Incremental Multi-manifold Out-of-Sample Data Prediction

Zhongxin Liu, Wenmin Wang*, Ronggang Wang
School of Electronic and Computer Engineering
Shenzhen Graduate School, Peking University
Lishui Road 2199, Nashan District, Shenzhen, China 518055
1201213663@sz.pku.edu.cn, {wangwm, rgwang}@ece.pku.edu.cn

***Abstract*—A lot of manifold learning algorithms have been developed, which are used to learn a low dimensional model on a manifold representing large numbers of data in high dimensionality. Multi-manifold learning algorithms have also been put forward to provide a compact representation when these data come from different classes, with different intrinsic dimensionalities. However, when unseen data samples are added to the data set, the necessity of retraining becomes a barrier to the application of multi-manifold learning algorithms as preprocessing step in predictive modeling. In this paper, an incremental out-of-sample data low dimensional coordinates prediction approach is proposed to solve the out-of-sample data problem for multi-manifold. The algorithm can learn a global low dimensional structure with randomly sampled data from each class in the first step, and can compute the low dimensional coordinates on the corresponding manifold for each new coming data effectively. The algorithm is evaluated using both synthetic and real-world datasets and the results are shown both qualitatively and quantitatively.**

Keywords-dimensionality reduction; multi-manifold; out-of-sample data; incremental; prediction

I. INTRODUCTION

The problem of dimensionality reduction arises in many applications, such as image processing, where it is natural to represent images as vectors in a high dimensional space. A large number of dimensionality reduction methods have been proposed. In contrast to linear dimensionality reduction techniques such as Principal Component Analysis (PCA) [12], manifold learning methods provide more powerful non-linear dimensionality reduction by preserving the local structure of the input data. Manifold learning algorithms are based on the assumption that the data lie on or around a single low dimensional manifold in a high dimensional Euclidean space.

Several powerful manifold learning techniques have been proposed, e.g., Semi-definite Embedding (SDE) [17], Isomap [14], Laplacian Eigenmaps [1] and Locally Linear Embedding (LLE) [10]. SDE aims to preserve distances and angles between all neighboring points. It is formulated as an instance of semi-definite programming, and is thus prohibitively expensive for large scale problems. Isomap constructs a dense matrix of approximate geodesic distances between all pairs of inputs, and aims to find a low dimensional space that best preserves these distances. Other algorithms, e.g., Laplacian Eigenmaps and LLE, focus only on preserving local neighborhood and relationships in the input space. They generate low-dimensional representations via manipulation of graph Laplacian or other sparse matrices related to the graph Laplacian.

Recently, multi-manifold learning algorithms have been proposed. These methods are introduced to discover a variety of the underlying low dimensional structures. In these methods, a probability density is associated, so that the collection of manifolds can be represented by a mixture of their associated density models. The generated models are formulated as latent variable models. Data points are filtered according to the intrinsic dimensionality of local manifold patch they are likely to belong to. Multiple manifolds in each such dimension-filtered set are detected. A hierarchical probabilistic model containing density models of the detected noisy manifolds is constructed.

In the use of such techniques, after unseen data samples are added to the data set, the retraining is a time-consuming process and no guarantee of the transformation into the exactly same coordinates. This presents a barrier to the application of manifold learning as a preprocessing step in predictive modeling.

Some works have been done to reveal the relationship between the coordinates of original high dimensional data and that of low dimensional manifold [7]. Once these low dimensional coordinates are known, the problem of learning this relationship reduces to nonlinear supervised regression. Some methods have been proposed to support nonlinear supervised regression, such as Gaussian processes and support vector regression. These techniques should apply a univariate technique individually to each output feature, i.e. each of the reduced dimensions if they are not based on the least squares technique. Recently, an inductive manifold learning method based on structured support vector machine has been put forward. This technique is able to solve the out-of-sample data problem for traditional manifold learning algorithms, such as Isomap, LLE and Laplacian Eigenmaps.

However, the problem of out-of-sample data for multi-manifold learning algorithms remains to be solved. As high dimensional data may lie on different manifolds with different dimensionalities intrinsically, it is necessary to propose an efficient and effective algorithm to solve this problem.

The contributions made in this paper are as follows. Firstly, the idea of incremental multi-manifold out-of-sample data prediction is put forward, to solve the out-of-sample data problem for datasets that are composed of different classes of data. Secondly, an incremental approach is proposed to predict the low dimensional coordinates of out-of-sample data for multi-manifolds. When given $\boldsymbol{x} \in \boldsymbol{X}$, which is the coordinate in the original high dimensional space, and its corresponding low dimensional data $\boldsymbol{y} \in \boldsymbol{Y}$ in the intrinsic manifold $\boldsymbol{c}$, the proposed method predict low dimensional coordinates for new data with the learned relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$.

978-1-4799-4143-8/14 $31.00 © 2014 IEEE
DOI 10.1109/WI-IAT.2014.137

Using this approach, the low dimensional coordinate can be easily learned from the known ones.

The remainder of this paper is organized as follows. In the next section the whole structure of the incremental multi-manifold prediction algorithm (IMMP) is discussed. Then the details of the IMMP algorithm and the rationale underlying the approaches are explained in section III. The experimental results using both synthetic and real world data are demonstrated in section IV. Finally the conclusions and future works are presented in section V.

II. INCREMENTAL MULTI-MANIFOLD PREDICTION (IMMP)

As discussed above, when the real world data, such as images, come from different classes, traditional manifold learning algorithms cannot generate an effective representation for them, because those algorithms assume the whole dataset to lie on one single intrinsic low dimensional manifold. Some multi-manifold learning algorithms have been proposed in recent years, but they cannot solve the out-of-sample data problem. When new data are observed, the whole dataset is to be recalculated. Therefore, the high computational intensity is not acceptable. Under this circumstance, the incremental multi-manifold prediction (IMMP) algorithm is proposed, which can both learn the global structure of multiple intrinsic low dimensional manifolds, and solve the computational intensity problem caused by out-of-sample data. An overall description of the IMMP algorithm is provided in this section.

The problem to be solved by this algorithm can be described as below. Let $\boldsymbol{X}$ be the original data with high dimensionality, which is composed of some kinds of data from different classes. The algorithm generates a global multi-manifold structure of these data. The set of low dimensional coordinates is denoted as $\boldsymbol{Y}$. When a new data point ***xnew*** comes, the algorithm can both learn the class that ***xnew*** belongs to and its low dimensional coordinate ***ynew***.

Therefore, this algorithm can be divided into two parts naturally. In the first part, two kinds of structures are learned from the input dataset. One of them is intra-manifold structure, which is composed of data that come from the same single class. This structure can preserve local properties of each single class effectively. The other is inter-manifold structure, which is calculated among different data from different classes. This structure can generate an effective representation among these multiple classes. In the second part, when a new data point comes, the intrinsic low dimensional coordinate is calculated according to the generated coordinates and original coordinates of sample data. This calculation process can avoid the problem of high computational intensity. It does not change anything of the low dimensional representations of sample data. Therefore, this process can both preserve the local structure and global structure of the whole dataset.

By using this algorithm, not only the class of new coming data is determined dynamically, but also the low dimensional representation is calculated effectively and efficiently. With the modification of some parameters to be determined by the users, the algorithm can be applied to a lot of circumstances.

III. DETAILS OF THE IMMP ALGORITHM

A. Learning Global Structure

As described above, in this algorithm, the input data are from different classes. Each data point is assumed to be lying on a low dimensional manifold. In this process, this algorithm learns a global structure from all the sample data for all classes.

There are two kinds of structures to be learned, i.e. intra-manifold structure and inter-manifold structure. Structure within a single manifold represents the local geometric properties inside a single manifold. On the other hand, the structure among different manifolds represents the global properties of all the data set.

In the first step, the whole input dataset is divided into several parts, each composed of data that are from a single class. Let $\boldsymbol{XI}$ denote the set of all input data for all classes, and $\boldsymbol{CI}$ be the set of classes that the elements of $\boldsymbol{XI}$ belong to. Each part of $\boldsymbol{XI}$ is denoted as $\boldsymbol{XI_k}$, $\boldsymbol{k}$ = 1, 2, … , $|\boldsymbol{CI}|$. The data of each set $\boldsymbol{XI_k}$ is assumed to be lying on a single low dimensional manifold. Correspondingly, the whole dataset lies on multiple low dimensional manifolds.

In the second step, the intra-manifold structure, i.e. the low dimensional manifolds of each single part $\boldsymbol{XI_k}$, $\boldsymbol{k}$ = 1, 2, … , $|\boldsymbol{CI}|$, is to be learned in the algorithm. For each data $\boldsymbol{x_i}$ in the set $\boldsymbol{XI_k}$, its nearest $\boldsymbol{m}$ neighbors are selected, where $\boldsymbol{m}$ is a parameter defined by users. The set of neighbors is denoted as $\boldsymbol{NXI}$. So the local structure to be learned is based on the set $\boldsymbol{LXI}$, which is the union of $\boldsymbol{NXI}$ and $\{\boldsymbol{x_i}\}$. The calculation of $\boldsymbol{LXI}$ is shown in (1).

$$LXI = NXI \cup \{x_i\} \tag{1}$$

For each pair of data $\boldsymbol{x_s}$ and $\boldsymbol{x_t}$ from the set $\boldsymbol{LXI}$, first the distance between them is calculated, and shown in (2).

$$dist(x_s, x_t) = \| x_s - x_t \| \tag{2}$$

Then the weight metrics between each pair of data $\boldsymbol{x_s}$ and $\boldsymbol{x_t}$ from the set $\boldsymbol{LXI}$ are generated according to this distance metric. This weight metric is denoted as $\boldsymbol{WXI}$. Calculation of the weight metrics are shown in (3), in which the parameter μ is defined by users.

$$WXI_k(x_s, x_t) = \exp\{-\frac{dist(x_s, x_t)}{\mu}\} \tag{3}$$

By repeating the procedure as described above, for each data $\boldsymbol{x_i}$ in the whole set $\boldsymbol{XI}$, the algorithm can learn the intra-manifold structure for each single manifold from each class of sample data. However, only by doing this, the global structure for the whole dataset of all the classes cannot be represented effectively. Consequently, in this paper, a procedure used to learn inter-manifold structures among different classes is proposed, as described below.

In the third step, the inter-manifold structure, i.e. the low dimensional structure between each pair of different classes $\boldsymbol{XI_a}$ and $\boldsymbol{XI_b}$, $\boldsymbol{a}$, $\boldsymbol{b}$ = 1, … , $|\boldsymbol{CI}|$, $\boldsymbol{a} \neq \boldsymbol{b}$, is to be learned in the algorithm.

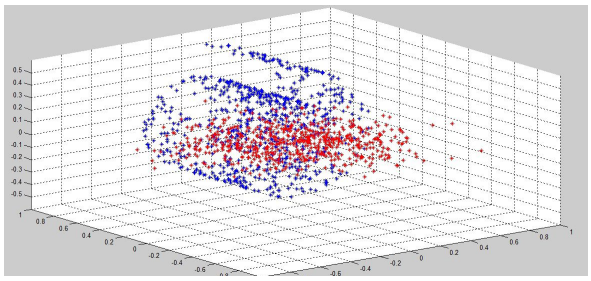

Fig. 1. The whole dataset of mixed data composed of Swiss roll and Gaussian distributed data.

For each pair of XI_a and XI_b, the matrix representing the weight metrics is denoted as WXI_{ab}, which is a sub-matrix of WXI, composed of the corresponding rows and columns of XI_a and XI_b, correspondingly. The weight metrics WXI_{ab} is based on the corresponding distance matrices, denoted by DXI_{ab}. For each x_a from XI_a and x_b from XI_b, the distance between them can be achieved from (4) below.

$$DXI_{ab}(x_a, x_b) = \| x_a - x_b \| \tag{4}$$

Then WXI_{ab} is calculated based on DXI_{ab}, as shown in (5). The right hand side of the equation is to calculate the trace of the matrix given in it. In this equation, the permutation matrix P permute the rows for matrix DXI_{ab} to maximize its trace.

$$WXI_{ab} = \arg\max_{P^T P=1} tr(P^T DXI_{ab}) \tag{5}$$

To get the result of WXI in the above equation, the SVD decomposition of the matrix DXI can be calculated simply, and the matrix representing singular values can be replaced by a diagonal matrix composed of ones on the diagonal, and then calculate the product of this new matrix with the two matrices representing singular vectors.

Through this approach, the inter-manifold weight metrics, shown in the matrix WXI can be gotten. Intra-manifold metrics are generated in the second step and inter-manifold metrics are gotten in the third step. These two kinds of metrics are calculated in different ways. The matrix WXI is used in the following step.

In the fourth step, a procedure of nonlinear dimensionality projection, based on Isomap algorithm, is used. The weight metrics WXI which is discussed above is used as the geodesic distance in the traditional learning algorithm. As the Isomap algorithm is a global method, it can preserve the global structure of the whole dataset. At the same time, the weight metrics inside each single class and among different classes are calculated in different methods, which cause the data within a single class influence more than the others. Therefore, local structures can also be preserved in this process.

A formal description of this global structure learning step is provided in algorithm 1.

B. Predict Low Dimensional Coordinates for Out-of-Sample Data

In this algorithm, after constructing the global structure of multiple low dimensional manifolds from the training set, low dimensional coordinates of new coming data are predicted according to that of training samples.

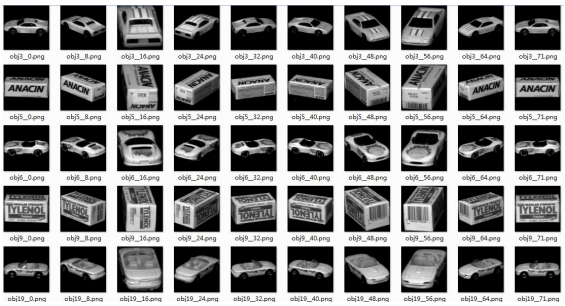

Fig. 2. An intuitive description of the dataset composed of 5 objects selected from the COIL-20 dataset.

In order to preserve the local properties for these multiple manifolds, the method we choose is to reconstruct the new coming data from its nearest neighbors. However, not all these neighbors are used. These nearest neighbors can be divided into several parts, each from a single class of object. Assume that if most of these nearest points come from class c, then this new coming data is assumed to be from class c as well. For each data x_i, assume the set of its nearest neighbors to be NXI. Then NXI can be divided into $|CI|$ parts. The determination of class of x_i is based on selecting the number k, which enables the size of $|NXI_k|$ to be the biggest. Different neighbor data points will be assigned different weights, as their distances to the new coming data are different. This weight is used to reconstruct the current data from its neighbors and to calculate its low dimensional coordinate as well.

Algorithm 1 Learning Global Structure

$YI \leftarrow$ Global-Structure(XI, CI)
BEGIN
Divide XI into $|CI|$ parts;
$XI = \cup\ XI_k$
For each $\mathbf{x_i} \in XI$, denote the set of its nearest m neighbors including itself as LXI and denote the class of x_i as k;
BEGIN
For each x_s, $x_t \in LXI$, calculate intra-class weight metrics;
BEGIN

$$dist(x_s, x_t) = \| x_s - x_t \|$$

$$WXI_k(x_s, x_t) = \exp\{-\frac{dist(x_s, x_t)}{\mu}\}$$

END
END
For each a, $b = 1, 2, \ldots, |CI|$, $a \neq b$, $x_a \in XI_a$, $x_b \in XI_b$, calculate inter-class weight metrics;
BEGIN

$$DXI_{ab}(x_a, x_b) = \| x_a - x_b \|$$

$$WXI_{ab} = \arg\max_{P^T P=1} tr(P^T DXI_{ab})$$

END
Perform Isomap algorithm on XI based on WXI;
END

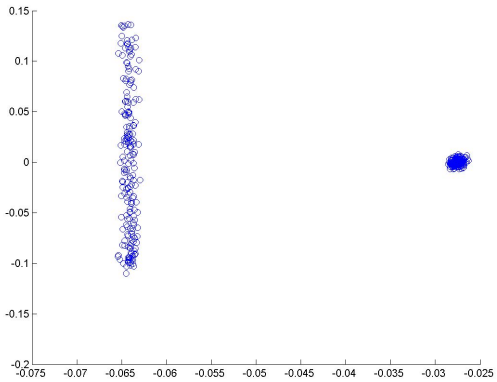

Fig. 3. Low dimensional structure of sample data for the mixed data composed of Swiss roll and Gaussian distributed data

Let ***xnew*** be the current new coming data, ***XN*** be the set of the nearest ***m*** neighbors of ***xnew***, and ***CI*** be the set of the classes that the elements of the data belong to, as described above. Obviously, ***XN*** is a subset of ***XI***. Let ***ynew*** be the corresponding low dimensional coordinate of ***xnew***, and ***YN*** be the set of corresponding low dimensional coordinates for each point in the set ***XN***. ***YN*** is a subset of ***YI***, correspondingly. Then the data point ***xnew***, the set ***CI***, and the sets both ***XN*** and ***YN*** are already known, while ***ynew*** is to be predicted using this algorithm.

In the first step, ***XN*** is divided into |***CI***| parts, each is composed of data from a single class. These parts are then denoted as $\boldsymbol{XN_i}$, $\boldsymbol{i}$ = 1, 2, … , |***CI***|. And the number of elements for each class $\boldsymbol{XN_i}$ is calculated. These numbers are denoted as $\boldsymbol{NUM_i}$, $\boldsymbol{i}$ = 1, 2, … , |***CI***|. Then the class ***c*** of this new coming data ***xnew*** can be derived from (6).

$$c = \arg\max_i | XN_i | \tag{6}$$

In this way, all the data from the set $\boldsymbol{XN_c}$ are used, while others are not considered as useful data for reconstructing the data ***xnew***.

In the second step, the weights for each data from the set $\boldsymbol{XN_c}$ are calculated. The weights are measured according to the distances between each point of the set $\boldsymbol{XN_c}$ and the data ***xnew***. For each data $\boldsymbol{x_i}$ from the set $\boldsymbol{XN_c}$, the distance is calculated using (7).

$$dist(x_i, xnew) = || x_i - xnew || \tag{7}$$

Then the weight for each data $\boldsymbol{x_i}$ is calculated according to the distances. The weight of data $\boldsymbol{x_i}$ is denoted as $\boldsymbol{W_i}$. And these weights can be derived from (8). In this equation, σ is a parameter to be determined by users.

$$W_i = \exp\{-\frac{dist(x_i, xnew)}{\sigma}\} \tag{8}$$

As these weights have been calculated, the last step is to predict the low dimensional coordinate ***ynew*** for data ***xnew***.

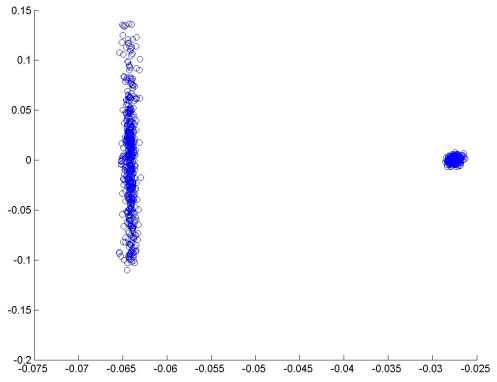

Fig. 4. Low dimensional structure for the mixed data composed of Swiss roll and Gaussian distributed data, including both training data and test data

Let $\boldsymbol{YN_c}$ be the set of corresponding low dimensional coordinates for data from the set $\boldsymbol{XN_c}$. For each data $\boldsymbol{y_i}$ from the set $\boldsymbol{YN_c}$, the weights for predicting ***ynew*** are the same as $\boldsymbol{W_i}$. Consequently, ***ynew*** can be calculated in the (9) as below.

$$ynew = \sum_i W_i \times y_i \tag{9}$$

The whole process to predict the low dimensional coordinates of each new coming data can be summarized in algorithm 2.

Algorithm 2 Prediction of Low Dimensional Coordinates

{***c***, ***ynew***} ← Dynamic-Prediction(***XI***, ***YI***, ***CI***, ***xnew***)
BEGIN
For each ***xnew***, denote its nearest neighbors in sample data as ***XN***, denote the low dimensional representation of ***XN*** as ***YN***;
BEGIN
$\boldsymbol{XN} = \cup\ \boldsymbol{XN_k}$
$\boldsymbol{YN} = \cup\ \boldsymbol{YN_k}$
$c = \arg\max_i | XN_i |$
END
For each $\boldsymbol{x_i}$ in $\boldsymbol{XN_c}$, calculate the weight between $\boldsymbol{x_i}$ and ***xnew***;
BEGIN
$dist(x_i, xnew) = || x_i - xnew ||$
$W_i = \exp\{-\frac{dist(x_i, xnew)}{\sigma}\}$
END
Predict the low dimensional coordinate ***ynew*** of ***xnew***;
$ynew = \sum_{y_i \in YN_c} W_i \times y_i$
END

TABLE I. CLASSIFICATION ACCURACY OF MIXED DATA COMPOSED OF SWISS ROLL AND GAUSSIAN DISTRIBUTED DATA

	Number of test data points	Number of correctly classified data points	Number of misclassified data points	Accuracy of classification
Swiss roll	400	362	38	90.50%
Gaussian distribute data	467	430	37	92.08%
Whole dataset	867	792	75	91.35%

IV. EXPERIMENTAL RESULTS

Two sets of data are used in our experiment, one of them is the mixed data composed of both Swiss roll and Gaussian distributed data, while the other is COIL data set. The former is synthetic data and the latter is real world data. Experimental results on these datasets are shown and discussed below.

A. *Mixed Data Com osed o ˙Swiss Roll and Gaussian Distributed Data*

The synthetic data we created is composed of 1300 data points, and can be divided into two parts. Firstly, 600 points are selected randomly on the surface of Swiss roll. Secondly, 700 points are selected randomly which obeys the Gaussian distribution.

As shown in Fig. 1, the intrinsic low dimensional structures of Swiss roll and Gaussian distributed data are different. The intrinsic structure of Swiss roll can be assumed to be a two-dimensional surface, while the Gaussian distributed data are from three-dimensional Euclidean space.

In this part of experiment, one-third of the mixed data, i.e. 433 data points are selected as training data. These training data are composed of 200 points selected randomly from the Swiss roll and 233 points selected randomly from the Gaussian distributed data points. The remaining 867 data points are used as the test data.

As discussed above, our algorithm is a two-step approach, the global structure of sample data is calculated in the first step, and the corresponding low-dimensional coordinates of test data are predicted in the second step. The result of the first step is shown in Fig. 3, and that of the second step is shown in Fig. 4.

As shown in our algorithm, the classes of new coming data are predicted dynamically. It is of great importance to guarantee that these data are classified correctly. Therefore the classification accuracy is tested.

The accuracy of classification for these test data is shown in table 1. As shown in this table, most of the test data in both classes are classified correctly. As discussed above, the intrinsic low dimensional structures of these two classes are different. Therefore the accuracies of classification of these two classes, which is relevant to the pair-wise distance inside each single class, are not the same.

TABLE II. CLASSIFICATION ACCURACY OF SELECTED DATA FROM COIL-20 DATASET

	Number of test images	Number of correctly classified images	Number of misclassified images	Accuracy of classification
Obj-3	48	46	2	95.83%
Obj-5	48	41	7	85.42%
Obj-6	48	36	12	75%
Obj-9	48	41	7	85.42%
Obj-19	48	44	4	91.67%
Whole dataset	240	208	32	86.67%

B. *COIL-20 Data*

COIL-20 is a dataset about images of Columbia University Image Library. This dataset is available in two versions. The version we used is that the background has been discarded and the image consists the smallest square that contains the object. This dataset is composed of 1440 images, representing 20 objects in different points of view.

In this part of experiment, five objects from COIL-20, which have the numbers 3, 5, 6, 9 and 19, are used. There are 360 images in total, as each object is composed of 72 images with similar points of view. Each instance is a 128*128 gray-scale image.

As shown in Fig. 2, the intrinsic low dimensional structures of these five classes are similar. The corresponding images with the same order in these classes are related to the same point of view. The images of each class form a sequence, which starts from a viewpoint and ends up with another viewpoint.

In this experiment, 24 viewpoints are selected to constitute the sample data. Therefore the sample data set is composed of 120 images, with 24 images from each class. And the remaining 240 images are used as test data. Each instance is viewed as a vector of size 1 * 16384.

Similar to the experiment of part ***A***, the algorithm is composed of both the global structure learning step and the low dimensional coordinates prediction step. The classification accuracy of the 240 images in the test data is shown in Table 2.

As shown in the table, most of the images are classified correctly. Although the low dimensional structures of these classes are similar, the classification accuracies among different classes are different as well. This is because the relation between each pair of data is relative to the pair-wise distances both inside each class and among different classes.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, an algorithm has been proposed to solve the out-of-sample data problem for multi-manifold

algorithms. The algorithm is composed of two parts, used to learn a global low dimensional multi-manifold structure and to predict low dimensional coordinates, respectively. In this algorithm, when new data come or some data change, it does not need to recalculate distances and weights between each pair of data in the whole dataset. As mentioned above, this algorithm can calculate the low dimensional coordinates of out-of-sample data effectively and efficiently.

However, some problems such as parameter selection remain to be solved. In the experiments, these parameters, such as the number of neighbors, are chosen by users, according to different situations in different problems. In our algorithm, when the class of a new data point has been observed before, its class and low dimensional presentation can be predicted efficiently. But when the new coming data point is from an unseen class, the algorithm may not solve it very well. This is also a problem remains to be solved.

For the future works, both manifold learning algorithms and multi-manifold learning algorithms will be further researched, especially to focus on how to select parameters more effectively and adaptively, such as the number of neighbors, and how to design new manifold learning algorithms.

ACKNOWLEDGMENT

This project was supported by Shenzhen Foundation for Basic Research JCYJ20120614150236998, by Shenzhen Engineering Lab on Intelligent Perception for Internet of Things, by Shenzhen Engineering Lab of Three-dimensional Media Technology, and by Shenzhen Peacock Plan.

REFERENCES

[1] M. Belkin, and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation,". Neural computation, vol 15, pp.1373-1396, 2003.

[2] D. L. Donoho, and C. Grimes, "Hessian Eigenmaps: Locally Linear Embedding Techniques for High-dimensional data," Proceedings of the National Academy of Sciences of the United States of America, vol. 100, pp. 5591-5596, 2003.

[3] A. B. Goldberg, X. Zhu, A. Singh and et al., "Multi-manifold Semi-supervised Learning," in the International Conference on Artificial Intelligence and Statistics, 2009, pp. 169-176.

[4] D. Gong, X. Zhao and G. Medioni, "Robust multiple manifolds structure learning," arXiv preprint arXiv:1206.4624, 2012.

[5] X. He and P. Niyogi, "Locality Preserving Projections," in Advances in Neural Information Processing Systems, 2003, pp.234-241.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with neural networks," Science, vol 313, pp. 504-507, 2006.

[7] K. Kim and D. Lee, "Inductive Manifold Learning Using Structured Support Vector Machine.", Pattern Recognition, vol. 47, pp. 470-479, 2014.

[8] T. Kohonen, "The Self-organizing Map," Proceedings of the IEEE, vol 78, pp. 1464-1480, 1990.

[9] S. Lafon and A. Lee, "Diffusion Maps and Coarse-graining: a Unified Framework for Dimensionality Reduction, Graph Partitioning, and Dataset Parameterization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 1393–1403, 2006.

[10] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, vol 290, pp. 2323–2326, 2000.

[11] L.K. Saul and S. T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," Journal of Machine Learning Research, vol. 4, pp. 119-155., 2004

[12] B. Schölkopf, A. Smola and K. R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," Neural Computation, vol 10 pp. 1299-1319, 1998.

[13] G. L. Scott and H. C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," Proceedings of the Royal Society of Biological Sciences, vol. 244, pp. 21-26, 1991..

[14] J. B. Tenenbaum, V. D. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, vol. 290, pp. 2319-2323, 2000.

[15] M. Torki, A. Elgammal and C. S. Lee, "Learning a Joint Manifold Representation from Multiple Data Sets," Proceedings of IEEE 2010 20th International Conference on Pattern Recognition (ICPR 2010), pp. 1068-1071, 2010.

[16] X. Wang, P. Tiňo and Fardal, "A Multiple Manifolds Learning Framework Based on Hierarchical Mixture Density Model," vol.5212 Berlin: Springer-Verlag Berlin, 2008, pp. 566-581.

[17] K. Q. Weinberger, F. Sha and L. K. Saul, "Learning a Kernel Matrix for Nonlinear Dimensionality Reduction," in International Conference on Machine Learning (ICML-04), 2004, pp. 839-846.

[18] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," SIAM Journal of Scientific Computing, vol 26, pp. 313-338, 2004.