On Local Intrinsic Dimension Estimation and Its Applications

Kevin M. Carter, *Student Member, IEEE*, Raviv Raich, *Member, IEEE*, and Alfred O. Hero III, *Fellow, IEEE*

***Abstract*—In this paper, we present multiple novel applications for local intrinsic dimension estimation. There has been much work done on estimating the global dimension of a data set, typically for the purposes of dimensionality reduction. We show that by estimating dimension locally, we are able to extend the uses of dimension estimation to many applications, which are not possible with global dimension estimation. Additionally, we show that local dimension estimation can be used to obtain a better global dimension estimate, alleviating the negative bias that is common to all known dimension estimation algorithms. We illustrate local dimension estimation's uses towards additional applications, such as learning on statistical manifolds, network anomaly detection, clustering, and image segmentation.**

***Index Terms*—Geodesics, image segmentation, intrinsic dimension, manifold learning, nearest neighbor graph.**

I. Introduction

TECHNOLOGICAL advances in both sensing and media storage have allowed for the generation of massive amounts of high-dimensional data and information. Consider the class of applications that generate these high-dimensional signals: e.g., digital cameras capture images at enormous resolutions; dozens of video cameras may be filming the exact same object from different angles; planes randomly drop hundreds of sensors into the same area to map the terrain. While this has opened a wealth of opportunities for data analysis, the problem of the *curse of dimensionality* has become more substantial, as many learning algorithms perform poorly in high dimensions. While the data in these applications may be represented in high dimensions, strictly based upon the immense capacity for data retrieval, it is typically concentrated on lower dimensional subsets—manifolds—of the measurement space. This allows for significant dimension reduction with minor or no loss of information. The point at which the data can be reduced with minimal loss is related to the *intrinsic dimensionality* of the manifold supporting the data. This measure may be interpreted as the minimum number of parameters required to describe the data [1].

When the intrinsic dimension is assumed constant over the data set, several algorithms [2]–[5] have been proposed to estimate the dimensionality of the manifold. In several problems of practical interest, however, data will exhibit varying dimensionality, as there may lie multiple manifolds of varying dimension within the data. This is easily viewed in images with different textures or in classification tasks in which data from different classes is generated by unique probability density functions (pdfs). In these situations, the local intrinsic dimension may be of more importance than the global dimension. In previous work [6], we illustrated the process of local dimension estimation, in which a dimension estimate is obtained for each sample within the data, rather than a single dimension estimate for the entire set.

In this paper, we focus on the applications of local dimension estimation. One immediate benefit is using local dimension to estimate the global dimension of a data set. To our knowledge, every method of estimating intrinsic dimension has expressed an issue with a negative bias. While insufficient sampling is a common source of this bias, a significant portion is a result of samples near the boundaries or edges of a manifold. These regions appear to be low dimensional when sampled and contribute a strong negative bias to the global estimate of dimension. We additionally utilize local dimension estimation for the purposes of dimensionality reduction. Typically, this has been presented for Riemannian manifolds in a Euclidean space [7]–[9], in which the data contain a single manifold of constant dimension lying in $\mathbb{R}^d$. We extend this to the problem of estimation and reduction of dimensionality on statistical manifolds, in which the points on the manifold are pdfs rather than points in a Euclidean space.

We continue by showing novel applications in which the exact dimension of the data is of no immediate concern, but rather the differences between the local dimensions. Dimensionality can be viewed as the number of *degrees of freedom* in a data set, and as such may be interpreted as a measure of data *complexity*. By comparing the local dimension of samples within a data set, we are able to identify different subsets of the data for analysis. For example, in a time-series data set, the intrinsic dimensionality may change as a function of time. By viewing each time step as a sample, we can identify changes in the system at specific time points. We illustrate this ability by finding anomalous activity in a router network. Additionally, the identification of subsets within the data allows for the immediate application of clustering and image segmentation. There has been much work presented on using fractal dimension estimation for image and tex-

Manuscript received November 05, 2008; revised July 20, 2009. . The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cedric Richard. This work was supported in part by the National Science Foundation under Grant CCR-0325571.

K. M. Carter is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: kmcarter@umich.edu).

R. Raich is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail: raich@eecs.oregonstate.edu).

A. O. Hero III is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: hero@umich.edu).

Digital Object Identifier 10.1109/TSP.2009.2031722

ture segmentation [10], [11]. In this paper, we do not make the model assumption that textures may be represented as a collection of fractals [12], and instead segment images using a novel method based on Euclidean dimension. We show that by using "neighborhood smoothing" [13] over the dimension estimates, we are able to find the regions that exhibit differing complexities, and use the smoothed dimension estimates as identifiers for the clusters/segments.

The organization of this paper is as follows: We give an overview of the two-dimension estimation algorithms we will utilize in our simulations in Section II. In Section III, we describe the process of neighborhood smoothing as a means of postprocessing for local dimension estimation. We illustrate the various novel applications of local dimension estimation in Section IV, including debiasing for global dimension estimation, manifold learning, anomaly detection, clustering, and image segmentation. Lastly, we offer a discussion and present areas for future work in Section V.

II. Dimension Estimation

We will now present two algorithms for dimension estimation: the k-nearest neighbor (k-NN) algorithm [4], [14] and the maximum likelihood estimator (MLE) method [5]. Please note that this paper makes no attempts to claim superiority of these algorithms over others. While there are many algorithms available for dimension estimation, we focus on these two as a means for illustrating the applications we later present. By utilizing two distinct methods, we hope to quell any concerns that our applications are algorithm dependent. For a thorough survey of intrinsic dimension estimation methods, we encourage the reader to view [15] and [16], as well as more recent work [17]–[20]

A. The k-Nearest Neighbor Algorithm for Dimension Estimation

Let $\boldsymbol{X}_n = \{x_1, \ldots, x_n\}$ be n independent identically distributed (i.i.d.) random vectors with values in a compact subset of $\mathbb{R}^d$. The (1-)nearest neighbor of x_i in $\boldsymbol{X}_n$ is given by

$$\arg \min_{x \in \boldsymbol{X}_n \setminus \{x_i\}} D(x, x_i)$$

where $D(x, x_i)$ is an appropriate distance measure between x and x_i; for the purposes of this paper, let us define $D(x, x_i) = \|x - x_i\|$, the standard Euclidean (L_2) distance. For a general integer $k \geq 1$, the k-nearest neighbor of a point is defined in a similar way. The k-NN graph assigns an edge between each point in $\boldsymbol{X}_n$ and its k-nearest neighbors. Let $\mathcal{N}_{k,i} = \mathcal{N}_{k,i}(\boldsymbol{X}_n)$ be the set of k-nearest neighbors of x_i in $\boldsymbol{X}_n$. The total edge length of the k-NN graph is defined as

$$L_{\gamma,k}(\boldsymbol{X}_n) = \sum_{i=1}^{n} \sum_{y \in \mathcal{N}_{k,i}} D(x, x_i)^{\gamma} \tag{1}$$

where $\gamma > 0$ is a power weighting constant.

For many data sets of interest, the random vectors $\boldsymbol{X}_n$ are constrained to lie on an m-dimensional Riemannian submanifold $\mathcal{M}$ of $\mathbb{R}^d$ ($m < d$). Under this framework, the asymptotic behavior of (1) is given as

$$L_{\gamma,k}(\boldsymbol{X}_n) = n^{\alpha(m)} c + \epsilon_n \tag{2}$$

where $\alpha(m) = (m - \gamma)/m$, c is a constant with respect to $\alpha(m)$ that depends on the Rényi entropy of the distribution of the manifold and ϵ_n is an error residual [6]. Note that for ease of notation, we will denote $\alpha(m)$ simply as α, except where the explicit expression is desirable (e.g., optimizing over m).

As noisy measurements can lead to inaccurate estimates, the intrinsic dimension $\hat{m}$ should be estimated using a nonlinear least squares solution. By calculating sampled graph lengths over varying values of n, the effect of noise ϵ_n can be diminished. In order to calculate graph lengths for differing sample sizes on the manifold, it is necessary to randomly subsample from the full set $\boldsymbol{X}_n = \{x_1, \ldots, x_n\}$, utilizing the nonoverlapping block bootstrapping method [21]. Specifically, let $\boldsymbol{X}'_n = \{x_{(1)}, \ldots, x_{(n)}\}$ be a spatially or temporally sorted version of $\boldsymbol{X}_n$, and let w be an integer satisfying $w < n/Q$. Define the blocks $\mathcal{B}_i = (x_{((i-1)w+1)}, \ldots, x_{(iw)})$, $i = 1, \ldots, n/w$. As such, we may now redefine $\boldsymbol{X}'_n = \{\mathcal{B}_1, \ldots, \mathcal{B}_{n/w}\}$.

Let $\{p_1, \ldots, p_Q\}$ be Q integers such that $1 \leq p_1 < \ldots, < p_Q \leq n/w$. For each value of $p \in \{p_1, \ldots, p_Q\}$, randomly draw N bootstrap datasets $\boldsymbol{X}_p^j$, $j = 1, \ldots, N$, with replacement, where the p blocks of data points within each $\boldsymbol{X}_p^j$ are chosen from the entire data set $\boldsymbol{X}'_n$ independently. From these samples, define $\boldsymbol{L}_n = \{L_{\gamma,k}(\boldsymbol{X}_p^1), \ldots, L_{\gamma,k}(\boldsymbol{X}_p^N)\}$, where $n = pw$.

Since c is dependent on m, it is necessary to solve for the minimum mean squared error, derived from (2), by minimizing over both c and integer values of $m \in \mathbb{Z}$

$$\hat{m} = \arg \min_{m \in \mathbb{Z}} \left\{ \min_{c} \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i} - n_i^{\alpha(m)} c \mathbf{1}\|^2 \right\} \tag{3}$$

where $n_i = p_i w$ and $\mathbf{1}$ is the vector of length n_i whose elements are all one. We solve over integer values of m, as we do not consider fractal dimensions for this algorithm. This improves accuracy by constraining the estimation space to discrete values rather than discretizing estimates in a continuous space. One can solve (3) in the following general manner.

```
for  m = 2 to d do
```

1: Calculate $\hat{c}(m)$ from the expansion of (3)

a) $\hat{c} = \min_c \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i}\|^2 - 2c \sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} + c^2 \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T \mathbf{1}$

$\Rightarrow \hat{c} = \sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} / \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T \mathbf{1}$

2: Calculate the error $\boldsymbol{\epsilon}(m)$ with m and $\hat{c}$ from step 1)

$\boldsymbol{\epsilon}(m) = \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i} - \hat{c} n_i^{\alpha(m)} \mathbf{1}\|^2$

```
end.
```

$\hat{m} = \arg\min_i \boldsymbol{\epsilon}(i)$

This nonlinear least squares solution yields the dimension estimate $\hat{m}$ based on the k-NN graphs.

B. The Maximum Likelihood Estimator for Intrinsic Dimension

The MLE method [5] for dimension estimation estimates the intrinsic dimension $\hat{m}$ from a collection of i.i.d. observations $\boldsymbol{X} = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$, $m \leq d$. Similar to the k-NN algorithm for dimension estimation, the MLE method assumes that

close neighbors lie on the same manifold. The estimator proceeds as follows, letting k be a fixed number of nearest neighbors to sample x_i

$$\hat{m}_k(x_i) = \left[\frac{1}{k-2} \sum_{j=1}^{k-1} \log \frac{T_k(x_i)}{T_j(x_i)} \right]^{-1} \tag{4}$$

where $T_k(x_i)$ is the distance from point x_i to its kth nearest neighbor in $\boldsymbol{X}$. The intrinsic dimension for the data set can then be estimated as the average over all observations

$$\hat{m}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{m}_k(x_i).$$

Algorithm 1 Local dimension estimation

Input: Data set $\boldsymbol{X} = \{x_1, \ldots, x_n\}$
1: **for** $i = 1$ to n **do**
2: Initialize cluster $\mathcal{C} = x_i$
3: **for** $k = 1$ to n'
4: Find the kth NN, $x_{k,i}$, of x_i
5: $\mathcal{C} \leftarrow \mathcal{C} \cup x_{k,i}$
6: **end for**
7: $\hat{m}(x_i) = \text{dimension}(\mathcal{C})$
8: **end for**

Output: Local dimension estimates $\hat{m}(x_i)$ for $i = 1, \ldots, n$

C. Local Dimension Estimation

While the MLE method inherently generates local dimension estimates for each sample $\hat{m}(x_i)$, the k-NN algorithm in itself is a global dimension estimator. We are able to adopt it (and any other dimension estimation algorithm) as a local dimension estimator by running the algorithm over a smaller neighborhood about each sample point. Define a set of n samples $\boldsymbol{X} = \{x_1, \ldots, x_n\}$ from the collection of manifolds $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$ such that each point x_i lies on manifold $\mathcal{M}_j$. Any small sphere or data cluster of samples $\mathcal{C} \subseteq \boldsymbol{X}$ centered at point x_i, with $|\mathcal{C}| = n' \leq n$, will contain samples from $M' \leq M$ distinct manifolds. As $n' \to 1$, all of the points in $\mathcal{C}$ will lie on a single manifold (i.e., $M' \to 1$). Intuitively speaking, as the cluster about point x_i is reduced in size, the local neighborhood defined by said cluster can be viewed as its own data set confined to a single manifold. Hence, we can use a global dimension estimation algorithm on a local subset of the data to estimate the local intrinsic dimension of each sample point. This can be performed as described in Algorithm 1, where "dimension($\mathcal{C}$)" refers to applying any method of dimension estimation to the data cluster $\mathcal{C}$.

One of the keys to local dimension estimation is defining a value of n'. There must be a significant number of samples in order to obtain a proper estimate, but it is also important to keep a small sample size as to (ideally) only include samples that lie on the same manifold. Currently, we arbitrarily choose n' based on the size of the data set. However, a more definitive method of choosing n' is grounds for future work.

We briefly note that our definition of "local" dimension estimation differs from that of the Fukunaga–Olsen algorithm [1]. Specifically, we aim to find a dimension estimate for each sample point, which accounts for sets consisting of multiple manifolds, while [1] used local subsets of the data to form a global estimate of dimension.

III. Neighborhood Smoothing

For the problem of local dimension estimation, results are often highly variable, where nearby samples from the same manifold may result in different dimension estimates. This issue can be a result of a variety of reasons, such as variability due to random subsampling in the k-NN algorithm, or variability due to the neighborhood size in the MLE method. When constructing a global dimension estimate, this variance is relatively insignificant, as the estimate is constructed as a function of the local estimates. For local dimension estimation, however, this variance is of significant concern, and we propose a variance reduction method known as neighborhood smoothing [13], which improves estimation accuracy.

An initial intuition for manifold learning algorithms is that samples that are "close" tend to lie on the same manifold, which extends to the assumption that they therefore have the same dimension. With this assumption in place, it follows that filtering by majority vote over the dimension estimates of nearby samples should smooth the estimator and reduce variance. This voting strategy is similar to the methods of mode filtering, bagging [22] and learning by rule ensembles [23]. Smoothing simply looks at the distribution of dimension estimates within each sample point's local neighborhood and reassigns each sample a dimension estimate equal to that with the highest probability within its neighborhood. Specifically

$$\hat{m} = \arg\max_l \mathrm{P}_{\mathcal{N}_i}[\hat{m} = l] \tag{5}$$

where $\mathrm{P}_{\mathcal{N}_i}$ is the probability over the neighborhood of the current sample $\mathcal{N}_i$. Given a finite number of samples $\{x_1, \ldots, x_n\}$, this may be empirically evaluated as

$$\hat{m}(x_i) = \arg\max_l \sum_{x_j \in \mathcal{N}_i} I(\hat{m}(x_j) = l) \tag{6}$$

where $I(\cdot)$ is the standard indicator function. This process may then be iterated until the set converges such that each estimate remains constant. This has the effect of implicitly incorporating the neighbors of each sample's neighbors to some extent, as the dimension estimates within a local region may change through iterations.

Intuitively, neighborhood smoothing is similar to iteratively imposing a k-NN classifier on the local dimension estimates—under the guise that at each iteration, sample x_i is a test sample and all points x_j, $j \neq i$ are appropriately labeled training samples. Similarly to k-NN classification, the key factor to smoothing is defining the neighborhood $\mathcal{N}_i$. If $\mathcal{N}_i$ is too large, oversmoothing will occur. The variance of the dimension estimates will drastically decrease, but there will be a strong bias which will remove the detection of coarsely sampled manifolds. As such, one cannot use a constant region about a point but must adapt that region to the statistics of the sample.

A. Adaptive Neighborhood Selection

Since the number of sample points on each manifold of a data set is generally unknown, using a constant number of smoothing samples is not a viable option; samples on a smaller manifold may have points from a disjoint manifold included in their smoothing neighborhood. One straightforward method for neighborhood selection is to define neighbors by some spherical region or ϵ-ball about each sample point. This is generally acceptable when the disjoint manifolds are easily separable, as the neighborhood does not adapt to the geometry of the manifold. When distinct manifolds lie near one another, or potentially intersect, it is necessary to further adapt the smoothing neighborhood beyond a spherical region. This is due to the fact that points on a nearby or intersecting manifold may be as close (or closer) to a sample as others on its own manifold. A spherical region may smooth over different manifolds, and the results will lead to the dimension estimates' "leaking" from one manifold to another.

Rather than defining neighborhoods through Euclidean distance, which will form only spherical regions about each sample point, we will define neighborhoods using a geodesic distance metric. This will adapt the neighborhood to the geometry of the manifold. The geodesic distance is defined as the shortest path between two points along the manifold and may be approximated with graph-based methods. For our purposes, this metric can be determined by taking each point and creating an edge to its k-NN. Then, using Dijkstra's shortest path algorithm (or any other algorithm for computing the shortest path), approximate the geodesic distances between each pair of points in the graph. Any points that remain unconnected are considered to have an infinite geodesic distance between them.

To define a local neighborhood, we can now simply choose the closest n_g points for which the geodesic distance is not infinite. This forms a nonspherical neighborhood that adapts to the curvature of the manifold, performing much better than spherical neighborhoods. Fig. 1 illustrates the difference in the neighborhoods (black stars) that are formed on the "swiss roll" manifold when using different proximity metrics. The Euclidean distance [Fig. 1(a)] forms a spherical neighborhood, including points that are separated from the sample in question (red diamond). The geodesic distance [Fig. 1(b)], however, forms a neighborhood considering points only in close proximity along the actual manifold. While all points in this example do exist on the same manifold, it is clear that defining neighborhoods along the manifold rather than in simple spherical regions reduces the probability of including samples from a nearby distinct manifold.

Illustrating the effects of neighborhood smoothing, we create a seven-dimensional data set that includes two distinct hyperspheres of intrinsic dimensions two and five, each containing 300 uniformly sampled points intersecting in three common dimensions. Fig. 2(a) shows the histogram of the local dimension estimates of each sample before any neighborhood smoothing was applied, while Fig. 2(b) shows the results after smoothing. One can clearly see that the wide histogram was correctly condensed to the proper local dimension estimates, even though the manifolds intersect. The use of the geodesic distance measure

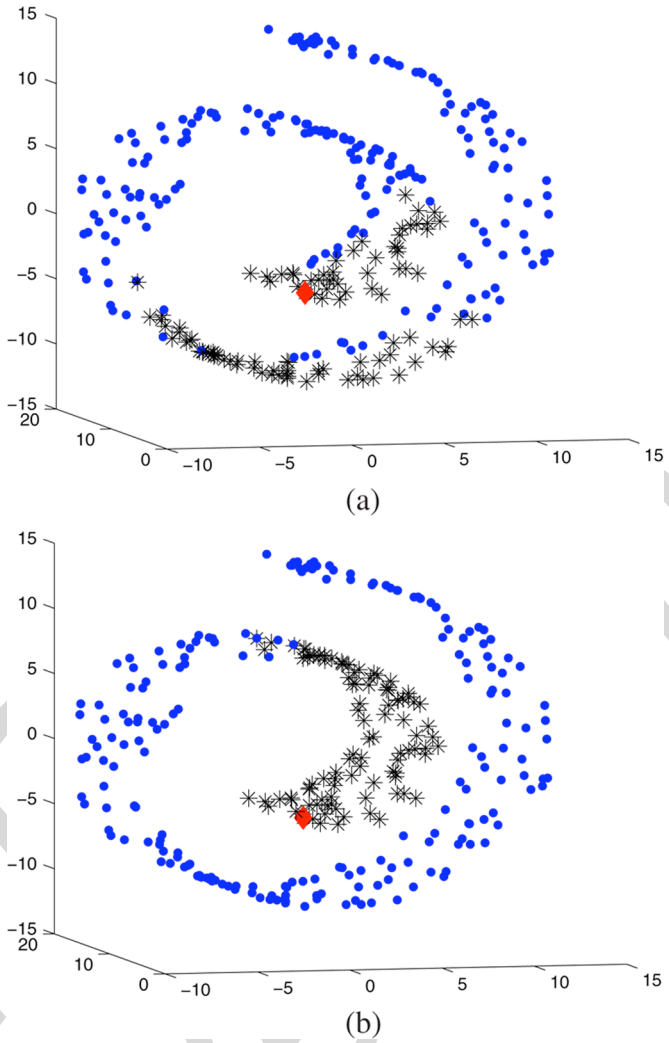


Fig. 1. Neighborhoods ($*$) of the sample in question ($\diamond$) defined by (a) Euclidean distance and (b) geodesic distance. (a) Spherical neighborhood and (b) adaptive neighborhood.

prevents smoothing across distinct manifolds, which lie closely together in Euclidean space.

It is important to note that, as with any form of postprocessing, neighborhood smoothing can only produce accurate results given sufficient input. The benefits of smoothing can be significantly diminished if the initial local dimension estimates are not sufficiently accurate. We note this explicitly because of the known issues with estimating large dimensions (e.g., $m > 20$). Because of variance issues due to insufficient samples and boundary effects, it is difficult to accurately estimate very large dimensions, and often the estimate can more appropriately be considered a measure of *complexity*, where the difference between m and $m+1$ is rather insignificant. This is important because no single dimension may dominate a given local neighborhood, yet smoothing will still assign a dimension estimate equal to the most represented dimension, which may indeed be inconsistent with the rest. We demonstrate this scenario with the example shown in Fig. 3, where smoothing would assign a dimension estimate of $m = 40$, which is the most represented dimension in the neighborhood. However, a more accurate dimension estimate could be considered $m = 33$

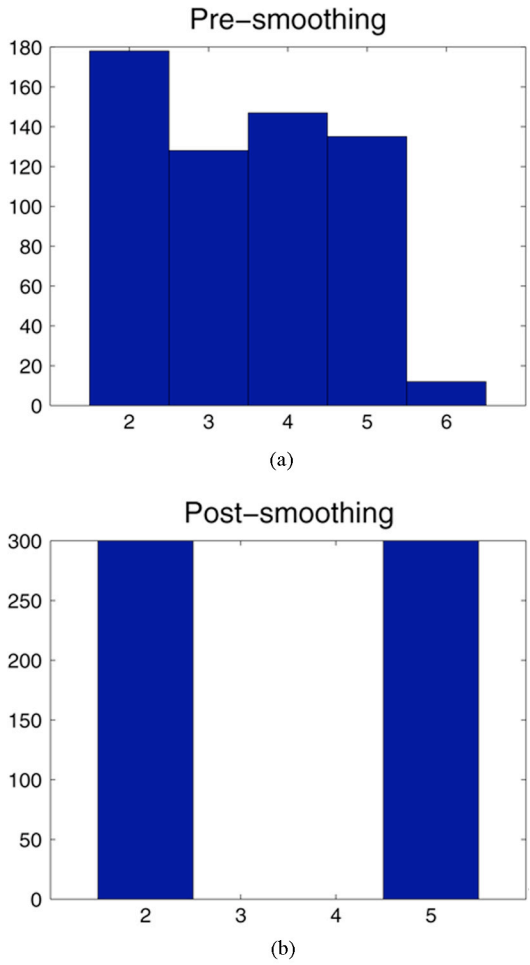


Fig. 2. Neighborhood smoothing applied to seven-dimensional data containing two spheres with intrinsic dimensions two and five.

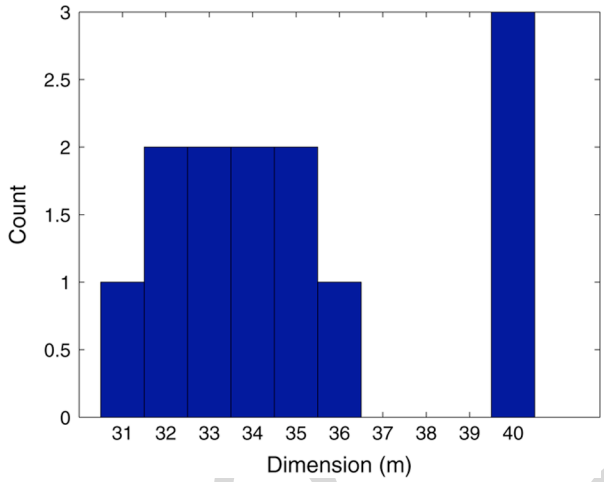


Fig. 3. Issues arise with neighborhood smoothing when estimating very large dimensions due to the variance of such estimates. In this example, smoothing would assign a dimension estimate of 40, although the more appropriate estimate would be 33 or 34.

or $m = 34$, as that would be more consistent with the majority of the samples. In these scenarios, it may be more appropriate to smooth over a histogram with user-defined bin sizes, corresponding to significant differences in complexity rather than individual dimensions. This is an area for future work.

IV. Applications

A. Debiasing Global Dimension Estimation

To our knowledge, a phenomenon common to all algorithms of intrinsic dimension estimation is a negative bias in the dimension estimate. It is believed that this is an effect of undersampling the high-dimensional manifold. While the bias due to lack of sufficient samples is inherent, we offer that the sample size is not the only source of bias; a significant portion is related to the depth of the data. Specifically, as data samples approach the boundaries of the manifold, they exhibit a lower intrinsic dimension. This issue becomes more prevalent as the dimension of the manifold increases and is directly related to the *curse of dimensionality*. Note that even manifolds that appear "empty" in their extrinsic-dimensional space (e.g., the Swiss roll) are filled and contain boundaries in the space of their intrinsic dimension.

Previous work [24] has demonstrated that as dimensionality increases, the nearest neighbor distances approach those of the most distant points; this will clearly have an adverse effect on neighborhood-based estimation algorithms. We are able to further correlate this effect on dimension estimation by calculating the depth of each sample and quantitatively analyzing the relationship between depth and dimension. We utilize the L_1-data depth algorithm developed in [25], which calculates depth $D_n(x)$ as the sum of all the unit vectors between the sample of interest $x \in \boldsymbol{X}$ and the rest of the data set $\boldsymbol{X} = \{x_1, \ldots, x_n\}$. Specifically

$$D_n(x) = 1 - \max\left(0, \left\| \sum_{x_i \neq x} e(x_i - x)/n \right\| - \sum_{x_i = x} \frac{1}{n}\right) \tag{7}$$

where $e(x_i - x) = (x_i - x)/\|x_i - x\|$ is the unit vector in the direction of $(x_i - x)$. This depth measure assigns the most interior points in the data set a depth value approaching one, while samples along the boundaries approach a depth of zero.

Using this measure, we illustrate the effect of data depth on dimension estimation in Fig. 4. The data set used was of 3000 points uniformly sampled on a six-dimensional hypercube. We utilize the MLE method for dimension estimation, and Fig. 4 illustrates the distribution of data depths for samples that estimate at different dimensions. It is clear that as the depth increased, so did the probability of estimating at a higher dimension, even to the point where the most deep points estimated at a dimension of seven (although we note that there were very few points with this estimate).

When estimating the global dimension of a data set, one can substantially reduce the negative bias by placing more emphasis on the local dimension of those points away from the boundaries, as they are more indicative of the true dimension of the manifold. Specifically, let the global dimension be estimated as follows:

$$\hat{m} = \frac{1}{\sum_j W_j} \sum_i W_i \hat{m}(x_i) \tag{8}$$

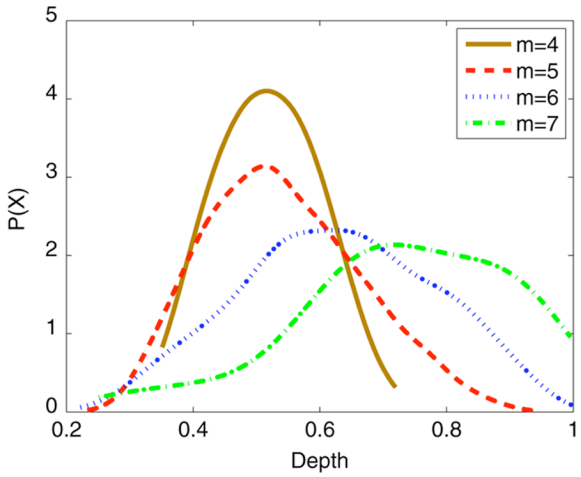


Fig. 4. PDFs of data depth based on estimated intrinsic dimension. Points with less depth estimate at a lower dimension, contributing to the overall negative bias.

where W_i is a weighting on each sample point. We offer two potential definitions of W_i, the first being a binary weighting

$$W_i = \begin{cases} 1, & D_n(x_i) \geq D_n(x_{(\beta \times n)}) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $0 \leq \beta \leq 1$ and $D_n(x_{(\beta \times n)})$ is the data depth of the $\beta \times n$ deepest point. Essentially this binary weight amounts to debiasing by averaging over the local dimension estimates of the deepest $\beta \times 100\%$ of points, where the threshold β is user defined. This is worthwhile for potentially large data sets, where there are enough samples to ignore a large portion of them. When this is not the case, let us make the definition

$$W_i = \mathrm{e}^{-(1-D_n(x_i))/c} \tag{10}$$

where c is a user-defined constant. This weighting may be viewed as a heat kernel, in which larger depths will yield higher weights. Unlike the binary weighting, which will ignore a large number of the data samples, this heat kernel weighting will utilize all samples (even those lying on a boundary) yet give preference to those with more depth in the manifold.

We now illustrate this debiasing ability in Fig. 5, in which we estimated the global dimension of the six-dimensional hypercube (3000 i.i.d. samples) over 200 unique trials. Fig. 5(a) shows the histogram of biased dimension estimates obtained by using the entire set for dimension estimation, while Fig. 5(b) estimates the correct global dimension each trial by using our debiasing method (8) with the binary weighting function (9) using $\beta = 0.5$.

To study the number of samples necessary to accurately estimate global dimension, we plot estimation results in Fig. 6. In this simulation, we plot the mean de-biased ($\beta = 0.5$) and unrounded dimension estimated over a 20-fold cross validation, based on differing number of samples on the six-dimensional hypercube. We can see that if rounded to the nearest integer, the debiased estimate will be correct on average with roughly 2500 samples. On the contrary, without debiasing, the estimation maintains a much stronger negative bias, never correctly estimating the dimension when rounded.

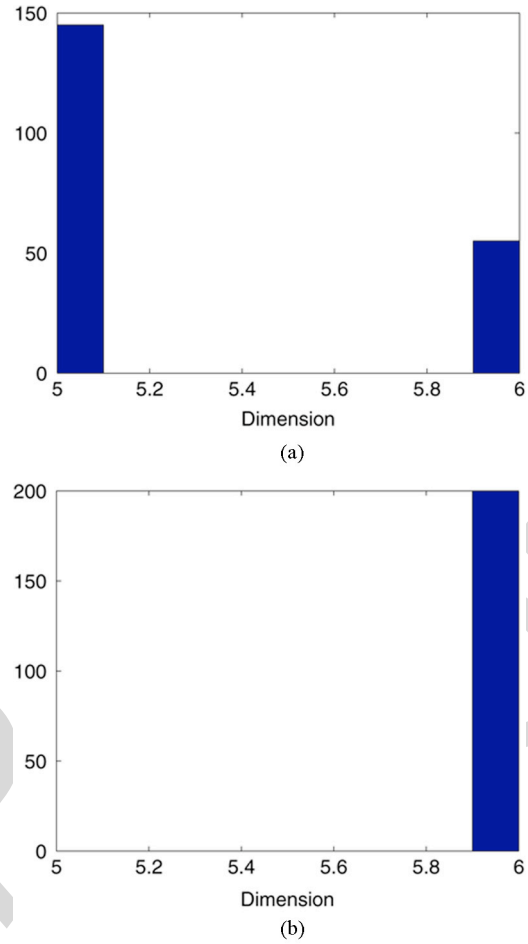


Fig. 5. Developing a debiased global dimension estimate by averaging over the 50% of points with the greatest depth on the manifold. (a) Biased results and (b) debiased results.

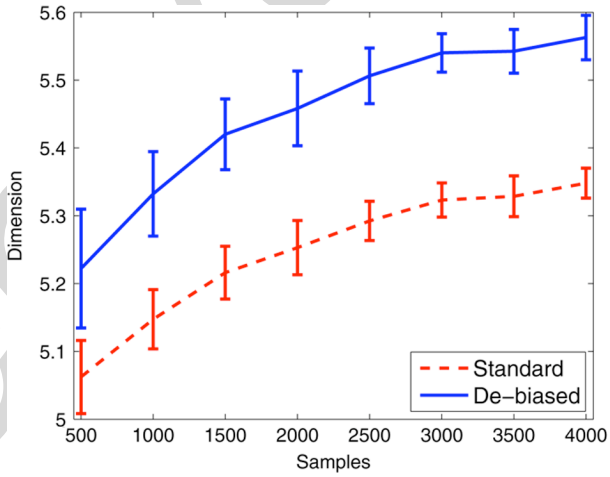


Fig. 6. Analysis of how many samples are necessary to appropriately estimate debiased global dimension. Plot shows mean dimension estimated over a 20-fold cv, with error bars at one standard deviation.

It is important to note that our method of debiasing is only applicable for data with a relatively low intrinsic dimension. When dealing with very high dimensional data, the probability of a sample lying near a boundary approaches one, and the value of the depth approximation becomes irrelevant. This is shown in

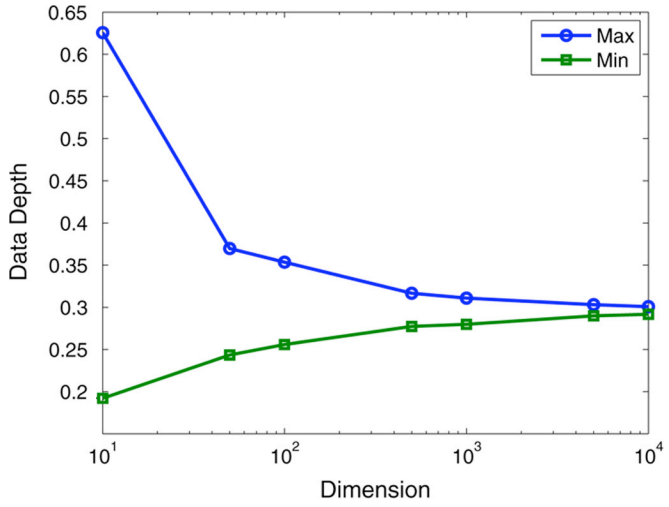


Fig. 7. As the intrinsic dimension increases, the maximum and minimum data depth of points in the set converge to the same value. This simulation was over a fivefold cross-validation with 400 uniformly sampled points on the unit cube.

Fig. 7 where the "deepest" and most "shallow" samples converge to the same depth value as the intrinsic dimension increases.

Prior work on estimating dimension through vector quantization [20] has reported robustness to negative bias. While not offering a distinct claim or proof of this robustness, the authors mention their algorithm obtains larger estimates for high-dimensional data than neighborhood-based methods. Theoretically, this method will suffer from similar bias issues due to the intrinsic geometry of the data, which is not explicitly accounted for in [20]. The improved performance reported is likely due to the cross-validation implemented. That said, the use of quantization error may indeed be *more* robust to negative bias than neighborhood-based methods, and this potential gain is worth further investigation.

B. Statistical Manifold Learning

Of particular interest in manifold learning is the intrinsic dimension to which one can reduce the dimensionality of a data set with minimal loss of information. This is typically presented for data that lie on a Riemannian submanifold of Euclidean space. We extend this application to the problem of learning on statistical manifolds [26], in which each point on the manifold is a pdf. Rather than defining distance with Euclidean metrics, we approximate the Fisher information distance—with the Kullback–Leibler divergence and Hellinger distance—which is the natural metric on statistical manifolds [27]. We illustrate the use of local dimension estimation in these learning tasks for the applications of flow cytometry analysis and document classification.

1) Flow Cytometry Analysis: In clinical flow cytometry, pathologists gather readings of fluorescent markers and light scatter off individual blood cells from a patient sample, leading to a characteristic multidimensional distribution that, depending on the panel of markers selected, may be distinct for a specific disease entity. The data from clinical flow cytometry can be considered multidimensional both from the standpoint of multiple characteristics measured for each cell, and from the standpoint of thousands of cells analyzed per sample. In previous work [28], [29], we have shown the ability to derive an information embedding of the statistical manifold defined by the space of pdfs, in which each patient's blood sample can be considered a realization of a pdf on said manifold. We developed Fisher information nonparametric embedding (FINE) as an informationgeometric method of dimensionality reduction based on Fisher information distances between pdfs [26]. Using FINE, we are able to embed the pdf realizations into a low-dimensional Euclidean space, in which each patient is represented by a single low-dimensional vector. In order to determine the dimension m for this embedding space, we first apply local dimension estimation to find the desired dimension of our embedding space.

Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ be a collection of data sets where $\boldsymbol{X}_i$ corresponds to the flow cytometer output of the ith patient. For our analysis, each patient has either chronic lymphocytic leukemia or mantle cell lymphoma, which display similar characteristics with respect to many expressed surface antigens but are generally distinct in their patterns of expression of two common B lymphocyte antigens CD23 and FMC7. We are interested in the intrinsic dimension of the statistical manifold, realized by $\mathcal{X}$, as that is what we plan to embed. We define the dissimilarity matrix D, where $D(i, j)$ is the symmetric Kullback–Leibler divergence approximation of the Fisher information distance between pdf estimates on $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ [30]. For this simulation, we estimated the pdfs with kernel density estimation methods, although any nonparametric method will suffice. By redefining our local dimension estimation algorithms to take the high-dimensional distance matrix—in the space of pdfs—as an input (which is not an issue, as both the k-NN and MLE methods are entirely based on nearest neighbor distances), we are able to estimate the intrinsic dimension of the statistical manifold. The local dimension estimation results are shown in Fig. 8, where we can see the intrinsic dimension is $m = [2, 3]$. This result can be interpreted as recognizing the two specific markers that most significantly differentiate between classes (i.e., $m = 2$) but also accounting for the fact that there still exist subtle differences between members of the same class, and some patients may not exhibit the expected response to specific antigens as strongly as others (i.e., $m = 3$).

After estimating the intrinsic dimension of the data set, we are able to embed each patient into an m-dimensional Euclidean space, as observed in Fig. 9. In this embedding, each point represents a single patient data set, which was originally six-dimensional with samples on the order of $n \sim 5000$. We can see that a two-dimensional unsupervised embedding gives a clear class separation, which enables effective clustering and classification of the data. This result is consistent with our dimension estimate of $m = [2, 3]$ and illustrates the effectiveness of local dimension estimation for learning on statistical manifolds.

2) Document Classification: Given a collection of documents of known class, we wish to best classify a document of unknown class. A common representation of a document is known as the *term frequency* representation. This is essentially a normalized histogram of word counts within the document. Specifically, let x_i be the number of times term i appears in a specific document. The PdF of that document can then be

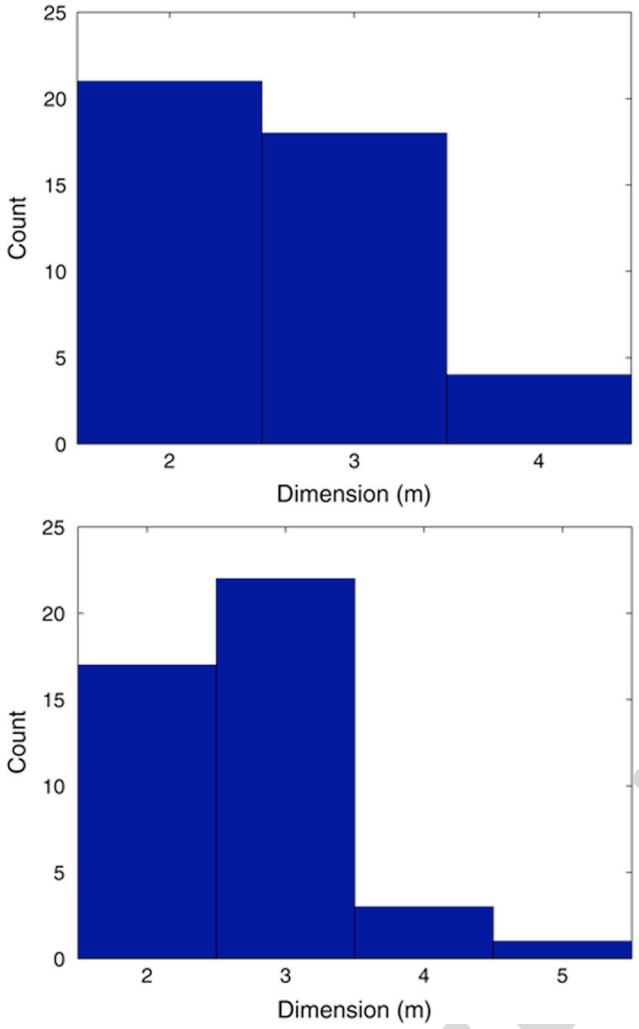


Fig. 8. Histogram of local dimension estimates for the statistical manifold defined by flow cytometry results of 43 patients with chronic lymphocytic leukemia or mantle cell lymphoma. (a) Local k-NN dimension estimates and (b) local MLE dimension estimates.

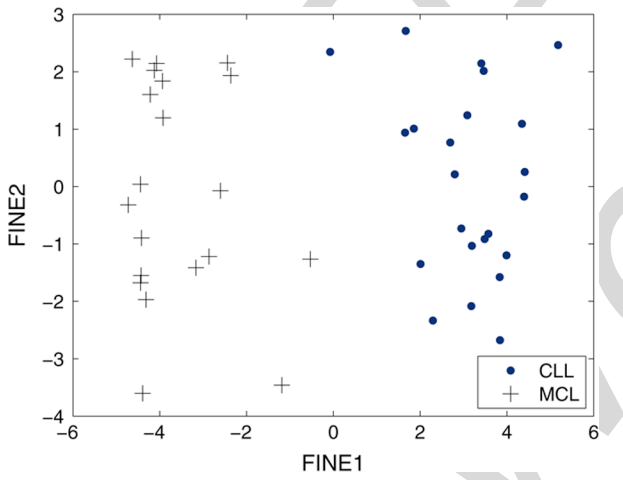


Fig. 9. The information-based embedding, determined by FINE, of the flow cytometry data set. Embedding into $m = 2$ dimensions yields linear separability between classes.

characterized as the multinomial distribution of normalized word counts, with the maximum likelihood estimate provided as

$$\hat{p}(x) = \left(\frac{x_1}{\sum_i x_i}, \ldots, \frac{x_N}{\sum_i x_i} \right). \tag{11}$$

For our illustration, we utilized the 20 Newsgroups[1] data set, which has an extrinsic dimension of $d = 26\,214$, which is the number of terms in its dictionary. This set contains postings from 20 separate newsgroups, and we wish to classify them by their highest domain (one of [comp.*, rec.*, sci.*, talk.*]). To perform this classification task, we first wish to alleviate the *curse of dimensionality* by reducing the data to a lower dimensional manifold. For this task we utilize FINE, approximating the Fisher information distance with the Hellinger distance, such that

$$D(i,j) = \sqrt{\sum_{l=1}^{N} (\sqrt{p_i(x_l)} - \sqrt{p_j(x_l)})^2}$$

where $p_i(x)$ is the estimate (11) of the pdf of document i.

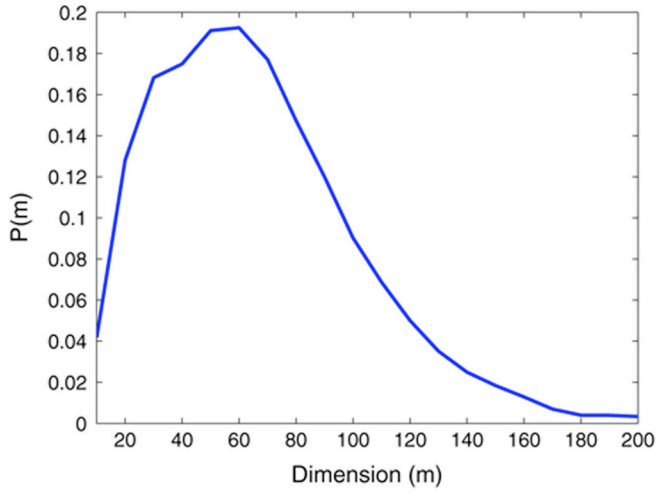
Experimental results have shown there are multiple submanifolds of differing dimension in the data set. In Fig. 10, we present the distribution of dimension estimates and compare that to classification performance at reduced dimension. Specifically, we used the MLE method with the matrix of Hellinger distances (between full-dimensional pdfs) to estimate the local dimension of each sample, then used FINE to embed a random subsampling of 1000 points of the data into a lower dimension. The distribution of these local dimension estimates over a 20-fold cross-validation is shown in Fig. 10(a). Next, we separated the embedded data into a training set of 800 samples and a test set of 200 samples. Results of the linear, "all vs. all" classification task (i.e., classify each test sample as one of 4 different potential classes) are shown in Fig. 10(b) as a function of the embedding dimension (over the same 20-fold cross-validation).

We observe that the apex of the classification rate curve ($m \in [20, 50]$) corresponds to the apex of the pdf curve of local dimension estimates ($m \in [30, 70]$), which illustrates that the local dimension estimation method was able to find an appropriate embedding dimension. Although the range $m \in [30, 70]$ seems to be large, it is important to remember the extrinsic dimension of the data is $d = 26\,214$, so we are able to adequately define the dimension of the manifold. We note that for this simulation, we did not utilize neighborhood smoothing due to the high-dimensional nature of the data, as previously explained. A pdf of the local dimension estimates is more beneficial towards analysis than arbitrarily setting a dimension that does not dominate the neighborhood.

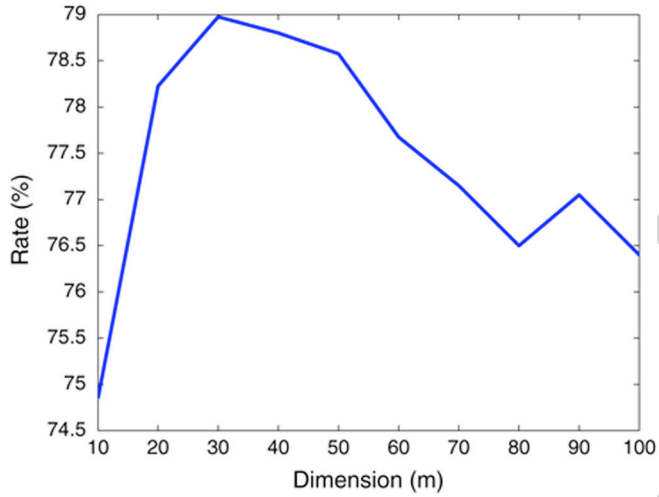
C. Network Anomaly Detection

Anomalies can be detected in router networks through the use of local dimension estimation [6]. Specifically, when only a few of the routers contribute disproportionately large amounts of traffic, there is a decrease in the intrinsic dimension of the entire network; that is, the space of traffic counts per router. Using

[1]http://people.csail.mit.edu/jrennie/20 Newsgroups/.



(a)



(b)

Fig. 10. Comparison of (a) pdf of local dimension estimates and (b) classification rate versus embedding dimension for the 20 Newsgroups data set. The optimal embedding dimension ranges from 20 to 50, which is in the same range as the apex of the local dimension estimation pdf.

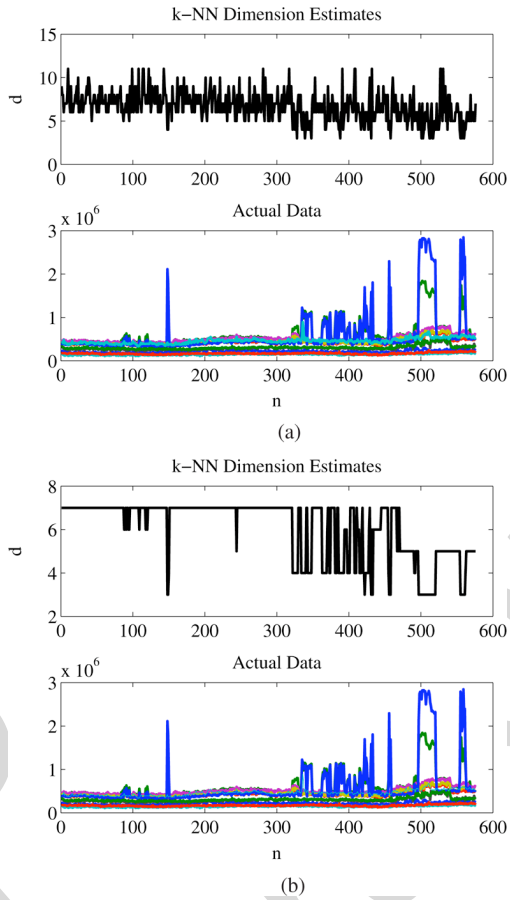


Fig. 11. Neighborhood smoothing applied to Abilene Network traffic data dimension estimation results. The x-axis represents time series changes in the network. (a) Before smoothing and (b) after smoothing.

neighborhood smoothing as a form of postprocessing, we are better able to locate the traffic anomalies, as the variance of the estimates is reduced. Fig. 11 illustrates the usage of neighborhood smoothing on the results of k-NN algorithm for local dimension estimation for anomaly detection. The data used are the number of packets counted on each of the 11 routers on the Abilene network, on January 1–2, 2005. Each sample is taken every 5 min, leading to 576 samples with an extrinsic dimension of $d = 11$.

Fig. 11(b) illustrates that neighborhood smoothing is able to preserve both the visually obvious ($n = 148$, $n > 300$) and nonobvious ($n = 87 - 120$) changes in network complexity. A detailed investigation of time $n = 244$, for example, reveals that the Sunnyvale router (SNVA) showed increased contribution from a single IP address. Large percentages (over half) of the overall packets had both source and destination IP 128.223.216.0/24 within port 119. This port showed increased activity on the Atlanta router as well. This change in dimensionality indicating anomalous activity would generally go unnoticed with the raw results of local dimension estimation due to the high variance [Fig. 11(a)].

We note the results shown in Fig. 11 are performed using nominal settings within the k-NN algorithm, which allows the algorithm to run quickly and accurately with neighborhood smoothing. We are able to generate results with much less variance than Fig. 11(a) by applying more averaging and bootstrapping, but this significantly increases computation time while still producing results with more variance than Fig. 11(b).

D. Clustering

As discussed previously, data sets often consist of multiple submanifolds of differing dimension. When the intrinsic dimension of these submanifolds becomes increasingly large, the value of the dimension may be interpreted as a measure of the *complexity* of the data. From this interpretation, we may use local dimension estimation to cluster data within a set by complexity. Specifically, we can define clusters through the use of recursive entropy estimation and neighborhood smoothing. As we increase the neighborhood size k, we incorporate more samples into our smoothing region, eventually oversmoothing between differing manifolds. By finding the point in which the smoothing regions extend into multiple manifolds, we can define clusters in the data. This point of change can be located

by analyzing the change in the entropy H of the dimension estimates as the region grows, such that

$$H = -\sum_j P_j \log P_j$$

where $P_j = (1)/(n)\sum_i^n I(\hat{m}(i) = j)$ is the empirical probability a sample estimates at dimension j.

When the regions are stable within each cluster, H will be constant. As the smoothing neighborhood incorporates additional manifolds, the entropy will leave its constant state and eventually $H \to 0$ as $k \to \infty$ (i.e., the region includes every point). With a priori knowledge of the distribution of dimensionality, one may choose a neighborhood size that yields an appropriate value of entropy. Without this knowledge, the point at which H leaves its constant state can be used as a threshold for defining clusters based on dimension. This process is similar to dual-rooted diffusion kernels method of clustering [31], in which the authors used the jump in nearest neighbor distance as a means to differentiate clusters.

For example, let $\boldsymbol{X} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$ is uniformly distributed in [0,1]^{m_i} ($m_i \in M$, a discrete set of integer values) and constant elsewhere. Hence, m_i is the intrinsic dimension of x_i. For our simulation, let $d = 13$ and $M = [2, 6, 10]$, and there are $n = 200$ samples for each value in M. After obtaining local dimension estimates, we apply neighborhood smoothing to differing neighborhood sizes and measure the entropy of the local dimension estimates at each size. The results are shown in Fig. 12, where the entropy exhibits the same pattern we previously described; after initially decreasing, H remains constant as k approaches the region size of each manifold ($n = 200$). As the smoothing covers multiple manifolds $k > 200$, the entropy decreases until the smoothing neighborhood eventually covers all manifolds simultaneously and $H = 0$. The histogram of local dimension estimates (with both k-NN and MLE methods), which is used to calculate the entropy, is shown in Fig. 13 to illustrate the evolution of the dimension estimates. It is clear that at $k = 100$, the three distinct clusters are represented, and this value also corresponds to the optimal entropy estimate given a priori knowledge that each dimension is represented with a constant probability of $P = (1)/(3)$, which yields the entropy value $H = 1.1$. Due to insufficient sampling, the actual value of the dimension estimates ([2,5,7] for the k-NN algorithm and [2,5,6] for the MLE method) differs from the true dimensions [2,6,10]. However, this is not of particular concern since the primary objective is to locate clusters of differing *complexity*. It is also worth noting that some samples are misidentified due to the overlapping nature of the three clusters, but the overall performance is respectable.

We note the dimension estimate obtained when smoothing over the entire set does not correspond to the global dimension of the data. Since we are using a majority voting method, the final value will be equal to the estimated dimension which is most represented (with simple tie-breaking rules). This is not necessarily equal to the global dimension, and is often not close to the dimension which best characterizes the entire data set (as in our example).

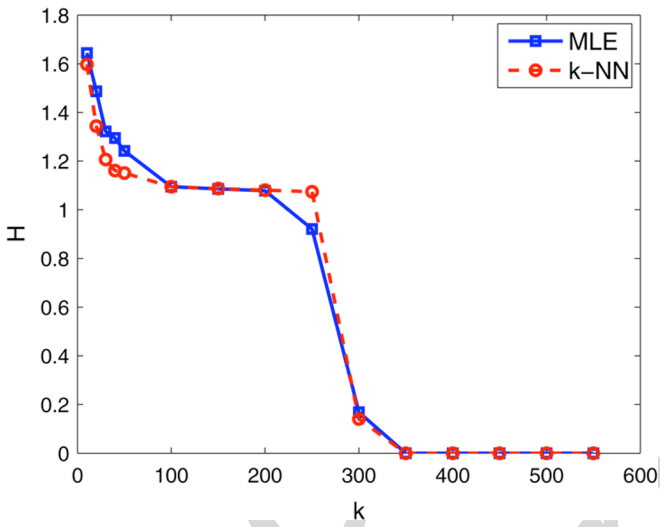


Fig. 12. The entropy of the local dimension estimates changes as a function of neighborhood size k.

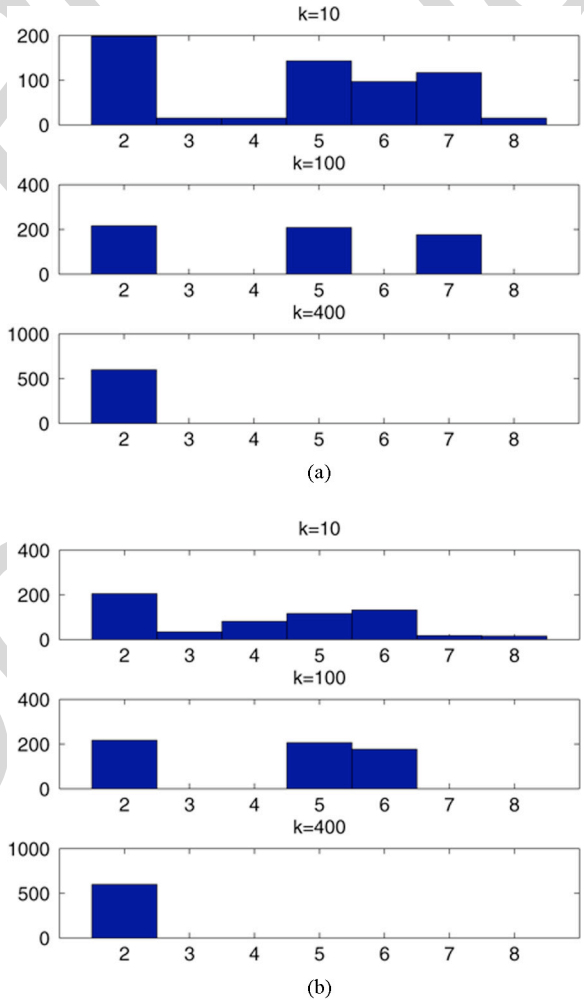


Fig. 13. Comparing dimension histograms of dimension estimates at various neighborhood sizes, we see that samples are clustered very well at $k = 100$, which corresponds to a constant point in the entropy plot shown in Fig. 12. (a) k-NN algorithm and (b) MLE method.

Let us now compare our clustering performance on a separate synthetic example. Consider the data set $\boldsymbol{X} = [x_1, \ldots, x_{400}]$ that consists of 200 points uniformly sampled on the "swiss roll" manifold and 200 points uniformly sampled on an intrinsically three-dimensional hypersphere. Hence, each $x_i \in \mathbb{R}^4$ (points sampled from the "swiss roll" have a constant value in the fourth dimension) and there are two distinct clusters formed. A visual representation of this set is illustrated in Fig. 14, and we compare our method of clustering by *complexity* using local dimension estimation with that of standard clustering methods—fuzzy c-means [32] and K-means [33]. To demonstrate clustering performance, we utilize the Jaccard index [34], which assesses the similarity between a predetermined set of class labels C and a clustering result K. Specifically

$$J(C,K) = \frac{a}{a+b+c}$$

where a is the number of pairs of points with the same class label in C and the same cluster label in K; b is the number of pairs that have the same label C but differ in K; and c is the number of pairs of points with the same cluster label in K but different class label in C. Essentially, the Jaccard index gives a rating in the range [0,1] in which "1" signifies complete agreement between the true labels C and the results K.

We show the results in Table I over a 20-fold cross-validation with i.i.d. realizations of $\boldsymbol{X}$. We see clustering by dimension estimation yields far superior performance to standard methods. While these methods aim to cluster by a variety of means, such as optimizing distances to centroids, dimension estimation simply assigns cluster labels based on the local dimensionality of each data point. In this simulation, we utilized a neighborhood size of $k = 25$ when smoothing, as larger values tended to incorporate both manifolds since they are so close to one another. We acknowledge that clustering by dimensionality is not applicable in many practical problems in which the different clusters exhibit the same dimensionality. However, in the realm of high-dimensional clustering, there may often exist an intrinsic difference in dimensionality, in which our method would be applicable.

1) Image Segmentation: After showing the ability to use local dimension estimation for clustering data by complexity, a natural extension is to apply the methods for the problem of image segmentation. Differing textures in images can be considered to have different levels of complexity (e.g., a periodic texture is less complex than a random one). This has been well stated in [12], where natural images and textures are viewed as a collection of fractals. For our purposes, we chose to ignore such model assumptions and see whether or not Euclidean dimension can be used towards image segmentation. The same framework as our clustering method applies.

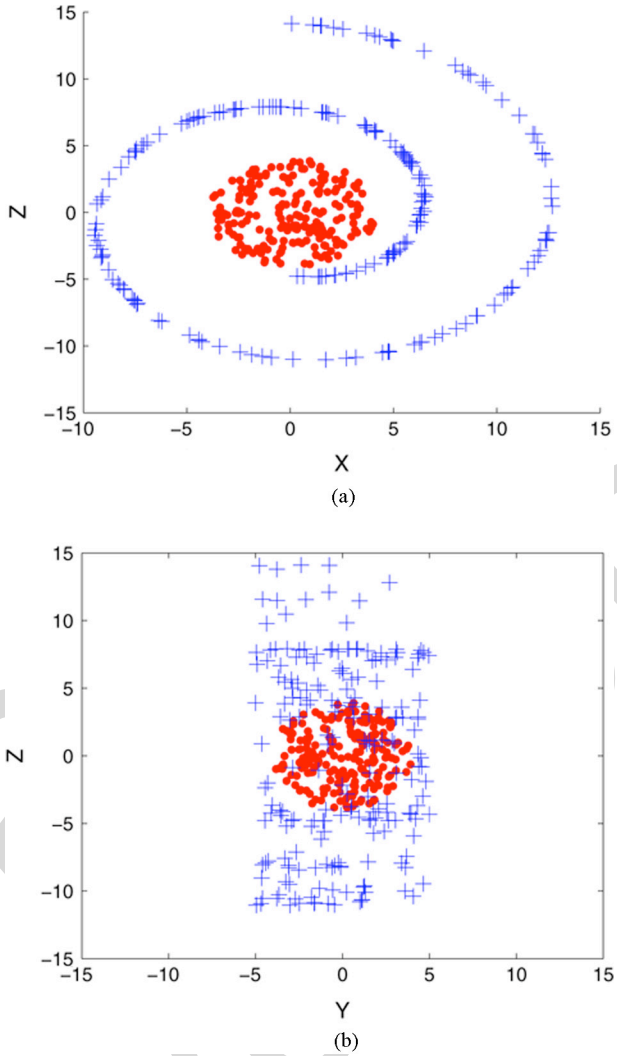


Fig. 14. Clustering based on local intrinsic dimensionality is useful for problems such as this, in which three-dimensional hypersphere (•) is placed "inside" the two-dimensional "swiss roll" (+). Side and front angles of set shown. (a) Side and (b) front.

TABLE I
COMPARISON OF VARIOUS CLUSTERING METHODS ON DATA SET CONSISTING OF "SWISS ROLL" AND THREE-DIMENSIONAL HYPERSPHERE MANIFOLDS. PERFORMANCE REPORTED BASED ON MEAN JACCARD INDEX OVER A 20-FOLD CROSS-VALIDATION

Method	Mean Jaccard
Dimension Estimation	0.7834
K-Means	0.4224
Fuzzy c-means	0.3607

Consider the satellite image of New York City[2] in Fig. 16(a), which has a resolution of 1452×1500. We wish to segment the image into land and water masses. To use local dimension estimation, we define $\boldsymbol{X} = \{x_1, \ldots, x_n\}$, where x_i is a 144-dimensional vector representing a rasterized 12×12 block of the image. After obtaining the local dimension estimates, we apply neighborhood smoothing and recursive entropy estimation as described above. The results, illustrated in Fig. 15(a), lead us to define an ideal neighborhood size of $k = 3500$, which is where

[2]http://newsdesk.si.edu/photos/sites_earth_from_space.htm.

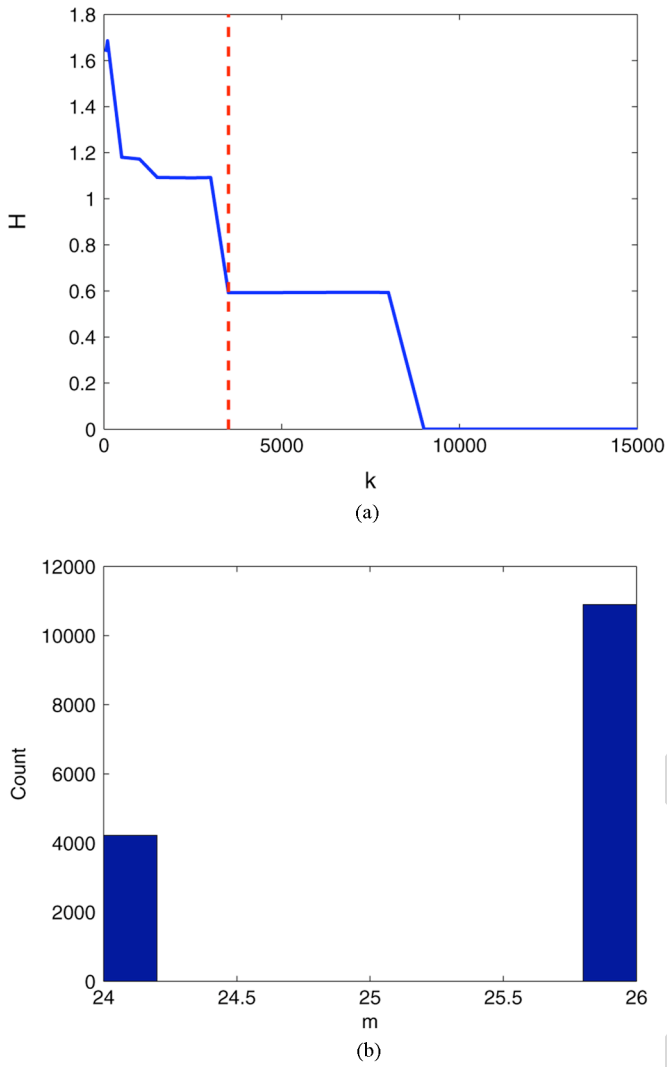


Fig. 15. Plotting the entropy of the dimension estimates suggests a neighborhood size of $k = 3500$, denoted by the dotted line, which yields two significant clusters in the dimension estimates. (a) Entropy versus k and (b) histogram of dimension estimates.

the entropy begins to remain constant for an extended period. This allows us to segment the image into two regions, defined by the complexity estimates shown in Fig. 15(b). The final segmentation can be viewed in Fig. 16(b), where the water is well separated from the land portions of the island of Manhattan and the surrounding boroughs. We note that this image is that of the smoothed local dimension estimates, uniformly scaled to the range [0,255].

We notice there is a relatively low resolution in our segmentation image, due to the large 12×12 blocks used for estimation. We can correct this by using a smaller pixel blocks; however, computational issues prevent us from estimating at much higher resolutions. We can alleviate this problem by estimating at a high resolution only in the areas that require such; this may be determined by using edge detection on the image of local dimension estimates as in Fig. 16(c). In the regions that are determined to contain edges, we resegment at a higher resolution—using 4×4 pixel blocks—with the same recursive entropy estimation process. The results are shown in Fig. 16(d); it is clear that this

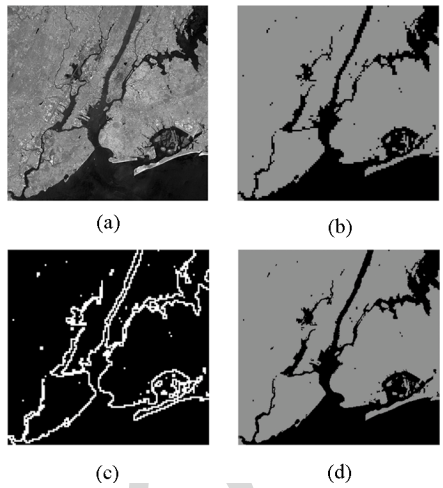


Fig. 16. By using local dimension estimation, neighborhood smoothing, and entropy estimation, we are able to segment the satellite image of New York City into water and land regions. After segmenting the image at a low resolution, we perform edge detection to find the regions that should be analyzed at a higher resolution, yielding a significantly more detailed segmentation. (a) New York City, (b) low-resolution segmentation, (c) edges of segmented image, and (d) high-resolution segmentation.

segmentation appears significantly less digitized and more detailed.

While the previous task was simply to segment water from land in an image, we detailed the "binary" task to demonstrate the process. The problem is easily extended to the multitexture case, which we illustrate in Fig. 17 with images of local dimension estimates scaled to the range [0,255]. In these cases, we segmented images of a sloth bear[3] and a panda bear cub[4] using the same techniques as previously described, only we utilized a high-resolution segmentation over the entire image along with small smoothing neighborhoods. This may give a finer segmentation than required (e.g., the bears are not segmented entirely as one object) but shows the potential segmentation power of local dimension estimation. If a coarser segmentation was desired, larger smoothing neighborhoods may be applied, similar to the previous case of New York City. We note that by no means are we suggesting that dimension alone is a superior means of image segmentation; we simply illustrate that there is a semblance of power to Euclidean dimension when segmenting natural images, and that dimension may be used in conjunction with other means for this complex task.

V. Conclusion

We have shown the ability to use local intrinsic dimension estimation for a myriad of applications. The negative bias in global dimension estimation is strongly influenced by the data depth of the samples on the manifold. By developing a global dimension estimator based on the local dimension estimates of the deepest points, we have shown the issue of the negative bias can be significantly reduced. Typically, dimension estimation is used for the purposes of dimensionality reduction of Riemannian manifolds in Euclidean space, and we have extended this to the

[3]http://newsdesk.si.edu/photos/nzp_sloth_bear.htm.

[4]http://newsdesk.si.edu/photos/nzp_panda_cub.htm.

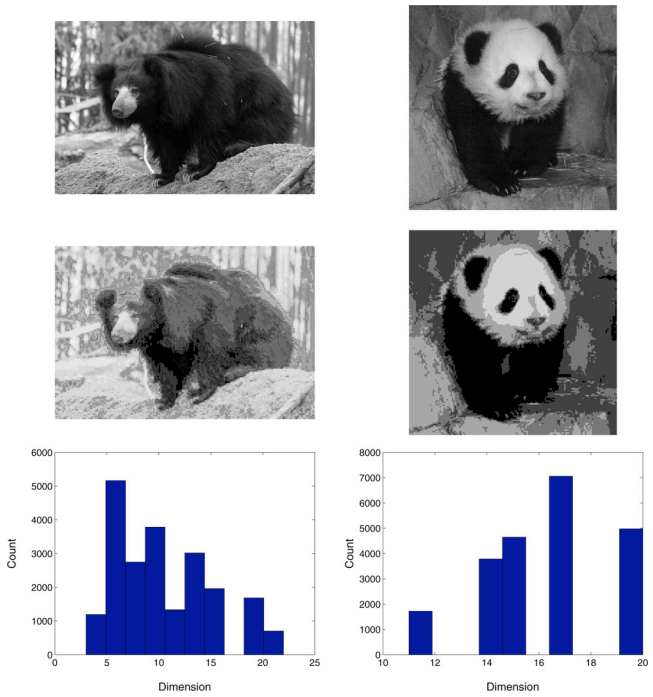


Fig. 17. Segmentation of multitexture images using local dimension estimation and neighborhood smoothing. The first row contains the original images, the second row contains the images of local dimension estimates (scaled to [0,255]), and the third row is the histogram of local dimension estimates.

problem of dimensionality reduction on statistical manifolds, illustrated with the examples of flow cytometry analysis and document classification.

By viewing dimension as a substitute for data complexity, we have applied local dimension estimation to problems that may not naturally be considered. Local dimension estimates can be used to find anomalous activity in router networks, as the overall complexity of the network is decreased when a few sources account for a disproportionate amount of traffic. We have also applied complexity estimation towards the problems of data clustering and image segmentation through the use of neighborhood smoothing. By finding the points in which entropy remains constant as the neighborhood size increases, we are able to optimally cluster the data.

Further analysis into the applications we have presented here is an area for future work. In terms of debiasing global dimension estimation, applying significant weight the interior points in averaging over local dimensions may result in large variance of the dimension estimate due to a small sample size. The bias–variance tradeoff and its optimization is of great importance and should be considered an area for future work. Additionally, we would like to further investigate using Euclidean dimension estimation (as opposed to fractal dimensions) for image segmentation, as we feel this is a very interesting application which has not been thoroughly researched. Specifically, we are interested in combining Euclidean dimension with other measures of textures in order to optimally segment a natural image.

ACKNOWLEDGMENT

The authors would like to thank B. Li from the University of Michigan for isolating the source of the anomalies we discovered in the Abilene data and Dr. W. G. Finn and the Department of Pathology, University of Michigan, for the cytometry data and diagnoses. They thank the reviewers of this paper for their significant contributions.

REFERENCES

[1] K. Fukunaga and D. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vol. C-20, Feb. 1971.
[2] F. Camastra and A. Vinciarelli, "Estimating the intrinsic dimension of data with a fractal-based method," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 24, pp. 1404–1407, Oct. 2002.
[3] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Neural Inf. Process. Syst.*, Vancouver, CA, Dec. 2002.
[4] J. Costa and A. O. Hero, *Statistics and Analysis of Shapes*. Cambridge, MA: Birkhauser, 2006, ch. Learning Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces, pp. 231–252.
[5] E. Levina and P. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Neural Inf. Process. Syst.*, Vancouver, CA, Dec. 2004.
[6] K. M. Carter, A. O. Hero, and R. Raich, "De-biasing for intrinsic dimension estimation," in *Proc. IEEE Statist. Signal Process. Workshop*, Aug. 2007, pp. 601–605.
[7] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, vol. 14.
[9] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 1, pp. 2323–2326, 2000.
[10] A. P. Petland, "Fractal-based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 661–674, 1984.
[11] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 72–77, Jan. 1995.
[12] B. B. Mandelbrot, *The Fractal Geometry of Nature*. San Francisco, CA: Freeman, 1982.
[13] K. M. Carter and A. O. Hero, "Variance reduction with neighborhood smoothing for local dimension estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2008, pp. 3917–3920.
[14] J. A. Costa and A. O. Hero, "Geodesic entropic graphs for dimension and entropy estimation in manifold learning," *IEEE Trans. Signal Process.*, vol. 52, pp. 2210–2221, Aug. 2004.
[15] F. Camastra, "Data dimensionality estimation methods: A survey," *Pattern Recognit.*, vol. 36, no. 12, pp. 2945–2954, 2003.
[16] V. I. Koltchinskii, *Empirical Geometry of Multivariate Data: A Deconvolution Approach*, vol. 28, no. 2, pp. 591–629, 2000.
[17] V. Pestov, "An axiomatic approach to intrinsic dimension of a dataset," *Neural Netw.*, vol. 21, no. 2–3, pp. 204–213, 2007.
[18] N. Tatti, T. Mielikainen, A. Gionis, and H. Mannila, "What is the dimension of your binary data?," in *Proc. 6th Int. Conf. Data Mining*, Hong Kong, 2006, pp. 603–612.
[19] M. Hein and J. Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in R^d," in *Proc. 22nd Int. Conf. Machine Learn.*, 2005, pp. 289–296.
[20] M. Raginski and S. Lazebnik, "Estimation of intrinsic dimensionality using high-rate vector quantization," in *Proc. 19th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 1105–1112.
[21] S. N. Lahiri, *Resampling Methods for Dependent Data*. New York: Springer, 2003.
[22] L. Breiman, "Bagging predictors," *Machine Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[23] J. H. Friedman and B. E. Popescu, Predictive learning via rule ensembles Stanford Univ., Tech. Rep., 2005.
[24] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *Proc. 7th Int. Conf. Database Theory*, Jerusalem, Israel, Jan. 1999, pp. 217–235.
[25] Y. Vardi and C.-H. Zhang, "The multivariate $L1$-median and associated data depth," in *Proc. Nat. Acad. Sci. USA*, 2000, vol. 97, pp. 1423–1426.

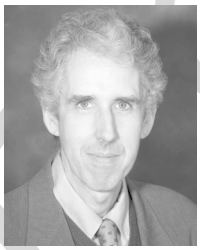
[26] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero, "Fine: Fisher information non-parametric embedding," *IEEE Trans. Pattern Anal. Machine Intell.* 2009 [Online]. Available: http://tbayes.eecs.umich.edu/kmcarter/papers/tpami_fine.pdf, to appear
[27] R. Kass and P. Vos, *Geometrical Foundations of Asymptotic Inference*, ser. Wiley Series in Probability and Statistics. New York: Wiley, 1997.
[28] W. G. Finn, K. M. Carter, R. Raich, and A. O. Hero, "Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: Treating flow cytometry data as high-dimensional objects," *Cytometry B, Clin. Cytometry*, vol. 76B, no. 1, pp. 1–7, Jan. 2009.
[29] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero, "Information preserving component analysis: Data projections for flow cytometry analysis," *IEEE J. Sel. Topics Signal Process. (Special Issue on Digital Image Processing Techniques for Oncology)*, vol. 3, no. 1, pp. 148–158, Feb. 2009.
[30] K. M. Carter, "Dimensionality reduction on statistical manifolds," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Jan. 2009.
[31] S. Grikshchat, J. A. Costa, A. O. Hero, and O. Michel, "Dual rooted-diffusions for clustering and classification on manifolds," in *Proc. 2006 IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2006, vol. 5.
[32] J. C. Bezdec, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[33] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[34] P. Jaccard, "The distribution of flora in the alpine zone," *New Phytologist*, vol. 11, pp. 37–50, 1912.

Kevin M. Carter (S'08) received the B.Eng. degree *(cum laude)* in computer engineering from the University of Delaware, Newark, in 2004. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2006 and 2009, respectively.

He is now a Member of Technical Staff at MIT Lincoln Laboratory, working on problems of network security and anomaly detection. His main research interests lies in manifold learning, with specific focus on statistical manifolds, information geometric approaches to dimensionality reduction, and intrinsic dimension estimation. His additional research interests include statistical signal processing, machine learning, and pattern recognition.

Raviv Raich (S'98–M'04) received the B.Sc. and M.Sc. degrees from Tel Aviv University, Tel Aviv, Israel, in 1994 and 1998, respectively, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 2004, all in electrical engineering.

Between 1999 and 2000, he was a Researcher with the Communications Team, Industrial Research, Ltd., Wellington, New Zealand. From 2004 to 2007, he was a Postdoctoral Fellow with the University of Michigan, Ann Arbor. Since fall 2007, he has been an Assistant Professor in the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis. His main research interest is in statistical signal processing, with specific focus on manifold learning, sparse signal reconstruction, and adaptive sensing. His other research interests lie in the area of statistical signal processing for communications, estimation and detection theory, and machine learning.



Alfred O. Hero III (S'79–M'84–SM'96–F'98) received the B.S. degree *(summa cum laude)* from Boston University, Boston, MA, in 1980 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1984, both in electrical engineering.

Since 1984, he has been with the University of Michigan, Ann Arbor, where he is a Professor in the Department of Electrical Engineering and Computer Science and, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. His recent research interests have been in areas, including inference in sensor networks, adaptive sensing, bioinformatics, inverse problems, and statistical signal and image processing.

He received an IEEE Signal Processing Society Meritorious Service Award (1998), an IEEE Signal Processing Society Best Paper Award (1998), and the IEEE Third Millenium Medal (2000). He was President of the IEEE Signal Processing Society (2006–2008) and is Director-Elect of IEEE for Division IX (2009).

On Local Intrinsic Dimension Estimation and Its Applications

Kevin M. Carter, *Student Member, IEEE*, Raviv Raich, *Member, IEEE*, and Alfred O. Hero III, *Fellow, IEEE*

***Abstract*—In this paper, we present multiple novel applications for local intrinsic dimension estimation. There has been much work done on estimating the global dimension of a data set, typically for the purposes of dimensionality reduction. We show that by estimating dimension locally, we are able to extend the uses of dimension estimation to many applications, which are not possible with global dimension estimation. Additionally, we show that local dimension estimation can be used to obtain a better global dimension estimate, alleviating the negative bias that is common to all known dimension estimation algorithms. We illustrate local dimension estimation's uses towards additional applications, such as learning on statistical manifolds, network anomaly detection, clustering, and image segmentation.**

***Index Terms*—Geodesics, image segmentation, intrinsic dimension, manifold learning, nearest neighbor graph.**

I. Introduction

TECHNOLOGICAL advances in both sensing and media storage have allowed for the generation of massive amounts of high-dimensional data and information. Consider the class of applications that generate these high-dimensional signals: e.g., digital cameras capture images at enormous resolutions; dozens of video cameras may be filming the exact same object from different angles; planes randomly drop hundreds of sensors into the same area to map the terrain. While this has opened a wealth of opportunities for data analysis, the problem of the *curse of dimensionality* has become more substantial, as many learning algorithms perform poorly in high dimensions. While the data in these applications may be represented in high dimensions, strictly based upon the immense capacity for data retrieval, it is typically concentrated on lower dimensional subsets—manifolds—of the measurement space. This allows for significant dimension reduction with minor or no loss of information. The point at which the data can be reduced with minimal loss is related to the *intrinsic dimensionality* of the manifold supporting the data. This measure may be interpreted as the minimum number of parameters required to describe the data [1].

When the intrinsic dimension is assumed constant over the data set, several algorithms [2]–[5] have been proposed to estimate the dimensionality of the manifold. In several problems of practical interest, however, data will exhibit varying dimensionality, as there may lie multiple manifolds of varying dimension within the data. This is easily viewed in images with different textures or in classification tasks in which data from different classes is generated by unique probability density functions (pdfs). In these situations, the local intrinsic dimension may be of more importance than the global dimension. In previous work [6], we illustrated the process of local dimension estimation, in which a dimension estimate is obtained for each sample within the data, rather than a single dimension estimate for the entire set.

In this paper, we focus on the applications of local dimension estimation. One immediate benefit is using local dimension to estimate the global dimension of a data set. To our knowledge, every method of estimating intrinsic dimension has expressed an issue with a negative bias. While insufficient sampling is a common source of this bias, a significant portion is a result of samples near the boundaries or edges of a manifold. These regions appear to be low dimensional when sampled and contribute a strong negative bias to the global estimate of dimension. We additionally utilize local dimension estimation for the purposes of dimensionality reduction. Typically, this has been presented for Riemannian manifolds in a Euclidean space [7]–[9], in which the data contain a single manifold of constant dimension lying in $\mathbb{R}^d$. We extend this to the problem of estimation and reduction of dimensionality on statistical manifolds, in which the points on the manifold are pdfs rather than points in a Euclidean space.

We continue by showing novel applications in which the exact dimension of the data is of no immediate concern, but rather the differences between the local dimensions. Dimensionality can be viewed as the number of *degrees of freedom* in a data set, and as such may be interpreted as a measure of data *complexity*. By comparing the local dimension of samples within a data set, we are able to identify different subsets of the data for analysis. For example, in a time-series data set, the intrinsic dimensionality may change as a function of time. By viewing each time step as a sample, we can identify changes in the system at specific time points. We illustrate this ability by finding anomalous activity in a router network. Additionally, the identification of subsets within the data allows for the immediate application of clustering and image segmentation. There has been much work presented on using fractal dimension estimation for image and tex-

Manuscript received November 05, 2008; revised July 20, 2009. . The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cedric Richard. This work was supported in part by the National Science Foundation under Grant CCR-0325571.

K. M. Carter is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: kmcarter@umich.edu).

R. Raich is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail: raich@eecs.oregonstate.edu).

A. O. Hero III is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: hero@umich.edu).

Digital Object Identifier 10.1109/TSP.2009.2031722

ture segmentation [10], [11]. In this paper, we do not make the model assumption that textures may be represented as a collection of fractals [12], and instead segment images using a novel method based on Euclidean dimension. We show that by using "neighborhood smoothing" [13] over the dimension estimates, we are able to find the regions that exhibit differing complexities, and use the smoothed dimension estimates as identifiers for the clusters/segments.

The organization of this paper is as follows: We give an overview of the two-dimension estimation algorithms we will utilize in our simulations in Section II. In Section III, we describe the process of neighborhood smoothing as a means of postprocessing for local dimension estimation. We illustrate the various novel applications of local dimension estimation in Section IV, including debiasing for global dimension estimation, manifold learning, anomaly detection, clustering, and image segmentation. Lastly, we offer a discussion and present areas for future work in Section V.

II. Dimension Estimation

We will now present two algorithms for dimension estimation: the k-nearest neighbor (k-NN) algorithm [4], [14] and the maximum likelihood estimator (MLE) method [5]. Please note that this paper makes no attempts to claim superiority of these algorithms over others. While there are many algorithms available for dimension estimation, we focus on these two as a means for illustrating the applications we later present. By utilizing two distinct methods, we hope to quell any concerns that our applications are algorithm dependent. For a thorough survey of intrinsic dimension estimation methods, we encourage the reader to view [15] and [16], as well as more recent work [17]–[20]

A. The k-Nearest Neighbor Algorithm for Dimension Estimation

Let $\boldsymbol{X}_n = \{x_1, \ldots, x_n\}$ be n independent identically distributed (i.i.d.) random vectors with values in a compact subset of $\mathbb{R}^d$. The (1-)nearest neighbor of x_i in $\boldsymbol{X}_n$ is given by

$$\arg \min_{x \in \boldsymbol{X}_n \setminus \{x_i\}} D(x, x_i)$$

where $D(x, x_i)$ is an appropriate distance measure between x and x_i; for the purposes of this paper, let us define $D(x, x_i) = \|x - x_i\|$, the standard Euclidean (L_2) distance. For a general integer $k \geq 1$, the k-nearest neighbor of a point is defined in a similar way. The k-NN graph assigns an edge between each point in $\boldsymbol{X}_n$ and its k-nearest neighbors. Let $\mathcal{N}_{k,i} = \mathcal{N}_{k,i}(\boldsymbol{X}_n)$ be the set of k-nearest neighbors of x_i in $\boldsymbol{X}_n$. The total edge length of the k-NN graph is defined as

$$L_{\gamma,k}(\boldsymbol{X}_n) = \sum_{i=1}^{n} \sum_{y \in \mathcal{N}_{k,i}} D(x, x_i)^{\gamma} \tag{1}$$

where $\gamma > 0$ is a power weighting constant.

For many data sets of interest, the random vectors $\boldsymbol{X}_n$ are constrained to lie on an m-dimensional Riemannian submanifold $\mathcal{M}$ of $\mathbb{R}^d$ ($m < d$). Under this framework, the asymptotic behavior of (1) is given as

$$L_{\gamma,k}(\boldsymbol{X}_n) = n^{\alpha(m)} c + \epsilon_n \tag{2}$$

where $\alpha(m) = (m - \gamma)/m$, c is a constant with respect to $\alpha(m)$ that depends on the Rényi entropy of the distribution of the manifold and ϵ_n is an error residual [6]. Note that for ease of notation, we will denote $\alpha(m)$ simply as α, except where the explicit expression is desirable (e.g., optimizing over m).

As noisy measurements can lead to inaccurate estimates, the intrinsic dimension $\hat{m}$ should be estimated using a nonlinear least squares solution. By calculating sampled graph lengths over varying values of n, the effect of noise ϵ_n can be diminished. In order to calculate graph lengths for differing sample sizes on the manifold, it is necessary to randomly subsample from the full set $\boldsymbol{X}_n = \{x_1, \ldots, x_n\}$, utilizing the nonoverlapping block bootstrapping method [21]. Specifically, let $\boldsymbol{X}'_n = \{x_{(1)}, \ldots, x_{(n)}\}$ be a spatially or temporally sorted version of $\boldsymbol{X}_n$, and let w be an integer satisfying $w < n/Q$. Define the blocks $\mathcal{B}_i = (x_{((i-1)w+1)}, \ldots, x_{(iw)})$, $i = 1, \ldots, n/w$. As such, we may now redefine $\boldsymbol{X}'_n = \{\mathcal{B}_1, \ldots, \mathcal{B}_{n/w}\}$.

Let $\{p_1, \ldots, p_Q\}$ be Q integers such that $1 \leq p_1 < \ldots, < p_Q \leq n/w$. For each value of $p \in \{p_1, \ldots, p_Q\}$, randomly draw N bootstrap datasets $\boldsymbol{X}_p^j$, $j = 1, \ldots, N$, with replacement, where the p blocks of data points within each $\boldsymbol{X}_p^j$ are chosen from the entire data set $\boldsymbol{X}'_n$ independently. From these samples, define $\boldsymbol{L}_n = \{L_{\gamma,k}(\boldsymbol{X}_p^1), \ldots, L_{\gamma,k}(\boldsymbol{X}_p^N)\}$, where $n = pw$.

Since c is dependent on m, it is necessary to solve for the minimum mean squared error, derived from (2), by minimizing over both c and integer values of $m \in \mathbb{Z}$

$$\hat{m} = \arg \min_{m \in \mathbb{Z}} \left\{ \min_{c} \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i} - n_i^{\alpha(m)} c \mathbf{1}\|^2 \right\} \tag{3}$$

where $n_i = p_i w$ and $\mathbf{1}$ is the vector of length n_i whose elements are all one. We solve over integer values of m, as we do not consider fractal dimensions for this algorithm. This improves accuracy by constraining the estimation space to discrete values rather than discretizing estimates in a continuous space. One can solve (3) in the following general manner.

```
for  m = 2 to d do
```

1: Calculate $\hat{c}(m)$ from the expansion of (3)

a) $\hat{c} = \min_c \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i}\|^2 - 2c \sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} + c^2 \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T \mathbf{1}$

$\Rightarrow \hat{c} = \sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} / \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T \mathbf{1}$

2: Calculate the error $\boldsymbol{\epsilon}(m)$ with m and $\hat{c}$ from step 1)

$\boldsymbol{\epsilon}(m) = \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i} - \hat{c} n_i^{\alpha(m)} \mathbf{1}\|^2$

```
end.
```

$\hat{m} = \arg\min_i \boldsymbol{\epsilon}(i)$

This nonlinear least squares solution yields the dimension estimate $\hat{m}$ based on the k-NN graphs.

B. The Maximum Likelihood Estimator for Intrinsic Dimension

The MLE method [5] for dimension estimation estimates the intrinsic dimension $\hat{m}$ from a collection of i.i.d. observations $\boldsymbol{X} = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$, $m \leq d$. Similar to the k-NN algorithm for dimension estimation, the MLE method assumes that

close neighbors lie on the same manifold. The estimator proceeds as follows, letting k be a fixed number of nearest neighbors to sample x_i

$$\hat{m}_k(x_i) = \left[\frac{1}{k-2} \sum_{j=1}^{k-1} \log \frac{T_k(x_i)}{T_j(x_i)} \right]^{-1} \tag{4}$$

where $T_k(x_i)$ is the distance from point x_i to its kth nearest neighbor in $\boldsymbol{X}$. The intrinsic dimension for the data set can then be estimated as the average over all observations

$$\hat{m}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{m}_k(x_i).$$

Algorithm 1 Local dimension estimation

Input: Data set $\boldsymbol{X} = \{x_1, \ldots, x_n\}$
1: **for** $i = 1$ to n **do**
2: Initialize cluster $\mathcal{C} = x_i$
3: **for** $k = 1$ to n'
4: Find the kth NN, $x_{k,i}$, of x_i
5: $\mathcal{C} \leftarrow \mathcal{C} \cup x_{k,i}$
6: **end for**
7: $\hat{m}(x_i) = \text{dimension}(\mathcal{C})$
8: **end for**

Output: Local dimension estimates $\hat{m}(x_i)$ for $i = 1, \ldots, n$

C. Local Dimension Estimation

While the MLE method inherently generates local dimension estimates for each sample $\hat{m}(x_i)$, the k-NN algorithm in itself is a global dimension estimator. We are able to adopt it (and any other dimension estimation algorithm) as a local dimension estimator by running the algorithm over a smaller neighborhood about each sample point. Define a set of n samples $\boldsymbol{X} = \{x_1, \ldots, x_n\}$ from the collection of manifolds $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$ such that each point x_i lies on manifold $\mathcal{M}_j$. Any small sphere or data cluster of samples $\mathcal{C} \subseteq \boldsymbol{X}$ centered at point x_i, with $|\mathcal{C}| = n' \leq n$, will contain samples from $M' \leq M$ distinct manifolds. As $n' \to 1$, all of the points in $\mathcal{C}$ will lie on a single manifold (i.e., $M' \to 1$). Intuitively speaking, as the cluster about point x_i is reduced in size, the local neighborhood defined by said cluster can be viewed as its own data set confined to a single manifold. Hence, we can use a global dimension estimation algorithm on a local subset of the data to estimate the local intrinsic dimension of each sample point. This can be performed as described in Algorithm 1, where "dimension($\mathcal{C}$)" refers to applying any method of dimension estimation to the data cluster $\mathcal{C}$.

One of the keys to local dimension estimation is defining a value of n'. There must be a significant number of samples in order to obtain a proper estimate, but it is also important to keep a small sample size as to (ideally) only include samples that lie on the same manifold. Currently, we arbitrarily choose n' based on the size of the data set. However, a more definitive method of choosing n' is grounds for future work.

We briefly note that our definition of "local" dimension estimation differs from that of the Fukunaga–Olsen algorithm [1]. Specifically, we aim to find a dimension estimate for each sample point, which accounts for sets consisting of multiple manifolds, while [1] used local subsets of the data to form a global estimate of dimension.

III. Neighborhood Smoothing

For the problem of local dimension estimation, results are often highly variable, where nearby samples from the same manifold may result in different dimension estimates. This issue can be a result of a variety of reasons, such as variability due to random subsampling in the k-NN algorithm, or variability due to the neighborhood size in the MLE method. When constructing a global dimension estimate, this variance is relatively insignificant, as the estimate is constructed as a function of the local estimates. For local dimension estimation, however, this variance is of significant concern, and we propose a variance reduction method known as neighborhood smoothing [13], which improves estimation accuracy.

An initial intuition for manifold learning algorithms is that samples that are "close" tend to lie on the same manifold, which extends to the assumption that they therefore have the same dimension. With this assumption in place, it follows that filtering by majority vote over the dimension estimates of nearby samples should smooth the estimator and reduce variance. This voting strategy is similar to the methods of mode filtering, bagging [22] and learning by rule ensembles [23]. Smoothing simply looks at the distribution of dimension estimates within each sample point's local neighborhood and reassigns each sample a dimension estimate equal to that with the highest probability within its neighborhood. Specifically

$$\hat{m} = \arg\max_l \mathrm{P}_{\mathcal{N}_i}[\hat{m} = l] \tag{5}$$

where $\mathrm{P}_{\mathcal{N}_i}$ is the probability over the neighborhood of the current sample $\mathcal{N}_i$. Given a finite number of samples $\{x_1, \ldots, x_n\}$, this may be empirically evaluated as

$$\hat{m}(x_i) = \arg\max_l \sum_{x_j \in \mathcal{N}_i} I(\hat{m}(x_j) = l) \tag{6}$$

where $I(\cdot)$ is the standard indicator function. This process may then be iterated until the set converges such that each estimate remains constant. This has the effect of implicitly incorporating the neighbors of each sample's neighbors to some extent, as the dimension estimates within a local region may change through iterations.

Intuitively, neighborhood smoothing is similar to iteratively imposing a k-NN classifier on the local dimension estimates—under the guise that at each iteration, sample x_i is a test sample and all points x_j, $j \neq i$ are appropriately labeled training samples. Similarly to k-NN classification, the key factor to smoothing is defining the neighborhood $\mathcal{N}_i$. If $\mathcal{N}_i$ is too large, oversmoothing will occur. The variance of the dimension estimates will drastically decrease, but there will be a strong bias which will remove the detection of coarsely sampled manifolds. As such, one cannot use a constant region about a point but must adapt that region to the statistics of the sample.

A. Adaptive Neighborhood Selection

Since the number of sample points on each manifold of a data set is generally unknown, using a constant number of smoothing samples is not a viable option; samples on a smaller manifold may have points from a disjoint manifold included in their smoothing neighborhood. One straightforward method for neighborhood selection is to define neighbors by some spherical region or ϵ-ball about each sample point. This is generally acceptable when the disjoint manifolds are easily separable, as the neighborhood does not adapt to the geometry of the manifold. When distinct manifolds lie near one another, or potentially intersect, it is necessary to further adapt the smoothing neighborhood beyond a spherical region. This is due to the fact that points on a nearby or intersecting manifold may be as close (or closer) to a sample as others on its own manifold. A spherical region may smooth over different manifolds, and the results will lead to the dimension estimates' "leaking" from one manifold to another.

Rather than defining neighborhoods through Euclidean distance, which will form only spherical regions about each sample point, we will define neighborhoods using a geodesic distance metric. This will adapt the neighborhood to the geometry of the manifold. The geodesic distance is defined as the shortest path between two points along the manifold and may be approximated with graph-based methods. For our purposes, this metric can be determined by taking each point and creating an edge to its k-NN. Then, using Dijkstra's shortest path algorithm (or any other algorithm for computing the shortest path), approximate the geodesic distances between each pair of points in the graph. Any points that remain unconnected are considered to have an infinite geodesic distance between them.

To define a local neighborhood, we can now simply choose the closest n_g points for which the geodesic distance is not infinite. This forms a nonspherical neighborhood that adapts to the curvature of the manifold, performing much better than spherical neighborhoods. Fig. 1 illustrates the difference in the neighborhoods (black stars) that are formed on the "swiss roll" manifold when using different proximity metrics. The Euclidean distance [Fig. 1(a)] forms a spherical neighborhood, including points that are separated from the sample in question (red diamond). The geodesic distance [Fig. 1(b)], however, forms a neighborhood considering points only in close proximity along the actual manifold. While all points in this example do exist on the same manifold, it is clear that defining neighborhoods along the manifold rather than in simple spherical regions reduces the probability of including samples from a nearby distinct manifold.

Illustrating the effects of neighborhood smoothing, we create a seven-dimensional data set that includes two distinct hyperspheres of intrinsic dimensions two and five, each containing 300 uniformly sampled points intersecting in three common dimensions. Fig. 2(a) shows the histogram of the local dimension estimates of each sample before any neighborhood smoothing was applied, while Fig. 2(b) shows the results after smoothing. One can clearly see that the wide histogram was correctly condensed to the proper local dimension estimates, even though the manifolds intersect. The use of the geodesic distance measure prevents smoothing across distinct manifolds, which lie closely together in Euclidean space.

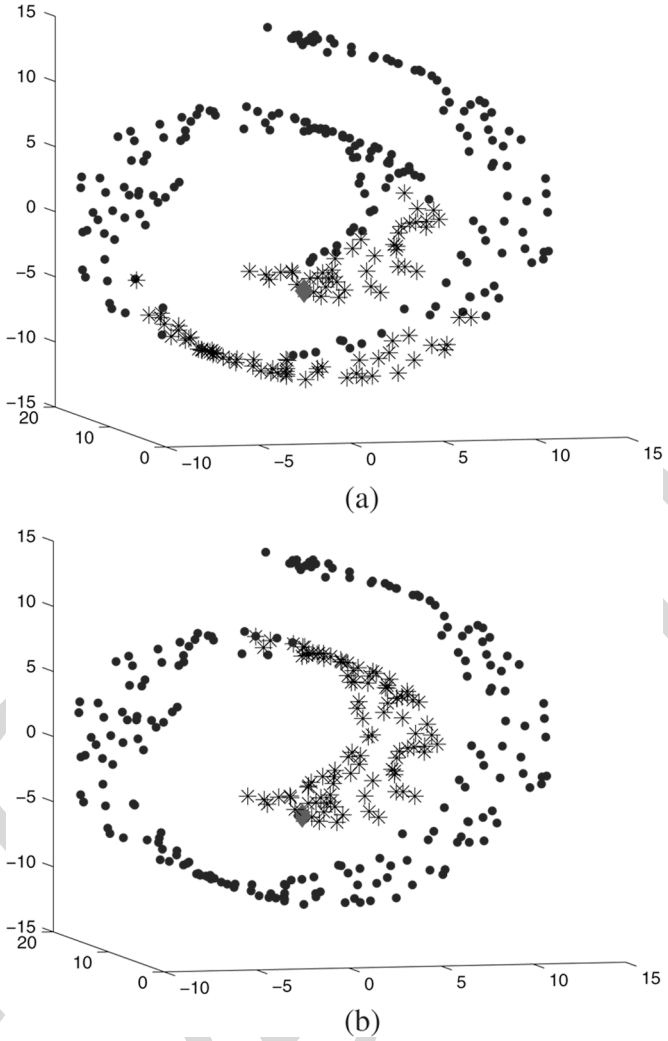


Fig. 1. Neighborhoods ($*$) of the sample in question ($\diamond$) defined by (a) Euclidean distance and (b) geodesic distance. (a) Spherical neighborhood and (b) adaptive neighborhood.

It is important to note that, as with any form of postprocessing, neighborhood smoothing can only produce accurate results given sufficient input. The benefits of smoothing can be significantly diminished if the initial local dimension estimates are not sufficiently accurate. We note this explicitly because of the known issues with estimating large dimensions (e.g., $m > 20$). Because of variance issues due to insufficient samples and boundary effects, it is difficult to accurately estimate very large dimensions, and often the estimate can more appropriately be considered a measure of *complexity*, where the difference between m and $m+1$ is rather insignificant. This is important because no single dimension may dominate a given local neighborhood, yet smoothing will still assign a dimension estimate equal to the most represented dimension, which may indeed be inconsistent with the rest. We demonstrate this scenario with the example shown in Fig. 3, where smoothing would assign a dimension estimate of $m = 40$, which is the most represented dimension in the neighborhood. However, a more accurate dimension estimate could be considered $m = 33$

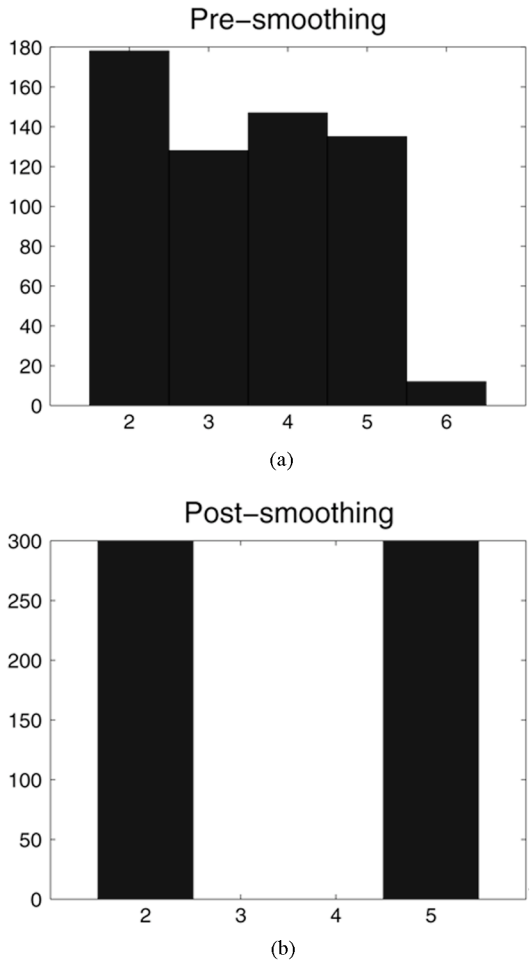


Fig. 2. Neighborhood smoothing applied to seven-dimensional data containing two spheres with intrinsic dimensions two and five.

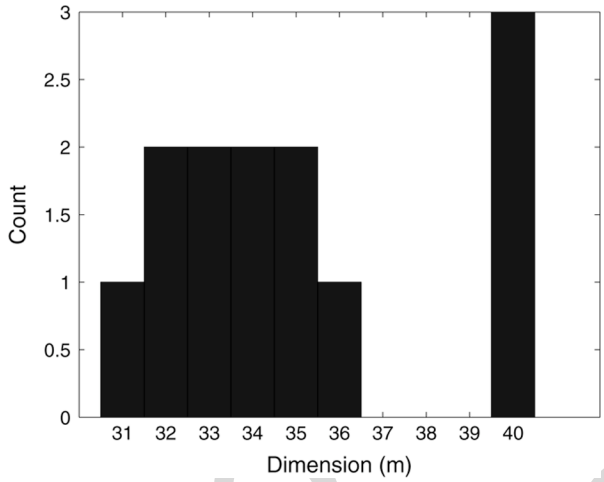


Fig. 3. Issues arise with neighborhood smoothing when estimating very large dimensions due to the variance of such estimates. In this example, smoothing would assign a dimension estimate of 40, although the more appropriate estimate would be 33 or 34.

or $m = 34$, as that would be more consistent with the majority of the samples. In these scenarios, it may be more appropriate to smooth over a histogram with user-defined bin sizes, corresponding to significant differences in complexity rather than individual dimensions. This is an area for future work.

IV. Applications

A. Debiasing Global Dimension Estimation

To our knowledge, a phenomenon common to all algorithms of intrinsic dimension estimation is a negative bias in the dimension estimate. It is believed that this is an effect of undersampling the high-dimensional manifold. While the bias due to lack of sufficient samples is inherent, we offer that the sample size is not the only source of bias; a significant portion is related to the depth of the data. Specifically, as data samples approach the boundaries of the manifold, they exhibit a lower intrinsic dimension. This issue becomes more prevalent as the dimension of the manifold increases and is directly related to the *curse of dimensionality*. Note that even manifolds that appear "empty" in their extrinsic-dimensional space (e.g., the Swiss roll) are filled and contain boundaries in the space of their intrinsic dimension.

Previous work [24] has demonstrated that as dimensionality increases, the nearest neighbor distances approach those of the most distant points; this will clearly have an adverse effect on neighborhood-based estimation algorithms. We are able to further correlate this effect on dimension estimation by calculating the depth of each sample and quantitatively analyzing the relationship between depth and dimension. We utilize the L_1-data depth algorithm developed in [25], which calculates depth $D_n(x)$ as the sum of all the unit vectors between the sample of interest $x \in \boldsymbol{X}$ and the rest of the data set $\boldsymbol{X} = \{x_1, \ldots, x_n\}$. Specifically

$$D_n(x) = 1 - \max\left(0, \left\| \sum_{x_i \neq x} e(x_i - x)/n \right\| - \sum_{x_i = x} \frac{1}{n}\right) \tag{7}$$

where $e(x_i - x) = (x_i - x)/\|x_i - x\|$ is the unit vector in the direction of $(x_i - x)$. This depth measure assigns the most interior points in the data set a depth value approaching one, while samples along the boundaries approach a depth of zero.

Using this measure, we illustrate the effect of data depth on dimension estimation in Fig. 4. The data set used was of 3000 points uniformly sampled on a six-dimensional hypercube. We utilize the MLE method for dimension estimation, and Fig. 4 illustrates the distribution of data depths for samples that estimate at different dimensions. It is clear that as the depth increased, so did the probability of estimating at a higher dimension, even to the point where the most deep points estimated at a dimension of seven (although we note that there were very few points with this estimate).

When estimating the global dimension of a data set, one can substantially reduce the negative bias by placing more emphasis on the local dimension of those points away from the boundaries, as they are more indicative of the true dimension of the manifold. Specifically, let the global dimension be estimated as follows:

$$\hat{m} = \frac{1}{\sum_j W_j} \sum_i W_i \hat{m}(x_i) \tag{8}$$

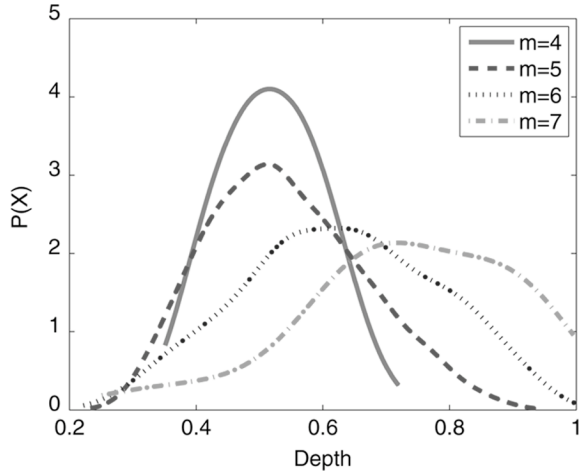


Fig. 4. PDFs of data depth based on estimated intrinsic dimension. Points with less depth estimate at a lower dimension, contributing to the overall negative bias.

where W_i is a weighting on each sample point. We offer two potential definitions of W_i, the first being a binary weighting

$$W_i = \begin{cases} 1, & D_n(x_i) \geq D_n(x_{(\beta \times n)}) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $0 \leq \beta \leq 1$ and $D_n(x_{(\beta \times n)})$ is the data depth of the $\beta \times n$ deepest point. Essentially this binary weight amounts to debiasing by averaging over the local dimension estimates of the deepest $\beta \times 100\%$ of points, where the threshold β is user defined. This is worthwhile for potentially large data sets, where there are enough samples to ignore a large portion of them. When this is not the case, let us make the definition

$$W_i = \mathrm{e}^{-(1-D_n(x_i))/c} \tag{10}$$

where c is a user-defined constant. This weighting may be viewed as a heat kernel, in which larger depths will yield higher weights. Unlike the binary weighting, which will ignore a large number of the data samples, this heat kernel weighting will utilize all samples (even those lying on a boundary) yet give preference to those with more depth in the manifold.

We now illustrate this debiasing ability in Fig. 5, in which we estimated the global dimension of the six-dimensional hypercube (3000 i.i.d. samples) over 200 unique trials. Fig. 5(a) shows the histogram of biased dimension estimates obtained by using the entire set for dimension estimation, while Fig. 5(b) estimates the correct global dimension each trial by using our debiasing method (8) with the binary weighting function (9) using $\beta = 0.5$.

To study the number of samples necessary to accurately estimate global dimension, we plot estimation results in Fig. 6. In this simulation, we plot the mean de-biased ($\beta = 0.5$) and unrounded dimension estimated over a 20-fold cross validation, based on differing number of samples on the six-dimensional hypercube. We can see that if rounded to the nearest integer, the debiased estimate will be correct on average with roughly 2500 samples. On the contrary, without debiasing, the estimation maintains a much stronger negative bias, never correctly estimating the dimension when rounded.

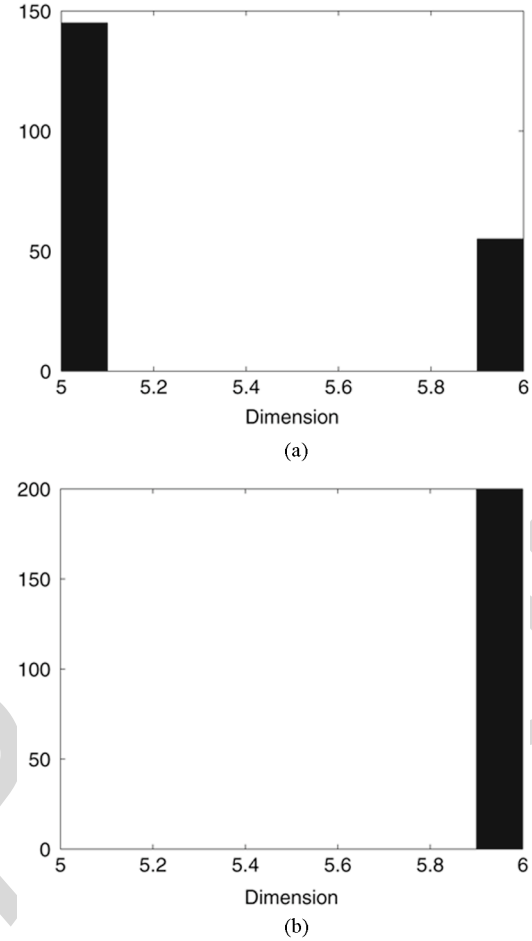


Fig. 5. Developing a debiased global dimension estimate by averaging over the 50% of points with the greatest depth on the manifold. (a) Biased results and (b) debiased results.

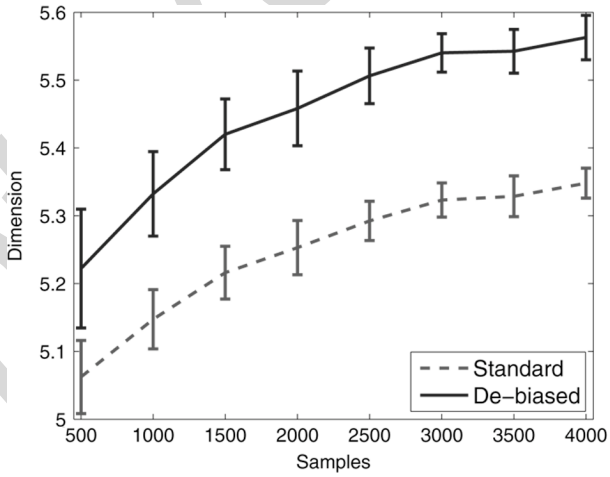


Fig. 6. Analysis of how many samples are necessary to appropriately estimate debiased global dimension. Plot shows mean dimension estimated over a 20-fold cv, with error bars at one standard deviation.

It is important to note that our method of debiasing is only applicable for data with a relatively low intrinsic dimension. When dealing with very high dimensional data, the probability of a sample lying near a boundary approaches one, and the value of the depth approximation becomes irrelevant. This is shown in

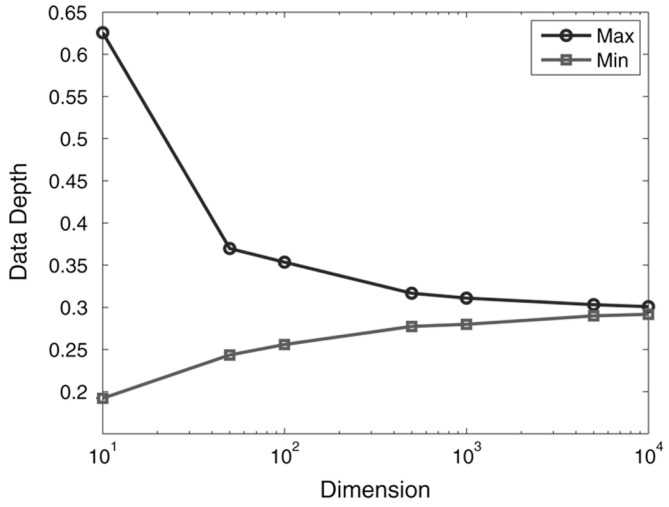


Fig. 7. As the intrinsic dimension increases, the maximum and minimum data depth of points in the set converge to the same value. This simulation was over a fivefold cross-validation with 400 uniformly sampled points on the unit cube.

Fig. 7 where the "deepest" and most "shallow" samples converge to the same depth value as the intrinsic dimension increases.

Prior work on estimating dimension through vector quantization [20] has reported robustness to negative bias. While not offering a distinct claim or proof of this robustness, the authors mention their algorithm obtains larger estimates for high-dimensional data than neighborhood-based methods. Theoretically, this method will suffer from similar bias issues due to the intrinsic geometry of the data, which is not explicitly accounted for in [20]. The improved performance reported is likely due to the cross-validation implemented. That said, the use of quantization error may indeed be *more* robust to negative bias than neighborhood-based methods, and this potential gain is worth further investigation.

B. Statistical Manifold Learning

Of particular interest in manifold learning is the intrinsic dimension to which one can reduce the dimensionality of a data set with minimal loss of information. This is typically presented for data that lie on a Riemannian submanifold of Euclidean space. We extend this application to the problem of learning on statistical manifolds [26], in which each point on the manifold is a pdf. Rather than defining distance with Euclidean metrics, we approximate the Fisher information distance—with the Kullback–Leibler divergence and Hellinger distance—which is the natural metric on statistical manifolds [27]. We illustrate the use of local dimension estimation in these learning tasks for the applications of flow cytometry analysis and document classification.

1) Flow Cytometry Analysis: In clinical flow cytometry, pathologists gather readings of fluorescent markers and light scatter off individual blood cells from a patient sample, leading to a characteristic multidimensional distribution that, depending on the panel of markers selected, may be distinct for a specific disease entity. The data from clinical flow cytometry can be considered multidimensional both from the standpoint of multiple characteristics measured for each cell, and from the standpoint of thousands of cells analyzed per sample. In previous work [28], [29], we have shown the ability to derive an information embedding of the statistical manifold defined by the space of pdfs, in which each patient's blood sample can be considered a realization of a pdf on said manifold. We developed Fisher information nonparametric embedding (FINE) as an informationgeometric method of dimensionality reduction based on Fisher information distances between pdfs [26]. Using FINE, we are able to embed the pdf realizations into a low-dimensional Euclidean space, in which each patient is represented by a single low-dimensional vector. In order to determine the dimension m for this embedding space, we first apply local dimension estimation to find the desired dimension of our embedding space.

Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ be a collection of data sets where $\boldsymbol{X}_i$ corresponds to the flow cytometer output of the ith patient. For our analysis, each patient has either chronic lymphocytic leukemia or mantle cell lymphoma, which display similar characteristics with respect to many expressed surface antigens but are generally distinct in their patterns of expression of two common B lymphocyte antigens CD23 and FMC7. We are interested in the intrinsic dimension of the statistical manifold, realized by $\mathcal{X}$, as that is what we plan to embed. We define the dissimilarity matrix D, where $D(i, j)$ is the symmetric Kullback–Leibler divergence approximation of the Fisher information distance between pdf estimates on $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ [30]. For this simulation, we estimated the pdfs with kernel density estimation methods, although any nonparametric method will suffice. By redefining our local dimension estimation algorithms to take the high-dimensional distance matrix—in the space of pdfs—as an input (which is not an issue, as both the k-NN and MLE methods are entirely based on nearest neighbor distances), we are able to estimate the intrinsic dimension of the statistical manifold. The local dimension estimation results are shown in Fig. 8, where we can see the intrinsic dimension is $m = [2, 3]$. This result can be interpreted as recognizing the two specific markers that most significantly differentiate between classes (i.e., $m = 2$) but also accounting for the fact that there still exist subtle differences between members of the same class, and some patients may not exhibit the expected response to specific antigens as strongly as others (i.e., $m = 3$).

After estimating the intrinsic dimension of the data set, we are able to embed each patient into an m-dimensional Euclidean space, as observed in Fig. 9. In this embedding, each point represents a single patient data set, which was originally six-dimensional with samples on the order of $n \sim 5000$. We can see that a two-dimensional unsupervised embedding gives a clear class separation, which enables effective clustering and classification of the data. This result is consistent with our dimension estimate of $m = [2, 3]$ and illustrates the effectiveness of local dimension estimation for learning on statistical manifolds.

2) Document Classification: Given a collection of documents of known class, we wish to best classify a document of unknown class. A common representation of a document is known as the *term frequency* representation. This is essentially a normalized histogram of word counts within the document. Specifically, let x_i be the number of times term i appears in a specific document. The PdF of that document can then be

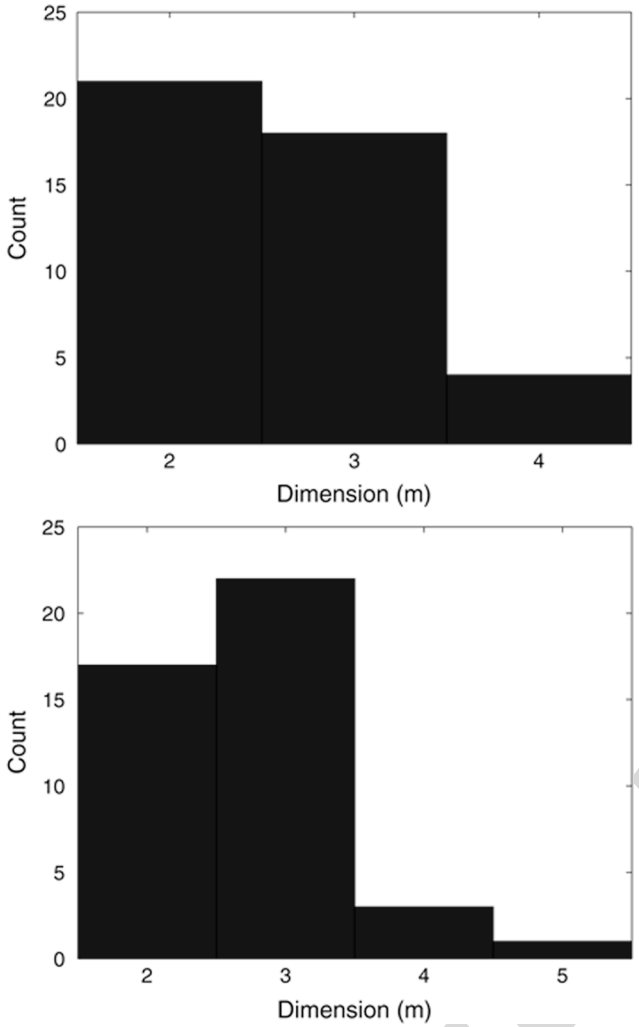


Fig. 8. Histogram of local dimension estimates for the statistical manifold defined by flow cytometry results of 43 patients with chronic lymphocytic leukemia or mantle cell lymphoma. (a) Local k-NN dimension estimates and (b) local MLE dimension estimates.

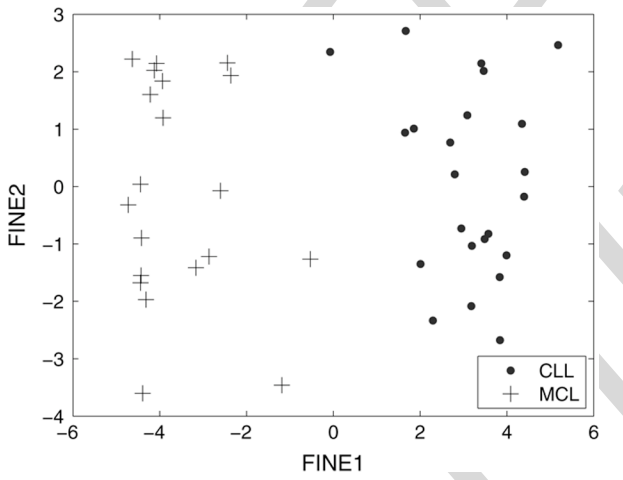


Fig. 9. The information-based embedding, determined by FINE, of the flow cytometry data set. Embedding into $m = 2$ dimensions yields linear separability between classes.

characterized as the multinomial distribution of normalized word counts, with the maximum likelihood estimate provided as

$$\hat{p}(x) = \left(\frac{x_1}{\sum_i x_i}, \ldots, \frac{x_N}{\sum_i x_i} \right). \tag{11}$$

For our illustration, we utilized the 20 Newsgroups[1] data set, which has an extrinsic dimension of $d = 26\,214$, which is the number of terms in its dictionary. This set contains postings from 20 separate newsgroups, and we wish to classify them by their highest domain (one of [comp.*, rec.*, sci.*, talk.*]). To perform this classification task, we first wish to alleviate the *curse of dimensionality* by reducing the data to a lower dimensional manifold. For this task we utilize FINE, approximating the Fisher information distance with the Hellinger distance, such that

$$D(i,j) = \sqrt{\sum_{l=1}^{N} (\sqrt{p_i(x_l)} - \sqrt{p_j(x_l)})^2}$$

where $p_i(x)$ is the estimate (11) of the pdf of document i.

Experimental results have shown there are multiple submanifolds of differing dimension in the data set. In Fig. 10, we present the distribution of dimension estimates and compare that to classification performance at reduced dimension. Specifically, we used the MLE method with the matrix of Hellinger distances (between full-dimensional pdfs) to estimate the local dimension of each sample, then used FINE to embed a random subsampling of 1000 points of the data into a lower dimension. The distribution of these local dimension estimates over a 20-fold cross-validation is shown in Fig. 10(a). Next, we separated the embedded data into a training set of 800 samples and a test set of 200 samples. Results of the linear, "all vs. all" classification task (i.e., classify each test sample as one of 4 different potential classes) are shown in Fig. 10(b) as a function of the embedding dimension (over the same 20-fold cross-validation).

We observe that the apex of the classification rate curve ($m \in [20, 50]$) corresponds to the apex of the pdf curve of local dimension estimates ($m \in [30, 70]$), which illustrates that the local dimension estimation method was able to find an appropriate embedding dimension. Although the range $m \in [30, 70]$ seems to be large, it is important to remember the extrinsic dimension of the data is $d = 26\,214$, so we are able to adequately define the dimension of the manifold. We note that for this simulation, we did not utilize neighborhood smoothing due to the high-dimensional nature of the data, as previously explained. A pdf of the local dimension estimates is more beneficial towards analysis than arbitrarily setting a dimension that does not dominate the neighborhood.

C. Network Anomaly Detection

Anomalies can be detected in router networks through the use of local dimension estimation [6]. Specifically, when only a few of the routers contribute disproportionately large amounts of traffic, there is a decrease in the intrinsic dimension of the entire network; that is, the space of traffic counts per router. Using

[1]http://people.csail.mit.edu/jrennie/20 Newsgroups/.

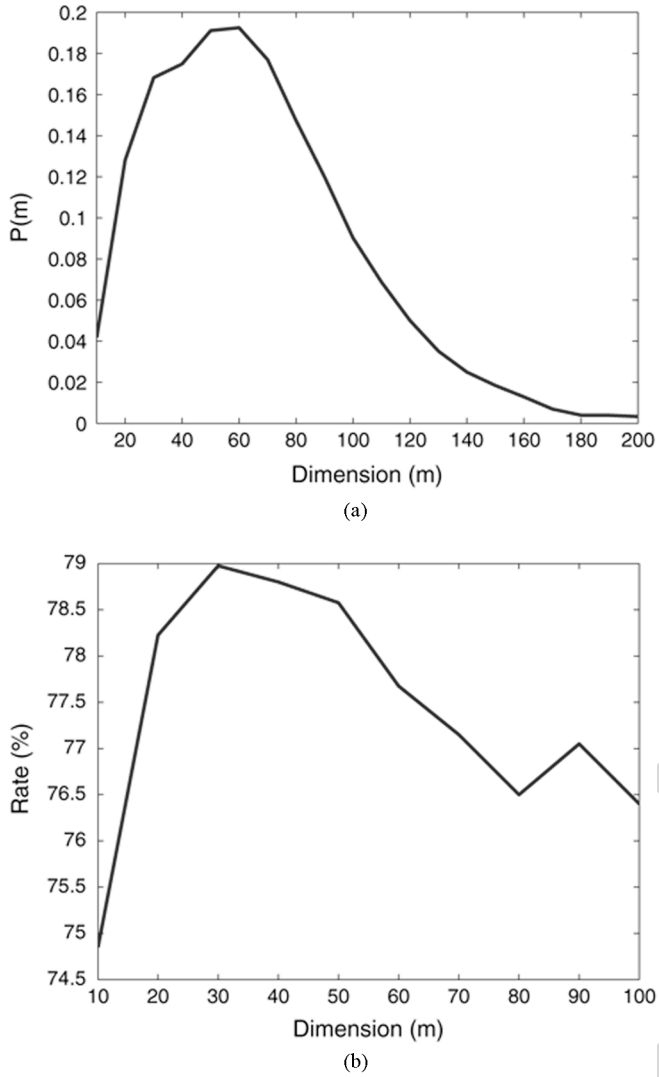


Fig. 10. Comparison of (a) pdf of local dimension estimates and (b) classification rate versus embedding dimension for the 20 Newsgroups data set. The optimal embedding dimension ranges from 20 to 50, which is in the same range as the apex of the local dimension estimation pdf.

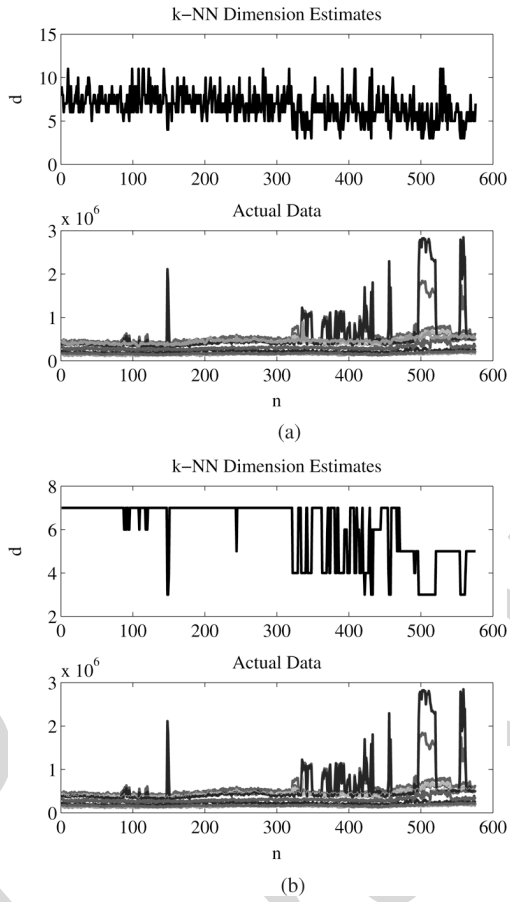


Fig. 11. Neighborhood smoothing applied to Abilene Network traffic data dimension estimation results. The x-axis represents time series changes in the network. (a) Before smoothing and (b) after smoothing.

neighborhood smoothing as a form of postprocessing, we are better able to locate the traffic anomalies, as the variance of the estimates is reduced. Fig. 11 illustrates the usage of neighborhood smoothing on the results of k-NN algorithm for local dimension estimation for anomaly detection. The data used are the number of packets counted on each of the 11 routers on the Abilene network, on January 1–2, 2005. Each sample is taken every 5 min, leading to 576 samples with an extrinsic dimension of $d = 11$.

Fig. 11(b) illustrates that neighborhood smoothing is able to preserve both the visually obvious ($n = 148$, $n > 300$) and nonobvious ($n = 87 - 120$) changes in network complexity. A detailed investigation of time $n = 244$, for example, reveals that the Sunnyvale router (SNVA) showed increased contribution from a single IP address. Large percentages (over half) of the overall packets had both source and destination IP 128.223.216.0/24 within port 119. This port showed increased activity on the Atlanta router as well. This change in dimensionality indicating anomalous activity would generally go unnoticed with the raw results of local dimension estimation due to the high variance [Fig. 11(a)].

We note the results shown in Fig. 11 are performed using nominal settings within the k-NN algorithm, which allows the algorithm to run quickly and accurately with neighborhood smoothing. We are able to generate results with much less variance than Fig. 11(a) by applying more averaging and bootstrapping, but this significantly increases computation time while still producing results with more variance than Fig. 11(b).

D. Clustering

As discussed previously, data sets often consist of multiple submanifolds of differing dimension. When the intrinsic dimension of these submanifolds becomes increasingly large, the value of the dimension may be interpreted as a measure of the *complexity* of the data. From this interpretation, we may use local dimension estimation to cluster data within a set by complexity. Specifically, we can define clusters through the use of recursive entropy estimation and neighborhood smoothing. As we increase the neighborhood size k, we incorporate more samples into our smoothing region, eventually oversmoothing between differing manifolds. By finding the point in which the smoothing regions extend into multiple manifolds, we can define clusters in the data. This point of change can be located

by analyzing the change in the entropy H of the dimension estimates as the region grows, such that

$$H = -\sum_j P_j \log P_j$$

where $P_j = (1)/(n)\sum_i^n I(\hat{m}(i) = j)$ is the empirical probability a sample estimates at dimension j.

When the regions are stable within each cluster, H will be constant. As the smoothing neighborhood incorporates additional manifolds, the entropy will leave its constant state and eventually $H \to 0$ as $k \to \infty$ (i.e., the region includes every point). With a priori knowledge of the distribution of dimensionality, one may choose a neighborhood size that yields an appropriate value of entropy. Without this knowledge, the point at which H leaves its constant state can be used as a threshold for defining clusters based on dimension. This process is similar to dual-rooted diffusion kernels method of clustering [31], in which the authors used the jump in nearest neighbor distance as a means to differentiate clusters.

For example, let $\boldsymbol{X} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$ is uniformly distributed in [0,1]^{m_i} ($m_i \in M$, a discrete set of integer values) and constant elsewhere. Hence, m_i is the intrinsic dimension of x_i. For our simulation, let $d = 13$ and $M = [2, 6, 10]$, and there are $n = 200$ samples for each value in M. After obtaining local dimension estimates, we apply neighborhood smoothing to differing neighborhood sizes and measure the entropy of the local dimension estimates at each size. The results are shown in Fig. 12, where the entropy exhibits the same pattern we previously described; after initially decreasing, H remains constant as k approaches the region size of each manifold ($n = 200$). As the smoothing covers multiple manifolds $k > 200$, the entropy decreases until the smoothing neighborhood eventually covers all manifolds simultaneously and $H = 0$. The histogram of local dimension estimates (with both k-NN and MLE methods), which is used to calculate the entropy, is shown in Fig. 13 to illustrate the evolution of the dimension estimates. It is clear that at $k = 100$, the three distinct clusters are represented, and this value also corresponds to the optimal entropy estimate given a priori knowledge that each dimension is represented with a constant probability of $P = (1)/(3)$, which yields the entropy value $H = 1.1$. Due to insufficient sampling, the actual value of the dimension estimates ([2,5,7] for the k-NN algorithm and [2,5,6] for the MLE method) differs from the true dimensions [2,6,10]. However, this is not of particular concern since the primary objective is to locate clusters of differing *complexity*. It is also worth noting that some samples are misidentified due to the overlapping nature of the three clusters, but the overall performance is respectable.

We note the dimension estimate obtained when smoothing over the entire set does not correspond to the global dimension of the data. Since we are using a majority voting method, the final value will be equal to the estimated dimension which is most represented (with simple tie-breaking rules). This is not necessarily equal to the global dimension, and is often not close to the dimension which best characterizes the entire data set (as in our example).

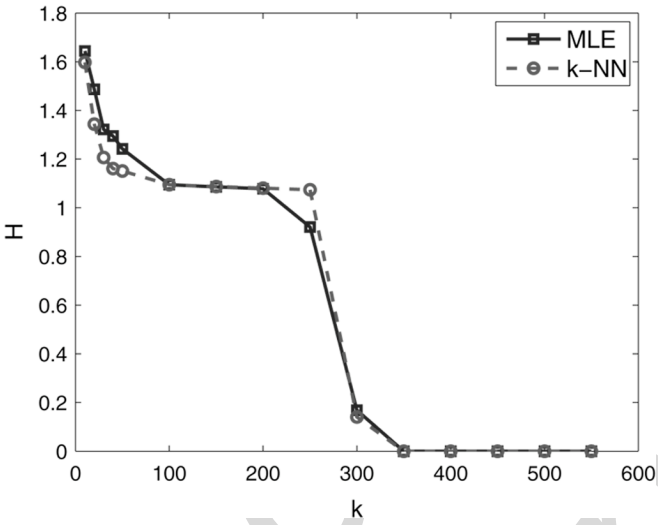


Fig. 12. The entropy of the local dimension estimates changes as a function of neighborhood size k.

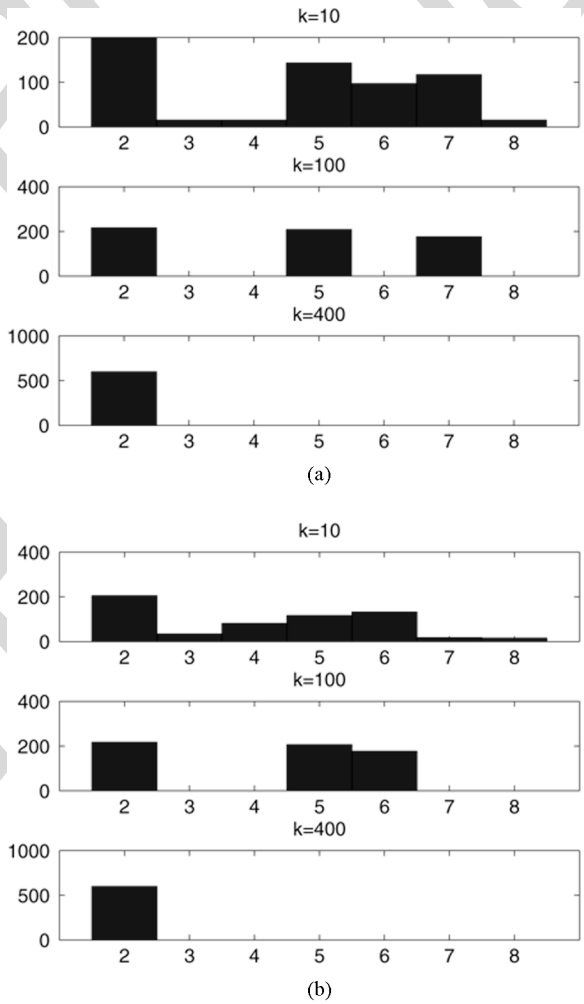


Fig. 13. Comparing dimension histograms of dimension estimates at various neighborhood sizes, we see that samples are clustered very well at $k = 100$, which corresponds to a constant point in the entropy plot shown in Fig. 12. (a) k-NN algorithm and (b) MLE method.

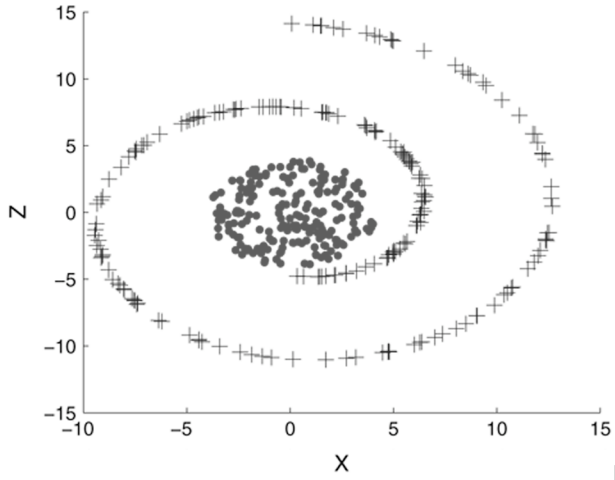
Let us now compare our clustering performance on a separate synthetic example. Consider the data set $\boldsymbol{X} = [x_1, \ldots, x_{400}]$ that consists of 200 points uniformly sampled on the "swiss roll" manifold and 200 points uniformly sampled on an intrinsically three-dimensional hypersphere. Hence, each $x_i \in \mathbb{R}^4$ (points sampled from the "swiss roll" have a constant value in the fourth dimension) and there are two distinct clusters formed. A visual representation of this set is illustrated in Fig. 14, and we compare our method of clustering by *complexity* using local dimension estimation with that of standard clustering methods—fuzzy c-means [32] and K-means [33]. To demonstrate clustering performance, we utilize the Jaccard index [34], which assesses the similarity between a predetermined set of class labels C and a clustering result K. Specifically

$$J(C,K) = \frac{a}{a+b+c}$$

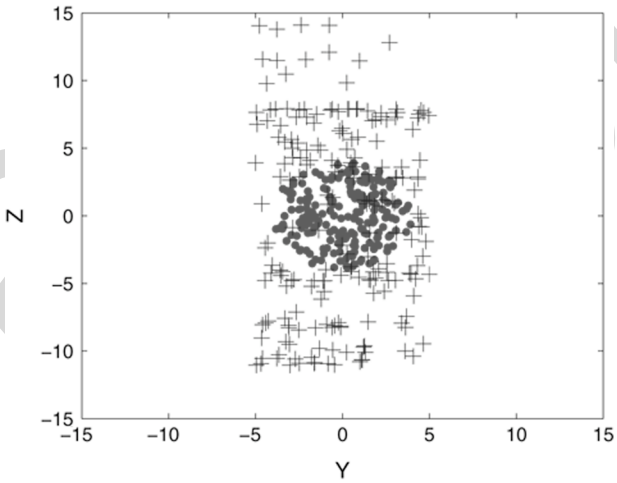
where a is the number of pairs of points with the same class label in C and the same cluster label in K; b is the number of pairs that have the same label C but differ in K; and c is the number of pairs of points with the same cluster label in K but different class label in C. Essentially, the Jaccard index gives a rating in the range [0,1] in which "1" signifies complete agreement between the true labels C and the results K.

We show the results in Table I over a 20-fold cross-validation with i.i.d. realizations of $\boldsymbol{X}$. We see clustering by dimension estimation yields far superior performance to standard methods. While these methods aim to cluster by a variety of means, such as optimizing distances to centroids, dimension estimation simply assigns cluster labels based on the local dimensionality of each data point. In this simulation, we utilized a neighborhood size of $k = 25$ when smoothing, as larger values tended to incorporate both manifolds since they are so close to one another. We acknowledge that clustering by dimensionality is not applicable in many practical problems in which the different clusters exhibit the same dimensionality. However, in the realm of high-dimensional clustering, there may often exist an intrinsic difference in dimensionality, in which our method would be applicable.

1) Image Segmentation: After showing the ability to use local dimension estimation for clustering data by complexity, a natural extension is to apply the methods for the problem of image segmentation. Differing textures in images can be considered to have different levels of complexity (e.g., a periodic texture is less complex than a random one). This has been well stated in [12], where natural images and textures are viewed as a collection of fractals. For our purposes, we chose to ignore such model assumptions and see whether or not Euclidean dimension can be used towards image segmentation. The same framework as our clustering method applies.



(a)



(b)

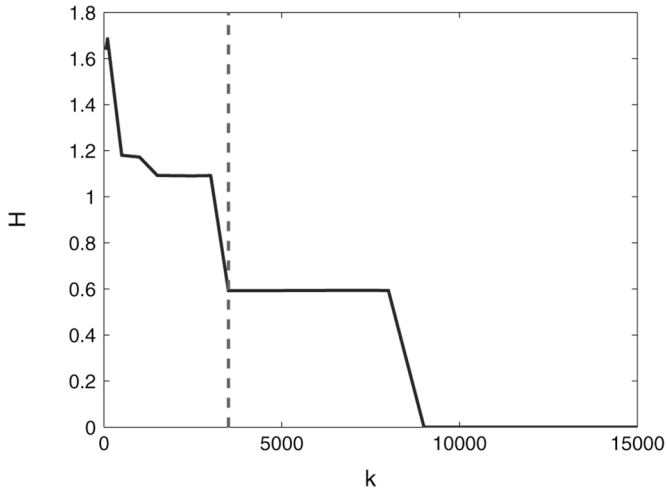
Fig. 14. Clustering based on local intrinsic dimensionality is useful for problems such as this, in which three-dimensional hypersphere (•) is placed "inside" the two-dimensional "swiss roll" (+). Side and front angles of set shown. (a) Side and (b) front.

TABLE I
COMPARISON OF VARIOUS CLUSTERING METHODS ON DATA SET CONSISTING OF "SWISS ROLL" AND THREE-DIMENSIONAL HYPERSPHERE MANIFOLDS. PERFORMANCE REPORTED BASED ON MEAN JACCARD INDEX OVER A 20-FOLD CROSS-VALIDATION

Method	Mean Jaccard
Dimension Estimation	0.7834
K-Means	0.4224
Fuzzy c-means	0.3607

Consider the satellite image of New York City[2] in Fig. 16(a), which has a resolution of 1452×1500. We wish to segment the image into land and water masses. To use local dimension estimation, we define $\boldsymbol{X} = \{x_1, \ldots, x_n\}$, where x_i is a 144-dimensional vector representing a rasterized 12×12 block of the image. After obtaining the local dimension estimates, we apply neighborhood smoothing and recursive entropy estimation as described above. The results, illustrated in Fig. 15(a), lead us to define an ideal neighborhood size of $k = 3500$, which is where

[2]http://newsdesk.si.edu/photos/sites_earth_from_space.htm.



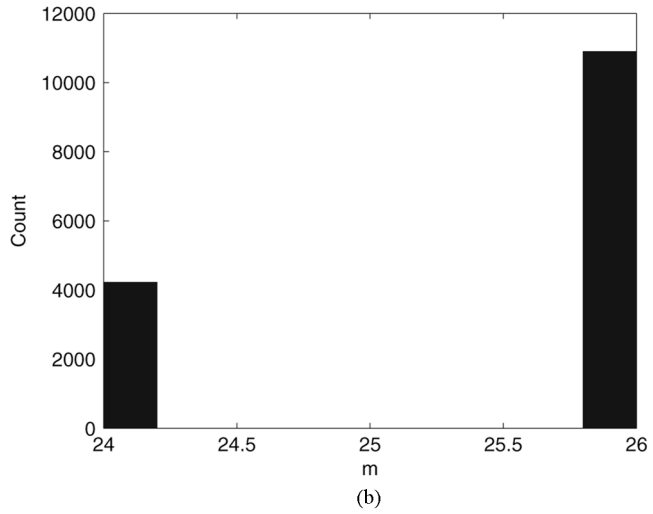


Fig. 15. Plotting the entropy of the dimension estimates suggests a neighborhood size of $k = 3500$, denoted by the dotted line, which yields two significant clusters in the dimension estimates. (a) Entropy versus k and (b) histogram of dimension estimates.

the entropy begins to remain constant for an extended period. This allows us to segment the image into two regions, defined by the complexity estimates shown in Fig. 15(b). The final segmentation can be viewed in Fig. 16(b), where the water is well separated from the land portions of the island of Manhattan and the surrounding boroughs. We note that this image is that of the smoothed local dimension estimates, uniformly scaled to the range [0,255].

We notice there is a relatively low resolution in our segmentation image, due to the large 12×12 blocks used for estimation. We can correct this by using a smaller pixel blocks; however, computational issues prevent us from estimating at much higher resolutions. We can alleviate this problem by estimating at a high resolution only in the areas that require such; this may be determined by using edge detection on the image of local dimension estimates as in Fig. 16(c). In the regions that are determined to contain edges, we resegment at a higher resolution—using 4×4 pixel blocks—with the same recursive entropy estimation process. The results are shown in Fig. 16(d); it is clear that this

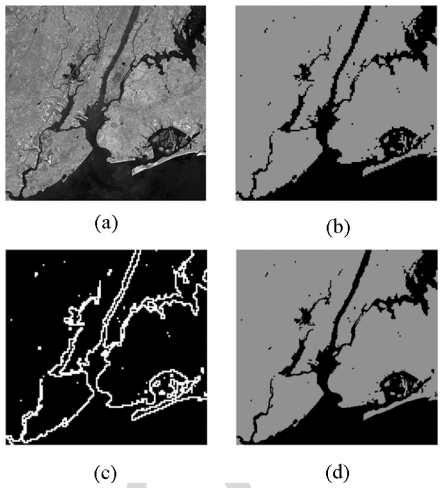


Fig. 16. By using local dimension estimation, neighborhood smoothing, and entropy estimation, we are able to segment the satellite image of New York City into water and land regions. After segmenting the image at a low resolution, we perform edge detection to find the regions that should be analyzed at a higher resolution, yielding a significantly more detailed segmentation. (a) New York City, (b) low-resolution segmentation, (c) edges of segmented image, and (d) high-resolution segmentation.

segmentation appears significantly less digitized and more detailed.

While the previous task was simply to segment water from land in an image, we detailed the "binary" task to demonstrate the process. The problem is easily extended to the multitexture case, which we illustrate in Fig. 17 with images of local dimension estimates scaled to the range [0,255]. In these cases, we segmented images of a sloth bear[3] and a panda bear cub[4] using the same techniques as previously described, only we utilized a high-resolution segmentation over the entire image along with small smoothing neighborhoods. This may give a finer segmentation than required (e.g., the bears are not segmented entirely as one object) but shows the potential segmentation power of local dimension estimation. If a coarser segmentation was desired, larger smoothing neighborhoods may be applied, similar to the previous case of New York City. We note that by no means are we suggesting that dimension alone is a superior means of image segmentation; we simply illustrate that there is a semblance of power to Euclidean dimension when segmenting natural images, and that dimension may be used in conjunction with other means for this complex task.

V. Conclusion

We have shown the ability to use local intrinsic dimension estimation for a myriad of applications. The negative bias in global dimension estimation is strongly influenced by the data depth of the samples on the manifold. By developing a global dimension estimator based on the local dimension estimates of the deepest points, we have shown the issue of the negative bias can be significantly reduced. Typically, dimension estimation is used for the purposes of dimensionality reduction of Riemannian manifolds in Euclidean space, and we have extended this to the

[3]http://newsdesk.si.edu/photos/nzp_sloth_bear.htm.

[4]http://newsdesk.si.edu/photos/nzp_panda_cub.htm.

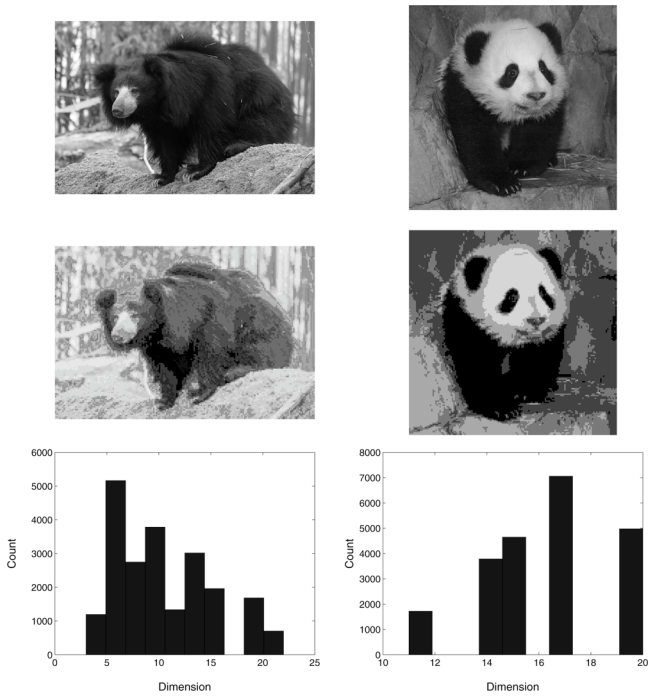


Fig. 17. Segmentation of multitexture images using local dimension estimation and neighborhood smoothing. The first row contains the original images, the second row contains the images of local dimension estimates (scaled to [0,255]), and the third row is the histogram of local dimension estimates.

problem of dimensionality reduction on statistical manifolds, illustrated with the examples of flow cytometry analysis and document classification.

By viewing dimension as a substitute for data complexity, we have applied local dimension estimation to problems that may not naturally be considered. Local dimension estimates can be used to find anomalous activity in router networks, as the overall complexity of the network is decreased when a few sources account for a disproportionate amount of traffic. We have also applied complexity estimation towards the problems of data clustering and image segmentation through the use of neighborhood smoothing. By finding the points in which entropy remains constant as the neighborhood size increases, we are able to optimally cluster the data.

Further analysis into the applications we have presented here is an area for future work. In terms of debiasing global dimension estimation, applying significant weight the interior points in averaging over local dimensions may result in large variance of the dimension estimate due to a small sample size. The bias–variance tradeoff and its optimization is of great importance and should be considered an area for future work. Additionally, we would like to further investigate using Euclidean dimension estimation (as opposed to fractal dimensions) for image segmentation, as we feel this is a very interesting application which has not been thoroughly researched. Specifically, we are interested in combining Euclidean dimension with other measures of textures in order to optimally segment a natural image.

ACKNOWLEDGMENT

The authors would like to thank B. Li from the University of Michigan for isolating the source of the anomalies we discovered in the Abilene data and Dr. W. G. Finn and the Department of Pathology, University of Michigan, for the cytometry data and diagnoses. They thank the reviewers of this paper for their significant contributions.

REFERENCES

[1] K. Fukunaga and D. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vol. C-20, Feb. 1971.
[2] F. Camastra and A. Vinciarelli, "Estimating the intrinsic dimension of data with a fractal-based method," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 24, pp. 1404–1407, Oct. 2002.
[3] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Neural Inf. Process. Syst.*, Vancouver, CA, Dec. 2002.
[4] J. Costa and A. O. Hero, *Statistics and Analysis of Shapes*. Cambridge, MA: Birkhauser, 2006, ch. Learning Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces, pp. 231–252.
[5] E. Levina and P. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Neural Inf. Process. Syst.*, Vancouver, CA, Dec. 2004.
[6] K. M. Carter, A. O. Hero, and R. Raich, "De-biasing for intrinsic dimension estimation," in *Proc. IEEE Statist. Signal Process. Workshop*, Aug. 2007, pp. 601–605.
[7] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, vol. 14.
[9] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 1, pp. 2323–2326, 2000.
[10] A. P. Petland, "Fractal-based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 661–674, 1984.
[11] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 72–77, Jan. 1995.
[12] B. B. Mandelbrot, *The Fractal Geometry of Nature*. San Francisco, CA: Freeman, 1982.
[13] K. M. Carter and A. O. Hero, "Variance reduction with neighborhood smoothing for local dimension estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2008, pp. 3917–3920.
[14] J. A. Costa and A. O. Hero, "Geodesic entropic graphs for dimension and entropy estimation in manifold learning," *IEEE Trans. Signal Process.*, vol. 52, pp. 2210–2221, Aug. 2004.
[15] F. Camastra, "Data dimensionality estimation methods: A survey," *Pattern Recognit.*, vol. 36, no. 12, pp. 2945–2954, 2003.
[16] V. I. Koltchinskii, *Empirical Geometry of Multivariate Data: A Deconvolution Approach*, vol. 28, no. 2, pp. 591–629, 2000.
[17] V. Pestov, "An axiomatic approach to intrinsic dimension of a dataset," *Neural Netw.*, vol. 21, no. 2–3, pp. 204–213, 2007.
[18] N. Tatti, T. Mielikainen, A. Gionis, and H. Mannila, "What is the dimension of your binary data?," in *Proc. 6th Int. Conf. Data Mining*, Hong Kong, 2006, pp. 603–612.
[19] M. Hein and J. Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in R^d," in *Proc. 22nd Int. Conf. Machine Learn.*, 2005, pp. 289–296.
[20] M. Raginski and S. Lazebnik, "Estimation of intrinsic dimensionality using high-rate vector quantization," in *Proc. 19th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 1105–1112.
[21] S. N. Lahiri, *Resampling Methods for Dependent Data*. New York: Springer, 2003.
[22] L. Breiman, "Bagging predictors," *Machine Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[23] J. H. Friedman and B. E. Popescu, Predictive learning via rule ensembles Stanford Univ., Tech. Rep., 2005.
[24] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *Proc. 7th Int. Conf. Database Theory*, Jerusalem, Israel, Jan. 1999, pp. 217–235.
[25] Y. Vardi and C.-H. Zhang, "The multivariate $L1$-median and associated data depth," in *Proc. Nat. Acad. Sci. USA*, 2000, vol. 97, pp. 1423–1426.

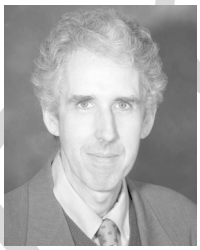
[26] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero, "Fine: Fisher information non-parametric embedding," *IEEE Trans. Pattern Anal. Machine Intell.* 2009 [Online]. Available: http://tbayes.eecs.umich.edu/kmcarter/papers/tpami_fine.pdf, to appear
[27] R. Kass and P. Vos, *Geometrical Foundations of Asymptotic Inference*, ser. Wiley Series in Probability and Statistics. New York: Wiley, 1997.
[28] W. G. Finn, K. M. Carter, R. Raich, and A. O. Hero, "Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: Treating flow cytometry data as high-dimensional objects," *Cytometry B, Clin. Cytometry*, vol. 76B, no. 1, pp. 1–7, Jan. 2009.
[29] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero, "Information preserving component analysis: Data projections for flow cytometry analysis," *IEEE J. Sel. Topics Signal Process. (Special Issue on Digital Image Processing Techniques for Oncology)*, vol. 3, no. 1, pp. 148–158, Feb. 2009.
[30] K. M. Carter, "Dimensionality reduction on statistical manifolds," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Jan. 2009.
[31] S. Grikshchat, J. A. Costa, A. O. Hero, and O. Michel, "Dual rooted-diffusions for clustering and classification on manifolds," in *Proc. 2006 IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2006, vol. 5.
[32] J. C. Bezdec, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[33] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[34] P. Jaccard, "The distribution of flora in the alpine zone," *New Phytologist*, vol. 11, pp. 37–50, 1912.

Kevin M. Carter (S'08) received the B.Eng. degree *(cum laude)* in computer engineering from the University of Delaware, Newark, in 2004. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2006 and 2009, respectively.

He is now a Member of Technical Staff at MIT Lincoln Laboratory, working on problems of network security and anomaly detection. His main research interests lies in manifold learning, with specific focus on statistical manifolds, information geometric approaches to dimensionality reduction, and intrinsic dimension estimation. His additional research interests include statistical signal processing, machine learning, and pattern recognition.

Raviv Raich (S'98–M'04) received the B.Sc. and M.Sc. degrees from Tel Aviv University, Tel Aviv, Israel, in 1994 and 1998, respectively, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 2004, all in electrical engineering.

Between 1999 and 2000, he was a Researcher with the Communications Team, Industrial Research, Ltd., Wellington, New Zealand. From 2004 to 2007, he was a Postdoctoral Fellow with the University of Michigan, Ann Arbor. Since fall 2007, he has been an Assistant Professor in the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis. His main research interest is in statistical signal processing, with specific focus on manifold learning, sparse signal reconstruction, and adaptive sensing. His other research interests lie in the area of statistical signal processing for communications, estimation and detection theory, and machine learning.



Alfred O. Hero III (S'79–M'84–SM'96–F'98) received the B.S. degree *(summa cum laude)* from Boston University, Boston, MA, in 1980 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1984, both in electrical engineering.

Since 1984, he has been with the University of Michigan, Ann Arbor, where he is a Professor in the Department of Electrical Engineering and Computer Science and, by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. His recent research interests have been in areas, including inference in sensor networks, adaptive sensing, bioinformatics, inverse problems, and statistical signal and image processing.

He received an IEEE Signal Processing Society Meritorious Service Award (1998), an IEEE Signal Processing Society Best Paper Award (1998), and the IEEE Third Millenium Medal (2000). He was President of the IEEE Signal Processing Society (2006–2008) and is Director-Elect of IEEE for Division IX (2009).