An Intrinsic Dimensionality Estimator from Near-Neighbor Information

KAJL W. PETTIS, THOMAS A. BAILEY, ANIL K. JAIN, MEMBER, IEEE, AND RICHARD C. DÜBÉS, MEMBER, IEEE

Abstract—The intrinsic dimensionality of a set of patterns is important in determining an appropriate number of features for representing the data and whether a reasonable two- or three-dimensional representation of the data exists. We propose an intuitively appealing, iterative estimator for intrinsic dimensionality which is based on near-neighbor information. We give plausible arguments supporting the consistency of this estimator. The method works well in identifying the real dimensionality for a variety of artificial data sets and is fairly insensitive to the number of samples and to the algorithmic parameters. Comparison between this new method and the global eigenvalue approach demonstrates the utility of our estimator.

Index Terms—Eigenvalues, interpoint distances, intrinsic dimensionality, near neighbor information, refines.

I. INTRODUCTION

The TOPOLOGICAL or intrinsic dimensionality of a point set refers to the minimum number of parameters needed to generate the point set. For example, points lying along a reasonably smooth curve are said to have intrinsic dimensionality one, independent of the dimensionality of the space in which the points are represented. Similarly, points on a plane or a surface with a few undulations are said to have intrinsic dimensionality two.

A knowledge of the intrinsic dimensionality of a set of patterns contributes to the solution of two important problems in pattern recognition. First, what is an appropriate number of features (measurements) for representing the data? That is, how many of the features are essential in the design of a pattern classifier? Second, does a reasonable two- or three-dimensional representation of the data exist? One would like to "look" at the data, but unless such a representation is a true picture of the original data, one can be led into improper decision rules and formulate misleading ideas about the data. An estimate of intrinsic dimensionality can also help evaluate the effectiveness of algorithms designed to unfold or flatten the original data representation.

We propose a new method for estimating the intrinsic dimensionality of a set of points based on near-neighbor information. That is, our method extracts local information about the true dimensionality from a given set of points in an L-dimensional space. We explain the mathematical basis for our method, explain some statistical and computational characteristics of the estimator, and demonstrate the method on several data bases. We also compare our method to the standard method based on eigenvalues of a covariance matrix.

II. BACKGROUND

Studies in intrinsic dimensionality can be dichotomized according to the type of input information, either proximity matrix 1 or set of pattern vectors. Intrinsic dimensionality algorithms produce two types of information: 1) a configuration of points, one per pattern (data items or pattern vector); 2) an estimate of the intrinsic dimensionality of the configuration generated in 1. Our review below shows that several algorithms have been suggested for 1), but little attention has been paid to 2). Existing algorithms rely entirely on the eigenvalues of a covariance matrix to estimate intrinsic dimensionality even though Bell [1] demonstrated the perils of such an approach over 10 years ago. Thus we are motivated to take a different approach in estimating intrinsic dimensionality.

If the input information to an intrinsic dimensionality algorithm consists of a proximity matrix, the relative positions of the points in the output configuration should reflect these proximities. Points close to one another in the configuration should be very similar; points far apart, very dissimilar. Algorithms for producing configurations with this property are usually called multidimensional scaling algorithms, the best known example of which is MDSICAL, due to Kruskal [2], [11] and Shepard [4].

The criterion for the goodness of the configuration produced by MDSICAL is called "stress" and it depends only on the input proximities and the distances between points in the configuration. When the rank order of the distances is the same as the rank order of the proximities, stress is zero. The adoption of rank order means that the input proximities are treated as ordinal data. Thus any mathematical measure of the goodness of the configuration can depend only on the distances in the configuration space. Specifically, Kruskal's stress is

\[ S_{\text{Kruskal}} = \frac{1}{N} \sum_{ij} (d_{ij} - d_{ij}^*)^2 / \left( \sum d_{ij}^* \right)^{3/2} \]

where \( d_{ij} \) is the (Minkowski) distance between the points representing data items i and j and the \( d_{ij}^* \) are the values of the

1 Each new column in a proximity matrix corresponds to a data item. If the entries are dissimilarities, such as Minkowski distances, large values represent very dissimilar data items, or points for removed from one another. If they are similarities, large values represent very similar, or close, data items.

Manuscript received July 14, 1977; revised November 9, 1977. This work was supported by NSF Grant ENG 76-11936.

The authors are with the Department of Computer Science, Michigan State University, East Lansing, MI 48824.

1979 IEEE
Cam and Andrews [6] extended Bennett's algorithm by introducing a cost function to make Bennett's rank-order criterion more sensitive to local data regions. The basic idea is still to maintain rank order of local distances in two spaces. The core of the algorithm is to search over all interpoint distances, with no preference given to the smaller distances. Shepard and Carroll [7] limit consideration to the smaller interpoint distances so as to pay more attention to local neighborhoods.

Another nonlinear projection was suggested by Sammon [8] who minimized a stress measure similar to Kruskal's. Since Sammon began with points in a high-dimensional space, he could incorporate distances between these points in the stress measure directly, as indicated below:

$$S_{\text{Sammon}} = \sum_{i < j} \left( d_{i} - d_{j} \right)^{2} / \sum_{i < j} d_{i}^{2}$$

where $d_{i}$ is the distance between patterns $i$ and $j$ in the original pattern space and $d_{ij}$ is that in the two- or three-dimensional configuration space. Sammon, like Kruskal, minimized stress with a gradient-descent procedure that followed Kruskal's algorithm. It followed it so closely that Kruskal [9] subsequently demonstrated how the configuration very similar to Sammon's could be generated from MDSICAL.

Another variation on the stress criterion was suggested by Chang and Lee [10]. The gradient descent used by Kruskal and Sammon moved all points in the configuration space simultaneously to minimize stress. Chang and Lee suggest minimizing stress by moving the points two at a time. This approach also attempts to preserve local structure while minimizing stress. Unfortunately, the amount of computation becomes prohibitive, even when only a moderate number of points are involved and the results are dependent on the order in which the points are paired. Chang and Lee propose a "frame" method for overcoming the computational problem in which a representative number of points is fixed in the configuration space and the remaining points are moved with respect to the fixed points. A similar idea was used by Kruskal and Hart [11] to study the intrinsic dimensionality of a large number of binary patterns.

Several other approaches to the general intrinsic dimensionality problem have been suggested, including Shepard and Carroll's index of conformation [7], Kruskal's indices of condensation [12], and Kruskal and Carroll's parametric mapping [13]. Shepard [14] has summarized existing methods in psychometrics for studying intrinsic dimensionality and proposed several exciting ideas. Volumes [15], [16] have been written on multidimensional scaling itself.

A unique approach to the study of intrinsic dimensionality was recently proposed by Schwartzman and Vidal [17] who use the minimum spanning tree (MST) as the "information invariant representing the data." The input data must consist of patterns in a high-dimensional space. The swarm of patterns is flattened into a low-dimensional space by replacing each point by a weighted average (called a barycenter) of the point and all points connected to it in the MST. This moving average transformation smooths the original surface containing the patterns. After restoring the length of the original MST by uniform scaling, the barycenter transformation is...
repeated until either the connectivity pattern of the MST is violated or until the variance of the interpoint distances stabilizes. After this interesting idea, it is little disappointing to see that the actual intrinsic dimensionality is determined from the number of significant eigenvalues, as in Bennett’s algorithm.

Two approaches for determining the actual intrinsic dimensionality of a set of patterns have been suggested. Fukunaga and Olsen [18] get around the inherent problem of using eigenvalues of a covariance matrix on a global scale by compiling tables indicating the intrinsic dimensionalities of local regions, as judged by the number of significant eigenvalues of covariance matrices computed from local data only. The main drawbacks of this approach are the difficulty in assembling several pieces of local information into a global picture of the data and the need for interactive computation. The only method for directly estimating intrinsic dimensionality available in the engineering literature was proposed by Trunk [19]. His method is based on a series of hypothesis tests and works as follows. An initial value of an integer parameter K is chosen and the K nearest neighbors in each pattern in the given set are identified. The subspace spanning the vectors from the Kth pattern to its K nearest neighbors is constructed for all patterns. The angle between the (K + 1)st nearest 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, the rule is to decide that the intrinsic dimensionality is K. Otherwise, K is incremented by 1 and the process is repeated.

Our method also uses nearest-neighbor information, but we base our approach on a density estimator and determine intrinsic dimensionality without iterating over the dimensionality.

III. Method

Several aspects of the proposed method for estimating intrinsic dimensionality are considered in this section. The mathematical motivation for the estimator is given first, followed by the algorithm itself. Several theoretical and computational properties of the estimator are then investigated.

A. Mathematical Basis

We begin with a set \((X_1, X_2, \ldots, X_N)\) of L-dimensional patterns assumed to be drawn independently and identically from unknown density \(p(x)\). Our objective is to estimate the intrinsic dimensionality \(d\) of the swarm of patterns by deriving an estimator \(\hat{d}\). The motivation for our estimator comes from the well-known [20] estimator of \(p(x)\) given by

\[
\hat{p}(x) = \frac{k(n)}{V}
\]

where \(k\) is the number of near neighbors to \(x\) within a hypersphere of radius \(V\) about \(x\) and \(V = V_2 R^L\) is the volume of the hypersphere. Here, \(V_2\) is the volume of the unit d-dimensional hypersphere given by

\[
V_2 = \frac{(\pi^{d/2})}{\Gamma(d/2 + 1)}.
\]

Substituting the expression for \(V\) into (1) and taking logarithms produces

\[
\log(\hat{p}(x)) = (1/d) \log(k) + \log[(n, V_2 R^L)^{1/(4\pi)}].
\]

If the last term in (2) were independent of \(k\), there would be a linear relationship between \(\log(k)\) and \(\log(\hat{p}(x))\) with a slope of \((1/d)\) and we could use (2) to estimate \(d\). However, \(\hat{p}(x)\) is not independent of \(k\) and \(V_2\) is not uniquely determined. Our strategy is to derive an equation similar to (2) which can be used to isolate \(d\).

Let \(\tau_{K,x}\) be the distance from \(x\) to the \(K\)th nearest neighbor of \(x\). If \(p(x)\) is continuous and nonzero at \(x\), then for sufficiently large \(n\) and small \(r\), the density function for \(\tau_{K,x}\) can be taken as (Appendix I

\[
f_{\tau_{K,x}}(r) = c n^{d/2 - 1} \frac{(\log(k) - 1)}{(k)} [\exp(c r^{d})], \quad \text{if } r > 0
\]

\[
= 0, \quad \text{else}
\]

where \(c = n p(x) V_2\). The expected value of \(\tau_{K,x}\) is

\[
E(\tau_{K,x}) = \int_0^\infty f_{\tau_{K,x}}(r) dr = \left[ \frac{k}{n p(x) V_2} \right]^{1/4d}
\]

Defining a sample-averaged distance to the \(K\)th nearest neighbor over the set of patterns as

\[
\tau_{\bar{K},x} = \frac{1}{n} \sum_{i=1}^{n} \tau_{K_i,x_i}
\]

From (4), the expected value of this average distance is

\[
E(\tau_{\bar{K},x}) = \left( \frac{1}{n} \sum_{i=1}^{n} \tau_{\bar{K},x_i} \right) = \frac{1}{G_{K,d}} k^{1/d} C_n
\]

where

\[
G_{K,d} = \frac{k^{1/d} \Gamma(1)}{\Gamma(1 + 1/d)}
\]

and

\[
C_n = \left( \frac{n}{\sum_{i=1}^{n} \log(p(x_i)) V_2} \right)^{1/d}
\]

Although \(C_n\) is sample-dependent, it is independent of \(k\). Taking logarithms in (5) yields an equation similar to (2)

\[
\log(G_{K,d}) + \log(\tau_{\bar{K},x}) = (1/d) \log(k) + \log(C_n).
\]

The term \(\log(G_{K,d})\), although not independent of \(k\), is close to 0 for all \(k\) and \(d\), as shown in Appendix II.

As an estimator for \(E(\tau_{\bar{K},x})\) we take our observed value of the random variable \(\hat{\tau}_{\bar{K},x}\) computed from the given sample. Using this observed value in (6) defines an estimator \(\hat{d}\) for \(d\)

\[
\log(G_{\hat{d},d}) + \log(\hat{\tau}_{\bar{K},x}) = (1/d) \log(\hat{K}) + \log(C_n).
\]

We can solve for \(d\) since a plot of \(\log(\hat{\tau}_{\bar{K},x})\) as a function of \(\log(k)\) will have a slope of \((1/d)\). The term \(C_n\) affects only the intercept of this plot and thus the underlying density \(p(x)\) need not be known to estimate \(d\).

We solve for \(d\) iteratively. The initial estimator \(\hat{d}_0\) is obtained by setting \(\log(G_{\hat{d}_0,d})\) to zero and fitting a least square
regression line to a plot of $\log(r_k)$ versus $\log(k)$ for $k = 1, \ldots, K$ for some $K$. We then use the value for $\log(G_k, d_k)$ and fit another regression line to obtain $\hat{d}_k$. We continue until reaching an $i$ for which

$$\hat{d}_i > \hat{d}_{i-1} + \epsilon$$

for some $\epsilon$. The estimator is $\hat{d} = \hat{d}_i$. In practice, $\hat{d}$ is rounded to the closest integer.

B. Algorithm

The actual algorithm used to compute $\hat{d}$ in (7) is straightforward, which is one advantage of the method. A flowchart is given in Fig. 1. Details of the computation are explained below.

The term $\log(k)$ in (7) refers to the term $\log(r_k)$ in (6)

$$\log(r_k) = \log \left( \frac{1}{n} \sum_{j \neq i} r_{k,j} \right)$$

(8)

In (7), the sum is from 1 to $n$. However, empirical studies showed that including samples on the edges of the swarm of samples, or outliers, substantially increased the variance and distorted the estimate of $d$. Several schemes can be used to identify outliers. We define $\max$ as the measure of the average distance $m_{\text{max}}$, and a measure of the spread of the distances $s_{\text{max}}$, as follows:

$$m_{\text{max}} = (1/n) \sum_{j \neq i} r_{k,j}$$

$$s_{\text{max}}^2 = (1/n - 1) \sum_{j \neq i} (r_{k,j} - m_{\text{max}})^2$$

The sum in (8) covers all $n^2$ values of $j$ for which $r_{k,j} < m_{\text{max}} + s_{\text{max}}$.

The following approximation for the first term in (7) is based on the Taylor series expansion for the logarithm of the gamma function [22] and is valid even for the worst case when $k = 1$ and $d = 2$

$$\log(G_k, d) = -d - 1 - \frac{(d - 1)(d - 2)}{12d^2} - \frac{(d - 1)^2}{12d^2} + O(1/d^4)$$

Equation (7) involves $\hat{d}$ in two places. The algorithm in Fig. 1 computes successive estimates of $d$ until the difference between the estimates is sufficiently small. We took $\epsilon = 0.01$ in our studies because $d$ should be an integer. The number of iterations through the loop almost never exceeded four. The adjustment required for $\log(k)$ in Fig. 1 is simply to add $\log(G_k, d)$ for each $k$. The equation used to compute $\hat{d}_k$ is obtained from the standard equation for the slope of a regression line:

$$\hat{d}_k = \frac{\sum_{k} (\log(k)) (\log(r_k)) - (\sum_{k} \log(k))(\sum_{k} \log(r_k))}{\sum_{k} (\log(k))^2 - (\sum_{k} \log(k))^2}$$

All sums are for $k = 1, 2, \ldots, K$.

Fig. 1. Procedure for computing $\hat{d}$.

The parameter $K$, the number of near neighbors used, is not critical. Our empirical studies show good results for a wide range of values of $K$, even for $K = 2$.

One other point about our method should be made. One could conceivably estimate $\hat{d}$ at each sample, then average the estimates to obtain a global estimate of $d$. Specifically, suppose $\hat{d}(r_{k,j})$ in (4) were replaced by $r_{k,j}$, the distance from $X_i$ to its $k$th nearest neighbor. Letting

$$c_i = \exp(X_i) \gamma_{ij}^{-1/2}$$

and taking logarithms, we obtain

$$\log(G_k, d) + \log(r_{k,j} - (1/2)(\log(k)) + \log(c_i)$$

As estimator, say $\hat{d}_{k,j}$, for $d$ could be obtained in the same way that $\hat{d}$ was formed in Fig. 1. Then one might use

$$\frac{(1/n) \sum_{k} \hat{d}_k}{\sum_{k} \hat{d}_k}$$

to estimate $\hat{d}$.

The difficulty with this approach is that the means of the individual terms $\hat{d}_{k,j}$ tend to be very large. In fact, one can show that, for $K = 2$,.

Heart

Set $k = 1$.

Get $x_{1,1}, \ldots, x_{1,n}$ in L-dimensional space.

Compute $r_{k,j}$ for all $k$.

Use (6) to compute $\log(k)$ for $k = 1, 2, \ldots, K$.

Compute $m_{\text{max}}$ for all $k$.

Remove outliers.

Compute $s_{\text{max}}$ for $d = 1$.

Adjust $\log(k)$ for $d = 1$.

Compute $\hat{d}_k$ for $k = 1, 2, \ldots, K$.

Stop.

Done.
\[ \hat{d}_{ni} = \left( \frac{r_{ni}}{r_{ni}/r_{ni}} \right)^{-1} \]

and \( \hat{d}(\hat{d}_{ni}) \) does not exist. For this reason, we chose the procedure that culminates in (7).

C. Distribution of \( \hat{d} \) for a Special Case

An exact distribution for the estimator \( \hat{d} \) in (7) is desired to study the behavior of the expected value and variance of the estimator. Unfortunately, determining the distribution for \( \hat{d} \) is difficult, even for simple underlying distributions. We have, however, obtained the distribution for \( \hat{d} \) when the underlying distribution is uniform over the interval \((0,1)\), \( L = d = 1 \), and \( n = 3 \). We will show that \( \hat{d} \) has reasonable properties for this special case.

If we use only two near neighbors, say the \( k \)th and \((k+1)\)th, (7) leads to

\[ \log G_{k,d} = \log \frac{\frac{r_{k}}{r_{k+i}}}{r_{k}} = (1/d) \log \frac{k + 1}{k} \]

Using the definition of \( G_{k,d} \) in (5)

\[ \log G_{k,d} = (1/d) \log \frac{k + 1}{k} \]

Substituting (10) into (9)

\[ (r_{k+1}/r_{k}) = \frac{1}{k+1} \]

Solving for \( \hat{d} \)

\[ \hat{d} = \frac{r_{k}}{\sum_{k+1}^{n+1} r_{k}} \]

In the special case being considered here, each pattern in the set \( \{X_1, X_2, X_3\} \) has at most two near neighbors. Setting \( k = 1 \)

in (1) shows that

\[ \hat{d} = \frac{r_{1}}{r_{1}/r_{2}} \]

We now determine the density function for \( \hat{d} \) in (12). We assume, without loss of generality, that the sample patterns are numbered so that

\[ X_1 \leq X_2 \leq X_3 \]

Define \( A = \max(X_3 - X_1, X_2 - X_1) \) and \( B = \min(X_1 - X_1, X_2 - X_3) \). The joint density function for \( A \) and \( B \) can be shown to be

\[ f_{A,B}(a, b) = \frac{1}{2}(1 - a - b), \quad \text{for } 0 \leq a \leq 1 \quad \text{and } \quad 0 \leq b \leq \min(1, 1 - a) \]

\[ = 0, \quad \text{else.} \]

The sample-averaged distances to the nearest neighbors are

\[ r_1 = (A + 2B)/3 \quad \text{and} \quad r_2 = (3A + 2B)/3. \]

From (12), the estimator is

\[ \hat{d} = (A + 2B)/2A \]

It follows that the density for \( \hat{d} \), which is pictured in Fig. 2, is

\[ f(\hat{d}) = 2(\hat{d} \cdot 0.5)^3, \quad \text{for } 0.5 \leq \hat{d} \leq 1.5 \]

\[ = 0, \quad \text{else.} \]

The mean and variance of \( \hat{d} \) are

\[ \mathbb{E}(\hat{d}) = \int 2 \ln(2) \cdot 0.886 \]

\[ \text{var}(\hat{d}) = 2 - 4 \ln(2)^2 = 0.078. \]

Thus \( \hat{d} \) provides a reasonable estimate of the true intrinsic dimensionality of one. As seen in Fig. 2, the range for \( \hat{d} \) is centered at this true value.
D. Asymptotic Behavior of $\hat{b}$ When $n = 2$

We now propose an intuitive argument showing the consistency of $\hat{b}$ when $n = 2$. The expected value of $\hat{b}$ in (12), based on $n$ samples, can be written as follows:

$$E(\hat{b}) = \int_{0}^{1} \int_{0}^{1} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(t_1 - t)^2}{2}\right) dt_1 dt_2$$

where $\exp_{\mu}^2(., .)$ denotes the joint density for the random variables

$$\exp_k = E(\hat{b}) \sum_{k=1}^{k-1} r_k X_k$$

for $k = 1, 2$.

Since $P_1$ and $P_2$ are sums of jointly distributed random variables, one can apply a Central Limit Theorem under certain conditions. Showing that such conditions are satisfied is very difficult in this case. We rely on the following intuitive argument.

For each $i$, $r_k X_i$ and $r_k X_i$ have a joint distribution whose covariance matrix is finite. Also, for a given $k$ and $i \neq j$, $r_k X_i$ and $r_k X_j$ are only weakly dependent. Under these conditions, it seems reasonable to apply the bivariate form of the Central Limit Theorem [21], which implies that the joint density of $P_1$ and $P_2$ approaches a $\delta$-function as $n$ grows large

$$f(\hat{b}) \rightarrow \delta[x - E(P_1), t - E(P_2)].$$

Therefore, as $n \rightarrow \infty$, $E(\hat{b})$ can be written as follows and evaluated with (5):

$$E(\hat{b}) \rightarrow \frac{1}{E(E(P_1)E(P_2))} = 1.$$

In addition, it is clear that

$$\text{var}(\hat{b}) \rightarrow 0.$$

Thus, under our assumptions, $\hat{b}$ is a consistent estimator of $b$ when $n = 2$.

IV. EXPERIMENTAL RESULTS

We applied our method to several sets of artificial data and compared the estimates of intrinsic dimensionality to the outcomes of global eigenvalue analyses. The results are summarized below. Our method performed consistently well, even when the eigenvalue analysis provided incorrect answers.

A. Data Generation

Six types of data were investigated.

1) Gaussian Data: Three sets of Gaussian data were generated from L-variate Gaussian distributions with zero-mean vectors and unit-covariance matrices. The three sets were for $L = 1, 2$, and 3. The unit-covariance matrices imply that $d = L$ for all three sets.

2) Surface Data: Two sets of uniform data were generated by randomly projecting the three-dimensional Gaussian data onto the surface of a sphere ($L = 2$, $d = 2$) and the two-dimensional data onto a circle ($L = 2$, $d = 1$). In both cases, the distribution of points over the surface can be shown to be uniform (see Appendix III).

3) Interior Data: Three more sets of data were generated by uniformly distributing points in the interior of a sphere ($L = 3$), interior of a circle ($L = 2$), and interior of an interval ($L = 1$), as explained in Appendix III. In all cases, $d = L$.

4) Filter Data: Bennett's original work [5] was motivated by trying to determine the minimum number of free system parameters required to describe a set of system outputs. We generated a data set in this spirit by choosing a point at random in the interior of a unit circle and treating the point $q$ as a pole of a digital filter. That is, letting $\tilde{q}$ be the complex conjugate of $q$, this point defines the z-domain transfer function

$$H(z) = \frac{z}{z - q} \frac{z}{z - \tilde{q}}$$

which has impulse response

$$h(n) = |q|^n \sin(\theta)^n \sin(\theta)^{-1}$$

if $n > 0$

where

$$q = |q| \exp(i\theta).$$

We observed $H(n)$ for $n = 1, 2, \ldots, 20$. Since $h(1) = 1$, we have $L = 19$. What is the "true" intrinsic dimensionality here? Since $q$ has a magnitude and phase angle that can be chosen independently, we might consider $d = 2$.

5) Helix Data: Helix data have been used in several studies on intrinsic dimensionality [8], [17]. Points were chosen at random along a helix described by the following equation:

$$x = \cos \theta$$
$$y = \sin \theta$$
$$z = 0.1(\theta)$$

$0 < \theta < 4\pi$.

For this data set, $L = 3$ and $d = 1$. A sketch of the helix is shown in Fig. 3.

6) Hyperellipsoidal Data: Five sets of data from three-dimensional Gaussian distributions were generated to study the effect of noise. The mean vectors of these distributions were the zero vector and the covariance matrices have the following form:

$$\begin{pmatrix}
  v_1^2 & 0 \\
  0 & v_2^2 \\
  0 & 0
\end{pmatrix}$$
The values of $q_2$, $q_3$, and $q_4$ and the corresponding $d$ values are given below:

a) $q_2 = 100$, $q_3 = q_4 = 1$, $d = 1$;

b) $q_2 = 10$, $q_3 = q_4 = 1$, $d = 1$ or $3$;

c) $q_2 = q_3 = q_4 = 1$, $d = 3$;

d) $q_2 = 1$, $q_3 = q_4 = 10$, $d = 2$ or $3$;

e) $q_2 = 1$, $q_3 = q_4 = 100$, $d = 2$.

Cases a) and b) represent noisy lines of different lengths. Case c) is a hypersphere and is included for comparison purposes. Finally, cases d) and e) represent noisy circles.

8. Results

Three parameters are associated with each method for generating data: $n$ (the number of patterns), $K$ (the maximum number of near neighbors used), and $M$ (the number of Monte Carlo runs for each situation). We let $M$ be 10, $n$ take on the values 5, 10, 20, 50, and 100, and $K$ be 2, 4, 9, and 19. We ran all cases for which $K < n$.

The mean and variance of $d$ for the ten Monte Carlo runs were computed for $n$, $K$, and data type fixed. The average values of $d$ are plotted in Figs. 4-9. For the first five data sets, the abscissas represent $n$, the number of patterns, and one curve is drawn for each $K$ and each data set. In almost all cases, the average $d$ is very close to $d$ when $n > 10$. The filter data present an unusual case in that the average $d$ is closer to 3 than to 2. Perhaps more samples are needed to discern the true intrinsic dimensionality. The ratio between the separation of the coils and the radius of the helix in Fig. 3 is small, making this a difficult problem. The plot of $d$ for the ten Monte Carlo runs shown in Fig. 8 indicates that our algorithm was able to obtain a good estimate of the intrinsic dimensionality for $n > 50$. For the hyperequipartition data, the results for $K = 4$ are given in Fig. 9. Each curve in this figure represents one of the five cases. For cases a), c), and e), $d$ approaches the desired values, while for cases b) and d), $d$ is within the range of expected values. Other values of $K$ give comparable results.

In all cases, the variance of $d$, computed over the ten Monte Carlo runs, decreased as $n$ increased. For $n = 100$, the largest variance occurred when $K = 2$ for the three-dimensional interior data; the range of the variance across all runs was (0.25, 0.001). Increasing $K$ tended to decrease the variance of $d$. Deleting the outliers also tended to decrease this variance, especially in the filter data. This effect was specially apparent for small values of $n$, at which outliers would have an unusually strong influence.

C. Comparison

Table 1 shows the two or three most significant eigenvalues (normalized so that all eigenvalues sum to 100) of the sample covariance matrices for one of the ten cases of each data set. That is, we estimated the covariance matrix from the given sample and computed its eigenvalues. The number of significant eigenvalues is normally taken as a measure of intrinsic dimensionality. The Gaussian data show the eigenvalues approaching one another as $n$ increases. Theoretically, of course, they should
Fig. 5. Average $\bar{d}$ for surface data.

Fig. 6. Average $\bar{d}$ for interior data.
Fig. 7. Average \( \hat{d} \) for linear data.

Fig. 8. Average \( \hat{d} \) for bell data.

Fig. 9. Average \( \hat{d} \) for hyperellipsoidal data (\( K = 4 \)).
be identical. Using eigenvalues to estimate intrinsic dimensionality requires judgment about which eigenvalues are "signifi-
cant." One hopes that the eigenvalues are either large or very small to simplify this judgment. For instance, the eigenvalues computed from the original image data show the same characteristics as those computed from the interior data, even though $d = 2$ for the surface data and $d = 3$ for the interior data. Our method discriminated well while the eigenvalue method did not. The accuracy inherent in the eigenvalue scheme is exemplified by the filter data. For $n = 100$, one could make arguments for one, two, or three significant eigenvalues.

V. DISCUSSION AND CONCLUSIONS

We have proposed a new method for directly estimating the intrinsic dimensionality of a set of patterns that requires only near-neighbor information. Our estimator has three major ap-
lications. First, it can follow one of the data-flattening algo-
rithms [5], [6], [10], [17] and provide a more realistic esti-
mate of intrinsic dimensionality than the global eigenvalue method, especially when the output of the data-flattening algo-
rithm is highly warped. Second, our estimator can guide the choice of an appropriate number of dimensions for repre-
senting the data. One can then concentrate on establishing a linear or nonlinear projection for the original patterns without worrying about the appropriate dimensionality of the new space. Third, one could have a data-flattening algorithm on the criterion of minimizing our estimator as opposed to mini-
mizing stress or minimizing variance of the point distances. We hope to investigate this possibility in the future.

Our method of estimating intrinsic dimensionality uses distances to the nearest neighbors and is thus dominated by local information. If the measurements are noisy or the hyper-
surface on which the patterns lie is thick, then using local information will positively bias our estimate. For example, a thick spherical shell will be indistinguishable from a solid sphere if only the first few neighbors are considered. The desired value of the intrinsic dimensionality, two, cannot be recovered either by global or local information alone. This effect is apparent in Fig. 9, where, for $K = 4$, the estimator $\hat{d}$ for case b increases from approximately 1 to approximately 3 as the number of samples increases. That is, for $n >> K$, local information dominates the estimator; while for $n = K$, about all samples are required to capture $K$ near neighbors.

Two effects observed in our empirical study require com-
ment. First, in all cases, the value of $K$ chosen did not mate-
rially affect $\hat{d}$ when $n$ grew large. For small sample size, the most erratic behavior in $\hat{d}$ was experienced when $K = 2$. Obviously, if the surface is wrinkled or has a high degree of curva-
ture, then the number of samples needed to obtain a good estimate of $d$ has to be large so that the local neighbor-
hood used for our estimate is small. Second, it appears that

edge effects negatively bias our estimator (Fig. 5) even though Edge effects are amelio-
rated when the number of samples increases.

The statistical properties of our estimator are not inter-
ded. Open question: the distribution of $\hat{d}_K$ for various $K$ and $n$ has not been worked out. We were able to show that $\hat{d}$ has reasonable properties in the simple case of three points uni-
f ormly distributed on an interval. We also proposed a $\alpha$

The algorithm for computing $\hat{d}$ is straightforward. It involves only three parameters, $\epsilon$, $K$, and Maxiter, and its performance is insensitive to wide ranges of values of these parameters. The computation is dominated by the determination of near neigh-
hrs, so the amount of computation varies as $n^2$. If one is computing near neighbors for some other purpose, such as clustering [23], decision-making [24], or data representations [8], [25], [26], the extra computational burden imposed by our algorithm is not great. In addition, several fast methods of computing near neighbors for small values of $d$ have recently been developed [27], [28] and our algorithm will benefit from their use.

Appendix I

DENSITY OF DISTANCE TO $K$TH NEAREST NEIGHBOR

The density function for $d_{x,y}$, the distance from $x$ to its $K$th nearest neighbor among $(x_1, x_2, \ldots, x_n)$, is required in Sec-

mion III-A, and is derived in this Appendix. The assumptions are as follows.

1) The underlying density $f(x)$ is constant over $S_K(r)$, a sphere of radius $r$ centered at $x$ whose volume is $V = \pi r^d/2$.

2) The distribution of $x_i$, $i = 1, 2, \ldots, n$, is performed by a sequence of $n$ Bernoulli trials with probability $p(x) \propto \pi r^d$ of "success," or falling within $S_K(r)$. Let $N_K(r)$ be a random variable denoting the number of samples in $S_K(r)$.

The previous assumption imply that

$$P[X_N(r) = m] = \frac{n!}{m!(n-m)!} [1 - p(r)^d]^m \cdot p(r)^d \cdot m \leq m \leq n. \quad (1.1)$$

The density function $f_{K,s}(r)$ for $d_{x,y}$ can be expressed in terms of $N_K(r)$ as follows:

$$f_{K,s}(r) = \lim_{\Delta r \to 0} \frac{P \left[ \frac{x \in S_K(r) \Delta r}{\Delta r} \right]}{\Delta r} = \frac{1}{(1/\Delta r) \ln \left[ \frac{N_K(r + \Delta r) - N_K(r)}{K-1} \right]} \cdot P[X_N(r) = k - 1].$$

Assumption 1) implies that the first factor is asymptotically

$$np(x) \Delta r \Delta r$$

where

$$\Delta r = V_{x,y} + (\Delta x)^d \cdot \Delta y \cdot \Delta y = \frac{V_{x,y} \Delta x^d}{2} + o(\Delta x).$$

Thus,

$$f_{K,s}(x) = np(x) \Delta x \Delta y \Delta z \cdot P[X_N(r) = k - 1].$$

The Poisson approximation to the binomial probability in (1.1) produces a usable form for the density. The approxima-

The
TABLE I

<table>
<thead>
<tr>
<th>Data set</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-D Gaussian</td>
<td>61.14</td>
<td>67.01</td>
<td>81.00</td>
<td>84.14</td>
<td>84.60</td>
</tr>
<tr>
<td></td>
<td>83.14</td>
<td>84.91</td>
<td>88.48</td>
<td>91.26</td>
<td>92.54</td>
</tr>
<tr>
<td></td>
<td>94.71</td>
<td>98.01</td>
<td>99.42</td>
<td>99.58</td>
<td>99.83</td>
</tr>
<tr>
<td>2-D Fano</td>
<td>62.43</td>
<td>67.04</td>
<td>80.63</td>
<td>84.93</td>
<td>85.32</td>
</tr>
<tr>
<td></td>
<td>83.32</td>
<td>85.92</td>
<td>89.37</td>
<td>91.07</td>
<td>97.20</td>
</tr>
<tr>
<td>3-D Surface</td>
<td>53.86</td>
<td>45.54</td>
<td>43.94</td>
<td>39.68</td>
<td>36.94</td>
</tr>
<tr>
<td>of Sphere</td>
<td>28.29</td>
<td>28.78</td>
<td>34.15</td>
<td>31.55</td>
<td>38.37</td>
</tr>
<tr>
<td>2-D on the</td>
<td>14.82</td>
<td>22.77</td>
<td>21.91</td>
<td>28.48</td>
<td>28.69</td>
</tr>
<tr>
<td>Circle</td>
<td>59.73</td>
<td>61.35</td>
<td>61.76</td>
<td>64.74</td>
<td>63.32</td>
</tr>
<tr>
<td>2-D Interior</td>
<td>90.82</td>
<td>51.63</td>
<td>41.34</td>
<td>41.26</td>
<td>40.28</td>
</tr>
<tr>
<td>of Sphere</td>
<td>53.23</td>
<td>48.65</td>
<td>48.82</td>
<td>50.02</td>
<td>57.28</td>
</tr>
<tr>
<td>2-D Interior</td>
<td>62.12</td>
<td>50.11</td>
<td>51.25</td>
<td>51.19</td>
<td>52.40</td>
</tr>
<tr>
<td>of Sphere</td>
<td>50.54</td>
<td>20.04</td>
<td>29.03</td>
<td>29.78</td>
<td>29.90</td>
</tr>
<tr>
<td>Filter</td>
<td>82.60</td>
<td>70.14</td>
<td>53.60</td>
<td>60.50</td>
<td>52.87</td>
</tr>
<tr>
<td></td>
<td>86.40</td>
<td>81.86</td>
<td>66.40</td>
<td>39.99</td>
<td>47.15</td>
</tr>
<tr>
<td>Neuron</td>
<td>67.35</td>
<td>92.2</td>
<td>88.2</td>
<td>44.68</td>
<td>43.1</td>
</tr>
<tr>
<td></td>
<td>27.7</td>
<td>4.5</td>
<td>10.1</td>
<td>12.7</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>1.8</td>
<td>5.8</td>
<td>15.7</td>
<td>6.3</td>
</tr>
<tr>
<td>Neuron</td>
<td>65.54</td>
<td>58.78</td>
<td>51.13</td>
<td>45.95</td>
<td>48.45</td>
</tr>
<tr>
<td></td>
<td>7.77</td>
<td>10.77</td>
<td>7.49</td>
<td>45.00</td>
<td>41.44</td>
</tr>
<tr>
<td></td>
<td>7.64</td>
<td>10.55</td>
<td>11.38</td>
<td>9.05</td>
<td>8.94</td>
</tr>
</tbody>
</table>

APENDIX II

**BOUNDS ON LOG(G_{k,d})**

In this Appendix, we demonstrate a bound required in Section III-A. We first show that for all \( k \geq 1 \) and \( d \geq 1 \)

\[
0 \leq \log(G_{k,d}) = \frac{d^{d-1}(e^{e-d})^{k+1}}{\Gamma(k+1)d} \leq \log \left( \frac{1}{d} \right) = \log(G_{1,d}).
\]

(2.1)

Substituting \( z \) for \( 1/d \) makes (2.1) equivalent to

\[
0 \leq g_k(z) \leq g_1(z),
\]

for \( 0 \leq z \leq 1 \)

where

\[
g_k(z) = z \log(k) + \log(\Gamma(k)) - \log(\Gamma(k+z)).
\]

Since \( g_k(0) = g_1(1) = 0 \), for all \( k \)

it will suffice to show that for \( 0 \leq z \leq 1 \), the second derivatives satisfy

\[
g_k''(z) \leq g_1''(z) < 0.
\]

(2.2)

However, (2.2) follows from Davis [22].

\[
g_k''(z) = -\sum_{i=1}^{\infty} (i+z)^{-2},
\]

In addition,

\[
\log(G_{k,d}) < 0.12
\]

the maximum occurring at \( d = 2.17 \). Thus, we conclude that

\[
0 \leq \log(G_{k,d}) < 0.12 \text{ for } k \geq 1 \text{ and } d \geq 1.
\]

**APPENDIX III

GENERATION OF UNIFORMLY DISTRIBUTED DATA ON THE SURFACE AND IN THE INTERIOR OF A HYPERSPHERE**

If \( r \) is an \( L \)-dimensional Gaussian random variable with mean \( \mu \) and covariance matrix \( \Sigma \), then because there is no preferred direction from the origin

\[
\frac{r}{|r|}
\]

is uniform over the interior of an \( L \)-dimensional hypersphere. [29]

And, if \( r \) is a random variable uniform on \((0, 1)\), then

\[
\frac{r}{|r|}
\]

is uniform over the interior of an \( L \)-dimensional hypersphere. Thus creating surface and interior data requires \( L \) and \( L+1 \) random numbers, respectively (one for each coordinate of \( z \) and a value of \( r \), for one pattern).

Straightforward rejection methods such as generating uniform points in a hypercube and rejecting points outside of the hyperspace are grossly inferior to this method even for moderate values of \( L(L > 3) \) since the ratio of the volume of a hypercube with radius 1 to the volume of a hypercube of
which approaches zero very rapidly as \( L \) increases. For \( L \to 10 \), about 4000 random numbers would be required to generate one pattern in the interior of a hypersphere by a rejection method as compared to 1 by our method.

References:


An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures

K. SAM SHANMUGA, SENIOR MEMBER, IEEE, FRED M. DICKY, MEMBER, IEEE, AND JAMES A. GREEN, STUDENT MEMBER, IEEE

Abstract—Edge detection and enhancement are widely used in image processing applications. In this paper we consider the problem of optimizing a frequency domain filter for detecting edges in digital pictures. The filter is optimum in that it produces maximum energy within a resolution interval at specified values in the vicinity of the edge.

We show that, in the continuous case, the filter transfer functions are specified in terms of the prolate spheroidal wave functions. In the discrete case, the filter transfer function is specified in terms of the sinc product of the first-order prolate spheroidal wave functions or in terms of the sampled values of an asymptotic approximation of the wave function. Both versions can be implemented via the fast Fourier transform (FFT). We show that the optimum filter is very effective for detecting blurred and noisy edges. Finally we compare the performance of the optimum edge-detection filter with other edge-detection filters using a variety of input images.

Index Terms—Edge detecting filters, edge enhancement, exponential approximation, digital picture processing, optimal edge detection, prolate-spheroidal wave functions.

I. INTRODUCTION

Edge detection is an important operation in a number of image processing applications such as in scene analysis and character recognition. Edges are defined at large and sudden changes in some image attribute, usually the brightness. The usual aim of edge detection is to locate edges belonging to boundaries of objects of interest. While the human eye performs this task quickly, the detection of edges is a complex task for automation. Some of the difficulties in edge detection are caused by noise in the image but much more so by the fact that edges are often blurred.

Many edge detection methods have been proposed for detecting and/or enhancing edges in digital images. Most of these procedures use local operations on the elements of the input picture.