Chapter 14

Face Recognition Using Tensor SVD

Human beings are very skillful at recognizing faces even when the facial expression, the illumination, the viewing angle, etc., vary. To develop automatic procedures for face recognition that are robust with respect to varying conditions is a challenging research problem that has been investigated using several different approaches. Principal component analysis (i.e., SVD) is a popular technique that often goes by the name "eigenfaces" [23, 88, 100]. However, this method is best when all pictures are taken under similar conditions, and it does not perform well when several environment factors are varied. More general bilinear models also have been investigated; see, e.g., [95].

Recently [102, 103, 104, 105], methods for multilinear analysis of image ensembles were studied. In particular, the face recognition problem was considered using a tensor model, the TensorFaces approach. By letting the modes of the tensor represent a different viewing condition, e.g., illumination or facial expression, it became possible to improve the precision of the recognition algorithm compared to the PCA method.

In this chapter we will describe a tensor method for face recognition, related to TensorFaces. Since we are dealing with images, which are often stored as $m \times n$ arrays, with m and n of the order 100–500, the computations for each face to be identified are quite heavy. We will discuss how the tensor SVD (HOSVD) can also be used for dimensionality reduction to reduce the flop count.

14.1 Tensor Representation

Assume that we have a collection of images of n_p persons, where each image is an $m_{i_1} \times m_{i_2}$ array with $m_{i_1} m_{i_2} = n_i$. We will assume that the columns of the images are stacked so that each image is represented by a vector in $\mathbb{R}^{n_i}$. Further assume that each person has been photographed with n_e different facial expressions.[36] Often one can have $n_i \geq 5000$, and usually n_i is considerably larger than n_e and n_p. The

[36] For simplicity here we refer to different illuminations, etc., as expressions.

collection of images is stored as a tensor,

$$\mathcal{A} \in \mathbb{R}^{n_i \times n_e \times n_p}. \tag{14.1}$$

We refer to the different modes as the image mode, the expression mode, and the person mode, respectively.

If, for instance we also had photos of each person with different illumination, viewing angles, etc., then we could represent the image collection by a tensor of higher degree [104]. For simplicity, here we consider only the case of a 3-mode tensor. The generalization to higher order tensors is straightforward.

Example 14.1. We preprocessed images of 10 persons from the Yale Face Database by cropping and decimating each image to 112×78 pixels stored in a vector of length 8736. Five images are illustrated in Figure 14.1.

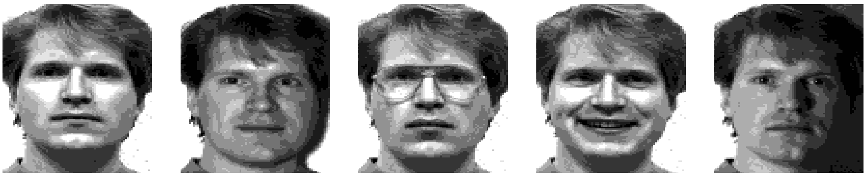


Figure 14.1. *Person* 1 *with five different expressions* (*from the Yale Face Database*).

Each person is photographed with a total of 11 different expressions. ■

The ordering of the modes is arbitrary, of course; for definiteness and for illustration purposes we will assume the ordering of (14.1). However, to (somewhat) emphasize the ordering arbitrariness, will use the notation $\times_e$ for multiplication of the tensor by matrix along the expression mode, and similarly for the other modes. We now assume that $n_i \gg n_e n_p$ and write the *thin HOSVD* (see Theorem 8.3 and (8.9)),

$$\mathcal{A} = \mathcal{S} \times_i F \times_e G \times_p H, \tag{14.2}$$

where $\mathcal{S} \in \mathbb{R}^{n_e n_p \times\, n_e \times\, n_p}$ is the core tensor, $F \in \mathbb{R}^{n_i \times\, n_e n_p}$ has orthonormal columns, and $G \in \mathbb{R}^{n_e \times n_e}$ and $H \in \mathbb{R}^{n_p \times n_p}$ are orthogonal.

Example 14.2. We computed the HOSVD of the tensor of face images of 10 persons, each with 10 different expressions. The singular values are plotted in Figure 14.2. All 10 singular values in the expression and person modes are significant, which means that it should be relatively easy to distinguish between expressions and persons. ■

The HOSVD can be interpreted in different ways depending on what it is to be used for. We first illustrate the relation

$$\mathcal{A} = \mathcal{D} \times_e G \times_p H,$$

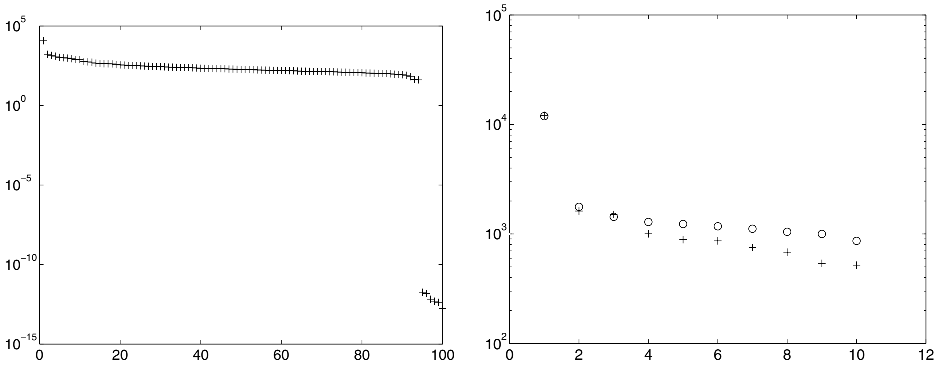
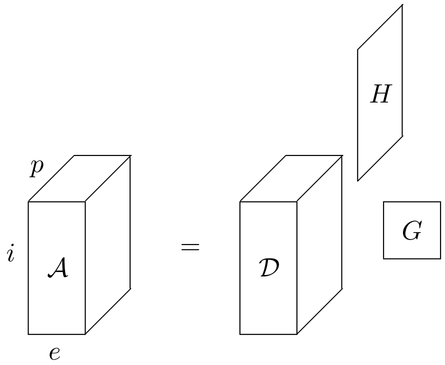


Figure 14.2. *The singular values in the image mode (left), the expression mode (right, +), and the person mode (right, circles).*

where $\mathcal{D} = \mathcal{S} \times_i F$:



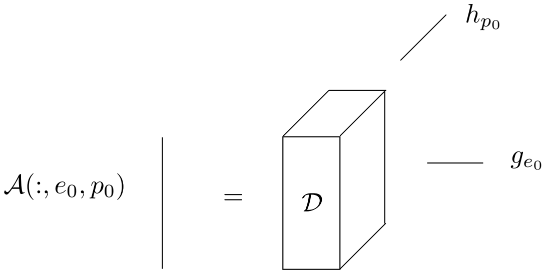
At this point, let us recapitulate the definition of tensor-matrix multiplication (Section 8.2). For definiteness we consider 2-mode, i.e., here e-mode, multiplication:

$$(\mathcal{D} \times_e G)(i_1, j, i_3) = \sum_{k=1}^{n_e} g_{j,k}\, d_{i_1,k,i_3}.$$

We see that fixing a particular value of the expression parameter, i.e., putting $j = e_0$, say, corresponds to using only the e_0th row of G. By doing the analogous choice in the person mode, we get

$$\mathcal{A}(:, e_0, p_0) = \mathcal{D} \times_e g_{e_0} \times_p h_{p_0}, \tag{14.3}$$

where g_{e_0} denotes the e_0th row vector of G and h_{p_0} the p_0th row vector of H. We illustrate (14.3) in the following figure:



We summarize this in words:

> The image of person p_0 in expression e_0 can be synthesized by multiplication of the tensor $\mathcal{D}$ by h_{p_0} and g_{e_0} in their respective modes. Thus person p_0 is uniquely characterized by the row vector h_{p_0} and expression e_0 is uniquely characterized by g_{e_0}, via the bilinear form
>
> $$\mathcal{D} \times_e g \times_p h.$$

Example 14.3. The MATLAB code

```
a=tmul(tmul(D,Ue(4,:),2),Up(6,:),3);
```

gives person 6 in expression 4 (happy); see Figure 14.3. Recall that the function `tmul(A,X,i)` multiplies the tensor `A` by the matrix `X` in mode `i`. ∎

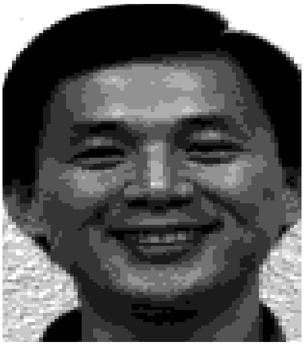


Figure 14.3. *Person* 6 *in expression* 4 (*happy*).

14.2 Face Recognition

We will now consider the classification problem as follows:

> Given an image of an unknown person, represented by a vector in $\mathbb{R}^{n_i}$, determine which of the n_p persons it represents, or decide that the unknown person is not in the database.

For the classification we write the HOSVD (14.2) in the following form:

$$\mathcal{A} = \mathcal{C} \times_p H, \qquad \mathcal{C} = \mathcal{S} \times_i F \times_e G. \tag{14.4}$$

For a particular expression e we have

$$\mathcal{A}(:, e, :) = \mathcal{C}(:, e, :) \times_p H. \tag{14.5}$$

Obviously we can identify the tensors $\mathcal{A}(:, e, :)$ and $\mathcal{C}(:, e, :)$ with matrices, which we denote A_e and C_e. Therefore, for all the expressions, we have linear relations

$$A_e = C_e H^T, \qquad e = 1, 2, \dots, n_e. \tag{14.6}$$

Note that the same (orthogonal) matrix H occurs in all n_e relations. With $H^T = \begin{pmatrix} h_1 & \dots & h_{n_p} \end{pmatrix}$, column p of (14.6) can be written

$$a_p^{(e)} = C_e h_p. \tag{14.7}$$

We can interpret (14.6) and (14.7) as follows:

> Column p of A_e contains the image of person p in expression e. The columns of C_e are *basis vectors for expression* e, and row p of H, i.e., h_p, holds the *coordinates* of the image of person p in this basis. Furthermore, *the same h_p holds the coordinates of the images of person p in all expression bases.*

Next assume that $z \in \mathbb{R}^{n_i}$ is an image of an unknown person in an unknown expression (out of the n_e) and that we want to classify it. We refer to z as a test image. Obviously, if it is an image of person p in expression e, then the coordinates of z in that basis are equal to h_p. Thus we can classify z by computing its coordinates in all the expression bases and checking, for each expression, whether the coordinates of z coincide (or almost coincide) with the elements of any row of H.

The coordinates of z in expression basis e can be found by solving the least squares problem

$$\min_{\alpha_e} \| C_e \alpha_e - z \|_2. \tag{14.8}$$

The algorithm is summarized below:

Classification algorithm (preliminary version)

```
% $z$ is a test image.
for $e = 1, 2, \dots, n_e$
    Solve $\min_{\alpha_e} \| C_e \alpha_e - z \|_2$.
    for $p = 1, 2, \dots, n_p$
        If $\| \alpha_e - h_p \|_2 <$ tol, then classify as person $p$ and stop.
    end
end
```

The amount of work in this algorithm is high: for each test image z we must solve n_e least squares problems (14.8) with $C_e \in \mathbb{R}^{n_i \times n_p}$.

However, recall from (14.4) that $\mathcal{C} = \mathcal{S} \times_i F \times_e G$, which implies

$$C_e = F B_e,$$

where $B_e \in \mathbb{R}^{n_e n_p \times n_p}$ is the matrix identified with $(\mathcal{S} \times_e G)(:, e, :)$. Note that $F \in \mathbb{R}^{n_i \times n_e n_p}$; we assume that n_i is considerably larger than $n_e n_p$. Then, for the analysis only, enlarge the matrix so that it becomes square and orthogonal:

$$\hat{F} = \begin{pmatrix} F & F^{\perp} \end{pmatrix}, \qquad \hat{F}^T \hat{F} = I.$$

Now insert $\hat{F}^T$ inside the norm:

$$\begin{aligned} \| C_e \alpha_e - z \|_2^2 = \| \hat{F}^T (F B_e \alpha_e - z) \|_2 &= \left\| \begin{pmatrix} B_e \alpha_e - F^T z \\ -(F^{\perp})^T z \end{pmatrix} \right\|_2^2 \\ &= \| B_e \alpha_e - F^T z \|_2^2 + \| (F^{\perp})^T z \|_2^2. \end{aligned}$$

It follows that we can solve the n_e least squares problems by first computing $F^T z$ and then solving

$$\min_{\alpha_e} \| B_e \alpha_e - F^T z \|_2, \qquad e = 1, 2, \ldots, n_e. \tag{14.9}$$

The matrix B_e has dimension $n_e n_p \times n_p$, so it is much cheaper to solve (14.9) than (14.8). It is also possible to precompute a QR decomposition of each matrix B_e to further reduce the work. Thus we arrive at the following algorithm.

Classification algorithm

Preprocessing step. Compute and save the thin QR decompositions of all the B_e matrices, $B_e = Q_e R_e$, $e = 1, 2, \ldots, n_e$.

```
% $z$ is a test image.

Compute $\hat{z} = F^T z$.

for $e = 1, 2, \ldots, n_e$
    Solve $R_e \alpha_e = Q_e^T \hat{z}$ for $\alpha_e$.
    for $p = 1, 2, \ldots, n_p$
        If $\| \alpha_e - h_p \|_2 <$ tol, then classify as person $p$ and stop.
    end
end
```

In a typical application it is likely that even if the test image is an image of a person in the database, it is taken with another expression that is not represented in the database. However, the above algorithm works well in such cases, as reported in [104].

Example 14.4. For each of the 10 persons in the Yale database, there is an image of the person winking. We took these as test images and computed the closest image in the database, essentially by using the algorithm above. In all cases the correct person was identified; see Figure 14.4. ∎

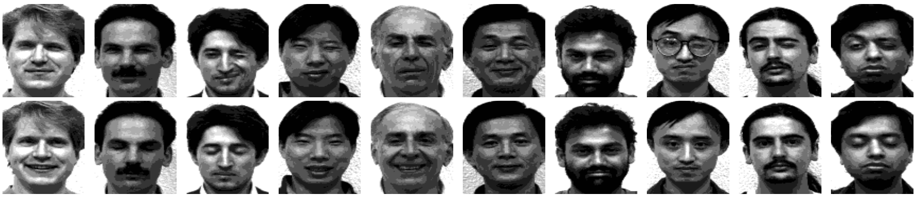


Figure 14.4. *The upper row shows the images to be classified, the bottom row the corresponding closest image in the database.*

14.3 Face Recognition with HOSVD Compression

Due to the ordering properties of the core, with respect to the different modes (Theorem 8.3), we may be able to truncate the core in such a way that the truncated HOSVD is still a good approximation of $\mathcal{A}$. Define $F_k = F(:, 1:k)$ for some value of k that we assume is much smaller than n_i but larger than n_p. Then, for the analysis only, enlarge the matrix so that it becomes square and orthogonal:

$$\widehat{F} = (F_k \, \widetilde{F}_\perp), \qquad \widehat{F}^T \widehat{F} = I.$$

Then truncate the core tensor similarly, i.e., put

$$\widehat{\mathcal{C}} = (\mathcal{S} \times_e G)(1:k, :, :) \times_i F_k. \tag{14.10}$$

It follows from Theorem 8.3, and the fact that the multiplication by G in the e-mode does not affect the HOSVD ordering properties in the i-mode, that

$$\|\widehat{\mathcal{C}} - \mathcal{C}\|_F^2 = \sum_{\nu=k+1}^{n_i} \sigma_\nu^{(i)}.$$

Therefore, if the rate of decay of the image mode singular values is fast enough, it should be possible to obtain good recognition precision, despite the compression. So if we use $\widehat{\mathcal{C}}$ in the algorithm of the preceding section, we will have to solve least squares problems

$$\min_{\alpha_e} \| \, \widehat{C}_e \alpha_e - z \, \|_2$$

with the obvious definition of $\widehat{C}_e$. Now, from (14.10) we have $\widehat{C}_e = F_k \widehat{B}_e$, where $\widehat{B}_e \in \mathbb{R}^{k \times n_p}$. Multiplying by $\widehat{F}$ inside the norm sign we get

$$\| \widehat{C}_e \alpha_e - z \|_2^2 = \| \widehat{B}_e \alpha_e - F_k^T z \|_2^2 + \| \widetilde{F}_\perp^T z \|_2^2.$$

In this "compressed" variant of the recognition algorithm, the operation $\hat{z} = F^T z$ is replaced with $\hat{z}_k = F_k^T z$, and also the least squares problems in the loop are smaller.

Example 14.5. We used the same data as in the previous example but truncated the orthogonal basis in the image mode to rank k. With $k = 10$, all the test images were correctly classified, but with $k = 5$, 2 of 10 images were incorrectly classified. Thus a substantial rank reduction (from 100 to 10) was possible in this example without sacrificing classification accuracy. ■

In our illustrating example, the numbers of persons and different expressions are so small that it is not necessary to further compress the data. However, in a realistic application, to classify images in a reasonable time, one can truncate the core tensor in the expression and person modes and thus solve much smaller least squares problems than in the uncompressed case.