Chapter 5

Concluding Remarks

5.1 Conclusions

This thesis is focused on the problem of object representation and classification using local visual features extracted from images. One of the main inconvenients faced by the use of visual features is the high dimensionality of data. Additionally, the data can be contaminated by noise, and not necessarily all visual features contribute to classification.

One of the goals of this thesis is the evaluation of linear transformations of data in order to model low dimensional pattern structures present in high dimensional data. That is, a whole chapter is focused on the use of unsupervised linear transformations of data. In this chapter we evaluate object recognition performances using low dimensional spaces using several object databases. In this sense, we can benefit from linear transformations if the projection improves or mantains the same information of the high dimensional space and produces reliable classifiers. As example, linear transformations can provide this benefit by reducing dimensionality preserving relevant information present in high dimensional spaces. Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF) and Weighted Non-negative Matrix Factorization (WNMF) are presented as 3 unsupervised linear transformations of data which are helpful for reducing data dimensionality. Independent Component Analysis (ICA) is another linear transformation which is introduced in order to reduce the complexity of estimating density functions.

PCA has been used for a long time as a linear transformation of data in order to reduce dimensionality preserving relevant information. It has been shown that visual local features can be expressed using positive representations and we find that PCA is not adapted for such representations. Mainly, when original positive high dimensional spaces are described in terms of low dimensional PCA coefficients, coefficients are also negative. As example, when color histograms are represented using a low dimensional PCA subspace, its projected bases are non understandable for a human observer. In this sense, we introduced a recent technique called Non-negative Matrix Factorization (NMF) which uses positive assumptions in order to find a low dimensional subspace.

The additive property resulting from the non-negativity constraints of NMF has

been shown to result in bases that represent local components of the original data (in this thesis we have seen local parts of digits, local parts of faces and local color components corresponding to local patches of color images). This parts-based representation leads us to compare NMF and PCA in typical object recognition scenarios as both approaches are based on different assumptions. However, we have experimented some unwanted behaviour of NMF when applied to data which is not uniformly distributed. This problem commonly appears when we deal with local data representations. We faced this problem introducing a modified approach of NMF. We introduced a weight matrix that minimizes the appearance of repeated bases in matrix $\mathbf{W}$ as it is based on the frequency of each vector used for training. Then, we decrease the influence of the most frequent training vectors as we want to reduce the appearance of redundant information. In this context, we compared NMF with respect to WNMF and the following contributions have been made,

- We have formalized a weighted version of NMF using a matrix which weighs vectors used for obtaining a NMF model (matrices $\mathbf{W}$ and $\mathbf{H}$).
- In problems where not uniformly distributed data is present, WNMF outperforms NMF depending on the dimensionality of the subspace. It is found by experiments that there is a threshold dimension which determines the outperformance of WNMF. However, this threshold dimension directly depends on the problem since it is related to the complexity of data.
- The threshold dimension of the subspace which determines the outperformance of WNMF with respect to NMF is obtained through the analysis of $\mathbf{W}$ bases or through the analysis of the reconstruction error of both NMF and WNMF techniques. However, no a priori information about original data can be used to know which is this threshold dimension.
- WNMF is a constrained version of NMF as introduces a weight matrix. Then, if we firstly perform a set of iterations using WNMF and we use this solution to iterate NMF, the final solution is better than the one obtained using WNMF. So that, we provide a good scheme where we are able to use only few iterations of WNMF to achieve a better starting point for NMF.

Once we introduced these three unsupervised linear transformations of data which have been mainly used for reducing high dimensional spaces to low dimensional ones, we performed several experiments in the context of object classification. We divided these experiments in three schemes:

1. **Reconstruction distances**: NMF and PCA search for a solution which minimizes the mean squared error (MSE) in terms of the reconstruction error. NMF is restricted to positive constraints. Then, a first attempt to perform classification using the reconstruction distances is shown. Using a wide variety of 10 data classes, we analyzed which technique (NMF or PCA) is adapted to each data class. Then, we created a unified framework which merges NMF and PCA models and is able to classify new data vectors. This is the first attempt to use a combined classifier using PCA and NMF. Experimental results demonstrate

that this unified framework outperforms results based on PCA and/or NMF alone.

2. **Parametric Model**: Using the coefficients of PCA and NMF, we estimated the probability density function of the projected original high dimensional space. This is done assuming that PCA is modeled using a Gaussian distribution and NMF using a Poisson distribution. Then, using this new probabilistic approach we faced the same classification problem of 10 data classes in order to find which technique can be used for each data class. Experimental results are acceptable and slighty better than using reconstruction distances.

3. **Nonparametric Model**: The k-nearest neighbor has been used as a nonparametric model for classifying the projected coefficients obtained using NMF and PCA. This is the most common approach to be used with PCA. However, NMF has not been used in such a context. We contributed with an extended analysis of several metric distances to be used in the projected space of NMF. Then, a comparison with PCA has been made in terms of performances with different object databases evaluating their robustness in front of occlusions.

In order to take advantage of the positive aspects of both techniques (PCA and NMF), we performed several experimental tests and the following contributions have been made,

- **Combined framework of PCA and NMF**: PCA and NMF/WNMF can be combined in an unified framework based on reconstruction distances and/or a parametric model. In a context of using several data classes, we create a model for each data class. We have experimentally observed that some data classes are better represented using NMF and some other ones with PCA. It seems that we can stablish a priori which data classes can be better represented using NMF: the most complex ones. A complex class is the one which contains several local behaviours and is very dispersed in the space.

- **Selection of a metric distance for NMF**: We have experimentally tested the use of traditional distance measures with the projected coefficients of NMF. We state that,

    1. **Earth Mover's Distance (EMD)**: Is the best metric distance to be used with NMF when occlusions are not present. However, the computational costs associated with this distance are extremely demanding.

    2. L_1 **metric distance**: Is the best metric distance to be used with NMF when occlusions are not present and EMD can not be used as it requires a huge amount of computational resources.

    3. **Cosine metric**: Is the best distance to be used with NMF when occlusions are present but not vice-versa. That is, if occlusions are not present is the worst metric distance. We tested two kinds of occlusions: handmade occlusions (digits) and real world occlusions (faces with sunglasses and a scarf).

4. **Good performance of NMF with cosine metric in front of occlusions**: The cosine metric in conjunction with NMF performs better than a commercial face recognition system, the FaceIt. This confirms the adaptability of $NMF + Cos$ for scenarios such as recognition of faces with the presence of natural occlusions.

- **Robustness of NMF in front of occlusions in a classification context**: NMF is a parts-based representation that has not been used for classification purposes. In this thesis, we performed several experimental tests with NMF and our main statement is that NMF is inherently robust under the presence of occlusions. This was a well-known fact in the context of data representation but we show that is also true in the context of object classification.

Apart of using PCA and NMF/WNMF which are based on linear data transformations and produce compact representations with usually less dimensions than the original high dimensional spaces, we introduce the Independent Component Analysis (ICA). ICA is used to transform a space in order to obtain reliable probability density functions. So that, ICA adapts a space in order to obtain independent components and estimate probability density functions for each independent component. In this context, we then propose to take advantage of independent component analysis in the context of statistical pattern classification. Since Bayesian classification makes use of the conditional densities, the choice of any representation oriented to simplifying density estimation necessarily implies the use of class-conditional representations. In this case, and under certain assumptions, independent component analysis can provide a framework where conditional independence can be assumed. Naive Bayes appears as the naturally associated classifier for this situation. In this sense, the following contribution has been made,

- A class-conditional representation is exposed to deal with different data classes. When independent component analysis is introduced as a representation, we obtain the context we named as class-conditional independent component analysis (CC-ICA). Adapting such a framework to classification results in a modified naive Bayes classifier. An experimental test showing the outperformance of CC-ICA is presented.

All these contributions are based on schemes which work with local features. The spatial arrangement of these local features is not taken into account. So that, all these approaches inherently contain a high degree of ambiguity. The second part of this thesis focuses on a possible new framework to deal with local features and take into account the neighborhood of each local feature since we want to decrease the level of ambiguity. We introduced the concept of k-tuple in order to represent the local appearance of an object at k different keypoints. The real contribution of this framework is the use of Independent Component Analysis to obtain factored joint distributions of tuples and work with a computationally tractable scheme. Several experimental tests have been performed and the method is specially adapted to cluttered environments with the presence of occlusions. Furthermore, the number of possible k-tuples is extremely huge and we propose to select a relevant subset of them. However, we

should be careful in order to select which tuples are used to learn the joint factored distributions. In this sense, we propose that,

- Objects which can suffer from occlusions should be represented using k-tuples with neighbor points in order to mantain the internal object structure. Then, recognition and detection will be robust under the presence of occlusions.
- Object which will not suffer from occlusions can be represented using k-tuples with distant points. Then, we will be modeling local features considering the whole structure of the object and it will not be robust under the presence of occlusions.

5.2 Future Work

The work covered in this thesis provides a number of areas of interest that may be worth further investigation. Among others, it may be interesting to consider the following lines for further research:

Non-negative Matrix Factorization issues

Non-negative Matrix Factorization (NMF) has been extended to its weighted version (WNMF) in order to decrease the level of redundancy obtained with NMF when applied to data which is not uniformly distributed. Even the outperformance of WNMF with respect to NMF, there are several issues that remain to be dealed. For example,

- Theoretical improvement of WNMF with respect to NMF. In this thesis, we have exposed WNMF as an alternative scheme to NMF which provides a better solution if the subspace dimension is correctly chosen. A good number of experimental tests have been done demonstrating this fact but it remains a theoretical approach to the reasons of this improvement.
- The convergence speed of NMF/WNMF should be improved. PCA is a direct scheme which is based on the eigenvectors and eigenvalues of the covariance matrix but NMF/WNMF is an iterative process and it requires a huge number of computational resources. We believe that the convergence speed could be improved by using non-random matrices as a starting point for NMF/WNMF. For example, we can use the input data vectors to extract some relevant information and use this information to initialize the input data matrices for NMF.
- It is stated in this thesis that NMF is well suited for complex data classes and PCA for simple data classes. However, it remains an exhaustive study reflecting this statement and explaining specifically what means *complex data classes*.

Independent Component Analysis issues

In general, relaxing the assumptions of class-conditional independent component analysis involves using a representation other than ICA for each class. A natural extension, would be to make use of mixtures of independent component analysers. And another possible extension would be to contemplate nonlinear independent component analysis.

The close link between sparsity and independence could allow to hold the independence assumption in the presence of sparse data not necessarily obtained through independent component analysis. As noted, NMF is a sparse model and prior assumptions on this model can force sparsity in the encodings. In this case, if an adequate density model is considered for the sparse and non-negative data, statistical classification could be simply extended to the non-negative context. This would combine the advantages of NMF as a representation with the advantages of statistical classifieres. It would be intetersting the use of a Class-Conditional Non-negative Matrix Factorization (CC-NMF).

Factored Joint Distributions of k-tuples issues

We proposed a joint distribution of k-tuples in order to model the spatial arrangement as well as the local information of the local features of objects for object detection and recognition purposes. The model factorization is achieved using Independent Component Analysis (ICA). However, it would be interesting to use NMF in such a context. And this would be possible if we are able to find an adequate density model for NMF.

Extracted local features correspond to local jets and local color information. However, we should test this framework using more relevant local information and really high dimensional features. But, in doing so, the computational costs of the joint distributions would also be increased. As example, instead of using the mean color descriptor of each RGB channel, we can use local color histograms.

A good topic to be dealed in such a framework is related to the tuple selection. As tuples of keypoints is the input source of information, they should be carefully selected. We show a first approach where tuples with neighbor keypoints, tuples with distant keypoints and tuples chosen at random are considered. However, it would be more interesting to chose those tuples that are discriminant for object detection/recognition purposes.

Appendix A

Principal Component Analysis

This appendix contains a detailed description of Principal Component Analysis (PCA).

A.1 PCA by Maximizing Variance

First we will derive PCA by maximizing the variance in the direction of principal vectors. Let us suppose that we have N M-dimensional vectors $\mathbf{x}_j$ aligned in the data matrix $\mathbf{X} \in \Re^{M \times N}$. Let $\mathbf{u}$ be a direction (a vector of length 1) in $\Re^M$. The projection of the j-th vector $\mathbf{x}_j$ onto the vector $\mathbf{u}$ can be calculated in the following way:

$$a_j =< \mathbf{x}_j, \mathbf{u} >= \mathbf{u}^T \mathbf{x}_j = \sum_{i=1}^{M} u_i x_{ij} \tag{A.1}$$

We want to find a direction $\mathbf{u}$ that maximizes the variance of the projections of all input vectors $\mathbf{x}_j$, $j = 1, \ldots N$.

It follows that the mean of the projections is

$$\bar{a} = \frac{1}{N} \sum_{j=1}^{N} a_j = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{M} u_i x_{ij} = \sum_{i=1}^{M} u_i \mu_i \tag{A.2}$$

and the variance is[1]:

$$\begin{aligned} \sigma^2 &= \frac{1}{N} \sum_{j=1}^{N} (a_j - \bar{a})^2 = \frac{1}{N} \sum_{j=1}^{N} \left(\sum_{i=1}^{M} u_i x_{ij} - \sum_{i=1}^{M} u_i \mu_i \right)^2 = \\ &= \frac{1}{N} \sum_{j=1}^{N} \left(\sum_{i=1}^{M} u_i \hat{x}_{ij} \right)^2 = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{M} \sum_{l=1}^{M} u_i \hat{x}_{ij} u_l \hat{x}_{lj} = \\ &= \sum_{i=1}^{M} \sum_{l=1}^{M} u_i u_l \frac{1}{N} < \hat{\mathbf{x}}_{i:}, \hat{\mathbf{x}}_{l:} >= \sum_{i=1}^{M} \sum_{j=1}^{M} u_i c_{il} u_l = \mathbf{u}^T \mathbf{C} \mathbf{u} \end{aligned} \tag{A.3}$$

[1] Subscript $\mathbf{x}_i$ denotes i-th *column* vector in the matrix $\mathbf{X}$, while $\mathbf{x}_{i:}$ denotes i-th *row* vector in the matrix $\mathbf{X}$.

Here, μ_i is the mean of the i-th row in the data matrix $\mathbf{X}$ and $\hat{x}_{ij}$ is the value of x_{ij} with subtracted μ_i. If the vector μ contains all row means, thus

$$\mu = [\mu_1, \ldots, \mu_M]^T = \frac{1}{N} \sum_{j=1}^{N} x_j \tag{A.4}$$

then[2]

$$\hat{\mathbf{X}} = \mathbf{X} - \mu \mathbf{1}_{1 \times N} \tag{A.5}$$

and $\mathbf{C}$ is the covariance matrix of $\mathbf{X}$, thus

$$\mathbf{C} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T \tag{A.6}$$

Our goal is to maximize σ^2 under the constraint that $\| \mathbf{u} \| = 1$. Therefore, by using the technique of Lagrange multipliers, we have to maximize the function

$$F(\mathbf{u}; \lambda) = \mathbf{u}^T \mathbf{C} \mathbf{u} - \lambda(\mathbf{u}^T \mathbf{u} - 1) = \sum_{i=1}^{M} \sum_{j=1}^{M} u_i c_{ij} u_j - \lambda \left(\sum_{i=1}^{M} u_i^2 - 1 \right) \tag{A.7}$$

A closed form solution of this maximization problem can be obtained in the following way:

$$\begin{aligned} \frac{\partial F}{\partial u_l} = \sum_{j=1}^{M} c_{lj} u_j + \sum_{i=1}^{M} u_i c_{il} - \lambda 2 u_l &= 0; \quad l = 1 \ldots M \\ \sum_{i=1}^{M} c_{li} u_i &= \lambda u_l; \quad l = 1 \ldots M \\ \mathbf{C}\mathbf{u} &= \lambda \mathbf{u} \end{aligned} \tag{A.8}$$

Therefore, to find $\mathbf{u}$ and λ that maximize (A.7) we have to compute the eigenvectors and the eigenvalues of the covariance matrix $\mathbf{C}$. The largest eigenvalue equals the maximal variance, while the corresponding eigenvector determines the direction with the maximal variance.

By performing *eigenvalue decomposition* (EVD) or *singular value decomposition* (SVD) of the covariance matrix $\mathbf{C}$ we can diagonalize $\mathbf{C}$:

$$\mathbf{C} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \tag{A.9}$$

in such a way that the orthonormal matrix $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N] \in \Re^{M \times N}$ contains the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_N$ in its columns and the diagonal matrix $\Lambda \in \Re^{N \times N}$ contains the eigenvalues $\lambda_1, \ldots, \lambda_N$ on its diagonal. We will assume that the eigenvalues and the corresponding eigenvectors are arranged with respect to the descending order of

[2] $\mathbf{1}_{M \times N}$ denotes a matrix of the dimension $M \times N$, where every element equals 1.

the eigenvalues, thus $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Therefore, the most of the variability of the input random vectors is contained in the first eigenvectors. Hence, the eigenvectors are called *principal vectors* (also *principal axes* or *principal directions*).

This approach to calculation of principal vectors is very clear and widely used. However, if the size of the data vector M is very large, which is often the case in the field of computer vision, the covariance matrix $\mathbf{C} \in \Re^{M \times M}$ (see equation A.6) becomes very large and eigenvalue decomposition of $\mathbf{C}$ becomes unfeasible. If the number of input vectors is smaller than the size of these vectors ($N < M$), PCA can be sped up using the following method proposed by Murakami and Kumar [92].

Instead of the covariance (outer product) matrix $\mathbf{C} \in \Re^{M \times M}$ the inner product matrix $\mathbf{C}' \in \Re^{N \times N}$ (divided by the number of the input vectors) is calculated:

$$\mathbf{C}' = \frac{1}{N} \hat{\mathbf{X}}^T \hat{\mathbf{X}} \tag{A.10}$$

The eigenvalues and the eigenvectors of the covariance matrix $\mathbf{C}$ can then be determined from the eigenvalues λ'_i and eigenvectors $\mathbf{u}'_i$ of the matrix $\mathbf{C}'$ as:

$$\lambda_i = \lambda'_i \tag{A.11}$$

$$\mathbf{u}_i = \frac{\hat{\mathbf{X}} \mathbf{u'}_i}{\sqrt{N \lambda'_i}}, \quad i = 1 \ldots N \tag{A.12}$$

Note that $\mathbf{C}'$ is much smaller than $\mathbf{C}$ when $N \ll M$. Thus, the eigendecomposition of the $M \times M$ matrix $\mathbf{C}$ has been reduced to the much more feasible eigendecomposition of the $N \times N$ matrix $\mathbf{C}'$.

A.2 Properties of PCA

The orthonormal matrix $\mathbf{U}$ containing the principal vectors can serve as a linear transformation matrix for projection from the high-dimensional input space to the low-dimensional feature space and vice versa. The columns of $\mathbf{U}$ are the basis vectors of the new low-dimensional coordinate frame expressed with the high-dimensional coordinates. Thus an input vector can be projected into the principal subspace using the transformation matrix $\mathbf{U}^T : \Re^M \rightarrow \Re^N$.

$$\mathbf{a} = \mathbf{U}^T \hat{\mathbf{x}} \tag{A.13}$$

Thus, the coefficients a_j are computed as the projections of the input image onto each principal vector:

$$a_j = < \hat{\mathbf{x}}, \mathbf{u}_j > = \sum_{i=1}^{M} u_{ij} \hat{x}_i, \quad j = 1 \ldots N \tag{A.14}$$

All the input vectors contained in the input matrix $\hat{\mathbf{X}}$ can thus be projected as $\mathbf{A} = \mathbf{U}^T \hat{\mathbf{X}}$. Since $\mathbf{A}$ is an orthonormal transformation of the mean centered $\hat{\mathbf{X}}$, the principal components are also centered around zero:

$$\mu_{\mathbf{A}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{a}_j = \frac{1}{N} \sum_{j=1}^{N} \mathbf{U}^T \hat{\mathbf{x}} = \mathbf{U}^T \frac{1}{N} \sum_{j=1}^{N} \hat{\mathbf{x}} = \mathbf{0} \tag{A.15}$$

Now, let us calculate the correlation matrix of $\mathbf{A}$:

$$\mathbf{C_A} = \frac{1}{N}\mathbf{A}\mathbf{A}^T = \frac{1}{N}\mathbf{U}^T\hat{\mathbf{X}}(\mathbf{U}^T\hat{\mathbf{X}})^T = \mathbf{U}^T\frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T\mathbf{U} = \\ = \mathbf{U}^T\mathbf{C}\mathbf{U} = \mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{U} = \mathbf{\Lambda} \tag{A.16}$$

Here, we replaced $\mathbf{C}$ with its diagonalized form (A.9) and considered the orthonormality of $\mathbf{U}$ (thus $\mathbf{U}^T\mathbf{U} = \mathbf{I}$). Therefore, the covariance matrix of the transformed data is the diagonal matrix $\mathbf{\Lambda}$, which contains the eigenvalues on its diagonal. This fact has two important implications. First, it proves that the transformed vectors are uncorrelated. Thus the redundancy caused by correlation between the input vectors has been removed. Secondly, it shows that the variance in the direction of the i-th principal axis (the variance of the i-th principal components) is equal to the i-th eigenvalue λ_i, thus $\frac{1}{N}\sum_{j=1}^{N} a_{ij}^2 = \lambda_i$.

An important property of the diagonalization (A.9) is that it preserves the trace of the matrix which is being diagonalized [89]. Since the sum of the diagonal elements of the covariance matrix is the sum of variances of the input vectors, this implies that the total variance of the input data has been preserved and equals the sum of all eigenvalues:

$$\begin{aligned} \mathrm{VAR}(\mathbf{X}) &= \sum_{i=1}^{M}\frac{1}{N}\hat{\mathbf{x}}_{i:}\hat{\mathbf{x}}_{i:}^T = \sum_{i=1}^{M} c_{ii} = \\ &= \sum_{i=1}^{N}\lambda_i = \sum_{i=1}^{N}\frac{1}{N}\mathbf{a}_{i:}\mathbf{a}_{i:}^T = \mathrm{VAR}(\mathbf{A}) \end{aligned} \tag{A.17}$$

Now we will explain how can $\mathbf{U}$ serve as a transformation matrix for projection of the coefficient vector back into the input space. This operation is called *reconstruction*. The coefficient vector $\mathbf{a}$ is reconstructed using the transformation matrix $\mathbf{U} : \Re^N \rightarrow \Re^M$:

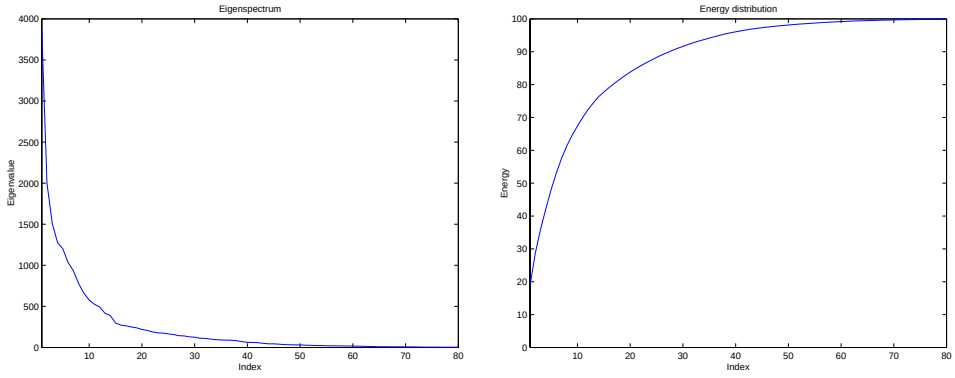
$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{a} = \sum_{j=1}^{N} a_j\mathbf{u}_j \tag{A.18}$$

Since N eigenvectors composing $\mathbf{U} \in \Re^{M\times N}$ span the same subspace in $\Re^M$ as all N input images composing $\mathbf{X} \in \Re^{M\times N}$, each input image from $\mathbf{X}$ can be perfectly reconstructed without any reconstruction error. What is more interesting to us, is how well an input image is reconstructed from a subset of principal components only.

To realize this, we first consider how the variance is distributed across the principal axes. This distribution is called the *eigenspectrum* and it is practically a plot of eigenvalues sorted in decreasing order. A typical eigenspectrum is depicted in figure (A.1). As one can observe, most of the variance is contained across the first few eigenvectors. This can also be measured with *energy*, which is defined as a fraction of the total variance. The energy contained in the first k eigenvectors can thus be calculated as

$$en_k = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{N}\lambda_i} \tag{A.19}$$

The energy plot obtained from the eigenvalues depicted in figure (A.1.a) is shown in figure (A.1.b). Again, it is evident that most of the energy is contained in a first few eigenvectors already.



(a) (b)

Figure A.1: Typical (a) eigenspectrum, (b) energy distribution.

From this we can conclude that we can obtain a good approximation of the input images by considering only a subset of eigenvectors associated with the largest eigenvalues. Therefore, from now on, we will consider only k, $k \ll N$, principal axes, thus $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \Re^{M \times k}$.

Now, an input vector is projected into the k-dimensional principal subspace using the transformation matrix $\mathbf{U}^T : \Re^M \rightarrow \Re^k$:

$$\begin{aligned} \mathbf{a} &= \mathbf{U^T}\hat{\mathbf{x}} = \mathbf{U^T}(\mathbf{x} - \mu) \\ a_j &=< \hat{\mathbf{x}}, \mathbf{u}_j >= \sum_{i=1}^{M} u_{ij}\hat{x}_i = \sum_{i=1}^{M} u_{ij}(x_i - \mu_i), \qquad j = 1 \ldots k \end{aligned} \tag{A.20}$$

and reconstructed using the transformation matrix $\mathbf{U} : \Re^{\mathbf{k}} \rightarrow \Re^{\mathbf{M}}$:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{U}\mathbf{a} = \sum_{j=1}^{k} a_j \mathbf{u}_j \\ \mathbf{y} &= \hat{\mathbf{y}} + \mu \end{aligned} \tag{A.21}$$

Thus, an input image is approximated with a linear combination of the first k principal vectors.

The reconstruction error (residual error) is equal to the difference between the input and the reconstructed vector:

$$\mathbf{e} = \hat{\mathbf{x}} - \hat{\mathbf{y}} = \sum_{j=1}^{N} a_j \mathbf{u}_j - \sum_{j=1}^{k} a_j \mathbf{u}_j = \sum_{j=k+1}^{N} a_j \mathbf{u}_j \tag{A.22}$$

The most commonly used error measure is the squared reconstruction error, which is defined as a square of the length of the residuum. Considering the orthonormality

of the eigenvectors $\mathbf{u}_j$ we obtain:

$$e = \|\mathbf{e}\|^2 = \| \sum_{j=k+1}^{N} a_j \mathbf{u}_j \|^2 = \sum_{j=k+1}^{N} a_j^2 \tag{A.23}$$

Thus, the squared reconstruction error is equal to the sum of squared discarded principal components. Since they are usually not known, the expected error can be approximated with expected values of the variance accross the discarded eigenvectors, which are equal to the corresponding eigenvalues:

$$E(e) = \sum_{j=k+1}^{N} \lambda_j \tag{A.24}$$

The expected error is thus equal to the sum of the discarded eigenvalues. This consideration confirms the fact that by maximizing the variance in the first (non-discarded) eigenvectors, the squared reconstruction error is being simultaneously minimized. These two assertions are indeed two main properties of PCA.

Therefore, for a given dimension of a subspace k, PCA finds such principal vectors $\mathbf{u}_l, l = 1 \ldots k$ and coefficient vectors $\mathbf{a}_j \in \Re^k, j = 1 \ldots N$ that minimize the total squared reconstruction error

$$e = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\hat{x}_{ij} - \sum_{l=1}^{k} u_{il} a_{lj} \right)^2 \tag{A.25}$$

Thus, as an alternative to the maximization of the variance, the principal vectors and the principal components can be estimated by minimizing the squared reconstruction error of equation (A.25). This is a nonlinear minimization problem and can be solved using several proposed algorithms for such a task, e.g., gradient descend algorithm [36] or neural networks [39]. Alternatively, the minimization can be performed by iterating the two-step procedure where first the coefficients are estimated and then the principal vectors are computed. Such an algorithm, which was derived from the probabilistic point of view, can be found in [114].

A.3 Gaussian Interpretation of PCA

In the computer vision literature one can find an interesting work [90] where a direct connection between Principal Component Analysis (PCA) and the Distance From Feature Space (DFFS) assuming that the principal subspace can be represented with a Gaussian distribution. As said before, the principal components obtained using PCA preserve the major linear correlations in the data and discard the minor ones. As noted in [90], there are two mutually exclusive and complementary subspaces: the principal subspace (or feature space) that contains the principal components and its orthogonal complementary subspace (the one that is spanned with the discarded components).

As shown in expression (A.23) and (A.25), the residual reconstruction error is defined as

$$e^2(\mathbf{x}) = \sum_{j=k+1}^{N} a_j^2 = \| \hat{\mathbf{x}} \|^2 - \sum_{i=1}^{k} a_i^2 \tag{A.26}$$

and can be easily computed from the first k principal components and the L_2 norm of the mean-normalized data vector $\hat{\mathbf{x}}$. The component of $\mathbf{x}$ which lies *in* the feature space F is referred to as the "distance-in-features-space" (DIFS) but is generally not a distance-based norm, but can be interpreted in terms of the probability distribution of the eigenvectors in the feature space. The orthogonal subspace ($\bar{F}$) is also called "distance-from-feature-space" (DFFS) which is a simple euclidean distance and is equivalent to the residual error $e(x)$ in equation (A.26)

An optimal approach for estimating high-dimensional Gaussian densities is to firstly assume that we have robustly estimated the mean $\bar{\mathbf{x}}$ and covariance Σ of the distribution from a given training set. Under this assumption, the likelihood of an input pattern $\mathbf{x}$ is given by

$$P(\mathbf{x}|\Omega) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right]}{(2\pi)^{N/2}|\Sigma|^{1/2}} \tag{A.27}$$

Assuming this Gaussian distribution, the only sufficient statistic for characterizing this likelihood is the *Mahalanobis* distance

$$d(\mathbf{x}) = \hat{\mathbf{x}}^T\Sigma^{-1}\hat{\mathbf{x}} \tag{A.28}$$

where $\hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$. Taking in mind that Σ is the covariance matrix of our distribution, we can use the eigenvalues and eigenvectors of Σ and rewrite Σ^{-1} in a diagonalized form

$$\begin{aligned} d(\mathbf{x}) &= \hat{\mathbf{x}}\Sigma^{-1}\hat{\mathbf{x}} && \text{(A.29)} \\ &= \hat{\mathbf{x}}^T[\mathbf{\Phi}\mathbf{\Lambda}^{-1}\mathbf{\Phi}^T]\hat{\mathbf{x}} && \text{(A.30)} \\ &= \mathbf{a}^T\mathbf{\Lambda}^{-1}\mathbf{a} && \text{(A.31)} \end{aligned}$$

where $\mathbf{a} = \mathbf{\Phi}^\mathbf{T}\hat{\mathbf{x}}$ are the new variables. In other words, a are the eigenvectors of the covariance matrix Σ. So that, the *Mahalanobis* distance can also be expressed in terms of the following sum

$$d(\mathbf{x}) = \sum_{i=1}^{N} \frac{a_i}{\lambda_i} \tag{A.32}$$

Usually, when dealing with high-dimensional data vectors, expression (A.32) is computationally infeasible. We therefore seek to estimate $d(\mathbf{x})$ using only k projections. Intuitively, an obvious choice for a lower-dimensional representation is the principal subspace indicated by PCA which captures the major degrees of statistical variability in the data. Therefore, one can divide the summation into two independent parts corresponding to the principal subspace (the first k projections) and its orthogonal complement subspace (the remaining projections):

$$d(\mathbf{x}) = \sum_{i=1}^{k} \frac{a_i^2}{\lambda_i} + \sum_{i=k+1}^{N} \frac{a_i^2}{\lambda_i} \tag{A.33}$$

In the first summation can be computed by projecting $\mathbf{x}$ onto the k-dimensional principal subspace F but the remaining terms in the second summation, however, can not be computed explicitly in practice because of the high-dimensionality of data. However, the *sum* of these terms is available and is in fact the DFFS quantity $e^2(\mathbf{x})$ which can be computed from expression (A.26). Therefore, based on the available terms, we can formulate an estimator for $d(\mathbf{x})$ as follows

$$\begin{aligned} \hat{d}(\mathbf{x}) &= \sum_{i=1}^{k} \frac{a_i^2}{\lambda_i} + \frac{1}{\rho}\left(\sum_{i=k+1}^{N} a_i^2\right) \\ &= \sum_{i=1}^{k} \frac{a_i^2}{\lambda_i} + \frac{e^2(\mathbf{x})}{\rho} \end{aligned} \tag{A.34}$$

where the term in the brackets is $e^2(\mathbf{x})$, which as we have seen can be computed using the first k principal components. So that, one can write the form of the likelihood based on $\hat{d}(\mathbf{x})$ as the product of two marginal and independent Gaussian densities,

$$\begin{aligned} \hat{P}(\mathbf{x}|\Omega) &= \left(\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{k}\frac{a_i^2}{\lambda_i}\right)}{(2\pi)^{k/2}\prod_{i=1}^{k}\lambda_i^{1/2}}\right)\cdot\left(\frac{\exp\left(-\frac{e^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-k)/2}}\right) \\ &= P_F(\mathbf{x}|\Omega)\hat{P}_{\bar{F}}(\mathbf{x}|\Omega) \end{aligned} \tag{A.35}$$

where $P_F(\mathbf{x}|\Omega)$ is the true marginal density in F-space and $\hat{P}_{\bar{F}}(\mathbf{x}|\Omega)$ is the estimated marginal density in the orthogonal complement $\bar{F}$-space. Then, the optimal value of ρ can be determined by minimizing a suitable cost function $J(\rho)$. See [90] for more information about this minimizing cost function. The optimal weight ρ^* is determined by [90]

$$\rho^* = \frac{1}{N-k}\sum_{i=k+1}^{N}\lambda_i \tag{A.36}$$

which is simply the arithmetic average of the eigenvalues in the orthogonal subspace $\bar{F}$. Then, we can conclude stating that once we select the k-dimensional principal subspace F (as indicated, for example, by PCA), the optimal estimate of the sufficient statistic $\hat{d}(\mathbf{x})$ has the form of expression (A.34) with ρ given by expression (A.36).

One of the typical behaviours in most of the applications it is to simply discard the $\bar{F}$-space component and simply work with $P_F(\mathbf{x}|\Omega)$. However, the use of the DFFS metric or equivalently the marginal density $P_{\bar{F}}(\mathbf{x}|\Omega)$ is critically imporntant in formulatin the likelihood of an observation $\mathbf{x}$ since there are an infinity of vectors which are not members of Ω which can have likely F-space projections.

A.4 Applications of PCA

The most famous application of PCA is the dimensionality reduction in order to formulate the same problem but without noise or with less noise in the data. A

frequently chosen criterion to decide over the dimensionality consists of thresholding the distribution of energy (see figure (A.1.b)) in order to preserve a fixed percentage of the variation. Typical percentages are above 90%. Observing figure (A.1.a) we can see that small variances in the data provided by small eigenvalues are often associated to noise. So, under certain simple assumptions, we can state that PCA reduces noise as well as dimension. Also, notice what happens when there exist eigenvalues taking zero values. Dropping their corresponding eigenvectors from the PCA bases has no effect on the mean square error or, what is equivalent, preserves 100% of the data variation, meaning that dimensionality is reduced and no information is lost.

Another application of PCA is data whitening. To whiten or sphere the data is to transform the data linearly so the components of the transformed vector are uncorrelated and have unit variance. The term 'white' comes from the fact that the power spectrum of white noise is constant over all frequencies, resembling the spectrum of white light which contains all colors. Whitening is frequently used as a preprocessing stage, since it can provide invariance to displacement and scale changes. Since PCA uncorrelates the data, one of its applications is to perform whitening. It is also known that statistical independence implies uncorrelation, so PCA can also be understood as one step towards a representation that yields statistically independent components [17]. If the data is Gaussian, the resulting components are in effect independent with unidimensional Gaussian distributions.

Nevertheless PCA is a powerful and simple technique we should not forget the natural limitations derived from its definition, mainly the fact that PCA fails to distinguish high order relationships between the data. This fact should be considered when using PCA as a dimensionality reduction technique previous to classification. Nevertheless, PCA performs successfully in several problems. Two reasons account for this achievement: the first is that in practice it is not unfrequent to find noise in the directions of small variance, the second is that classes are frequently found to be distributed along the main directions of variance.