Manifold Learning of 3D Objects

Varsha Pendyala
ee13b1022@iith.ac.in

***Abstract*—Manifold learning helps in uncovering the intrinsic structure of data, thereby bringing the high dimensional data into a low dimensional basis. In this project, intrinsic dimension of data is found through Grassberger-Procaccia (GP) algorithm. Later, LLE is used to embed this data on a low-dimensional subspace. The other important problem of finding the inverse mapping from low dimensional data to high dimensional data is also explored. RBFs have been used to interpolate the inverse mapping using the discrete set of mappings from high to low dimensions which is obtained through LLE.**

***Keywords*—Intrinsic dimension, ambient dimension, non-linear dimensionality reduction, LLE, RBFs**

I. Introduction

Most of the data that we encounter in our day-to-day life in its crude form is high dimensional. There are various problems associated in dealing with high dimensional data. The problems include: Visualization, Storage, Processing time, Curse of dimensionality. So, there is a great necessity in engineering the ways to represent this data using low dimensional basis. It starts with answering the following questions: Can we always find a low dimensional representation for any kind of data? If it is possible, how much lower is that dimensionality compared to that in unprocessed representation? The answer to the first question is true in most of the cases. Because, any system that generates the data follows certain model and hence we can always bring this data into such a representation where it is purely governed by the degrees of freedom(dof) that the system generating it allows. Or in other words, the data has an intrinsic dimensionality which may be different from ambient dimensionality (the no. of dimensions in raw data representation).

Manifold learning deals with learning this intrinsic structure of the data in order to circumvent above mentioned problems associated with large number of dimensions. There are various approaches for intrinsic dimension (ID) estimation from the data itself. In this project, Grassberger-Procaccia (GP) algorithm based on correlation dimension has been used. This ID serves as the input parameter for many manifold learning algorithms. In this implementation, Local Linear Embedding (LLE) approach has been used for manifold learning.

The other side of the problem that bears great significance is, finding the inverse mapping from low dimensional data to high dimensional data. The solution to this problem will enable us to do all the tasks of data processing in low dimension and get back to higher dimension only for human consumption. Data processing in low dimensions is not only profitable in terms of memory usage, but is also advantageous in terms of mitigating the curse of dimensionality. In addition, when we learn the low dimensional manifold, smoothing over the data is achieved. Hence, finding the inverse map lets us compute the denoised high dimensional data. In this project, RBFs have been used to interpolate the inverse mapping.

II. Intrinsic Dimension

In mathematical terms, few methods describe this intrinsic dimensionality as the topological dimension. Topological dimension is the basis dimension of the local linear approximation of hypersurface on which the data resides.

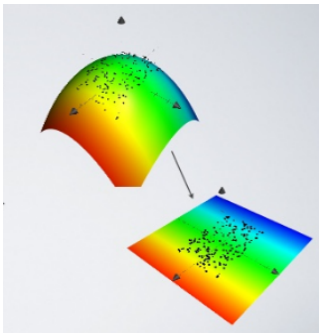

Fig. 1.

In Fig 1, though we see that the first surface has 3D representation, we can find its equivalent 2D representation by preserving the local neighbourhood. Here, the dimension of local linear approximation of hypersurface on which the data resides is two.

Ideally, intrinsic dimension(ID) of the data equals dof of the system that is generating this data. However, the system may not have been modelled in all the applications and hence guessing ID from the problem statement alone is not possible. In such cases, we resort to certain mathematical approaches that estimates ID from the data itself. For example, in PCA eigen value decomposition of covariance matrix of the data (in linear case) is performed to find the number of dominant eigen values which equals the ID of the data. Similarly, we have several approaches to estimate ID of data in non-linear case as well. Local methods estimate ID by preserving the properties of local neighborhood of data points. In one of the approaches, tessellation of the data space is performed to do PCA locally, so that intrinsic structure of data is not corrupted. Also, neural networks called Topology Representing Networks (TRNs) are used to find the reference vectors which closely reflect the structure of input data distribution, thereby enabling to estimate ID. There are global methods that include Kernal PCA, MDS methods which tend to preserve as much as possible the distances or the order of distances among the data points.

In this project, Grassberger-Procaccia (GP) algorithm has been used for ID estimation. This allows to estimate the correlation dimension which belongs to a dimension family called fractal. Let Ω be a set of points X of cardinality N. If the correlation integral C_m(r) is defined as:

$$C_m(r) = \lim_{N\to\infty} \frac{2}{N(N-1)} \sum_{1\le i<j\le N} I(||X_j - X_i|| \le r) \quad (1)$$

where I is an indicator function which holds 1 iff condition holds, 0 otherwise. Then the correlation dimension D of Ω is equal to:

$$D = \lim_{r\to 0} \frac{ln(C_m(r))}{ln(r)} \quad (2)$$

D: correlation dimension, is found to be close to intrinsic dimension of the data. However, it has been proved that in order to get an accurate estimate of the dimension D, the set cardinality N has to satisfy the following inequality:

$$D < 2log_{10}N \quad (3)$$

This inequality implies that the number of points N, necessary to get accurate estimate grows exponentially when the dimension to estimate increases. Hence, an empirical approach which works with small cardinality sets is implemented.

Consider the set Ω whose intrinsic dimension has to be estimated. The procedure has following steps:

- Another set $\Omega^{'}$ with same cardinality N is generated from uniformly distributed data points from a d-dimensional hypercube. We assume ID of $\Omega^{'}$ to be d.
- The correlation dimension(D) is measured by GP algorithm.
- Previous steps are repeated for T different values of d, thus obtaining C=(Di,di) , i=1,2,...T
- The best fitting of points of C by non-linear functions is performed. (Thus we create a look up table for a particular N)
- Finally the correlation dimension of Ω is computed, and by using the curve-fitting, intrinsic dimension of Ω is estimated.

III. Local Linear Embedding

LLE uses local linearity property of the data points to find their low dimensional representation. Like PCA and MDS, this algorithm is simple to implement, and its optimization do not involve local minima. At the same time, however, it is capable of generating highly nonlinear embeddings. The LLE algorithm, summarized in Fig.2, is based on simple geometric intuitions. Suppose the data consist of N real-valued vectors $\vec{X}_i$, each of dimensionality D, sampled from some underlying manifold. Provided there is sufficient data (such that the manifold is well-sampled), we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point

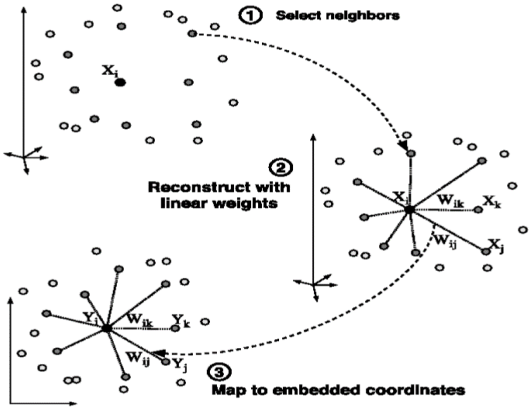

Fig. 2. LLE

from its neighbors. Reconstruction errors are measured by the cost function:

$$\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij}\vec{X}_j|^2 \quad (4)$$

To compute the weights W_{ij}, we minimize the cost function subject to two constraints: first, that each data point $\vec{X}_i$ is reconstructed only from its neighbors, enforcing W_{ij} = 0 if $\vec{X}_j$ does not belong to the set of neighbors of $\vec{X}_i$; second, that the rows of the weight matrix sum to one:$\sum_j W_{ij}$ = 1. The optimal weights W_{ij} subject to these constraints are found by solving a least-squares problem. By design, the reconstruction weights W_{ij} reflect intrinsic geometric properties of the data that are invariant to rotations, rescalings, and translations. We therefore expect their characterization of local geometry in the original data space to be equally valid for local patches on the manifold. In particular, the same weights W_{ij} that reconstruct the ith data point in D dimensions should also reconstruct its embedded manifold coordinates in d dimensions. In the final step of the algorithm, each high-dimensional observation $\vec{X}_i$ is mapped to a low-dimensional vector $\vec{Y}_i$ representing global internal coordinates on the manifold. This is done by choosing d-dimensional coordinates $\vec{Y}_i$ to minimize the embedding cost function:

$$\Phi(Y) = \sum_i |\vec{Y}_i - \sum_j W_{ij}\vec{Y}_j|^2 \quad (5)$$

The embedding cost in Eq. 5 defines a quadratic form in the vectors $\vec{Y}_i$. Subject to constraints that make the problem well-posed, it can be minimized by solving a sparse N X N eigenvalue problem, whose bottom d nonzero eigen vectors provide an ordered set of orthogonal coordinates centered on the origin.

IV. Inverse Mapping

We consider a finite set of n data points x(1),...,x(n) $\subset R^D$ that lie on a bounded low-dimensional smooth manifold M $\subset$

R^D, and we assume that a nonlinear mapping has been defined for each point $x^{(i)}$,

$$\Phi_n : M \subset R^D \rightarrow R^d \tag{6}$$

$$x^{(i)} \longmapsto y^{(i)} = \Phi_n(x^{(i)}), i = 1, 2, ..., n \tag{7}$$

However Φ_n^{-1} is only defined on the existing data. The goal is to generate a numerical extension of Φ_n^{-1} to all of $\Phi(M) \subset R^d$. We interpolate each coordinate function, ϕ_i^{-1}, i=1,...,D independently of each other. Here we use RBFs to construct these inverse mappings. For each coordinate function ϕ_i^{-1}, we define $\phi_i^{\dagger}$ to be the RBF interpolant to the data $(y^{(j)}, x^{(j)})$:

$$for\ all\ y \in \Phi(M), \Phi_i^{\dagger}(y) = \sum_{j=1}^{n} \alpha_i^{(j)} k(y, y^{(j)}) \tag{8}$$

The function k in (8) is the kernel that defines the radial basis functions, k(z,w)=g($||z - w||$). The weights, $\alpha_i^{(1)}, ..., \alpha_i^{(n)}$, are determined by imposing the fact that the interpolant be exact at the nodes $y^{(1)}$,...,$y^{(n)}$, and thus are given by the solution of the linear system:

$$\begin{bmatrix} k(y^{(1)}, y^{(1)}) & \dots & k(y^{(1)}, y^{(n)}) \\ \vdots & \ddots & \vdots \\ k(y^{(n)}, y^{(1)}) & \dots & k(y^{(n)}, y^{(n)}) \end{bmatrix} \begin{bmatrix} \alpha_i^{(1)} \\ \vdots \\ \alpha_i^{(n)} \end{bmatrix} = \begin{bmatrix} x_i^{(1)} \\ \vdots \\ x_i^{(n)} \end{bmatrix} \tag{9}$$

We can combine the D linear systems (9) by concatenating all the coordinates in the right-hand side of (9), and the corresponding unknown weights on the left-hand side of (9) to form the system of equations:

$$\begin{bmatrix} k(y^{(1)}, y^{(1)}) & \dots & k(y^{(1)}, y^{(n)}) \\ \vdots & \ddots & \vdots \\ k(y^{(n)}, y^{(1)}) & \dots & k(y^{(n)}, y^{(n)}) \end{bmatrix} \begin{bmatrix} \alpha_i^{(1)} & & \alpha_D^{(1)} \\ \vdots & \dots & \vdots \\ \alpha_i^{(n)} & & \alpha_D^{(n)} \end{bmatrix} = \begin{bmatrix} x_i^{(1)} & & x_D^{(1)} \\ \vdots & \dots & \vdots \\ x_i^{(n)} & & x_D^{(n)} \end{bmatrix} \tag{10}$$

which takes the form KA = X, where $K_{i,j}$ = k($y^{(i)}, y^{(j)}$), $A_{i,j}$=$\alpha_j^{(i)}$, and $X_{i,j}$ = $x_j^{(i)}$. Let us define the vector k(y,) = [k(y,$y^{(1)}$)...k(y,$y^{(n)}$)]^T. The approximate inverse at a point y $\in \Phi(M)$ is given by:

$$\Phi^{\dagger}(y)^T = k(y,)^T A = k(y,)^T K^{-1} X \tag{11}$$

V. Experiments and Results

A. DataSet

A 3d object in nurbs representation is considered. To learn the manifold of this object, we need to have slightly distorted versions of the same object. For this, the property of affine invariance of nurbs is utilized. It implies, we can perform affine transformation of the object by applying the same on its control points. Thus, each version of the 3d object is given by an affine transformation over original control points. N=1000 affine transformations of the object is created.

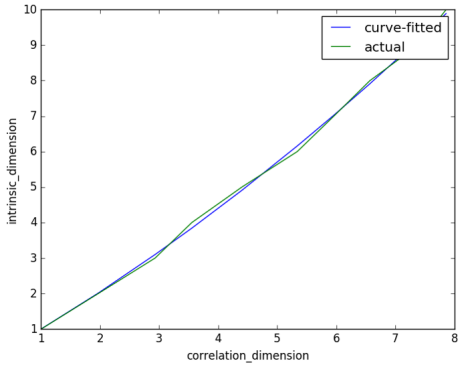

Fig. 3.

B. Intrinsic dimension estimation

The intrinsic dimension of this data generated is estimated using Grassberger-Procaccia (GP) algorithm. The empirical approach described in the section II is carried out with N=1000 data points. Curve fitting between correlation dimension estimate and intrinsic dimension is done using quadratic polynomial. This is done for d=1,2,...,10. To validate the efficiency of this approach, it is tested with hypercube data of different dimensions. Hence, ideally d should be equal to D. Fig 4 gives the results from GP algorithm.

D	Correlation_dimension	Intrinsic_dimension = d
11	8.67	11.21
13	9.8	13.16
15	10.68	14.7
17	11.5	16.2
19	12.79	18.7
23	14.48	22.29
26	15.5	24.6
28	16.17	26
30	16.3	26.4

Fig. 4.

Now, this approach is used on the 3D data that is generated through affine transformations. Hence an intrinsic dimension of 9 is expected. While the correlation dimension estimate of the 3D data generated is found to be 6.59, the corresponding intrinsic dimension estimate is 7.9 (through curve-fitting).

C. Manifold learning

After getting the ID estimate, LLE is used to get the low dimensional representation of the data. The number of neighbors considered in LLE is 6. However, we will see that LLE is performed with various values of d in order to evaluate the efficiency of reconstruction.

D. Inverse Mapping

Of the 1000 data points, 900 are used to estimate the interpolation coefficients for inverse mapping through RBFs with cubic kernel. Once, the interpolation coefficients are

calculated, remaining 100 low dimensional data points are used to find their corresponding high-dimensional representation. Since, we also have the actual high-dimensional representation of these data points, efficiency of reconstruction can be found. For each data point $y^{(i)} \in R^d$ in this set of cardinality 100, error_measure(i) = $\frac{mean((S(i)-\hat{S}(i))^2)}{mean(S(i))}$ is computed, where $S(i)$, $\hat{S}(i)$ are actual and estimated surfaces respectively. Finally, mean(error_measure) is computed for various values of d. Fig 5 shows the mean error measures when reconstruction is done with different ds.

d	Error measure
10	0.28
15	0.19
20	0.21
25	0.16
30	0.13
40	0.11

Fig. 5.

VI. Conclusion

Manifold learning approach called LLE is used to get the low dimensional representation of 3d objects. Later, these mappings from high to low dimensional data is put to use in estimating the interpolation matrix required for inverse mapping estimation through RBFs. Finally, the efficiency of the method is tested by dividing the data into two sets of cardinality N1=900 and N2=100. The set with N1 points is used to estimate the interpolation function, which is later validated through an error measure by means of set with N2 points. The above error measures are obtained when no restriction is put on the affine transformed surface w.r.t to the original surface. However, in scenarios where we have more controlled 3d in real settings, the error is expected to decline greatly.

References

[1] Inverting non linear dimensionality reduction with scale-free radial basis function interpolation. Nathan D.Monnig, Bengt Fornberg, Franois G.Meyer

[2] Nonlinear Dimensionality Reduction by Locally Linear Embedding. Sam T. Roweis and Lawrence K. Saul

[3] Intrinsic Dimension Estimation of Data: AN Approach Based on Grassberger-Procaccia's Algorithm. FRANCESCO CAMASTRA and ALESSANDRO VINCIARELLI

[4] Data Dimensionality Estimation Methods: A survey. Francesco Camastra

[5] CHAPTER 2 Basic properties of splines and B-splines - UiO