POLITECNICO DI TORINO

Data Science Lab

Exercises

DataBase and Data Mining Group

Andrea Pasini, Elena Baralis

Theory questions

- Which statement is true?
  a) To limit over-fitting, the accuracy of a classification model must be computed on the training set
  b) To limit over-fitting, the accuracy of a classification model must be computed on a set of unlabeled data
  c) To limit over-fitting, the accuracy of a classification model must be computed on a test set with a completely different data distribution from the training set
  d) None of the previous statements is true.

Theory questions

- Solution: d)

Classification

- Given the following confusion matrix

		predicted a	predicted b	predicted c	predicted d
actual	a	10	0	0	0
actual	b	0	5	0	1
actual	c	0	2	10	0
actual	d	1	1	0	5

- Q1: compute the accuracy score
- Q2: compute F-Measure (F1) of class b

		predicted			
		a	b	c	d
actual	a	10	0	0	0
	b	0	4	0	4
	c	0	4	10	0
	d	0	2	0	6

Q1: compute the accuracy score

Q2: compute F-Measure (F1) of class b

Solution

accuracy = (10+4+10+6)/(30+4+4+2) = 30/40 = 0.75

p(b) = 4/(4+4+2) = 0.4

r(b) = 4/(4+4) = 0.5

f1 = 2*(p*r)/(p+r) = 2*(0.2)/(0.9)

Regression

- Given the following dataset, with 2 features (x0, x1) and 3 data points:
  - X = [[2, 4], [1, 2], [2, 0]]

- Apply to X the following multinomial regression pipeline
  - Feature extraction step
    - $[x_0, x_1, x_0^2, x_1^2, x_0 x_1]$
  - Regression parameters (to be applied on the extracted features)
    - B = [0, 2, 0, 1, 1/2], Bias=1

- **Q1**: What is the output vector with the predictions?
  - y_pred = [?]

- **Q2**: Given ground the ground truth predictions
  - y_truth = [28, 9, 5]
  - Compute the Mean Absolute Error (MAE) of the obtained predictions (y_pred)

X = [[2, 4], [1, 2], [2, 0]]

$[x_0, x_1, x_0^2, x_1^2, x_0x_1]$

B = [0, 2, 0, 1, 1/2], Bias=1

Solution (Q1):

[0, 2, 0, 1, 1/2]

X_poly = [[2, 4, 4, 16, 8], [1, 2, 1, 4, 2], [2, 0, 4, 0, 0]]

Apply the model:

y_pred = [0+8+0+16+4, 0+4+0+4+1, 0+0+0+0+0] + 1

= [28, 9, 0] + 1

= [29, 10, 1]

y_truth = [28, 9, 5]

y_pred = [29, 10, 1]

Solution (Q2):

MAE = 1/3 * (|28-29|+|9-10|+|5-1|) = (1 + 1 + 4)/3 = 2

Computation of indices

- Given the labels predicted by a clustering algorithm and ground truth labels:
  - y_true = [1, 1, 1, 2]
  - y_pred = [3, 3, 1, 1]
- Compute the Rand Index score (RI)
- $$RI = \frac{TP+TN}{\binom{n}{2}}$$
  - where TP = number of pairs of elements that are in the same set in y_true and in the same set in y_pred
  - TN = number of pairs of elements that are in different sets in y_true and different sets in y_pred
  - n = number of data points

0, 1, 2, 3
y_true = [1, 1, 1, 2]
y_pred = [3, 3, 1, 1]

together in y_true

together in y_pred

	true	pred	TP	TN
0-1	1	1	1	
0-2	1	0		
0-3	0	0		1
1-2	1	0		
1-3	0	0		1
2-3	0	1		

TP = 1
TN = 2

$RI = \frac{TP+TN}{\binom{n}{2}}$ = 3/6 = 0.5

PoliTo DBMG

- Given the following distance matrix (each cell describes the distance between two points)

	a	b	c	d	e	f	g
a		6	4	7	8	3	6
b	6		6	3	7	7	6
c	4	6		7	7	3	9
d	7	3	7		6	8	4
e	8	7	7	6		7	8
f	3	7	3	8	7		6
g	6	6	9	4	8	6	

- Apply DBSCAN clustering. Hyperparameters:
  - Epsilon = 5. Minpoints = 2.

Clustering

- Q1: Label each point with B(border), C (core), N(noise)

a	b	c	d	e	f	g

- Q2: Assign a cluster id to each point

a	b	c	d	e	f	g

- Q3: Compute the silhouette score of point g

	a	b	c	d	e	f	g
a		6	4	7	8	3	6
b	6		6	3	7	7	6
c	4	6		7	7	3	9
d	7	3	7		6	8	4
e	8	7	7	6		7	8
f	3	7	3	8	7		6
g	6	6	9	4	8	6	

Epsilon = 5. Minpoints = 2

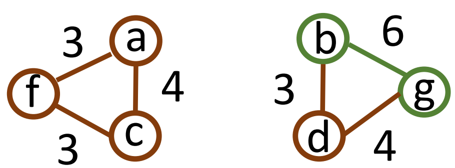
a	b	c	d	e	f	g
C	B	C	C	N	C	B

Clusters

a	b	c	d	e	f	g
1	2	1	2	-1	1	2

silh(g)?

1. Draw graph with distances



2. identify core points
   - a, c, f, d
3. identify border points
   - b, g
4. Identify clusters and noise points
5. Silhouette
   - inter(g) = (6+4)/2 = 5
   - extra(g, c1) = (ag+cg+fg)/3
     = (6 + 9 + 6)/3 = 7
   - silh(g) = (extra – inter)/max(extra, inter)
     = (7-5)/(7) = 2/7

Python-related questions

- Given two Numpy vectors
  - X with shape (100, 50)
  - y with shape (50,)

a) `np.sqrt(((X-y)**2).sum(axis=1))`
is the euclidean distance between rows of X and y and the result has shape (100, 1)

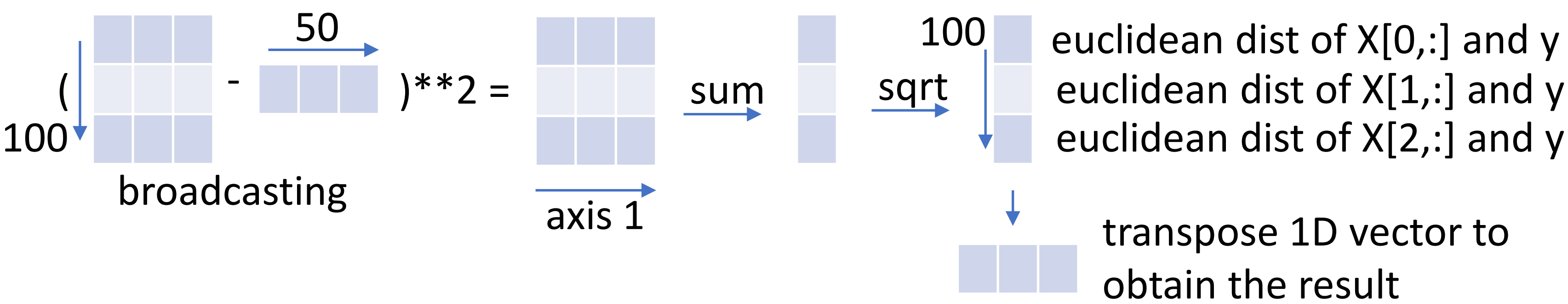
b) `np.sqrt(((X-y)**2).sum(axis=1))`
is the euclidean distance between rows of X and y and the result has shape (100,)

c) `np.sqrt(((X-y).sum(axis=1))**2)`
is the euclidean distance between rows of X and y and the result has shape (100, 1)

d) `np.sqrt(((X-y)**2).sum(axis=0))`
is the euclidean distance between rows of X and y and the result has shape (100,)

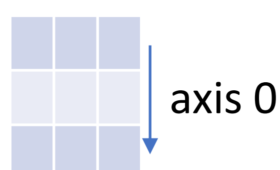
X with shape (100, 50)
y with shape (50,)

Analyze the code for a), b):
np.sqrt(((X-y)**2).sum(axis=1))



Since the result is 1-dimensional, result will have shape: (100,)
Answer **b** is correct.

Analyze the code for c):
np.sqrt(((X-y).sum(axis=1))**2) -> wrong because the square is computed after the sum of the differences
Analyze the code for d):
np.sqrt(((X-y)**2).sum(axis=0)) -> wrong because the sum is performed along axis 0



Python-related questions

- Given a Dataframe with four columns (category, year, month, #subscriptions)

a) df[['category', 'year']].pivot_table('#subscriptions', index='category', columns='year', aggfunc='mean')

returns information about the average number of subscriptions for each combination of category and year

b) df.groupby(by=['category']).sum().unstack()

returns information about the total number of subscriptions for each combination of category and year

c) df.pivot_table('#subscriptions', index='category', columns='year', aggfunc='sum')

returns information about the maximum number of subscriptions for each combination of category and year

d) df.drop(columns='month').groupby(by=['category', year']).sum().unstack()

returns information about the total number of subscriptions for each combination of category and year

e) None of the previous answers is correct

Answer: d) is correct