters around its centroids. hence it will not be able to find the natural clusters here. ✓



← DBSCAN & singly linkage & complete linkage

DBSCAN would perform well here as it can handle clusters of different shapes and sizes like the ones we have here. Especially as the natural clusters are well seperated. ✓

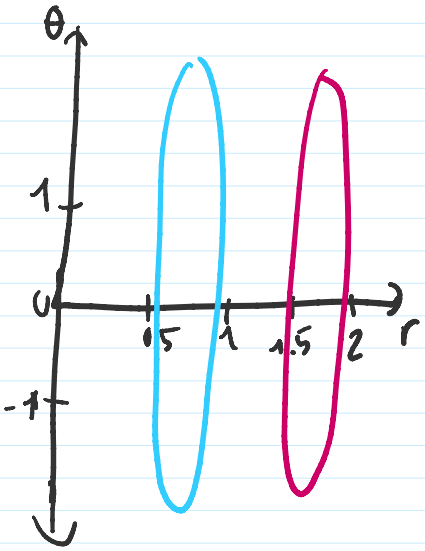
Singly linkage will also perform well as the points of each natural cluster are very close to each other so when the algorithm needs to decide which clusters to merge with, it will be able to find the natural clusters. Complete linkage will do the same by the same process

Are you sure about complete linkage? There exist points from differen natural clusters that are closer to each other than some points within the same natural clusters.

b. We could use something like a polar cordinates

θ ↑

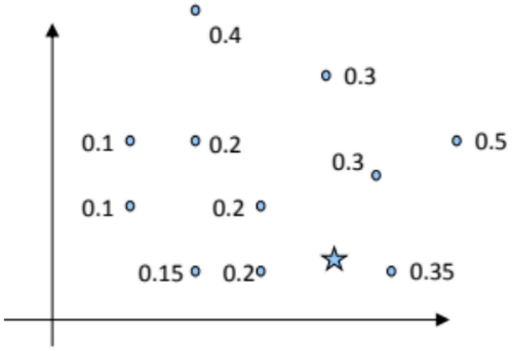
Then, we are going to have two well seperated natural clusters. ✓



Then, we are going to have two well seperated natural clusters. ✓

2. kNN regression
   a. Determine the target value of the record marked by a star using ***k*NN regression** with the choice of $k = 4$ (without using distance-weights)!
   b. Name at least **three further algorithms** that can solve **regression** problems!

(10%)



8%

a. $\frac{0.2 + 0.2 + 0.3 + 0.35}{4} = 0.2625$ ✓

b. Decision tree, linear regression, logistic regression.

Logistic regression is not a regression algorithm despite its name! It is a binary classification algorithm as you also note later!

3. Are the following statements **true or false**? **Explain** your answer! **(15%)**

12%

a. Using AdaBoost algorithm, if the *j*th classifier correctly classifies the *i*th record then the weight of the *i*th record will be certainly reduced in the *(j+1)*th step.

True. The correctly classified record's weight has to decrease so the incorrectly classified ones' can increase.

Not necessarily! If the classifier has negative importance then the weight of the misclassified record will increase in the next step.

b. Perceptron algorithm is a universal function approximator.

False. A neural network with one hidden layer can do it according to the universal approximation theorem, not a perceptron. ✓

c. One-vs-one strategy is computationally more expensive than the one-vs-rest strategy.

True. For a problem with N classes, one-vs-one has to make $\binom{N}{2}$ binary classifications. One-vs-rest makes N classifiers. ✓

d. SMOTE is an efficient undersampling method.

False. It is an oversampling method. ✓

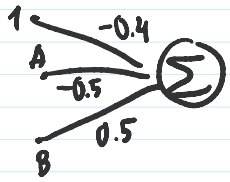
e. Logistic regression is a binary classification algorithm.

True. It still predict the probability that a new record belongs to the labeled class. We can combine multiple logistic regression models to perform multiple class classification using OvO or OvR strategy. ✓

4. Represent the "(**NOT** *A*) **AND** *B*" Boolean function with a perceptron or show that it is not possible to do so. In the latter case, construct a neural network with one hidden layer. Use the usual activation function!

10%

(10%)



$y = -0.5A + 0.5B - 0.4$ with the usual activation function ✓

5. Hierarchical clustering
   a. Using the following **distance** matrix draw the dendrograms corresponding to the single linkage (MIN) and complete linkage (MAX) clustering algorithms.
   b. What advantages does hierarchical clustering have in general compared to K-means algorithm?
   c. What advantages and disadvantages do single and complete linkage techniques have compared to one another?

20%

(20%)

Item	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

a. Single linkage

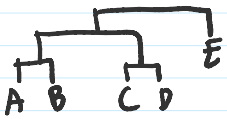
Complete linkage

a. Single linkage

Item	A∩B	C	D	E
A∩B	0	2	2	3
C	2	0	1	5
D	2	1	0	3
E	3	5	3	0

Item	A∩B	C∩D	E
A∩B	0	2	3
C∩D	2	0	3
E	3	3	0

Item	A∩B∩C∩D	E
A∩B∩C∩D	0	3
E	3	0



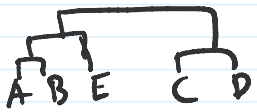
✓

Complete linkage

Item	A∩B	C	D	E
A∩B	0	2	4	3
C	2	0	1	5
D	4	1	0	3
E	3	5	3	0

Item	A∩B	C∩D	E
A∩B	0	4	3
C∩D	4	0	5
E	3	5	0

Item	A∩B∩E	C∩D
A∩B∩E	0	5
C∩D	5	0



✓

b. We don't have to decide the number of clusters before running the algorithm
The dendogram it produces carries hierarchial information.
Can possibly deal better with non-spherical clusters.
Deal better with clusters of different sizes.
Does not have to worry about choosing initial centroids. ✓

c. Singly linkage handles elliptical shapes better, but it is more sensitive for noise and outliers.

Complete linkage is less sensitive for noise and outliers, but it tends to divide large spherical clusters. ✓

6. We aim to minimize the squared error with regularization term regarding a linear regression model:

$$((w_0 + \boldsymbol{w}^T\boldsymbol{x}) - y)^2 + \lambda||\boldsymbol{w}||_2^2,$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_p)$: vector of weights (parameters, coefficients), $\boldsymbol{x}$: feature vector, λ: regularization parameter, $||\boldsymbol{w}||_2$ denotes the Euclidean length of the vector $\boldsymbol{w}$. Note that the constant term (w_0) is not regularized.

a. What is the role of regularization?
b. Give the formula for one update step of stochastic gradient descent method for w_0 and for w_i $(i \neq 0)$, with η learning rate for a training record $(x_1, x_2, \ldots, x_p)$ with target variable y.
c. What does the learning rate refer to? What are the advantages and disadvantages of using a small/large learning rate?
d. What is the difference between gradient descent method and stochastic gradient descent method?

(25%)

18%

a. To avoid overfitting.

b. Let $\vec{x} = (x_1, \dots, x_p)$. $Err^2 = (w_0 + \vec{w}^T \vec{x}) - y)^2 + \lambda \|w\|^2$

$$\vec{w}_1 = w_0 - \rho \left(\begin{matrix} \frac{\partial Err^2|\vec{x}}{\partial w_0} \\ \vdots \\ \frac{\partial Err^2|\vec{x}}{\partial x_p} \end{matrix} \right) \Bigg|_{\vec{w}_0}$$

$$\vec{w}_{i+1} = w_i - \rho \left(\begin{matrix} \frac{\partial Err^2|\vec{x}}{\partial w_0} \\ \vdots \\ \frac{\partial Err^2|\vec{x}}{\partial x_p} \end{matrix} \right) \Bigg|_{\vec{w}_i}$$

w_0 and w_i are scalars not vectors. What are the derivatives?

c. Learning rate refers to the amount by which the weights will change at every step

Using a large learning rate can makes our algorithm faster, but it might miss/skip over the maximal minima we are trying to reach. On the other hand, using a small learning rate can fix that problem, but it is going to make the algorithm slower.

d. The gradient used in stochastic gradient descent is calculate for one random record, not the entire dataset like the gradient descent.