Exploring the Use of a Self-tuning Diffusion Map Framework

Friedenberg, David
Statistics and Information Analysis Group
Battelle Memorial Institute
Columbus, OH 43201 USA
friedenbergd@battelle.org

Nugent, Rebecca
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213 USA
rnugent@stat.cmu.edu

Introduction

Dimensionality reduction is an important step in being able to process and understand large, high-dimensional data. With data acquisition and storage capabilities skyrocketing, the ability to quickly and precisely extract meaningful structure while discarding extraneous information is often imperative before any type of inference can be made from the data. In this paper, we introduce a novel method for automatically extracting a low-dimensional representation of complicated, often non-linear structure from high-dimensional datasets of varying local density. *Self-tuning diffusion maps* extend the previous literature on diffusion maps, a dimensionality reduction tool, to adapt to fluctuating local densities in the data. This improvement allows for better detection of structure in the presence of significant density variations, irregular sampling, or possible hierarchical structure. We demonstrate the power of this method with several example datasets that are difficult to analyze using standard diffusion maps and show that the self-tuning diffusion map is better able to disentangle the complicated structure. Once we have calculated the STDM, standard methods can be used to make accurate inferences in the reduced space.

In the next section, we first review diffusion maps and their subsequent dimensionality reduction of the data and then introduce the difficulty of choosing good tuning parameters. We then introduce the self-tuning diffusion map and possible parameter choices while providing some insights on its advantages. We next demonstrate the method using examples from clustering, classification, and regression and a cross-validation algorithm is suggested for choosing the self-tuning parameters in supervised learning problems. We conclude with a summary of the advantages of the self-tuning diffusion map and directions for future work.

Diffusion Maps

In this section we provide a brief overview of diffusion maps, which have been successfully used for tasks such as data parameterization (e.g. [1]), regression (e.g. [2]), and high-dimensional density estimation (e.g. [3]). Diffusion maps aim to measure the "connectivity" of a dataset and project data into diffusion space, in which the Euclidean distance between two observations is small if the observations are highly connected in the original feature space and large otherwise. Assume we have n observations each with p attributes. We define a weighted graph on the data where each observation

is a node and the edge between two observations x and y is defined as

$$w(x,y) = \exp(-d(x,y)^2/\epsilon) \tag{1}$$

where $d(x,y)$ is the application-dependent dissimilarity between x and y, often just Euclidean distance. Next, we construct a fictive Markov random walk ([4]) over the weighted graph where the transition probability of going from x to y in one-step is $p_1(x,y) = w(x,y)/\sum_z w(x,z)$. From this, we then construct the transition matrix P where $P_{ij} = p_1(i,j)$. The transition probabilities will be close to zero unless the two observations are similar. This transition matrix construction is also used in several spectral clustering algorithms. While there is a strong connection between spectral clustering and diffusion maps ([5],[6]), diffusion maps can be used for a wide variety of applications and are more flexible in the choice of dimensionality (more details later). The tuning parameter ϵ controls how quickly $w(x,y)$ decays to zero and needs to be selected either automatically (e.g. cross-validation) or by the user. A proposed default ([7]) is to take the median distance to the Kth nearest neighbor in the dataset, i.e.,

$$\epsilon(K) = \text{median}(d(x,x_K)) \tag{2}$$

where K is a small fraction of n, usually around 1%. For notational brevity, we will often simply refer to this as ϵ. The parameter ϵ could also be determined by some other measure or optimized over a grid of possible values.

Since P is the transition matrix for a Markov random walk, it is straightforward to calculate the transition probability after t steps using entries from the matrix P^t. We can also calculate the stationary distribution of the random walk, denoted $\phi_0(.)$. We use these quantities to define the *diffusion distance* between two observations at the scale t:

$$D_t^2(x,y) = \sum_z \frac{(p_t(x,z) - p_t(y,z))^2}{\phi_0(z)} \tag{3}$$

This distance calls two observations close if their t-step conditional distributions are close. The diffusion map at scale t projects the data into m-dimensional diffusion space such that the Euclidean distance between two observations in diffusion space approximates their diffusion distance in the original coordinate system. The mapping is defined as follows:

$$\Psi_t : x \rightarrow [\lambda_1^t\psi_1(x), \lambda_2^t\psi_2(x), \ldots, \lambda_m^t\psi_m(x)] \tag{4}$$

with λ_j and ψ_j the eigenvalues and right eigenvectors respectively of P. In this paper we use the multi-scale diffusion map as defined in [8]. The multi-scale diffusion map simultaneously considers all possible paths between observations across all time scales t making it more robust to structure at different time scales. To do this, we replace λ_j^t with $\sum_{t=1}^{\infty}\lambda_j^t = \lambda_j/(1-\lambda_j)$ which we call the jth *eigenmultiplier*. Additionally, the use of the multi-scale diffusion map eliminates having to choose t. In practice, we still need to choose m, the number of diffusion dimensions (with an upper bound of n). Where appropriate, m is chosen using cross-validation or some other measure of risk. In the absence of such a measure, m can be chosen by examining the drop-off in the eigenmultipliers, similar to choosing the number of principal components based on a scree plot. As with principal components, we seek the smallest m that accurately captures the structure of the data. However, diffusion maps are often better able to capture non-linear and complex structure in a way that principal components cannot. For more details on diffusion maps, see [1] and [9].

Self-tuning Diffusion Maps

The global tuning parameter ϵ represents the size of the neighborhood in which two observations are considered similar. A small value of ϵ corresponds to a large similarity neighborhood; increasing ϵ shrinks the neighborhood and decreases the number of pairs of similar observations. This global parameter can be problematic if the structure in the data is dependent on different local densities. For example, in Figure 1, we see four dense regions of observations embedded in a sparser background. The largest distance between two observations in a dense area may be smaller than the distance between an observation in the background and its nearest neighbor. Here it would be difficult to capture the structure of the data using a global ϵ to define a similarity neighborhood irregardless of the region's local density.

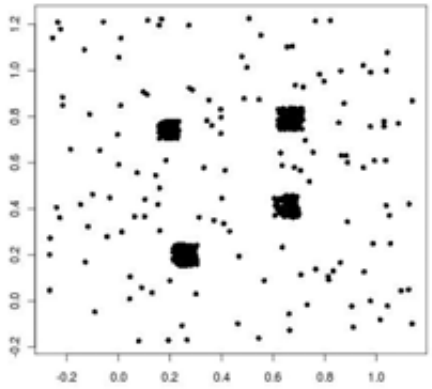

Figure 1: Four very dense groups embedded in a noisy background. Differences in local densities may cause difficulty in capturing the structure using standard diffusion map techniques.

We previously commented on the connections between spectral clustering and diffusion maps; a variant of spectral clustering called self-tuning spectral clustering (STSC, [10]) incorporates local scaling parameters in order to capture group structure with different densities. In STSC, $w(x,y) = exp(-d(x,y)^2/\epsilon)$ is replaced with $w(x,y) = exp(-d(x,y)^2/\epsilon_x\epsilon_y)$ where ϵ_x and ϵ_y represent measures of the local density of observations x and y respectively. In [10], the authors suggest using

$$\epsilon_x(K) = d(x, x_K) \tag{5}$$

where x_K is the Kth nearest neighbor of x (with $K = 7$ as a default). Again, we will often just refer to ϵ_x. The self-tuning approach adjusts to the local density; in dense regions, two observations have to be closer to be considered similar than if they were in a sparse area. For instance, in Figure 1, observations in the dense regions would be assigned smaller ϵ_x values than those in the sparse background. This approach has been successful in spectral clustering and has been proposed for possible use with diffusion maps as well ([11]). In this paper, we explore the self-tuning diffusion map framework (not exclusive to clustering) and comment on possible advantages and disadvantages.

We extend the self-tuning framework to diffusion maps by incorporating the local scale parameters into the dissimilarity matrix. Let

$$d'(x,y) = d(x,y)/\sqrt{\epsilon_x\epsilon_y}. \tag{6}$$

Then

$$w'(x,y) = exp(-d'(x,y)^2/1) = exp(-d(x,y)^2/\epsilon_x\epsilon_y). \quad (7)$$

Using $d'(x,y)$ as our dissimilarity measure and setting the global $\epsilon = 1$ constructs a self-tuning diffusion map (STDM). The STDM eliminates the global ϵ but still leaves the problem of choosing m. Additionally, we need to choose K to calculate $\epsilon_x(K)\ \forall\ x$, the local scaling parameters for all observations. While [10] suggest the form in Equation (5) and setting $K = 7$, we believe this choice will not scale well for datasets of different sizes (among other possible disadvantages). As an alternative, one could choose K to be a small percentage of n (the approach used to select a default ϵ in the `DiffusionMap` package in `R`; [7]). This ϵ_x may be more robust since it will scale with the size of the dataset. In certain applications, it might be desirable to optimize ϵ_x over some measure of risk; we explore this option in subsequent examples. We could also use a completely different measure of local density, for example, a kernel density estimate.

After specifying the form of ϵ_x, we proceed as before, first creating the transition matrix P where $p_1(x,y) = w'(x,y)/\sum_z w'(x,z)$ and then constructing the self-tuning (multi-scale) diffusion map from the eigenvalues and eigenvectors of P. The self-tuning diffusion distance is then defined as in Equation (3) using the dissimilarity metric from Equation (6). The Euclidean distances between the self-tuning diffusion map coordinates will approximate the self-tuning diffusion distance between the observations in the original coordinate system.

Examples

Although motivated by the use of self-tuning parameters in spectral clustering, this paper explores the performance of STDM in a more general framework. We include applications in clustering, classification, and regression. For easy visualization, we present several two-dimensional examples with nonlinear structure and varying local density. We also include a higher dimensional classification problem, the identification of glass fragments via their composition. In all examples, we compare the performance of the self-tuning diffusion map against the standard use of diffusion maps (DM). Euclidean distance is used for $d(x,y)$.

Clustering

We return to the Figure 1 dataset, originally found in [10]. This type of structure can cause problems for standard clustering methods because of both non-standard cluster shapes and fluctuations in local density. We first compute a standard diffusion map using slightly altered defaults (to ensure graph connectedness) in the `DiffusionMap` package in `R` ([7]). Using the 95% dropoff in eigenmultipliers, we choose $m = 7$. We then cluster the observations' diffusion coordinates with k-means (using the correct number of clusters). K-means is a common clustering algorithm that partitions the observations into roughly spherical groups ([12]); k-means has been suggested by [1] as the appropriate clustering technique for use with diffusion coordinates because it minimizes distortion (in a lossy compression framework) in the embedded space. Next we calculate the self-tuning diffusion map using ϵ_x as defined in Equation (5) with $K = 7$ and $m = 5$. The self-tuning diffusion coordinates are then also clustered using k-means (again with the correct k).

Figure 2 compares both the projected diffusion coordinates and the final cluster assignment of standard diffusion maps (top row) and self-tuning diffusion maps (bottom row). The fluctuating

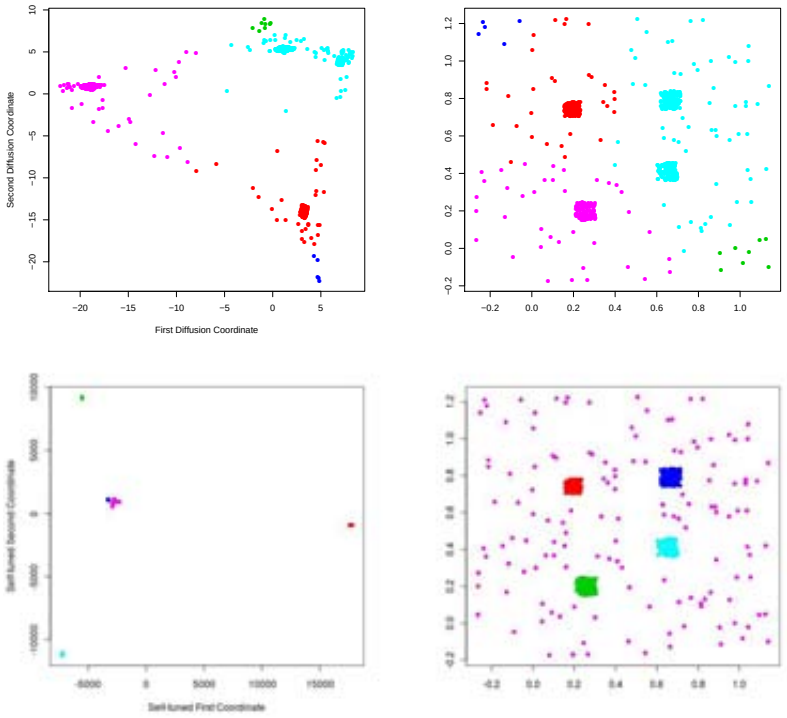

Figure 2: All subfigures colored by cluster assignment: a) DM coordinates; b) DM assignment of original observations; c) STDM coordinates; d) STDM assignment of original observations.

local density causes the standard DM to essentially partition the feature space into sections. Two of the dense regions are grouped; two corners of the background noise are labeled clusters (likely due to slightly larger random separation from the remaining background observations). In contrast, the STDM separates the five groups very well in the projected space (excepting perhaps the background observations and the upper right blue cluster). The final cluster assignments correctly match the structure found in the data. We also reduced the number of diffusion dimensions from $m = 7$ to $m = 5$. Additionally note the separation in the STDM coordinates is such that any standard clustering method would likely recover the true structure; we are not restricted to k-means. In fact, we have seen in practice that hierarchical linkage methods can often give superior performance ([13]).

Previous work ([13]) has shown several advantages to clustering in diffusion coordinates (both standard and self-tuning). One benefit is the separation of the choice of the number of eigenvectors m from the choice of the number of clusters k. Many spectral algorithms (for example, [14]) select m to be equal to k, which itself is often unknown apriori. Often, this choice corresponds to a higher dimensionality than necessary (resulting in a suboptimal dimensionality reduction). In extreme cases, these extra dimensions can adversely affect the quality of the clustering solution. In contrast, using STDMs, we choose the number of dimensions that most succinctly capture the important structure in the data. This choice could be made heuristically (as done above); [13] proposes an automatic method based on prediction strength ([15]). In addition, the resulting dimension reduction can make tasks like choosing the number of clusters k easier via improved visualization and analysis of the reduced space. Furthermore, since STDMs typically separate the clusters more clearly, clustering solutions tend to be less dependent on the choice of algorithm. Often simple algorithms on an STDM are sufficient

to capture the structure (when complicated procedures are necessary on the original coordinates). Further exploration was done on other examples with varying local density from [10]. In all cases, k-means on a DM was largely unable to recover the structure; using an STDM instead resulted in perfect recovery.

In some sense, clustering is a difficult application for the STDM framework. STDMs requires choosing tuning parameters which is not as straightforward when we cannot directly optimize a well-agreed-upon criterion such as a cross-validation score. In addition to the tuning parameters, the user still also needs to select the number of clusters k. Making this choice is outside the scope of this paper; however, we believe the use of STDMs can make this choice easier.

Classification

Next we examine the use of STDMs in a classification setting where the presence of labeled data allows for a more straightforward choice of optimal tuning parameters. One method for choosing the local scaling parameters is cross-validation. We utilize the following algorithm for classifying observations using STDMs.

1. For each $K \in \{1, \dots K^*\}$:
   (a) Compute the STDM_K using $\epsilon_x(K) = d(x, x_K)$ and the m selected by the 95% dropoff in eigenmultipliers.
   (b) Partition the data randomly into R subsets.
   (c) For $r = 1, 2, ..., R$,
      i. Remove subset S_r from STDM_K and train a classifier on the remaining observations.
      ii. Predict the labels for S_r using the classifier from the previous step and compute the misclassification rate M_r.
   (d) Find the average misclassification rate over the subsets: $\bar{M}_R(K) = \frac{1}{R}\sum_{r=1}^{R} M_r$
2. Choose the K with the smallest $\bar{M}_R(K)$ (and corresponding m value).

Alternatively, we could separate the choice of K and m by cross-validating over both parameters (with a significant increase in computational cost). In addition, the algorithm could easily incorporate other choices for ϵ_x.

To demonstrate this algorithm, we return to the example in Figure 1. The four dense region sizes are 116, 111, 150, and 109; we have 136 background observations. We search over a grid of nearest neighbor values for K from 1 to 80 and find that for any K from 3 to 22 (all corresponding to $m = 5$), linear discriminant analysis on the STDM has a zero percent misclassification rate. This large range of optimal K values is likely an artifact of the artificial dataset. We do note, however, that using $K = 1$ (one-near-neighbor) does not work well, likely because it is too sensitive to outliers. As K increases, ϵ_x increases, and we lose the ability to discern fluctuations in the local density.

We now compare performance of DM and STDM for a real-world high-dimensional classification problem. The glass identification dataset ([16]) is comprised of 214 glass samples with nine attributes[1] each, corresponding to the refractive index and the chemical contents of each sample. The dataset was

[1] Each attribute was standardized by subtracting the mean and dividing by the standard deviation.

obtained from the UCI Machine Learning Repository [17]. There are seven different types of glass – two types for building windows, two types for vehicle windows, containers, tableware and headlamps. Being able to differentiate the types is important in practice; glass left at a crime scene can be used as evidence if it can be properly identified. We compare classifying the DM and the STDM. In both cases, cross-validation was used to select the optimal tuning parameter, respectively ϵ and K. The number of eigenvectors m was selected using the 95% dropoff in eigenmultipliers. We again use linear discriminant analysis as our classifier but any number of classification algorithms could be used (given the clearer separation in the projected diffusion space).

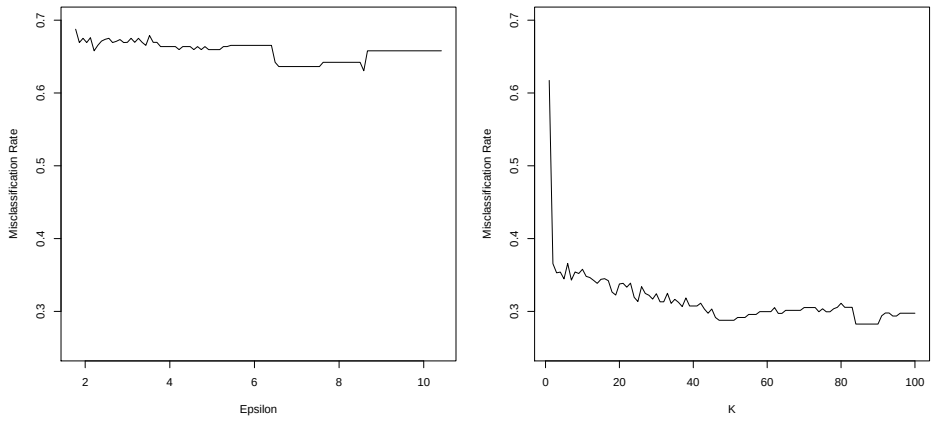

Figure 3: Choosing tuning parameters via four-fold cross validation: a) DM's misclassification rate as a function of ϵ (global); b) STDM's misclassification rate as a function of K (local).

Figure 3 shows the four-fold cross-validation results for both the DM (left) and the STDM (right). The DM misclassification rate remains between 60% and 70% regardless of ϵ value; the STDM misclassification rate is high for $K = 1$ (as expected), and then drops immediately, remaining in the neighborhood of 30% as K increases. Using STDMs with the cross-validation selected local ϵ_x, we achieve an error rate of 28%, far less than the 63% corresponding to the DM with cross-validation selected ϵ. In this example, the STDM is better at differentiating between coarse differences in the types of glass (e.g. building windows and car windows) but also the fine differences within types (e.g. the two different types of building windows). In the classification framework, STDMs seem better able to negotiate the (possibly hierarchical) structure in the data. Although cross-validation can be used to select optimal tuning parameters, our results so far have indicated that performance may not be strongly dependent on the exact choice of $\epsilon_x(K)$; it may be enough just to identify a range of appropriate K values.

Regression

STDMs can also be used to transform predictor variables for regression problems. As in the classification setting, the tuning parameter can be selected using cross-validation. To demonstrate, we examine an irregularly sampled spiral embedded in two dimensions, seen in Figure 4. The response variable is a noisy version of the spiral parameter which traces the path of the spiral. Clearly, a simple linear regression will not be able to capture this structure. Again, we use cross-validation to

choose tuning parameters (ϵ, K) over a grid for DMs and STDMs using the smallest mean-squared error (MSE) as our criterion. The number of diffusion dimensions m was chosen using the 95% eigenmultiplier dropoff (but could also be chosen by cross-validation at an additional computational cost.)

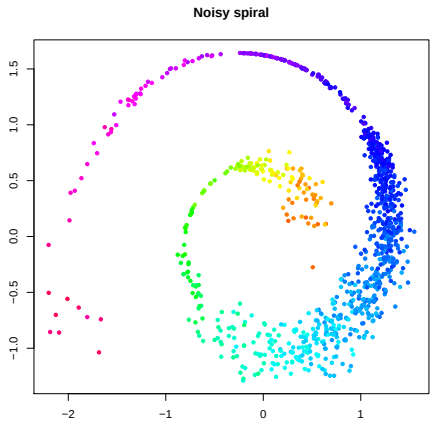

Figure 4: Spiral color represents the intensity of the response; local density varies with the response

Figure 5 shows the cross-validation results for the DM (left) and the STDM (right). For both, the MSE initially drops substantially and then rises as ϵ, K respectively increase. After the initial drop, it appears that the STDM almost always performs better than the DM regardless of K, ϵ values (again indicating that it may not be necessary to identify an exact optimal tuning parameter). The best STDM representation uses $K = 6$ for an MSE of 9.26 which is considerably lower than the best DM MSE (12.50, $\epsilon = 0.0598$). In contrast to the DM, the STDM adapts to the irregular sampling and better traverses the spiral, adjusting to both the sparse and dense regions.

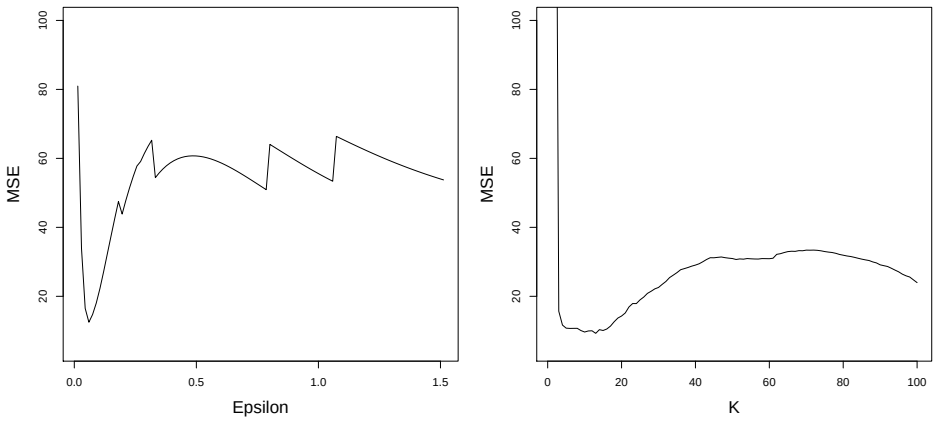

Figure 5: a) Cross-validation selects $\epsilon = .0598$; DM MSE = 12.50 b) Cross-validation selects $K = 6$; STDM MSE = 9.26

Conclusion

In this paper we introduced the self-tuning diffusion map, a flexible variant of standard diffusion maps, which we have shown to be an effective tool for simultaneously reducing the dimensionality of complex datasets while preserving the intrinsic and often complicated structures within the data. STDMs extend the established and successful diffusion map framework to account for local density fluctuations and data structures which may occur at different scales. We demonstrate the power of STDMs using examples from clustering, classification, and regression where standard diffusion maps using a global tuning parameter cannot adapt to local differences in the data. By introducing a series of local parameters, STDMs can more successfully navigate local variations to produce an accurate low-dimensional representation of the underlying data structures. While our examples are limited to clustering, classification and regression, STDMs can be applied to a wide variety of other applications including, but not limited to, high-dimensional density estimation, data parameterization, and image analysis.

We presented results that selected local tuning parameters based on nearest neighbor distances and via cross-validation algorithms in supervised learning settings. We referenced other possible methods for choosing these local parameters including borrowing tools from the adaptive density estimation literature. Future work will explore the performance of different choices. In addition, for unsupervised applications (e.g. clustering) where cross-validation may not be possible, further investigation of appropriate methods to choose the tuning parameters is needed. Regardless, the self-tuning diffusion map appears to potentially be a powerful, flexible addition to the diffusion map framework.

REFERENCES

[1] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9): 1393-1403, 2006.

[2] J.W. Richards, P.E. Freeman, A.B. Lee, and C.M. Schafer. Exploiting Low-Dimensional Structure in Astronomical Spectra. ApJ, 691:32-42, January 2009.

[3] S.M. Buchman, A.B. Lee, and C.M. Schafer. High-dimensional Density Estimation via SCA: An Example in the Modelling of Hurricane Tracks. *ArXiv e-prints*, Jul 2009.

[4] S.M. Ross. *Introductory to Probability Models.* Academic press, December 2006.

[5] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering, and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems 18*, pages 955-962. MIT Press. 2005.

[6] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering, and reaction. In *Applied and Computational Harmonic Analysis: Special issue on Diffusion Maps and Wavelets*, page 2006. 2006.

[7] J.W. Richards. *diffusionMap: Diffusion map*, 2009. R package version 1.0-0.

[8] J.W. Richards, P.E. Freeman, A.B. Lee, and C. M. Schafer. Accurate parameter estimation for star formation history in galaxies using SDSS spectra. MNRAS, 399: 1044-1057, October 2009.

[9] R.R. Coifman and S. Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *App. Comput. Harmon. Anal.*, 21(5): 31-52, July 2006.

[10] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In Lawrence K. Saul, Yair Weiss, and Leon Bottou, editors. *Advances in Neural Information Processing Systems 17*, pages 1601-1608. MIT Press, Cambridge, MA, 2005.

[11] G. Bhat and D.G. Arnold. Diffusion maps and radar data analysis. In E.G. Zelnio and F.D. Garber, editors, *Algorithms for Synthetic Aperture Radar Imagery XIV*, volume 6568, pages 439-464. 2007.

[12] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*, Academic press, 1979.

[13] D. Friedenberg. Adaptive Cluster Detection. *PhD Thesis*, Carnegie Mellon University, 2010.

[14] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849-856. MIT Press, 2001.

[15] R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3): 511-528, September 2005.

[16] I.W. Evett and E.J. Spiehler. Rule induction in forensic science. Technical report, Central Research Establishment, Home Office Forensic Science Service, 1987.

[17] A. Asuncion and D.J. Newman. UCI Machine Learning Repository, 2007.

ABSTRACT

Diffusion maps are a powerful tool for identifying complicated structure and reducing dimensionality in a wide variety of applications. Representing the connectivity of a data set, diffusion maps project observations into a space in which standard methods can more easily model the structure. These maps rely heavily on the choice of a global tuning parameter ϵ that dictates the threshold for similarity. Often, however, using a global tuning parameter does not capture structure that may be a function of fluctuations in the local density. For example, a dense region embedded in background noise would be difficult to capture with standard diffusion maps. We present a flexible self-tuning diffusion map framework that incorporates local tuning parameters to capture this type of structure (if present). We illustrate the self-tuning diffusion framework using examples from clustering, classification, and regression. Where appropriate, a cross-validation algorithm is employed to choose local tuning parameters. Use of the self-tuning diffusion map greatly improves the recovery of structure in the presence of varying local density.