COMP47750 Tutorial

Nearest Neighbour Classifiers

Pádraig Cunningham

School of Computer Science

Tutorial Q1

1. The table below shows three examples from the Penguins dataset. The two labelled examples are one Adelie and one Gentoo. The type of the query example is not known.

   The four descriptive features are:

   - Bill Length: numeric, with range [30,60]mm
   - Bill Depth: numeric with range [10,20]mm
   - Flipper Length: numeric with range [170, 230]mm
   - Body Mass: numeric with range [3,000, 6,000]g

Example: *x1*	
Bill Length	39.1
Bill Depth	18.7
Flipper Len	181
Body Mass	3,750
Type	Adelie

Example: *x2*	
Bill Length	50.2
Bill Depth	14.3
Flipper Len	218
Body Mass	5,700
Type	Gentoo

Query: *q*	
Bill Length	39.5
Bill Depth	17.4
Flipper Len	186
Body Mass	3,800
Type	???

Tutorial Q1

Example: *x1*	
Bill Length	39.1
Bill Depth	18.7
Flipper Len	181
Body Mass	3,750
Type	Adelie

Example: *x2*	
Bill Length	50.2
Bill Depth	14.3
Flipper Len	218
Body Mass	5,700
Type	Gentoo

Query: *q*	
Bill Length	39.5
Bill Depth	17.4
Flipper Len	186
Body Mass	3,800
Type	???

a) Normalise all numeric features to the range [0,1]

b) Propose an appropriate global distance function for comparing examples such as the above.

c) Use your proposed distance function to calculate the distances between the query example *q* and the two labelled examples. Which class label would a 1-NN classifier assign to the query based on the distances?

Tutorial Q1a Normalisation

	X1	X2	Q	Max	Min	X1'	X2'	Q'	D(Q,X1)	D(Q,X2)
Bill L	39.1	50.2	39.5	30	60	0.30	0.67	0.32	0.01	0.66
Bill D	18.7	14.3	17.4	10	20	0.87	0.43	0.74	0.13	0.30
Flip	181	218	186	170	230	0.18	0.80	0.27	0.08	0.72
BM	3,750	5,700	3,800	3000	6000	0.25	0.90	0.27	0.02	0.88
	Adelie	Gentoo	?					Sum	0.24	2.56

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

$$X1'_{BL} = \frac{39.1 - 30}{60 - 30} = 0.30$$

$$X2'_{BD} = \frac{14.3 - 10}{20 - 10} = 0.43$$

Tutorial Q1b&c

	X1	X2	Q	Max	Min	X1'	X2'	Q'	D(Q,X1)	D(Q,X2)
Bill L	39.1	50.2	39.5	30	60	0.30	0.67	0.32	0.01	0.66
Bill D	18.7	14.3	17.4	10	20	0.87	0.43	0.74	0.13	0.30
Flip	181	218	186	170	230	0.18	0.80	0.27	0.08	0.72
BM	3,750	5,700	3,800	3000	6000	0.25	0.90	0.27	0.02	0.88
	Adelie	Gentoo	?					Sum	0.24	2.56

(b) Manhattan Distance (could also use Euclidean)

$$D(Q,X) = \sum_{f \in F} |Q_f - X_f|$$

(c) Q much closer to X1, Q is an Adelie.

Tutorial Q2a

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 2.

a. What class would a 3-NN classifier assign to **q**?

Example	Class	Distance to q
x1	over	1.5
x2	under	2.8
x3	over	1.8
x4	under	2.9
x5	under	2.2
x6	under	3.0
x7	under	2.4
x8	over	3.2
x9	over	3.6

Example	Class	Distance to q
x1	over	1.5
x3	over	1.8
x5	under	2.2
x7	under	2.4
x2	under	2.8
x4	under	2.9
x6	under	3.0
x8	over	3.2
x9	over	3.6

Sort by distance, smallest first

- Over = 2 votes
- Under = 1 vote

➡ Label **q** as 'over'

Tutorial Q2b

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 2.

b. What class would a 4-NN classifier assign to **q**?

Example	Class	Distance to q
x1	over	1.5
x2	under	2.8
x3	over	1.8
x4	under	2.9
x5	under	2.2
x6	under	3.0
x7	under	2.4
x8	over	3.2
x9	over	3.6

Example	Class	Distance to q
x1	over	1.5
x3	over	1.8
x5	under	2.2
x7	under	2.4
x2	under	2.8
x4	under	2.9
x6	under	3.0
x8	over	3.2
x9	over	3.6

Sort by distance, smallest first

- Over = 2 votes
- Under = 2 votes

➡ Tie!

Note top-ranked examples are both 'over'

Tutorial Q2c

- Pairwise distances between 9 labelled training examples and a new query example **q**, for the system described in Question 2.

c. What class would a <u>weighted</u> 4-NN classifier assign to **q**?

Example	Class	Distance to q	Weight
x1	over	1.5	1/1.5 = 0.666
x3	over	1.8	1/1.8 = 0.555
x5	under	2.2	1/2.2 = 0.454
x7	under	2.4	1/2.4 = 0.417
x2	under	2.8	...
x4	under	2.9	...
x6	under	3.0	...
x8	over	3.2	...
x9	over	3.6	...

Sort by distance, smallest first.
Calculate weight as inverse distance.

- Over = 0.666 + 0.555 = 1.221
- Under = 0.454 + 0.417 = 0.871

➡ Label **q** as 'over'

Tutorial Q3

- Two examples from a Case-based reasoning (CBR) system for estimating the price of second-hand cars are described by 6 features:

Example: *x1*

Manufacturer	Ford
Model	Fiesta
Engine Size	1,100
Fuel	Petrol
Mileage	65,000
Condition	Excellent
Price	€3,100

Example: *x2*

Manufacturer	Citroen
Model	C3
Engine Size	1,800
Fuel	Diesel
Mileage	37,000
Condition	Fair
Price	€4,500

a. Normalise all numeric features to the range [0,1]. Assume that the feature ranges are: Engine Size 1,000 to 3,000; Mileage 1,000 to 100,000.

b. Propose a suitable global distance function. Assume that Condition is an ordinal feature that has the possible values {Poor, Fair, Good, Excellent},

c. Use this measure to calculate the distance between *x1* and *x2*.

Tutorial Q3a

a. Normalise all numeric features to the range [0,1]. Note that you can assume that the feature ranges for: Engine Size is 1,000 to 3,000; Mileage is 1,000 to 100,000.

- Min-max normalisation:
  Use min and max values for a given feature to rescale to the range [0,1]

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Example: *x1*

Manufacturer	Ford
Model	Fiesta
Engine Size	**(1100-1000)/ (3000-1000) = 0.05**
Fuel	Petrol
Mileage	**(65000-1000)/ (100000-1000) = 0.646**
Condition	Excellent

Example: *x2*

Manufacturer	Citroen
Model	BX
Engine Size	**(1800-1000)/ (3000-1000) = 0.4**
Fuel	Diesel
Mileage	**(37000-1000)/ (100000-1000) = 0.364**
Condition	Fair

Tutorial Q3b

Feature	Type	Local Distance Function
Manufacturer	Categorical	Overlap function
Model	Categorical	Overlap function
Engine Size	Numeric	Absolute difference (after normalisation)
Fuel	Categorical	Overlap function
Mileage	Numeric	Absolute difference (after normalisation)
Condition	Ordinal {Poor, Fair, Good, Excellent}	Absolute relative rank difference (normalised)

Sum over local distance on each feature:

Manufacturer + Model + Engine Size + Fuel + Mileage + Condition

Tutorial Q3c

Example: *x1* (Normalised)

Manufacturer	Ford
Model	Fiesta
Engine Size	0.05
Fuel	Petrol
Mileage	0.646
Condition	Excellent

Example: *x2* (Normalised)

Manufacturer	Citroen
Model	C3
Engine Size	0.4
Fuel	Diesel
Mileage	0.364
Condition	Fair

Calculate *D(x1,x2)*

Feature	**Difference**
Manufacturer	1
Model	1
Engine Size	|0.05-0.4| = 0.35
Fuel	1
Mileage	|0.646-0.364| = 0.282
Condition	|4-2|/4 = 0.5

```
Dist = 1 + 1 + 0.35 + 1
+ 0.282 + 0.5 = 4.132
```

* subject to rounding

Tutorial Q4

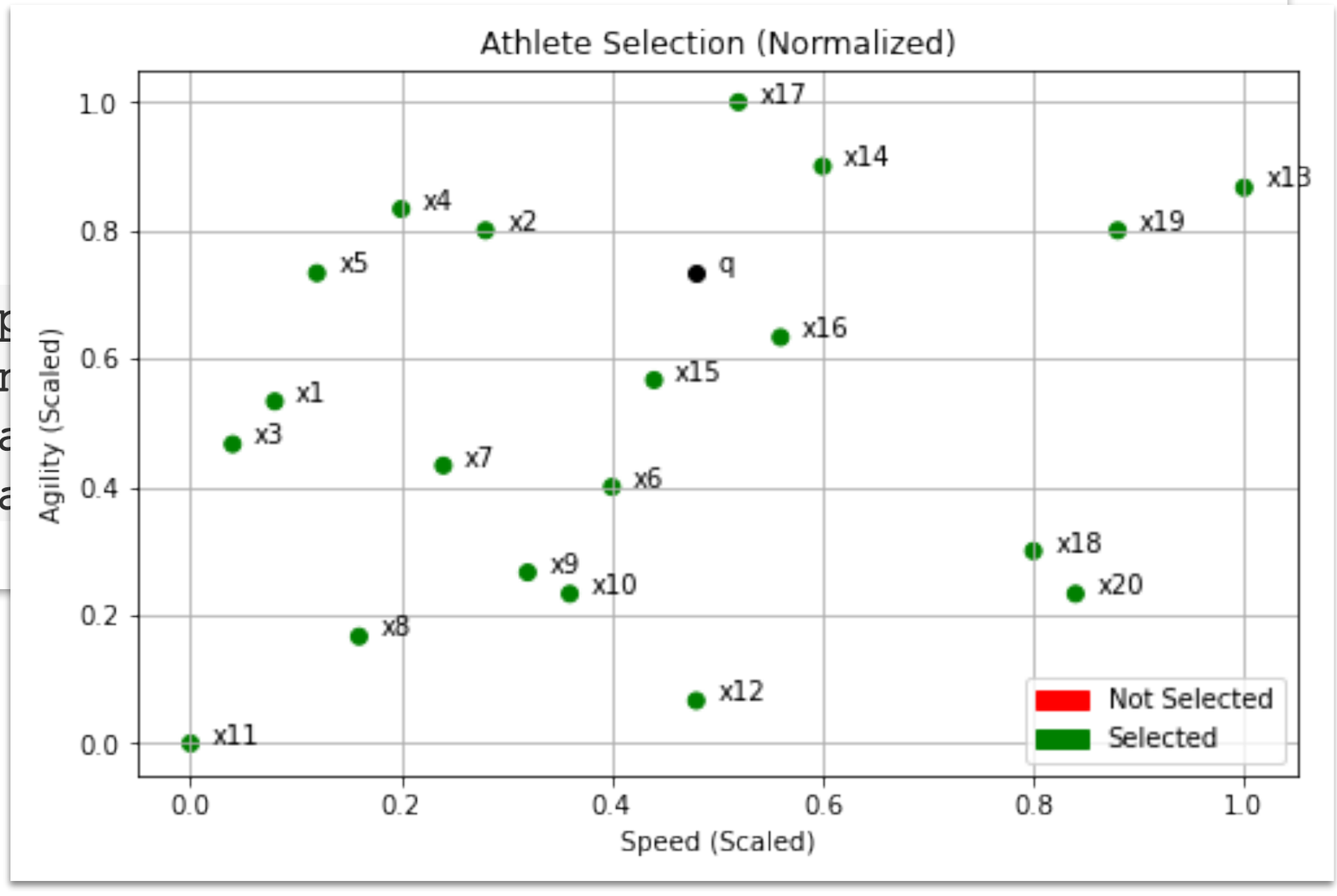
- See Solution notebook on Brightspace.
- Change the metric used by k-NN to correlation to see if it will predict the other class.

```
house_C_kNN = KNeighborsClassifier(n_neighbors=1, metric='correlation')
house_C_kNN.fit(X,y)
print('Query is classified as',house_C_kNN.predict([q])[0] )

Query is classified as C1
```

Tutorial Q5

- See Solution notebook on Brightspace.
- In the Data Normalisation example in the 02-kNN Notebook replace the N(0,1) scaler with a min-max scaler.

```
athlete = pd.read_csv('AthleteSelection.csv',index_col = 'Athlete')
y = athlete.pop('Selected').values
X = athlete.values
names = athlete.index
q = [5.0,7.5]

In [20]:
from sklearn import prep
mm_scaler = preprocessir
X_scaled = mm_scaler.tra
q_scaled = mm_scaler.tra
q_scaled
```



Tutorial Q6

- See Solution notebook on Brightspace.

```
algs = ['brute', 'ball_tree', 'kd_tree']
Out[15]:
KNeighborsClassifier(n_neighbors=3)
In [20]:
for alg in algs:
    HTRU_kNN = KNeighborsClassifier(n_neighbors=3, algorithm = alg)
    HTRU_kNN.fit(X_train,y_train)
    t_start = time.time()
    acc = HTRU_kNN.score(X_test,y_test)
    t = time.time()-t_start
    print('Time: %5.2f Accuracy: %5.2f, Algorithm: %s' % (t, acc,
alg))

Time:  2.47 Accuracy:  0.98, Algorithm: brute
Time:  1.25 Accuracy:  0.98, Algorithm: ball_tree
Time:  0.53 Accuracy:  0.98, Algorithm: kd_tree
```