Chapter 10
Robust Face Recognition

10.1 Introduction

In human perception, the role of sparse representation has been studied extensively. Investigators have revealed that in both low-level and mid-level human vision, many neurons in the visual pathway are selective for recognizing a variety of specific stimuli, such as color, texture, orientation, scale, and even view-tuned object images [Olshausen and Field, 1997,Serre, 2006]. Considering these neurons to form an overcomplete dictionary of base signal elements at each visual stage, the firing of the neurons with respect to a given input image is typically highly sparse.

As we discussed in the earlier part of the book, the original goal of sparse representation was not inference nor classification *per se*, but rather representation and compression of signals, potentially using lower sampling rates than the Shannon-Nyquist bound. Therefore, the algorithm performance was measured by the sparsity of the representation and the fidelity to the original signals. Furthermore, individual base elements in the dictionary were not assumed to have any particular semantic meaning – they were typically chosen from standard bases (e.g., Fourier, Wavelet, Curvelet, Gabor), or even generated from random matrices. Nevertheless, the sparsest representation *is* naturally discriminative: amongst all subsets of base vectors, it would select the subset which most compactly expresses the input signal and rejects all other possible but less compact representations.

In this chapter, we exploit the discriminative nature of sparse representation to perform *classification*. Instead of using the generic dictionaries

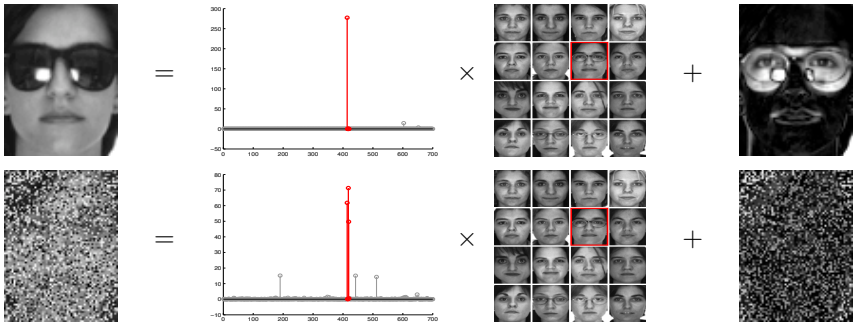


Figure 10.1. **Overview of our approach.** Our method represents a test image (left), which is potentially occluded (top) or corrupted (bottom), as a sparse linear combination of all the training images (middle) plus sparse errors (right) due to occlusion or corruption. Red (darker) coefficients correspond to training images of the correct individual. Our algorithm determines the true identity (indicated with a red box at second row and third column) from 700 training images of 100 individuals (7 each) in the standard AR face database.

mentioned above, we represent a test sample using a data-driven dictionary, whose base elements are *the training samples themselves*. If sufficient training samples are available from each class, it will be possible to represent the test sample as a linear combination of just those training samples from the same class. This representation is naturally sparse, involving only a small fraction of the overall training database. We will see that in many problems of interest, it is actually the *sparsest* linear representation of the test sample in terms of this dictionary, and can be recovered efficiently via sparse optimization. Seeking the sparsest representation therefore automatically discriminates between the various classes present in the training set. Figure 10.1 illustrates this simple idea using face recognition as an example. Sparse representation also provides a simple yet surprisingly effective means of rejecting invalid test samples not arising from any class in the training database: these samples' sparsest representations tend to involve many dictionary elements, spanning multiple classes.

We will motivate and study this new approach to classification within the context of automatic face recognition. Human faces are arguably the most extensively studied object in image-based recognition. This is partly due to the remarkable face recognition capability of the human visual system [Sinha et al., 2006], and partly due to numerous important applications for face recognition technology [Zhao et al., 2003]. In addition, technical issues associated with face recognition are sufficiently representative of object recognition and even data classification in general. In this chapter, the application of sparse representation and compressed sensing to face recognition yields new insights into compensating gross image error or facial occlusion in the context of face recognition.

It has been known that facial occlusion or disguise poses a significant obstacle to robust real-world face recognition [Leonardis and Bischof, 2000, Martinez, 2002, Sanja et al., 2006]. This difficulty is mainly due to the unpredictable nature of the error incurred by occlusion: it may affect any part of the image, and may be arbitrarily large in magnitude. Nevertheless, this error typically corrupts only a fraction of the image pixels, and is therefore sparse in the standard pixel space basis. When the error has such a sparse representation, it can be handled uniformly within the classical sparse representation framework (see Figure 10.1 for an example). Yet in our experiment, we further discovered that as the dimension of the problem grows higher, sparsity solvers such as ℓ_1-minimization seem to be able to recover dense error with ease. In this context, the general theory of sparse representation and compressive sensing falls short in explaining the phenomena of dense error correction with a special kind of dictionaries, called the *cross-and-bouquet* model. We will discuss the conditions in which ℓ_1-minimization guarantees to recover dense error approaching 100% under the cross-and-bouquet model.

10.2 Classification Based on Sparse Representation

A basic problem in object recognition is to use labeled training samples from L distinct object classes to correctly determine the class to which a new test sample belongs. We arrange the given n_i training samples from the i-th class as columns of a matrix $\boldsymbol{A}_i \doteq [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$. In the context of face recognition, we will identify a $w \times h$ grayscale image with the vector $\boldsymbol{v} \in \mathbb{R}^m$ ($m = wh$) given by stacking its columns. Then the columns of $\boldsymbol{A}_i$ are the training face images of the i-th subject.

An immense variety of statistical models have been proposed for exploiting the structure of the $\boldsymbol{A}_i$ for recognition. One particularly simple and effective approach models the samples from a single class as lying on a linear subspace. Subspace models are flexible enough to capture much of the variation in real datasets. In particular in the context of face recognition, it has been observed that images of a face under varying lighting and expression lie on a special low-dimensional subspace [Belhumeur et al., 1997, Basri and Jacobs, 2003a], often called a *face subspace*. This is the only prior knowledge about the training samples we will be using in proposing our solution using sparse representation.

Given sufficient training samples of the i-th object class, $\boldsymbol{A}_i = [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, any new (test) sample $\boldsymbol{y} \in \mathbb{R}^m$ from the same class approximately lie in the linear span of the training samples associated with object i:

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{v}_{i,1} + \alpha_{i,2}\boldsymbol{v}_{i,2} + \cdots + \alpha_{i,n_i}\boldsymbol{v}_{i,n_i}, \tag{10.2.1}$$

for some scalars $\alpha_{i,j} \in \mathbb{R}, j = 1, 2, \ldots, n_i$.

Since the membership i of the test sample is initially unknown, we define a new matrix $\boldsymbol{A}$ for the entire training set as the concatenation of the n training samples from all L object classes:

$$\boldsymbol{A} \doteq [\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_L] = [\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2}, \ldots, \boldsymbol{v}_{L,n_L}]. \tag{10.2.2}$$

Then the linear representation of $\boldsymbol{y}$ can be rewritten in terms of all training samples as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o \quad \in \mathbb{R}^m, \tag{10.2.3}$$

where $\boldsymbol{x}_o = [0, \cdots, 0, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}, 0, \ldots, 0]^* \in \mathbb{R}^n$ is a coefficient vector whose entries are zero except those associated with the i-th class.

This motivates us to seek the sparsest solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ via sparse optimization, such as ℓ_1-minimization:

$$\hat{\boldsymbol{x}} = \arg\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{10.2.4}$$

Given a new test sample $\boldsymbol{y}$ from one of the classes in the training set, we first compute its sparse representation $\hat{\boldsymbol{x}}$ via (10.2.4). Ideally, the nonzero entries in the estimate $\hat{\boldsymbol{x}}$ will be all associated with the columns of $\boldsymbol{A}$ from a single object class i, and we can easily assign the test sample $\boldsymbol{y}$ to that class. However, noise and modeling error may lead to small nonzero entries associated with multiple object classes (for example, see Figure 10.1 bottom case). Based on the global, sparse representation, one can design many possible classifiers to resolve this. For instance, we can classify $\boldsymbol{y}$ based on how well the coefficients associated with all the training samples of each object reproduce $\boldsymbol{y}$.

More specifically, for each class i, let $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ be the characteristic function which selects the coefficients associated with the i-th class. For $\boldsymbol{x} \in \mathbb{R}^n$, $\delta_i(\boldsymbol{x}) \in \mathbb{R}^n$ is a new vector whose only nonzero entries are the entries in $\boldsymbol{x}$ that are associated with class i. Using only the coefficients associated with the i-th class, one can approximate the given test sample $\boldsymbol{y}$ as $\hat{\boldsymbol{y}}_i = \boldsymbol{A}\delta_i(\hat{\boldsymbol{x}})$. We then classify $\boldsymbol{y}$ based on these approximations by assigning it to the object class that minimizes the residual between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}_i$:

$$\min_i \; r_i(\boldsymbol{y}) \doteq \|\boldsymbol{y} - \hat{\boldsymbol{y}}_i\|_2. \tag{10.2.5}$$

Algorithm 10.1 below summarizes the complete recognition procedure.

Example 10.2.1. *(ℓ_1-Minimization vs. ℓ_2-Minimization) To illustrate how Algorithm 1 works, we randomly select half of the* $2,414$ *images in the Extended Yale B database as the training set, and the rest for testing. In this example, we subsample the images from the original* 192×168 *to size* 12×10*. The pixel values of the downsampled image are used as 120-D features – stacked as columns of the matrix* $\boldsymbol{A}$ *in the algorithm. Hence matrix* $\boldsymbol{A}$ *has size* 120×1207*, and the system* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ *is underdetermined. Figure*

Algorithm 10.1 : Sparse Representation-based Classification (SRC)

1: **Input:** a matrix of training samples $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_L] \in \mathbb{R}^{m \times n}$ for L classes, a test sample $\boldsymbol{y} \in \mathbb{R}^m$.
2: Normalize the columns of A to have unit ℓ^2-norm.
3: Solve the ℓ^1-minimization problem (10.2.4).

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{10.2.6}$$

4: Compute the residuals $r_i(\boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{A}\,\delta_i(\hat{\boldsymbol{x}})\|_2$ for $i = 1, \ldots, L$.
5: **Output:** identity$(\boldsymbol{y}) = \arg\min_i r_i(\boldsymbol{y})$.

10.2 top illustrates the sparse coefficients recovered by Algorithm 1 for a test image from the first subject. The figure also shows the features and the original images that correspond to the two largest coefficients. The two largest coefficients are both associated with training samples from subject 1. Figure 10.2 bottom shows the residuals with respect to the 38 projected coefficients $\delta_i(\hat{\boldsymbol{x}}_1)$, $i = 1, 2, \ldots, 38$. With 12×10 downsampled images as features, Algorithm 1 achieves an overall recognition rate of 92.1% across the Extended Yale B database. Whereas the more conventional minimum ℓ^2-norm solution to the underdetermined system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is typically quite dense, minimizing the ℓ^1-norm favors sparse solutions, and provably recovers the sparsest solution when this solution is sufficiently sparse. To illustrate this contrast, Figure 10.3 top shows the coefficients of the same test image given by the conventional ℓ^2-minimization, and Figure 10.3 bottom shows the corresponding residuals with respect to the 38 subjects. The coefficients are much less sparse than those given by ℓ^1-minimization (in Figure 10.2), and the dominant coefficients are not associated with subject 1. As a result, the smallest residual in Figure 10.3 does not correspond to the correct subject.

10.3 Robustness to Occlusion or Corruption

In many real-world scenarios, the test image $\boldsymbol{y}$ could be partially occluded or corrupted. In this case, the linear model (10.2.3) should be modified as

$$\boldsymbol{y} \;=\; \boldsymbol{y}_o + \boldsymbol{e}_o \;=\; \boldsymbol{A}\,\boldsymbol{x}_o + \boldsymbol{e}_o, \tag{10.3.1}$$

where $\boldsymbol{e}_o \in \mathbb{R}^m$ is a vector of errors – a fraction, ρ, of its entries are nonzero. The nonzero entries of $\boldsymbol{e}_o$ represent which pixels in $\boldsymbol{y}$ are corrupted or occluded. The locations of corruption can differ for different test images and are not known to the algorithm. The errors may have arbitrary magnitude

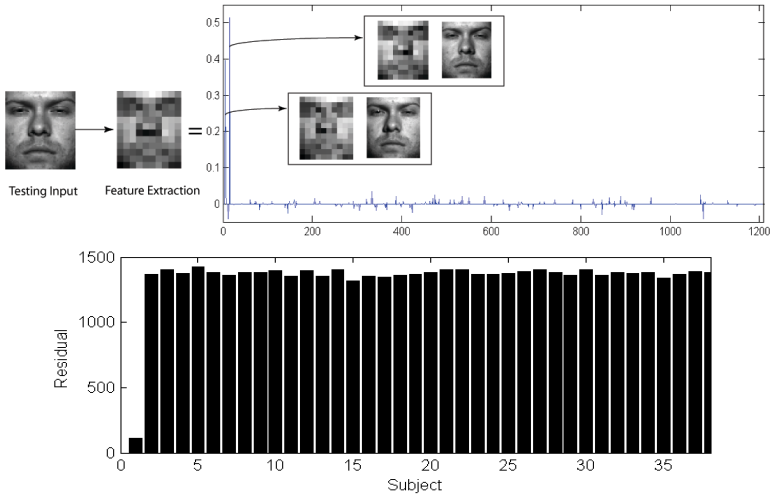


Figure 10.2. **A valid test image.** Top: Recognition with 12×10 downsampled images as features. The test image $\boldsymbol{y}$ belongs to subject 1. The values of the sparse coefficients recovered from Algorithm 1 are plotted on the right together with the two training examples that correspond to the two largest sparse coefficients. Bottom: The residuals $r_i(\boldsymbol{y})$ of a test image of subject 1 with respect to the projected sparse coefficients $\delta_i(\hat{\boldsymbol{x}})$ by ℓ^1-minimization. The ratio between the two smallest residuals is about 1:8.6.

and therefore cannot be ignored or treated with techniques designed for small noise.

A fundamental principle of coding theory [MacWilliams and Sloane, 1981] is that *redundancy* in the measurement is essential to detecting and correcting gross errors. Redundancy arises in object recognition because the number of image pixels is typically far greater than the number of subjects that have generated the images. In this case, even if a fraction of the pixels are completely corrupted, recognition may still be possible based on the remaining pixels. On the other hand, traditional feature extraction schemes discussed in the previous section would discard useful information that could help compensate for the occlusion. In this sense, no representation is more redundant, robust, or informative than the original images. Thus, when dealing with occlusion and corruption, we should always work with the highest possible resolution, performing downsampling or feature extraction only if the resolution of the original images is too high to process.

Of course, redundancy would be of no use without efficient computational tools for exploiting the information encoded in the redundant data. The difficulty in directly harnessing the redundancy in corrupted raw images has led researchers to instead focus on *spatial locality* as a guiding principle for robust recognition. Local features computed from only a small fraction of the image pixels are clearly less likely to be corrupted by occlusion than

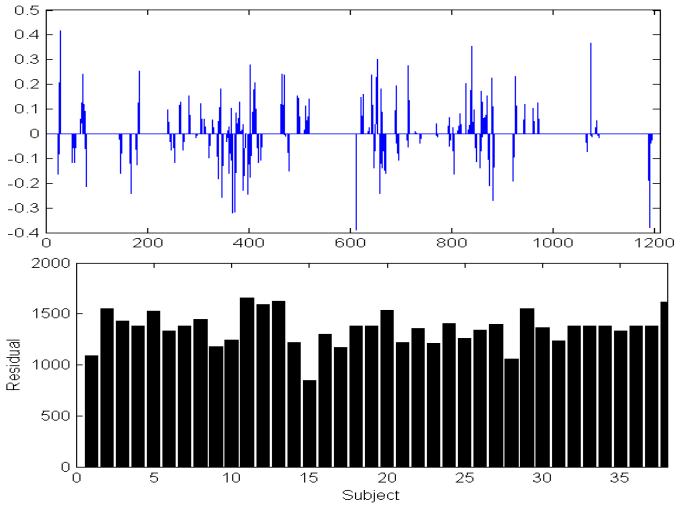


Figure 10.3. **Non-sparsity of the ℓ^2-minimizer.** Top: Coefficients from ℓ^2-minimization, using the same test image as Figure 10.2. The recovered solution is not sparse and hence less informative for recognition (large coefficients do not correspond to training images of this test subject). Bottom: The residuals of the test image from subject 1 with respect to the projection $\delta_i(\hat{\boldsymbol{x}})$ of the coefficients obtained by ℓ^2-minimization. The ratio between the two smallest residuals is about 1:1.3. The smallest residual is not associated with subject 1.

holistic features. In face recognition, methods such as ICA [Kim et al., 2005] and LNMF [Li et al., 2001] exploit this observation by adaptively choosing filter bases that are locally concentrated. Local Binary Patterns [Ahonen et al., 2006] and Gabor wavelets [Lades et al., 1993] exhibit similar properties, since they are also computed from local image regions. A related approach partitions the image into fixed regions and computes features for each region [Pentland et al., 1994, Martinez, 2002]. Notice, though, that projecting onto locally concentrated bases transforms the domain of the occlusion problem, rather than eliminating the occlusion. Errors on the original pixels become errors in the transformed domain, and may even become less local. The role of feature extraction in achieving spatial locality is therefore questionable, since *no bases or features are more spatially localized than the original image pixels themselves.* In fact, the most popular approach to robustifying feature-based methods is based on randomly sampling individual pixels [Leonardis and Bischof, 2000].

Now, let us show how the sparse representation classification framework can be extended to deal with occlusion. Let us assume that the corrupted pixels are a relatively small portion ρ of the total image pixels. Then the error vector $\boldsymbol{e}_o$, like the vector $\boldsymbol{x}_o$, should be sparse nonzero entries. Since

$\boldsymbol{y}_o = \boldsymbol{A}\boldsymbol{x}_o$, we can rewrite (10.3.1) as

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{A}, & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_o \\ \boldsymbol{e}_o \end{bmatrix} \doteq \boldsymbol{B}\,\boldsymbol{w}_o. \tag{10.3.2}$$

Here, $\boldsymbol{B} = [\boldsymbol{A},\ \boldsymbol{I}] \in \mathbb{R}^{m\times(n+m)}$, so the system $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{w}$ is always underdetermined and does not have a unique solution for $\boldsymbol{w}$. However, in theory, the correct generating $\boldsymbol{w}_o = [\boldsymbol{x}_o,\ \boldsymbol{e}_o]$ has at most $n_i + \rho m$ nonzeros. We might therefore hope to recover $\boldsymbol{w}_o$ as the sparsest solution to the system $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{w}$. As before, we attempt to recover the sparsest solution $\boldsymbol{w}_o$ via sparse optimization, such as solving the following ℓ^1-minimization problem:

$$\hat{\boldsymbol{w}} = \arg\min \|\boldsymbol{w}\|_1 \quad \text{subject to} \quad \boldsymbol{B}\boldsymbol{w} = \boldsymbol{y}. \tag{10.3.3}$$

Algorithm 10.2 summarizes the complete recognition procedure.

Algorithm 10.2: Robust Sparse Representation-based Classification

1: **Input:** a matrix of training samples $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_L] \in \mathbb{R}^{m\times n}$ for L classes, a test sample $\boldsymbol{y} \in \mathbb{R}^m$, (and an optional error tolerance $\epsilon > 0$.)

2: Normalize the columns of $\boldsymbol{A}$ to have unit ℓ^2-norm.

3: Solve the ℓ^1-minimization problem:

$$\begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{e}} \end{bmatrix} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \|\boldsymbol{e}\|_1 \quad \text{subject to} \quad [\boldsymbol{A}, \boldsymbol{I}] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{e} \end{bmatrix} = \boldsymbol{y}. \tag{10.3.4}$$

4: Compute the residuals $r_i(\boldsymbol{y}) = \|\boldsymbol{y} - \hat{\boldsymbol{e}} - \boldsymbol{A}\,\delta_i(\hat{\boldsymbol{x}})\|_2$ for $i = 1, \ldots, L$.

5: **Output:** identity$(\boldsymbol{y}) = \arg\min_i r_i(\boldsymbol{y})$.

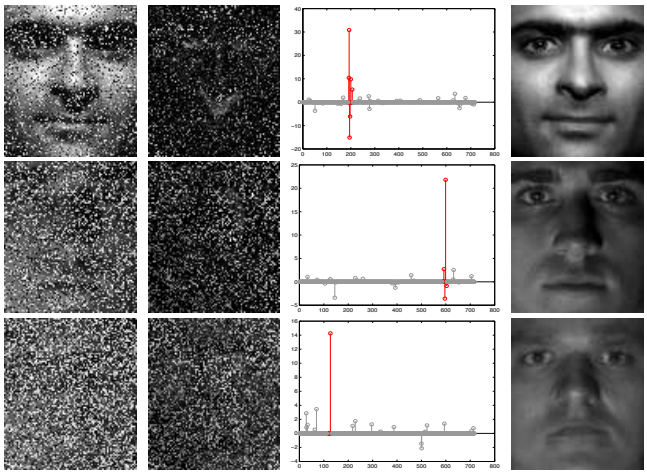
More generally, one can assume that the corrupting error $\boldsymbol{e}_o$ has a sparse representation with respect to some basis $\boldsymbol{A}_e \in \mathbb{R}^{m\times n_e}$. That is, $\boldsymbol{e}_o = \boldsymbol{A}_e\boldsymbol{u}_o$ for some sparse vector $\boldsymbol{u}_o \in \mathbb{R}^m$. Here, we have chosen the special case $\boldsymbol{A}_e = \boldsymbol{I} \in \mathbb{R}^{m\times m}$ as $\boldsymbol{e}_o$ is assumed to be sparse in the natural pixel coordinates. If the error $\boldsymbol{e}_o$ is instead more sparse with respect to another basis, e.g., Fourier or Haar, we can simply redefine the matrix $\boldsymbol{B}$ by appending $\boldsymbol{A}_e$ to $\boldsymbol{A}$ and instead seek the sparsest solution $\boldsymbol{w}_o$ to the equation:

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{w} \quad \text{with} \quad \boldsymbol{B} = [\boldsymbol{A},\ \boldsymbol{A}_e] \quad \in \mathbb{R}^{m\times(n+n_e)}. \tag{10.3.5}$$

In this way, the same formulation can handle more general classes of sparse corruption.

Experimental verification of the algorithm.

We test the robust version of SRC applied to face recognition using the Extended Yale B Face Database. We choose Subsets 1 and 2 (717 images,



(a) (b) (c) (d)

Figure 10.4. **Illustration of recovered sparse representation and sparse error under random corruption.** (a) Test images $\boldsymbol{y}$ from Extended Yale B, with random corruption. Top row: 30% of pixels are corrupted, Middle row: 50% corrupted, Bottom row: 70% corrupted. (b) Estimated errors $\hat{\boldsymbol{e}}_1$. (c) Estimated sparse coefficients $\hat{\boldsymbol{x}}_1$. (d) Reconstructed images $\boldsymbol{y}_r$. SRC correctly identifies all three corrupted face images.

normal-to-moderate lighting conditions) for training, and Subset 3 (453 images, more extreme lighting conditions) for testing. Without occlusion, this is a relatively easy recognition problem. This choice is deliberate, in order to isolate the effect of occlusion. The images are resized to 96×84 pixels, so in this case $\boldsymbol{B} = [\boldsymbol{A},\ \boldsymbol{I}]$ is an $8,064 \times 8,761$ matrix, a manageable size for most computers.

We then corrupt a percentage of randomly chosen pixels from each of the test images, replacing their values with iid samples from a uniform distribution. The corrupted pixels are randomly chosen for each test image and the locations are unknown to the algorithm. We vary the percentage of corrupted pixels from 0% to 90%. Figure 10.4 shows several example test images. To the human eye, beyond 50% corruption, the corrupted images (Figure 10.4(a) second and third rows) are barely recognizable as face images; determining their identity seems out of the question. Yet even in this extreme circumstance, SRC correctly recovers the identity of the subjects.

We quantitatively compare our method to four popular techniques for face recognition in the vision literature. The Principal Component Analysis (PCA) approach of [Turk and Pentland, 1991] is not robust to occlusion. Although there are many variations to make PCA robust to corruption or

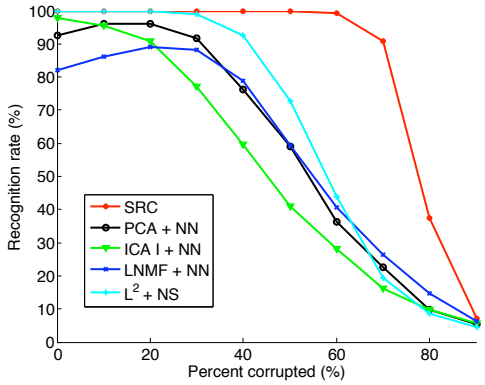


Figure 10.5. **Recognition rates under random corruption.** The recognition rate across the entire range of corruption for various algorithms. SRC (red curve) significantly outperforms others, performing almost perfectly up to 60% random corruption (see table below).

incomplete data, some of which have been applied to robust face recognition, e.g., [Sanja et al., 2006], here we use the basic PCA to provide a standard baseline for comparison. The remaining three techniques are designed to be more robust to occlusion. Independent Component Analysis (ICA) architecture I [Kim et al., 2005] attempts to express the training set as a linear combination of statistically independent basis images. Local Nonnegative Matrix Factorization (LNMF) [Li et al., 2001] approximates the training set as an additive combination of basis images, computed with a bias toward sparse bases. For PCA, ICA and LNMF, the number of basis components is chosen to give the best performance over the range $\{100, 200, 300, 400, 500, 600\}$. Finally, to demonstrate that the improved robustness is really due to the use of the ℓ^1-norm, we compare to a least-squares technique that first projects the test image onto the subspace spanned by all face images, and then performs nearest subspace.

Figure 10.5 plots the recognition performance of SRC and its five competitors, as a function of the level of corruption. We see that the algorithm dramatically outperforms others. From 0% upto 50% occlusion, SRC correctly classifies all subjects. At 50% corruption, none of the others achieves higher than 73% recognition rate, while the proposed algorithm achieves 100%. Even at 70% occlusion, the recognition rate is still 90.7%. This greatly surpasses the theoretical bound of worst-case corruption (13.3%) that the algorithm is ensured to tolerate. Clearly, the worst-case analysis is too conservative for random corruption.

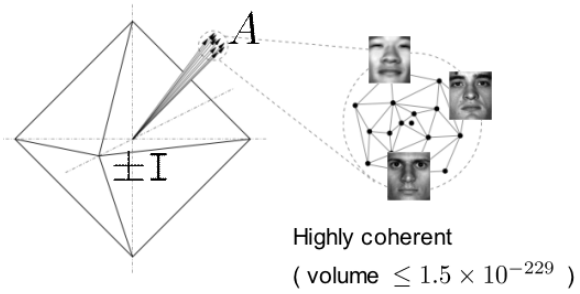


Figure 10.6. The cross-and-bouquet model for face recognition. The raw images of human faces expressed as columns of $\boldsymbol{A}$ are clustered with small variance.

10.4 Dense Error Correction with the Cross and Bouquet

In this section, we will take a closer look at the sparsity model (10.3.2). Recall in the classical sparse representation theory, one of the conditions for the successful recovery of a sparse signal is that the dictionary under which the sparsity is represented must be sufficiently incoherent. However, the dictionary $\boldsymbol{B} = [\boldsymbol{A}, \boldsymbol{I}] \in \mathbb{R}^{m\times(n+m)}$ is quite special. In its first part, the matrix $\boldsymbol{A}$ consists of column vectors that represent the pixel values of all face images. As m grows higher, the convex hull spanned by all the face image vectors becomes an extremely tiny portion of the unit sphere in $\mathbb{R}^m$, which means they are highly correlated. As an example shown in Figure 10.6, the face images lie in $\mathbb{R}^m$ where $m = 8,064$, and all the image vectors are contained within a spherical cap of volume $\leq 1.5 \times 10^{-229}$. These vectors are tightly bundled together as a "bouquet." In the second part of $\boldsymbol{B}$, $\boldsymbol{I}$ is a standard m-by-m identity matrix, which is also called a standard pixel basis. Then $\boldsymbol{I}$ and its negative copy $-\boldsymbol{I}$ form a "cross" in $\mathbb{R}^m$, also illustrated in Figure 10.6. We call this type of dictionaries a *cross and bouquet* (CAB) model.

The CAB model belongs to a special class of sparse representation problems where the dictionary is a concatenation of two or more sub-dictionaries. Examples include the merger of wavelet and heavy-side dictionaries [Chen et al., 2001] and the combination of texture and cartoon dictionaries in morphological component analysis [Elad et al., 2005]. However, in contrast to most other examples, not only is the CAB dictionary as a whole inhomogeneous as we discussed above, in fact the ground-truth signal $(\boldsymbol{x}_o, \boldsymbol{e}_o)$ is also very inhomogeneous, namely, the sparsity of $\boldsymbol{x}_o$ is limited by the number of training images per subject for the purpose of recognition, while we would like to handle as dense corruption error $\boldsymbol{e}_o$ as possible, to guarantee good error correction performance. The above experiment is indeed a concrete demonstration that shows sparse optimization

such as ℓ_1-minimization seems to be able to recover very dense error $\boldsymbol{e}_o$. This further contradicts our understanding in the classical sparse representation theory, where the corruption error to be recovered is typically assumed to be sparse.

The reason sparse optimization could recover even dense error is mainly due to the special nature of the sparsity of the signal $\boldsymbol{x}_o$, which is called *weak proportional growth*. Again, assume the signal

$$\boldsymbol{w}_o = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{e}_o,$$

where $\boldsymbol{e}_o \in \mathbb{R}^m$ is a vector of error of arbitrary magnitude. We also assume the columns of $\boldsymbol{A}$ are i.i.d. samples from a Gaussian distribution: $\boldsymbol{A} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N] \in \mathbb{R}^{m\times N}$, where $\boldsymbol{v}_i \sim_{iid} N(\boldsymbol{\mu}, \frac{\nu^2}{m}\boldsymbol{I}_m)$, and $\|\boldsymbol{\mu}\|_2 = 1$, $\|\boldsymbol{\mu}\|_\infty \leq C_\mu m^{-1/2}$.

Definition 10.4.1 (Weak Proportional Growth)**.** *A sequence of signel-error problems $(\boldsymbol{x}_o, \boldsymbol{e}_o)$ exhibits weak proportional growth with parameters $\delta > 0$, $\rho \in (0,1)$, $C_0 > 0$, and $\eta_0 > 0$, denoted as $WPG_{\delta,\rho,C_0,\eta_0}$, if as $m \to \infty$,*

$$\frac{N}{m} \to \delta, \quad \frac{\|\boldsymbol{e}_o\|_0}{m} \to \rho, \quad \|\boldsymbol{x}_o\|_0 \leq C_0 m^{1-\eta_0}. \tag{10.4.1}$$

In other words, in the weak proportional growth scenario, $\|\boldsymbol{e}_o\|_0$ grows linearly with respect to m, but $\|\boldsymbol{x}_o\|_0$ is sublinear.

Theorem 10.4.2 (Dense Error Correction with the Cross and Bouquet)**.** *For any $\delta > 0$, there exists $\nu_0(\delta) > 0$ such that if $\nu < \nu_0$ and $\rho < 1$, in $WPG_{\delta,\rho,C_0,\eta_0}$ with $\boldsymbol{A}$ distributed according to (10.4.1), if the error support and the signs of nonzero elements are chosen uniformly at random then as $m \to \infty$, the probability of successfully recovering $(\boldsymbol{x}_o, \boldsymbol{e}_o)$ via Algorithm 10.2 approaches to one.*

That is, as long as the bouquet is sufficiently tight, under the assumption of weak proportional growth, asymptotically ℓ_1-minimization recovers any non-negative sparse signal from almost any error with support size less than 100%! A detailed proof of this theorem can be found in [Wright and Ma, 2010]. Although in general sparse representation problems may not satisfy the weak proportional growth assumption, the assumption is valid in the face recognition example, whereby the number of training samples per subject n_i usually does not grow proportionally with the dimension of the image.

10.5 Notes and References

Face recognition when the query image is not aligned with the gallery: simultaneous face alignment and recognition. Relationships with face verification etc.

Exercises

10.1 (Robust Face Recognition*). *Download the Extended Yale B database. Using the cropped face image set in the database to form a gallery set and a query set. Code a robust face recognition system and demonstrate its performance in the same setting discussed in the experiment of this chapter.*

10.2 (Randomfaces*). *In the literature, there are facial feature extraction methods that reduce the dimensionality of face images according to some linear transformations. In this exercise, we will implement two well-established methods, and compare their performance in terms of recognition accuracy with the results using random projection in compressive sensing.*

1. *Code a function that extracts Eigenface features. Demonstrate the recognition accuracy of robust face recognition in Exercise 10.2 in the Eigenface space with respect to different feature dimensions.*
2. *Code a function that extracts Fisherface features. Demonstrate its recognition accuracy with respect to different feature dimensions.*
3. *Code a function that extracts lower-dimensional features using random projection. This is called* Randomface *features. Demonstrate its recognition accuracy with respect to different feature dimensions, and compare with those of Eigenface and Fisherface features.*

10.3 (Receiver Operating Charactristic (ROC)*). *In the presence of potential irrelevant test samples, it is important to evaluate the performance of a classifier not only based on the* true positive rate*, but often more importably on* false positive rate*. The curve that measures the true positive rates under various false positive rates is known as the* receiver operating charactristic *(ROC) curve.*[1]

In this exercise, code a program that plots a representative ROC curve of the robust face recognition algorithm. Exclude half of the subject classes from the gallery set of the Extended Yale B database, and designate them as outlying subjects. Implement the outlier rejection rule based on the sparse coefficient concentration, and plot the ROC curve with respect to different threshold values of the concentration index.

[1]There are different definitions of the ROC curve. There are four basic performance rates: true positive, false positive, true negative, and false negative.