10 Clustering of Multi-Component Analytical Data for Olive Oils

learning objectives:

- how chemists analyze samples of olive oils for eight fatty acids, and from this can determine from which of nine regions in Italy the oils come
- the difference in classification ability between back-propagation and Kohonen learning
- how a Kohonen network can "associate" the analyses and the regions and learn to classify an oil sample as to region, given the analysis
- the way to select the appropriate Kohonen network architecture
- the significance of empty spaces in a Kohonen map, and how to deal with cases where objects are mapped into empty spaces.

10.1 The Problem

Monitoring the origins of goods is an important application of analytical chemistry in the food industry as well as among consumer protection groups.

The problem of determining geographical origin is not as hard as it sounds, because you are generally choosing from a limited set of possibilities; in fact it may be better if we rephrase the problem: how can we show that the object actually comes (or does not come) from the place named on the label. Hence, the problem is reduced to a standard **classification**.

In addition, there is the problem of how to present such results to the customer. Consumers, especially consumer activists, are not likely to accept a short answer like "yes" or "no".

In fact, there are lots of reasons for giving clear and easy-to-understand presentations of analytical results; so, the chosen classification method must show a clear picture, and must be easy to justify; it should be robust enough to allow an easy classification of unknown objects. Such a robust approach to the classification problem can be made by mapping the original multivariate objects into a two-dimensional plane and assigning (or trying to assign) the clusters of object projections formed on the map to the sought categories.

Before a procedure is accepted as a reliable classification technique, it still has to be tested with additional "unknown" objects. If the proposed mapping procedure does not provide a reliable classification, either the mapping method or the representation of the objects has to be changed.

Before the arrival of neural networks, the best method for mapping multivariate data into a two-dimensional plane was *Principal Component Analysis* (PCA). This method first calculates the correlation matrix, then diagonalizes it to obtain the eigenvalues and eigenvectors. Finally, it transforms the original data into new ones by using the matrix of eigenvectors as a transformation matrix. The map is obtained by plotting the transformed data against whatever two of the new components bring the largest portion of the information into the correlation matrix (Figure 10-1). Although the entire procedure can be made completely transparent to the user, such complex statistical calculations are hard to explain to the general public.

Therefore, a simpler method seems to be desirable. We will show here that this can be achieved by a Kohonen neural network.

Of course, if only the classification of objects is needed, the back-propagation method can be used just as well. In the following section, the same set of data is treated by both methods in a number of networks, each having a different architecture.

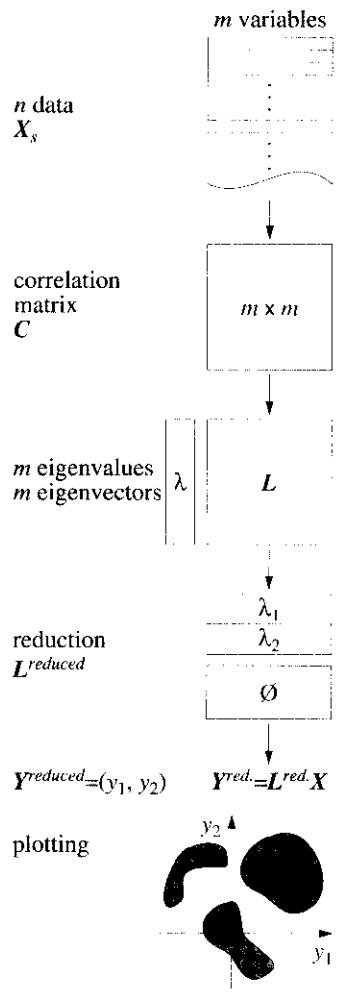

Figure 10-1: The flow of activities necessary to map data by a principal component analysis (PCA).

10.2 The Data

In order to show how problems of classifying multivariate objects can be treated by neural networks, we will use a dataset that has been extensively studied by various statistical and pattern recognition methods. This data set consists of analytical data from 572 Italian olive oils produced in nine different regions of Italy (Reference 10-1). For each oil, a chemical analysis determined the percentage of the following eight different fatty acids:

- palmitic
- palmitoleic
- stearic
- oleic
- linoleic
- arachidic (eicosanoic)
- linolenic
- eicosenoic

This formidable analytical work produced a matrix of almost five thousand values (572 x 8 = 4576) (each of which a result of a careful analysis!).

Because the proportions of some fatty acids may differ by two orders of magnitude, all values belonging to a given variable were normalized.

Table 10-1 shows how the Italian regions were numbered, and how many different oils were analyzed from each part. Figure 10-2 shows the Italian regions and their numbers.

Figure 10-2: Italian regions used in this study.

no.	region	no. of samples
1	North Apulia	25
2	Calabria	56
3	South Apulia	206
4	Sicily	36
5	Inner Sardinia	65
6	Coastal Sardinia	33
7	East Liguria	50
8	West Liguria	50
9	Umbria	51
total		572

Table 10-1: Origin and number of samples of each oil used in this study.

(The data, which are widely known among chemometricians, were kindly supplied to us by Professor Forina of the University of Genoa, Italy, for which we express our sincere thanks.)

Various groups of researchers have investigated this particular dataset using a number of statistical treatments and inspection methods, including various clustering methods, principal component and discriminate analysis. The results of these studies are mainly

published in the journals of chemometrics and analytical chemistry. Some of these studies are listed in Section 10.6.

10.3 Preliminary Exploration of Possible Networks

We started by trying a simple classification of all 572 objects of four different back-propagation and four different Kohonen neural networks. We just wanted to see how different types of networks would react to these data. The results are summarized in Table 10-2; the back-propagation trials were aimed at finding the most promising architecture (i.e., the best recall), and the Kohonen to estimate the proper size for the resulting map.

network	type	dimension	weights	time [min]	errors or conflicts
1	BPN	8 x 5 x 1	51	15.0	138
2	BPN	8 x 10 x 1	101	20.0	71
3	BPN	8 x 5 x 9	108	45.0	23
4	BPN	8 x 8 x 9	153	75.0	15
5	Kohonen	10 x 10 x 8	800	1.0	24
6	Kohonen	15 x 15 x 8	1800	2.0	14
7	Kohonen	17 x 17 x 8	2512	2.5	14
8	Kohonen	20 x 20 x 8	3200	3.0	12
(Kohonen networks used the criterion of Equation (6.1))					

Table 10-2: Characteristics of neural networks used in the preliminary step. (BPN: back-propagation network; time measured on IBM 387 compatible)

In the back-propagation network, "best recall" was defined as the smallest number of objects that cannot be learned after 400 epochs.

Two criteria were used to evaluate the Kohonen maps: the number of conflicts, and the amount of empty space. A *conflict* occurs when two objects from different classes trigger the same neuron. The number of conflicts is given in Table 10-2. An *empty space* is a neuron that is not triggered by any of the 572 oils. Too many empty spaces indicate either that the network is too large for the given set, or that the network did not spread the objects well enough across the projection plane.

In designing a back-propagation network, we have to decide; first, whether to use one or nine output lines; and second, how many neurons should be in the hidden layer. The answer to the first question is not hard: the networks with nine outputs perform far better than those with one (in Table 10-2, compare the number of errors in networks 3 and 4 against 1 and 2).

But the question about the number of neurons in the hidden layer is much harder. Fortunately, there is no need to fix the number of hidden neurons exactly at the optimum value in small networks like the ones used here. The time needed to learn the classification using the entire group of 572 objects was on the order of ten minutes on a SUN Sparc workstation. Hence, computational time is not a limiting problem. However, a network too large can increase the time considerably, with no payback in improved results; finally, we chose back-propagation networks with about 150 weights. (Don't forget to count the bias in the number of weights (see Section 2.5). Biases on Figure 10-3 are shown, as usual, as black squares.

Considering this particular problem, it can be said that the (20 x 20) neuron matrix is slightly too large for the given data set, since many of the objects are rather similar to each other and end up firing the same neuron. If we later separate the data into a training set of 250 and a test set of 322 objects, the (20 x 20) matrix would be populated too sparsely. Therefore (and for another reason that will be explained in the next section), a slightly smaller matrix was eventually chosen.

Another reason for selecting these particular back-propagation and Kohonen networks was the fact that other studies on the classification of these oils (including Principal Component Analysis (PCA), K-nearest neighbor technique (KNN), SIMCA, or three-distance clustering (3-DC)) gave a comparable number – 10 to 20 – of misclassifications or errors. The numbers of bad recalls and conflicts ("errors or conflicts" in Table 10-2) are comparable with the results from these other methods.

The objects that cause the errors or conflicts are either wrongly assigned or contain excessive experimental error. In any case, in a pool of almost 600 complex experiments, it is hard to make fewer than 2% errors in the analytical determinations.

Table 10-2 also contains the time each network needs to learn the 572 objects. The training time of the Kohonen networks is more than one order of magnitude smaller than for the back-propagation

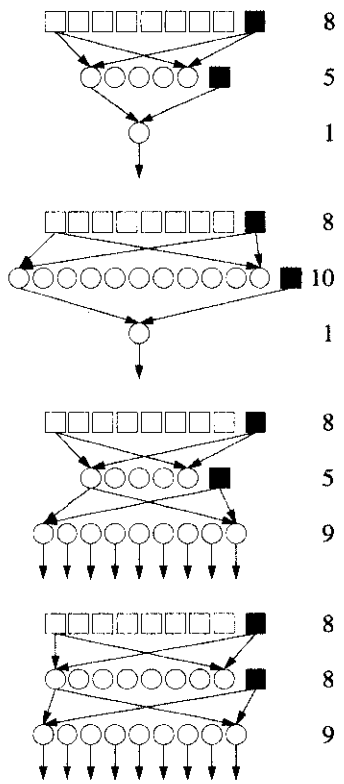

Figure 10-3: The back-propagation networks used in the preliminary study.

networks, in spite of the fact that the number of weights in the Kohonen networks is an order of magnitude larger (Figure 10-4).

The lower computation times of the Kohonen networks are due to the lower number of learning periods (epochs) required for the completion of the training: between 6 and 15, compared to 400 for the back-propagation network.

The architecture of Kohonen networks used in this study can be visualized in terms of square bricks; the base of such a brick accommodates the neurons (each of which has eight closest neighbors). The height of the brick depends on the number of weights in each neuron, which in turn is related to the number of inputs (the number of variables representing each object).

In our case, each object is represented by eight variables (the compositions of eight different fatty acids). Thus, all Kohonen neural networks are eight units high; they differ only by their bases, the areas where the future maps will be formed. All four Kohonen networks used in this study are shown schematically in Figure 10-4.

The inputs to the networks are treated as eight-dimensional column vectors (shown at the left-hand side of each network in Figure 10-4). The same input vector (complete analysis of an olive oil) goes to **all** neurons in the network simultaneously. However, **one component** of the input vector is connected to **one layer** of weights (shown darker in Figure 10-4).

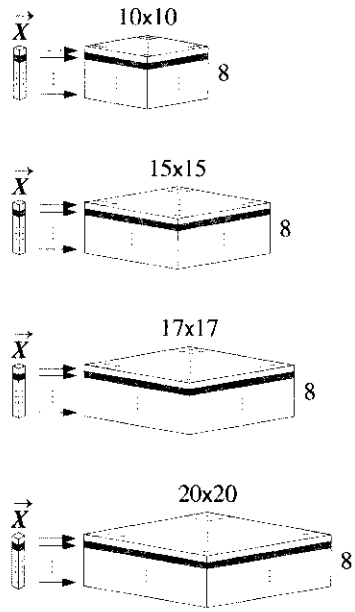

Figure 10-4: The Kohonen networks used in this study.

NOTE: the map of oil samples was produced by training the Kohonen network **without** considering toroidal boundary conditions (see Section 6.2) when making the corrections. Thus, the projection was not made onto the surface of a torus but onto a normal two-dimensional plane.

Figure 10-5 shows the map obtained from 572 records of olive oil data with the (20 x 20 x 8) Kohonen network. With respect to geographical origin, regions which are topologically close to each other are mapped into areas of the map that are also close together.

For example, the Sardinian oils – both those from the inner and those from the coastal regions, classes 5 and 6 – are separated from the rest by an obvious empty region. The oils from the northern parts – the Liguria and Umbria regions, oils 7, 8, and 9 – form a tight cluster in the upper part of the map. Those from the southern parts of Italy – from Apulia, Calabria, and Sicily, oils 1, 2, 3, and 4 – are again clearly separated from the rest by an U-shaped region extending from left to right.

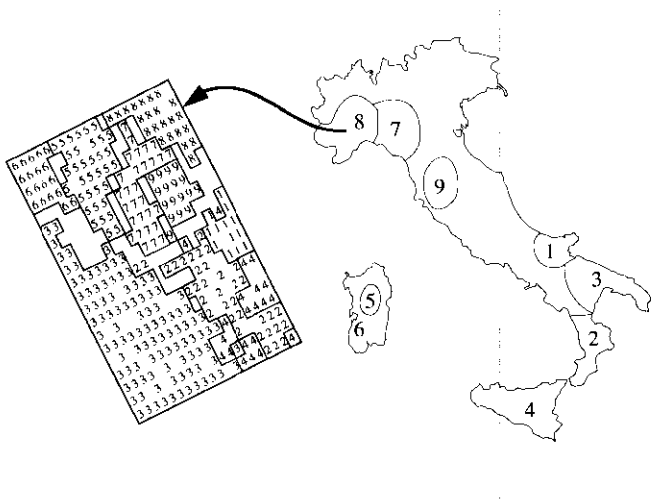

Figure 10-5: Mapping the data for olive oils on the (20 x 20) neural network.

The only significant inconsistency in the Kohonen map (compared with the actual map of oil growing regions in Italy) is the position of the South Apulian oils (class 3), which is in the lower left instead of the lower-right corner of the map. However, this correspondence between the geographical map and the Kohonen map in this example is purely fortuitous.

This clustering is remarkably good, considering the simple learning scheme used to produce it. Not only is there a clear gap between the southern and the northern oils, there is even a clear separation between the two types of Sardinian oils, and of those two types from the rest. We can safely say that there is a clear correlation between the topology of labeled regions in the Kohonen map and the actual positions of the oil-producing areas in Italy.

Two groups that are not as homogeneous as others are the Sicilian and Calabrian oils, which intersect each other. It is interesting to note that most of the errors in classification are produced by the oils from these two regions. This exception is quite instructive: even if excellent results are produced overall, we should remember that our input objects do not actually have any explicit geographical information embedded in their representation.

10.4 Learning to Make Predictions

All this preliminary work was done to observe the behavior of different networks with a certain group of data, rather than to inspect the data itself. In the second part of this study, we will find out how good our networks are at **learning**. We will divide the data into a training set and a test set. We will ignore the possibility of overtraining (Section 8.6.2); therefore we will not need a control set for signaling when to stop the training. With such simple networks as these, it is sufficient to let the learning process run until no further improvements can be detected.

Table 10-3 shows how the data were split into the two sets. We used two conditions: first, both sets should be approximately of the same size (if possible, the test set should be slightly larger than the training set); second, the training set should be as homogeneous as possible (it should contain approximately the same number of objects from each class). An good division is 270 objects for training and 302 for testing (the 270 comprise 30 objects from each of the 9 classes). But not all classes have equal numbers of data, so there has to be some tradeoff between smaller and larger groups. Table 10-3 shows the final distribution of the data objects.

no.	region	label	training	test
1	North Apulia	NA	15	10
2	Calabria	CA	35	21
3	South Apulia	SA	40	166
4	Sicily	SI	20	16
5	Inner Sardinia	IS	30	35
6	Coastal Sardinia	CS	20	13
7	East Liguria	EL	30	20
8	West Liguria	WL	30	20
9	Umbria	UM	30	21
total			250	322

Table 10-3: Splitting the data into the training and test group.

The objects within each group were selected at random with no prior inspection of their behavior, or knowledge of how they group together.

The training set was first used with the four back-propagation networks described above (Table 10-2). Although it was clear from the beginning that the first two networks (with only one output neuron

each) will yield worse predictions than those with nine, they were included in the experiment to show examples of bad design (Table 10-4).

All learning procedures were carried out by applying the set of equations given in Section 8.7 and using value of 0.2 and 0.4 for the learning rate η and the momentum μ.

network	dimension	errors	
		recall from 250	predictions from 322
1	8 x 5 x 1	104 (42)	121 (38)
2	8 x 10 x 1	89 (36)	117 (36)
3	8 x 5 x 9	5 (22)	25 or 27 (8)
4	8 x 8 x 9	1 (0.4)	31 (10)

Table 10-4: Prediction ability of different back-propagation neural networks (percentages in parentheses).

The networks with only one output neuron gave almost 40% wrong predictions and thus clearly are inadequate for a nine-class identification. On the other hand, it is surprising that the network with eight neurons in the hidden layer is less successful in learning compared to the one having five (as demonstrated by the number of errors in predictions for the test set). Although the difference is not very large, this demonstrates that larger networks do not necessarily yield better performance.

Thus, the best performer is the (8 x 5 x 9) network, which gives only 25 wrong classifications for the 322 data in the test set. This 92% prediction ability can be considered quite good. The number 27 shown together with number 25 in Table 10-4 means that two more objects would be classified wrongly if a stricter criterion were selected for class membership. The issue of criteria for class membership will be explained soon.

Now let's compare the desired output (targets) and the actual output values.

For a neural network with nine output neurons, we have to provide nine-element target vectors $\boldsymbol{Y}$. In the ideal case, only one output neuron should have an output signal equal to 1 (associated with a particular input class), while all other output neurons produce zero. The 9-variable targets are coded as follows:

class		NA	CA	SA	SI	IS	CS	EL	WL	UM	
Y (class 1)	= (	1	0	0	0	0	0	0	0	0	)
Y (class 2)	= (	0	1	0	0	0	0	0	0	0	)
Y (class 3)	= (	0	0	1	0	0	0	0	0	0	)
Y (class 4)	= (	0	0	0	1	0	0	0	0	0	)
Y (class 5)	= (	0	0	0	0	1	0	0	0	0	)
Y (class 6)	= (	0	0	0	0	0	1	0	0	0	)
Y (class 7)	= (	0	0	0	0	0	0	1	0	0	)
Y (class 8)	= (	0	0	0	0	0	0	0	1	0	)
Y (class 9)	= (	0	0	0	0	0	0	0	0	1	)

The actual output from the networks was seldom like this; more often the output values of neurons that should be zero were around 0.1, and the values of the neurons signaling the correct class had values ranging from 0.95 to 0.5.

All of the 25 wrong predictions produced the largest signal on the **wrong** neuron. But when a stricter criterion is used, the two additional cases have to be considered. Both have their largest output on a neuron signaling the **correct** class, but in one case this largest signal was only about 0.45, with all other signals being much smaller; in the second case, the largest signal of the correct neuron was 0.9, but the second largest signal was well over 0.55.

Strictly speaking, the total error in both of the anomalous answers was more than 0.5; so, we could simply dismiss them as being wrong. But "right" and "wrong" are relative terms in science; if we consider only the neuron with the largest signal, regardless of its absolute strength, both can be regarded as correct answers.

The second learning experiment was set up on the Kohonen networks (see Table 10-2). These were trained by the same 250-member training set as used for the back-propagation networks, and then tested for predictions with the remaining 322 member test set.

Because a Kohonen network is usually not used for making predictions, we have to discuss in more detail how this can be done.

To review, the result of the Kohonen network is a **two**-dimensional map of assigned neurons, each of which carries a "tag" or "label" of the object that excited it at the final recall test. If more than one object excites one neuron, it is hoped that all of them belong to the same class. Even more so, it is expected that the neurons most excited by objects of the same class will form clusters or small regions on such a map.

If a test object falls into such a cluster, it can be classified as belonging to the group corresponding to this cluster. The region where the objects of a certain class excite the neurons can form tight borders with regions formed by other classes, or such regions can be separated by empty spaces, corresponding to neurons not excited by any object from the training set. Sometimes, empty spaces appear within the region of a class (Figure 10-6). An input object that maps into an empty space may still be classifiable, as we will now see.

Along tight borders of two or more regions, it can easily happen that the **same neuron is excited by two objects belonging to different classes**. Such cases are called *conflicts* and the neurons are called *conflicting neurons*. Conflicts can occur in the recall process, but much more often they happen during the later prediction phase. If the input object excites a neuron in the wrong region, this is clearly a conflict situation. On the other hand, if the excited neuron corresponds to an empty region, the class membership of the **neighboring** neurons can help us decide whether to consider it a conflict.

Before trying to settle this question, let's inspect the predictions made by different Kohonen networks. Table 10-5 shows the prediction abilities of eight Kohonen networks. Four were obtained from our preliminary investigation and four from additional investigations. This table contains the correct and wrong classifications, as well as the numbers of empty spaces.

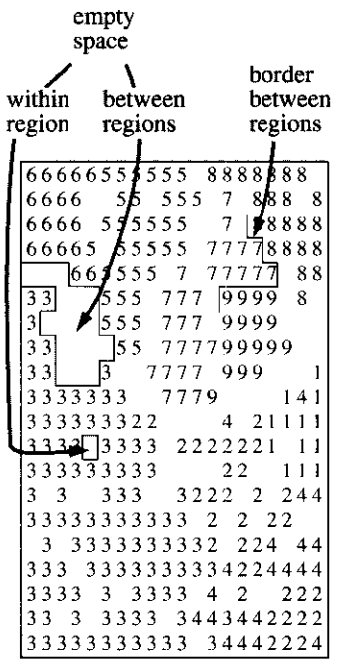

Figure 10-6: Borders between regions, and empty spaces between and within the regions representing classes.

no.	dimension	network map size	learning: empty spaces	learning: conflicts	predictions: correct	predictions: hits into empty space	predictions: hits into wrong space
1	20 x 20 x 8	400	193	0	216	108	8
2	17 x 17 x 8	289	108	2	215	96	11
3	15 x 15 x 8	225	64	1	251	60	11
4	10 x 10 x 8	100	19	2	280	25	17
5	7 x 7 x 8	49	5	9	290	3	29
6	5 x 5 x 8	25	2	27	285	0	37
7	4 x 4 x 8	16	2	33	247	0	75
8	3 x 3 x 8	9	0	57	267	0	55

Table 10-5: Performance of various Kohonen networks in learning and predicting geographical origins of olive oils.

All objects were mapped by exactly the same Kohonen learning procedure with the same training set of 250 objects, the only difference being the size of the network.

The **learning** produced only two conflicting neurons in the (10 x 10) and (17 x 17) networks, one conflicting neuron in the (15 x 15) network, and none in the (20 x 20) network. Thus, it can be said that a Kohonen network of adequate size has a good recall or good recognition ability.

The **prediction** ability of each network was tested with the same set of 322 oils, and it was found that an increase of the size of the Kohonen maps improves the prediction ability from 17 to only 8 mistakes; however, the number of hits into empty spaces increases at the same time by a factor of four: from 25 to 108.

The factor of four correlates with the increase the size in the Kohonen map from 100 to 400 neurons. It makes sense that the number of empty spaces ("unused" neurons) will increase, as the network size increases, and that the number of hits into empty spaces also depends on network size. As Figure 10-7 shows, the latter relationship is linear – within limits. For networks much larger than (20 x 20), there will be many more empty spaces, but because of the limited number of data the number of hits into empty spaces will level off. With very small maps, the number of empty spaces cannot be linearly related to the number of empty space hits because there wouldn't be any. Rather, the number of conflicts increases (Table 10-5).

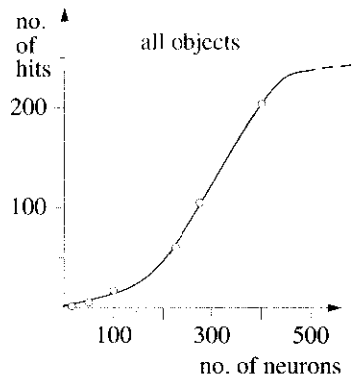

Figure 10-7: Number of hits into empty space as a function of map size.

Based on this discussion, we can see, that our network sizes were well chosen.

Because there always exist a certain number of repulsive interactions between certain classes, it is hard to generate a map completely covered by hits with no empty spaces. To investigate this, we generate four more Kohonen networks producing 49-, 25-, 16-, and 9-neuron maps. It was found that only the (3 x 3) map does not contain any empty spaces; the (4 x 4) and the (5 x 5) maps have two, while the (7 x 7) map has five empty spaces. So we see that, in designing Kohonen networks a compromise between the number of hits into empty spaces and the number of conflicts has to be found.

As Table 10-6 shows, when we test the network for prediction, there are quite a number of hits into empty spaces. It seems therefore worthwhile to explore these cases in more detail.

In order to make a guess about class membership for objects that map into empty spaces, we can try the K-nearest neighbor (KNN) technique, which determines the class by counting a number k of

closest neighbors, with the majority determining the class of the central object.

class	no. of objects	network 10 x 10	15 x 15			17 x 17	20 x 20
1	10	0	1	(1	–	–)	3	5
2	21	3	6	(4	2	–)	7	9
3	166	20	32	(32	–	–)	59	52
4	16	2	6	(3	–	3)	7	8
5	35	0	1	(1	–	–)	3	11
6	13	1	2	(–	–	2)	0	4
7	20	0	5	(3	2	–)	5	10
8	20	0	7	(7	–	–)	10	6
9	21	0	0	(–	–	–)	3	2
total	322	26	60	(51	4	5)	96	108

Table 10-6: Number of hits into empty spaces in predictions on the 322-object test set.

A detailed count was made for all 60 empty-space hits of the (15 x 15) map. The numbers of correct, undecided and wrong classifications for each class are given in parentheses in Table 10-6.

It was found that 51 of the hits would be classified correctly based on the majority vote of the eight closest neighbors. Four were undecided, which means that an equal number of these neighbors belong to two different classes; and only five would be classified wrongly on this basis. Now, we can add these figures to the corresponding figures (Table 10-5) for correct and wrong answers, 251 and 11, respectively, predicted by the (15 x 15). Altogether, we infer that by considering the KNN decision for empty space hits **in addition** to the predictions made by hits in labeled areas, the (15 x 15) network produces 302 (= 51 + 251) correct, four undecided, and 16 (= 5 + 11) wrong classifications.

Figure 10-8 shows the case where an unlabeled neuron, i.e., an empty space, has two neighbors from class 2, two neighbors from class 3 and four additional blanks.

Comparing Figures 10-8 and 10-9, it appears that the neighborhoods of the wrong hits were quite different from those of the hits into "empty space". Working with a well-diversified test set (Table 10-3), we were able to check predictions involving a variety of topological configurations.

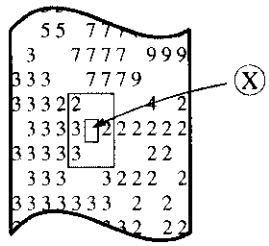

Figure 10-8: Out of eight neighbors of an unlabeled neuron, two are members of class 2 and two are members of class 4; the rest are blanks. The prediction cannot be made.

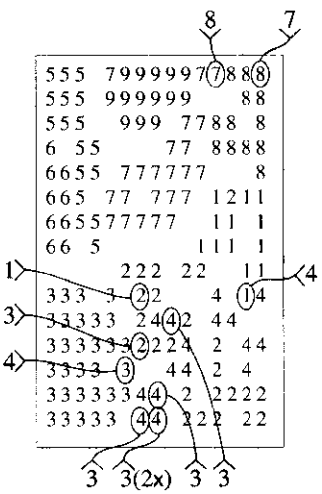

Figure 10-9: The 11 cases in the (15 x 15) network for which the predictions were wrong.

The (15 x 15) network makes a total of 11 wrong classifications (see Table 10-5); Figure 10-9 shows all neurons (circles) corresponding to wrong hits during the 322-object test.
In the eleven "wrongs" shown in Figure 10-9, it can be seen that

- the West Liguria oil (8) has been assigned as an East Ligurian one (7), and vice versa,
- both Sicilian oil mistakes (4) were predicted on the borders of Sicilian oils (once with North Apulia, (1), and once with South Apulia, (3)),
- all six South Apulia oil mistakes were made within (five cases) or in contact with (one case) the North Apulia oils, (4),
- only one case: the North Apulian oil, class (1), triggered a neuron far away from its own group. But even this "far away shot" places the North Apulian sample into the Calabrian region, (2), on the border with the South Apulian ones.

It is encouraging to note that some valuable information can be obtained even from the wrong predictions. All the errors involve objects being out into neighboring classes in the geographical sense; it turns out that the identification of origins as "Northern Italy" or "Southern Italy" was made 100% correctly for all samples.

10.5 Concluding Remarks

The example presented in this chapter was selected to show the advantages of the Kohonen network. It is useful when the topology of the classes is of interest; you may also use it in all preliminary researches where the number of clusters and the relations among them are not known. Other uses of Kohonen networks are discussed in Chapters 11 and 19. In many cases however, the back-propagation methods can give considerably better results than a Kohonen network.

10.6 References and Suggested Readings

10-1. M. Forina and C. Armanino, "Eigenvector Projection and Simplified Non-linear Mapping of Fatty Acid Content of Italian Olive Oils", *Ann. Chim. (Rome)* **72** (1982) 127 – 143; M. Forina, E. Tiscornia, ibid. 144 – 155.

10-2. M. P. Derde and D. L. Massart, "Extraction of Information from Large Data Sets by Pattern Recognition", *Fresenius' Z. Anal. Chem.* **313** (1982) 484 – 495.

10-3. M. P. Derde and D. L. Massart, "Supervised Pattern Recognition: the Ideal Method?", *Anal. Chim. Acta* **191** (1986) 1 – 16.

10-4. J. Zupan and D. L. Massart, "Evaluation of the 3-D Method with the Application in Analytical Chemistry", *Anal. Chem.* **61** (1989) 2098 – 2182.

10-5. D. L. Massart and L. Kaufmann, *Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*, Wiley, New York, USA, 1983.

10-6. B. Everitt, *Clustering Analysis*, Heineman, London, UK, 1975.

10-7. D. L. Massart, B. G. M. Vandeginste, L. M. C. Buydens, S. De Jong, P. J. Lewi and J. Smeyers-Verbeke, *Handbook of Chemometrics and Qualimetrics: Part A*, Elsevier, Amsterdam, NL, 1997.

10-8. B. G. M. Vandeginste, D. L. Massart, L. M. C. Buydens, S. De Jong, P. J. Lewi and J. Smeyers-Verbeke, *Handbook of Chemometrics and Qualimetrics: Part B*, Elsevier, Amsterdam, NL, 1998.

10-9. J. Zupan, M. Novic, X. Li and J. Gasteiger, "Classification of Multicomponent Analytical Data of Olive Oils Using Different Neural Networks", *Anal. Chim. Acta* **192** (1994) 219 – 234.

10-10. X. Li, J. Gasteiger and J. Zupan, "On the Topology Distortion in Self-Organizing Feature Maps", *Biol. Cybern.* **70** (1993) 189 – 198.

10-11. B. Kocjancic and J. Zupan, "Application of a Feed-Forward Artificial Neural Network as a Mapping Device", *J. Chem. Inf. Comput. Sci.* **37** (1997) 985 – 989.

10-12. J. Zupan, M. Novic and I. Ruisanchez, "Tutorial: Kohonen and Counterpropagation Artificial Neural Networks in Analytical Chemistry", *Chemom. Intell. Lab. Syst.* **38** (1997) 1 – 23.