(3) Most information retrieval systems change with time, when new documents are added. In LSI this necessitates the updating of the SVD of the term-document matrix. Unfortunately, it is quite expensive to update an SVD. The Lanczos-based method, on the other hand, adapts immediately and at no extra cost to changes of A.

5. Classification and pattern recognition

5.1. Classification of handwritten digits using SVD bases

Computer classification of handwritten digits is a standard problem in pattern recognition. The typical application is automatic reading of zip codes on envelopes. A comprehensive review of different algorithms is given in LeCun, Bottou, Bengio and Haffner (1998).

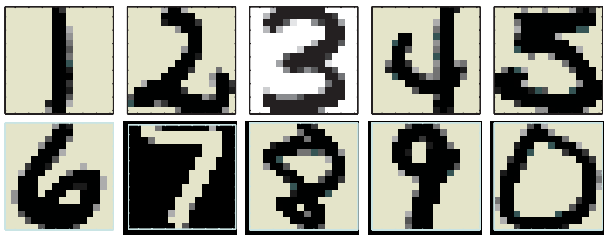

Figure 5.1. Handwritten digits from the US Postal Service database.

In Figure 5.1 we illustrate handwritten digits that we will use in the examples in this section.

We will treat the digits in three different, but equivalent ways:

(1) 16×16 grey-scale images,

(2) functions of two variables,

(3) vectors in $\mathbb{R}^{256}$.

In the classification of an unknown digit it is necessary to compute the distance to known digits. Different distance measures can be used, perhaps the most natural is Euclidean distance: stack the columns of the image in a vector and identify each digit as a vector in $\mathbb{R}^{256}$. Then define the distance function

$$\text{dist}(x, y) = \|x - y\|_2.$$

An alternative distance function can be based on the cosine between two vectors.

In a real application of recognition of handwritten digits, *e.g.*, zip code reading, there are hardware and real time factors that must be taken into account. In this section we will describe an idealized setting. The problem is:

Given a set of of manually classified digits (the training set), classify a set of unknown digits (the test set).

In the US Postal Service database, the training set contains 7291 handwritten digits, and the test set has 2007 digits.

When we consider the training set digits as vectors or points, then it is reasonable to assume that all digits of one kind form a cluster of points in a Euclidean 256-dimensional vector space. Ideally the clusters are well separated and the separation depends on how well written the training digits are.

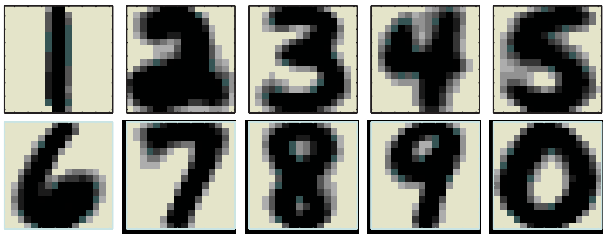

Figure 5.2. The means (centroids) of all digits in the training set.

In Figure 5.2 we illustrate the means (centroids) of the digits in the training set. From this figure we get the impression that a majority of the digits are well written (if there were many badly written digits this would demonstrate itself as diffuse means). This means that the clusters are rather well separated. Therefore it is likely that a simple algorithm that computes the distance from each unknown digit to the means should work rather well.

A simple classification algorithm

Training. Given the training set, compute the mean (centroid) of all digits of one kind.

Classification. For each digit in the test set, compute the distance to all ten means, and classify as the closest.

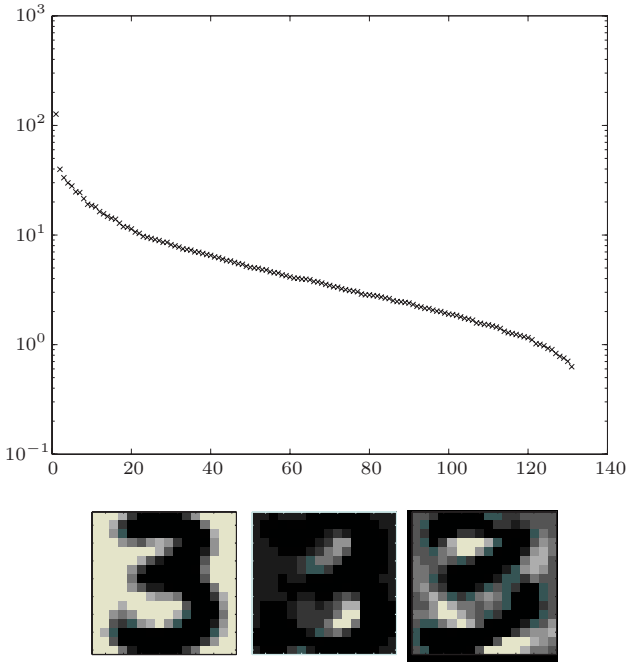

Figure 5.3. Singular values (top), and the first three singular images (vectors) computed using the 131 3s of the training set (bottom).

It turns out that for our test set the success rate of this algorithm is around 75%, which is not good enough. The reason is that the algorithm does not use any information about the variation of the digits of one kind. This variation can be modelled using the SVD.

Let $A \in \mathbb{R}^{m\times n}$, with $m = 256$, be the matrix consisting of all the training digits of one kind, the 3s, say. The columns of A span a linear subspace of $\mathbb{R}^m$. However, this subspace cannot be expected to have a large dimension, because if it had, then the subspaces of the different kinds of digits would intersect.

The idea now is to 'model' the variation within the set of training digits of one kind using an orthogonal basis of the subspace. An orthogonal basis can be computed using the SVD, and A can be approximated by a sum of rank-one matrices (3.9),

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

for some value of k. Each column in A is an image of a digit 3, and therefore the left singular vectors u_i are an orthogonal basis in the 'image space of 3s'. We will refer to the left singular vectors as 'singular images'. From the matrix approximation properties of the SVD (Theorem 3.4) we know that the first singular vector represents the 'dominating' direction of the data matrix. Therefore, if we fold the vectors u_i back to images, we expect the first singular vector to look like a 3, and the following singular images should represent the dominating variations of the training set around the first singular image. In Figure 5.3 we illustrate the singular values and the first three singular images for the training set 3s.

The SVD basis classification algorithm will be based on the following assumptions.

(1) Each digit (in the training and test sets) is well characterized by a few of the first singular images of its own kind. The more precise meaning of 'a few' should be investigated by experiment.

(2) An expansion in terms of the first few singular images discriminates well between the different classes of digits.

(3) If an unknown digit can be better approximated in one particular basis of singular images, the basis of 3s say, than in the bases of the other classes, then it is likely that the unknown digit is a 3.

Thus we should compute how well an unknown digit can be represented in the ten different bases. This can be done by computing the residual vector in *least squares problems* of the type

$$\min_{\alpha_i} \left\| z - \sum_{i=1}^{k} \alpha_i u_i \right\|,$$

where z represents an unknown digit, and u_i the singular images. We can write this problem in the form

$$\min_{\alpha} \|z - U_k \alpha\|_2,$$

where $U_k = \begin{pmatrix} u_1 & u_2 & \cdots & u_k \end{pmatrix}$. Since the columns of U_k are orthogonal, the solution of this problem is given by $\alpha = U_k^T z$, and the norm of the residual vector of the least squares problems is

$$\|(I - U_k U_k^T) z\|_2. \tag{5.1}$$

It is interesting to see how the residual depends on the number of terms in the basis. In Figure 5.4 we illustrate the approximation of a nicely written 3 in terms of the 3-basis with different numbers of basis images. In Figure 5.5 we show the approximation of a nice 3 in the 5-basis.

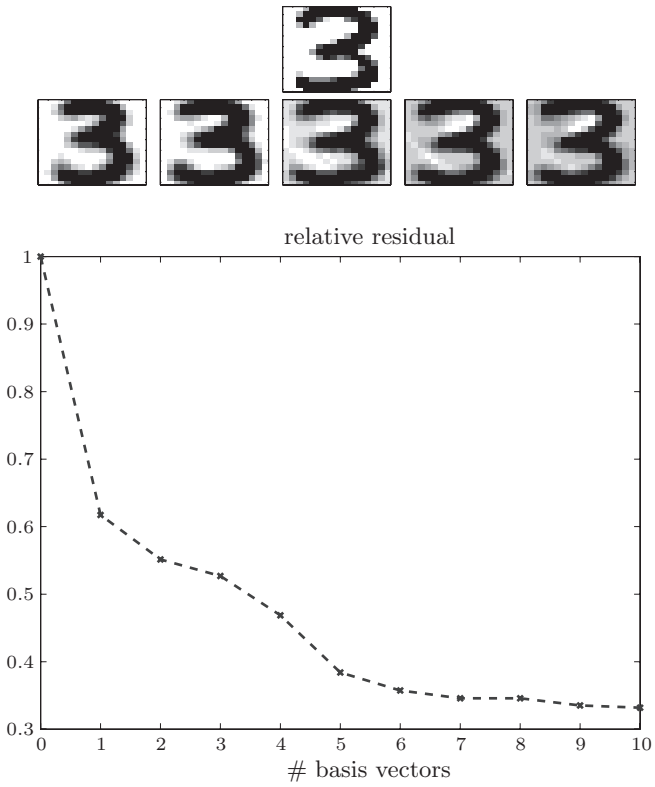

Figure 5.4. Unknown digit (nice 3) and approximations using 1, 3, 5, 7, and 9 terms in the 3-basis (top). Relative residual $\|(I - U_k U_k^T)z\|_2/\|z\|_2$ in least squares problem (bottom).

From Figures 5.4 and 5.5 we see that the relative residual is considerably smaller for the nice 3 in the 3-basis than in the 5-basis.

It is possible to devise several classification algorithm based on the model of expanding in terms of SVD bases. Below we give a simple variant.

An SVD basis classification algorithm

Training. For the training set of known digits, compute the SVD of each class of digits, and use k basis vectors for each class.

Classification. For a given test digit, compute its relative residual in all ten bases. If one residual is significantly smaller than all the others, classify as that. Otherwise give up.

The algorithm is closely related to the SIMCA method (Wold 1976, Sjöström and Wold 1980).