WEKꓘƎM: A Study in Fractal Dimension and Dimensionality Reduction

[Extended Abstract] *

Elena Eneva
Center for Automated Learning and Discovery, Carnegie Mellon University
eneva@cs.cmu.edu

Krishna Kumaraswamy
Center for Automated Learning and Discovery, Carnegie Mellon University
skkumar@cs.cmu.edu

Matteo Matteucci
Center for Automated Learning and Discovery, Carnegie Mellon University
matteo@cs.cmu.edu

ABSTRACT

In this paper we investigate the relationship between several dimensionality reduction methods and the intrinsic dimensionality of the data in the reduced space, as estimated by the fractal dimension. We show that a successful dimensionality reduction/feature extraction algorithm projects the data into a feature space with dimensionality close to the intrinsic dimensionality of the data in the original space and preserves topological properties.

Keywords

Fractal Dimension, Intrinsic Dimensionality, Feature Reduction, PCA, Factor Analysis, Neural Networks

1. INTRODUCTION

Many real-life datasets have a large number of features, some of which are often highly correlated. These correlated attributes contribute to the increase of complexity of any treatment that can be applied to a dataset (spatial indexing in a DBMS, density estimation, knowledge retrieval in data mining processes, etc). This is referred to as the *curse of dimensionality* or as the *empty space phenomenon*. Moreover if there are linear correlations in the data (very likely in high dimensions), the optimal mean integrated squared error when estimating the data density will be very large even if the sample size is arbitrarily large.

However, a phenomenon which appears high–dimensional, and thus complex, can actually be governed by a few simple variables/attributes (sometimes called *hidden causes* or *latent variables*), and the many degrees of freedom are due to a variety of factors like noise or presence of irrelevant variables. Provided that this influence does not mask entirely the original structure, a good dimensionality reduction method should be able to "filter out" the unuseful dimensions and recover the original variables or an equivalent set of them.

Often vector spaces suffer from large differences between their embedding dimensionality and their intrinsic dimensionality. We define the **embedding dimensionality** of a dataset as the number of attributes of the dataset (its address space). The **intrinsic dimensionality** of a phenomenon (and also of the data retrieved from it) is defined as the real number of dimensions in which the points can be embedded while preserving the distances among them [4]. For example, a plane embedded in a 50-dimensional space has intrinsic dimension 2 and embedding dimension 50. This is in general the case in real applications and it has lead to attempts to measure the intrinsic dimension using concepts such as "fractal dimension" [5].

From a purely geometrical point of view, the intrinsic dimension would be the dimension m of the manifold that embeds a sample of an unknown distribution in a d–dimensional space ($d \gg m$), satisfying certain smoothness constraints. If the variables of a dataset are independent, then its intrinsic dimensionality is the embedding dimensionality. However, when there is a correlation between two or more variables, the intrinsic dimensionality of the dataset is reduced accordingly. For example, each polynomial correlation (linear, quadratic, etc.) reduces the intrinsic dimensionality by a unit. Other types of correlations can reduce the intrinsic dimension by different amounts, even fractional.

By knowing the intrinsic dimensionality of a dataset, it is possible to decide how many attributes are required to characterize the dataset. According to some research [8] [9], correlation fractal dimension can be used to approximate the intrinsic dimension $\mathscr{D}$ of a dataset and to discard some attributes (dimensions) until $\mathscr{D}$ is approached. Note that the fractal dimension $\mathscr{D}$ of a dataset cannot exceed its embedding dimensionality d and that there are $\lceil \mathscr{D} \rceil$ attributes which cannot be determined from the dataset. Since $\mathscr{D} \leq d$, there are at least $d - \lceil \mathscr{D} \rceil$ attributes which can be correlated with the others. Our claim is that a "good" subset of the original variables (or an equivalent set) projects the dataset into a space with fractal dimension close to the intrinsic dimension of the original dataset, thus preserving its

*A full version of this paper is available as *WEKꓘƎM: A Comparative Study of Dimensionality Reduction Methods and Fractal Dimension* at `http://www.cs.cmu.edu/~eneva/interest.htm` topological properties.

Some applications of dimensionality reduction are discussed in section 2. In section 3 we introduce a formalization of the feature reduction problem, and in section 4 we summarize the state of the art in dimensionality reduction. The approach used to prove our claim is described in section 5. The experimental results of our study are described in section 6 and a brief conclusion of the work and ideas for possible extensions to our comparative study are presented in section 7.

2. APPLICATIONS

Dimensionality reduction is an important issue in many types of applications. In *database applications*, multidimensional index structures (e.g., R* tree, SS-tree) are used for information retrieval and multi–modal queries. As the dimensionality of the records increases, query performances in the index structure degrade and the resulting complexity becomes comparable to sequential scanning. Two of the major research directions in finding an efficient way to overcome this problem are to develop indexing structures designed for high dimensional vector spaces, or to reduce the dimensionality so that the existing indexing techniques achieve satisfactory levels of performance.

In *classification applications*, high dimensional data are involved when large feature vectors are generated in order to describe complex objects and to distinguish between them. For a classifier, the estimation of the class probability distribution in sparsely sampled high dimensional spaces affects the reliability of the obtained classification results. Usually the easiest way to avoid these problems is to reduce the feature space by selecting a subset of the features using the classification performance on the training set as a measure for selecting the features. A more appropriate technique is to map the feature space into a lower dimensional space, preserving as much as possible the topological structure of the original space, and train the classifier in this new space [1].

In *statistical applications,* multivariate density estimation with finite samples is difficult to accomplish. Computational approaches for density estimation based on the maximum likelihood (e.g., EM algorithm) are quite slow, result in many suboptimal solutions, and depend strongly on initial conditions. In many cases it may be advantageous to map the data into a lower dimensional space first and solve the estimation problem in this new space.

3. PROBLEM DEFINITION

The two main approaches used for dealing with the curse of dimensionality are *Vector Quantization*, and *Dimensionality Reduction.* Both approaches propose an approximate solution to the density estimation of the data distribution.

In vector quantization, the given training examples are approximated using a small number of prototype vectors $C = \{c_1, c_2, ..., c_m\}$ with $m \ll n$, where n is the number of training examples in the d–dimensional space. Note that in this case a distribution in a d–dimensional space is approximated by a collection of points (prototypes) in the same space, leading to the so called zero–order approximation since this low–dimensional mapping/encoding is toward a space with dimension zero (set of points). More complex approximations can be used projecting on a 1 or 2 dimensional space, and in this case we have first–order or second–order approximations producing more compact encoding for nonuniform distributions.

In dimensionality reduction, we have an encoding function G and a decoding function F, where G maps from the input space $\mathbb{R}^d$ to a lower–dimensional feature space $\mathbb{R}^m$, and F maps from $\mathbb{R}^m$ back to the original space $\mathbb{R}^d$. In this paper we focus on this approach. Our objective is to find a mapping G from a d-dimensional input (sample) space $\mathbb{R}^d$ to some m-dimensional output space $\mathbb{R}^m$, where $m \ll d$,

$$G(x) : \mathbb{R}^d \rightarrow \mathbb{R}^m$$

producing a low dimensional encoding $z = G(x)$ for every input vector x. A "good" mapping G should act as a low dimensional *encoder* of the original (unknown) data distribution. Also, there should exist another "inverse" mapping

$$F(z) : \mathbb{R}^m \rightarrow \mathbb{R}^d$$

producing the decoding $x' = F(z)$ of the original input x. Thus the overall mapping for such an encoding–decoding process is

$$x' = F(G(x)).$$

We know from set theory that $||\mathbb{R}^d|| = ||\mathbb{R}||$ for any $d \in \mathbb{N}$, which means that we can map invertibly and continuously $\mathbb{R}^d$ into $\mathbb{R}$, for example using z-ordering or Hilbert curves. In principle, this would allow us to find a (nonlinear) continuous mapping from $\mathbb{R}^d$ into $\mathbb{R}^m$, with $m \ll d$, preserving all the information. Of course, due to finite precision this is of no practical application and we can only try to find the best approximation mapping. To do this, we need to specify a class of approximating functions (mappings):

$$f(x, \omega) = F(G(x))$$

parametrized by a vector of parameters ω and then to find a function (in this class) that minimizes the risk

$$R(\omega) = \int L(x, x')p(x)dx = \int L(x, f(x, \omega))p(x)dx,$$

where $p(x)$ is the true distribution of the data. In practice, since $p(x)$ is unknown, we use the empirical distribution of the data to approximate $p(x)$. Commonly, the loss function is the *squared error distortion*

$$L(x, f(x, \omega)) = ||x - f(x, \omega)||^2,$$

where $||\ ||$ denotes the usual L_2 norm on $\mathbb{R}^d$.

4. STATE OF THE ART

Common approaches used for dimensionality reduction rely on some parametric and non–parametric models. Other methods reduce the original feature space by heuristically dropping some of the attributes. In this section we describe some of these dimensionality reduction methods we used to support our hypothesis.

4.1 Principal Components Analysis

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called *principal components.* The first principal component accounts for as much of the variability in the data as possible, and each successive component accounts for as much of the remaining variability as possible.

In PCA, the data is summarized as a linear combination of an orthonormal set of vectors. Let $\{x_i\}_{i=1}^{n}$ be a sample in $\mathbb{R}^d$ with mean $\bar{x}$ and covariance Σ, with spectral decomposition $\Sigma = U \Lambda U^T$ (where U is orthogonal and Λ is diagonal). The principal component transformation $y = U^T(x - \bar{x})$ yields a reference system in which the sample has mean 0 and covariance matrix Λ containing the eigen–values of Σ. One can now discard the variables with small variance by projecting on the subspace spanned by the first few principal components, and obtain a good approximation of the original sample. The key property of PCA is that it attains the best linear mapping $x \in \mathbb{R}^d \longrightarrow x^* \in \mathbb{R}^m$ in the sense of the least squared sum of errors of the reconstructed data.

4.2 Factor Analysis

Factor Analysis is a statistical method for modeling the covariance structure of high dimensional data using a small number of latent variables [6]. It assumes that the data is a linear combination of real-valued uncorrelated Gaussian sources (factors). After the linear combination, each component of the data vector is also assumed to be corrupted with additional Gaussian noise (see Figure 1).

In maximum likelihood factor analysis, a d-dimensional real-valued vector x is modeled using an m-dimensional vector of real-valued factors, z, where $m \ll d$. The generative model is given by:

$$x = \Lambda z + u$$

where Λ is known as the *factor loading matrix.* The factors are assumed to be $\mathcal{N}(0, I)$ distributed. The d dimensional random variable u is distributed $\mathcal{N}(0, \Psi)$, where Ψ is a diagonal matrix. According to this model, x is therefore normally distributed with mean 0 and covariance $\Lambda\Lambda' + \Psi$.

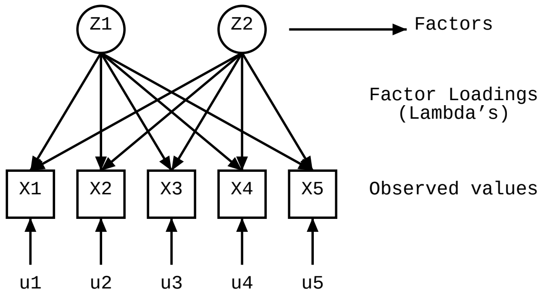

Figure 1: Simple path diagram for a factor analysis model

The goal of factor analysis is to find the Λ and Ψ that best model the covariance structure of x. The factor variables z model correlations between the elements of x, while the u variable accounts for the independent noise in each element of x. The m factors play the same role as the principal components in PCA: they are information projections of the data. We first subtract the mean of the data and then model the data as

$$x - \mu = \Lambda z + u.$$

We use the *EM-algorithm* to determine the values of Λ and Ψ and determine the reduced space [6].

4.3 Self–Supervised Multi Layer Perceptrons

Neural networks can be used to implement some statistical methods as well as a mapping between two vector spaces. The Self–Supervised MLP architecture (a.k.a. autoencoder) implements this mapping using two layers of linear perceptrons with d input, m hidden units and d output trained to replicate the input in the output layer minimizing the squared sum of errors with backpropagation. This approach is called *self-supervised*, referring to the fact that during the training each output sample vector is identical to the input sample vector.

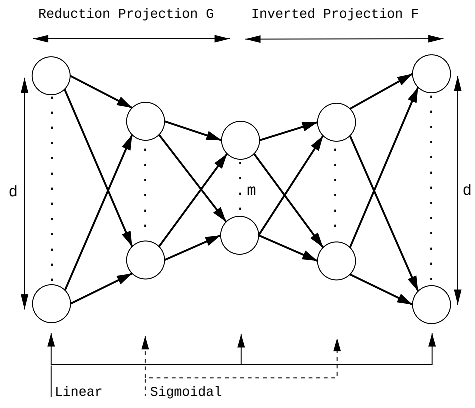

Figure 2: The Self–Supervised MLP architecture

A bottleneck MLP with a *single hidden layer* effectively performs *linear* PCA, even with nonlinear hidden units [3]. In fact, in order to effectively implement a nonlinear dimensionality reduction, the mapping functions F and G must both be nonlinear. Let us assume that F is restricted to be linear (as in the one hidden layer neural network model), though G may be nonlinear. The process of dimensionality reduction consists in finding functions F and G that are (approximately) functional inverses of each other. Since the inverse of a nonlinear function cannot be linear, therefore, if either function is linear, the other must also be linear.

Linear Self–Supervised MLP can be extended to implement non–linear PCA, using non–linear activation functions and more hidden layers (see Figure 2). In this approach, the subnet which leads to the hidden bottleneck implements the reduction projection on the feature space and the following subnet implements the inversion of such projection. Notice that the backpropagation approach does not take advantage of the inverse relationship between the structure of F and the structure of G.

5. APPROACH

In this paper we investigate the relationship between fractal dimension and feature reduction (for approximation and classification tasks). Our claim is that a good dimensionality reduction/feature extraction algorithm projects the data onto a feature space with dimensionality close to the intrinsic dimensionality of the data, preserving its topological properties. We show that, using a real dataset, this holds true, and therefore the fractal dimensionality of the resulting feature space can be used as an a-priori index for the final performance of the classifier/index method build on it.

Since dimensionality reduction implies a transformation of the feature space, we expect that the results of topology–based classification methods like K-nearest neighbors (KNN) which rely on a distance metric in the feature space will

	Fractal Dimension	KNN Error	DT Error	Reconstruction Error
Original Space	5.0047	23.90	25.74	0.0
PCA Reduced Space	3.3509	45.94	36.54	30.02
FA Reduced Space	3.6111	34.35	40.74	32.76
NN Reduced Space	3.7104	24.94	31.32	21.48
High Variance Reduced Space	1.3609	47.68	41.30	78.91
Random Reduced Space	1.7433	51.04	52.66	78.91

Table 1: The Fractal Dimension and the Performance Evaluation of the Methods

be more correlated to the fractal dimension with respect to other methods which rely on mutual information like decision trees (DT).

We use the described dimensionality reduction methods and two simple baseline methods – a reduced space with n randomly chosen features, where $n = \lceil \mathscr{D} \rceil$, and a reduced space with the n features with highest variances. The second baseline method is the easiest way to minimize the reconstruction error and explain most of the variance in the data without applying any transformations.

For practical applications, data preprocessing is often one of the most important stages in the development of a solution, and the choice of preprocessing steps can often have a significant effect on generalization performance [2]. The preprocessing we do consists of a linear rescaling of the input variables since in our dataset different variables have typical values which differ significantly, and the typical sizes of the inputs do not reflect their relative importance in determining the required outputs. We treat each of the input variables independently, and for each variable x_i we calculate its mean $\bar{x}_i$ and variance σ_i^2 with respect to the training set and we then define a set of rescaled variables given by

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i}.$$

We compute the *fractal dimension* of the datasets (see Section 6) using the correlation integral and the *classification errors* on the original data using all the attributes computed by a DT and KNN [1]. The first one is the measure we use to characterize the topological properties of the data and the second one gives us an idea of the expected performances of the feature reduction if it could keep all the relevant information and the topological structure.

Our conjecture for choosing the reduced set of features is that $\lceil \mathscr{D} \rceil$ will be a good heuristic. Having chosen the size of the reduced feature sets, we run the dimensionality reduction methods (PCA, Factor Analysis, and Neural Networks), keeping the defined number of features.

After applying dimensionality reduction methods, we measure their performance in terms of the reconstruction and classification powers of the reduced feature set. At the same time, we estimate the change in the topology structure of the reduced feature space using the fractal dimension of the projected data. We are interested in finding a correlation between the two characteristics of the feature reduction procedure: information preserved by the extracted features and the fractal dimension.

[1]The DT classification is based only on the information given from a single attribute in predicting the class, while the KNN classification uses the topology structure to determine the class.

6. EXPERIMENTAL RESULTS

In order to validate our approach we apply the dimensionality/feature reduction methods to a real dataset of classified protein images with 862 records and 84 attributes each[2] [7]. We scale the data according to the method proposed in the previous section and compute the fractal dimension of the original and the rescaled dataset (using the correlation integral).

As expected, since the units of measure for different variables in the original dataset are not comparable, without scaling we obtain a low fractal dimension of 1.6912. After scaling the dataset we obtain a less skewed distribution and the fractal dimension of the dataset increases to 5.0047.

According to the fractal dimension, the intrinsic dimensionality of the dataset of protein images is 2 for the non–rescaled version and 5 for the rescaled one. Therefore, we use 5 as the dimension of the reduced space. For each of the reduced feature spaces, we compute the FD using the correlation integral. The results for the fractal dimension and the performance evaluation of these spaces[3] are reported in Table 1.

The results of the KNN classification task confirm our claim. In Figure 3 we observe a monotonically decreasing trend such that the higher the fractal dimension, the lower the errors. Notice how, with rescaled data, the Neural Network non–linear reduced space performs the best, after the original dataset. This suggest that the Neural Network has captured the instrinsic non–linearity in the data.

The relationship we were looking for is present also when we use DT to classify the data (see Figure 4). The space obtained by a few features randomly selected from the original feature set has the highest classification error among all the spaces. This space is an outlier in the general trend to have lower error as the fractal dimension increases. A possible explanation for this is that the features randomly selected were minimally useful for classification and this heavily affects a classification method that relies only on mutual information.

The reconstruction error results also show the correlation with the fractal dimension of the reduced space and the relative topological properties (see Figure 5). However, the reconstruction error is not always the most important criterion which we want to minimize. For example, noisy/random features with high variance which do not carry any information about the classification of the data instances (and can be, therefore, unimportant for some tasks) can be completely

[2]Since the dataset has few records, we use the slope of the linear part in the correlation integral to compute the fractal dimension otherwise the box–counting method would not give an accurate estimate of it.

[3]We use 5–fold cross–validation and t–test to verify the statistical significance of the different results.

omitted and the significant penalty which the reconstruction error gives to a dimensionality reduction method for not being able to recreate those features should not be a concern for us.

7. CONCLUSIONS & FUTURE WORK

In this paper we presented a comparative study of dimensionality reduction using fractal dimension and different kinds of methods. We showed that a good dimensionality reduction method preserves the intrinsic fractal dimensionality of the data and the closer the intrinsic dimensionality of the dataset in the reduced space is to the intrinsic dimensionality of the dataset in the original space, the better is the classification achieved with the reduced feature set.

As future work it will be interesting to investigate how we can effectively choose the dimensionality of the target space by some heuristic measures extracted from the original data. An idea arising from the current work is a method for selecting relevant components in PCA space. We could start with the first principal component, compute the FD of the new space with just that component, then add another component, recompute the FD of the two components, and continue doing this until we see a flattening in the fractal dimension, meaning that more features don't change the intrinsic dimensionality of the dataset. Another possible extension is to consider other methods for dimensionality reduction, such as principal curves, Self Organinzing Maps, etc.

8. REFERENCES

[1] S. D. Backer, A. Naud, and P. Scheunders. Nonlinear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters*, 19:711–720, 1998.

[2] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, England, 1995.

[3] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59:291–294, 1988.

[4] E. Chvez, G. Navarro, R. Baeza-Yates, and J. L. Marroqun. Proximity searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, sep 2001.

[5] C. Faloutsos and I. Kamel. Beyond uniformity and independence: Analysis of R- trees using the concept of fractal dimension. In *Proc. ACM SIGACT-SIGMOD-SIGART PODS*, pages 4–13, 1994.

[6] Z. Ghahramani and G. Hinton. The em algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1997.

[7] R. F. Murphy, M. V. Boland, and M. Velliste. Towards a systematics for protein subcellular location: Quantitative description of protein localization patterns and automated analysis of fluorescence microscope images. In *Proc Int Conf Intell Syst Mol Biol (ISMB 2000)*, pages 251–259, 2000.

[8] M. Schroeder and F. Chaos. *Fractal, Chaos and Power Laws.* Freeman, New York, 1991.

[9] C. Traina, A. J. M. Traina, and C. Faloutsos. Distance exponent: A new concept for selectivity estimation in metric trees. Technical Report CMU-CS-99-110, Carnegie Mellon University, 1999.

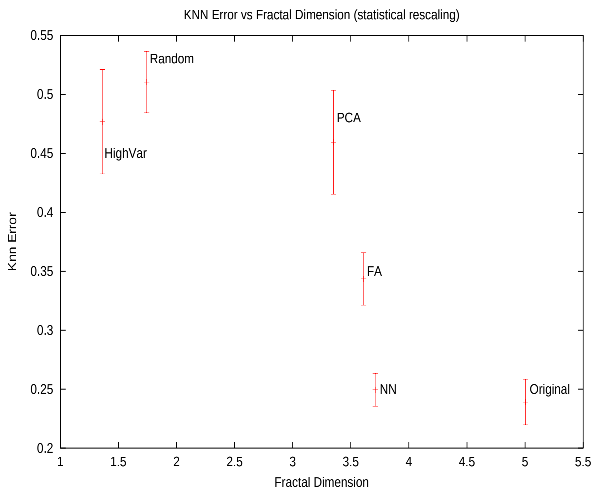

Figure 3: Correlation between KNN error and fractal dimension

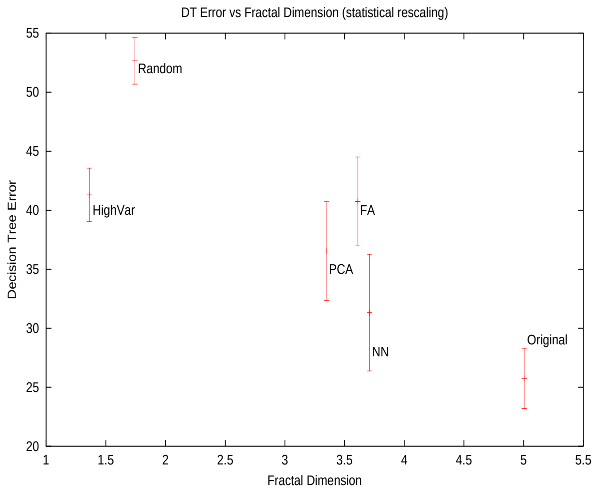

Figure 4: Correlation between DT error and fractal dimension

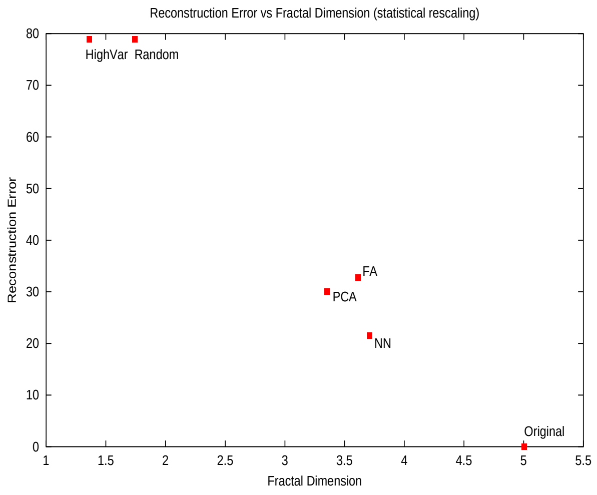

Figure 5: Correlation between reconstruction error and fractal dimension