$$S_L(z_1, z_2, \ldots, z_N) = \sum_{(i,i')\in\mathcal{N}} (d_{ii'} - ||z_i - z_{i'}||)^2 - \tau \sum_{(i,i')\notin\mathcal{N}} ||z_i - z_{i'}||, \tag{14.106}$$

where $\tau = 2wD$. The first term in (14.106) tries to preserve local structure in the data, while the second term encourages the representations $z_i, z_{i'}$ for pairs (i, i') that are non-neighbors to be farther apart. Local MDS minimizes the stress function (14.106) over z_i, for fixed values of the number of neighbors K and the tuning parameter τ.

The right panel of Figure 14.44 shows the result of local MDS, using $k = 2$ neighbors and $\tau = 0.01$. We used coordinate descent with multiple starting values to find a good minimum of the (nonconvex) stress function (14.106). The ordering of the points along the curve has been largely preserved,

Figure 14.45 shows a more interesting application of one of these methods (LLE)[15]. The data consist of 1965 photographs, digitized as 20×28 grayscale images. The result of the first two-coordinates of LLE are shown and reveal some variability in pose and expression. Similar pictures were produced by local MDS.

In experiments reported in Chen and Buja (2008), local MDS shows superior performance, as compared to ISOMAP and LLE. They also demonstrate the usefulness of local MDS for graph layout. There are also close connections between the methods discussed here, spectral clustering (Section 14.5.3) and kernel PCA (Section 14.5.4).

14.10 The Google PageRank Algorithm

In this section we give a brief description of the original *PageRank* algorithm used by the Google search engine, an interesting recent application of unsupervised learning methods.

We suppose that we have N web pages and wish to rank them in terms of importance. For example, the N pages might all contain a string match to "statistical learning" and we might wish to rank the pages in terms of their likely relevance to a websurfer.

The *PageRank* algorithm considers a webpage to be important if many other webpages point to it. However the linking webpages that point to a given page are not treated equally: the algorithm also takes into account both the importance (*PageRank*) of the linking pages and the number of outgoing links that they have. Linking pages with higher *PageRank* are given more weight, while pages with more outgoing links are given less weight. These ideas lead to a recursive definition for *PageRank*, detailed next.

[15] Sam Roweis and Lawrence Saul kindly provided this figure.

Let $L_{ij} = 1$ if page j points to page i, and zero otherwise. Let $c_j = \sum_{i=1}^{N} L_{ij}$ equal the number of pages pointed to by page j (number of outlinks). Then the Google *PageRanks* p_i are defined by the recursive relationship

$$p_i = (1-d) + d\sum_{j=1}^{N}\big(\frac{L_{ij}}{c_j}\big)p_j \tag{14.107}$$

where d is a positive constant (apparently set to 0.85).

The idea is that the importance of page i is the sum of the importances of pages that point to that page. The sums are weighted by $1/c_j$, that is, each page distributes a total vote of 1 to other pages. The constant d ensures that each page gets a *PageRank* of at least $1-d$. In matrix notation

$$\mathbf{p} = (1-d)\mathbf{e} + d\cdot\mathbf{L}\mathbf{D}_c^{-1}\mathbf{p} \tag{14.108}$$

where $\mathbf{e}$ is a vector of N ones and $\mathbf{D}_c = \mathrm{diag}(\mathbf{c})$ is a diagonal matrix with diagonal elements c_j. Introducing the normalization $\mathbf{e}^T\mathbf{p} = N$ (i.e., the average *PageRank* is 1), we can write (14.108) as

$$\begin{aligned}\mathbf{p} &= \big[(1-d)\mathbf{e}\mathbf{e}^T/N + d\mathbf{L}\mathbf{D}_c^{-1}\big]\mathbf{p}\\ &= \mathbf{A}\mathbf{p}\end{aligned} \tag{14.109}$$

where the matrix $\mathbf{A}$ is the expression in square braces.

Exploiting a connection with Markov chains (see below), it can be shown that the matrix $\mathbf{A}$ has a real eigenvalue equal to one, and one is its largest eigenvalue. This means that we can find $\hat{\mathbf{p}}$ by the power method: starting with some $\mathbf{p} = \mathbf{p}_0$ we iterate

$$\mathbf{p}_k \leftarrow \mathbf{A}\mathbf{p}_{k-1}; \quad \mathbf{p}_k \leftarrow N\frac{\mathbf{p}_k}{\mathbf{e}^T\mathbf{p}_k}. \tag{14.110}$$

The fixed points $\hat{\mathbf{p}}$ are the desired *PageRanks*.

In the original paper of Page et al. (1998), the authors considered *PageRank* as a model of user behavior, where a random web surfer clicks on links at random, without regard to content. The surfer does a random walk on the web, choosing among available outgoing links at random. The factor $1-d$ is the probability that he does not click on a link, but jumps instead to a random webpage.

Some descriptions of *PageRank* have $(1-d)/N$ as the first term in definition (14.107), which would better coincide with the random surfer interpretation. Then the page rank solution (divided by N) is the stationary distribution of an irreducible, aperiodic Markov chain over the N webpages.

Definition (14.107) also corresponds to an irreducible, aperiodic Markov chain, with different transition probabilities than those from he $(1-d)/N$ version. Viewing *PageRank* as a Markov chain makes clear why the matrix $\mathbf{A}$ has a maximal real eigenvalue of 1. Since $\mathbf{A}$ has positive entries with

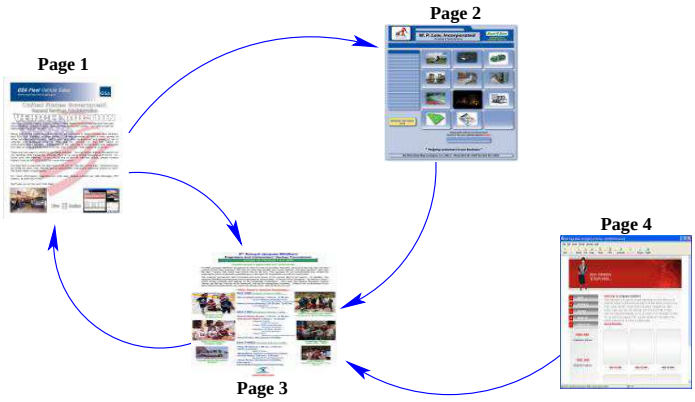

FIGURE 14.46. *PageRank algorithm: example of a small network*

each column summing to one, Markov chain theory tells us that it has a unique eigenvector with eigenvalue one, corresponding to the stationary distribution of the chain (Bremaud, 1999).

A small network is shown for illustration in Figure 14.46. The link matrix is

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{14.111}$$

and the number of outlinks is $\mathbf{c} = (2, 1, 1, 1)$.

The *PageRank* solution is $\hat{\mathbf{p}} = (1.49, 0.78, 1.58, 0.15)$. Notice that page 4 has no incoming links, and hence gets the minimum *PageRank* of 0.15.

Bibliographic Notes

There are many books on clustering, including Hartigan (1975), Gordon (1999) and Kaufman and Rousseeuw (1990). K-means clustering goes back at least to Lloyd (1957), Forgy (1965), Jancey (1966) and MacQueen (1967). Applications in engineering, especially in image compression via vector quantization, can be found in Gersho and Gray (1992). The k-medoid procedure is described in Kaufman and Rousseeuw (1990). Association rules are outlined in Agrawal et al. (1995). The self-organizing map was proposed by Kohonen (1989) and Kohonen (1990); Kohonen et al. (2000) give a more recent account. Principal components analysis and multidimensional scaling are described in standard books on multivariate analysis, for example, Mardia et al. (1979). Buja et al. (2008) have implemented a powerful environment called Ggvis for multidimensional scaling, and the user manual