MACHINE LEARNING

EXERCISES

Elements of non-parametric techniques

All the course material is available on the web site
Course web site:

Exercise 1

Given the following patterns belonging to three different classes A, B, and C:

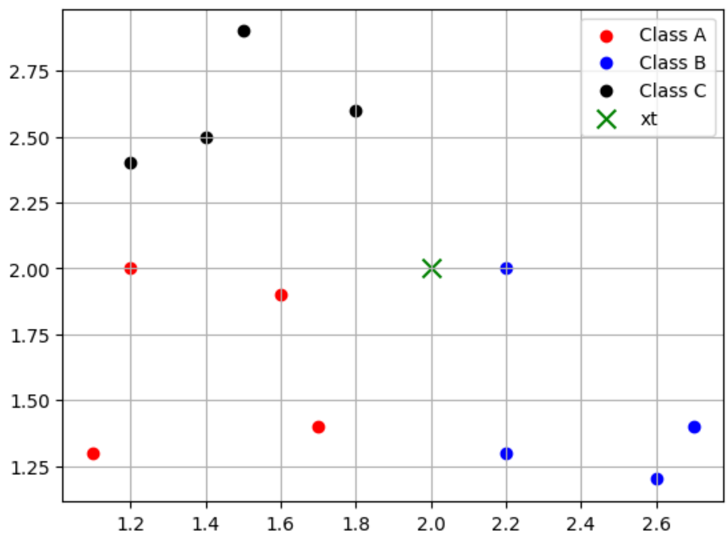
A	1.1 1.3	1.7 1.4	1.2 2.0	1.6 1.9
B	2.7 1.4	2.6 1.2	2.2 2.0	2.2 1.3
C	1.4 2.5	1.2 2.4	1.8 2.6	1.5 2.9

A) Classify the pattern x_t=(2; 2)' using the *k*NN classifier with *k*= 1, ..., 4 using the *Euclidean* and the *Manhattan* distance.

B) Use the "*leave-one-out*" method to select the best value of the "k" parameter between k=1 and k=4, using the *Euclidean* distance. The "*leave-one-out*" method works as follows:

1. Given the training set D with *n* patterns (12 patterns in this exercise)
2. for i=1, ..., *n,* use the training set {D -{x_i}} and classify the pattern x_i left out.
3. Compute the error probability (number of errors for the classifications of the *n* patterns left out)

You should use the above "*leave-one-out*" method for k=1 and k=4 and then select the value of the k parameter that provides the minimum error.



Class A: red points; Class B: blue points; Class C: black points.

Distance matrices

Squared *Euclidean* distances

	a1	a2	a3	a4
x_t	1.30	0.45	0.64	0.17

	b1	b2	b3	b4
x_t	0.85	1.00	0.04	0.53

	c1	c2	c3	c4
x_t	0.61	0.80	0.40	1.06

Manhattan distances

	a1	a2	a3	a4
x_t	1.60	0.90	0.80	0.50

	b1	b2	b3	b4
x_t	1.30	1.40	0.20	0.90

	c1	c2	c3	c4
x_t	1.10	1.20	0.80	1.40

	A1	**A2**	**A3**	**A4**	**B1**	**B2**	**B3**	**B4**	**C1**	**C2**	**C3**	**C4**
A1	0.	0.37	0.5	0.61	2.57	2.26	1.7	1.21	1.53	1.22	2.18	2.72
A2	0.37	0.	0.61	0.26	1.	0.85	0.61	0.26	1.3	1.25	1.45	2.29
A3	0.5	0.61	0.	0.17	2.61	2.6	1.	1.49	0.29	0.16	0.72	0.9
A4	0.61	0.26	0.17	0.	1.46	1.49	0.37	0.72	0.4	0.41	0.53	1.01
B1	2.57	1.	2.61	1.46	0.	0.05	0.61	0.26	2.9	3.25	2.25	3.69
B2	2.26	0.85	2.6	1.49	0.05	0.	0.8	0.17	3.13	3.4	2.6	4.1
B3	1.7	0.61	1.	0.37	0.61	0.8	0.	0.49	0.89	1.16	0.52	1.3
B4	1.21	0.26	1.49	0.72	0.26	0.17	0.49	0.	2.08	2.21	1.85	3.05
C1	1.53	1.3	0.29	0.4	2.9	3.13	0.89	2.08	0.	0.05	0.17	0.17
C2	1.22	1.25	0.16	0.41	3.25	3.4	1.16	2.21	0.05	0.	0.4	0.34
C3	2.18	1.45	0.72	0.53	2.25	2.6	0.52	1.85	0.17	0.4	0.	0.18
C4	2.72	2.29	0.9	1.01	3.69	4.1	1.3	3.05	0.17	0.34	0.18	0.

Solution

A) Classify the pattern x_t with values of *k*= 1, ..., 4 using the *Euclidean* and the *Manhattan* distance.

Squared *Euclidean* distances

	a1	a2	a3	a4
x_t	1.30	0.45	0.64	0.17

	b1	b2	b3	b4
x_t	0.85	1.00	0.04	0.53

	c1	c2	c3	c4
x_t	0.61	0.80	0.40	1.06

Classification result

k	A	B	C	Classification
1		1		B
2	1	1		A/B*
3	1	1	1	A/B/C*
4	2	1	1	A

* → the tie can be broken at random (i.e., assign the pattern at random to one of the classes)

Manhattan distances

	a1	a2	a3	a4
x_t	1.60	0.90	0.80	0.50

	b1	b2	b3	b4
x_t	1.30	1.40	0.20	0.90

	c1	c2	c3	c4
x_t	1.10	1.20	0.80	1.40

k	A	B	C	Classification
1		1		B
2	1	1		A/B *
3	2	1		A or A/B/C *
4	2	1	1	A

* → the tie can be broken at random (i.e., assign the pattern at random to one of the classes)

B) Use the "*leave-one-out*" method to select the best value of the "k" parameter between k=1 and k=4, using the *Euclidean* distance.
We compute the Euclidean distances between each pattern and all the other ones (except itself). For k=1 and k=4, then we compute the classification of the left-out pattern (in rows).
Here are the Euclidean distances between each pattern and all the other ones. In red, we highlighted the closest samples when k=1, while in bold all the k nearest neighbors with k=4.

	A1	**A2**	**A3**	**A4**	**B1**	**B2**	**B3**	**B4**	**C1**	**C2**	**C3**	**C4**
A1	0.	**0.37**	**0.5**	**0.61**	2.57	2.26	1.7	**1.21**	1.53	1.22	2.18	2.72
A2	**0.37**	0.	**0.61**	**0.26**	1.	0.85	**0.61**	**0.26**	1.3	1.25	1.45	2.29
A3	**0.5**	0.61	0.	**0.17**	2.61	2.6	1.	1.49	**0.29**	**0.16**	0.72	0.9
A4	0.61	**0.26**	**0.17**	0.	1.46	1.49	**0.37**	0.72	**0.4**	0.41	0.53	1.01
B1	2.57	**1.**	2.61	1.46	0.	**0.05**	**0.61**	**0.26**	2.9	3.25	2.25	3.69
B2	2.26	**0.85**	2.6	1.49	**0.05**	0.	**0.8**	**0.17**	3.13	3.4	2.6	4.1
B3	1.7	**0.61**	1.	**0.37**	**0.61**	0.8	0.	**0.49**	0.89	1.16	**0.52**	1.3
B4	1.21	**0.26**	1.49	0.72	**0.26**	**0.17**	**0.49**	0.	2.08	2.21	1.85	3.05
C1	1.53	1.3	**0.29**	0.4	2.9	3.13	0.89	2.08	0.	**0.05**	**0.17**	**0.17**
C2	1.22	1.25	**0.16**	0.41	3.25	3.4	1.16	2.21	**0.05**	0.	**0.4**	**0.34**
C3	2.18	1.45	0.72	0.53	2.25	2.6	**0.52**	1.85	**0.17**	**0.4**	0.	**0.18**
C4	2.72	2.29	**0.9**	1.01	3.69	4.1	1.3	3.05	**0.17**	**0.34**	**0.18**	0.

	k=1	**k=4**
A1	A	A
A2	A (50%) or B (50%)	A (75%) or B (25%)
A3	C	A (50%) or C (50%)
A4	A	A
B1	B	B
B2	B	B
B3	A	A (50%) or B (50%)
B4	B	B
C1	C	C
C2	C	C
C3	C	C
C4	C	C

Error (k=1) = from 2/12 to 3/12
Error (k=4) = from 0 to 3/12

Let's compute the expected error for the first and the second case.

For k=1, the probability of making 2 or 3 mistakes is the same, so the expected error is just 2.5/12, approximately **0.21**

For k=4, we need to compute the probability of making 0 mistakes, 1, 2, or 3.
If we want 0 mistakes, we need A2 to be classified as A, A3 as A, and B3 as B, so the probability is 75% x 50% x 50% = 3/16

If we want 1 mistake, we need to enumerate all the three cases.

- A2 is wrong, the other 2 are correct, P = ¼ x ½ x ½ = 1/16
- A3 is wrong, the other 2 are correct, P = ¾ x ½ x ½ =3/16
- B3 is wrong, the other 2 are correct, P= 3/16

… and sum them up: P=7/16

If we want 3 mistakes, we must have that all samples are wrongly classified, so P=1/16

For 2 mistakes, then you're left with P=1-3/16-7/16-1/16=5/16
(but you can double-check it by again enumerating all the cases with 2 mistakes)

So, the expected error is = 1/12 * 1/16 * (0 * 3 + 1 * 7 + 2 * 5 + 1*3) = 20/16 * 1/12 = 1,25/12 = 0,10

So, the best choice for minimizing the (expected) error rate is k=4.

Exercise 2

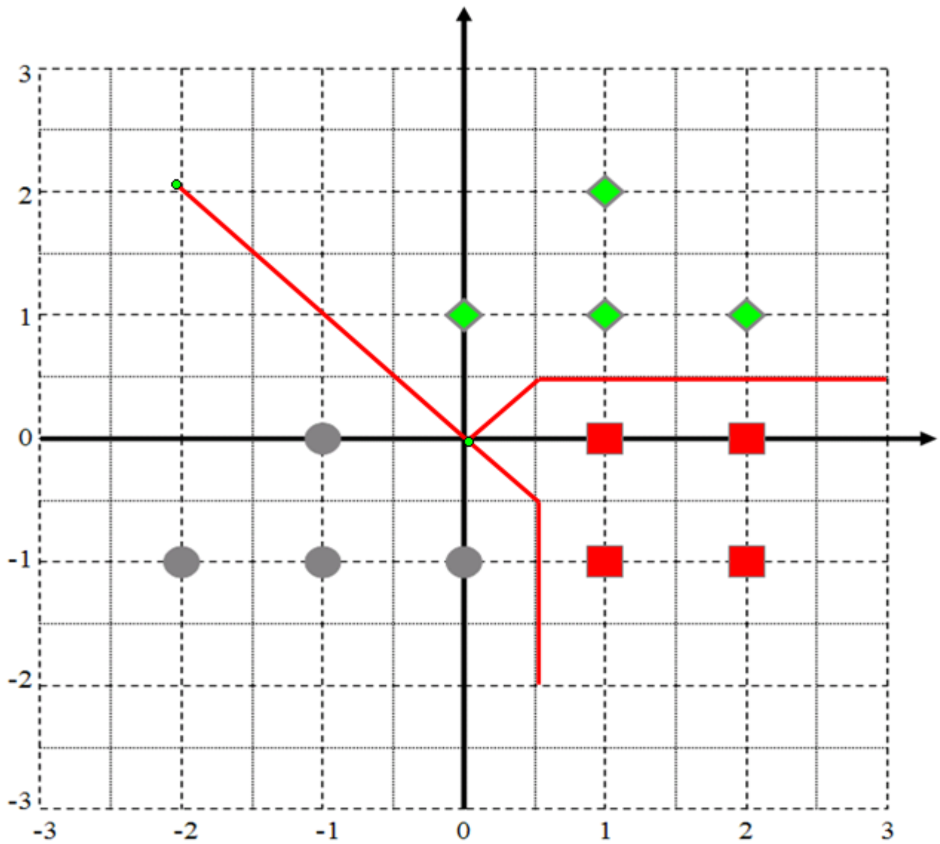
Given the patterns:

Class	ω_1				ω_2				ω_3			
	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
	0.0 1.0	1 1	2 1	1 2	1 0	2 0	1 -1	2 -1	0 -1	-1 -1	-2 -1	-1 0

1) Classify $x_t = (0.5, 0.4)^t$ using the kNN classifier and the Euclidean distance for k=1 and k=3.
2) Estimate the optimal value between k=1 and k=3 using the Manhattan metric. Explain the leave-one-out (LOO) procedure used to estimate the best k value.

If the test pattern is equidistant from different training patterns, consider those patterns with a lower index to break ties. The same goes for classes (if two or more classes have the same number of votes, choose the class with the lowest index). Use the same strategy also to break ties within the LOO procedure.

Below is the Voronoi Diagram/decision function (for k=1, using the Euclidean distance):



Distance matrices

	ω_1				ω_2				ω_3			
	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
x	0.61	0.61	2.61	2.81	0.41	2.41	2.21	4.21	2.21	4.21	8.21	2.41

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0	1	2	2	2	3	3	4	2	3	4	2
P2	1	0	1	1	1	2	2	3	3	4	5	3
P3	2	1	0	2	2	1	3	2	4	5	6	4
P4	2	1	2	0	2	3	3	4	4	5	6	4
P5	2	1	2	2	0	1	1	2	2	3	4	2
P6	3	2	1	3	1	0	2	1	3	4	5	3
P7	3	2	3	3	1	2	0	1	1	2	3	3
P8	4	3	2	4	2	1	1	0	2	3	4	5
P9	2	3	4	4	2	3	1	2	0	1	2	2
P10	3	4	5	5	3	4	2	3	1	0	1	1
P11	4	5	6	6	4	5	3	4	2	1	0	2
P12	2	3	4	4	2	3	3	5	2	1	2	0

Solution

1. Evaluate the Euclidean distance between the test pattern and the training patterns

	ω_1				ω_2				ω_3			
	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
x	**0.61**	**0.61**	2.61	2.81	**0.41**	2.41	2.21	4.21	2.21	4.21	8.21	2.41
	ω_1	ω_1			ω_2							

- $k=1$, $x \rightarrow \omega_2$
- $k=3$, $x \rightarrow \omega_1$

2. The optimal value of k in {1,3} can be estimated using the leave-one-out approach. According to this method, it is possible to classify each training pattern using all the remaining ones.

   Evaluate the distances between all the training patterns using the given metric:

$$d(p_a, p_b) = |a_1 - b_1| + |a_2 - b_2|$$

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	k=1	k=3
P1	0	**1**	**2**	**2**	2	3	3	4	2	3	4	2	ω_1	ω_1
P2	**1**	0	**1**	**1**	1	2	2	3	3	4	5	3	ω_1	ω_1
P3	**2**	**1**	0	2	2	**1**	3	2	4	5	6	4	ω_1	ω_1
P4	**2**	**1**	**2**	0	2	3	3	4	4	5	6	4	ω_1	ω_1
P5	2	**1**	2	2	0	**1**	**1**	2	2	3	4	2	ω_1	ω_2
P6	3	2	**1**	3	**1**	0	2	**1**	3	4	5	3	ω_1	ω_2
P7	3	**2**	3	3	**1**	2	0	**1**	1	2	3	3	ω_2	ω_2
P8	4	3	**2**	4	2	**1**	**1**	0	2	3	4	5	ω_2	ω_2
P9	**2**	3	4	4	2	3	**1**	2	0	**1**	2	2	ω_2	ω_1
P10	3	4	5	5	3	4	2	3	**1**	0	**1**	**1**	ω_3	ω_3
P11	4	5	6	6	4	5	3	4	**2**	**1**	0	**2**	ω_3	ω_3
P12	**2**	3	4	4	**2**	3	3	5	2	**1**	2	0	ω_3	ω_1

Err (k=1) = 3/12
Err (k=3) = 2/12 → the best result is obtained for k=3.

Exercise 3

Given the two-dimensional training points $\mathbf{x}_{\text{tr}}$, along with their labels $\mathbf{y}_{\text{tr}}$, and a set of test examples $\mathbf{x}_{\text{ts}}$, with their labels $\mathbf{y}_{\text{ts}}$

$$\mathbf{x}_{\text{tr}} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 3 & 2 \\ 1 & 2 \\ 3 & -1 \end{bmatrix}, \quad \mathbf{y}_{\text{tr}} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{\text{ts}} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y}_{\text{ts}} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix},$$

classify the points in $\mathbf{x}_{\text{ts}}$ with a k-NN algorithm with k=1, using the L2 distance as the distance metric. The distance matrix computed by comparing $\mathbf{x}_{\text{ts}}$ against $\mathbf{x}_{\text{tr}}$ is given below:

```
[[2.   2.   2.24 1.   2.83]
 [3.16 3.16 1.   1.   3.16]
 [1.41 1.41 3.61 2.24 3.16]
 [2.24 3.   2.   0.   3.61]]
```

- Compute the classification error. In case of equal (minimum) distances between a given test sample and a subset of the training points, assign the test sample to the class of the first point of the training set (from left to right in the distance matrix).
- Plot the decision function of the given k-NN classifier.

Solution

It is not difficult to see that the minimum distances per row are indexed as [3 2 0 3]:

```
[[2.   2.   2.24 1.   2.83]
 [3.16 3.16 1.   1.   3.16]
 [1.41 1.41 3.61 2.24 3.16]
 [2.24 3.   2.   0.   3.61]]
```

Recalling that the training labels are: [2 2 0 1 0], the test samples are thus classified as: yc = [1 0 2 1]. The true labels are: [0 0 2 1], and thus, the classification error is ¼ = 25%.