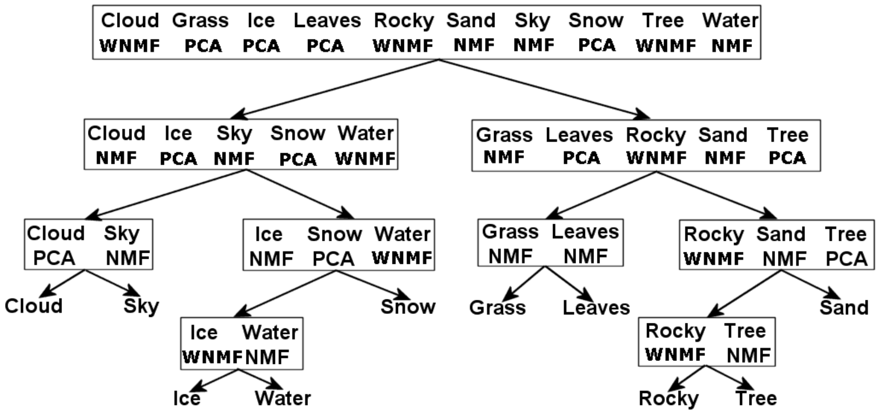


Figure 3.12: Optimal hierarchical representation that classifies our original 10 data classes. Given a node in the tree and having a local color histogram to classify between the right or the left leaves, we obtain the reconstruction distance of this vector and we choose the leaf that contains the minimal distance. Under the name of each data class, we have the name of the technique used to represent the data.

one can think of. That is, we classify a data vector only analyzing the class labels of the k nearest neighbors. So that, k-nearest neighbors approach induces to use a distance. However, the distance of choice will affect to the shape of the neighborhood, for instance, choosing the Euclidean metric is equivalent to choosing hyperspheric neighborhoods, the L^{∞} metric is equivalent to choosing hypercubic neighborhoods, etc.

In this section, in contrast to the two previous ones, we want to work with the projected coefficients of both PCA and NMF techniques. Taking as a reference the projected vectors obtained with PCA and NMF, we will perform k-nearest neighbors in order to classify new unseen data vectors. PCA has been widely analyzed in this context. Usually, one has to choose L_2 metric distance in the projected space to perform classification. And as known a priori, this should be the best metric distance to be used with PCA since it assumes data gaussianity. However, NMF has not an associated metric distance to be used with its nonnegative coefficients. So that, this motivates to explore which metric distance can be used in combination with NMF.

Firstly, we chose the MNIST digit database [76] in order to evaluate different metric distances with NMF. This digit database has been previously used with PCA (see results in `http://yann.lecun.com/exdb/mnist/`). Furthermore, it is a widely used database with a huge number of training and testing digits. So that, in order to provide statistical evidence of our results, we find desirable to test NMF in such a scenario. We should note that this experiment evaluates PCA and NMF in their respective subspaces but both representations are global. For such reason, we do not test the weighted version of NMF. In this framework, NMF has been compared to PCA with the main goal to find a good metric to be used with NMF in its positive subspace.

Secondly, we tested PCA and NMF in a common application, the recognition of faces. PCA has been widely used in such scenario a [7, 142] and it seems natural to compare PCA versus NMF in this scenario. Furthermore, we have to take into account that NMF has been presented to the scientific community as a technique that can extract features from images. And more particularly, the proposed example where the NMF has been tested was an image database of faces. Again, faces were represented using a global approach (each face is a data vector). So that, it makes no sense to apply the WNMF technique. Also, with this experiment we want to show the different performances that we can obtain with NMF when we use a variety of subspace metrics.

Choosing a Metric Distance for NMF

NMF can be seen as a unsupervised feature extraction technique. In fact, it has been discovered in order to extract features from data [77]. Here, we want to use NMF as another alternative reduction technique that can be used for classification. The problem here is that PCA has a natural metric to be used in the subspace, the L_2 metric. But we have to define a metric distance between two projected vectors obtained using NMF. Up to now, it does not exist any metric distance that can be used properly with the projected vectors obtained by NMF. Since NMF is a recent technique, it does not provide a natural metric to work with its positive projected vectors. Is for this reason that a metric distance must be defined or chosen in the positive subspace described by NMF in order to work in an optimal manner. This section presents experimental evaluations of traditional distance measures in the context of digit recognition when using PCA and NMF. We have selected the MNIST digit database [76] because it is a well-known database with a huge number of training and testing vectors. So that, reliable statistics about the performance of NMF can be extracted. We have to note that the aim of this comparison is not to obtain the best classifier of the MNIST digit database. With this current analysis, we want to show that it is possible to define a metric distance when we use NMF. Also, we want to show that we can use NMF and obtain better classification results with respect to PCA.

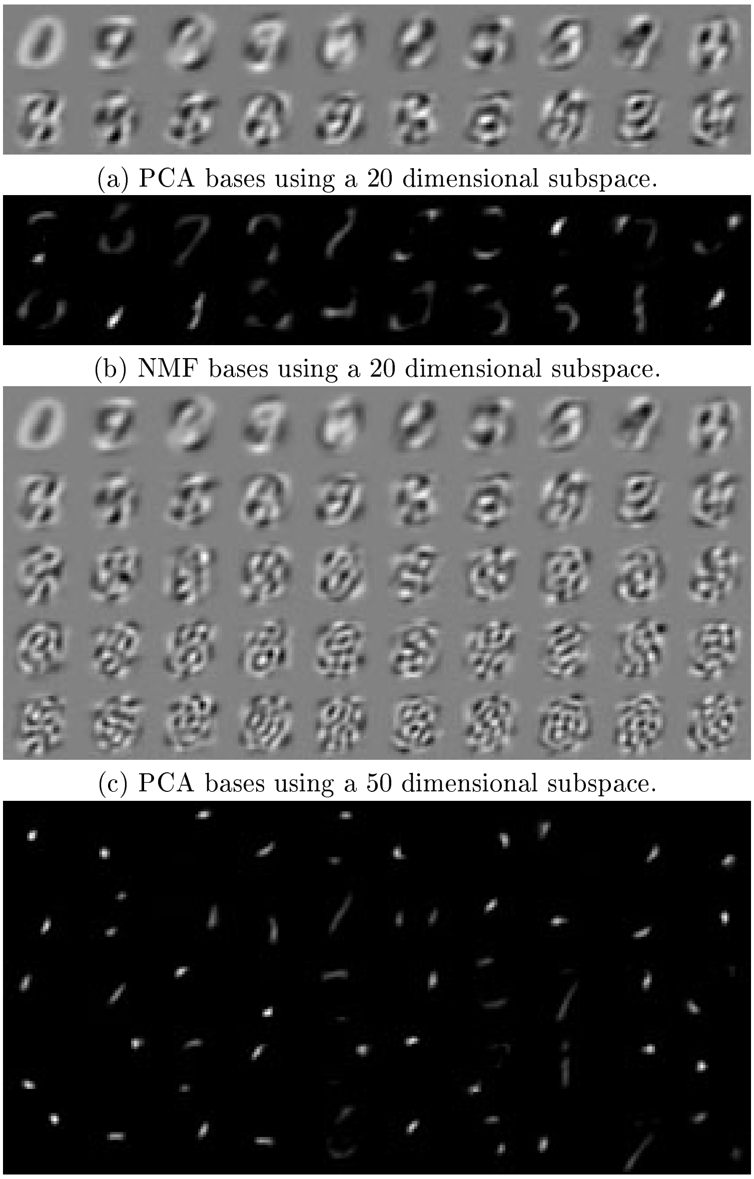
There are several methods that have been tested with the MNIST digit database and most of them are based on preprocessing the input images in order to reduce some distortion effects. In our case, we have used the original 28×28 images without any modification. We have randomly selected $4,000$ training vectors (400 of each digit) to learn the PCA and NMF models. The reason of selecting only $4,000$ training vectors instead of a large number of them is because NMF needs to work with matrices of size $4,000 \times 784 \times 8 = 24$ Mb ($784 = 28 \times 28$ and 8 is the precision needed by a double number). We have estimated that having $4,000$ training vectors to obtain our NMF models is a good trade-off between time of calculation and accuracy in results. In section (C.4) we can find some examples of the MNIST digit database and we will see that some of them are complex to be identified even for us.

We have learned both PCA and NMF models and we have obtained a set of bases that can be seen in figure (3.13). Figure (3.13) shows the bases obtained if we decide to work with a 20 dimensional subspace and a 50 dimensional subspace. As we see in this figure, the main difference between PCA bases and NMF bases is that NMF is a parts-based representation and PCA a global one. The NMF bases of the 50 dimensional subspace depict a sparse set of pixels. As we see, nearly all the NMF bases contain localized parts of digits. When the subspace is described by only 20 bases, the NMF bases do not depict this behaviour. The obtained bases are agrupations of different parts. Up to this point, we can think that the dimension of the subspace is crucial in order to obtain a parts-based representation or not. The other interesting thing to appreciate from these figures is the fact that when we increment the subspace dimension from 20 to 50, PCA holds the same initial bases. However, when we increment the subspace dimension for NMF, its bases change from one subspace to the other. The NMF bases obtained in the 20 dimensional subspace have nothing to do with the ones obtained in the 50 dimensional one.

Analyzing more carefully all the NMF bases obtained in figure (3.13), we see that some of them share common pixels. Figure (3.13.d) shows sparse bases and it seems that they are independent. Here we say independent in the sense that they do not share pixels. However, taking a look to some of the bases of figure (3.13) we see that some of them are very similar or contain a high degree of correlation. As an example, two of the bases that contain similar pixels are the bases shown in figure (3.14). The previous presented weighted non-negative matrix factorization (WNMF) reduced the effect of repeated information of the **W** bases. Here, as we see, some of the obtained bases contain repeated information (figure (3.14)) and it can be a good problem where we can test the weighted version of NMF. So that, the learned bases can present non-negligible correlations or other higher order effects. However, in order to take profit of the correlation between different bases, we want to find out a metric distance that can be adapted to such a problem.

In section (2.1.3) we explained the Earth Mover's Distance (EMD). This distance is interesting because we can define a cost matrix between each component of the input vectors. This means that if we have a 50 dimensional space where we know that some of the components are very correlated between them, we can define a cost matrix that reflects this level of correlation. So that, we can use a cost matrix in conjunction with EMD to evaluate the distance between two correlated vectors and we should be able to compare these two vectors more efficiently than using a traditional distance measure. EMD is well suited to this problem because we can explicitly define a distance between our NMF bases and create a cost matrix used in the minimization problem of expression (2.58).

After learning our PCA and NMF models as explained before, we present the experimental results with the testing MNIST database of $10,000$ images. Once we have our learned models, we project the training database of $60,000$ images and we store these projections. After this, given a new unseen digit image of our testing database, we obtain its projection and we search for its nearest neighbors using the projections of the training vectors. So that, the metric distance is an important clue to find a correct identity for new unseen digit images. We used the well-known L_2 metric distance with Principal Component Analysis (PCA) (that we already know to be the optimal metric distance to be used with it) as well as L_1 and cosine metrics. We used L_2, L_1, cosine and the earth mover's distances with NMF. The cost matrix that must be defined when we use EMD is created as follows: given a basis of matrix $\mathbf{W}$ (b_i) and another basis of matrix $\mathbf{W}$ (b_j), we define the distance between these two bases as $d_{ij} = 1/\text{dist}_{\text{COS}}(b_i, b_j)$. We used the correlation (or cosine) metric between



(a) PCA bases using a 20 dimensional subspace.

(b) NMF bases using a 20 dimensional subspace.

(c) PCA bases using a 50 dimensional subspace.

(d) NMF bases using a 50 dimensional subspace.

Figure 3.13: Different set of bases obtained using PCA and NMF in two dimensional subspaces, 20 and 50. We can appreciate the parts-based representation of NMF and the holistic representation of PCA.

two bases because we are evaluating how correlated are two bases. As we suggested before, NMF can generate correlated bases with some pixels in common. So that, distance d_{ij} represents the following idea: When two bases are correlated, the cost

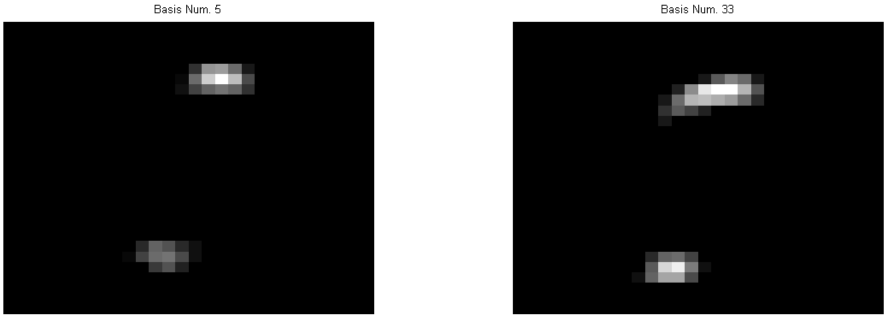


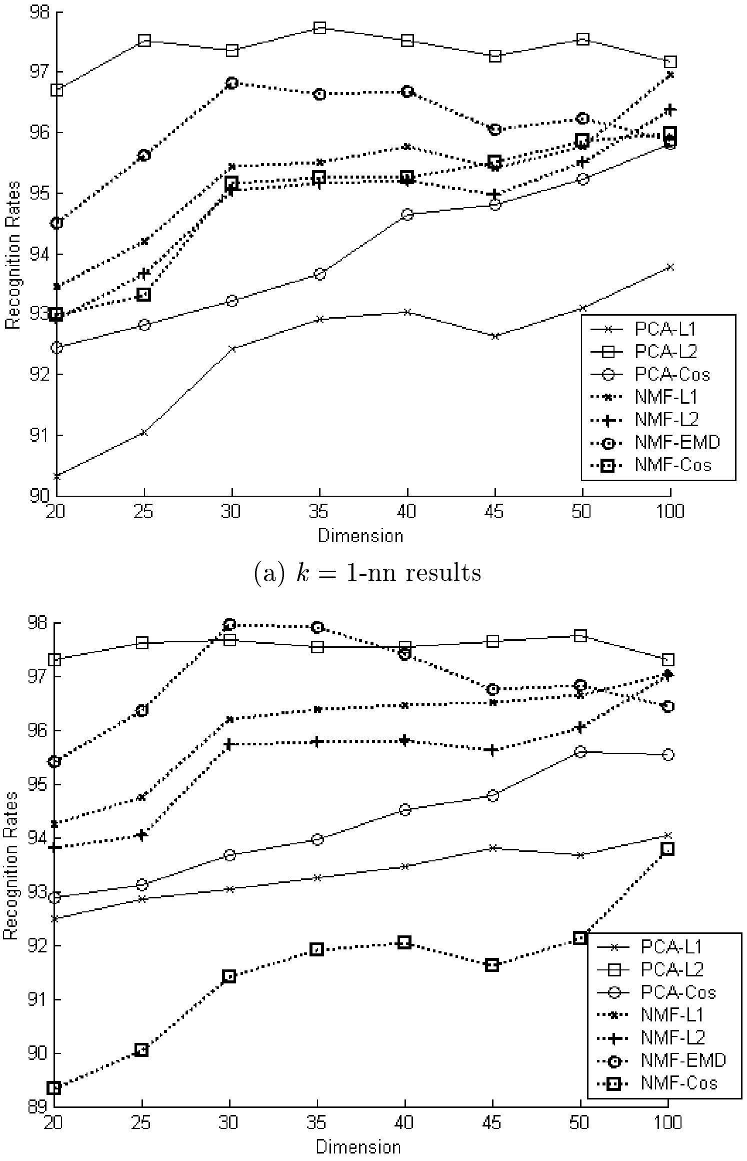
Figure 3.14: 2 different bases obtained with NMF in the 50 dimensional subspace described in figure (3.13.d). As seen, these two bases share some spatial pixels.

associated to them should be lower than the case of two uncorrelated bases. We have to note that PCA has not been tested with the EMD metric because EMD assumes positive representations (see expression (2.54)) and PCA does not provide such a representation. All techniques have been tested using two k nearest neighbor classifiers ($k = 1$ and $k = 5$). Table (3.22) shows the recognition rates for this experiment where we can see different metric distances in conjunction with PCA and NMF. In order to have a visual idea of which is the best combination of metric distance and technique, we show figure (3.15).

Method	20D	25D	30D	35D	40D	45D	50D	100D
PCA + L1 1nn	90.32	91.04	92.42	92.92	93.04	92.63	93.11	93.78
PCA + L1 5nn	92.48	92.86	93.03	93.25	93.45	93.81	93.67	94.03
PCA + L2 1nn	96.70	97.52	97.36	97.72	97.51	97.26	97.54	97.17
PCA + L2 5nn	97.29	97.63	97.68	97.54	97.53	97.64	97.75	97.29
PCA + Cos 1nn	92.45	92.81	93.21	93.65	94.65	94.81	95.23	95.82
PCA + Cos 5nn	92.87	93.13	93.67	93.97	94.51	94.78	95.59	95.53
NMF + L1 1nn	93.46	94.21	95.43	95.51	95.76	95.42	95.77	96.95
NMF + L1 5nn	94.25	94.76	96.21	96.37	96.45	96.52	96.65	97.03
NMF + L2 1nn	92.91	93.65	95.03	95.15	95.21	94.97	95.51	96.38
NMF + L2 5nn	93.81	94.04	95.73	95.77	95.79	95.61	96.05	97.02
NMF + EMD 1nn	94.51	95.62	96.82	96.62	96.67	96.05	96.23	95.87
NMF + EMD 5nn	95.42	96.36	97.97	97.92	97.41	96.76	96.83	96.43
NMF + Cos 1nn	92.98	93.32	95.15	95.25	95.24	95.52	95.87	95.98
NMF + Cos 5nn	89.32	90.04	91.42	91.92	92.04	91.63	92.11	93.78

Table 3.22: Recognition rates using the MNIST database with the PCA and NMF techniques using several metric distances (L_1, L_2, Cos and EMD). We also used a $k = 1$ and $k = 5$ nearest neighbor classifiers.

Figure (3.15) shows all the recognition rates of all possible combinations between metric distances and techniques (PCA or NMF) against the dimensionality of the subspace. As seen in figure (3.15), the combination of PCA with L_2 provides the best recognition rates when a $k = 1$ nearest neighbor classifier is used. And, as expected, the combination between PCA and L_1 or the cosine metrics results in a



(a) k = 1-nn results

(b) k = 5-nn results

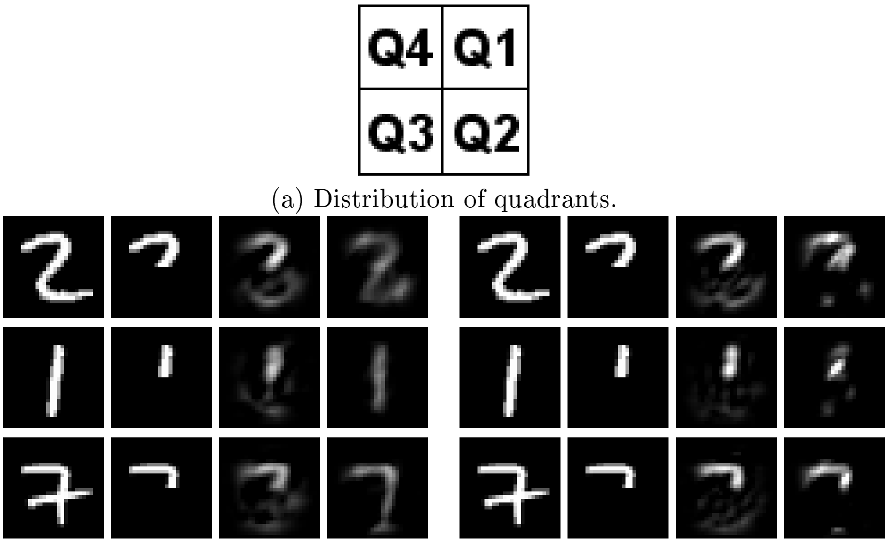
Figure 3.15: Recognition rates using the MNIST database with PCA and NMF techniques. This graphical representation of results corresponds to table (3.22). We analyzed all subspace dimensions of 20, 25, 30, 35, 40, 45, 50 and 100. (a) is the graphical result of using a $k = 1$ nearest neighbor classifier, (b) is the graphical result of using a $k = 5$ nearest neighbor classifier.

bad combination because the recognition rates are very poor. In the case of using NMF, the EMD is the best metric distance to use with it since we obtain the best recognition rates in front of L_1, L_2 and the cosine distance. Furthermore, there are two particular cases where $NMF + EMD$ outperforms $PCA + L_2$ when using a $k = 5$ nearest neighbor classifier. It is interesting to note that the performance of $NMF + EMD$ decreases according to the dimensionality of the subspace. It seems that when a high-dimensional subspace is used, EMD is not a good metric distance. But this fact is easy to understand because a high-dimensional subspace generates a set of NMF bases which do not contain correlations between their pixels. That is, the bases do not share common pixels because they tend to be non-correlated. Thus, we can conclude stating that when no occlusions are present in our digit images, EMD is the best metric distance to use in conjunction with NMF among all possible distances (L_1, L_2, EMD, Cosine) because it obtains the best recognition rates and, in some particular dimensions, is better than $PCA + L_2$. If we have to decide the best combination of method (PCA or NMF) and metric distance to be used in the subspace when occlusions are not present, we have to choose $PCA + L_2$ in front of $NMF + EMD$ because $PCA + L_2$ is slightly better.

Previous experiment compares the performance obtained using PCA and NMF and the combination of different metric distances in the subspace of both projection techniques. It seems clear that the introduction of NMF does not provide a real improvement over the traditional PCA since the combination of NMF with EMD improves the recognition rates of $PCA + L_2$ in only two particular cases: when using a subspace of 30 and 35 dimensions. Since NMF is a parts-based representation and PCA a holistic one, occlusions should be a good test to appreciate how significant could be to use NMF in a classification framework. We have considered two levels of occlusions in our experiments. According to the distribution of quadrants observed in figure (3.16.a), we have occluded our testing images using one quadrant (25% of the area of one digit) and two quadrants (50% of the are of one digit). Figure (3.16) contains three examples of digit reconstructions using two different dimensional subspaces (20 and 50 dimensions) using PCA and NMF. As seen in this figure, PCA always introduces noise in the reconstruction images because it is a global technique even if we are working on a low or high dimensional subspace.

Recognition rates under the presence of different levels of occlusions are shown in tables (3.23,3.24). We also present a visual representation of this table in order to have a quick notion of our results. Under the presence of a 25% degree of occlusion, our results demonstrate that $NMF + Cos$ is better than $PCA + L_2$. EMD is also a good candidate distance to be used in conjunction with NMF because it is better than $PCA + L_2$. But we have to note that this behaviour is true only in a low dimensional subspace (20 dimensional subspace). When the testing digits present a 50% level of occlusion, differences between performances of $PCA + L_2$ and $NMF + Cos$ are also present. Now, with this huge level of occlusion, $NMF + Cos$ is really the best combination in front of $PCA + L_2$. With a 50% of occlusion, EMD can also be used with NMF, but only in low dimensional subspaces (20 dimensions).

We have presented NMF as an alternative technique to PCA but, as seen, if we want to classify vectors using their projections, we have to be very careful when we select an appropiate metric distance. As known from previous studies, L_2 is



(a) Distribution of quadrants.

(b) 20D space

(c) 50D space

Figure 3.16: Reconstruction examples with occlusions with PCA and NMF in two different dimensional subspaces. Original digit is the first column, the second column reflects the occlusion applied to each digit, third column reflects the PCA reconstruction and last column the NMF reconstruction.

Method	Q1	Q2	Q3	Q4	Q1+Q2	Q2+Q3	Q3+Q4	Q4+Q1
PCA + L1 1nn	85.13	83.32	81.56	83.51	47.19	55.06	50.28	52.83
PCA + L1 5nn	88.71	86.11	84.06	84.93	52.81	59.01	54.08	55.23
PCA + L2 1nn	87.92	86.12	84.47	86.82	54.02	60.93	55.43	57.32
PCA + L2 5nn	90.73	89.37	87.81	89.74	57.38	62.39	59.47	61.62
PCA + Cos 1nn	83.41	82.28	80.91	81.17	45.82	54.85	51.65	51.71
PCA + Cos 5nn	85.03	83.41	83.16	83.01	47.03	56.20	52.18	53.12
NMF + L1 1nn	85.82	83.58	82.23	84.53	51.97	57.67	52.31	55.04
NMF + L1 5nn	88.85	86.92	84.51	88.14	55.84	60.12	56.79	58.37
NMF + L2 1nn	84.66	82.49	80.31	82.35	48.82	56.94	49.42	52.63
NMF + L2 5nn	87.37	85.21	83.92	85.61	53.09	58.42	53.12	54.41
NMF + EMD 1nn	90.63	88.93	85.84	86.52	55.73	61.41	56.04	57.91
NMF + EMD 5nn	92.53	91.23	90.63	90.38	61.42	64.18	61.74	63.74
NMF + Cos 1nn	91.94	90.35	87.42	87.94	57.21	62.64	59.47	59.73
NMF + Cos 5nn	93.14	91.42	90.73	91.21	62.37	66.14	65.64	67.42

Table 3.23: Results with occlusions in a 20D subspace.

the best metric distance to be used with PCA since PCA obtains an optimal linear dimensionality reduction scheme with respect to the mean squared error (MSE). When occlusions are not present in the testing data images, NMF has some difficulties to outperform the combination between PCA and L_2 even using a wide variety of metric distances. However, it seems clear that EMD is the best choice to be used with NMF among all the possible metric distances. However, when occlusions are present in our images, $PCA + L_2$ decreases its performance because it is a global technique and NMF takes advantage of its local behaviour. Under the presence of occlusions, EMD is a good choice when the dimensionality of the subspace created by NMF is low (i.e.

Method	Q1	Q2	Q3	Q4	Q1+Q2	Q2+Q3	Q3+Q4	Q4+Q1
PCA + L1 1nn	82.39	81.75	80.21	82.49	55.21	57.19	56.32	49.29
PCA + L1 5nn	83.71	84.13	82.39	83.71	56.77	58.61	57.72	51.91
PCA + L2 1nn	92.36	89.74	87.41	91.38	58.01	65.08	59.31	60.52
PCA + L2 5nn	93.75	91.87	89.63	92.58	62.53	67.34	63.52	63.43
PCA + L2 1nn	81.91	80.32	80.11	81.98	53.16	54.91	55.27	50.91
PCA + L2 5nn	82.16	83.39	82.92	82.19	54.72	57.11	57.89	52.87
NMF + L1 1nn	87.12	83.32	82.24	85.78	54.32	55.72	53.16	47.92
NMF + L1 5nn	89.52	84.65	84.23	87.17	56.16	57.81	54.68	52.87
NMF + L2 1nn	84.83	83.18	81.32	83.84	56.62	57.31	57.43	51.32
NMF + L2 5nn	86.37	85.48	83.23	86.67	59.42	58.97	60.83	56.76
NMF + EMD 1nn	85.42	81.41	81.32	82.47	52.54	56.36	51.65	48.27
NMF + EMD 5nn	88.31	83.68	84.14	84.39	56.03	57.12	53.82	49.83
NMF + Cos 1nn	91.52	88.81	86.53	87.31	66.76	64.73	70.65	60.08
NMF + Cos 5nn	94.21	91.03	91.32	90.31	69.63	68.52	72.39	66.22

Table 3.24: Results with occlusions in a 50D subspace.

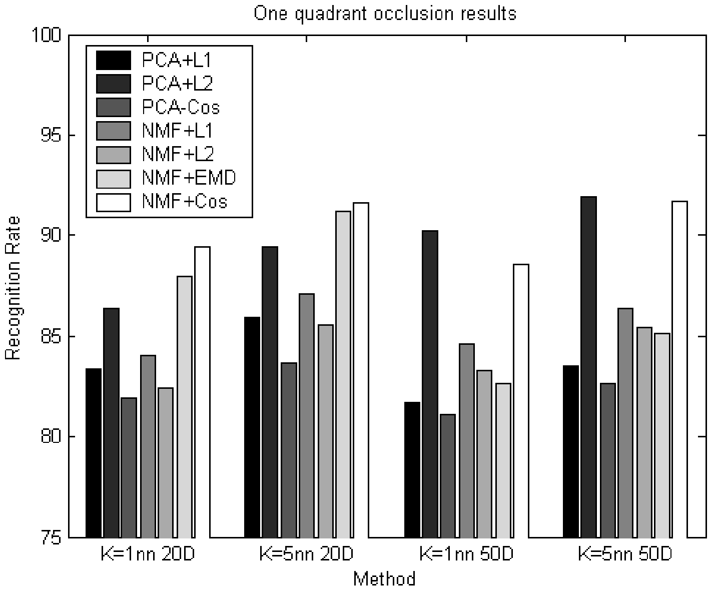
20 dimensions) because it outperforms $PCA + L_2$. But, it is interesting to see that the worst combination of NMF and a metric distance ($NMF + Cos$) results into the best combination when occlusions are introduced. Surprisingly, $NMF+Cos$ is always better than the other combinations of methods and metric distances.

In terms of computational costs, PCA is a fast technique when we project new unseen data samples. However, NMF is based on an iterative process. Furthermore, EMD requires a minimization process to obtain the optimal distance between two positive vectors. So that, the combination of NMF and EMD results in a good scheme but the computational resources are very demanding. Thus, when no occlusions are present in our data images, it is clear that we have to keep using the traditional $PCA+L_2$ and when occlusions are present, we can choose $NMF + Cos$ or $NMF + EMD$. But from our results, it seems clear that $NMF + Cos$ is faster than $NMF + EMD$.

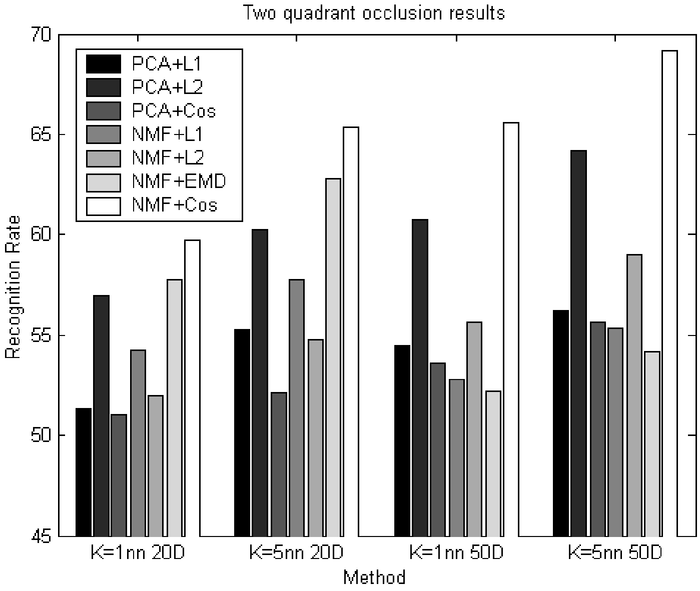
Recognition of Faces using NMF

In [79], Lee and Seung presented the NMF technique to the computer vision community. They stated that is a parts-based representation because it is able to extract parts of objects without any kind of supervision. Their objects were face images, so that, the parts that they were able to extract were eyes, nose, etc. It is interesting to see that even they presented this technique to extract parts of face images, they do not use this technique for face classification. It is natural to think that maybe such a representation is better than the traditional PCA or maybe it is more adapted for this specific problem.

In the previous section, we evaluated a set of metric distances to be used with NMF. We found that EMD can be used with NMF but it requires a huge amount of computational resources. Furthermore, in the previous section, we found that we can obtain parts of digits and it seems that, under the presence of occlusions, NMF can be a good solution in terms of recognition rates. The occlusions dealed in the previous section were artificial and handmaded and it is clear that this kind of occlusions can not be found in real life situations. It is for this specific reason that we present some experimental results with a face database containing different kinds of real occlusions. We evaluate the performances of PCA and NMF techniques in a face classification framework.



(a) One quadrant occlusion results



(b)Two quadrant occlusion results

Figure 3.17: Recognition rates with different levels occlusions using two different subspaces (20 and 50 dimensions) and using two $k - nn$ nearest neighbor classifiers ($k = 1$ and $k = 5$).

Face recognition and classification is one of the most challenging problems to be solved in the computer vision community and can be seen under different perspectives. Until now, several methods and sophisticated approaches have been developed in order to obtain the best recognition results using some specific face databases. Due to this huge number of methods and face databases, there is no uniform way to establish the best method because nearly all of them have been designed to work with some specific face poses. Even though, some of these methodologies have lead to the development of a great number of commercial face recognition systems. Most of the face recognition algorithms can be classified into two classes, image template based or geometry feature based. Template based methods compute a measure of correlation between new faces and a set of template models to estimate the face identity. Several well-known statistical techniques have been used to define a template model, such as Support Vector Machines (SVM) [146], Linear Discriminant Analysis (LDA) [7, 43], Principal Component Analysis (PCA) [90, 142] and Independent Component Analysis (ICA) [64]. Usually, these approaches are focused on extracting global features, and occlusions are difficult to handle. Geometry features-based methods analyze explicit local facial features, and their geometric relationships. Some examples of these methods are the active shape model [75], the elastic bunch graph matching algorithm for face recognition [158] and the Local Feature Analysis (LFA) [105].

Since PCA has been widely used for face recognition and classification and NMF has been introduced in the context of extracting a parts-based representation using faces, this section wants to address the problem of recognizing frontal faces captured under different illumination conditions and with the presence of natural occlusions such as individuals wearing sunglasses and / or scarfs. We applied the same framework as in the previous section but instead of using images with digits we use frontal faces. We also evaluated the use of different metric distances in the subspace described by PCA and NMF. In order to obtain comparable results with the most important techniques, we used a face database that has been extensively used by the computer vision community, the AR face database (see appendix C.3).

For our experimental results, we firstly analyzed two leading techniques used in the computer vision community: FaceIt and Bayesian techniques. The FaceIt technique is a successful commercial face recognition system and it is mainly based on the Local Feature Analysis (LFA) [105] technique. The Bayesian technique was developed by Moghaddam and Pentland [90] in order to model large non-linear variations in facial appearance due to self-occlusion and self-shading using a PCA approach as a probability density estimation tool. It is interesting to see the results obtained by Gross et al. in [50] where they have compared both techniques, FaceIt and Bayesian, using the well-known AR face database. Since they provide recognition rates based on this face database, we also want to test the NMF in such representation.

Appendix C.3 contains a detailed description of the AR face database with some examples. But due to the high dimensionality of the original face images, we reduced all face images to a size of 40×48. Thus, our representation becomes more manageable. As PCA and NMF will be directly based on the pixels of each face image, a pose normalization has been applied in order to align all face images. We have manually localized both eye positions in every image and we have normalized all faces according to this information. Furthermore, in order to avoid external influences of background, we have defined an elliptical region that removes possible pixel artifacts. Figure (3.18) shows an example of an individual taken under different ambient conditions and the elliptical region considered. The size of each reduced image is 40 × 48 pixels and if we consider the elliptical region, each image is represented using 1505 pixels. The elliptical region considered has been extracted after analyzing all images of the database. We rejected all those pixels that do not have a relevant statistical influence in a face.

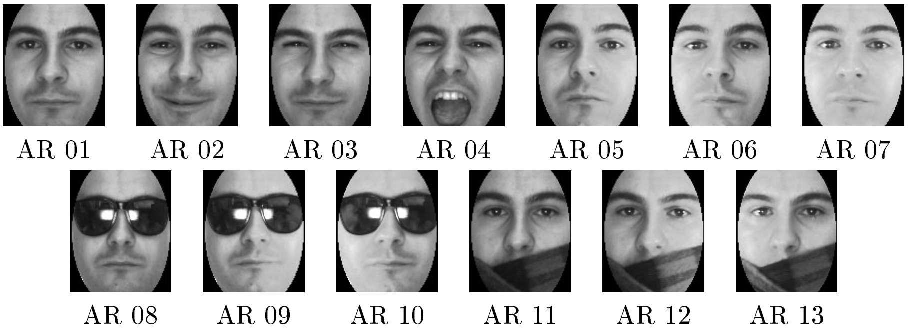
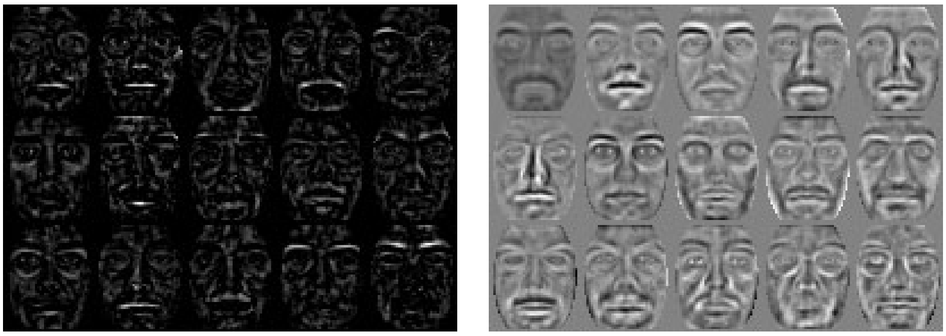


Figure 3.18: Conditions of an individual of the AR face database: (1) neutral, (2) smile, (3) anger, (4) scream, (5) left light on, (6) right light on, (7) both lights on, (8) sunglasses, (9) sunglasses/left light, (10) sunglasses/right light, (11) scarf, (12) scarf/left light, (13) scarf/right light

Here, face images are represented in a 1505 dimensional space and are projected in a low dimensional subspace of 50, 100 and 150 dimensions. We reduced the original space to three different dimensions in order to have a general idea of how results can change when the dimensionality of the subspace is changed. As in the previous section, we have to note that the parts-based representation provided by NMF should be reflected in the results. In order to see the obtained NMF bases of our face images, figure (3.19) shows some of the bases. Again, we can see that NMF provides a sparse representation that tends to be parts-based instead of the global one provided by PCA.

As explained in appendix (C.3), each individual consists of 13 different poses and one of them is a neutral face pose. So that, we selected the neutral face pose to train our face model. Since there are two images of each pose, we took the two neutral face images of each individual to train our model.

Images labelled as AR02, AR03 and AR04 are the ones that reflect different facial expressions. They contain smile, anger and scream expressions. Table (3.25) shows the results obtained using PCA and NMF with respect to FaceIt and Bayesian techniques. The first impression is that L_2 distance is not the most suitable metric when we work with NMF. And, again, both L_1 and cosine metrics could be a good choice. Expression AR02 is better classified by FaceIt technique and AR03 is better classified when we use NMF in a high dimensional space. But it seems clear that expression AR04 is a very difficult one because neither PCA nor NMF are able to obtain acceptable results. For this particular face expression (scream), both FaceIt and Bayesian schemes are better than PCA or NMF.



(a) NMF bases.

(b) PCA bases.

Figure 3.19: Bases obtained by both techniques, PCA and NMF.

		Facial expressions		
		AR 02	AR 03	AR 04
FaceIt		0.96	0.93	0.78
Bayesian		0.72	0.67	0.41
PCA-50+L2 Norm		0.67	0.82	0.18
NMF-50 +	L2 Norm	0.61	0.78	0.14
	L1 Norm	0.72	0.80	0.19
	Correlation	0.73	0.77	0.18
PCA-100+L2 Norm		0.80	0.88	0.24
NMF-100 +	L2 Norm	0.62	0.85	0.09
	L1 Norm	0.85	0.91	0.29
	Correlation	0.89	0.90	0.28
PCA-150+L2 Norm		0.83	0.90	0.29
NMF-150 +	L2 Norm	0.66	0.87	0.09
	L1 Norm	0.88	0.92	0.30
	Correlation	0.93	0.95	0.36

Table 3.25: Facial expression results. This table reflects how both techniques can deal with facial expressions. Note that scream expression (AR 04) is hard to recognize.

We have also considered a set of different illumination conditions because it is an important factor to take into account in a face recognition system. This illumination conditions are reflected in images AR05, AR06 and AR07. Table (3.26) shows that PCA can not deal with illumination conditions as good as NMF. Furthermore, it is interesting to note that when the dimensionality of the subspace is increased, NMF improves FaceIt and Bayesian techniques.

One of the topics to be dealed with face recognition is occlusions. Faces from AR02 to AR07 contain different facial expressions and different lighting conditions but there are not occlusions. Now, we want to evaluate a set of natural occlusions where faces are occluded using a scarf or sunglasses. AR08 contains sunglasses that occlude both eyes and AR09 and AR10 also contain sunglasses but they also contain different lighting conditions. So, as it is natural to think, these two last image faces

		Expression with lighting changes		
		AR 05	AR 06	AR 07
FaceIt		0.95	0.93	0.86
Bayesian		0.77	0.74	0.72
PCA-50+L2 Norm		0.77	0.76	0.57
NMF-50 +	L2 Norm	0.91	0.84	0.67
	L1 Norm	0.93	0.87	0.69
	Correlation	0.94	0.89	0.76
PCA-100+L2 Norm		0.86	0.86	0.69
NMF-100 +	L2 Norm	0.94	0.85	0.67
	L1 Norm	0.97	0.94	0.87
	Correlation	0.99	0.94	0.88
PCA-150+L2 Norm		0.85	0.87	0.71
NMF-150 +	L2 Norm	0.93	0.84	0.64
	L1 Norm	0.98	0.97	0.92
	Correlation	0.99	0.96	0.91

Table 3.26: Illumination results. This table reflects how both techniques manage illumination changes in faces. Note that in this case, NMF obtains the best classification results when the number of dimensions starts to be considerable (100 or 150).

(AR09 and AR10) should be very difficult to identify since they contain occlusions and they are also affected by lighting conditions. The other natural occlusion to be considered under this scheme is the one that contains a scarf. This means that the mouth is occluded and part of the face can not be recognized. AR12 and AR13 also consider a scarf but with the addition of some lightings. Tables (3.27) and (3.28) show all the results obtained when we consider these two kinds of occlusions.

Under the presence of sunglasses, recognition rates decrease considerably as can be seen in table (3.27). This means that eyes are a very important facial feature to take into account when we classify faces. It is interesting to note that when sunglasses are considered without the presence of lighting influences (AR08), NMF obtains the best recognition results. But, when lighting conditions are present in faces in conjunction with partial occlusions (AR09 and AR10), the Bayesian technique performs better than the other methods. Thus, NMF is a good choice when partial occlusions are present and without the presence of lighting conditions. It seems that NMF can deal with local changes in an image but not with a more general change of the scene. Table (3.28) shows a similar behaviour when we analyze faces that contain a scarf. That is, NMF works very good when we analyze an image with a scarf (AR11) but when the presence of a scarf in conjunction with lighting conditions is present in an image, NMF decreases its performance. It should be mentioned that performance of NMF is comparable to the FaceIt technique in the particular case of considering a scarf.

In general, the first impression of these experiments is that NMF performs better than PCA in the same dimensional subspace. This behaviour was expected because PCA is based on a global transformation of the original space and NMF is based on a local one and our face database is mainly composed of face occlusions. Also, from the results obtained in the previous section with the MNIST digit database, we

		Expressions with occlusions (sunglasses)		
		AR 08	AR 09	AR 10
FaceIt		0.10	0.08	0.06
Bayesian		0.34	0.35	0.28
PCA-50+L2 Norm		0.16	0.12	0.18
NMF-50 +	L2 Norm	0.16	0.10	0.12
	L1 Norm	0.19	0.10	0.20
	Correlation	0.23	0.12	0.17
PCA-100+L2 Norm		0.23	0.15	0.22
NMF-100 +	L2 Norm	0.14	0.11	0.12
	L1 Norm	0.24	0.15	0.21
	Correlation	0.32	0.19	0.24
PCA-150+L2 Norm		0.26	0.16	0.24
NMF-150 +	L2 Norm	0.17	0.12	0.09
	L1 Norm	0.31	0.21	0.23
	Correlation	0.38	0.21	0.23

Table 3.27: Occlusion results when sunglasses are present. Note that in this case, NMF only is better when we use a high dimensional feature space and no lighting conditions are considered. When lighting conditions are considered, Bayesian approach obtains the best recognition rates.

		Expressions with occlusions (scarf)		
		AR 11	AR 12	AR 13
FaceIt		0.81	0.73	0.71
Bayesian		0.46	0.43	0.40
PCA-50+L2 Norm		0.44	0.38	0.37
NMF-50 +	L2 Norm	0.47	0.35	0.28
	L1 Norm	0.59	0.35	0.32
	Correlation	0.61	0.45	0.35
PCA-100+L2 Norm		0.59	0.50	0.47
NMF-100 +	L2 Norm	0.47	0.36	0.25
	L1 Norm	0.66	0.55	0.46
	Correlation	0.76	0.62	0.59
PCA-150+L2 Norm		0.62	0.57	0.48
NMF-150 +	L2 Norm	0.53	0.31	0.24
	L1 Norm	0.73	0.57	0.48
	Correlation	0.75	0.62	0.56

Table 3.28: Occlusion results when a scarf is present. In this case, FaceIt obtains the best recognition results. And we have to note that NMF is better than the Bayesian approach in this situation. Again, NMF is always better than PCA.

were able to predict this performance. Thus, it turns out that when we consider local effects as occlusions, changes in expression or even changes in illumination, PCA is not able to deal with these effects as well as NMF. In terms of performances NMF has comparable recognition rates with respect to FaceIt and Bayesian techniques and, in some situations, is even better than these two methods. The reason of this high performance is mainly justified because NMF is able to represent data using a parts-based representation. Finally, it is clear from these experiments that L_2 metric distance is the worst metric to be considered in conjunction with NMF and, as expected from our previous section, the cosine metric is the best one.

One of the most interesting things to analyze in this framework is how we can describe the local features of males and females. It is common to think that the local facial features that describe a male are different from the local facial features that describe a female. So that, this means that NMF can be a suitable technique in order to capture these local differences. And this motivates to create a gender classifier based on the NMF technique and when a testing face is correctly classified according to its gender, we can use this information to recognize the face using a more specific classifier. Again, we want to compare performance of NMF with respect to PCA, so that, we learned two gender classifiers: one with PCA and the other one with NMF. For these gender classifiers, we have used the same subspace parameters as before. Figure (3.20) shows the gender classification results when using 50, 100 and 150 dimensional subspaces.

Figure (3.20) depicts a general behaviour of PCA and NMF techniques: females are better recognized in the following face situations AR02, AR03, AR05, AR06 and AR07 and males in the other ones. The reason of these recognition differences is not clear but it seems that each gender has some particular facial local features and maybe this fact could affect to obtain these recognition differences between genders. We should say that when face images contain occlusions as images AR08, AR09 and AR10, gender recognition performs very bad since the recognition performance decreases considerably.

Considering that NMF is based on capturing local behaviours, we can think that a more specific classifier based only on males or females should improve the initial recognition rates presented before. So that, we learned both PCA and NMF models in order to perform gender classification using the same internal parameters as in the previous experiments. Tables (3.29,3.30,3.30,3.32) shows the obtained results.

In general, with the addition of a gender classifier, both techniques (PCA and NMF) are slightly improved. This improvement is not very significant in face images that contain complex occlusions such as those faces that contain sunglasses or a scarf. However, theses results motivate to build a face classifier divided into a global gender detector and two specific face classifiers, one for males and another for females. This configuration must work better than only considering a universal face classifier because NMF is based on the representation of local features. So that, using two gender classifiers, NMF represent each gender more properly. Figure (3.21) summarizes previous results showing all the recognition rates obtained according to the method used (PCA or NMF) in conjunction with their internal parameters. We have to note that the overall recognition rate of the FaceIt technique is 65.83% and 52.42% for the Bayesian one.

From the analysis of figure (3.21), we can appreciate that the introduction of a gender classifier improves the whole recognition rates even using PCA or NMF. Obviously, this behaviour is justified because it is more easy to classify a face between a male and a female than recognizing the face directly. But it is clear that this improvement is more remarkable in low dimensional subspaces.

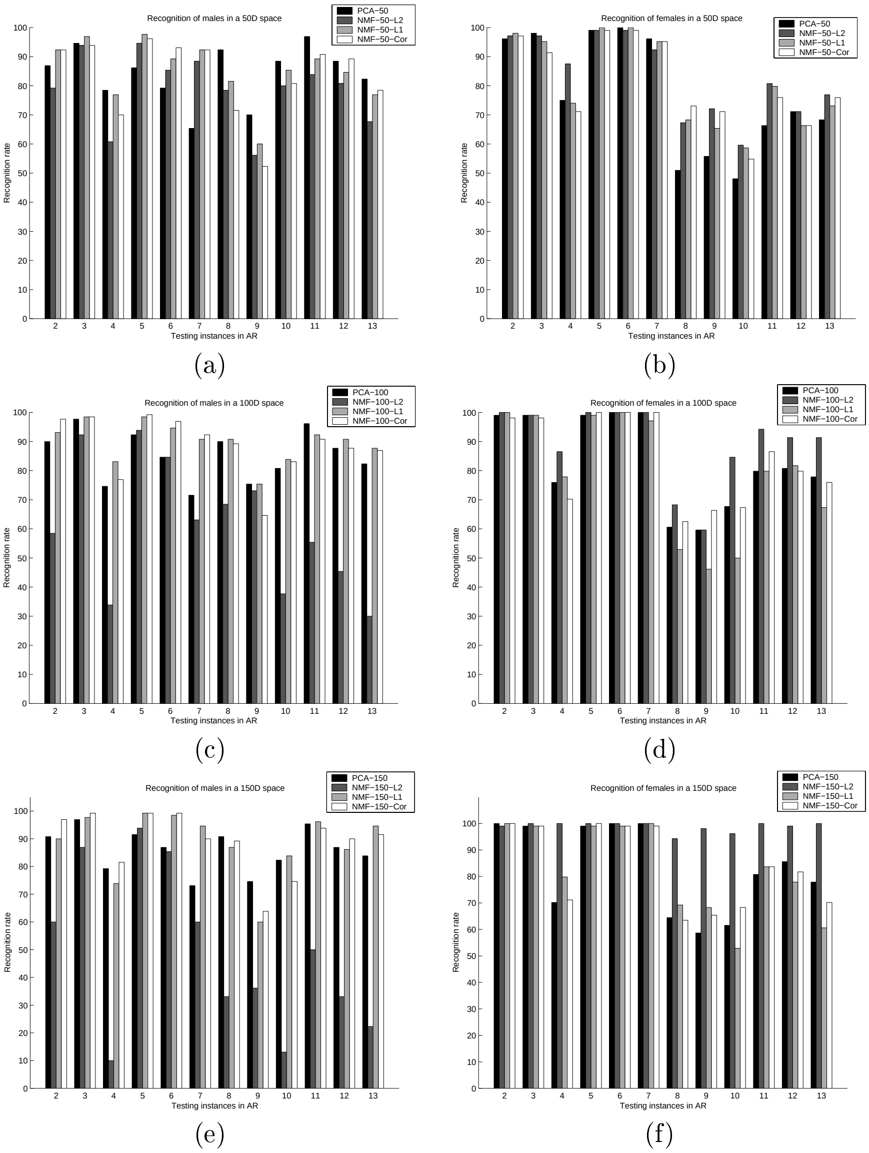


Figure 3.20: Gender classification results when we classify males (a) and females (b) in a 50 dimensional feature subspace. (c) and (d) are the results in a 100 dimensional subspace and (e) and (f) in a 150 dimensional subspace.

If we directly compare the overall results of PCA and NMF with respect to FaceIt and Bayesian techniques, we can state that performances are comparable depending on the subspace dimension. The best configuration of our scheme is the one that uses NMF in a 150 dimensional subspace using the cosine metric distance where we obtain a recognition rate of 66.74% that is greater than the recognition rate of 65.87%

		Expression		
		AR 02	AR 03	AR 04
PCA-50+L2 Norm		0.74	0.87	0.22
NMF-50 +	L2 Norm	0.68	0.83	0.17
	L1 Norm	0.81	0.87	0.25
	Correlation	0.85	0.84	0.25
PCA-100+L2 Norm		0.83	0.91	0.28
NMF-100 +	L2 Norm	0.65	0.87	0.14
	L1 Norm	0.90	0.92	0.31
	Correlation	0.91	0.94	0.34
PCA-150+L2 Norm		0.84	0.91	0.28
NMF-150 +	L2 Norm	0.70	0.88	0.13
	L1 Norm	0.90	0.93	0.33
	Correlation	0.93	0.94	0.35

Table 3.29: Expression results when we consider a previous gender classifier. This table must be compared with table (3.25) where we can appreciate some improvements.

		Expression		
		AR 05	AR 06	AR 07
PCA-50+L2 Norm		0.82	0.82	0.62
NMF-50 +	L2 Norm	0.91	0.86	0.68
	L1 Norm	0.94	0.89	0.76
	Correlation	0.96	0.92	0.84
PCA-100+L2 Norm		0.86	0.86	0.70
NMF-100 +	L2 Norm	0.92	0.89	0.73
	L1 Norm	0.97	0.96	0.89
	Correlation	0.98	0.97	0.92
PCA-150+L2 Norm		0.86	0.87	0.71
NMF-150 +	L2 Norm	0.94	0.89	0.69
	L1 Norm	0.98	0.98	0.92
	Correlation	0.98	0.97	0.93

Table 3.30: Illumination results when we consider a previous gender classifier. This table must be compared with table (3.26). In this particular case, AR 07 is specially improved in low dimensional spaces.

obtained by the FaceIt technique.

It seems that the combination of NMF in conjunction with the cosine distance is a good scheme to work with this whole set of face conditions. It is surprising to find that NMF can outperform a commercial technique as FaceIt or the Bayesian one in some of the face expressions. We think that this is justified because these two well-known techniques have been designed to work with faces that contain specific changes in expression, but not the whole range of conditions that we have exposed. And again, we have to note that the cosine distance seems to be the best metric distance to be used with NMF when our images contain natural occlusions.

		Expression		
		AR 08	AR 09	AR 10
PCA-50+L2 Norm		0.21	0.13	0.20
NMF-50 +	L2 Norm	0.20	0.10	0.14
	L1 Norm	0.24	0.14	0.22
	Correlation	0.29	0.17	0.24
PCA-100+L2 Norm		0.25	0.16	0.23
NMF-100 +	L2 Norm	0.16	0.15	0.13
	L1 Norm	0.26	0.17	0.21
	Correlation	0.35	0.21	0.25
PCA-150+L2 Norm		0.27	0.17	0.25
NMF-150 +	L2 Norm	0.18	0.13	0.10
	L1 Norm	0.32	0.20	0.24
	Correlation	0.36	0.24	0.26

Table 3.31: Occlusion results when we consider sunglasses and a previous gender classifier. This table must be compared with table (3.27). We can see that in this particular case, recognition rates are not really improved.

		Expression		
		AR 11	AR 12	AR 13
PCA-50+L2 Norm		0.52	0.44	0.40
NMF-50 +	L2 Norm	0.51	0.34	0.31
	L1 Norm	0.60	0.41	0.37
	Correlation	0.67	0.51	0.45
PCA-100+L2 Norm		0.64	0.55	0.49
NMF-100 +	L2 Norm	0.50	0.32	0.24
	L1 Norm	0.71	0.56	0.51
	Correlation	0.79	0.62	0.57
PCA-150+L2 Norm		0.63	0.57	0.51
NMF-150 +	L2 Norm	0.53	0.36	0.27
	L1 Norm	0.73	0.58	0.52
	Correlation	0.79	0.65	0.61

Table 3.32: Occlusion results when we consider a scarf and a previous gender classifier. This table must be compared with table (3.28). In this case, recognition rates present a general improvement.

3.6 Estimation of the Original Probability Density Function

Two techniques which are based on linear transformations of data have been presented. PCA and NMF/WNMF are strictly based on reducing the dimensionality of a feature space in order to produce a compact representation. This compact representation usually contains less dimensions than the original feature space. Then, the *curse of dimensionality* problem is alleviated. The main motivation of this dimensionality reduction is the fact that the original feature space is high dimensional and estimation of probability density functions is not reliable. In fact, we will need a huge

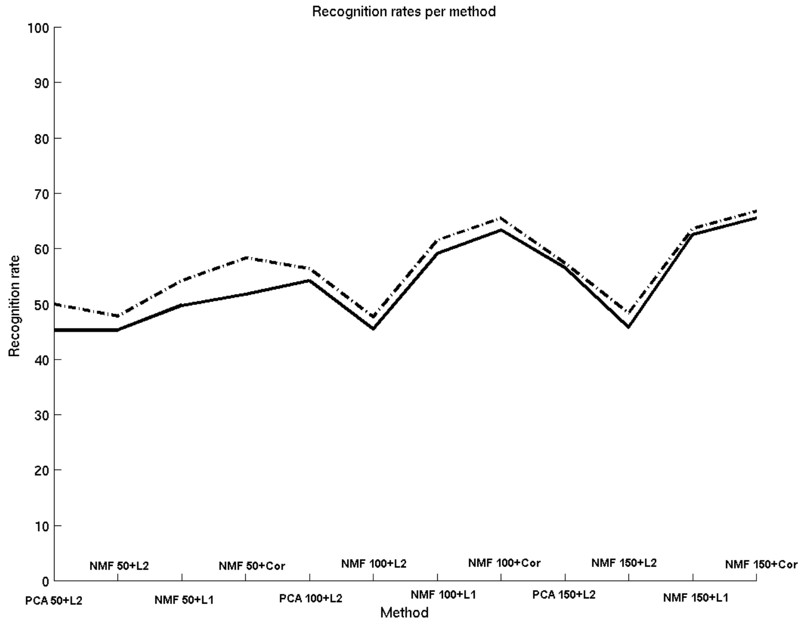


Figure 3.21: Recognition rates according to the method used. Solid line indicates the recognition rates obtained without using any gender information and the dashed line indicates the recognition rates when considering gender information. The best method according to the whole set of face situations is the Non-negative Matrix Factorization in a 150 dimensional space using the correlation distance as a metric obtaining a recognition rate of 66.74%.

amount of data samples in a high dimensional space to obtain reliable probability density function estimators.

Once we reduced the original feature space using PCA or NMF, we can try to use the projected coefficients to estimate a viable probability density function for the original space. This has been done in section (3.5.2). However, it remains an open question: is it possible to estimate a probability density function in the original space without reducing data dimensionality using general data? The term general data refers to positive and negative values of data.

In this section we introduce another technique which is based on a linear transformation of data. This technique, the Independent Component Analysis (ICA), will help us to estimate probability density functions. We will explain Independent Component Analysis and extend its definition to a broader problem: the use of multiple object classes. We have noticed that ICA is not adapted to problems which contain several object classes. Then, we propose to use the Class-Conditional Independent Component Analysis (CC-ICA). A final example using the Class-Conditional ICA is shown.

3.6.1 Independent Component Analysis

If we assume our data is the result of linearly combining nongaussian and mutually independent latent variables with an unknown mixing matrix, independent component analysis (ICA) is the statistical technique which reveals these hidden factors by defining a generative model on the observed data. In this case, the latent variables are called *independent components* or *sources* of the observed data.

Blind Source Separation (BSS, also known as Blind Signal Separation) is the classical application of the ICA model, and one of the main motors for all initial research on ICA [67]. BSS consists of recovering unobserved signals or *sources* from several observed mixtures. The *cocktail-party problem*, a paradigmatic BSS situation, provides a clarifying picture of the ICA context. Imagine a room with D people talking simultaneously and M microphones placed in different room locations. In this case, the original speech signals, or sources, can be represented by an M-dimensional random vector (one per person in the room) and the recorded sound signals which actually is our observed data is represented by a D-dimensional random vector (one per microphone). The problem here is to estimate the original speech signals from the recorded signals. If we omit time delays, noise and other extra factors to simplify our model, we can linearly approximate the mixing function. Also, it is not unrealistic to assume that the speech signals are statistically independent. This is equivalent to assume that speech waveforms corresponding to different persons are statistically independent signals. Since this waveform is nongaussian (speech waveforms are generally nongaussian), we are under the assumptions of the linear ICA model, and ICA provides a solution for this problem.

Here, we will resctrict ourselves to what is commonly known as the *basic* ICA model [64]. This is the linear instantaneous noise-free mixing model classic in ICA, opposed to several extensions which consider nonlinear mixing, inclusion of explicit observational noise or time dependency. In the following notation, we will assume that our data is zero-centered. In practice, this situation can be always achieved by previously subtracting the global mean from the working dataset. Given a set of observations represented by the D dimensional random vector $\mathbf{x}$, assume the following generative model

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.36}$$

where the latent variables or *independent components* s_m in vector $\mathbf{s} = (s_1, \dots s_M)^T$ are assumed to be independent and the $D \times M$ basis matrix $\mathbf{A}$ is unknown. Independent component analysis consists of estimating both matrix and independent components, when we only observe $\mathbf{x}$. Following the BSS application explained before, the independent components are also known as sources and the basis matrix, usually called mixing matrix. The pseudoinverse of matrix $\mathbf{A}$ which we will represent as $\mathbf{W}$, is called the filter or projection matrix and provides an alternative expression for ICA,

$$\mathbf{W}\mathbf{x} = \mathbf{s} \tag{3.37}$$

This expression provides an alternative definition for ICA, less rigorous but far more illustrative. Given a dataset $\mathbf{x}$, ICA searches for the linear transformation of the data $\mathbf{W}$, such that the projected variables are as independent as possible.

It has been shown [32] that if certain assumptions hold, the ICA model is completely identifiable, i.e. there exists a solution to the problem of estimating the mixing matrix and components. These assumptions are,

- *The independent components are assumed statistically independent.* Knowledge of one component gives us no information on the value of any other component. This is the main principle on which ICA rests. This is why ICA is such a powerful method with applications in many different areas.

  Basically, random variables $y_1, y_2, \ldots y_n$ are said to be independent if information on the value of y_i does not give any information on the value of y_j for $i \neq j$. Technically, independence can be defined by the probability densities. Let us denote by $p(y_1, y_2, \ldots y_n)$ the joint probability density function (pdf) of the y_i, and by $p(y_i)$ the marginal pdf of y_i, i.e., the pdf of y_i when it is considered alone. The we say that the y_i are *independent* if and only if the joint pdf is factorizable in the following way:

  $$p(y_1, y_2, \ldots, y_n) = p(y_1)p(y_2)\ldots p(y_3) \tag{3.38}$$

  So that, this is the main advantage of ICA: we can work with the marginal densities because we assume that they are independent.

- *The independent components must have nongaussian distributions.* Intuitively, one can say that the gaussian distributions are "too simple". The higher-order cumulants are zero for gaussian distributions, but such higher-order information is essential for estimation of the ICA model [64]. Thus, ICA is essentially impossible if the observed variables have gaussian distributions.

- *The unknown mixing matrix is square.* This situation, in which $M = D$, is called the *complete case*. In this case $\mathbf{W} = \mathbf{A}^{-1}$ and estimation is greatly simplified. This assumption is not a necessary condition and can sometimes be relaxed. When this is done, two situations can arise. If we consider less sources than observations ($M < D$) we have that, though $\mathbf{W}$ can be completely determined, $\mathbf{A}$ contains uncertainty. In this case, the common approach is to previously perform dimensionality reduction using for instance PCA in a preprocessing stage, and then restrict the problem to the complete case. The second situation is when there are more independent components than dimensions in the data ($M > D$) and it is referred to as ICA with *overcomplete* bases. In this case, estimation is much more complicated and estimation methods less developed [82].

The ICA model also contains some ambiguities. From equation (3.36) we know that the independent components are zero-centered but we can not determine the variances (energies) of the independent components. This is due to the fact that both $\mathbf{s}$ and $\mathbf{A}$ are unknown so any scalar multiplier on one of the sources can be cancelled by dividing the corresponding column of $\mathbf{A}$ by the same scalar. To overcome this situation, the magnitudes of the independent components are considered fixed. For instance, considering that each independent component s_m has unit variance: $Es_m^2 = 1$. Notice that this restriction still leaves an ambiguity in the sign, because the multiplication of an independent component by -1 does not affect the model. Fortunately, this is insignificant in most of the situations.

Another important ambiguity in the model is that the order in the components can not be determined. Given any permutation matrix $\mathbf{P}$ the model given in equation (3.36) is equivalent to $\mathbf{x} = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}$. In many applications an order for the sources is necessary, so different ordering criterions can be used. The norm of the columns $\mathbf{A}$ can be understood as the contributions of the different sources to the variance of $\mathbf{x}$, so an order reminiscent to that of PCA would be to number the independent components in decreasing order of the norm of the columns of mixing maxtrix $\mathbf{A}$. As it is also known that measures of nongaussianity play a significative role in ICA estimation [64]. So another possibility is to order te sources according to their nongaussianity. Here, the obtained order of independent components would be related to the order given by projection pursuit. Nevertheless, none of these approaches is definitive and ordering the independent components is absolutely problem-dependent. Actually, imposing a hierarchy on the independent components would break the nature of ICA: if no sources gives information on another source, does it have any sense to sort them?

Since the ICA problem can be seen under different perspectives, there are several methods to obtain the parameters of expression (3.37). With this thesis, we only want to present ICA as a method to perform this parameter estimation without entering to the internal details of ICA. Following this, we only present one approach for ICA estimation: the classical maximum lilelihood (ML) method. This algorithm has been extensively tested and improved. Its main drawbacks are its computational load and the heuristic fact it does not generalize properly to high dimensions. Although we used the FastICA algorithm [65] in our experiments, we introduce maximum likelihood estimation since it is a natural approach to the statistical parameter estimation problem we are faced with (the parameters are the components of the filter matrix) and it illustrates clearly a basic point we want to make: ICA is the representation in which the product of the marginal probabilities of the projected features best approximates the probability of the original features, times a constant value.

If we assume independence on the sources, expression (3.36) can be expressed as

$$p(\mathbf{x}) = |\det \mathbf{W}| p(\mathbf{s}) = |\det \mathbf{W}| \prod_{m=1}^{M} p_m(s_m) = |\det \mathbf{W}| \prod_{m=1}^{M} p_m(\mathbf{w}_m^T \mathbf{x}) \tag{3.39}$$

where we used the change of variables that define a density transformation. This density transformation is explained as follows: If we assume that both $\mathbf{x}$ and $\mathbf{y}$ are n- dimensional random vectors that are related by the vector mapping

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{3.40}$$

for which the inverse mapping

$$\mathbf{x} = \mathbf{g}^{-\mathbf{1}}(\mathbf{y}) \tag{3.41}$$

exists and is unique. It can be shown that the density $p_y(\mathbf{y})$ of $\mathbf{y}$ is obtained from the density $p_x(\mathbf{x})$ of $\mathbf{x}$ as follows:

$$p_y(\mathbf{y}) = \frac{1}{|\det J\mathbf{g}(\mathbf{g}^{-1}(\mathbf{y}))|} p_x(\mathbf{g}^{-1}(\mathbf{y})) \tag{3.42}$$

Here $J\mathbf{g}$ is the *Jacobian matrix*

$$J\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_1} \\ \frac{\partial g_1(\mathbf{x})}{\partial x_2} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1(\mathbf{x})}{\partial x_n} & \frac{\partial g_2(\mathbf{x})}{\partial x_n} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{3.43}$$

and $g_j(\mathbf{x})$ is the jth component of the vector function $\mathbf{g}(\mathbf{x})$. In the special case where the transformation (3.40) is linear and nonsingular so that $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{x} = \mathbf{A}^{-\mathbf{1}}\mathbf{y}$, the formula (3.42) simplifies to

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{1}{|\det\ \mathbf{A}|} p_x(\mathbf{A}^{-1}\mathbf{y}) \tag{3.44}$$

So that, returning to expression (3.39) where we had that p_m is the unidimensional marginal distribution of the m-th independent component and, if we resctrict ourselves to the complete model, $\mathbf{W} = \mathbf{A}^{-\mathbf{1}}$. The last equality of expression (3.39) uses that if $\mathbf{w}_m$ is the vector corresponding to the m-th row of matrix $\mathbf{W}$, then $s_m = \mathbf{w}_m^T\mathbf{x}$. Remember that maximum likelihood searches for the parameter values that give highest probabilities to the observations so we now assume we have N observations of feature vector $\mathbf{x}$ which we note by $\mathbf{x}^1, \ldots \mathbf{x}^N$, then, assuming sample independence, the likelihood $L(\mathbf{W})$ is obtained as the product of (3.39),

$$L(\mathbf{W}) = \prod_{n=1}^{N} |\det\ \mathbf{W}| \prod_{m=1}^{M} p_m(\mathbf{w}_m^T\mathbf{x}^n) \tag{3.45}$$

and the algebraically simpler log-likelihood,

$$\log\ L(\mathbf{W}) = \sum_{n=1}^{N}\sum_{m=1}^{M} \log\ p_m(\mathbf{w}_m^T\mathbf{x}^n) + N|\det\ \mathbf{W}| \tag{3.46}$$

Notice that this sum can be replaced by the (frequential) expectation operator if both sides of the equation are divided by the number of samples, yielding the following equivalent expression,

$$\frac{1}{N}\log\ L(\mathbf{W}) = E\{\sum_{m=1}^{M} \log\ p_m(\mathbf{w}_m^T\mathbf{x})\} + |\det\ \mathbf{W}| \tag{3.47}$$

There are several algorithms for maximizing this expression such as gradient methods, natural gradient methods, fixed-point algorithms or even an application of the expectation-maximization (EM) algorithm. By differentiating expression (3.47) with respect to $\mathbf{W}$ we obtain the following expression for the gradient,

$$\frac{1}{N}\frac{\partial \log\ L}{\partial \mathbf{W}} = E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^{\mathbf{T}}\} + (\mathbf{W}^T)^{-1} \tag{3.48}$$

where g is the component-wise vector function whose components $1 \leq m \leq M$ are defined as,

$$g_m(s) = \frac{\partial \log p_m(s)}{\partial s} = \frac{1}{p_m(s)} \frac{\partial p_m(s)}{\partial s} \tag{3.49}$$

These functions are also called *score* functions of the distribution p. Equation (3.48) yields the following gradient descent iteration for Maximum Likelihood (ML) estimation,

$$\Delta \mathbf{W} \propto E\{\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^{\mathbf{T}}\} + \{\mathbf{W}^{\mathbf{T}}\}^{-1} \tag{3.50}$$

This algorithm was first derived by Bell and Sejnowski [8] from an information theoretic approach that yields the same results. The main drawback of this algorithm is its slow convergence, both theoretically (as with most gradient descent procedures) and computationally (inversion of $\mathbf{W}$ is required on each step). This situation can be attenuated by using the natural or relative gradient, which amounts to multiplying the right hand side by $\mathbf{W}^{\mathbf{T}}\mathbf{W}$. This algorithm makes use of the fact that, for our problem, the parameter space (nonsingular matrices) has a Riemannian instead of an Euclidean metric structure [1]. The natural gradient descent iteration for ML estimation is better conditioned than its gradient version,

$$\Delta \mathbf{W} \propto (\mathbf{I} + E\{\mathbf{g}(\mathbf{W}\mathbf{x})(\mathbf{W}\mathbf{x})^{\mathbf{T}}\})\mathbf{W} \tag{3.51}$$

We still have not treated a basic and necessary condition for the implementation of any ML estimation procedure: the choice of the distributions. Since the independent components are themselves unknown, so are their distributions. The most common approach in this case is to restrict the densities to a particular family (taking a parametric approach). Of course, not any family since this choice affects the consistency of the estimator. Results for stability analysis and local consistency of ML for ICA estimation have been derived [64] notably showing that accurate density estimation is not absolutely necessary so simple models can be applied. Based on these results several choices for the distributions have been proposed [64, 29].

Remember from the assumptions made about an ICA model that we should restrict ourselves to nongaussian densities. These densities can be further divided into two groups, *subgaussian* and *supergaussian*, based on the value of the fourth order statistic known as *kurtosis* defined as,

$$K(s) = E\{s^4\} - 3 \tag{3.52}$$

for a zero mean, unit variance random vector (true for the independent components). Its value is proportional to the concentration of the variable around zero. It can be seen that $K(s)$ is zero if s is Gaussian. From here that negative kurtotic variables are said to have subgaussian (or platykurtotic) distributions, and positive kurtotic supergaussian (or leptokurtotic) distributions. Kurtosis measures the *peakiness* of a distribution. In the range of unimodal distributions the uniform distribution can be considered the least "peaky", the Dirac delta its opposite. Having made this distinction, the simplest form of effective family densities for maximum likelihood ICA estimation have a single parameter [8, 64]: a binary parameter which decides whether the distribution of the m-th independent component is sub or supergaussian and, having decided this, assigns a predefined (sub or supergaussian) fixed density to s_m. The most extended moderately supergaussian density is the *Laplacian* or *double-exponential* density,

$$p_m(s_m|\alpha) = \frac{1}{\sqrt{2}\alpha} e^{-\frac{\sqrt{2}|s_m|}{\alpha}} \qquad (3.53)$$

This density has the undesirable property of not being differentiable in the origin, so a smooth approximation of the logarithm of this density is introduced,

$$p_m^+(s_m) = \alpha^+ - 2\text{log}\ \cosh(s_m) \qquad (3.54)$$

where α^+ is fixed in order to make the function the logarithm of a probability density. Replacing expression (3.54) in (3.49) we have that the score function for this choice is,

$$g_m^+(s_m) = -2\tanh(s_m) \qquad (3.55)$$

For the subgaussian case, the following log-density is proposed,

$$p_m^-(s_m) = \alpha^- - (\frac{s_m^2}{2} - \log\ \cosh(s_m)) \qquad (3.56)$$

where α^- is also a normalizing constant and the corresponding score density is given by,

$$g_m^-(s_m) = \tanh\ (s_m) - s_m \qquad (3.57)$$

Of course, more precise parametric models can be studied for the component densities. A highly flexible model with the interesting property of joining the sub and supergaussian cases in a single parametrization is the result of using the *generalized Gaussian* distribution [64]. In this case, the component is assumed to belong to the following family of distributions,

$$p_m^G(s_m, \alpha_m) = \frac{\alpha_m}{2\lambda\Gamma(\frac{1}{\alpha_m})} e^{-\frac{|s_m|}{\lambda_m}{}^{\alpha}_{m}} \qquad (3.58)$$

where the real positive number α_m controls the "peakiness" and is often referred to as the Gaussian exponent of the distribution, λ_m depends on α_m and the variance (here fixed to 1), and Γ is the Gamma function given by

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \qquad (3.59)$$

The only parameter we need to estimate is the value of α_m for each component. The score function for expression (3.58) is

$$g_m^G(s_m, \alpha_m) = |(s_m)|^{\alpha_m - 1} \text{sign}\ (s_m) \qquad (3.60)$$

In practice, estimation of α_m can be done using the kurtosis of the corresponding component $K(s)$ [64].

3.6.2 Class Conditional Independent Component Analysis

As seen, we can use PCA to reduce the dimensionality of a given feature space and use Independent Component Analysis (ICA) to estimate probability density functions using the projected coefficients. From a statistical perspective, the maximum likelihood approach states that the product of the marginal densities of the projected data (independent components) best fits the global distribution for the observations. If we assume that we are in the context of the basic ICA model (see expression (3.37)) it implies that independent components satisfy that $p(s) \cong \prod_m p(s_m)$ and estimation of the D-dimensional density of our features ($\mathbf{x}$) in domain space can be approximated estimating M unidimensional densities in the projected space since

$$p(\mathbf{x}) = |\det\ \mathbf{W}| p(\mathbf{s}) \cong |\det\ \mathbf{W}| \prod_{m=1}^{M} p(s_m) \tag{3.61}$$

with $\mathbf{W}$ the ICA filter matrix and assuming random vector $\mathbf{x}$ is zero-centered ($E\{\} = 0$). Though tempting, straightforward application of ICA for classification, i.e. unsupervisedly learning ICA from the dataset and working with the marginal densities of the independent components is incorrect. The Bayesian classification scheme makes use of the class-conditional densities and it would be desirable to take into account data classes. This representation is called class-conditional ICA (CC-ICA).

We will refer to a learning technique as class-conditional when its parameters depend on any given class, as opposed to a global representation, usually estimated from all available samples regardless of their labels. In the case of a nonsingular linear representation, and for a certain class C^k, what we have is that filter and basis matrices are class-dependent $\mathbf{W} = \mathbf{W^k}$ and $\mathbf{A} = \mathbf{A^k}$. Since class-conditional representations are adapted to the class they are able to learn patterns that otherwise would be lost. For instance, for a given reconstruction error class-conditional PCA used for dimensionality reduction would surely allow a more compact set of features for the description of a certain class than global PCA. This is because global PCA takes into account extra-class variances as well as intra-class. The counterpart of this choice is that, instead of a single representation, we have to learn as many representations as classes there are. More importantly, class-conditional representations fail to model the relationship among classes, for instance discriminability. If necessary, these relations have to be learnt using further techniques such as feature selection, which is not straightforward considering that different classes are represented with different features.

In terms of bayesian theory, given a random feature vector $\mathbf{x}$, to classify it as belonging to one of K classes $C^1, C^2, \dots C^K$. To assume the distribution is known is equivalent to assume that we know the probability of a class, given the outcome $\mathbf{x}$. For a class C^k this distribution, noted by $P(C^k|\mathbf{x})$, is named the *posterior* or *a posteriori* probability of the class. Although it is not the only alternative it seems natural to assign to $\mathbf{x}$ the class with maximum posterior probability. This assignment is known as the Bayes or Maximum A Posteriori (MAP) decision rule [39]:

$$C_{MAP} = \arg\max\ _{k=1,\dots K} P(C^k|\mathbf{x}) \tag{3.62}$$

Bayes' theorem provides an alternative expression for the posterior probability in terms of quantities which are often easier to implement. And taking in mind that this theorem for random variables x and y states that,

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{3.63}$$

so that, expression (3.62) has another alternative

$$C_{MAP} = \text{arg max}_{\ k=1,\dots K} P(\mathbf{x}|C^k)P(C^k) \tag{3.64}$$

Now that the Bayesian theory has been introduced we can state that an important characteristic of class-conditional linear feature extractors is that they can be simply included within a Bayesian classification scheme. Returning to the formulation of ICA of expression (3.39) and considering that we can reformulate it in terms of data classes, we have that the class-conditional probability is

$$p(\mathbf{x}|C^k) = |\det \mathbf{W}^k| p(\mathbf{s^k}|C^k) = |\det \mathbf{W}^k| p^k(\mathbf{s}) \tag{3.65}$$

if $\mathbf{s}^k = \mathbf{W^k x}$. For this case, the Maximum A Posteriori (MAP) solution (see expression (3.64)) takes the following form,

$$C_{MAP} = \text{arg max}_{\ k=1,\dots K} |\det \mathbf{W}^k| p^k(\mathbf{s})P(C^k) \tag{3.66}$$

This simplicity is not true with other classifiers such as the nearest neighbor classifier. The distance of a test sample to members of different classes is performed in different feature spaces so it is very complex to compare them in order to choose the label that corresponds to the sample with the nearest distance. In this case, making the distance invariant to the particular representantations makes us lose whatever we have gained through the choice of representation.

ICA assumes that the extracted features are statistically independent. Now, we have K classes, so that, if we wish to make use of the independence assumption for the class-conditional probabilities, we are then obliged to use class-conditional representations (CC-ICA). The basic CC-ICA model is estimated from the training set for each class. If $\mathbf{W}^k$ and $\mathbf{s}^k$ are the ICA filter matrix and the independent components for class C^k, then from (3.37)

$$\mathbf{s}^k = \mathbf{W}^k(\mathbf{x} - \bar{\mathbf{x}}^{\mathbf{k}}) \tag{3.67}$$

where $\mathbf{x} \in C^k$ and $\bar{\mathbf{x}}^k$ is the class mean, estimated from the training set. Most ICA methods require, or at least advise, data whitening as preprocessing. Since some simple denoising is also recommended, dimensionality reduction and whitening through PCA is a very common practice as a preprocessing stage for ICA. In this case, $\mathbf{W}^k$ can be decomposed as

$$\mathbf{W}^k = \mathbf{B}^k \mathbf{D}^{k^{-1/2}} \mathbf{V}^k \tag{3.68}$$

where $\mathbf{V}^k$ and $\mathbf{D}^k$ are the matrices composed by the eigenvectors and eigenvalues of the class covariance matrix, and $\mathbf{B}^k$ the ICA unmixing matrix. See [17] for more details. Through CC-ICA, we have a space where all class-conditional probabilities can be assumed independent or at least, where the error involved with working with the marginal probabilities instead of the whole distribution is minimized.

We can use any Bayesian classifier based on the extracted features through equation (3.66). But this is pointless: taking into account the independence assumption we observe that a modified naive Bayes is the natural choice when working with CC-ICA: CC-ICA has in naive Bayes a naturally associated classifier. We can see this by using our ICA representation (3.61) into the MAP solution (3.66),

$$C_{MAP} = \arg\max{}_{k=1,\ldots K} |\det \mathbf{W}^k| \prod_{m=1}^{M^k} p^k(s_m) P(C^k) \tag{3.69}$$

If dimensionality is sufficiently high the product on the right-hand side of this equation, generally made up of values lower than 1, will be very close to zero, so the logarithm of the likelihoods will be used whenever possible. Also, unless stated otherwise, classes will be considered equiprobable so the MAP solution becomes the maximum likelihood (ML) solution,

$$C_{ML} = \arg\max{}_{k=1,\ldots K} \sum_{m=1}^{M^k} \log p^k(s_m) + \log |\det \mathbf{W}^k| \tag{3.70}$$

The constant in the right-hand side of the equation can be regarded as a normalizing constant, necessary to compare conditional probabilities calculated in different representations. We can further estimate this constant considering that

$$|\det \mathbf{W}^k| = |\det \mathbf{B}^k||\det \mathbf{D}^{k^{-1/2}}||\det \mathbf{V}^k| = \prod_{m=1}^{M^k} \frac{1}{\sqrt{\lambda_m^k}} \tag{3.71}$$

where λ_m^k are the eigenvalues of the covariance matrix of class C^k. See [17] for more details. To this point, we have assumed that the dimensionality of the original data (D) which will usually be a large value, is equal to the dimensionality in the CC-ICA representations (M^k). This is not always the case since PCA whitening generally conveys some kind of dimensionality reduction making it impossible to directly calculate the determinant of a no longer square matrix. In this case, we can assume that the conditional distribution of the measurements is approximated by the distribution of the principal components. This approximation can be arbitrarily good if a sufficient number of components are considered since the reconstruction error is bounded by the sum of the eigenvalues and tends to zero. By replacing (3.71) into (3.70) we have that samples will be classified using the version of naive Bayes,

$$C_{ML} = \arg\max{}_{k=1,\ldots K} \sum_{m=1}^{M^k} \log p^k(s_m) - \frac{1}{2} \sum_{m=1}^{M^k} \log \lambda_m^k \tag{3.72}$$

which can be written as

$$C_{ML} = \arg\max{}_{k=1,\ldots K} \sum_{m=1}^{M^k} \log p^k(s_m) + v^k \tag{3.73}$$

with $v^k = -0.5 \sum_m^{M^k} \log \lambda_m^k$.

We arrived to this modified naive Bayes classifier first by stating the advantages ICA poses for density estimation, then motivating the need for a class-conditional approach and finally by replacing the conditional independence assumptions on the conditional densities within the Bayesian classification scheme. If we reverse the reasoning, results are identical but maybe, easier to understand. Let us suppose we want to improve naive Bayes. One of the possible ways for doing this might be reinforcing the class-conditional independence assumption made by this classifier. This can be done using ICA on each of the classes, and we are back to where we started. The goodness of our method should be measured, besides in terms of absolute performance when compared with other classifiers, in terms of improvement it represents for naive Bayes. This will be shown in the following experimental results.

However, our classifier (3.72) still requires the estimation of the densities $p^k(s_m)$, where in this case $s_m = \mathbf{w}_m^{k^T}(\mathbf{x} - \bar{\mathbf{x}}^{\mathbf{k}})$ with $\mathbf{w}_m^k$ the m-th row of the filter matrix $\mathbf{W}^k, \mathbf{x} \in C^k$ and $\bar{\mathbf{x}}^{\mathbf{k}}$ the class mean. This estimation is simplified not only by being unidimensional but also by prior information we have on the independent components. We know that the independent components have zero mean and unit variance. We also know they are highly nongaussian, and we can easily find out if they sub or super Gaussian. All this knowledge can restrict density estimation to particular families. It is not the scope of this thesis to find the best family of estimators for the independent components but we will explain one of the possible estimators. See [17] for detailed information about more complex estimators. Here, we will consider the problem of estimating $p(s)$ where s is a zero mean, unit variance nongaussian random variable.

One of the most common nonparametric density estimation techniques is the Gaussian kernel approach, that adapted to our current situation results in

$$p(s) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\sqrt{(2\pi)}\sigma} exp^{-\frac{1}{2}\frac{(s-s_n)^2}{\sigma}} \tag{3.74}$$

where s_n are each of the N samples in the training set (in CC-ICA, the components of the training set with a certain class label). The kernel width can be selected as $\sigma = [\frac{12}{N}]^{\frac{1}{5}}$ as suggested in [131]. In the case of sparse data, this kernel method can cause the probability to drop to zero. This can be solved by increasing the value of σ at the cost of eventually over-smoothing the estimate.

If the independent components are very sparse, a quite robust parametrization which provides an accurate approximation of very sparse data was introduced by Hivarinen in [65],

$$p(s) = \frac{1}{2} \frac{(\alpha+2)[\alpha(\alpha+1)/2]^{(\alpha/2+1)}}{\sqrt{\alpha(\alpha+1)/2} + |s|^{(\alpha+3)}} \tag{3.75}$$

as $\alpha \to \infty$ this approaches the Laplace density. The parameters are estimated as follows:

$$\alpha = \frac{2 - k + \sqrt{k(k+4)}}{2k-1} \tag{3.76}$$

where $k = p(0)^2$ and $p(0)$ can be estimated using a suitable kernel.

Another family of density estimators can be considered, the semiparametric models. Since we work with unidimensional data, Gaussian mixture models can be used and the model can be estimated through the expectation-maximization (EM) algorithm [38]. In our case,

$$p(s) = \sum_{j=1}^{J} p(j) \frac{1}{\sqrt{(2\pi)}\sigma_j} exp^{-\frac{1}{2}\frac{(s-\mu_j)^2}{\sigma_j}} \tag{3.77}$$

Gaussian mixture models might require far too many mixture components in order to accurately estimate a highly supergaussian component. So other, more appropiate distributions can be used within the mixture. The EM can also be used to estimate a *mixture of Laplacians.* A mixture of two zero-mean Laplace distributions proves easy to estimate and highly adaptive to strong variations in the level of sparsity. These would be modelled as,

$$p(s) = \sum_{j=1}^{2} p(j) \frac{1}{\sqrt{2}\alpha_j} e^{-\frac{\sqrt{2}|s|}{\alpha_j}} \tag{3.78}$$

These three proposed methods, the nonparametric, the parametric and the semiparametric can be combined. In practice, all these approaches, if adequately estimated, yield approximate results. In cases where strong variations of nongaussianity are present, the parametric approaches were greatly affected. Also, performance of the kernel approach was very sensible to the choice of an adequate kernel width. So generally the semiparametric approach was taken.

3.6.3 Application of ICA/CC-ICA

One of the main drawbacks of using local color distributions as salient features is the difficulty of constructing a good model, due to their high dimensionality. For example, if we consider a single 8-bin histogram per color spectrum resulting feature dimensionality is $8^3 = 512$ as done in the previous experiments. A first approach is to classify data using metric techniques such as nearest neighbor techniques. And if we consider probabilistic classifiers, high dimensionality forces nonparametric density estimators such as Gaussian kernels or naive Bayes. Here, once we presented Independent Component Analysis (ICA) and the Class-Conditional Independent Component Analysis, we show how we can use these techniques to improve classical approaches to recognition in a pharmaceutical object recognition system. This object image database has been previously used in section (3.1). We obtain a set of perceptually salient keypoints of pharmaceutical products using the well-known Harris operator as described in section (2.2). Then, local color histograms are extracted as described in section (3.1).

We will use CC-ICA in the context of object classification of pharmaceutical products. This can be compared with the results obtained in section (3.1). There, global and local approaches are used to classify pharmaceutical products but we have seen that results were not very satisfactory. Here, we use a larger database of pharmaceutical products. More precisely, we will use 400 pharmaceutical products. This database is shown in appendix (C.1). However, the use of CC-ICA implies to manage with robust probability density functions. So that, we will require a considerable amount of data vectors. Is for this reason we increased the amount of data vectors by taking into account the neighborhoods of the detected keypoints, and extracting more sample histograms from these neighborhoods.

Up to this point, we are able to represent an object $\mathbf{H}$ (i.e. an image) belonging to one of K possible classes C^k, through the local histograms extracted from its L detected keypoints and neighbors $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_L]$. Now, L is about $L = 500$. The training histograms for a certain class consists of the representative histograms for all the training objects of the class. For instance, given 2 images of a given object, with maybe 100 and 150 selected local histograms, respectively. We are representing this image with 250 512-dimensional samples.

In order to compare histograms, we chose the χ^2 distance because it has been extensively used for histogram comparisons (see section 2.1.3). We have to remember that in section (3.1) we used L_2 between local color histograms which is also used in this experiment. Also, it seems natural to use this distance within the nearest neighbor approach, which can be adapted to our case making use of a voting scheme: given an image of an object $\mathbf{H}_{Test}$, with L representative histograms, calculate the distance of these histograms to all histograms in the training set and assign the most voted class label to this object.

Within the Bayesian context, if the local histograms are assumed independent and the priors equiprobable, then we can apply the Maximum A Posteriori (MAP) rule for this particular problem results in,

$$C^{ML} = \arg\max_{k=1\ldots K} p(\mathbf{H}|C^k) = \arg\max_{k=1\ldots K} \prod_{l=1}^{L} p(\mathbf{h}_l|C^k) \tag{3.79}$$

The class-conditional probabilities $p(\mathbf{h}_l|C^k)$ can be estimated using several methods but we have to consider the limitations exposed to the estimation by the high dimensionality of $\mathbf{h}_l$. In general, these high-dimensional situations restrict this estimation to nonparametric kernel methods, because other approaches usually impose too many restrictions on the data, for instance Gaussianity [17]. Also, semiparametric methods and their estimation algorithms become increasingly nonstable with dimensionality. But the precision in the estimation of the class-conditional probabilities is decisive on the performance of the classifier.

Two experiments were performed, the first one illustrates the properties of the CC-ICA representation for color distributions, mainly independence and sparsity. In the second experiment we test CC-ICA classification for a large set of pharmaceutical products and compare this scheme with other nonparametric and probabilistic approaches.

For the first experiment we used two images of similar objects except for the color distribution, as can be seen in figure (3.22). The images correspond to two milk boxes of the same brand. The full cream milk has predominant rose tones and we will refer to it as f-milk. The semi skimmed milk box has mainly green tones and we will refer to it as ss-milk. We have chosen these two objects because though they have the same color in a large portion of the image, their design is almost identical, and each of them contains a specific color tonality. Ideally and for a particular object, the ICA representation will provide a sparse coding for the color distribution of this object. As mentioned, this means that when a test histogram is recognized as belonging to this object it will have values close to zero in most of the components, and consequently have a high probability. For this experiment, a dataset of 144 representative 8-bin color histograms ($D = 512$) was extracted from both images using a predefined grid. Dimension was reduced from the original color histogram space of 512 dimensions to 35 using PCA and preserving a 99.9% of the total variation of the original data. We obtained the ICA representation for the f-milk and estimated the one dimensional densities corresponding to each independent component using a mixture of 2 zero-mean Laplacians.

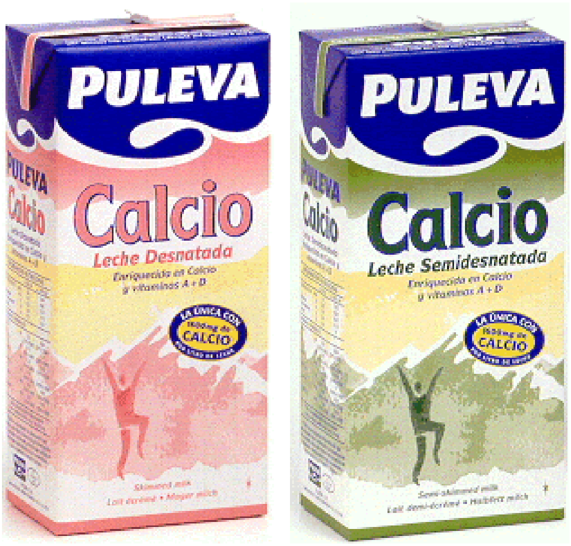


Figure 3.22: Full cream milk box with predominant rose tones (f-milk) and semi skimmed milk box with predominant green tones (ss-milk).

We then manually selected a component from the f-milk ICA representation to illustrate independence and sparsity. Manual selection has to be performed because the ICA representation does not provide a natural hierarchy on its components. In this case we chose indpendent component number 19 due to the fact it represents a connected color distribution inside the object. This makes visualization more clear, but any other component would do. Figure (3.23) illustrates the activations of this component. The straight line in figure (3.23.a) shows the value of component 19 for the representative histograms of the f-milk. These values correspond to a sparse distribution (concentrated around zero), and from these values, the density of the component was estimated. The dotted line in the same figure shows the value of component 19 when the representative histograms of the ss-milk are projected into the f-milk ICA representation. The ss-milk histograms randomly activate this component, yielding a low probability in the sparse distribution learnt from the f-milk histograms.

From figure (3.23.a) we can deduce that the projection of histogram 108 of the f-milk has the highest absolute value on component 19. Figure (3.23.b) plots the

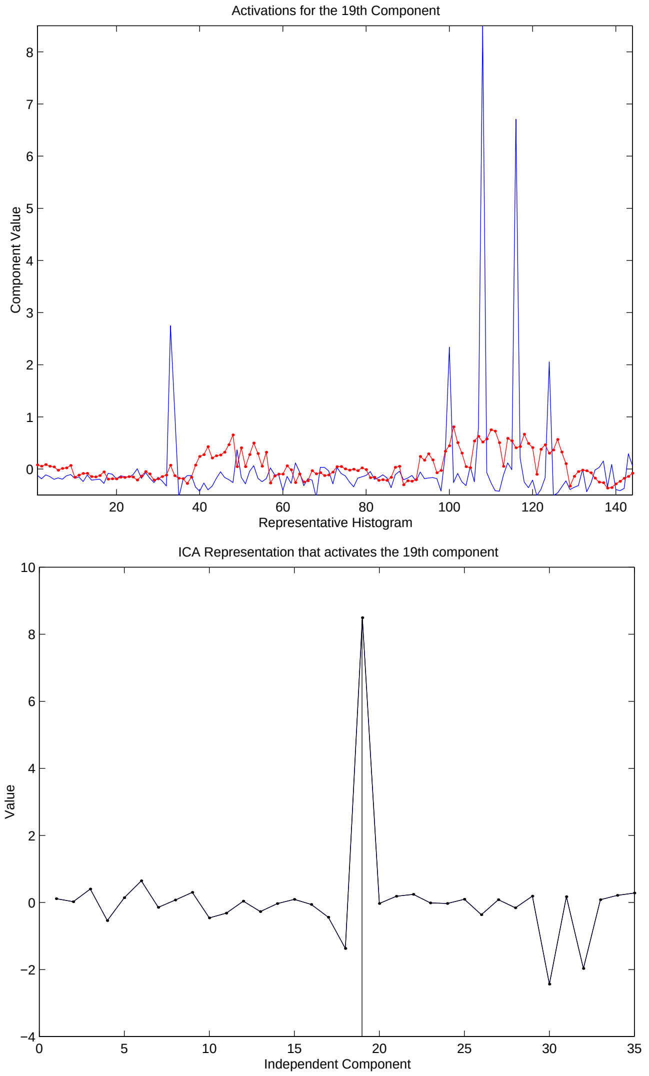
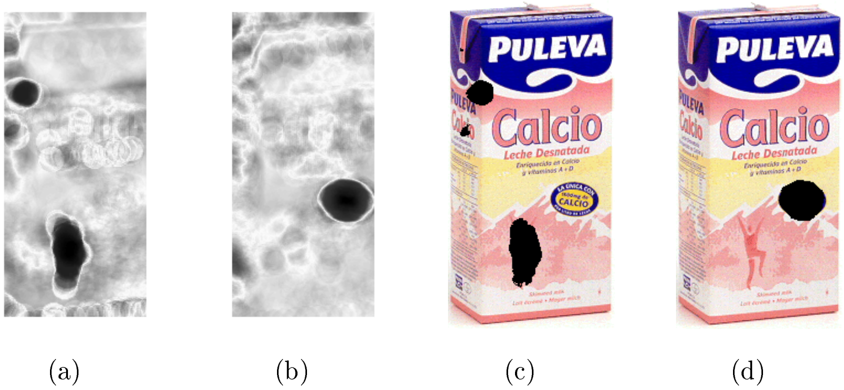


Figure 3.23: Shows the value of the 19^{th} independent component for the representative histograms of the full milk image (continuous line) and for the representative histograms of the semi skimmed milk image (line with dots). (b) Shows how this component is the only one activated upon the appearance of a certain color distribution in the full milk image.

projection values of all the other components of this histogram. Since most of them are near zero, the probability for this histogram will be high. Histogram 108 is then a highly representative histogram for the f-milk. This is confirmed when we find out that histogram 108 corresponds to a neighborhood inside the human figure of th f-milk. ICA is gathering in the 19^{th} component the dark pink that corresponds to the color distribution of the human figure in the f-milk image.

In figure (3.24.a) and (3.24.b) the probability map for components 19 and 21 of the f-milk ICA representation are shown. These maps were calculated projecting the color histogram in a neighborhood of every point of the image in this representation and then calculating the probability of this projection. So low probability values correspond to color distributions that activate the components. It can be seen how component 19 effectively captures the color distribution surrounding the human figure, while component 21 captures the color distribution around an ellipsoidal blue and yellow tag shared by both images. Figures (3.24.c) and (3.24.d) are the result of multiplying the image of the f-milk with a threshold of the probability map.



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

Figure 3.24: (a,b) Probability maps for component 19, activated by the color distribution of the human figure, and component 21 activated by the color distribution of the "calcio" tag. (c,d) Thresholded probability map shown over original image.

With this experiment it is observed the way the ICA representation separates color distributions in the input data, providing a sparse coding for each of these separated components. When we try to code an unlearnt color distribution with this coding, sparsity is lost so probabilities drop. It is observed how this sparse coding can be effectively used for distinguishing the most dissimilar and unique regions between objects.

The second experiment shows the CC-ICA performance for object recognition using local color histograms. For this experiment we counted with 2400 images of 400 different pharmaceutical products (the classes) with dark background. There a total of six images per class. These products present several color ambiguities so there is a lot of class overlap (some products are very similar only differing in reduced regions) and, in all the images, the background color is black and the illumination controlled. Figures (C.2,C.3,C.4,C.5) show an instance of each pharmaceutical product used in the experiments.

From the six instances, five were used to train our statistical models in order to increase the accuracy of the probability estimation. From each image, we extracted a large amount of representative 8-bin local color histograms (around 150 interesting local regions per image) with the explained keypoint detector. So 400 ICA models were estimated using CC-ICA from approximately 750 samples per class. The results of the classification are presented in table (3.33) where we can check that CC-ICA outperforms all the other statistical classification methods. The results are presented in terms of ranks where, for example, rank five means that the test product was correctly classified within the top five probabilities. For instance, ICA classified correctly 99.0% of the test images, if the top four positions are considered (rank 4). Particularly interesting is comparison with the Gaussian kernel approach to density estimation: any difference between this classifier and the CC-ICA approach can only be blamed on the level of accuracy of the density estimation. We also compare the local approach with the global approach in order to point out the advantages of searching for local cues. A nearest neighbor technique with euclidean distance was used for classification (NNL2). In our problem, the background can be easily subtracted so both cases, with and without background, were considered. As expected, not considering the background performed better. But neither performed comparably to the local approaches. Actually, the recognition rates of the global approach is what led us to focus this particular problem with local strategies. Table (3.33) also includes these results.

For our object recognition experiment, the estimation of the 400 projection matrices took around 20 hours on a dual Pentium III with 850 Mhz, and the density estimation only fifteen minutes. Testing is straightforward since for each test object, K projections are needed and a simple algebraic operation obtains the probability on each component of the projected data. The probabilities are then added and compared for classification.

Method	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
10 Nearest Neighbour L2 based technique	90.25%	91.75%	95.50%	96.25%	97.25%
10 Nearest Neighbour χ^2 based technique	92.25%	94.25%	96.75%	97.25%	98.50%
Gaussian kernel based technique	89.75%	91.25%	93.50%	96.00%	96.50%
Class-conditional ICA	**96.00%**	**98.25%**	**98.75%**	**99.00%**	**99.00%**
Global histograms (NNL2 - with background)	80.50%	83.50%	85.25%	86.00%	86.00%
Global Histograms (NNL2 - no background)	82.75%	85.00%	86.00%	86.50%	86.50%

Table 3.33: Recognition percentages in the first five places for the six compared classification methods in the object recognition experiment.

This experiment demonstrates how we can work in the original space of local color histograms by estimating a set of probability density functions. ICA helps us to estimate these probability density functions because we will be able to assume each component to be independent. And as seen in table (3.33), the performance obtained using CC-ICA is better than other approaches. So that, we believe that this approach can be used in those problems where several object classes should be distinguished among them and data vectors lie in a high dimensional space.

3.7 Conclusions

This thesis is focused on the problem of object representation and classification using local features. At the beginning of this chaper, we presented an experiment where a local approach is compared with a holistic one. Thanks to this experiment, we realized that the outperformance of local approaches with respect to global ones is a real evidence.

Then, we focused the rest of the chapter on exposing different nonsupervised linear transformations and examined their advantages and drawbacks. PCA, NMF, WNMF and ICA have been explained in order to solve several encountered drawbacks. At this point we should be able to state, at least experimentally, if classification can benefit from each of these methods. In the case of PCA we observe that this technique is "blind" beyond statistics of second order, restricting its capacity to learn complex data. The main advantage of PCA is its ability to reduce dimensionality, preserving the reconstruction error. We can conclude that statistical classification can benefit from PCA since classification can be more accurate and less computationally demandind on low dimensions and, if low variance directions correspond to noise, discarding them does not affect classification.

The way NMF might benefit classification comes from a whole different standpoint. NMF can be seen as a technique to preserve the mean square error (MSE) as PCA but assuming positive data without orthogonal components. It can be said that reconstruction is worse than PCA because NMF is only restricted to positive data. PCA can be used for general data but we have seen that feature vectors described in terms of positive data are better represented using NMF. Local color histograms have been used along this thesis and are a good example of positive descriptions to be used with NMF.

We have experimentally tested that NMF is not well suited for not uniformly data distributions. Then, we introduced a weighted version of NMF that overcomes this disadvantage. Since feature vectors which come from local data distributions are usually not uniformly distributed, WNMF outperforms NMF in such a scenario. Several tests have been performed which compare NMF with respect to WNMF. From this set of tests, we can conclude stating that WNMF outperforms NMF if data vectors come from a not uniformly data distribution. This outperformance is only manifested when the subspace is expressed using a relative amount of dimensions. For low dimensional subspaces, NMF is still better than WNMF. Also, experiments demonstrate that after learning a representation using WNMF, we can take this solution to continue learning with NMF and the final solution would be better than using only NMF.

We have empirically tested this set of linear transformations in different contexts. Firstly, we used the reconstruction distances to perform object classification. Taking advantage that these linear transformations come out from different assumptions, we are able to merge them in a unique classifier. We realize that PCA is well suited for compact data classes and NMF for dispersed ones. Then, using the projected coefficients of the subspaces described by each linear transformation, we tried to estimate the probability density function of the original space. In the same context as in the previous experiment, we merged both techniques in order to take advantage of the positive aspects of each of them. This parametric modelization performs better than using direct reconstruction distances.

Finally, a nonparametric approach is used to perform object classification. Then, we needed to define a metric distance to be used with NMF. We performed an extended analysis using a database of handwritten digits where we compared NMF and PCA using several metric distances. When occlusions are not present in our data set, Earth Mover's Distance seems very appropiate to be used with NMF. However, the computational costs required for such a combination (NMF and EMD) are explosive. When occlusions are present in our data set, the cosine metric distance seems very appropiate to be used with NMF as it outperfoms PCA. Here is where we state that PCA can not be used under the presence of occlusions and NMF is inherently adapted for such problem since it is a parts-based representation. As we occluded handwritten digits using handmade occlusions, we performed an extended analysis using face images in order to evaluate the robustness of NMF in front of natural occlusions (individuals wearing sunglasses and/or a scarf) and decide upon the reliability of PCA with respect to NMF. This experiment demonstrates that NMF outperforms PCA. We believe that this outperformance is due to the reliance of NMF on a parts-based representation.

PCA, NMF and WNMF are three nonsupervised linear transformations that help us to reduce the dimensionality of a high dimensional feature space. Then, using the coefficients of the subspace we are able to perform classification and alleviate the curse of dimensionality problem. We have evaluated different strategies to perform object classification using this dimensionality reduction. In the particular case that we want to work directly with the original feature vectors, that is, to work in a high dimensional space we can use the formulation of naive bayes. As seen, results are not very satisfactory and we introduced Independent Component Analysis (ICA) to overcome this problem. We formulated ICA in the context of distinguishing between data classes as an extension of the bayesian formulation. So that, we formulated ICA in order to make use of the class-conditional densities, this is the so-called Class-Conditional ICA (CC-ICA). We present an experiment where we used CC-ICA to estimate probability density functions of 400 object classes and results are very satisfactory since we achieve the best recognition results.