

**Research Article**

# **Analysis of Linear and Nonlinear Dimensionality Reduction Methods for Gender Classification of Face Images**

SAMARASENA BUCHALA\*†, NEIL DAVEY†, TIM M GALE†‡, RAY J FRANK†

† Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK.  
Email: {S.Buchala, N.Davey, T.Gale, R.J.Frank}@herts.ac.uk

‡ Department of Psychiatry, QEII Hospital, Welwyn Garden City, AL7 4HQ, UK.

Data in many real world applications are high dimensional and learning algorithms like neural networks may have problems in handling high dimensional data. However, the Intrinsic Dimension is often much less than the original dimension of the data. Here, we use fractal based methods to estimate the Intrinsic Dimension and show that a nonlinear projection method called Curvilinear Component Analysis can effectively reduce the original dimension to the Intrinsic Dimension. We apply this approach for dimensionality reduction of the face images data and use neural network classifiers for Gender Classification.

*Keywords:* Intrinsic Dimension; Dimensionality Reduction; Curvilinear Component Analysis; Principal Component Analysis; Gender Classification

## **1 Introduction**

High dimensional data usually contain redundancies and may have many irrelevant variables. Classifiers like neural networks may need huge networks, with many free parameters, to cover the high dimensional data. Networks, on such datasets, even if successfully trained, often perform badly on their test sets. This bad generalization may be due to the large number of free parameters representing irrelevant information. To learn relevant information from such datasets, a large number of datapoints would be needed, which is often impractical, and the training time needed for learning also increases to a great extent. This problem with high dimensional data is often referred in the literature as “curse of dimensionality” (Bellman, 1961).

The intrinsic dimension which is the true dimension, of the data may be much smaller than the original data dimension. The problem with high dimensional data can be circumvented by reducing the data to its Intrinsic Dimension.

Principal Component Analysis (PCA) (Jolliffe, 2002) and Independent Component Analysis (ICA) (Comon, 1994) are linear projection methods and are the most popular statistical methods for dimensionality reduction. Being linear methods, they work perfectly well on the linear data. However real world data are often nonlinear, in which case linear techniques are not appropriate. Here, we use a powerful recent nonlinear projection method, Curvilinear Component Analysis (CCA), for dimensionality reduction and show that it is possible to reduce the data to its Intrinsic Dimension. We apply this technique on face images data and use two classifiers, Multi Layer Perceptron (MLP), and Support Vector Machine (SVM) with a linear kernel, for Gender Classification. We also investigate different methods for estimating Intrinsic Dimension.

## 2 Intrinsic Dimension

Intrinsic Dimension (ID) can be defined as the minimum number of free variables required to define the data without any significant information loss.

Due to correlations among the data, linear and nonlinear, a  $D$  dimensional data set may actually lie on a  $d$  dimensional manifold ( $D \geq d$ ) and the ID of such data is said to be  $d$ . For example a plane embedded in a three-dimensional space, as shown in Figure 1(a) has an ID of 2, as the two axes variables are linearly dependent. Figure 1(b) shows the well known three dimensional horseshoe data distribution. However, any point in the data can be defined by a linear axis and a curvilinear axis, indicating that it's ID is 2.

Dimensionality reduction algorithms reduce the data to a user defined dimension but do not inform about the number of dimensions the data should be reduced to. ID estimation can be a prior step to dimensionality reduction. There are a few methods in the literature, for estimating the ID, which are mainly based on the fractal dimension. As the name suggests a fractal dimension can be a non integer value. The box counting dimension, information dimension and the correlation dimension are the popular fractal methods.

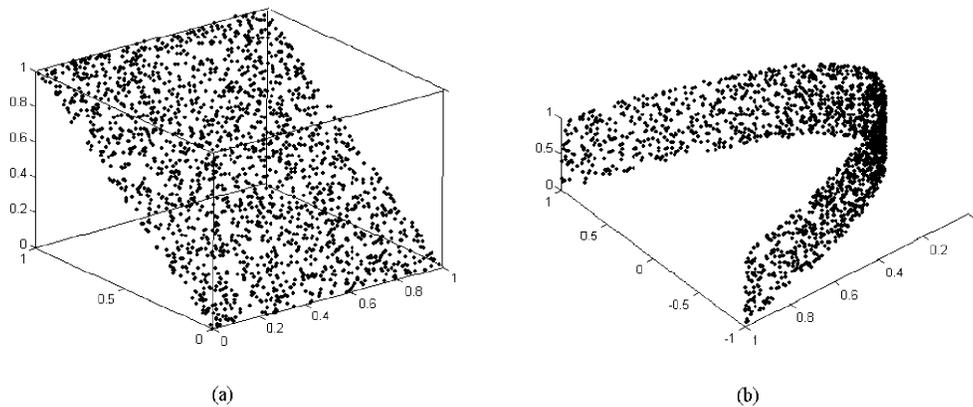


Figure 1. (a) A two dimensional plane embedded in a three dimensional space has an ID value of 2. (b) Three dimensional horseshoe data distribution with an ID value of 2.

### 2.1 Box counting dimension

ID estimation using different approaches leads to different solutions, depending upon the complexity of the problem, and there is no unique definition of dimension. According to Camastra (Camastra & Vinciarelli, 2001), the so called box counting dimension also known as the capacity dimension (Mandelbrot, 1977) is the most popular one.

The box counting dimension is based on the following idea (Baker & Gollub, 1990): consider a one-dimensional line of length  $r$ . The number of one-dimensional boxes,  $V(l)$ , of length  $l$ , required to cover this line can be given as  $r(1/l)$ . Similarly, to cover a two dimensional box of length  $r$  the number of two-dimensional boxes required is  $r^2(1/l)^2$ . In general, for a  $d$  dimensional figure the number of boxes needed would be  $r^d(1/l)^d$ .

Taking logarithms gives the following equation.

$$d = \frac{\log V(l)}{\log r + \log(1/l)} \tag{1}$$

In the limit of small  $l$ , the term involving  $r$  becomes negligible. The box counting dimension can then be written as

$$d_B = \lim_{l \rightarrow 0} \frac{\log V(l)}{\log(1/l)} \tag{2}$$

The above equation can be approximated by measuring the slope of the plot of the logarithmic values of the number of required boxes (non-empty boxes), to cover the data, and the inverse of the edge length of the box. The illustration of box counting method on the horse shoe data distribution of 2000 data points is shown in Figure 2. The data space is covered by a three dimensional box as shown in Figure 2(a). The box is divided into 8 boxes as shown in Figure 2(b) and further into 64 boxes as shown in Figure 2(c) and the number of non-empty boxes is calculated at each stage. Figure 2(d) shows the box counting plot for the horse shoe data. The slope of the linear part of the curve, which is the box counting dimension, is calculated as 1.9792.

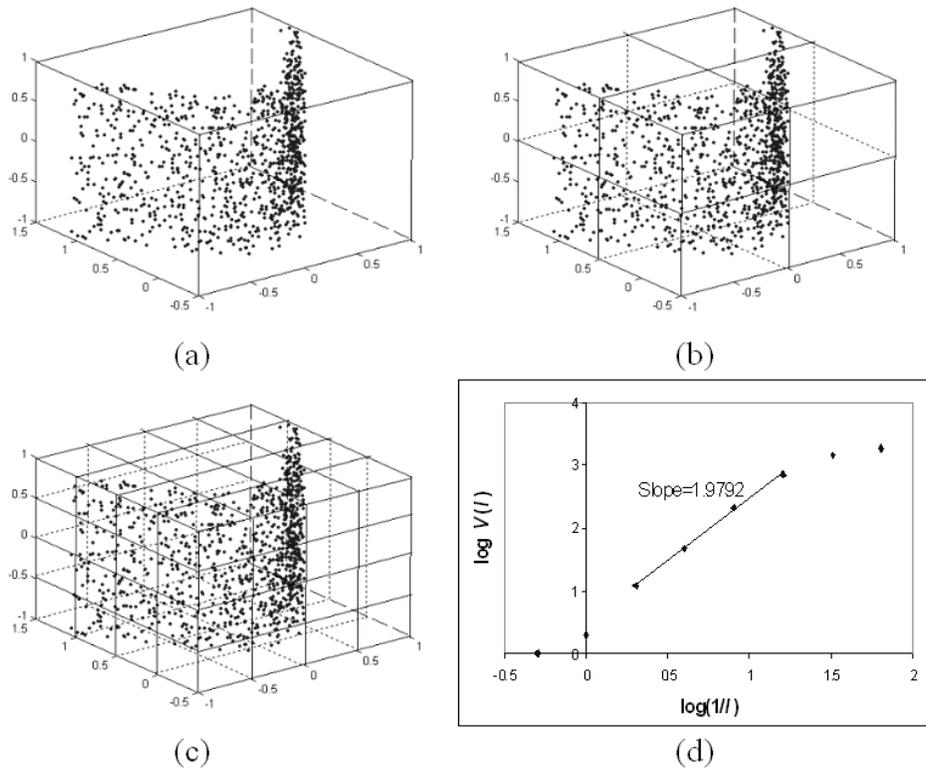


Figure 2. (a) A box of edge length 2 covering the whole horse shoe distribution. (b) The box is divided into 8 equal boxes. (d) The box is further divided into 64 equal boxes (c) A plot of logarithmic values of the number of non-empty boxes and the inverse of the edge length of the box.

## 2.2 Information dimension

The Box counting dimension, though it analyzes the geometrical structure of the data, ignores the distribution of the points on this structure. It discounts the information of the number of points in the box. The Information Dimension overcomes this problem. It uses Shannon’s information formula to quantify the information conveyed by one box (Theiler, 1990). The Average Information can be given by the following equation:

$$I(l) = -\sum_{i=1}^M p_i \log p_i \quad (3)$$

where  $P_i$  is the probability that a point falls in the  $i^{th}$  box, and can be defined as  $P_i = N_i / N$  if  $N$  is the total number of points and  $N_i$  the total number of points in the  $i^{th}$  box.

The Information Dimension is then defined as:

$$d_l = \lim_{l \rightarrow 0} \frac{I(l)}{\log(1/l)} \quad (4)$$

The above equation can be approximated by measuring the slope of the plot of the logarithmic values of the Average Information and the inverse of the edge length of the box. Figure 3 shows the Information Dimension plot of a 2000 data point horseshoe distribution. The Information Dimension is the slope of the linear part of the curve and is measured at 1.9211.

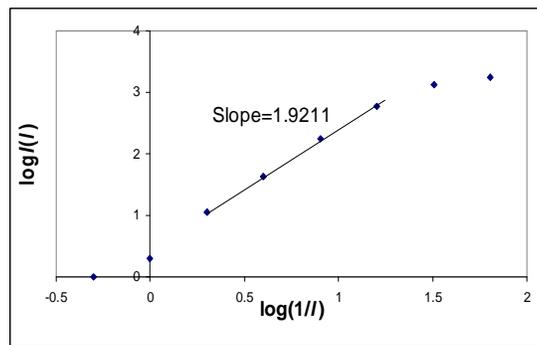


Figure 3. Information Dimension plot of the horse shoe data. It is calculated as the slope of the linear part of the curve.

In the case of uniform distribution of points, where each box holds equal number of points, the Information Dimension is equivalent to Box Counting Dimension.

### 2.3 Correlation dimension

Both the Box Counting Dimension and the Information Dimension are impractical to implement on high dimensional data due to the computational load involved, as the number of boxes increases exponentially with the dimensionality of the data. The correlation dimension (Grassberger & Procaccia, 1983) developed by Grassberger and Procaccia is a better alternative for high dimensional data. This method assumes that the data is spatially correlated. A measure of this property is called the *correlation integral*  $C(l)$ . It can be calculated by using Equation (5).

$$c(l) = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N K \quad \text{where} \quad \begin{cases} K = 1, & \text{if } d_{i,j}^X \leq l \\ K = 0, & \text{if } d_{i,j}^X > l \end{cases} \quad (5)$$

Where  $N$  is the number of data points,  $l$  is the length variable and  $d_{i,j}^X$  is the Euclidean distance between points  $x_i$  and  $x_j$  in the dataset  $X$ .

The idea is that in a  $d$  dimensional dataset, the total number of pair wise points closer to each other than length  $l$  is proportional to  $l$  raised to  $d$ . From this assumption the correlation dimension  $d$  can be calculated from Equation (6).

$$d_c = \lim_{l \rightarrow 0} \frac{\log(C(l))}{\log(l)} \tag{6}$$

The above equation can be approximated by calculating the slope of the graph plotted of the logarithmic values of the Correlation Integral and length  $l$ . Figure 4(a) shows the Correlation Dimension plot of a 2000 data point horseshoe distribution. The correlation Dimension shown in Figure 4(b) is the slope of the linear part of the curve shown in Figure 4(a), and is measured at 1.8768.

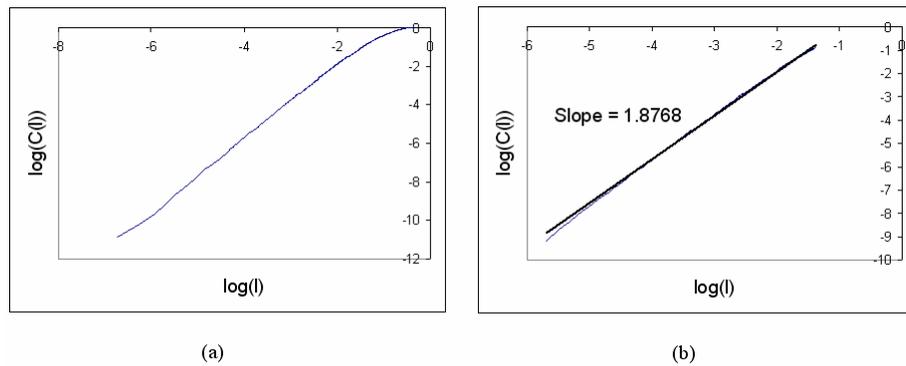


Figure 4. (a) Correlation Dimension plot of the horse shoe data. (b) The Correlation Dimension is calculated as the slope of the linear part of the curve.

The three dimensions discussed are shown to be related by the following inequality:

$$d_c \leq d_l \leq d_B \tag{7}$$

The values of the three dimensions are usually closer in practice and if the points are uniformly distributed, the equality of Equation (7) holds (Grassberger & Procaccia, 1983). The three dimensions calculated for the horseshoe data also satisfies the above inequality.

Accurate ID estimation of most real world data, including our face images data, is difficult because of the availability of only few data points and the noise in the data. However this does not mean that ID estimation is not useful. A rough estimation of ID can be done by using any of the above methods, where after few dimension values near to the estimated ID are considered and the dimension which gives the best result can be selected as the true dimension of the data.

### 3 Dimensionality reduction

Many techniques for dimensionality reduction have been proposed in the literature. However, Principal Component Analysis (PCA) (Jolliffe, 2002) and more recently Independent Component Analysis (ICA) (Comon, 1994) are the ones mostly used. PCA, especially, is a well known technique in the field of Face Recognition (Sirovich & Kirby, 1987), (Turk & Pentland, 1991). These are linear techniques and look for linear dependencies in the data. Nonlinear methods such as Nonlinear Multidimensional Scaling (Shepard & Carroll, 1965) and Sammon’s Nonlinear Mapping (Sammon, 1969) have the ability to reduce the

dimensionality of nonlinear data. However, these methods suffer from huge computational costs and the inability to unfold strongly nonlinear data (Demartines & Herault, 1997). We use a recent algorithm called Curvilinear Component Analysis (CCA) proposed by Demartines and Herault (Demartines & Herault, 1997), which overcomes some of the shortcomings of the other mentioned methods and has the ability to reduce the dimensionality of strongly nonlinear data.

### 3.1 Curvilinear component analysis

The structure of the CCA network consists of two layers, the first one of which performs vector quantization on the dataset and the second layer called the projection layer performs a topographic mapping of the structure obtained by the vector quantization layer. The projection layer is a free space, which takes the shape of the submanifold of the data.

While dimensionality reduction methods reduce the dimension of the data, vector quantization methods reduce the number of data points. The main purpose of vector quantization in CCA is to reduce the computational cost. As our face images dataset is relatively small (400 faces), we do not perform vector quantization and hence we discuss, here, only the projection part of the CCA.

The idea of CCA is to preserve distances in the input and output spaces; all the possible distances between points in the input space should match the respective distances in the output space. However, preservation of larger distances may not be possible in the case of nonlinear data, as a global unfolding of the manifold is required to reduce the dimension. In this case, it is important that at least local (smaller) distances should be preserved. For this, CCA uses a neighbourhood function which ensures the condition of distance matching is satisfied for smaller distances while it is relaxed for larger distances. Preservation of smaller distances (local mapping), may then lead to the stretching of larger distances (global unfolding).

The projection layer of CCA minimizes an error function which is given as

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (d_{i,j}^X - d_{i,j}^Y)^2 F_{\lambda}(d_{i,j}^Y) \quad \forall j \neq i \quad (8)$$

Where  $d_{i,j}^X$  and  $d_{i,j}^Y$  are the Euclidean distances between points  $i$  and  $j$  in the input space  $X$  and output space  $Y$  respectively.  $F_{\lambda}(d_{i,j}^Y)$  is the neighbourhood function, selected such that it favours smaller distances over larger ones. Minimizing the error function with respect to the point  $Y_i$  in the output space by a normal stochastic gradient would give the following update rule.

$$\nabla Y_i = \alpha(t) \sum_{j=1}^N \frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y} [2F_{\lambda}(d_{i,j}^Y) - (d_{i,j}^X - d_{i,j}^Y)F'_{\lambda}(d_{i,j}^Y)](Y_i - Y_j) \quad \forall j \neq i \quad (9)$$

$\alpha(t)$ , the learning rate, and the neighbourhood function  $F_{\lambda}(d_{i,j}^Y)$  can be time varying.

The stochastic gradient update method of Equation (9) can be conceived as selecting a point  $Y_i$  in the output space, while the remaining points are pinned. The selected point is moved (updated) according to the average influence of all the pinned points. This method of updating has the following drawbacks (Demartines & Herault, 1997).

1. The computational cost is of the order of  $O(N^2)$  as all the possible  $N(N-1)/2$  distances need to be calculated at each time step.

2. The sum of all influences may lead to an averaging effect, which leads to a small update amount resulting in slow convergence.

For these reasons CCA uses a different update method, where the selected point is pinned while the remaining points are moved according to its influence. Then, by ignoring the derivative part of Equation (9), the update rule of CCA can be written as:

$$\nabla Y_j = \alpha(t) F_\lambda(d_{i,j}^Y) \frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y} (Y_j - Y_i) \quad \forall j \neq i \quad (10)$$

The algorithm for projection of the training data can be summarized as follows.

Calculate the Euclidean distances between all pairs of points in the input space.

Initialize the points in the output space randomly or using PCA.

Initialize epoch  $t=0$

For each epoch  $t$ ,

Begin

Calculate  $\bullet(t)$  and  $\bullet$ .

For each point  $r_j$  in the output space,

Begin

$$Y_j = Y_j + \nabla Y_j \quad \text{where} \quad \nabla Y_j = \alpha(t) F_\lambda(d_{i,j}^Y) \frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y} (Y_j - Y_i) \quad \forall j \neq i$$

End

Increment  $t$

End

Mapping of a new point (test data) from the input space  $X$  to the output space  $Y$ , in CCA, involves reducing the error function of Equation (8) and is iterative in the same sense as the actual learning process. However, the update rule is the stochastic gradient of Equation (9) without the derivative part. The algorithm for projecting a new point can be summarized as follows.

Calculate the Euclidean distances between the new test point and all the training points.

Initialize the test point in the output space randomly or using PCA.

Initialize epoch  $t=0$ .

For each epoch  $t$ ,

Begin

Calculate  $\bullet(t)$  and  $\bullet$ .

$$Y_i = Y_i + \nabla Y_i \quad \text{where} \quad \nabla Y_i = \alpha(t) \sum_{j=1}^N F_\lambda(d_{i,j}^Y) \frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y} (Y_i - Y_j) \quad \forall j \neq i$$

Increment  $t$

End

We use the first few variables obtained by the PCA projection, for initialization of the points in the output space. This initialization, rather than a random one, induces some prior information about the submanifold of the data. The learning rate and the neighbourhood width are calculated as an exponential decay.

### 3.1.1 Projection evaluation

The quality of a projection can be evaluated by the “ $dy-dx$ ” representation (Demartines & Herault, 1997). It is a plot of all the possible distances in the input space,  $dx$ 's, versus their respective distances in the output space,  $dy$ 's. For a linear projection the “ $dy-dx$ ” plot should be linear. Figure 5(a) shows the projection of the plane in a three dimensional space, of Figure 1(a), in a two dimensional space. This “ $dy-dx$ ” plot, shown in Figure 5(b) indicates a linear projection as the  $dy$ 's and  $dx$ 's are proportional at all scales. However, for a nonlinear projection a complete distance match at all scales may not be possible. Figure 6(a) shows the projection of the horseshoe data, of Figure 1(b), in a two dimensional space. The projection is nonlinear with only small  $dy$ 's matching  $dx$ 's, shown in Figure 6(b). Unfolding can be observed as ( $dy > dx$ ) occurring for larger distances.

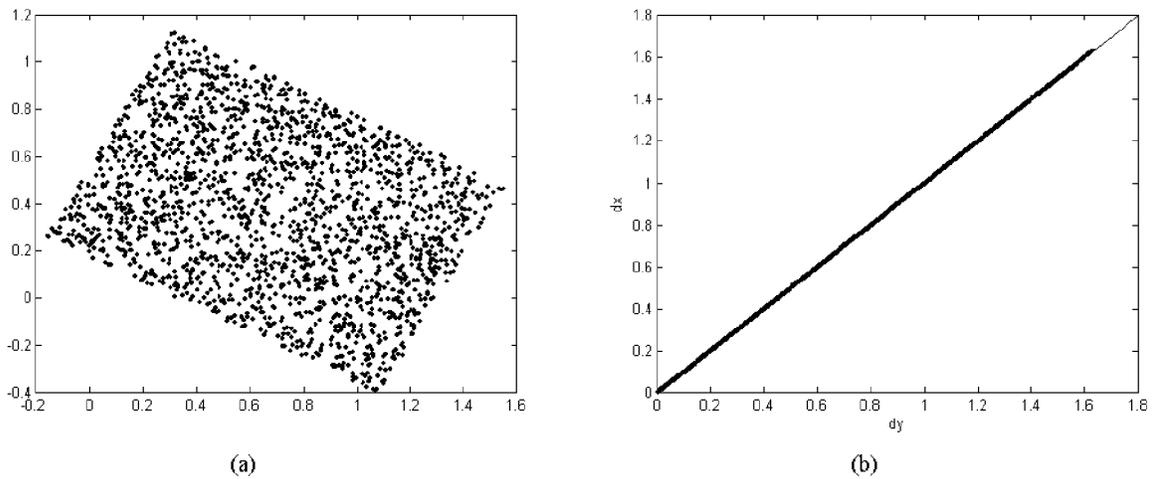


Figure 5. (a) Projection of Figure 1(a) from a three dimensional space to a two dimensional space by CCA (b) The “ $dy-dx$ ” representation indicates a complete linear projection with no unfolding.

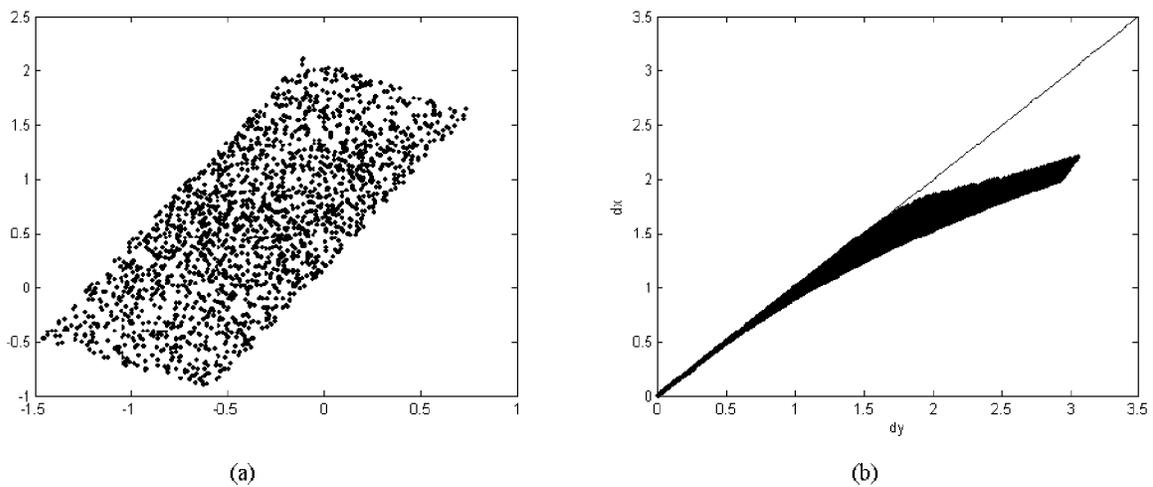


Figure 6. (a) Projection of the horseshoe data of Figure 1(b) from a three dimensional space to a two dimensional space by CCA. (b) The “ $dy-dx$ ” representation indicates a nonlinear projection with unfolding ( $dy > dx$ ) occurring at higher scales.

#### 4 Gender classification

Two different datasets are used. The first one consists of 100 distinct adult, frontal face grey scale images (50 male and 50 female). The face images are from the following databases: AR (Martiniz & Benavente, 1998), FERET (Phillips *et al.*, 1998), and JAFFE (Lyons *et al.*, 1998). The dataset consists of faces of different races and age groups, taken under different lighting conditions. Some examples are shown in Figure 7.



Figure 7. Examples of the raw face images of *dataset1*

Taking the midpoint of the two eyes as a reference point a  $60 \times 90$  part is extracted from each of the  $128 \times 128$  face images. Histogram equalization is applied on the extracted images to normalize for different lighting conditions. Some of the extracted and histogram equalized faces are shown in Figure 8. The dimensionality of this dataset is 5400. We refer to this resultant dataset as *dataset1*.



Figure 8. The first row faces are the  $60 \times 90$  extractions of the original  $128 \times 128$  face images. The second row shows face images after histogram equalization.

The second dataset, a much larger dataset, was used by Sun *et al.* (Sun *et al.*, 2002). This dataset consists of 400 adult frontal face grey scale images (200 male and 200 female) each with  $100 \times 100$  resolution. All face images were warped to the same scale, orientation and position, by geometric alignment of centres of the eyes and corners of the mouth. Histogram Equalization was then applied on the resultant images. The dimensionality of this dataset is 10000. We refer to this dataset as *dataset2*.

##### 4.1 Dataset1

This dataset is randomly divided into 5 subsets, with each subset having 80 (40 male and 40 female) for training and 20 (10 male and 10 female) for testing. The test sets are not overlapped with their respective training sets and other test sets.

Intrinsic Dimensionality of this dataset is calculated using the *correlation dimension* method discussed in Section 2.3. As stated earlier, ID estimation of real world data is difficult. Figure 9(a) shows the Correlation

Dimension plot for *dataset1*. As the plot is not linear like the plot of the horseshoe data shown in Figure 4, we select different intervals and measure the slope of the linear fit of that interval. The ID values from these plots of different intervals are different; the ID estimation of the plot in Figure 9(b) is 7 while it is 11 in both Figure 9(c) and 9(d). We select the worst case dimension 11 as the Intrinsic Dimension of the *dataset1*.

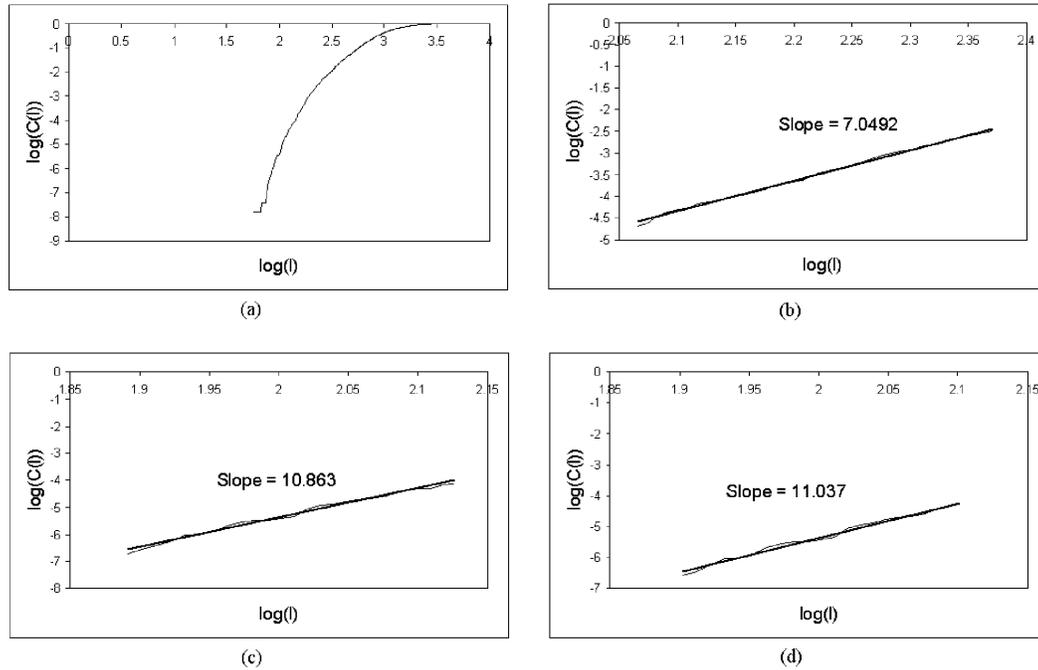


Figure 9. (a) The Correlation Dimension plot for *dataset1*. The ID estimation varies at different intervals. The interval taken in (b) gives ID as 7, while (c) and (d) gives ID as 11.

As the ID value of 11 selected for this dataset can be considered as only a rough estimation, we tried different dimensions around this value.

We use two classifiers, a MLP and a SVM with a linear kernel (Other kernels like Radial Basis function and Polynomial function are tried, but linear kernel produced better results). The average error rates over the 5 sets on this dataset, for different CCA dimensions are shown in Table 1. The error on CCA data with 6 dimensions was quite high with both MLP and SVM and the error went down as the dimension is increased. The minimum dimension with optimum result is 14. The “*dy-dx*” plots of the CCA projections, shown in Figure 10 can explain the results of Table 1. Figure 10(a) shows CCA projection to 6 dimensions. The plot is distorted, with distance linearity occurring only at very small distances, indicating a bad projection. The projection quality improves as the dimension is increased. Figure 10(d) and 10(e) with CCA projections to 14 and 16 dimensions respectively, has distance linearity occurring at larger distances.

Table 1. Average Error Rates Over 5 Testsets of *Dataset1*, with Different CCA Dimensions

Method	MLP (%)	SVM (%)
CCA-6	40	43
CCA-8	31	28
CCA-10	31	24
CCA-12	28	24
CCA-14	23	17
CCA-15	26	17
CCA-16	25	20
CCA-18	26	19

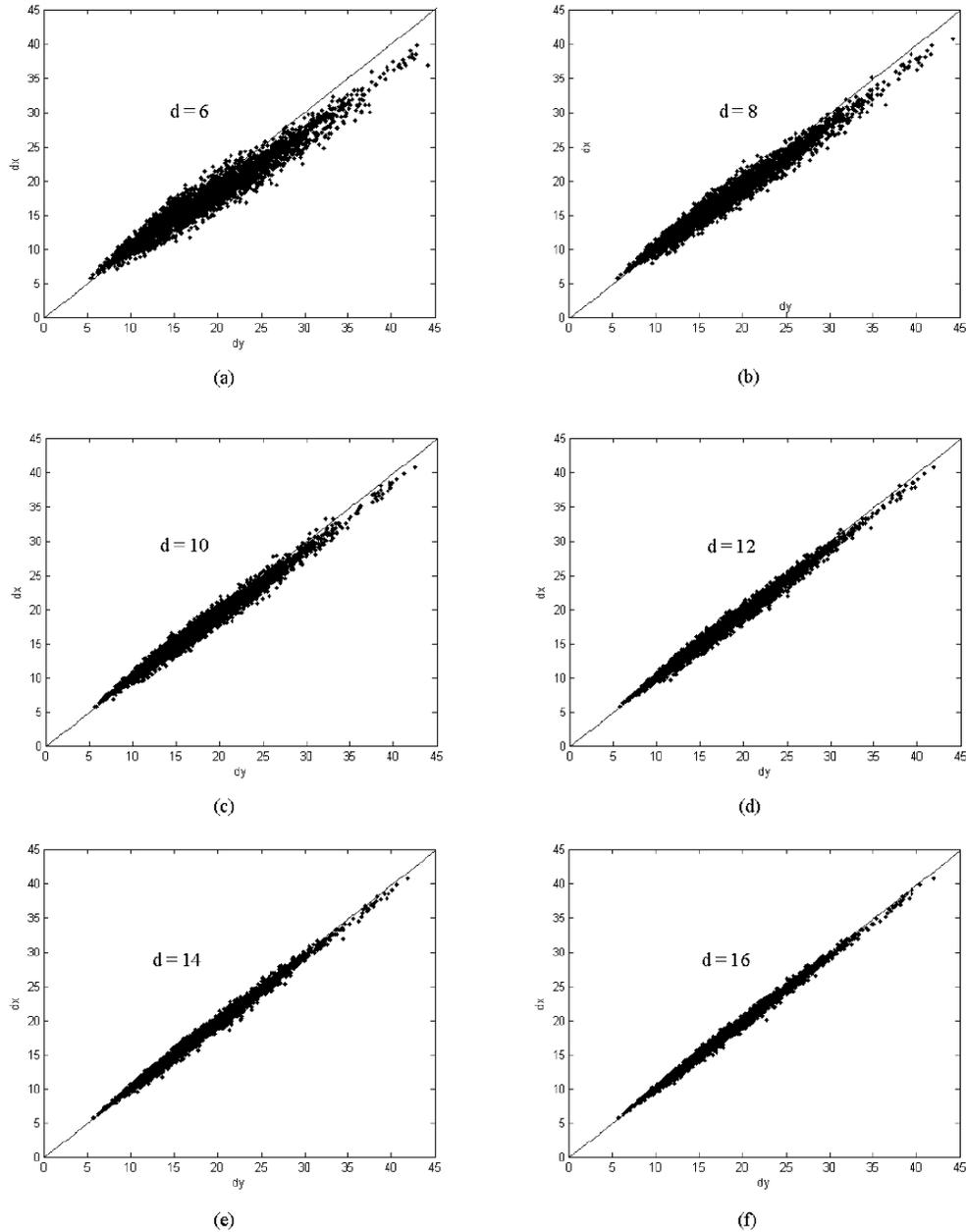


Figure 10. The “ $dy-dx$ ” plots of CCA projections to (a) 6 dimensions (b) 8 dimensions (c) 10 dimensions (d) 12 dimensions (e) 14 dimensions (f) 16 dimensions. Projection to 6 dimensions is more distorted with only few starting small distances being linear. The projection quality improves as the dimension is increased.

For comparison, we tried classification on data obtained by PCA reduction. For an  $N$  data point dataset, there will be  $N-1$  meaningful *Principal Components*. More details can be found in (Sirovich & Kirby, 1987). As this dataset has 80 faces there will be 79 meaningful principal components. However the first 67 components accounted for 95% of the total variance of the data. By projecting the data onto these 67 components we were able to reduce the 5400 dimensional data to a 67 dimensional data. We refer to this data as PCA-67 data. We also tried classification on the actual data, without any dimensionality reduction, and we refer to this data as RAW data. Table 2 shows that both dimensionality reduction approaches produced better results than the RAW data, with PCA-67 faring better than CCA-14. For comparison, another PCA reduction to 14

dimensions is obtained, by projecting the data onto the first 14 components. We refer to this as the PCA-14 data. The classification in Table 2 shows that PCA-14 performance is worse than that on the RAW data. The “ $dy-dx$ ” plots of PCA-67 and PCA-14 data shown in Figure 11 explains their performance. The PCA-14 plot, in Figure 11(b), shows mismatch of distances at all scales. It can also be seen, from Table 2, that the SVM gave a better classification than the MLP.

Table 2. Average Error Rates Over 5 Testsets of *Dataset1*

Method	MLP (%)	SVM (%)
RAW	30	27
PCA-67	19	12
CCA-14	23	17
PCA-14	34	32

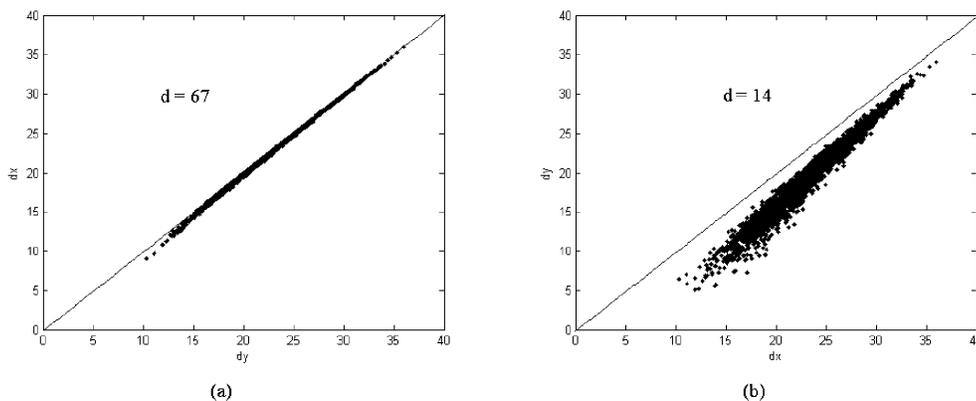


Figure 11. The “ $dy-dx$ ” plots of PCA projections to (a) 67 dimensions (b) 14 dimensions. The plot is distorted for 14 dimensions with nonlinearity of distances at all scales, while the plot for 67 dimensions is mostly linear.

#### 4.2 Dataset2

This dataset is divided into 5 subsets with each subset having 320 faces (160 male and 160 female) for training, 50 faces (25 male and 25 female) for testing, and 30 faces (15 male and 15 female) as a validation set. The validation sets are used for stopping criteria for the training of the MLP.

A rough ID estimation was performed, similar to the process for *dataset1*. The ID is measured as approximately 14. Again different CCA dimensions are tried as shown in Table 3. The PCA gave 273 components accounting for 95% of the total variance of the data. The projection of the data onto these 273 components resulted in a 273 dimensional data. We refer to this data as PCA-273. Table 3 shows the average error rates over 5 testsets. It can be seen that the average error rates for CCA above 10 dimensions is similar, however the minimum dimension with optimum result is CCA-12. The performance of the PCA-273 is similar to that of CCA-12. There is not much difference in the classification performances of the MLP and SVM.

Table 3. Average Error Rates Over 5 Testsets of *Dataset2*

Method	MLP (%)	SVM (%)
PCA-273	6.55	6.25
CCA-6	13.75	11.25
CCA-8	9.5	9.25
CCA-10	8.25	8
CCA-12	6.75	7
CCA-14	6.75	7.5
CCA-16	6.5	7.5
CCA-18	8.25	7
CCA-20	8.75	7.5
CCA-22	7.75	7.5

## 5 Discussion

PCA projection to account for 95% of the total variance of the data resulted in 67 dimensions for *dataset1*, while it resulted in 273 dimensions for *dataset2*. As the number of data points increases, the number of such PCA dimensions also increases. This, however, does not necessarily mean that the ID also increases, and our results show similar ID estimation for both datasets. CCA is able to successfully reduce the original dimension to the ID for both datasets.

If the “*dy-dx*” plot of the CCA-14 of *dataset1*, Figure 10(e) is considered, the projection can be seen as reasonably linear, with no strong unfolding ( $dy > dx$ ). The larger distances in the original dataspace are replicated with good fidelity in the output space. This indicates the projection of the data in a 14 dimensional space by CCA is not strongly nonlinear. In contrast the PCA projection in a 14 dimensional space is highly distorted as shown by Figure 11(b). This shows the inability of the PCA to deal with even slight nonlinearities.

CCA favours smaller distances over larger ones. It can be seen in all the “*dy-dx*” plots of Figure 10, that smaller distances in both input and output spaces are matched. Even in a distorted plot of Figure 10(a), there are few small distances that are matched. PCA projections, in contrast, seem to favour larger distances. The “*dy-dx*” plot of PCA projection in a 14 dimensional space, shown in Figure 11(b), shows distortion at all scales. However, the smaller distances are more distorted than the larger distances. Even in a fairly uniform PCA projection in a 67 dimensional space, shown in Figure 11(a), there is a slight mismatch in smaller distances. This may suggest a bad local mapping by PCA.

Three different fractal methods, which consider different properties of data to estimate the Intrinsic Dimension, are also investigated. Correlation Dimension, which considers spatial correlation property of the data, is found to be feasible in the case of high dimensional data, due to its relatively lesser computational cost.

Finally, based on our experiments, we make the following conclusions:

1. The ID of our face images data is much lower than their original dimension.
2. Linear methods like PCA are unable to effectively reduce the nonlinear data to its ID, whereas nonlinear methods like CCA can effectively do this.
3. Classification in the ID space works.

## **Acknowledgment**

We thank Zehang Sun for providing with the face images data which we referred as *dataset2* in our experiments.

## **References**

- G. L. Baker and J. P. Gollub, *Chaotic dynamics an introduction*, ed., Cambridge University Press, 1990.
- R. E. Bellman, *Adaptive control processes: A guided tour*, ed., Princeton University Press, 1961.
- F. Camastra, and A. Vinciarelli, "Intrinsic dimension estimation of data: An approach based on grassberger-proccaccia's algorithm", *Neural Processing Letters*, **14** (1), 2001, pp. 27-34.
- P. Comon, Independent component analysis -A new concept? *Signal Processing*, **36**, 1994, pp. 287-314.
- P. Demartines, and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets", *IEEE Transactions on Neural Networks*, **8** (1), 1997, pp. 148-154.
- P. Grassberger and I. Proccacia, "Measuring the strangeness of strange attractors", *Physica D*, **9**, 1983, pp.189-208.
- I. T. Jolliffe, *Principal Component Analysis*, 2nd. ed. New York, Springer-Verlag, 2002.
- M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets", *IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 1998.
- B. Mandelbrot, *Fractals: Form, chance and dimensions*, ed. San Francisco, W H Freeman & Co., 1977.
- A. M. Martiniz and R. Benavente, The AR face database, CVC, 1998.
- P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms", *Image and Vision Computing*, **16** (5), 1998, pp. 295-306.
- J. W. Sammon, "A nonlinear mapping algorithm for data structure analysis", *IEEE transactions Computers*, **C-18** (5), 1969, pp. 401-409.
- R. N. Shepard and J. D. Carroll, "Parametric representation of nonlinear data structures", in Krishnaiah, P. R. (Ed.) *International Symposium on Multivariate Analysis*, Academic Press, 1965.
- L. Sirovich and M. Kirby, "Low -dimensional procedure for the characterization of human faces", *Journal of the Optical Society of America A*, **4**, 1987, pp. 519-524.
- Z. Sun, X. Yuan, G. Bebis, and S. J. Louis, Neural-Network-based gender classification using genetic search for eigen-feature selection. *IEEE international joint conference on neural networks*, 2002.

J. Theiler, "Estimating fractal dimension", *Journal of the Optical Society of America A-Optics and Image Science*, 7 (6). 1990, pp. 1055-1073.

M. Turk and A. Pentland, "EigenFaces for recognition", *J.Cognitive Neuroscience*, 3, 1991, pp. 71-86.

### Biography

	<p><b>Samarasena Buchala</b> received a B.Tech degree in Metallurgical Engineering from the Regional Engineering College Warangal, India, presently known as National Institute of Technology, Warangal in 1999. He received an MSc from the Robert Gordon University, UK in 2002. Currently he is working towards a PhD in the School of Computer Science, University of Hertfordshire, UK.</p> <p>His PhD is an inter-disciplinary project that investigates computational methodologies, for modeling psychological theories of face perception and categorisation, with a specific emphasis on gender. His primary research interests are face perception, visual cognition, pattern recognition, and machine learning.</p>
	<p><b>Neil Davey</b> received a BSc degree in Mathematics from the University of Manchester in 1978, an MSc in Mathematical Logic from the University of London in 1979, an MSc in Computing from Brunel University in 1989 and a PhD from the University of Hertfordshire in 2004.</p> <p>He is a Lecturer at the University of Hertfordshire, where he undertakes research into Connectionism and Neural Networks.</p>
	<p><b>Tim M. Tim Gale</b> received a PhD in Psychology from the University of Hertfordshire. He is currently Research Lead for Hertfordshire Partnership NHS Trust and is a Visiting Research Fellow in the Department of Computer Science, University of Hertfordshire.</p> <p>His principal interest is in the processes underlying visual object recognition and this work involves work with normal human subjects, brain-injured patients and computational models. He also has interests in abnormal psychology (dementia, obsessive compulsive disorder and depression) and in cognitive psychology (biases in risk perception, knowledge representation) and is actively involved in research collaborations under all these areas.</p>
	<p><b>Ray J. Frank</b> received a B.Sc degree in Physics from Bedford College, University of London in 1973 and M.Sc's in Astrophysics and Computer Science form Queen Mary College, London in 1977 and City University, London in 1986. He has been a Principal Lecturer in Computer Science at the University of Hertfordshire since 1992. He was seconded to Nortel U.K under the Department of Trade and Industry (DTI U.K.) Senior Academic in Industry Scheme (SAIS) awards of 1992-93.</p> <p>He is the author and co-author of numerous papers and co-author of a book on computer programming as well as co-inventor for two patents. He is also a consultant for Great North Eastern Railways in computer control systems. His teaching interest are in neural networks, cognitive modeling and real time systems His research interests include neural networks, time series processing and genetic algorithms.</p>