Multiple Manifolds Learning Framework Based on Hierarchical Mixture Density Model

Xiaoxia Wang, Peter Tiňo, and Mark A. Fardal

School of Computer Science, University of Birmingham, UK
Dept. of Astronomy, University of Massachusetts, USA
{X.Wang.1,P.Tino}@cs.bham.ac.uk, fardal@fcrao1.astro.umass.edu

Abstract. Several manifold learning techniques have been developed to learn, given a data, a single lower dimensional manifold providing a compact representation of the original data. However, for complex data sets containing multiple manifolds of possibly of different dimensionalities, it is unlikely that the existing manifold learning approaches can discover all the interesting lower-dimensional structures. We therefore introduce a hierarchical manifolds learning framework to discover a variety of the underlying low dimensional structures. The framework is based on hierarchical mixture latent variable model, in which each submodel is a latent variable model capturing a single manifold. We propose a novel multiple manifold approximation strategy used for the initialization of our hierarchical model. The technique is first verified on artificial data with mixed $1-$, $2-$ and $3-$dimensional structures. It is then used to automatically detect lower-dimensional structures in disrupted satellite galaxies.

1 Introduction

In the current machine learning literature, the manifold learning has been predominantly understood as learning (a single) underlying low-dimensional manifold embedded in a high dimensional data space, where the data points are assumed to be aligned (up to some noise) along the manifold. Typical representatives of such approaches are principal component analysis (PCA)[1], self-organizing mapping (SOM)[2], locally linear embedding (LLE)[3] and Isomap [4]. In these methods, intrinsic (manifold) dimension d is either treated as a prior knowledge given by user or as a parameter to be estimated. Estimating the intrinsic dimension of a data set (without the structure learning considerations) is discussed e.g. in [5] [6], [7]. To our best knowledge, there has been no systematic work on dealing with situations when the data is aligned along multiple manifolds of various dimensionalities, potentially corrupted by noise.

In this paper we propose a a framework for learning multiple manifolds. With each manifold we associate a probability density (generative model), so that the collection of manifolds can be represented by a mixture of their associated density models. These generative models are formulated as latent variable model along the lines of [8] or [9]. Our proposed approach consists of several steps. First,

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 566–581, 2008.

we filter data points according to the intrinsic dimensionality of the local manifold patch they are likely to belong to (modulo some manifold aligned "noise"). Then we detect multiple manifolds in each such dimension-filtered set. Finally, we construct a hierarchical probabilistic model containing density models of the detected noisy manifolds. We illustrate our framework on learning multiple manifolds with dimension $d = 1$ and $d = 2$ embedded in a 3-dimensional space.

The paper is organized as follows: the next section briefly reviews related work on manifold learning, intrinsic dimension estimation and hierarchical latent variable modeling; Section 3 describes our framework of learning multiple manifolds based on probability density modeling; Section 4 contains experimental results on artificial data and a data set produced by realistic galaxy collision models. Finally, section 5 concludes the paper and discusses the directions of our future work.

2 Related Work

Some manifold learning algorithms are designed to find a single function $\mathbf{y} = g(\mathbf{x}, \mathbf{U})$ representing the mapping between high dimensional observation $\mathbf{x}$ and its low dimensional representation $\mathbf{y}$. Principal component analysis (PCA) implements the transformation function by linear projection $\mathbf{Ux}$. The d orthonormal principal axes vectors $\mathbf{u}_i$ in the observation (data) space form the matrix $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d)$. In contrast, the Generative Topographic Mapping (GTM) [9] is a probabilistic reformulation of the self-Organizing Map (SOM) [2]. It represents the non-linear mapping from a low-dimensional latent space to the high-dimensional data space as a generalized linear regression model $\mathbf{W}\Phi(\mathbf{y})$, where $\Phi(\mathbf{y})$ consists of M fixed basis functions $\{\phi_j(\mathbf{y}), j = 1, \ldots, M\}$, $\mathbf{W}$ is $D \times M$ weights matrix of outputs of basis functions (D is the dimensionality of the data space). A probabilistic generative model of PCA called probabilistic principal component analysis (PPCA) was also proposed in [8]. Other approaches, like locally linear embedding (LLE) [3], Isomap [4] and Laplacian eigenmaps [10], learn the embedding without formulating an explicit mapping. LLE and Laplacian eigenmaps compute the low dimensional representation preserving the local neighborhood structure in data space. Isomap applies multidimensional scaling (MDS) to estimated geodesic distance between points. To generalise the results of LLE, Saul and Roweis proposed in [11] a probabilistic model for the joint distribution $p(\mathbf{x}, \mathbf{y})$ over the input and embedding spaces, which also provides a way to generalise the results from Isomap or Laplacian eigenmaps.

As in the case of manifold learning, most intrinsic dimensionality estimators assume that all the data points are aligned along a single 'manifold'. In [6], local PCA is applied to each node of the optimal topology preserving map (OPMT), intrinsic dimension is the average over the number of eigenvalues which approximates the intrinsic dimensionality at data clusters. Levina and Bickel [5] also average the estimated dimension over all observation. A point level dimensionality estimator proposed in [7] is appealing due to the ability to deal with manifolds

of different dimensionality. The authors first represent data point by a second order, symmetric, non-negative definite tensor, whose eigenvalues and eigenvectors fully describe the local dimensionality and orientation at each point. A voting procedure accumulates votes from its neighbors and provides an estimate of local dimensionality.

Finally, we review some examples of hierarchical model and its structure estimation strategy. To reveal the interesting local structures in a complex data set, a hierarchical visualization algorithm based on a hierarchical mixture of latent variable models is proposed in [12]. The complete data set is visualized at the top level with clusters and subclusters of data points visualized at deeper level. Tino and Nabney [13] extended this visualization by replacing the latent variable model by GTM (generative topographic mapping) so that the nonlinear projection manifolds could be visualized. Structures of the hierarchy in these visualization systems are built interactively. A non-interactive hierarchy construction was proposed in [14].

3 Multiple Manifolds Learning Framework

In this section, a multiple manifolds learning framework is proposed to learn from the dataset of points aligned along different manifolds of different dimensionalities, as shown in figure 1. Although the methods presented in the previous section could easily learn manifolds from either the first or the second set, no methods have been developed for learning their mixture. To identify these manifolds, we **(1)** cluster them by their intrinsic dimensions d; **(2)** use the data with same intrinsic dimension to discover and construct the d dimensional surfaces by the multi-manifolds learning algorithm presented later, then initialize a latent variable model for each manifold, **(3)** build a hierarchical mixture model consisting of the generative model for manifolds.

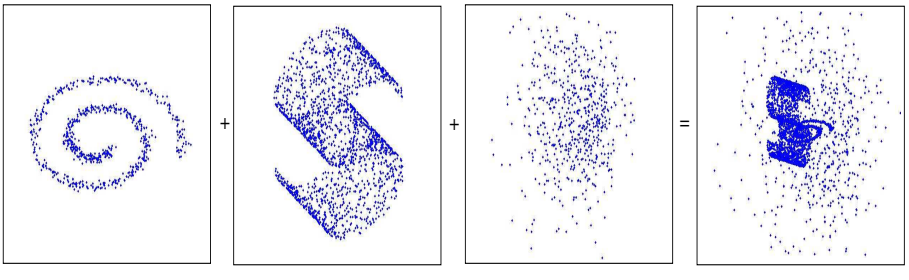


Fig. 1. Multiple manifolds example: 700 3D points aligned along a 1D manifold, 2000 3D points from lying on a 2D manifold and 600 3D points generated form a mixture of 3 Gaussians

3.1 Intrinsic Dimension Estimation

In the first step, we estimate each point's intrinsic dimension and cluster the entire dataset according to the intrinsic dimensions found. The intrinsic dimension of a point is revealed by the dimensionality of the local manifold patch on which the point is laying. With the assumption that manifold is locally linear, we represent the local patch by the covariance matrix of the points on it. In our implementation, points on $\mathbf{x}_i$'s local patch are $\mathbf{x}_i$'s K nearest neighbours in the dataset (denoted by ζ). The set of K nearest neighbors of $\mathbf{x}_i$ is denoted by $\mathcal{K}(\mathbf{x}_i, \zeta)$. Decomposing the (local) covariance matrix as $\sum_i^D \lambda_i \mathbf{u}_i \mathbf{u}_i^T$, we obtain the orthogonal components as eigenvectors $\mathbf{u}_i$ and their corresponding eigenvalues λ_i (we rescale them, so that $\sum_i^D \lambda_i = 1$). The covariance matrix, in the 3D data example, is rewritten as

$$\sum_{i=1}^{3} \lambda_i \mathbf{u}_i \mathbf{u}_i^T = S_1 \mathbf{u}_1 \mathbf{u}_1^T + \frac{1}{2} S_2 (\mathbf{u}_1 \mathbf{u}_1^T + \mathbf{u}_2 \mathbf{u}_2^T) + \frac{1}{3} S_3 (\mathbf{u}_1 \mathbf{u}_1^T + \mathbf{u}_2 \mathbf{u}_2^T + \mathbf{u}_3 \mathbf{u}_3^T),$$

where $\mathbf{u}_1 \mathbf{u}_1^T$, $1/2(\mathbf{u}_1 \mathbf{u}_1^T + \mathbf{u}_2 \mathbf{u}_2^T)$ and $1/3(\mathbf{u}_1 \mathbf{u}_1^T + \mathbf{u}_2 \mathbf{u}_2^T + \mathbf{u}_3 \mathbf{u}_3^T)$ are the covariance matrices of the structures having intrinsic dimension $d = 1$, $d = 2$ and $d = 3$ respectively. Therefore the saliences of these structures are computed as $S_1 = \lambda_1 - \lambda_2$, $S_2 = 2(\lambda_2 - \lambda_3)$ and $S_3 = 3\lambda_3$. Note that $S_1 + S_2 + S_3 = 1$. Intrinsic dimension of the point is then $d = \arg\max_i S_i$.

The intrinsic dimension estimator is a variation of the approach in [7], where in contrast to [7], we operate directly on the tangent spaces of the underlying manifold.

Figure 2 demonstrates the performance of our intrinsic dimension estimator on the multiple manifolds dataset in figure 1. Therefore the whole set $\zeta = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is substituted by a partition $\zeta^1 \cup \ldots \zeta^d \ldots \cup \zeta^D$ with d indicating the intrinsic dimension of the subset.

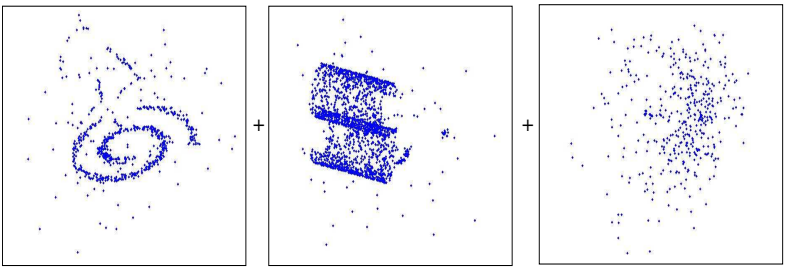


Fig. 2. Three intrinsic-dimension-filtered subsets of data from Fig. 1

3.2 Multi-manifolds Learning Algorithm

The goal of our multiple manifolds learning algorithm is to represent manifolds contained in each set $\zeta^d (d < D)$ by latent variable models, respectively. We use

the notation $\mathcal{M}_d$ to denote the set of manifolds' generative models $p(\mathbf{x}|d,q)$, where q is the index for the manifold found with dimension d. In the proposed work, the latent variable model we used is GTM [9]. In the following, we first briefly introduce GTM and then demonstrate the local optima problem in GTM's training. This motivates us to propose a novel robust manifold learning algorithm which provides a better initialization aligned along the non-linear manifold.

Generative Model for Noisy Manifolds. Generative topographic mapping (GTM) represents the non-linear transformation by a generalized linear regression model $f^{d,q} = \mathbf{W}^{d,q}\Phi^{d,q}(\mathbf{y}^{d,q})$. The latent variable space is a d-dimensional (hyper)cube and is endowed with a (prior) probability distribution $p(\mathbf{y}^{d,q})$ - a set of delta functions. Noise model $p(\mathbf{x}|\mathbf{y}^{d,q})$ is a radially-symmetric Gaussian distribution with mean $f^{d,q}(\mathbf{y}^{d,q})$ and covariance $1/\beta^{d,q}\mathbf{I}$, where $\beta^{d,q} > 0$. The generative model $p(\mathbf{x}|d,q)$ can be obtained by integrating over the latent variables

$$\begin{aligned} p(\mathbf{x}|d,q) &= \frac{1}{Z^{d,q}} \sum_{z=1}^{Z^{d,q}} p(\mathbf{x}|\mathbf{y}_z^{d,q}, \mathbf{W}^{d,q}, \beta^{d,q}) \\ &= \frac{1}{Z^{d,q}} (\frac{\beta^{d,q}}{2\pi})^{(D/2)} \exp\{-\frac{\beta^{d,q}}{2} ||f^{d,q}(\mathbf{y}_z^{d,q}) - \mathbf{x}||^2\} \end{aligned} \tag{1}$$

where $Z^{d,q}$ denotes the number of the latent variable $\mathbf{y}^{d,q}$, the map $f^{d,q}$ is defined above and $\Phi^{d,q}(\mathbf{y}_k^{d,q})$ is a column vector

$$\Phi^{d,q}(\mathbf{y}_z^{d,q}) = [\phi_1(\mathbf{y}_z^{d,q}), \phi_2(\mathbf{y}_z^{d,q}), \ldots, \phi_M^{d,q}(\mathbf{y}_z^{d,q})]^T \tag{2}$$

We can in principle determine $\mathbf{W}^{d,q}$ and $\beta^{d,q}$ by maximizing the log likelihood function through E-M. But the optimization procedure cannot guarantee the global optimum in case of a strong non-linear manifold structure, because original GTM is typically initialized through a linear global PCA.

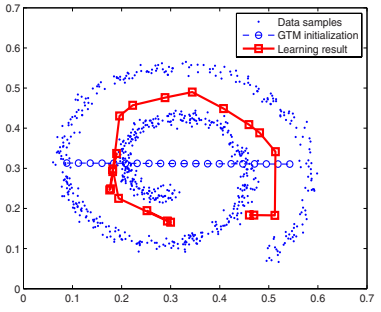
We demonstrate the performance of GTM on a strong non-linear manifold data, spiral dataset. This set of 700 points were generated from the following distribution of two dimensional (x_1, x_2) points:

$$x_1 = 0.02t\sin(t) + 0.3 + \epsilon_{x_1}; \qquad x_2 = 0.02t\cos(t) + 0.3 + \epsilon_{x_2} \tag{3}$$

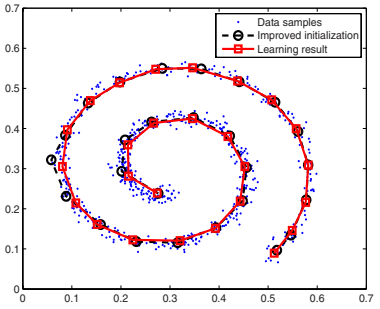
where $t \sim \mathcal{U}(3,15)$, $\epsilon_{x_1} \sim \mathcal{N}(0, 0.01)$, $\epsilon_{x_2} \sim \mathcal{N}(0, 0.01)$, $\mathcal{U}(a,b)$ is the uniform distribution over the interval (a,b) and $\mathcal{N}(0,\delta)$ is the zero-mean Gaussian distribution with standard deviation δ.

In figure 3(a), we illustrate the initialization and training result of classical GTM on the dataset described above. The initial Gaussian centers are obtained by mapping the one dimensional latent variables pre-defined through global PCA. The training result with this initialization approximates the non-linear spiral data manifold poorly. In contrast to figure 3(a), figure 3(b) shows an improved fit by the GTM initialized by the method described below.

Identifying One Manifold. Here we propose a novel strategy to identify the embedded manifolds and capture their (high) non-linear structure in the learning



(a) Initialization and learning result of classical GTM



(b) GTM initialized with the new manifold learning approach

Fig. 3. Learning results from different GTM initializations

procedure. In our proposed framework, we describe the low dimensional latent manifold by an oriented graph $\mathcal{G} = (\mathcal{Y}, \mathcal{E})$, where $\mathcal{Y}$ represents the graph vertices and $\mathcal{E}$ represents the edges. We label the i-th vertex in $\mathcal{Y}$ by a d-dimensional point $\mathbf{y}_i \in \mathcal{R}^d$. The directed edges from the vertex $\mathbf{y}_i$ are collected in $\mathbf{E}_i$ that can be also thought of as the set of destination vertices. The coordinates of these destination vertices (children of the source/parent vertex $\mathbf{y}_i$ and denoted by $Ch(\mathbf{y}_i)$) can be calculated as

$$Ch(\mathbf{y}_i) = \mathbf{y}_i + l \times \mathbf{e}_i^{\mathbf{o}} \tag{4}$$

where $\mathbf{e}_i^{\mathbf{o}}$ is a collection of unit directional vectors of the outgoing edges, and l represents the length of the edge (fixed to be 1 in our implementation). We denote the outgoing edges from vertex $\mathbf{y}_i$ as $\mathbf{E}_i = \{\mathbf{e}_i^{\mathbf{o}}\}$ for simplicity.

An example of a graph structure for $d = 2$ is illustrated in figure 4. We describe our manifold learning approach based on this case. Partitioning all the vertices according to the number of their parents, we obtain $\mathcal{Y} = \mathcal{Y}_0 \cup \mathcal{Y}_1 \cup \mathcal{Y}_2$, where $\mathcal{Y}_0$, $\mathcal{Y}_1$, $\mathcal{Y}_2$ represent collections of vertices with $0-$, $1-$ and $2-$ parents, respectively. The set $\mathcal{Y}_0$ contains the unique parentless vertex in the graph. We refer to this vertex as the origin of the graph and set its coordinates to $\mathbf{0} \in \mathcal{R}^d$. The edges from the origin correspond to $2d = 4$ outgoing edges (orthonormal vectors) denoted by $\mathbf{E}_0$ and specified as $\{(1,0),(-1,0),(0,1),(0,-1)\}$ in the implementation.

Knowing $\mathcal{Y}_0$ and $\mathbf{E}_0$, we could retrieve all the vertices in this graph by eq. (4) from their parents and whose outgoing edges $\mathbf{E}_i$. And we obtain the outgoing edges by the following equation:

$$\mathbf{E}_i = \begin{cases} \mathbf{E}_0 & \mathbf{y}_i \in \mathcal{Y}_0 \\ \mathbf{E}_{Pa(i)} \setminus -\mathbf{E}_i^I & \mathbf{y}_i \in \mathcal{Y}_1 \\ \mathbf{E}_i^I & \mathbf{y}_i \in \mathcal{Y}_2 \end{cases} \tag{5}$$

where $\mathbf{E}_i^I$ denotes the directions of the incoming edges of vertex $\mathbf{y}_i$, $Pa(i)$ stands for vertex $\mathbf{y}_i$'s only parent when $\mathbf{y}_i \in \mathcal{Y}_1$, $A \backslash B$ is the operation of removing items in A if they are also in B and $-B$ returns the opposite directions of B.

A mapped graph $\mathcal{G}^m = (\mathcal{X}, \mathcal{E}^x) = F(\mathcal{G})$ in high dimensional space $\mathcal{R}^D$ (here $D = 3$) is also illustrated in figure 4. The vertices $\{\mathbf{y}_i^m\}$ and directed edges $\{\mathbf{E}_i^m\}$ in the high dimensional graph are obtained by mapping the vertices $\{\mathbf{y}_i\}$ and the edges $\{\mathbf{E}_i\}$ respectively. The vertex mapping $F_y(\mathbf{y}_i)$ is described as follows:

$$\begin{aligned} &\mathcal{X}_0 = F_y(\mathcal{Y}_0) \\ &Ch(\mathbf{y}_i^m) = F_y(Ch(\mathbf{y}_i)) = \mathbf{y}_i^m + L \times F_e(\mathbf{E}_i, \mathbf{y}_i) \end{aligned} \tag{6}$$

where $\mathcal{X}_0$ is the origin of the mapped graph $\mathcal{G}^m$. F_e is the edge mapping and L is the fixed edge length in the mapped graph. The mapped edges in the data space are then obtained via the mapping

$$F_e(\mathbf{E}_i, \mathbf{y}_i) = \begin{cases} \mathbf{M}_{\mathbf{y}_i^m} \mathbf{E}_0^m & \mathbf{y}_i \in \mathcal{Y}_0 \\ \mathbf{M}_{\mathbf{y}_i^m} \mathbf{E}_{pa(i)}^m \backslash -\mathbf{E}_i^{mI} & \mathbf{y}_i \in \mathcal{Y}_1 \\ \mathbf{M}_{\mathbf{y}_i^m} \mathbf{E}_i^{mI} & \mathbf{y}_i \in \mathcal{Y}_2 \end{cases} \tag{7}$$

where $\mathbf{M}_{\mathbf{y}_i^m}$ is the projection matrix onto the manifold patch around the point $\mathbf{y}_i^m$ and given by

$$\mathbf{M}_{\mathbf{y}_i^m} = \mathbf{B}_{\mathbf{y}_i^m} \mathbf{B}_{\mathbf{y}_i^m}^T, \tag{8}$$

where $\mathbf{B}_{\mathbf{y}_i^m}$ denotes the matrix of basis vectors which are the first d eigenvectors with largest eigenvalues corresponding to principle directions of the neighborhood of the vertex $\mathbf{y}_i^m$. In eq. (7), $\mathbf{E}_0^m$ denotes the orthonormal outgoing edges of the mapped origin $\mathcal{X}_0$.

We present an algorithm to simultaneously learn the graph $\mathcal{G}$ and the associated mapped graph $\mathcal{G}^m$ from a dataset ζ^d.

- Initialization
  Learning of the mapped graph is initialized by specifying $\mathcal{X}_0$, L in eq. (6), and $\mathbf{E}_0^m$ in eq. (7). In a dataset without outliers (isolated points not lying "close" to any apparent manifold), the origin $\mathcal{X}_0$ of the mapped graph can be associated with any randomly chosen point. Since the presence of outliers cannot be ruled out, we include outlier detection (steps 1, 2) in the algorithm **initialization**.
- Recursive learning
  With the initialization, we set the origins $\mathcal{Y}_0$ and $\mathcal{X}_0$ to be the current generations of graphs $\mathcal{G}$ and $\mathcal{G}^m$, denoted by CG and CG^m. The learning procedure forms the graphs together iteratively from the current generation to its next generation (NG and NG^m) until the boundary of the manifold is detected.
  - $[NG, NG^m] = Redun\_remove(NG, NG^m)$
    This procedure removes duplicate vertices in NG and NG^m. The duplicate vertices are generated in the learning procedure, because vertices may have more than one parent. For example, in the graph $\mathcal{G}$, any vertex $\mathbf{y}_s \in \mathcal{Y}_2$ is a child of two parents. It is then learnt as a set of vertices $\{\mathbf{y}_{s1}, \mathbf{y}_{s2}\} \in NG$ from the current generation. Since these vertices

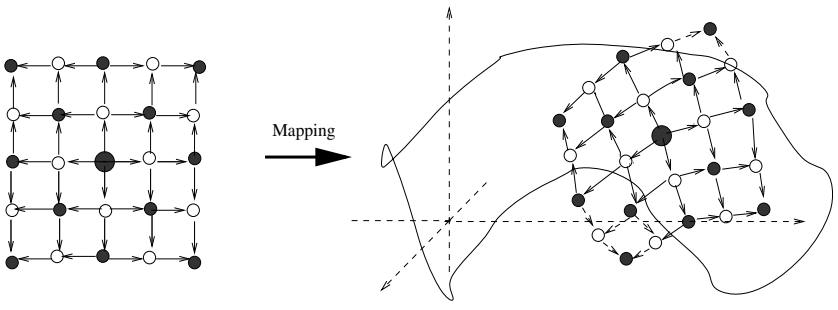


Fig. 4. The directed graph representing 2-dimensional latent space and its mapped graph lying on 2 dimensional manifold in the 3 dimensional data space

Initialization

1. For all $\mathbf{x}_i \in \zeta^d$ do:
   (a) Count the presence of $\mathcal{K}(\mathbf{x}_i, \zeta)$ in dataset ζ^d, i.e., k =the size of $\{\mathcal{K}(\mathbf{x}_i, \zeta) \cap \zeta^d\}$,
   (b) If $k > \alpha_1 K$ $(0 < \alpha_1 < 1)$, then $\mathcal{X}_0 = \mathbf{x}_i$, break;
2. EndFor
3. If $\mathcal{X}_0 != null$, then
   (a) Find $\mathcal{K}(\mathcal{X}_0, \zeta^d)$ with parameter K_2 (neighborhood size).
   (b) $\mathbf{E}_0^m = \{\mathbf{u}_1, \ldots, \mathbf{u}_d, -\mathbf{u}_1, \ldots, -\mathbf{u}_d\}$, where $\{\mathbf{u}_1, \ldots, \mathbf{u}_d\}$ are the first d principal directions of $\mathcal{K}(\mathcal{X}_0, \zeta^d)$.
   (c) L is set to be the average of the distance from the neighbours to $\mathcal{X}_0$.
4. EndIf

Learning

1. $NG = \{\}$; $NG^m = \{\}$
2. While $CG != null$ do:
   (a) i. For all $\mathbf{y}_i \in CG$
   A. $NG = \{NG; Ch(\mathbf{y}_i)\}$;
   B. $NG^m = \{NG^m; F_y(Ch(\mathbf{y}_i)\}$
   ii. EndFor
   (b) $[NG, NG^m] = Redun\_remove(NG, NG^m)$
   (c) $[CG, CG^m] = Boud\_check(NG, NG^m)$
   (d) $\mathcal{Y} = \{\mathcal{Y}; CG\}$;
   (e) $\mathcal{X} = \{\mathcal{X}; CG^m\}$;
3. EndWhile

have the same coordinates, we replace this set by $\mathbf{y}_s$. The corresponding mapped set $\{\mathbf{y}^m_{s1}, \mathbf{y}^m_{s2}\} \in NG^m$ should also be replaced by a single vertex $\mathbf{y}^m_s \in \mathcal{Y}^m$ even though the mapped vertices may not overlap in the data space.

Learning ::*Redun_remove*(NG, NG^m)

1. For all $\{\mathbf{y}_{s1}, \mathbf{y}_{s2}\} \in NG$ of the same coordinate
   (a) Replace them by $\mathbf{y}_s$ with the same coordinate.
   (b) Collapse the corresponding $\mathbf{y}^m_{s1}, \mathbf{y}^m_{s2}$ into its mean with the notation $F_y(\mathbf{y}_s)$.
   (c) Collect incoming edges of $\mathbf{y}_s$ and $\mathbf{y}^m_s$ by their directions in $\mathbf{E}^I_s$ and $\mathbf{E}^{Im}_s$
2. EndFor

- $[CG, CG^m] = Boud\_check(NG, NG^m)$
  This procedure checks for the manifold boundary. The learning procedure should stop when the boundary is reached. Close to the boundary, the local density of points around the current point decreases rapidly. In such situations the set of nearest neighbors overlaps significantly (determined by parameter K_3) with the set of already visited points on the manifold.

Learning ::*Boud_check*(NG, NG^m)

1. $CG = \{\}$; $CG^m = \{\}$;
2. For all $\mathbf{y}^m_j \in NG^m$
   (a) Find $\mathcal{K}(\mathbf{y}^m_j, \zeta^d)$ and $\mathcal{O}$ = the size of $\{\mathcal{K}(\mathbf{y}^m_j, \zeta^d) \cap \mathcal{K}(CG^m, \zeta^d)\}$.
   (b) If $\mathcal{O} < K_3$, then
      i. $CG = \{CG, \mathbf{y}_j\}$;
      ii. $CG^m = \{CG^m, \mathbf{y}^m_j\}$
   (c) EndIf
3. EndFor

Initializing GTM with Learnt Graphs. Each detected manifold will be modeled by a single GTM initialized using the graphs $\mathcal{G}$ and $\mathcal{G}^m$. Vertices in graph $\mathcal{G}$ are the low dimensional latent variables and vertices in the mapped graph $\mathcal{G}^m$ are the images the latent variables under the GTM model. Therefore, we determine $\mathbf{W}^{d,q}$ by minimizing the error

$$Err = \frac{1}{2} \sum_{z=1}^{Z^{d,q}} ||\mathbf{W}^{d,q} \Phi(\mathbf{y}^{d,q}_z) - F_y(\mathbf{y}^{d,q}_z)||^2. \tag{9}$$

The parameter $\beta^{d,q}$ is initialized to be the larger of the average of the $d+1$ eigenvalues from PCA applied on each vertex in $\mathcal{Y}^m$ and the square of half of the grid spacing of the mapped vertices $\mathcal{Y}^m$ in the data space.

As an example, figure 3(b) shows the result of GTM fitting the spiral data after being initialized with our manifold learning technique. With this initialization, we can see that a little or no further GTM training is required.

Learn Multiple Manifolds. Since the proposed manifold learning algorithm explores the dataset starting from a single "seed" point and then "crawls" on the manifold, points in the data space which are not connected in the same manifold will be left unvisited. For datasets having more than one underlying manifolds, we utilize an iterative procedure (see below) to explore the unvisited set $\tilde{\zeta}$ (entire filtered set at the beginning), learn the manifolds and initialize the corresponding generative models one by one.

1. $\tilde{\zeta}^d = \zeta^d (1 \leq d < D)$; $q = 0$
2. While $\tilde{\zeta}^d > K_2$
   (a) Learn $\mathcal{G}$ and $\mathcal{G}^m$ from $\tilde{\zeta}^d$; $q = q + 1$.
   (b) Initialize $\mathbf{W}^{d,q}$, $\beta^{d,q}$ in $p(\mathbf{x}|d,q)$ associating with the $\mathcal{G}$ and $\mathcal{G}^m$.
   (c) $\tilde{\zeta}^d = \tilde{\zeta}^d \backslash \mathcal{K}(\mathcal{X}, \tilde{\zeta}^d)$
3. EndWhile

3.3 Hierarchical Mixture Model

In this section, we formulate a two-level hierarchical model $\mathcal{T}$ and the EM algorithm to fit $\mathcal{T}$ to the entire data set $\zeta = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.

Model Formulation. We formulate the hierarchical model $\mathcal{T}$ by first mixing the models $p(\mathbf{x}|, d, q) \in \mathcal{M}_d$ at the second level hierarchy with $\pi_{q|d}$, for each intrinsic dimensionality d, $\sum_q \pi_{q|d} = 1$ $(q > 0)$. If $\mathcal{M}_d = null$, then we set $q = 0$ and $\pi_{0|d} = 0$. We the mix these intrinsic dimensionality groups with π_d at the first level hierarchy. Thus we obtain:

$$p(\mathbf{x}|\mathcal{T}) = \sum_{d=1}^{D} \pi_d \sum_{q \in \mathcal{M}_d} \pi_{q|d} p(\mathbf{x}|d,q). \tag{10}$$

The probabilistic models in $\mathcal{M}_d$ $(1 \leq d < D)$ are formulated in eq. (1). We use a Gaussian mixture (11) to model the data points collected in ζ^D.

$$\begin{aligned} p(\mathbf{x}|D,1) &= \sum_{z=1}^{Z^{D,1}} p(z) p(\mathbf{x}|z;D,1) \\ &= \sum_{z=1}^{Z^{D,1}} p(z) \frac{1}{\sqrt{||2\pi\mathbf{\Sigma}_z||}} \exp\{-\frac{1}{2}(\mathbf{x}-\mu_z)^T \mathbf{\Sigma}_z^{-1}(\mathbf{x}-\mu_z)\} \end{aligned} \tag{11}$$

where $Z^{D,1}$ is the number of Gaussian components, μ_z and $\mathbf{\Sigma}_z$ are the mean and covariance of the $z-$th component, and $p(z)$'s are the mixing coefficients with $\sum_{z=1}^{Z^{D,1}} p(z) = 1$.

EM Algorithm. The mixing coefficients at each hierarchical level and the parameters of each submodel can be determined by maximizing the log likelihood function of the model (10).

$$\mathcal{L} = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathcal{T}) \tag{12}$$

We use binary assignment variables $\upsilon_{n,d}$ to represent that the data $\mathbf{x}_n$ belongs to the group of manifolds having dimension d, and $\upsilon_{n,q|d}$ to represent the situation that data $\mathbf{x}_n$ is generated from q-th manifold in the group with dimension d. Even if this was known, we still need to decide which latent space center $\mathbf{y}_z^{d,q} \in \mathcal{Y}^{d,q}, z = 1, 2, \ldots, Z^{d,q}$ in the latent variable model $p(\mathbf{x}|d,q)$ $(1 \leq d < D)$ corresponds to the Gaussian that generated $\mathbf{x}_n$, We represent this by indicator variables $\upsilon_{n,z}^{d,q}$.

For $d = D$, $\mathcal{M}_D$ contains a single unconstrained Gaussian mixture model. The complete data likelihood function reads

$$\mathcal{L}_c = \sum_{n=1}^{N} \sum_{d=1}^{D} \upsilon_{n,d} \sum_{q \in \mathcal{M}_d} \upsilon_{n,q|d} \sum_{z=1}^{Z^{d,q}} \upsilon_{n,z}^{d,q} \ln\{\pi_{q|d}\pi_d p(\mathbf{x}_n|d,q)\} \tag{13}$$

Taking the expectation (with respect to the posterior given the data) over all types of hidden variables, we arrive at the expected complete-data likelihood

$$< \mathcal{L}_c > = \sum_{n=1}^{N} \sum_{d=1}^{D} R_{d|n} \sum_{q \in \mathcal{M}_d} R_{q|d,n} \sum_{z=1}^{Z^{d,q}} R_{z|q,d,n} \ln\{\pi_d \pi_{q|d} p(\mathbf{x}_n, |d,q)\} \tag{14}$$

The M-step of the EM algorithm involves maximizing (14) w.r.t. the parameters.

We obtain the following updates:

$$\tilde{\pi}_d = \frac{1}{N} \sum_{n=1}^{N} R_{d|n} \tag{15}$$

$$\tilde{\pi}_{q|d} = \frac{\sum_{n=1}^{N} R_{d|n} R_{q|d,n}}{\sum_{n=1}^{N} R_{d|n}} \tag{16}$$

$$p(z) = \frac{\sum_{n=1}^{N} R_{D|n} R_{z|D,n}}{\sum_{n=1}^{N} \sum_{z=1}^{Z^{D,1}} R_{d|n} R_{z|d,n}} \tag{17}$$

As for the manifold models, using eq. (1) and (2), we have

$$\sum_{n=1}^{N} R_{nd} \sum_{z=1}^{Z^{d,q}} R_{z|d,qn} (\mathbf{W}^{d,q} \Phi(\mathbf{y}_z^{d,q}) - \mathbf{x}_n) \Phi^T(\mathbf{y}_z^{d,q}) = 0 \tag{18}$$

The responsibilities $R_{d|n}$ and $R_{z|n,d,q}$ are calculated with the current (old) weight and inverse variance parameters of the probabilistic models $p(\mathbf{x}|d,q)$.

Written in matrix notation, we have to solve

$$(\mathbf{\Phi}^{d,q})^T\mathbf{B}^{d,q}(\mathbf{\Phi}^{d,q})^T(\mathbf{W}^{d,q})^T = (\mathbf{\Phi}^{d,q})^T\mathbf{R}^{d,q}\mathbf{T} \tag{19}$$

for $\mathbf{W}^{d,q}$.

The above system of linear equations involves the following matrices:

- $\mathbf{\Phi}$ is a $Z^{d,q} \times M^{d,q}$ matrix with elements $(\mathbf{\Phi}^{d,q})_{ij} = \phi_j(\mathbf{y}_z^{d,q})$.
- $\mathbf{T}$ is a $N \times D$ matrix storing the data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ as rows.
- $\mathbf{R}^{d,q}$ is a $Z^{d,q} \times N$ matrix containing, for each latent space center $\mathbf{y}_z^{d,q}$, and each data point $\mathbf{x}_n$, scaled responsibilities $(\mathbf{R}^{d,q})_{zn} = R_{d|n}R_{q|n,d}R_{z|q,d,n}$
- $\mathbf{B}^{d,q}$ is a $Z^{d,q} \times Z^{d,q}$ diagonal matrix with diagonal elements corresponding to responsibilities of latent space centers for the whole data sample ζ, where $(\mathbf{B})_{ii} = \sum_{n=1}^{N} R_{d|n}R_{q|d,n}R_{z|q,d,n}$.

The GTM mapping $f^{d,q}$ can be regularized by adding a regularization term to the likelihood (1). Inclusion of the regularizer modifies eq.(19) to

$$[(\mathbf{\Phi}^{d,q})^T\mathbf{B}^{d,q}(\mathbf{\Phi}^{d,q})^T + \frac{\alpha^{d,q}}{\beta^{d,q}}\mathbf{I}](\mathbf{W}^{d,q})^T = (\mathbf{\Phi}^{d,q})^T\mathbf{R}^{d,q}\mathbf{T} \tag{20}$$

where $\mathbf{I}$ is the $Z^{d,q} \times Z^{d,q}$ identity matrix.

Finally, the re-estimation formulation

$$\frac{1}{\beta^{d,q}} = \frac{\sum_{n=1}^{N} R_{d|n}R_{q|d,n}\sum_{z=1}^{Z^{d,q}} R_{z|n,d,q}||\mathbf{W}^{d,q}\phi(\mathbf{y}_z^{d,q}) - \mathbf{x}_n||^2}{D\sum_{n=1}^{N} R_{d|n}R_{q|d,n}} \tag{21}$$

where $\mathbf{W}^{d,q}$ is the "new" weight matrix computed by solving (19) in the last step.

We also obtain

$$\mu_z = \frac{\sum_{n=1}^{N} R_{D|n}R_{z|D,n}\mathbf{x}_n}{\sum_{n=1}^{D} R_{D|n}} \tag{22}$$

$$\mathbf{\Sigma}_z^D = \frac{\sum_{n=1}^{N} R_{D|n}R_{z|D,n}(\mathbf{x}_n - \mu_z)(\mathbf{x}_n - \mu_z)^T}{\sum_{n=1}^{N} R_{D|n}} \tag{23}$$

In the E-step of the EM algorithm, we estimate the latent space responsibilities $R_{z|n,d,q}$ in submodels for manifolds and component responsibilities $R_{z|D}$ in Gaussian mixture model. Model responsibilities in each group $R_{q|n,d}$ and the group responsibilities $R_{d|n}(1 \leq d \leq D)$ are specified as well.

$$R_{z|d,q,n} = \frac{p(\mathbf{x}_n|\mathbf{y}_z, d, q)}{\sum_{z'=1}^{Z^{d,q}} p(\mathbf{x}_n|\mathbf{y}'_z, d, q)} \tag{24}$$

$$R_{z|D,n} = \frac{p(z)p(\mathbf{x}_n|z, D)}{\sum_{z'=1}^{Z^D} p(z')p(\mathbf{x}_n|z, d, q)} \tag{25}$$

$$R_{q|d,n} = \frac{\pi_{q|d}p(\mathbf{x}_n|d, q)}{\sum_{q' \in \mathcal{M}_d} \pi_{q'|d}p(\mathbf{x}_n|d, q')} \tag{26}$$

$$R_{d|n} = \frac{\pi_d \sum_{q \in \mathcal{M}_d} \pi_{q|d}p(\mathbf{x}|d, q)}{\sum_{d'=1}^{D} \pi_{d'} \sum_{q' \in \mathcal{M}_{d'}} \pi_{q'|d'}p(\mathbf{x}_n|d', q')} \tag{27}$$

Parameter Initialization. There are two groups of parameters to be initialized before running EM algorithm to fit $\mathcal{T}$ to the data.

An unconstrained GMM is used to model the set ζ^D. The corresponding parameters including Gaussian centers and covariance can be initialized e.g. by a simple K-means algorithm. Parameters in latent variable model $p(\mathbf{x}|d, q)$, where $d < D$ are initialized using our multi manifold learning algorithm described in the previous section.

4 Experiments

Multi-manifolds with Varying Dimensions. We first present the multiple manifolds learning results on the dataset illustrated in figure 1. Figure 2 shows the filtered subsets of data points with respect to their intrinsic dimensionality. Note that because the manifolds cross, there is a gap splitting the single $1d$ manifold into two parts. The points in the gap were taken to the set of intrinsically 2-dimensional points. Given a strong prior knowledge concerning connectiveness of the data manifolds, we could deal with such situations, however in the absence of such information we would prefer to have several isolated components dictated by the data.

We use our multi manifold initialization alongside with EM algorithm to fit the full hierarchical model to the dataset. The results shown in figure 5. The manifolds of varying dimensionality and shape were correctly identified.

Identifying Streams and Shells in Disrupted Galaxies. Recent exciting discoveries of low-dimensional structures such as shells and streams of stars in the vicinity of large galaxies led the astronomers to investigate the possibility that such structures are in fact remnants of disrupted smaller satellite galaxies. One line of investigation models the disruption process using realistic particle models of both the large galaxy (e.g. M31) and the disrupted one (e.g. M32) [15,16]. Realistic set of initial conditions are applied and the results of particle simulations are compared with the observed structures for each initial condition setting. This is of most importance for understanding the disruption process and prediction of the future continuation of the satellite disruption. We will show that

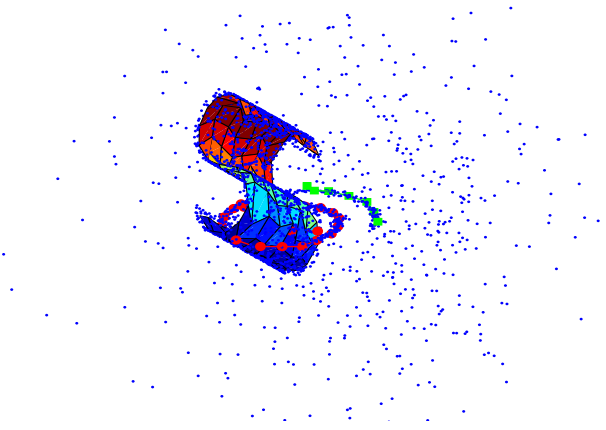
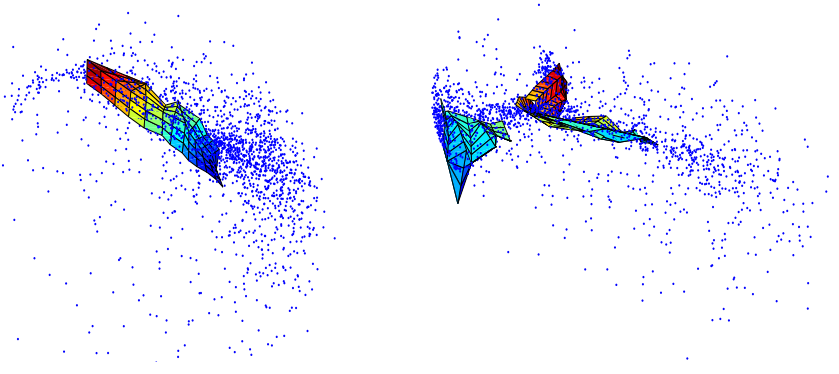


Fig. 5. Manifolds learnt from the artificial dataset dataset in figure 1. We use parameters $K = 20, \alpha_1 = 0.8, K_2 = 20, K_3 = 18$.



(a) (b)

Fig. 6. Identified 2-dimensional manifolds in disrupted satellite galaxy at an early (a) and later (b) stages of disruption by M31

our multi manifold learning methodology can be used for automatic detection of low dimensional streams and shells in such simulations, thus enabling automated inferences in large scale simulations.

Using one realistic setting of the particle model [15,16] for satellite galaxy disruption by M31, we obtained several stages of the disrupted satellite, modeled by approximately 30,000 particles (stars). In each stage we applied the multi manifold learning. In figure 6 we show (along with the stars (particles)) the detected two-dimensional manifold structures ("skeletons of the mixture components) in an early and a later stage of the disruption process. 1- and 3-dimensional structures are not shown. In the early stage a single stream was automatically detected, whereas in the later stage a stream and two shells were correctly identified. Two-dimensional structures are of most importance in such investigations,

but our system can be used for investigation of structures across a variety of dimensions. It can also be used to build a hierarchical mixture model for the full system (large galaxy + a satellite) for the purposes of principled comparison of the simulated system with the real observations (in the projected plane). The is a matter for our future work.

5 Conclusions and Discussion

We presented a novel hierarchical model based framework to learn multiple manifolds of possibly different intrinsic dimensionalities. We first filter the data points with respect to the intrinsic dimensionality of the manifold patches they lie on. Then our new multi manifold learning algorithm is applied to each such filtered dataset of dimensionality d to detect d-dimensional manifolds along which the data are aligned. This is later used to initialize generative latent variable models representing noisy manifolds underlying the data set. The generative models are combine in a hierarchical mixture representing the full data density. The proposed approach is significantly different from the current manifold learning approaches which typically assume that the whole data set is sampled from a single low dimensional manifold, which may not always be realistic.

As with other manifold learning approaches, parameter selection (e.g. neighborhood size) can be an issue. Model selection approaches can be used to select the appropriate values for a given application, but obviously much more work is required in this direction. In this paper we present a proof of concept and show that our multi manifold learning framework can be potentially applied in interesting application domains, such as astronomy.

Acknowledgments. We would like to thank Somak Raychaudhury and Arif Babul for stimulating discussions.

References

1. Moore, B.: Principal component analysis in linear systems: Controllability, observability, and model reduction. IEEE Transactions on Automatic Control, 17–32 (February 1981)
2. Kohonen, T. (ed.): Self-Organization and Associative Memory. Springer, Heidelberg (1997)
3. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
4. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(5500), 2319–2323 (2000)
5. Levina, E., Bickel, P.: Maximum likelihood estimation of intrinsic dimension. In: Advances in NIPS (2005)
6. Bruske, J., Sommer, G.: Intrinsic dimensionality esitmation with optimally topology preserving maps. IEEE Transactions on Pattern Analysis and Machine Intelligence, 572–575 (May 1998)

7. Mordohai, P., Medioni, G.: Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. In: International Joint Conference on Artifcial Intelligence, pp. 798–803 (2005)
8. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology) (3), 611–622 (1999)
9. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: The generative topographic mapping. Neural Computation 10(1), 215–234 (1998)
10. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. In: Neural Computation, pp. 1373–1396 (June 2003)
11. Saul, L.K., Roweis, S.T.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Journal of machine learning research, 119–155 (2003)
12. Bishop, C.M., Tipping, M.E.: A hierarchical latent variable model for data visualization. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 281–293 (1998)
13. Tino, P., Nabney, I.: Hierarchical GTM: constructing localized non-linear projection manifolds in a principled way. IEEE Transactions on Pattern Analysis and Machine Intelligence (in print, 2001)
14. Williams, C.: A MCMC approach to hierarchical mixture modelling. In: Advances in Neural Information Processing Systems 12, pp. 680–686. MIT Press, Cambridge (2000)
15. Fardal, M.A., Babul, A., Geehan, J.J., Guhathakurta, P.: Investigating the andromeda stream - ii. orbital fits and properties of the progenitor. Monthly Notices of the Royal Astronomical Society 366, 1012–1028 (2006)
16. Fardal, M., Guhathakurta, P., Babul, A., McConnachie, A.W.: Investigating the andromeda stream - iii. a young shell system in m31. Monthly Notices of the Royal Astronomical Society 380, 15–32 (2007)