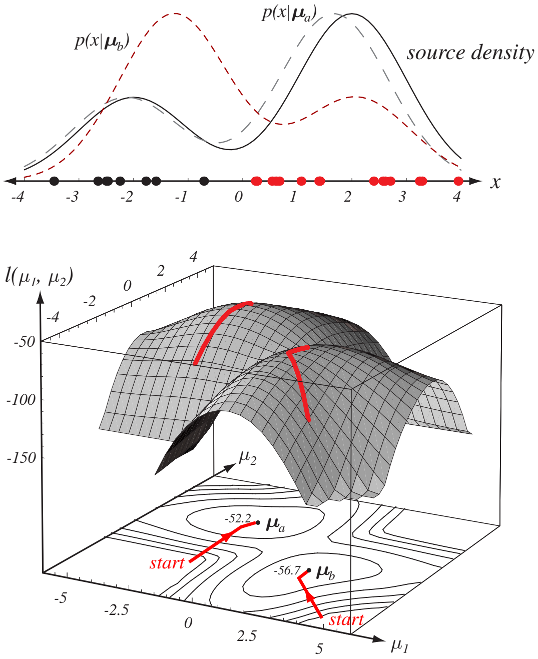

FIGURE 10.1. (Above) The source mixture density used to generate sample data, and two maximum-likelihood estimates based on the data in the table. (Bottom) Log-likelihood of a mixture model consisting of two univariate Gaussians as a function of their means, for the data in the table. Trajectories for the iterative maximum-likelihood estimation of the means of a two-Gaussian mixture model based on the data are shown as red lines. Two local optima (with log-likelihoods −52.2 and −56.7) correspond to the two density estimates shown above. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

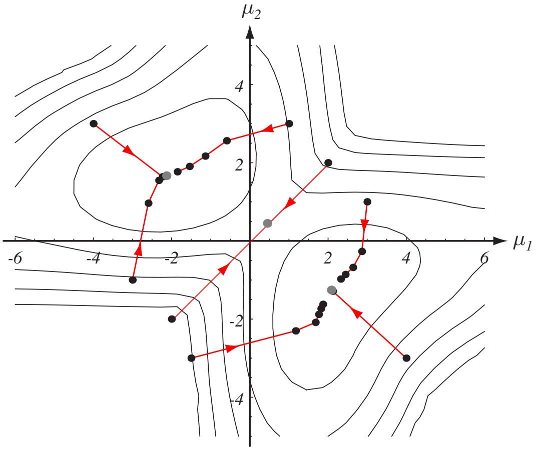

FIGURE 10.2. The *k*-means clustering procedure is a form of stochastic hill climbing in the log-likelihood function. The contours represent equal log-likelihood values for the one-dimensional data in Fig. 10.1. The dots indicate parameter values after different iterations of the *k*-means algorithm. Six of the starting points shown lead to local maxima, whereas two (i.e., $\mu_1(0) = \mu_2(0)$) lead to a saddle point near $\boldsymbol{\mu} = \mathbf{0}$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

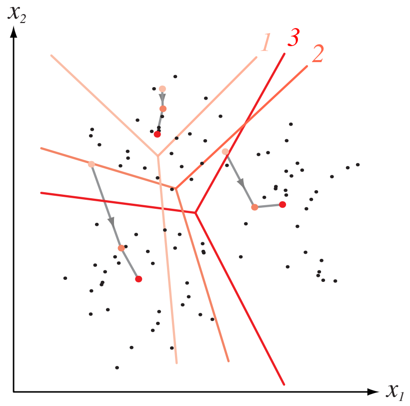

FIGURE 10.3. Trajectories for the means of the *k*-means clustering procedure applied to two-dimensional data. The final Voronoi tesselation (for classification) is also shown—the means correspond to the "centers" of the Voronoi cells. In this case, convergence is obtained in three iterations. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

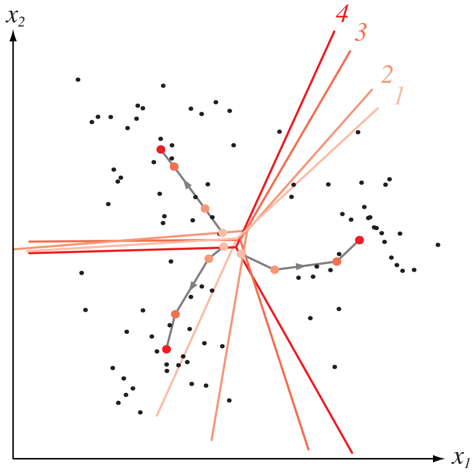

FIGURE 10.4. At each iteration of the fuzzy k-means clustering algorithm, the probability of category memberships for each point are adjusted according to Eqs. 32 and 33 (here $b = 2$). While most points have nonnegligible memberships in two or three clusters, we nevertheless draw the boundary of a Voronoi tesselation to illustrate the progress of the algorithm. After four iterations, the algorithm has converged to the red cluster centers and associated Voronoi tesselation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

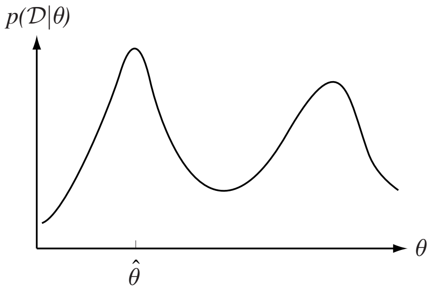

FIGURE 10.5. In a highly skewed or multiple peak posterior distribution such as illustrated here, the maximum-likelihood solution $\hat{\theta}$ will yield a density very different from a Bayesian solution, which requires the integration over the full range of parameter space θ. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

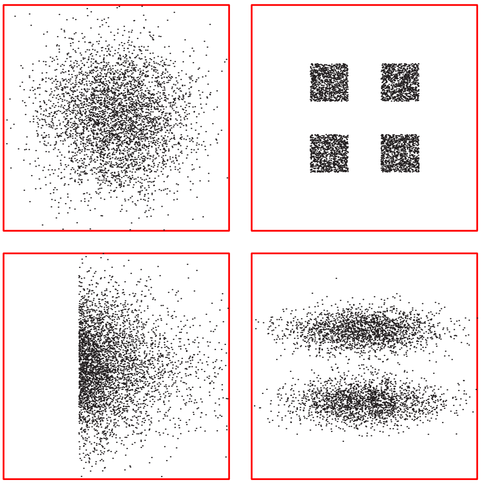

FIGURE 10.6. These four data sets have identical statistics up to second-order—that is, the same mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. In such cases it is important to include in the model more parameters to represent the structure more completely. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

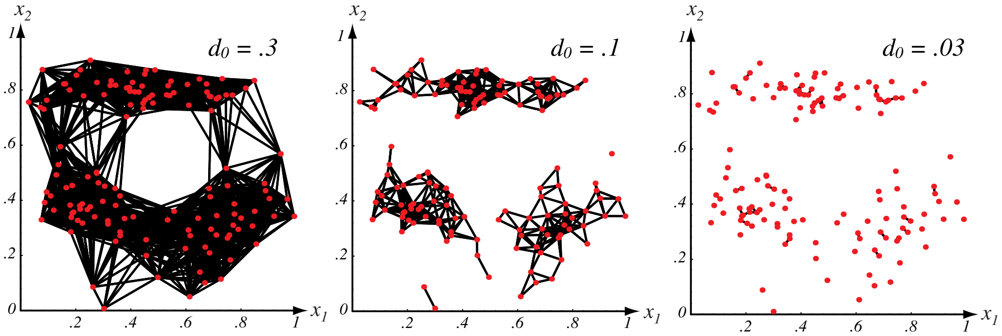

FIGURE 10.7. The distance threshold affects the number and size of clusters in similarity based clustering methods. For three different values of distance d_0, lines are drawn between points closer than d_0—the smaller the value of d_0, the smaller and more numerous the clusters. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

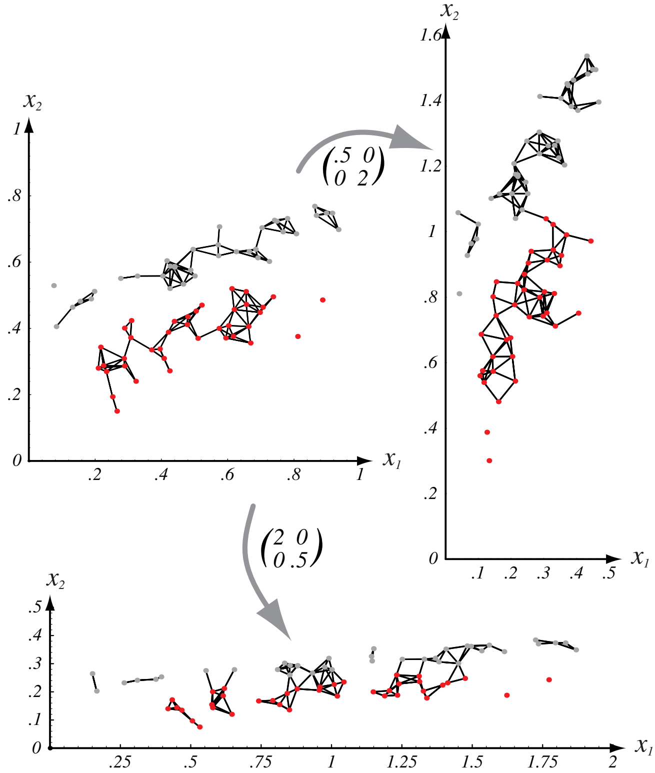

FIGURE 10.8. Scaling axes affects the clusters in a minimum distance cluster method. The original data and minimum-distance clusters are shown in the upper left; points in one cluster are shown in red, while the others are shown in gray. When the vertical axis is expanded by a factor of 2.0 and the horizontal axis shrunk by a factor of 0.5, the clustering is altered (as shown at the right). Alternatively, if the vertical axis is shrunk by a factor of 0.5 and the horizontal axis is expanded by a factor of 2.0, smaller more numerous clusters result (shown at the bottom). In both these scaled cases, the assignment of points to clusters differ from that in the original space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

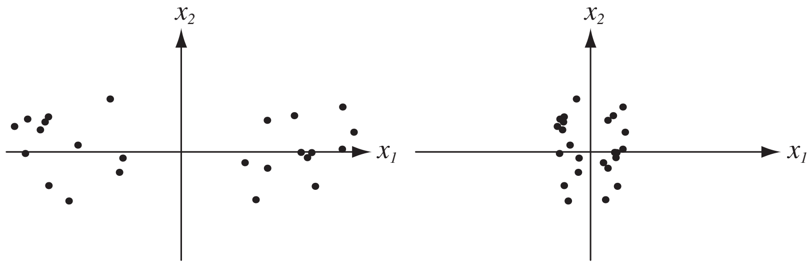

FIGURE 10.9. If the data fall into well-separated clusters (left), normalization by scaling for unit variance for the full data may reduce the separation, and hence be undesirable (right). Such a normalization may in fact be appropriate if the full data set arises from a single fundamental process (with noise), but inappropriate if there are several different processes, as shown here. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

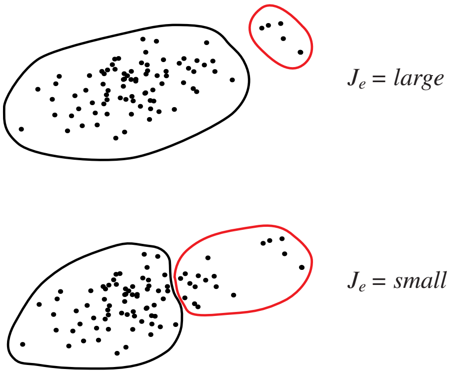

FIGURE 10.10. When two natural groupings have very different numbers of points, the clusters minimizing a sum-squared-error criterion J_e of Eq. 54 may not reveal the true underlying structure. Here the criterion is smaller for the two clusters at the bottom than for the more natural clustering at the top. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

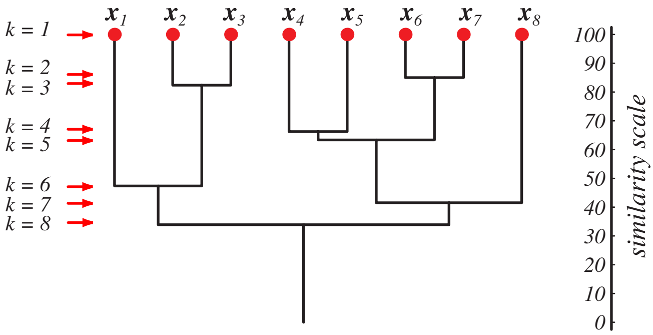

FIGURE 10.11. A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters. Here, at level 1 all eight points lie in singleton clusters; each point in a cluster is highly similar to itself, of course. Points $\mathbf{x}_6$ and $\mathbf{x}_7$ happen to be the most similar, and are merged at level 2, and so forth. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

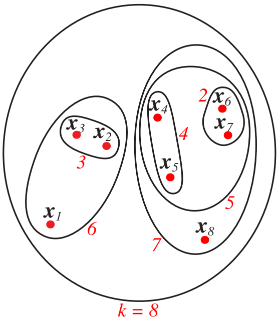

FIGURE 10.12. A set or Venn diagram representation of two-dimensional data (which was used in the dendrogram of Fig. 10.11) reveals the hierarchical structure but not the quantitative distances between clusters. The levels are numbered by k, in red. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

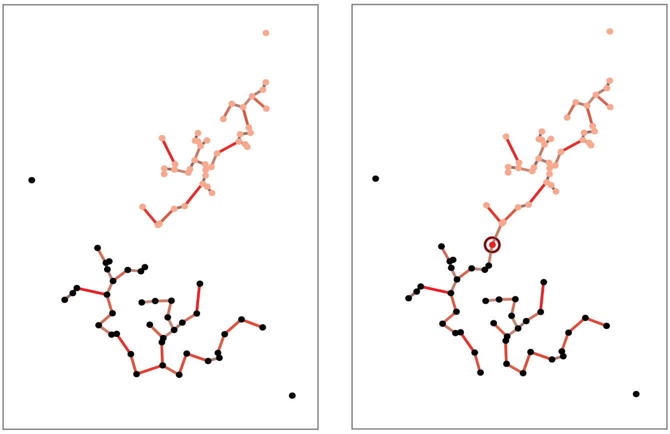

FIGURE 10.13. Two Gaussians were used to generate two-dimensional samples, shown in pink and black. The nearest-neighbor clustering algorithm gives two clusters that well approximate the generating Gaussians (left). If, however, another particular sample is generated (circled red point at the right) and the procedure is restarted, the clusters do not well approximate the Gaussians. This illustrates how the algorithm is sensitive to the details of the samples. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

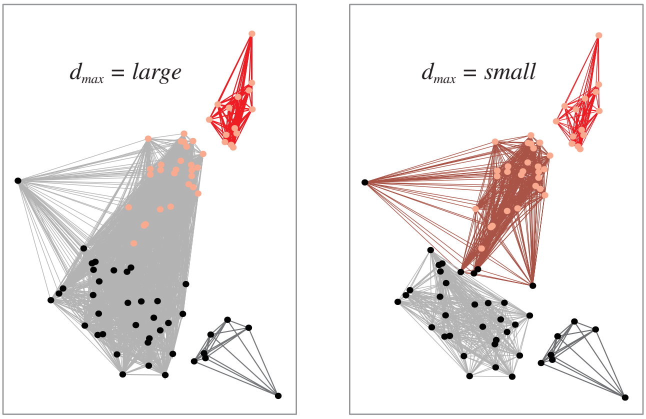

FIGURE 10.14. The farthest-neighbor clustering algorithm uses the separation between the most distant points as a criterion for cluster membership. If this distance is set very large, then all points lie in the same cluster. In the case shown at the left, a fairly large d_{max} leads to three clusters; a smaller d_{max} gives four clusters, as shown at the right. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

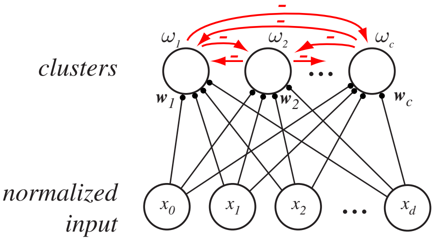

FIGURE 10.15. The two-layer network that implements the competitive learning algorithm consists of $d + 1$ input units and c output or cluster units. Each augmented input pattern is normalized to unit length (i.e., $\|\mathbf{x}\| = 1$), as is the set of weights at each cluster unit. When a pattern is presented, each of the cluster units computes its net activation $net_j = \mathbf{w}_j^t\mathbf{x}$; only the weights at the most active cluster unit are modified. (The suppression of activity in all but the most active cluster units can be implemented by competition among these units, as indicated by the red arrows.) The weights of the most active unit are then modified to be more similar to the pattern presented. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

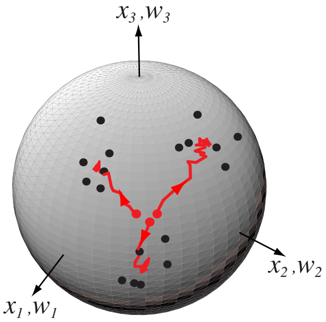

FIGURE 10.16. All of the two-dimensional patterns have been augmented and normalized and hence lie on a two-dimensional sphere in three dimensions. Likewise, the weights of the three cluster centers have been normalized. The red curves show the trajectory of the weight vectors, which start at the red points and end at the center of a cluster. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

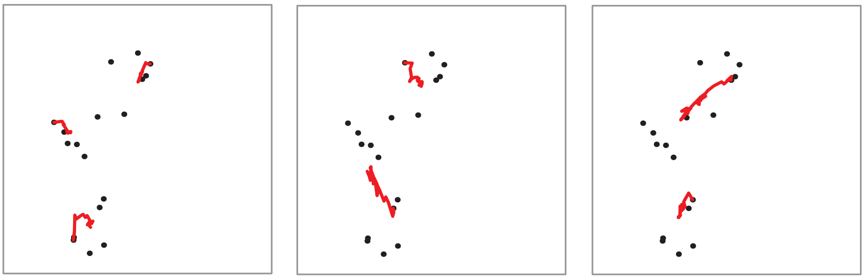

FIGURE 10.17. In leader-follower clustering, the number of clusters and their centers depend upon the random sequence of presentations of the points. The three simulations shown employed the same learning rate η, threshold θ, and number of presentations of each point (50), but differ in the random sequence of presentations. Notice that in the simulation on the left, three clusters are generated, whereas only two are generated in the other simulations. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

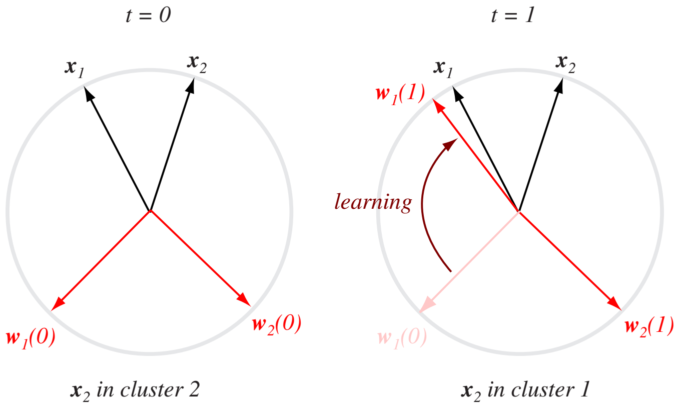

FIGURE 10.18. Instability and recoding can occur during competitive learning, as illustrated in this simple case of two patterns and two cluster centers. Two patterns, $\mathbf{x}_1$ and $\mathbf{x}_2$, are presented to a 2-2 network of Fig. 10.15 represented by two weight vectors. At $t = 0$, $\mathbf{w}_1$ happens to be most aligned with $\mathbf{x}_1$ and hence this pattern belongs to cluster 1; likewise, $\mathbf{x}_2$ is most aligned with $\mathbf{w}_2$ and hence it belongs to cluster 2, as shown at the left. Next, suppose pattern $\mathbf{x}_1$ is presented several times; through the competitive learning weight update rule, $\mathbf{w}_1$ moves to become closer to $\mathbf{x}_1$. Now $\mathbf{x}_2$ is most aligned with $\mathbf{w}_1$, and thus it has changed from class 2 to class 1. Surprisingly, this recoding of $\mathbf{x}_2$ occurs even though $\mathbf{x}_2$ was not used for weight update. It is theoretically possible that such recoding will occur numerous times in response to particular sequences of pattern presentations. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

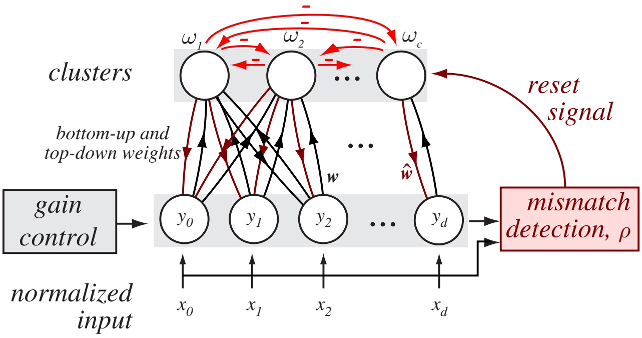

FIGURE 10.19. A generic adaptive resonance network has inputs and cluster units, much like a network for performing competitive learning. However, the input and the category layers are fully interconnected by *both* bottom-up and top-down connections with weights. The bottom-up weights, denoted $\mathbf{w}$, learn the cluster centers while the top-down weights, $\hat{\mathbf{w}}$, learn expected input patterns. If a match between input and a learned cluster is poor (where the quality of the match is specified by a user-specified vigilance parameter ρ), then the active cluster unit is suppressed by a reset signal, and a new cluster center can be recruited. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

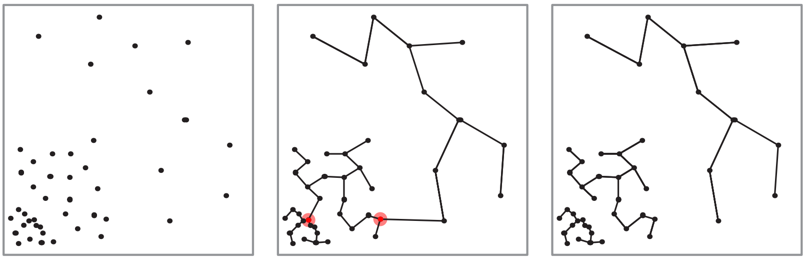

FIGURE 10.20. The removal of inconsistent edges—ones with length significantly larger than the average incident upon a node—may yield natural clusters. The original data are shown at the left, and its minimal spanning tree is shown in the middle. At virtually every node, incident edges are of nearly the same length. Each of the two nodes shown in red are exceptions: their incident edges are of very different lengths. When the two such inconsistent edges are removed, three clusters are produced, as shown at the right. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

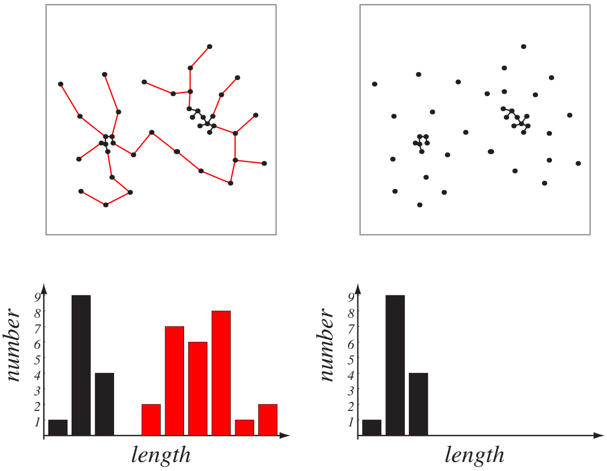

FIGURE 10.21. A minimal spanning tree is shown at the left; its bimodal edge length distribution is evident in the histogram below. If all links of intermediate or high length are removed (red), the two natural clusters are revealed (right). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

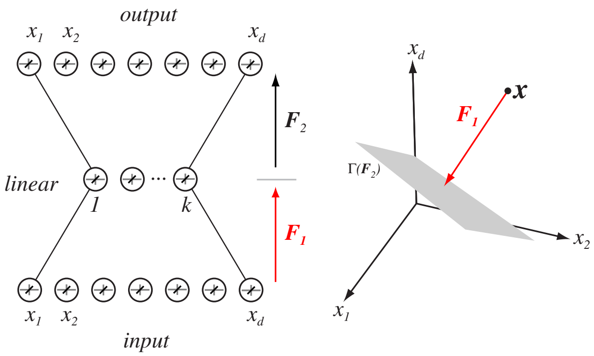

FIGURE 10.22. A three-layer neural network with linear hidden units, trained to be an auto-encoder, develops an internal representation that corresponds to the principal components of the full data set. The transformation $\mathbf{F}_1$ is a linear projection onto a k-dimensional subspace denoted $\Gamma(\mathbf{F}_2)$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

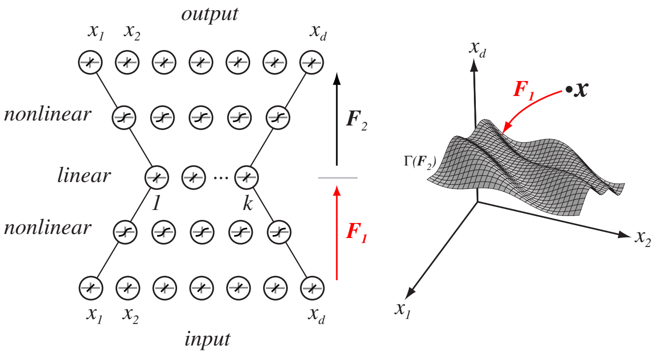

FIGURE 10.23. A five-layer neural network with two layers of nonlinear units (e.g., sigmoidal), trained to be an auto-encoder, develops an internal representation that corresponds to the nonlinear components of the full data set. The process can be viewed in feature space (at the right). The transformation $\mathbf{F}_1$ is a nonlinear projection onto a k-dimensional subspace, denoted $\Gamma(\mathbf{F}_2)$. Points in $\Gamma(\mathbf{F}_2)$ are mapped via $\mathbf{F}_2$ back to the the d-dimensional space of the original data. After training, the top two layers of the net are removed and the remaining three-layer network maps inputs $\mathbf{x}$ to the space $\Gamma(\mathbf{F}_2)$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

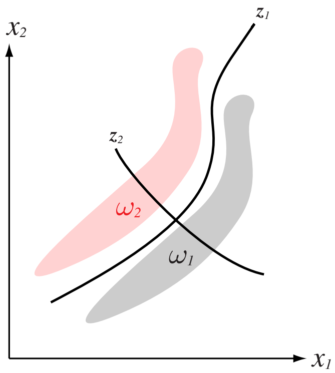

FIGURE 10.24. Features from two classes are as shown, along with nonlinear components of the full data set. Apparently, these classes are well-separated along the line marked z_2, but the large noise gives the largest nonlinear component to be along z_1. Preprocessing by keeping merely the largest nonlinear component would retain the "noise" and discard the "signal," giving poor recognition. The same defect can arise in linear principal components, where the coordinates are linear and orthogonal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

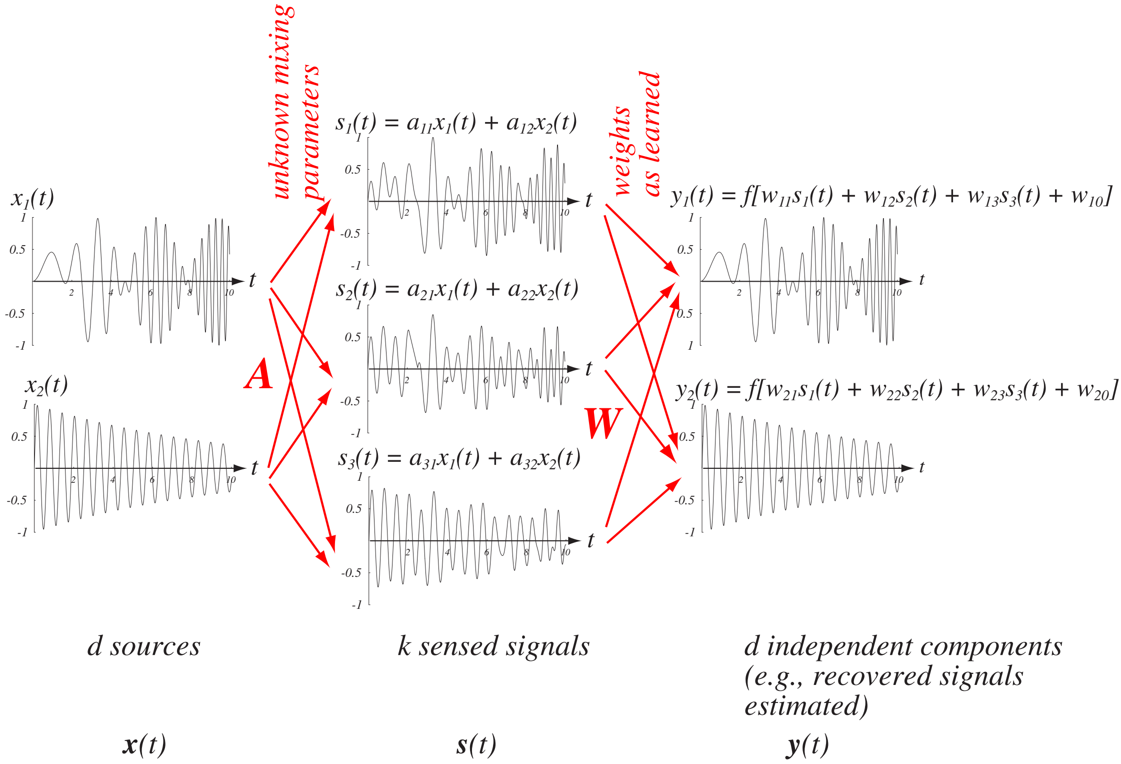
unknown mixing parameters
A
weights as learned
W
$x_1(t)$
$x_2(t)$
$s_1(t) = a_{11}x_1(t) + a_{12}x_2(t)$
$s_2(t) = a_{21}x_1(t) + a_{22}x_2(t)$
$s_3(t) = a_{31}x_1(t) + a_{32}x_2(t)$
$y_1(t) = f[w_{11}s_1(t) + w_{12}s_2(t) + w_{13}s_3(t) + w_{10}]$
$y_2(t) = f[w_{21}s_1(t) + w_{22}s_2(t) + w_{23}s_3(t) + w_{20}]$
t
1
0.5
-0.5
-1
2
4
6
8
10
d sources
k sensed signals
d independent components (e.g., recovered signals estimated)
$\boldsymbol{x}(t)$
$\boldsymbol{s}(t)$
$\boldsymbol{y}(t)$

FIGURE 10.25. Independent component analysis (ICA) is an unsupervised method that can be applied to the problem of blind source separation. In such problems, two or more source signals (assumed independent) $x_1(t), x_2(t), \cdots, x_d(t)$ are mixed linearly to yield sum signals $s_1(t), s_2(t), \cdots, s_k(t)$, where $k \geq d$. (This figure illustrates the case $d = 2$ and $k = 3$.) Given merely the sensed signals $\mathbf{x}(t)$ and an assumed number of components, d, the task of ICA is to find independent components in $\mathbf{s}$. In a blind source separation application, these are merely the source signals. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

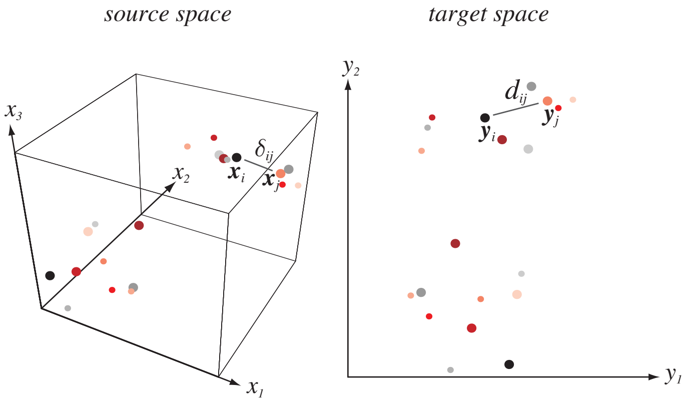

FIGURE 10.26. The figure shows an example of points in a three-dimensional space being mapped to a two-dimensional space. The size and color of each point $\mathbf{x}_i$ matches that of its image, $\mathbf{y}_i$. Here we use simple Euclidean distance, that is, $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ and $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$. In typical applications, the source space usually has high dimensionality, but to allow easy visualization the target space is only two- or three-dimensional. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

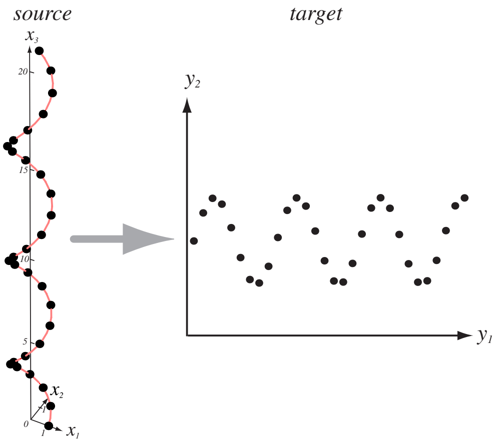

FIGURE 10.27. Thirty points of the form $\mathbf{x} = (\cos(k/\sqrt{2}), \sin(k/\sqrt{2}), k/\sqrt{2})^t$ for $k = 0, 1, \ldots, 29$ are shown at the left. Multidimensional scaling using the J_{ef} criterion (Eq. 109) and a two-dimensional target space leads to the image points shown at the right. This lower-dimensional representation shows clearly the fundamental sequential nature of the points in the original source space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

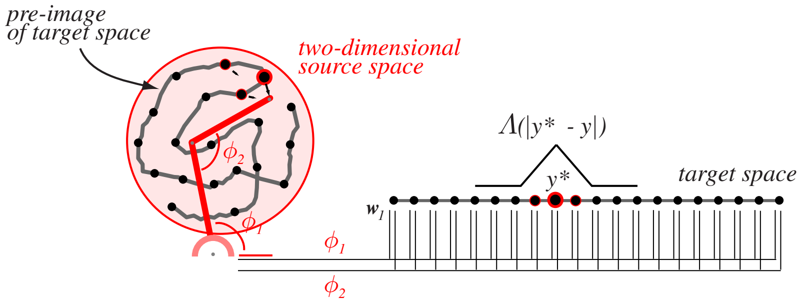

FIGURE 10.28. A self-organizing map from the (two-dimensional) disk source space to the (one-dimensional) line of the target space can be learned as follows. For each point y in the target line, there exists a corresponding point in the source space that, if sensed, would lead to y being most active. For clarity, then, we can link theses points in the source; it is as if the image line is placed in the source space. We call this the pre-image of the target space. At the state shown, the particular sensed point leads to y^* begin most active. The learning rule (Eq. 113) makes its source point move toward the sensed point, as shown by the small arrow. Because of the window function $\Lambda(|y^* - y|)$, the pre-image of points adjacent to y^* are also moved toward the sensed point, thought not as much. If such learning is repeated many times as the arm randomly senses the whole source space, a topologically correct map is learned. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

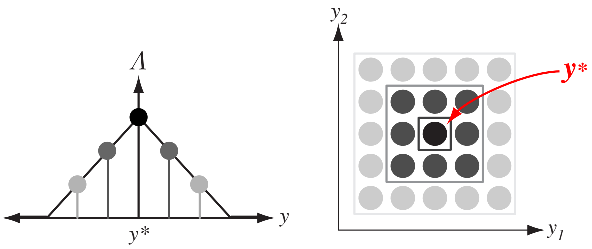

FIGURE 10.29. Typical window functions for self-organizing maps for target spaces in one dimension (left) and two dimensions (right). In each case, the weights at the maximally active unit, $\mathbf{y}^*$, in the target space get the largest weight update while units more distant get smaller update. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

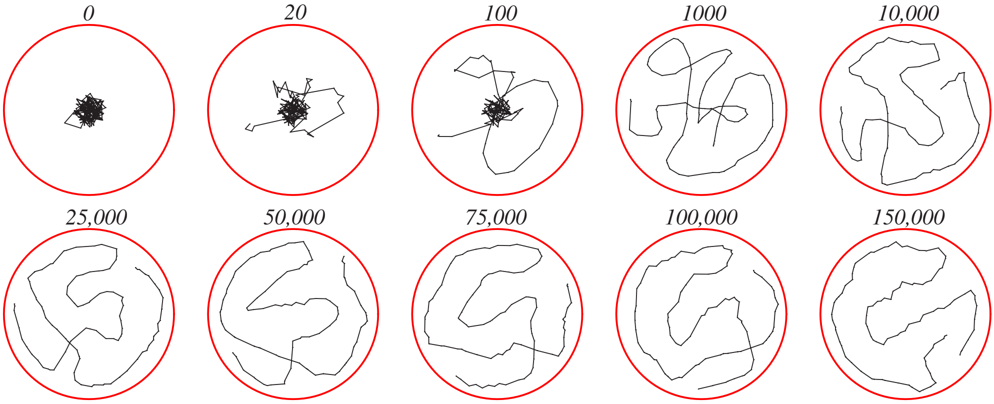

FIGURE 10.30. If a large number of pattern presentations are made using the setup of Fig. 10.28, a topologically ordered map develops. The number of pattern presentations is listed. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

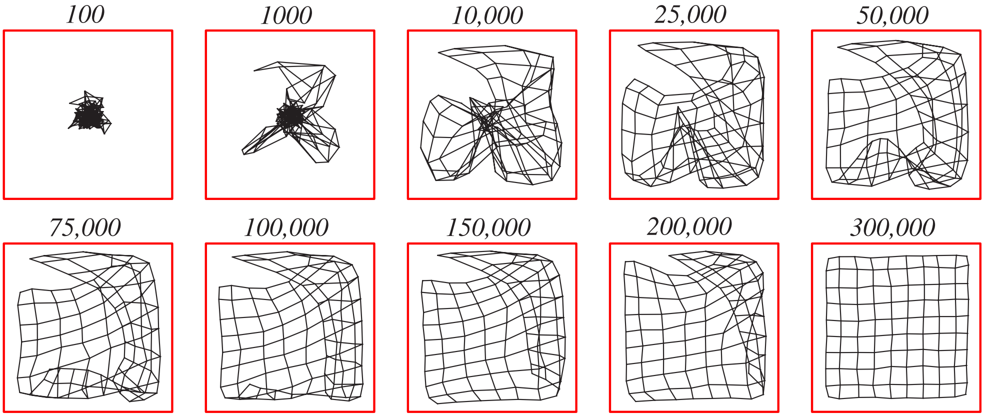

FIGURE 10.31. A self-organizing feature map from a square source space to a square (grid) target space. As in Fig. 10.28, each grid point of the target space is shown atop the point in the source space that leads maximally excites that target point. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

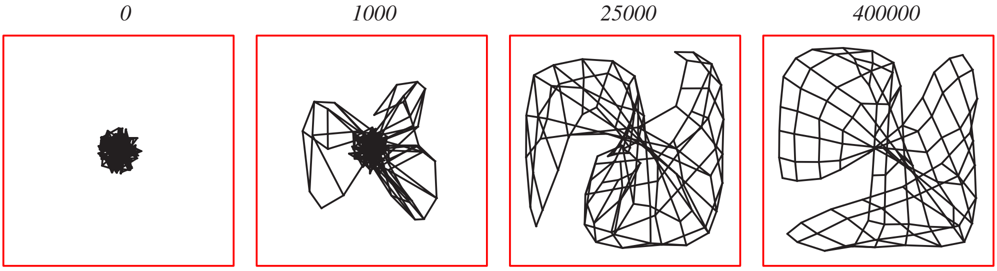

FIGURE 10.32. Some initial (random) weights and the particular sequence of patterns (randomly chosen) lead to kinks in the map; even extensive further training does not eliminate the kink. In such cases, learning should be restarted with randomized weights and possibly a wider window function and slower decay in learning. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

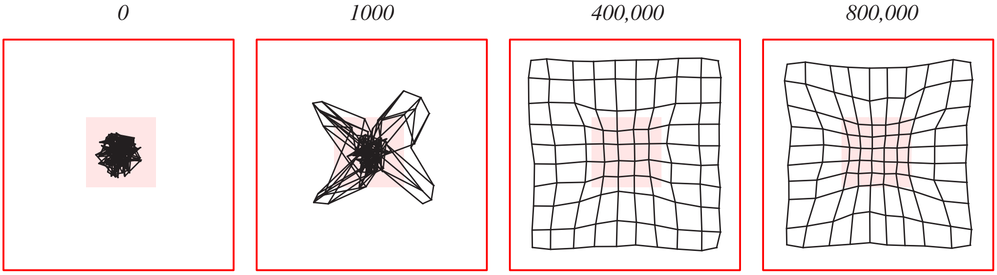

FIGURE 10.33. As in Fig. 10.31 except that the sampling of the input space was not uniform. In particular, the probability density for sampling a point in the central square region (pink) was 20 times greater than elsewhere. Notice that the final map devotes more nodes to this center region than in Fig. 10.31. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.