Dimensionality Estimation of Datasets

Fahad Ashraf

***Abstract*—Dimensionality estimations of datasets is an important problem in the field of pattern recognition and knowledge discovery. In this paper, different methods for estimating intrinsic dimensionality are discussed with focus on global methods specifically fractal-based methods. A fractal-based approach using the Grassberger-Procaccia (GP) algorithm is discussed and as the GP algorithm doesn't preform well on high dimensionality datasets, an empirical procedure that improves the orignal alogrithm has been described.**

I. Introduction

In pattern recognition problems, data is represented as vectors of dimension d. The data is then embedded in the vector space $\mathbb{R}^d$ but this does not mean that the intrinsic dimensionality (ID) of the dataset is d. The ID of a data set is the minimum number of free variables needed to represent the data without information loss.

II. Local Methods

Local methods try to estimate the topological dimension of the data manifold. The definition of topological dimension was given by Brouwer [1] in 1913. Topological dimension is the basis dimension of the local linear approximation of the hypersurface on which the data resides, i.e. the tangent space. The topological dimension is the number of dimensions of the tangent space at each point. For example, a sphear has a two-dimensional tangent space that can also be viewd as a two-dimensional manifold and since the ID of a sphear is three, the topological dimension represent the lower bound of ID. Sometimes the topological is simply called local dimension, this is why the methods which estimate topological dimensions are called local methods. The basic algorithm to estimate the topological dimension was proposed by Fukunaga and Olsen [2]. Other approaches to the Fukunaga-Olsen's algorithm have been proposed to estimate locally ID. Among them the Near Neighbor Algorithm [3] and the methods based on Topological Representing Networks (TRN) [4].

A. Fukunaga-Olsen's algorithm

The alogrithm is based on the observation that for the vectors embedded in a linear subspace, the dimension is equal to the number of non-zero eigenvalues of the covariance matrix. The basic idea of the algorithm is to examine the data in many small subregions and from this estimate the intrinsic dimensionality. For each region the eigenvalues of the local convariance matrix are computed. Then, The eigenvalues are normalized by dividing them by the largest eigenvalue. The intrinsic dimensionality is then defined as the number of normalized eigenvalues that are larger that a threshold T. The values T is based on heuristic approach such as 0.05 and 0.01. It is not possible to fix the threshold values which would be best for every problem.

B. The Near Neighbor Algorithm

The use of near neighbor algorthm to estimate ID was first done by Trunk [5]. This method consist to following steps An initial value of an integer parameter k is chosen and the k nearest neighbors to each patternthe given data set are identified. The subspace spanning the vectors from the i^{th} pattern to its k nearest neighbors is constructed for all patterns. The angle between the $(k+1)^{th}$ near neighbor of pattern i and the subspace constructed for pattern i is then computed for all i. If the average of these angles is below a threshold, ID is k. Otherwise, k is incremented by 1 and the processrepeated. The drawback of Trunk's method is that a fixed value for the threshold can not be determined.

C. TRN-based methods

Topology Representing Network (TRN) is a unsupervised neural network proposed by Martinetz and Schulten [4]. They proved that TRN are optimal topology preserving maps i.e TRN preserves in the map the topology originally present in the data. Bruske and Sommer [6] proposed to improve Fukunaga-Olsen's algorithm using TRN in order to perform the Voronoi tesselation of the data space. The algorithm proposed by Bruske and Sommer consist of the following process. An optimal topology preserving map G, using TRN, is computed. Then, for each neuron $i \in G$, a PCA is performed on the set Qi consisting of the differences between the neuron i and all of its mi closest neurons in G. Bruske-Sommer's algorithm shares with Fukunaga-Olsen's one the same limitations: since none of the eigenvalues of the covariance matrix will be null due to noise, it is necessary to use heuristic thresholds in order to decide whether an eigen-value is significant or not.

III. Global Methods

Global methods try to estimate the ID by making use of the whole dataset and the main difference between local methods and global methods is that local methods rely only on the information contained in the neighborhood of each data sample where as global methods make use of the whole data set.

Global methods can be grouped in three big families: Projection techniques, Multidimensional Scaling Methods and Fractal-Based Methods.

A. Projection Techniques

Projection methods rely on projecting the data on the best subspace and minimizing projection error. These methods can be divided into two groups: linear and non-linear.

Principal Component Analysis (PCA) [7], [8] is a widely used linear method. PCA projects the data along the directions of maximal variance. The method consists of computing

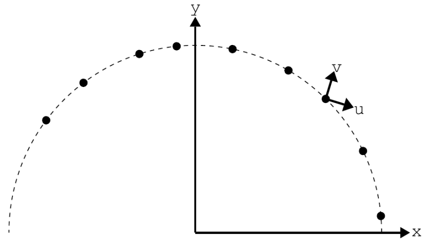

Fig. 1. Ω Data Set. The data set is formed by points lying on the upper semicirconference of equation $x^2 + y^2 = 1$. The ID of Ω is 1. Neverthless PCA yields two non-null eigenvalues. The principal components are indicated by u and v.

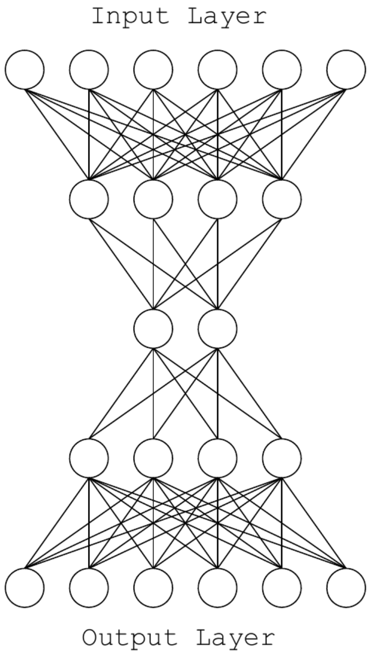

Fig. 2. A Neural Net for Nonlinear PCA

eigenvalues and eigenvectors of the covariance matrix of data. Each of the eigenvectors is called a principal component. ID is given by the number of the non-null eigenvalues. But there are some problems, PCA is an inadequate estimator, since it tends to overestimate the ID [17]. As shown in Fig. 1, a data set formed by points lying on a circumference for PCA has dimension 2 rather than 1.

To solve this problem, non-linear algorithms have been proposed. There are two different approaches to get a non-linear PCA: an autoassociative approach (Nonlinear PCA) [5,18] and the one based on the use of Mercer kernels (Kernel PCA) [19]. Nonlinear PCA is performed by means of a five-layers neural network. The neural net has a typical bottleneck structure, shown in Fig. 2.

The first (input) and the last (output) layer have the same number of neurons, while the remaining hidden layers have less neuron than the first and the last ones. The second, the third and the fourth layer are called respectively mapping, bottle-neck and demapping layer. Mapping and demapping layers have usually the same number of neurons. The number of the neurons of the bottleneck layer provides an ID estimate. The targets used to train Nonlinear PCA are simply the input vector themselves. The network is trained with the backpropagation algorithm, minimizing the square error. As optimization algorithm, the conjugate-gradient algorithm [9] is generally used. Though nonlinear PCA performs better than linear PCA in some contexts [10], it presents drawbacks when estimating ID. As underlined by Malthouse [11], the projections onto curves and surfaces are suboptimal. Besides, NLPCA cannot model curves or surfaces that intersect themselves. Kernel PCA consists of making a nonlinear projection of the data set, by means of an appropriate positive definite function (Mercer kernel) [12] in a new space (Feature Space). Then the eigenvalues of the covariance matrix in the Feature Space are computed and ID is given by the number of the non-null eigenvalues. The performance of the method is heavily influenced by the kernel choice [13]. Moreover, due to the data noise, last eigenvalues, even if very small, are not null. Therefore it is necessary to ignore the eigenvalues whose magnitude is lower than a threshold value that can be only fixed in a heuristic way. Among projection techniques it is worth mentioning the Whitney reduction network recently proposed by Broomhead and Kirby [7], [14]. This method is based on Whitney's concept of good projection [26], namely a projection obtained by means of an injective mapping. An injective mapping between two sets U and V is a mapping that associate a unique element of V to each element of U . As pointed out in [7], finding projections, by means of injective mappings, can be difficult and can sometimes involve empirical considerations.

B. Multidimensional Scaling Methods

Multidimensional Scaling (MDS) [27,28] methods are projection techniques that tend to preserve the distance among data as much as possible. Therefore data that are close in the original data set should be projected in such a way that their projections, in the new space (output space), are still close. Among multidimensional scaling algorithms, the best known example is MDSCAL, by Kruskal [29] and Shepard [30]. The criterion for the goodness of the projection used by MDSCAL is the $stress$.

This depends only on the distances between data. When the rank order of the distances in the output space is the same as the rank order of the distances in the original data space then the $stress$ is zero.

Kruskal's stress S_K is:

$$S_K = \Big[\frac{\sum_{i<j} [rank(d(x_i,x_j)) - rank(D(x_i,x_j))]^2}{\sum_{i<j} rank(d(x_i,x_j))^2} \Big]^{\frac{1}{2}} \qquad (1)$$

where $d(x_i, x_j)$ is the distance between the data x_i and x_j and the $D(x_i, x_j$ is the distance of the projections of the same data in the output space. When the stress is zero a perfect projection exists. Stress is minimized by iteratively moving the data in the output space from their initially randomly chosen positions according to a gradient-descent algorithm.

The intrinsic dimensionality is determined in the following way. The minimum stress for projections of different dimensionalities is computed. Then a plot of the minimum stress versus dimensionality of the output space is performed. ID is the dimensionality value for which there is a knee or a flattening of the curve. Kruskal and Shepard's algorithm presents a main drawback. The knee or the flattening of the curve could not exists.

IV. Fractal-Based Methods

Fractal dimension is a statistical quantity that gives an indication of how completely a geometric object appears to fill space as one zooms down to finer and finer scales. It is commonly used in image analysis [33] and chaos theory [41]. Fractal-based techniques are global methods that have been successfully ap- plied to estimate the attractor dimension of the underlying dynamic system generating time series [44]. It has also been applied in machine learning and data mining [2], [4]. The idea behind that, is to use the fractal dimension of a data set as an estimate of its intrinsic dimension. As defined by Fukunaga [9], if all elements of a d-dimensional data set lie entirely within an m-dimensional subspace then the data set has an intrinsic dimension of $m(m < d)$. From this, it can also be said that only m independent variables are requried to describe the data set without any information loss. A number of methods can be used for fractal dimension calculation but the most popular ones are $Box - Counting$ and $Correlation$ dimension.

A. Box-Counting Dimension

The box-counting dimension is a simplified version of Haussdorff dimension [5]. The Box-Counting dimension D_B of a set Ω is defined as follows: if $v(r)$ is the number of the boxes of size r needed to cover Ω, then D_B is

$$D_B = \lim_{r \to 0} \frac{\ln(v(r))}{\ln(\frac{1}{r})} \tag{2}$$

Unfortunately, the box-counting dimension can be computed only for low-dimensional sets because the algorithmic complexity grows exponentially with the set dimension.

B. Correlational Dimension

Correlational dimension is a good alternative to the $Box - Counting$ method. Due to its computational simplicity, the Correlation dimension is successfully used to estimate the dimension of attractors of dynamical systems. The Correlation dimension is defined as follows:
let $\Omega = X_1, X_2, X_3, ..., X_N$ be a set of points in $\mathbb{R}_n$ of cardinality N. If the correlation integral $C_m(r)$ is defined as:

$$C_m(r) = \lim_{N \to \infty} \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I(\parallel X_j - X_i \parallel \leq r) \tag{3}$$

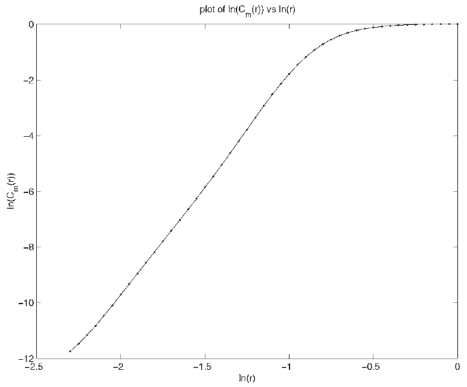

Fig. 3. Plot $\ln(C_m(r))$ versus $\ln(r)$

where I is an indicator function then the Correlation dimension D of Ω is:

$$D = \lim_{r \to 0} \frac{\ln(C_m(r))}{\ln(r)} \tag{4}$$

It has been proven [11] that the correlation dimension is a lower bound of the box-counting dimension, but, because of noise, the difference between the two is negligible in real applications. Some methods [28], [27] have been studied to obtain an optimal estimate for the correlation dimension, but all of these techniques work only when the correlation integral assumes the given form in the equation (3). These methods generally require some heuristics to set the radius r [30]. Therefore, Camastra and Vinciarelli [1S] used the original procedure (GP algorithm) proposed by Grassberger and Procaccia that consists of plotting $\ln(C_m(r))$ versus $\ln(r)$ and measuring the slope of the linear part of the curve (3).

It has been proven [6], [26] that, in order to get an accurate estimate of the dimension D, the set cardinality N has to satisfy the following inequality

$$D = 2 \log_{10} N \tag{5}$$

Inequality (5) shows that the number N of data points needed to accurately estimate the dimension of a D-dimensional set is at least $D^{\frac{10}{2}}$. But this leads to huge values of N. The effect of N on the measure of the dimension can be seen in Table I.

This table reports the value of the measures of D obtained using the GP algorithm over sets of points randomly distributed in 10-dimensional hypercubes (supposed to have $D = 10$). The dimension is measured for different values of N and the error with respect to the supposed true dimension is reduced by increasing the number of data points from 10^3 to $10^{\frac{10}{2}} = 10^5$. To handle this issue and improve the reliability of the measure for low values of N, Camastra and Vinciarelli proposed an empirical procedure as described in the below section.

TABLE I. DEPENDENCE OF THE ESTIMATED CORRELATION DIMENSION ON THE NUMBER OF DATA POINTS USED (THE ACTUAL DIMENSION OF DATA IS 10)

Points number	Estimated dimension
1000	7.83
2000	7.94
5000	8.30
10000	8.56
30000	9.11
100000	9.73

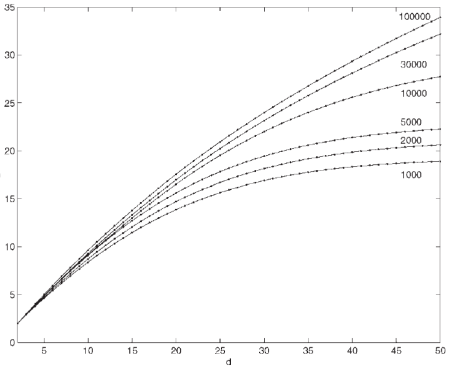

Fig. 4. Reference curves for different values of the cardinality N.

C. Empirical Procedure

Consider the set Ω of cardinality N. The empirical procedure (EP) consists of the following steps:

1) A set Ω' , whose ID d is known, with the same cardinality N as Ω is created. For instance, Ω' could be constituted by N points randomly generated in a d-dimensional hypercube.
2) The correlation dimension D of Ω' is measured with the GP algorithm.
3) The previous steps are repeated for T different values of d. The set $C = \{(d_i, D_i) : i = 1, 2, ..., T\}$ is obtained.
4) A best-fitting to the points in C is performed. A plot (reference curve) Γ of D versus d is generated (see Fig. 4). The reference curve allows to estimate the value of D when d is known.
5) The correlation dimension D of Ω is computed and, using Γ, the intrinsic dimension of Ω can be estimated.

The above method is based on the following assumptions:

1) Γ depends on N.
2) Since the GP algorithm gives close estimates for sets of the same dimensionality and cardinality, the dependence of Γ on the Ω' sets used for its setup is negligible.

TABLE II. ID ESTIMATION BY THE GP ALGORITHM AND EP OF 8-DIMENSIONAL AND 23-DIMENSIONAL DATA SETS

Points	GP (d=8)	EP (d=8)	GP (d=23)	EP (d=23)
1000	6.83	7.86	14.99	22.95
2000	6.94	7.75	15.76	22.48
5000	7.42	7.98	17.09	23.21
10000	7.51	8.20	18.04	22.43
30000	7.65	8.13	19.10	23.20
100000	7.83	8.03	19.78	23.24

This approach offers the following advantages: it allows one to estimate the ID of high-dimensional data, unlike TRN-based method. Moreover, the proposed approach is based on the estimation of a fractal dimension and, therefore, allows one to obtain noninteger values. This latter advantage is quite important, since, due to the presence of noise, real data can sometimes lie within a $fractal$-$like$ submanifold, whose dimension is usually noninteger.

D. Experimental Results of EP

The EP was tested by first creating reference curves for different values of the cardinality N, then by using each of them to estimate the dimension of data sets of the same cardinality and known dimension. During our experimentation, the sets Ω used for the reference curve setup were formed by randomly generated points in a d-hypercube. A plot was generated for the following cardinality values: 1,000, 2,000, 5,000, 10,000, 30,000, and 100,000. Correspondence to each value, a pair (d, D) was calculated for

$$d \in \{2, 3, 5, 10, 15, 18, 20, 25, 28, 30, 33, 35, 38, 40, 43, 45, 48, 50\} \tag{6}$$

The plot function is estimated by a multilayer-perceptron (MLP) [1]. Its structure was set up by means of the $Bayesianinformationcriterion$ [25]. The resulting reference curves can be seen in Fig. 4 In order to test the method, several sets (with cardinalities corresponding to the values indicated above) were created composed of random points generated in hypercubes in spaces with dimension 8 and 23. These sets were assumed to have ID 8 and 23, respectively, and were not used to generate the reference curves Γ. Following the procedure described in Section 4, the GP algorithm was first applied for each set, then the plot Γ corresponding to the same cardinality as the set being measured was used to compute the ID. The results are reported in Table II.

The table shows the dimension estimation obtained with the GP algorithm and with the empirical procedure proposed here. Indeed, a remarkable improvement is obtained when the cardinality is low. Afterwards, in order to validate the EP

TABLE III. Estimation of the Attractor Dimension of the Series D and A of Santa Fe Competition

Points	**GP (D)**	**EP (D)**	**GP (A)**	**EP (A)**
1000	7.54	8.84	2.00	2.01
2000	7.87	8.90	2.01	2.02
5000	8.13	8.83	2.03	2.03
10000	8.25	9.09	2.03	2.03
30000	8.48	9.07		

procedure, the data set A [14] and D [23] of the Santa Fe time series competition were considered. Data Set A is a real data time series generated by a Lorenz-like chaotic system, implemented by $NH_3 - FIR$ lasers. The data set D is a synthetic time series generated by a particle motion, simulated on a computer, with nine freedom degrees. The goal of the experimentation was to estimate, with the GP procedure, the attractor dimension of time series A and D.

In order to estimate the attractor dimension, they used the method of delays [17], [22] to the data set A, considering its first 1,000, 2,000, 5,000, 10,000 points. The results, obtained with the GP and EP algorithms, are reported in Table III.

Since the value of the fractal dimension of the attractor of Lorenz's system is approximately 2:06, the result can be considered satisfactory. They then applied the method of delays to the data set D, considering the first 1,000, 2,000, 5,000, 10,000, 30,000 points. The results, with GP and EP, are shown in Table III. Since the system that generated the data D has 9 degrees of freedom, the result can be considered particularly satisfactory.

V. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. F. Arend Heyting, "Collected works of l.e.j brouwer, north holland elsevier," 1975.

[2] D. R. O. K. Fukunaga, "An algorithm for finding intrinsic dimensionality of data," in *IEEE Transactions on Computers 20*, 1976, pp. 165 – 171.

[3] T. J. K. Pettis, T. Bailey, "An intrinsic dimensionality estimator from near-neighbor information," in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1979, pp. 25–37.

[4] K. S. T. Martinetz, "Topology representing networks, neural networks," 1994, pp. 507–522.

[5] G. V. Trunk, "Statistical estimation of the intrinsic dimensionality of a noisy signal collection," in *IEEE Transaction on Computers 25*, 1976, pp. 165–171.

[6] G. S. J. Bruske, "Intrinsic dimensionality estimation with optimally topology preserving maps," in *IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (5)*, 1998, pp. 572–575.

[7] M. Kirby, "Geometric data analysis: An empirical approach to dimensionality reduction and the study of patterns," in *John Wiley and Sons*, 2001.

[8] I. T. Jollife, "Principal component analysis," in *Springer-Verlag*, 1986.

[9] W. T. V. W. H. Press, B. P. Flannery, "Numerical recipes: The art of scientific computing," in *Cambridge University Press*, 1989.

[10] R. J. B. D. Fotheringhame, "Nonlinear principal component analysis of neuronal spike train data," in *Biological Cybernetics 77*, 1997, pp. 282–288.

[11] E. C. Malthouse, "Limitations of nonlinear pca as performed with generic neural networks," in *IEEE Transaction on Neural Networks 9 (1)*, 1998, pp. 165–173.

[12] P. R. C. Berg, J. P. R. Christensen, "Harmonic analysis on semigroups," in *Springer-Verlag*, 1984.

[13] F. Camastra, "Kernel methods for unsupervised learning, phd thesis progress report," in *University of Genova*, 2002.

[14] M. K. D. S. Broomhead, "A new approach to dimensionality reduction: Theory and algorithms," in *SIAM Journal of Applied Mathematics 60 (6)*, 2000, pp. 2114–2142.