An Intrinsic Dimensionality Estimator from Near-Neighbor Information

KARL W. PETTIS, THOMAS A. BAILEY, ANIL K. JAIN, MEMBER, IEEE, AND RICHARD C. DUBES, 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, noniterative 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 true dimensionality for a variety of artificial data sets and is fairly insensitive to the number of samples and to the algorithmic parameters. Comparisons between this new method and the global eigenvalue approach demonstrate the utility of our estimator.

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

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 can 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, explore 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.

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.

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 item 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 Ball [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 MDSCAL, due to Kruskal [2], [3] and Shepard [4].

The criterion for the goodness of the configuration produced by MDSCAL 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

$$S_{\text{Kruskal}} = \left\{ \left[\sum_{i<j} (d_{ij} - d_{ij})^2 \right] \Big/ \sum_{i<j} d_{ij}^2 \right\}^{1/2}$$

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 row and 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 far removed from one another. If they are similarities, large values represent very similar, or close, data items.

0162-8828/79/0100-0025$00.75

d_{ij} that would make the rank order of the interpoint distances the same as that of the proximities. The d_{ij} are obtained by solving the monotone regression problem. When $d_{ij} = d_{ij}$ for all i and j, stress is zero and a perfect configuration exists, according to the stress measure.

Stress is minimized by iteratively moving the points in the configuration from their initial randomly chosen positions according to a gradient-descent procedure that uses empirical acceleration coefficients. This procedure creates a configuration of points in a space of fixed dimensionality. The points are moved so as to achieve a monotone relation between interpoint distances and the information in the proximity matrix. The algorithm ceases when the magnitude of a gradient vector is relatively small. The intrinsic dimensionality itself is determined from a plot of the minimum stress versus dimensionality of the configuration space. One looks for a knee, or a flattening of the curve. In practice, a stress of 5 or 10 percent is considered "good" on an objective basis. Even a larger stress is acceptable if the researcher can interpret the axes in the configuration space and use the configuration itself to further the ends of the analysis.

When the input configuration consists of a set of patterns in high-dimensional space, one can generate a proximity matrix by computing, say, euclidean distances and proceeding as in MDSCAL. However, the approach usually taken with this type of input is to try to "unfold" the data, or flatten the swarm of patterns into a lower dimensional space. This approach can be viewed as a nonlinear projection. The first complete algorithm of this sort was suggested by Bennett [5] who was attempting to reduce the dimensionality of a signal space and uncover the number of parameters underlying signal generation. Bennett introduced an idea which has been adopted by almost all subsequent workers in intrinsic dimensionality. Bennett's idea was based on his observation that if points are uniformly distributed inside a sphere of radius r in an L-dimensional space and if

$$R_L = |X_1 - X_2|/(2r)$$

where X_1 and X_2 are random variables representing points in this sphere and R_L is the (normalized) euclidean distance between them, called the interpoint distance, then the variance of R_L is a decreasing function of L, which may be expressed as

$$L \operatorname{var}(R_L) \approx \text{constant}$$

where $\operatorname{var}(R_L)$ is the variance of R_L. Thus increasing the variance of the interpoint distances has the effect of decreasing the dimensionality of the representation, or "flattening" the swarm of patterns.

Bennett's algorithm involves two stages. The first stage moves the patterns (in the original pattern space) so as to increase the variance of the interpoint distances. The second stage adjusts the positions of the patterns so as to make the rank orders of interpoint distances in local regions the same in both spaces. These stages are repeated, with suitable normalizations, until the variance of the interpoint distances levels off, which indicates a flattening of the surface containing the patterns. The actual intrinsic dimensionality of the flattened surface is determined by the number of significant eigenvalues of the covariance matrix computed in the configuration space.

Chen 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 the two spaces. By contrast, MDSCAL uses rank orders of 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.

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

where d_{ij}^* 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 a configuration very similar to Sammon's could be generated from MDSCAL.

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 continuity [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 Schwartzmann 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 smoothes 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 a 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 to each pattern in the given set are identified. The subspace spanning the vectors from the ith pattern to its k nearest neighbors is constructed for all patterns. The angle between the $(k+1)$st 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, 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 near-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 $(X_1, X_2, \cdots, X_n)$ of L-dimensional patterns assumed to be drawn independently and governed by unknown density $p(\cdot)$. 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

$$\hat{p}(x) = \frac{k/n}{V} \tag{1}$$

where k is the number of near neighbors to x within a hypersphere of radius R_k about x and $V = V_d R_k^d$ is the volume of the hypersphere. Here, V_d is the volume of the unit d-dimensional hypersphere given by

$$V_d = \frac{(\pi)^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}.$$

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

$$\log(R_k) = (1/d)\log(k) + \log[(nV_d\hat{p}(x))^{-(1/d)}]. \tag{2}$$

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

Let $r_{k,x}$ be the distance from x to the kth 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 $r_{k,x}$ can be taken as (Appendix I)

$$f_{k,x}(r) = cdr^{d-1}\frac{(cr^d)^{k-1}}{\Gamma(k)}\exp(-cr^d), \quad \text{if } r>0$$
$$= 0, \quad \text{else} \tag{3}$$

where $c = np(x)\,V_d$. The expected value of $r_{k,x}$ is

$$E(r_{k,x}) = \int_0^\infty r f_{k,x}(r)\,dr = \frac{\Gamma\left(k+\frac{1}{d}\right)}{k^{1/d}\Gamma(k)}\left[\frac{k}{np(x)\,V_d}\right]^{1/d}. \tag{4}$$

Define a sample-averaged distance to the kth nearest neighbor over the set of patterns as

$$\bar{r}_k = (1/n)\sum_{i=1}^{n} r_{k,X_i}.$$

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

$$E(\bar{r}_k) = (1/n)\sum_{i=1}^{n} E(r_{k,X_i}) = \frac{1}{G_{k,d}}k^{1/d}C_n \tag{5}$$

where

$$G_{k,d} = \frac{k^{1/d}\Gamma(k)}{\Gamma\left(k+\frac{1}{d}\right)}$$

and

$$C_n = (1/n)\sum_{i=1}^{n}[np(X_i)\,V_d]^{-1/d}.$$

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

$$\log(G_{k,d}) + \log E(\bar{r}_k) = (1/d)\log(k) + \log(C_n). \tag{6}$$

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 $E(\bar{r}_k)$ we take the observed value of the random variable $\bar{r}_k$ computed from the given sample. Using this observed value in (6) defines an estimator $\hat{d}$ for d

$$\log(G_{k,\hat{d}}) + \log(\bar{r}_k) = (1/\hat{d})\log(k) + \log(C_n). \tag{7}$$

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

We solve for $\hat{d}$ iteratively. The initial estimator $\hat{d}_0$ is obtained by setting $\log(G_{k,\hat{d}})$ to zero and fitting a least square

regression line to a plot of $\log(\bar{r}_k)$ versus $\log(k)$ for $k = 1, \cdots, K$ for some K. We then use the value for $\log(G_{k,\hat{d}_0})$ and fit another regression line to obtain $\hat{d}_1$. We continue until reaching an i for which

$$|\hat{d}_i - \hat{d}_{i-1}| < \epsilon$$

for some ϵ. 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 LOGR(k) in Fig. 1 refers to the term $\log(\bar{r}_k)$ in (7)

$$\text{LOGR}(k) = \log\left[(1/n') \sum_j r_{k,X_j}\right]. \tag{8}$$

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 sample variance and distorted the estimate of d. Several schemes can be used to identify outliers. We define a measure of the average distance $m_{\max}$, and a measure of the spread of the distances $s_{\max}$, as follows:

$$m_{\max} = (1/n) \sum_{j=1}^{n} r_{K,X_j}$$

$$s^2_{\max} = (1/n - 1) \sum_{j=1}^{n} (r_{K,X_j} - m_{\max})^2.$$

The sum in (8) covers all n' values of j for which

$$r_{K,X_j} \leqslant m_{\max} + s_{\max}.$$

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 $d = 2$

$$\log(G_{k,d}) = \frac{d-1}{2kd^2} + \frac{(d-1)(d-2)}{12k^2d^3} - \frac{(d-1)^2}{12k^3d^4} - \frac{(d-1)(d-2)(d^2+3d-3)}{120k^4d^5} + 0(1/k^5).$$

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 LOGR(k) in Fig. 1 is simply to add $\log(G_{k,\hat{d}_{i-1}})$ for each k. The equation used to compute $\hat{d}_i$ is obtained from the standard equation for the slope of a regression line.

$$\hat{d}_i = \left[\frac{K \sum (\log k)(\log \bar{r}_k) - \left(\sum \log k\right)\left(\sum \log \bar{r}_k\right)}{K \sum (\log k)^2 - \left(\sum \log k\right)^2}\right]^{-1}.$$

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

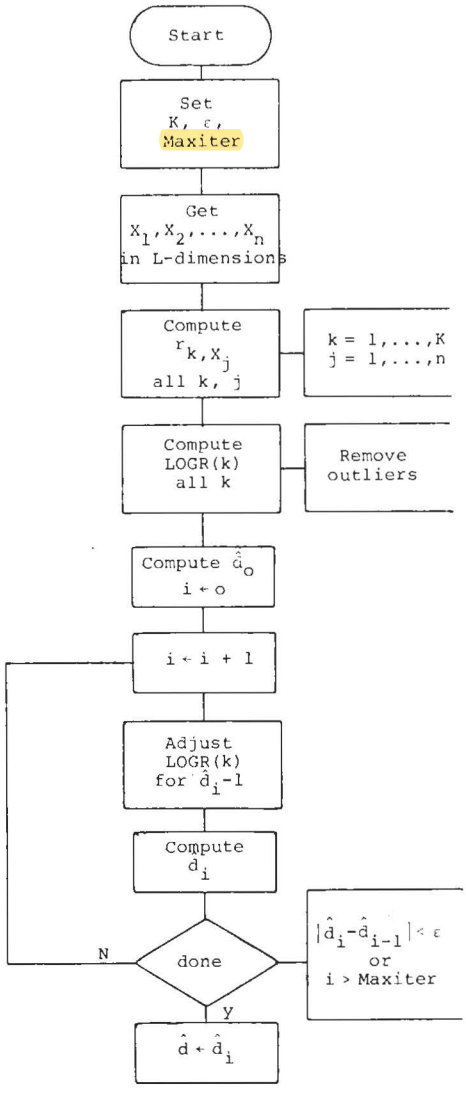


Fig. 1. Procedure for computing $\hat{d}$.

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 d at each sample, then average the estimates to obtain a global estimate of d. Specifically, suppose $E(r_{k,x})$ in (4) were replaced by r_{k,X_i}, the distance from X_i to its kth nearest neighbor. Letting

$$c_i = [np(X_i)\, V_d]^{-1/d}$$

and taking logarithms, we obtain

$$\log(G_{k,d}) + \log(r_{k,X_i}) = (1/d)\log(k) + \log(c_i).$$

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

$$(1/n) \sum_i \hat{d}_{X_i}$$

to estimate d.

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

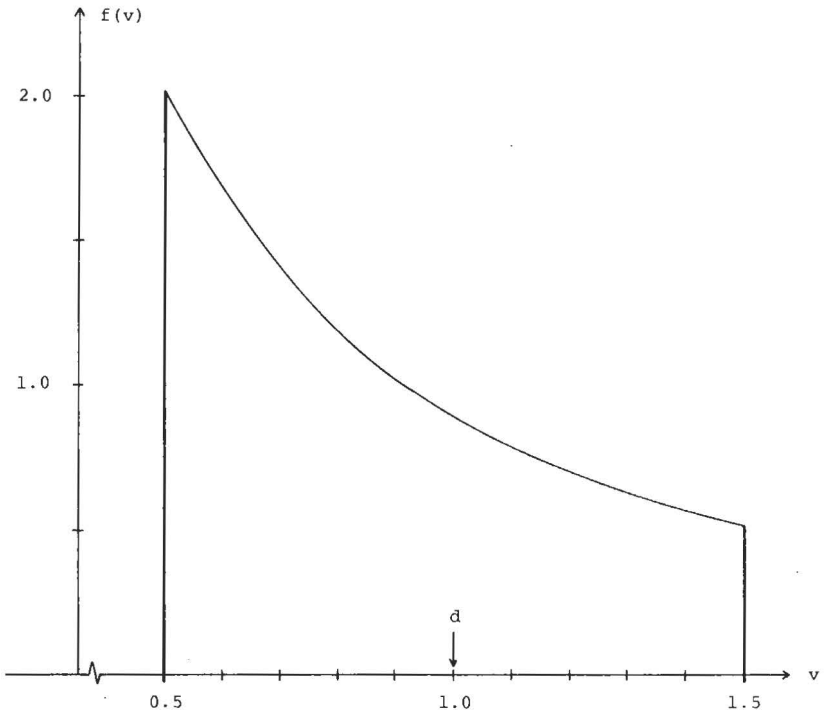


Fig. 2. Density of $\hat{d}$ in a special case (Section III-C).

$$\hat{d}_{X_i} = [(r_{2,X_i}/r_{1,X_i}) - 1]^{-1}$$

and $E(\hat{d}_{X_i})$ 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 kth and $(k+1)$st, (7) leads to

$$\log \frac{G_{k+1,\hat{d}}}{G_{k,\hat{d}}} + \log \frac{\bar{r}_{k+1}}{\bar{r}_k} = (1/\hat{d}) \log \frac{k+1}{k}. \tag{9}$$

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

$$\log \frac{G_{k+1,\hat{d}}}{G_{k,\hat{d}}} = (1/\hat{d}) \log \frac{k+1}{k} - \log \left[1 + \frac{1}{k\hat{d}}\right]. \tag{10}$$

Substituting (10) into (9)

$$(\bar{r}_{k+1}/\bar{r}_k) = 1 + \frac{1}{k\hat{d}}.$$

Solving for $\hat{d}$

$$\hat{d} = \frac{\bar{r}_k}{k(\bar{r}_{k+1} - \bar{r}_k)}. \tag{11}$$

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 (11) shows that

$$\hat{d} = \bar{r}_1/(\bar{r}_2 - \bar{r}_1). \tag{12}$$

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

$$X_1 \leqslant X_2 \leqslant X_3.$$

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

$$f_{A,B}(a, b) = 12(1 - a - b), \quad \text{for } 0 \leqslant a \leqslant 1 \text{ and } 0 \leqslant b \leqslant \min(a, 1 - a)$$
$$= 0, \quad \text{else.}$$

The sample-averaged distances to the nearest neighbors are $\bar{r}_1 = (A + 2B)/3$ and $\bar{r}_2 = (3A + 2B)/3$. 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

$$f(v) = 2/(v + 0.5)^2, \quad \text{for } 0.5 \leqslant v \leqslant 1.5$$
$$= 0, \quad \text{else.}$$

The mean and variance of $\hat{d}$ are

$$E(\hat{d}) = 2 \ln (2) - 0.5 = 0.886$$

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

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 $\hat{d}$ When $K = 2$

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

$$E(\hat{d}) = \int_0^\infty \int_0^\infty \frac{1}{(t/s) - 1} f^{(n)}_{\bar{r}_1, \bar{r}_2}(s, t)\, ds\, dt$$

where $f^{(n)}_{\bar{r}_1, \bar{r}_2}(.,.)$ denotes the joint density for the random variables

$$\bar{r}_k = (1/n) \sum_{i=1}^{n} r_{k, X_i}, \qquad \text{for} \quad k = 1, 2.$$

Since $\bar{r}_1$ and $\bar{r}_2$ 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_{1,X_i} and r_{2,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 $\bar{r}_1$ and $\bar{r}_2$ approaches a δ-function as n grows large

$$f^{(n)}_{\bar{r}_1, \bar{r}_2}(s, t) \xrightarrow[n \to \infty]{} \delta[s - E(\bar{r}_1), t - E(\bar{r}_2)].$$

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

$$E(\hat{d}) \to \frac{1}{[E(\bar{r}_1)/E(\bar{r}_2)] - 1} = d.$$

In addition, it is clear that

$$\operatorname{var}(\hat{d}) \xrightarrow[n \to \infty]{} 0.$$

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

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 radially projecting the three-dimensional Gaussian data onto the surface of a sphere ($L = 3, d = 2$) 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).

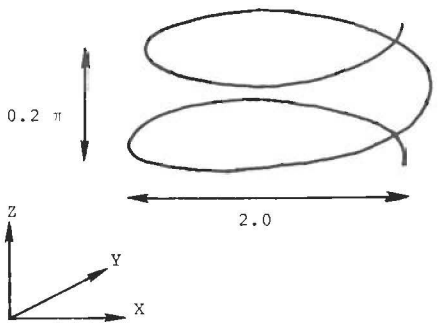


Fig. 3. Sketch of helix data.

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

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

which has impulse response

$$h(n) = |q|^{n-1} \frac{\sin(n\theta)}{\sin(\theta)}, \qquad \text{if} \quad n \geq 0$$

where

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

We observed $h(n)$ for $n = 1, 2, \cdots, 20$. 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:

$$x = \cos\theta$$
$$y = \sin\theta$$
$$z = 0.1(\theta)$$
$$0 \leq \theta \leq 4\pi.$$

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:

$$\begin{pmatrix} \sigma_x^2 & & \circ \\ & \sigma_y^2 & \\ \circ & & \sigma_z^2 \end{pmatrix}$$

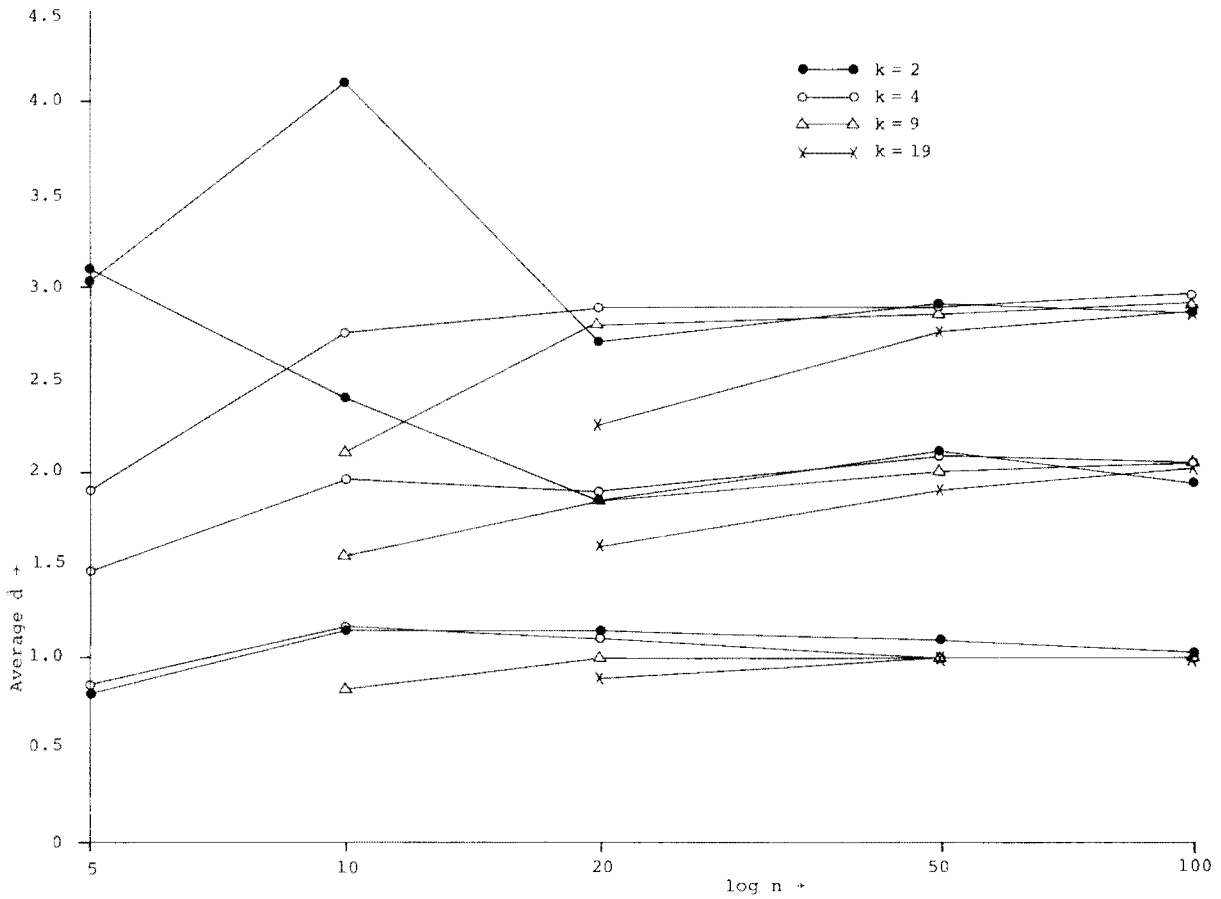


Fig. 4. Average $\hat{d}$ for Gaussian data.

The values of σ_x, σ_y, and σ_z and the corresponding d values are given below.

a) $\sigma_x = 100$, $\sigma_y = \sigma_z = 1$, $d = 1$;
b) $\sigma_x = 10$, $\sigma_y = \sigma_z = 1$, $d = 1$ or 3;
c) $\sigma_x = \sigma_y = \sigma_z = 1$, $d = 3$;
d) $\sigma_x = 1$, $\sigma_y = \sigma_z = 10$, $d = 2$ or 3;
e) $\sigma_x = 1$, $\sigma_y = \sigma_z = 100$, $d = 2$.

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.

B. 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 $\hat{d}$ for the ten Monte Carlo runs were computed for n, K, and data type fixed. The average values of $\hat{d}$ 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 $\hat{d}$ is very close to d when $n \geq 10$. The filter data present an unusual case in that the average $\hat{d}$ 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 $\hat{d}$ 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 $n \geq 50$. For the hyperellipsoidal 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), $\hat{d}$ approaches the desired values, while for cases b) and d), $\hat{d}$ is within the range of expected values. Other values of K give comparable results.

In all cases, the variance of $\hat{d}$, 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 $\hat{d}$. 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 I 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

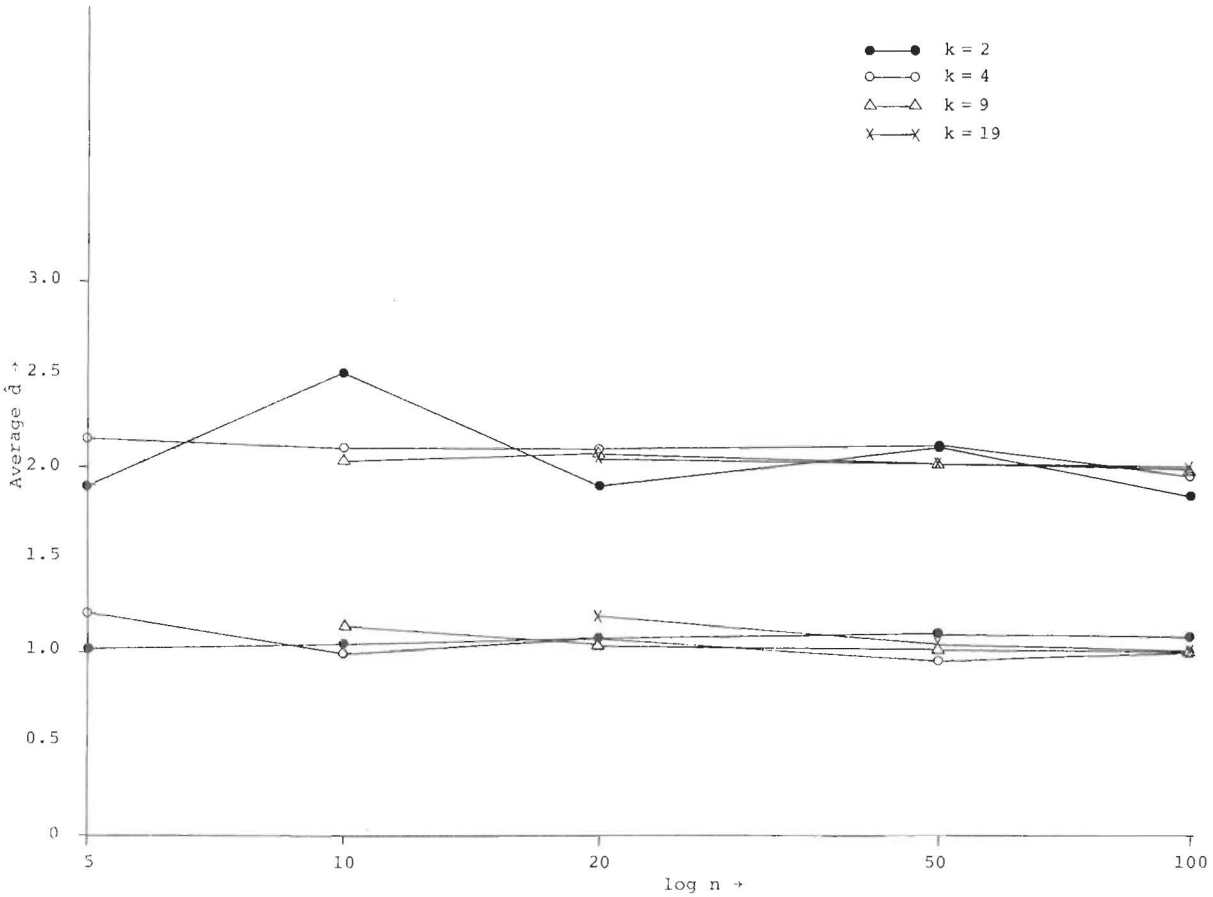


Fig. 5. Average $\hat{d}$ for surface data.

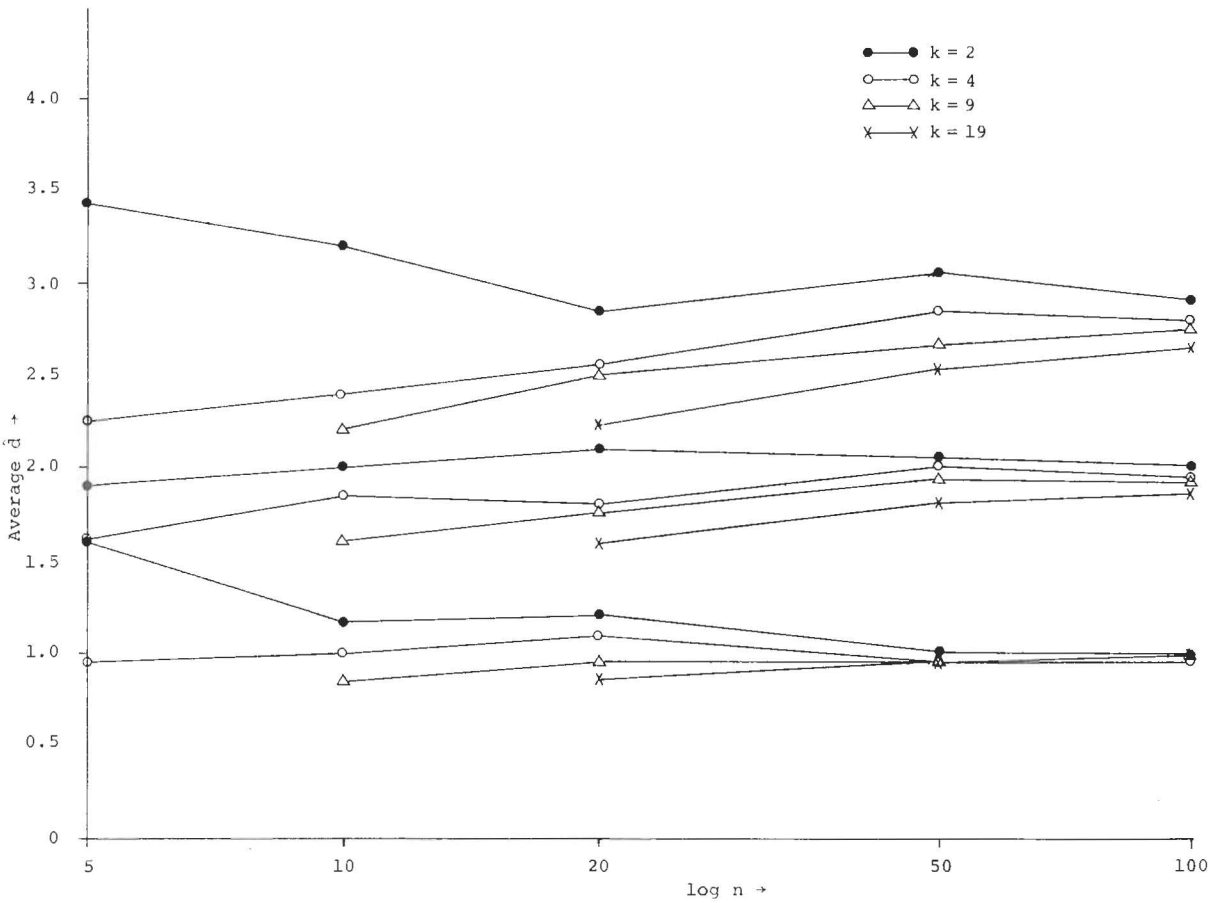


Fig. 6. Average $\hat{d}$ for interior data.

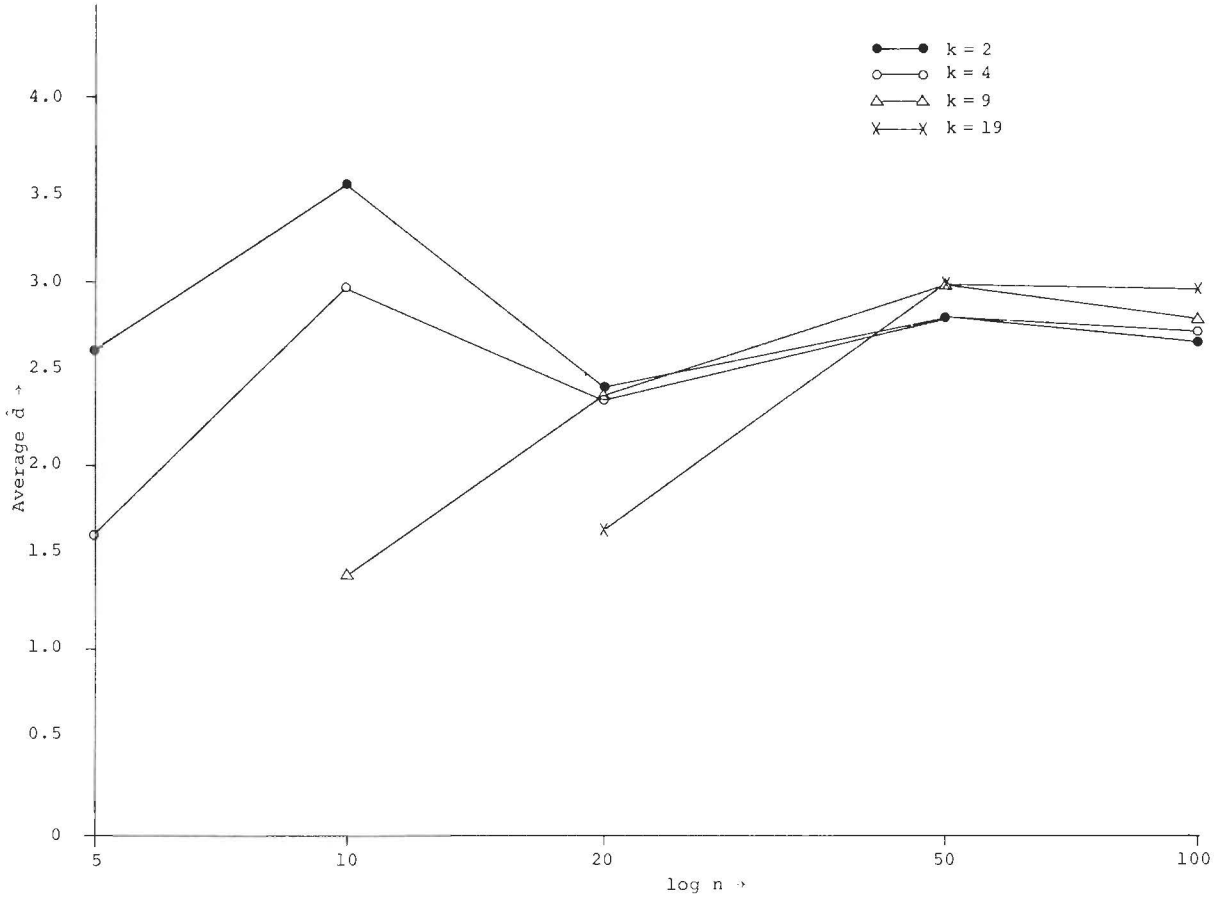


Fig. 7. Average $\hat{d}$ for filter data.

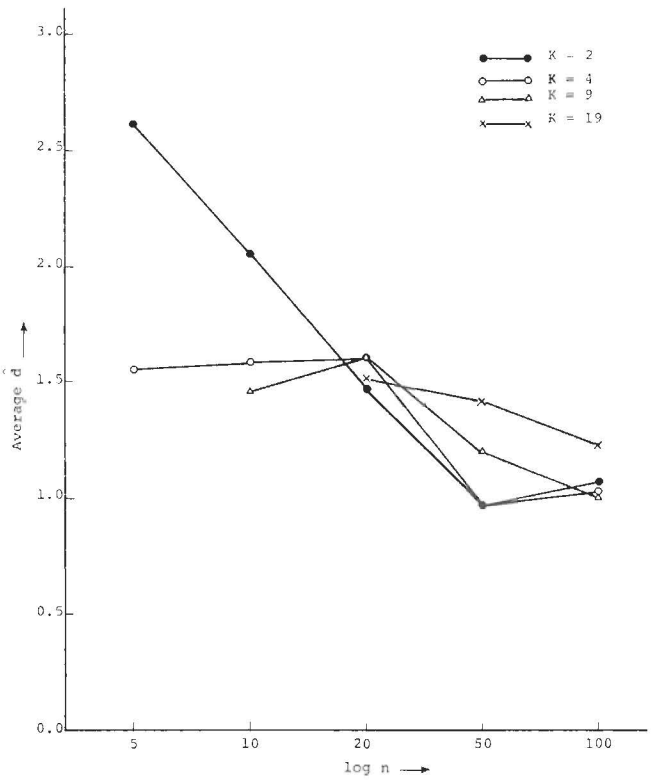


Fig. 8. Average $\hat{d}$ for helix data.

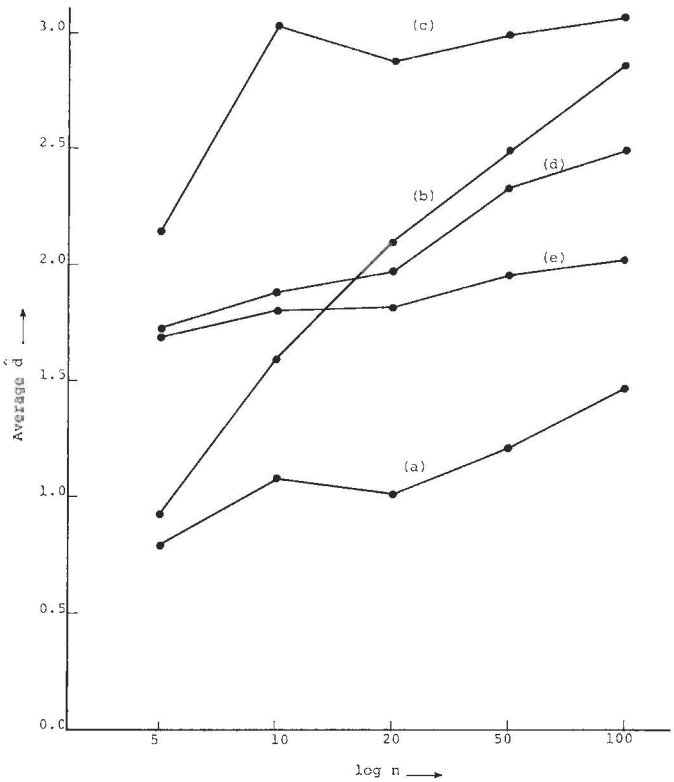


Fig. 9. Average $\hat{d}$ for hyperellipsoidal data ($K = 4$).

be identical. Using eigenvalues to estimate intrinsic dimensionality requires judgement about which eigenvalues are "significant." One hopes that the eigenvalues are either large or very small to simplify this judgement. For instance, the eigenvalues computed from the three-dimensional surface data show the same character 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 failed. The difficulty 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 main applications. First, it can follow one of the data-flattening algorithms [5], [6], [10], [17] and provide a more realistic estimate of intrinsic dimensionality than the global eigenvalue method, especially when the output of the data-flattening algorithm is highly warped. Second, our estimator can guide the choice of an appropriate number of dimensions for representing 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 base a data-flattening algorithm on the criterion of minimizing our estimator as opposed to minimizing stress or minimizing variance of interpoint distances. We hope to investigate this possibility in the future.

Our estimator exhibited good results for a variety of artificial data. Of special interest was its performance on surface data where the global eigenvalue method failed. Our method does not require interaction between user and data during computation, so it is suitable for a wider variety of situations and computer installations than some other nonglobal methods [18].

Our method of estimating intrinsic dimensionality uses distances to the nearest neighbors and thus is dominated by local information. If the measurements are noisy or the hypersurface 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 $n \gg K$, local information dominates the estimator; while for $n \sim K$, almost all samples are required to capture K near neighbors.

Two effects observed in our empirical study require comment. First, in all cases, the value of K chosen did not materially 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 curvature, then the number of samples needed to obtain a good estimate of d has to be large so that the local neighborhood used for our estimate is small. Second, it appears that edge effects negatively bias our estimator (Fig. 6) even though our algorithm excludes the outliers. Edge effects are ameliorated when the number of samples increases.

The statistical properties of our estimator are not immediately apparent. The distribution of $\hat{d}$ for arbitrary 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 uniformly distributed on an interval. We also proposed a plausible argument for consistency of $\hat{d}$ when $K = 2$.

The algorithm for computing $\hat{d}$ is straightforward. It involves only three parameters, ϵ, 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 neighbors, so the amount of computation varies as n^2L. 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 L have recently been developed [27], [28] and our algorithm will benefit from their use.

Appendix I

Density of Distance to kth Nearest Neighbor

The density function for $r_{k,x}$, the distance from x to its kth nearest neighbor among $(X_1, X_2, \cdots, X_n)$, is required in Section III-A, and is derived in this Appendix. The assumptions are as follows.

1) The underlying density $p(\cdot)$ is constant over $S_x(r)$, a sphere of radius r centered at x whose volume is $V = V_d r^d$.

2) Observing the samples which fall in $S_x(r)$ is equivalent to performing a sequence of n Bernoulli trials with probability $p(x)\,V$ of "success," or falling within $S_x(r)$. Let $N_x(r)$ be a random variable denoting the number of samples in $S_x(r)$. The previous assumptions imply that

$$\Pr[N_x(r) = m] = \binom{n}{m} [p(x)\,V]^m [1 - p(x)\,V]^{n-m}, \qquad 0 \leqslant m \leqslant n. \quad (1.1)$$

The density function $f_{k,x}(\cdot)$ for $r_{k,x}$ can be expressed in terms of $N_x(r)$ as follows:

$$\begin{aligned} f_{k,x}(r) &= \lim_{\Delta r \to 0} (1/\Delta r) \Pr(r \leqslant r_{k,x} \leqslant r + \Delta r) \\ &= \lim_{\Delta r \to 0} (1/\Delta r) \Pr[N_x(r + \Delta r) = k \,|\, N_x(r) = k - 1] \\ &\quad \cdot \Pr[N_x(r) = k - 1]. \end{aligned}$$

Assumption 1) implies that the first factor is asymptotically $np(x)\,\Delta V$ where

$$\Delta V = V_d(r + \Delta r)^d - V_d(r)^d = V_d(dr^{d-1}\Delta r) + 0(\Delta r).$$

Thus,

$$f_{k,x}(r) = np(x)\,V_d\,dr^{d-1} \Pr[N_x(r) = k - 1].$$

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

TABLE I
MOST SIGNIFICANT EIGENVALUES (NORMALIZED)

Data set	n = 5	10	20	50	100
3-Dim Gaussian	61.14	47.08	41.00	42.14	44.06
	33.14	34.91	38.18	31.28	32.61
	5.71	18.01	20.82	26.58	23.33
2-Dim Gaussian	62.67	76.08	70.63	54.93	52.72
	37.32	23.92	29.37	45.07	47.28
3-Dim Surface of Sphere	55.86	48.54	43.94	39.69	36.94
	29.26	28.70	34.15	31.55	34.37
	14.88	22.77	21.91	28.76	28.69
2-Dim on the Circle	59.72	66.37	56.16	58.74	53.72
	40.28	33.63	43.84	41.26	46.28
3-Dim Interior of Sphere	50.24	49.85	45.82	39.03	37.58
	33.22	30.11	35.15	31.19	33.40
	16.54	20.04	19.03	29.78	29.02
2-Dim Interior of Circle	63.60	70.14	53.60	60.01	52.87
	36.40	29.86	46.40	39.99	47.13
Filter	67.3	92.2	78.4	64.8	53.1
	27.7	4.5	10.0	13.7	24.3
	5.0	1.8	5.8	10.7	8.3
Helix	64.57	58.78	51.13	45.95	49.45
	27.77	30.67	37.49	45.00	41.61
	7.66	10.55	11.38	9.05	8.94
Hyperellipsoidal					
(a)	99.98	99.97	99.98	99.98	99.98
(b)	98.24	97.87	98.04	98.46	98.43
(c)	73.78	60.84	46.96	39.72	38.77
	18.22	22.48	32.08	33.48	31.89
	8.00	16.67	20.97	26.79	29.33
(d)	77.06	75.23	53.83	55.43	51.95
	22.80	24.43	45.77	43.94	47.43
(e)	77.13	75.47	54.02	55.78	52.27
	22.87	24.53	45.97	44.22	47.72

$$\Pr[N_x(r) = k-1] \approx \frac{[np(x)\,V]^{k-1}}{\Gamma(k)} \exp[-np(x)\,V].$$

Substituting and letting

$$c = np(x)\,V_d$$

produces the final equation for the density

$$f_{k,x}(r) = \frac{cdr^{d-1}(cr^d)^{k-1}}{\Gamma(k)} \exp(-cr^d).$$

The Poisson approximation is valid if n is large, $p(x)\,V_d r^d$ is small, and $np(x)\,V_d r^d$ is moderate.

APPENDIX 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 $d \geq 1$

$$0 \leq \log(G_{k,d}) = \log \frac{k^{1/d}\Gamma(k)}{\Gamma\left(k+\frac{1}{d}\right)} \leq \log \frac{1}{\Gamma\left(1+\frac{1}{d}\right)} = \log(G_{1,d}). \tag{2.1}$$

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

$$0 \leq g_k(z) \leq g_1(z), \quad \text{for} \quad 0 \leq z \leq 1$$

where

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

Since

$$g_k(0) = g_k(1) = 0, \quad \text{for all} \quad k$$

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

$$g_1''(z) \leq g_k''(z) \leq 0. \tag{2.2}$$

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

$$g_k''(z) = -\sum_{i=k}^{\infty} (i+z)^{-2}.$$

In addition,

$$\log(G_{1,d}) \leq 0.12$$

the maximum occurring at $d \approx 2.17$. Thus, we conclude that $0 \leq \log(G_{k,d}) \leq 0.12$ for $k \geq 1$ and $d \geq 1$.

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

If z is an L-dimensional Gaussian random variable with mean vector 0 and covariance matrix $\sigma^2 I$, then because there is no preferred direction from the origin

$$\frac{z}{\|z\|}$$

is uniform over the interior of an L-dimensional hypersphere. [29]. Also, if r is a random variable uniform on (0, 1), then

$$\frac{z}{\|z\|} r^{(1/L)}$$

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 hypersphere are grossly inferior to this method even for moderate values of $L (L \geq 3)$ since the ratio of the volume of a hypersphere with radius 1 to the volume of a hypercube of

side 2 is

$$\frac{\pi^{L/2}}{\Gamma\left(\frac{L}{2}+1\right)2^{L}}$$

which approaches zero very rapidly as L increases. For $L = 10$, about 4000 random numbers would be required to generate one pattern in the interior of a hypersphere by a rejection method as compared to 11 by our method.

References

[1] G. H. Ball, "Data analysis in the social sciences: What about the details?," in *Proc. Fall Joint Computer Conf.*, 1965, pp. 533-554.

[2] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1-27, Mar. 1964.

[3] ——, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, pp. 115-129, June 1964.

[4] R. N. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function," *Psychometrika*, vol. 27, pp. 125-140, June 1962.

[5] R. S. Bennett, "The intrinsic dimensionality of signal collections," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 517-525, Sept. 1969.

[6] C. K. Chen and H. C. Andrews, "Nonlinear intrinsic dimensionality computations," *IEEE Trans. Comput.*, vol. C-23, pp. 178-184, Feb. 1974.

[7] R. N. Shepard and J. D. Carroll, "Parametric representation of nonlinear data structures," in *Multivariate Analysis*, P. R. Krishnaiah, Ed. New York: Academic, 1966, pp. 561-592.

[8] J. W. Sammon, Jr., "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-18, pp. 401-409, May 1969.

[9] J. B. Kruskal, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-20, p. 1614, Dec. 1971.

[10] C. L. Chang and R. C. T. Lee, "A heuristic relaxation method for nonlinear mapping in cluster analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 197-200, Mar. 1973.

[11] J. B. Kruskal and R. E. Hart, "A geometric interpretation of diagnostic data from a digital machine," *Bell Syst. Tech. J.*, vol. 45, pp. 1299-1338, 1966.

[12] J. B. Kruskal, "Linear transformations of multivariate data to reveal clustering," in *Multidimensional Scaling*, vol. 1, *Theory*, R. N. Shepard, A. K. Romney, and S. B. Nerlove. New York: Seminar Press, 1972.

[13] J. B. Kruskal and J. D. Carroll, "Geometrical models and badness-of-fit functions," in *Multivariate Analysis-II*, P. R. Krishnaiah, Ed. New York: Academic, 1969, pp. 639-671.

[14] R. N. Shepard, "Representation of structure in similarity data—problems and prospects," *Psychometrika*, vol. 39, pp. 373-421, Dec. 1974.

[15] A. K. Romney, R. N. Shepard, and S. B. Nerlove, *Multidimensional Scaling*, vol. 2, *Applications*. New York: Seminar Press, 1972.

[16] ——, *Multidimensional Scaling*, vol. I, *Theory*. New York: Seminar Press, 1972.

[17] D. H. Schwartzmann and J. J. Vidal, "An algorithm for determining the topological dimensionality of point clusters," *IEEE Trans. Comput.*, vol. C-24, pp. 1175-1182, Dec. 1975.

[18] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vol. C-20, pp. 176-183, Feb. 1971.

[19] G. V. Trunk, "Statistical estimation of the intrinsic dimensionality of a noisy signal collection," *IEEE Trans. Comput.*, vol. C-25, pp. 165-171, Feb. 1976.

[20] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973, p. 87.

[21] L. Breiman, *Probability*. Reading, MA: Addison-Wesley, 1968, pp. 237-238.

[22] P. J. Davis, "Gamma function and related functions," *Handbook of Mathematical Functions*, M. Abramowitz and I. A. Stegun, Eds. Washington, DC: U.S. Government Printing Office, 1965, pp. 253-293.

[23] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Trans. Comput.*, vol. C-22, pp. 1025-1034, Nov. 1973.

[24] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21-26, Jan. 1967.

[25] T. W. Calvert, "Nonorthogonal projections for feature extraction in pattern recognition," *IEEE Trans. Comput.*, vol. C-19, pp. 447-452, May 1970.

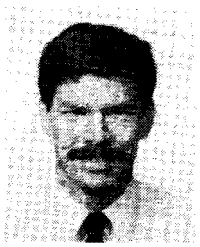
[26] R. C. T. Lee, J. R. Slagle, and H. Blum, "A triangulation method for the sequential mapping of points from N-space to two-space," *IEEE Trans. Comput.*, vol. C-26, pp. 288-292, Mar. 1977.

[27] J. H. Friedman, F. Baskett, and L. J. Shustek, "An algorithm for finding nearest neighbors," *IEEE Trans. Comput.*, vol. C-25, pp. 1000-1006, Oct. 1975.

[28] T. P. Yunck, "A technique to identify nearest neighbors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 678-683, Oct. 1976.

[29] M. E. Muller, "A note on a method for generating points uniformly on N-dimensional spheres," *Commun. Ass. Comput. Mach.*, vol. 2, pp. 19-20, Apr. 1959.

Karl W. Pettis was born in Gainesville, FL, on October 24, 1954. He received the B.S. degree in mathematics in 1975 and the M.S. degree in computer science in 1977 from Michigan State University, East Lansing, MI. He is now a doctoral student at Yale University, New Haven, CT, in computer science.



Thomas A. Bailey was born in Chicago, IL, on September 29, 1942. He received the B.S. degree from Alma College, Alma, MI, in 1964 and the M.S. degree in physics from the University of Colorado, Boulder, in 1969. He is currently completing requirements for the Ph.D. degree in computer science at Michigan State University, East Lansing.

From 1969 to 1976 he was a member of the teaching faculty at Alma College. Since 1976 he has been a Research Assistant with the Department of Computer Science at Michigan State University. His current research interests concern applications of random graph theory to questions of cluster validity.

Mr. Bailey is a member of the Association for Computing Machinery.

Anil K. Jain (S'70-M'72) was born in Basti, India, on August 5, 1948. He received the B. Tech. degree with distinction from Indian Institute of Technology, Kanpur, in 1969, and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, Columbus, in 1970 and 1973, respectively. He was a recipient of the National Merit Scholarship in India.

From 1971 to 1972 he was a Research Associate at the Communications and Control Systems Laboratory, Ohio State University, work-

ing on dimensionality and sample size problems in statistical pattern recognition. Then, from 1972 to 1974, he was an Assistant Professor in the Computer Science section at Wayne State University, Detroit, MI. In 1974 he joined the Computer Science Department at Michigan State University where he is currently an Associate Professor. His researeh interests are in the areas of pattern recognition and image processing.

Dr. Jain is a member of the Association for Computing Machinery, the Pattern Recognition Society, and Sigma Xi.

Richard C. Dubes (S'58-M'64) was born in Chicago, IL, in 1934. He received the B.S. degree from the University of Illinois, Urbana, in 1956, and the M.S. and Ph.D. degrees from Michigan State University, East Lansing, in 1959 and 1962, respectively, all in electrical engineering.

In 1956 and 1957 he was a member of the Technical Staff of the Hughes Aircraft Company, Culver City, CA. From 1957 through 1968 he served as Graduate Assistant, Research Assistant, Assistant Professor, and Associate Professor in the Electrical Engineering Department at Michigan State University. In 1969 he joined the Computer Science Department at Michigan State University and became Professor in 1970. He is the author of *The Theory of Applied Probability* (Prentice-Hall, 1968) and several technical papers and reports. He has served as a consultant to the Lear-Siegler Corporation, Grand Rapids, MI, and the J. M. Richards Laboratory, Detroit, MI. His areas of technical interest include pattern recognition, clustering, decision theory, and application of data analysis methods to the medical area.

Dr. Dubes is a member of the Pattern Recognition Society and Sigma Xi.

An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures

K. SAM SHANMUGAM, SENIOR MEMBER, IEEE, FRED M. DICKEY, 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 spatial frequency domain filters for detecting edges in digital pictures. The filter is optimum in that it produces maximum energy within a resolution interval of specified width in the vicinity of the edge.

We show that, in the continuous case, the filter transfer function is specified in terms of the prolate spheroidal wave function. In the discrete case, the filter transfer function is specified in terms of the sampled values of the first-order prolate spheroidal wave function 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.

Manuscript received March 13, 1978, revised May 1, 1978. This work was partially supported by the Nuclear Regulatory Commission under Contract NRC-04-77-133.

K. S. Shanmugam is with the Department of Electrical Engineering, Wichita State University, Wichita, KS 67208.

F. M. Dickey is with the Electro-Optics Group, Boeing Wichita Co., Wichita, KS 67210.

J. A. Green was with the Department of Electrical Engineering, Wichita State University, Wichita, KS 67208. He is now with Microtech Inc., Wichita, KS.

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 as 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 easily, the detection of edges is a complex task to automate. 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

0162-8828/79/0100-0037$00.75