Manifold Learning for Multivariate Variable-Length Sequences With an Application to Similarity Search

Shen-Shyang Ho, *Member, IEEE*, Peng Dai, *Member, IEEE*, and Frank Rudzicz, *Member, IEEE*

***Abstract*— Multivariate variable-length sequence data are becoming ubiquitous with the technological advancement in mobile devices and sensor networks. Such data are difficult to compare, visualize, and analyze due to the nonmetric nature of data sequence similarity measures. In this paper, we propose a general manifold learning framework for arbitrary-length multivariate data sequences driven by similarity/distance (parameter) learning in both the original data sequence space and the learned manifold. Our proposed algorithm transforms the data sequences in a nonmetric data sequence space into feature vectors in a manifold that preserves the data sequence space structure. In particular, the feature vectors in the manifold representing similar data sequences remain close to one another and far from the feature points corresponding to dissimilar data sequences. To achieve this objective, we assume a semisupervised setting where we have knowledge about whether some of data sequences are similar or dissimilar, called the instance-level constraints. Using this information, one learns the similarity measure for the data sequence space and the distance measures for the manifold. Moreover, we describe an approach to handle the similarity search problem given user-defined instance level constraints in the learned manifold using a consensus voting scheme. Experimental results on both synthetic data and real tropical cyclone sequence data are presented to demonstrate the feasibility of our manifold learning framework and the robustness of performing similarity search in the learned manifold.**

***Index Terms*— Application, embedding, feature extraction, isometric feature mapping (ISOMAP), longest common subsequence (LCSS), metric learning, similarity learning, similarity search, tropical cyclone.**

I. Introduction

MANY applications require comparing sequence data, including financial time series, audio sequences, and DNA sequences. With the advent of modern mobile technology, we are collecting more complex multivariate data sequences for pattern mining and analysis. For example, one might measure the similarity of multidimensional trajectories of moving objects or extract patterns from multiple sensors at various locations over time. The multivariate and variable-length nature of such data sequences makes their comparison and analysis challenging. In particular, conventional notions of similarity or distance between two data sequences (of different length, see Section III-A) are not metric. In many cases (if not all), the triangle inequality property is not valid in the data sequence space with such conventional distance measures. Without the triangle inequality, convergence theorems for metric spaces cannot be used for data sequences.

In this paper, we propose a framework to embed multivariate arbitrary-length data sequences into a manifold (Section II). Our proposed algorithm transforms the data sequences in a nonmetric data sequence space into feature vectors in a manifold that preserves the data sequence space structure. In particular, the feature vectors in the manifold representing similar data sequences remain close to one another and far from the feature points corresponding to dissimilar data sequences. To achieve this objective, we assume a semisupervised setting. In this setting, we have knowledge of the similarity (or dissimilarity) of sequences within a subset of data, called the instance-level constraints [1]. Using this information, one learns the similarity measure for the data sequence space and the distance measures for the manifold. Fig. 1 shows an example of tropical cyclone trajectory data (wind intensity and pressure not known) projected into the learned manifold. One observes that those in close proximity in the data sequence space remain close in the manifold. The four solid line trajectories correspond to the four circle representations in the manifold. The two dotted line trajectories correspond to diamond representations in the manifold.

To demonstrate the usefulness of the learned manifold, we propose a solution for similarity search in the learned manifold. Our approach allows a user query such as "List all tropical cyclones that are *similar* to tropical cyclones $s_1, s_2, \ldots, s_k$ and *dissimilar* to tropical cyclones $d_1, d_2, \ldots, d_l$." This user query requires the user to specify the tropical cyclones represented by multivariate variable-length sequences as the instance-level constraints. It enables users to perform: 1) event data sequence clustering or categorizing based on knowledge of limited number of events and 2) similar events identification for data retrieval and analysis. For example, a scientist provides the query system with a small set of tropical cyclones that have similar trajectories and wind intensity time series but traveling at different speeds; the system then returns tropical cyclone events from the past 20 years that exhibit similar characteristics and related satellite data for further analysis.

The outline of this paper is as follows. In Section II, we motivate and describe the manifold-learning problem for

Manuscript received October 1, 2013; revised January 15, 2015; accepted January 24, 2015. Date of publication March 13, 2015; date of current version May 16, 2016. This work was supported by NASA Advanced Information Systems Technology (AIST) program, the Singapore Ministry of Education under Grant RG41/12 and Grant RG18/14, and the NTU Start-Up-Grant.

S.-S. Ho is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ssho@ntu.edu.sg).

P. Dai and F. Rudzicz are with the Toronto Rehabilitation Institute—UHN, Toronto, ON M5G 2A2, Canada, and also with the Department of Computer Science, University of Toronto, Toronto, ON M5S 2J7, Canada (e-mail: derekpengdai@gmail.com; frank@cs.toronto.edu).

Digital Object Identifier 10.1109/TNNLS.2015.2399102

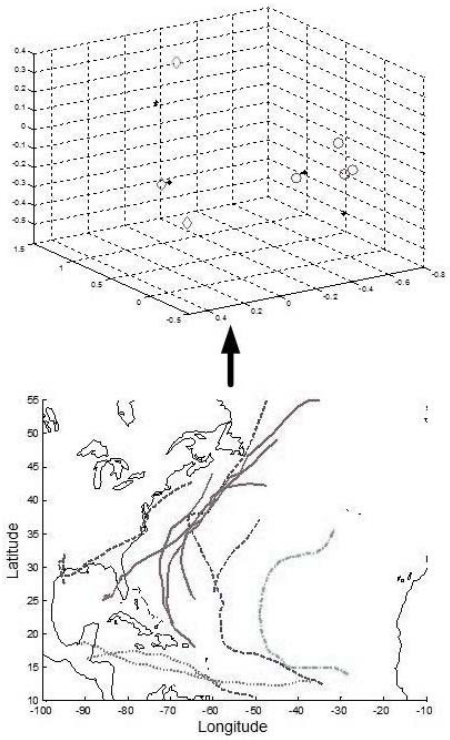

Fig. 1. Projecting multivariate arbitrary-length tropical cyclone trajectory sequences into the learned 3-D manifold. Feature values (e.g., wind intensity and pressure) are not shown.

data sequences. In Section III, we describe our proposed approach in detail and some needed background knowledge. In Section IV, we first describe the tropical cyclone data sequences from 2000 to 2008 used in the similarity search problem. Then, we describe how multivariate variable-length synthetic data are generated based on the tropical cyclone data sequences for the analysis of our proposed manifold-based similarity search approach. Experimental results are then presented to demonstrate the feasibility of our manifold learning framework and the robustness of performing similarity search in the learned manifold.

II. Manifold Learning for Data Sequences

Conventional manifold learning refers to nonlinear dimensionality reduction methods based on the assumption that [high dimensional] input data are sampled from a smooth manifold [2], so that one can embed these data into the [low dimensional] manifold while preserving some structural (or geometric) properties that exist in the original input space. This smooth manifold is a space that locally resembles a Euclidean space R^n. However, note that in practice, a manifold is nothing more than the underlying support of a data distribution, which is known only through a finite sample [3]. An embedding is a representation of a (topological) object (such as a manifold or a graph) in a certain space $[\cdots]$, such that its (topological or structural) properties are preserved [3].

Some of the more representative manifold learning algorithms (see [2] and references therein) are the isometric feature mapping (ISOMAP) [4], the locally linear embedding (LLE) [5], and the Laplacian eigenmaps [6]. Recently, Lin *et al.* [7] proposed a dimensionality reduction approach that seeks an embedding function whose gradient field is close to the parallel field relating to local isometric property in the data manifold. Manifold representations have been used in applications related to image pattern recognition [8], speech recognition [9], and text classification [10]. Li *et al.* [11] proposed understanding the dynamical process from video sequences in a manifold motivated by Lin *et al.* [12] on embedding a time series in another time-series manifold focusing on temporal correlation.

The general manifold learning framework proposed in this paper is as follows. Given a finite set X consisting of multivariate arbitrary-length data sequences that are partitioned into similar data sequence set S' and dissimilar data sequence set D', we learn a mapping

$$f : (X, d_1) \rightarrow (M, d_2)$$

such that d_1 is the learned similarity measure used in X and d_2 is a learned distance metric used in M, a low-dimensional (learned) manifold. Before performing similarity search given the instance-level constraints and a set of unlabeled data sequences U, we learn a mapping

$$g : (X \cup U, d_1) \rightarrow (M', d_2)$$

such that d_1 and d_2 are previously learned, and M' is a low-dimensional manifold. In addition, g and f preserve the same structural property induced by X. In particular, the representations of data sequences in S' should be close to one another and far away from the representations of data sequences in D' in both M and M'. Note that M and M' do not need to be the same metric space. While any conventional nonlinear dimensionality reduction approach (see [2]) can be used in the data sequence embedding, we use the distance preserving ISOMAP [4] in our discussion. It is an extension of linear embedding approaches (e.g., multidimensional scaling (MDS) [13]), which learns the global nonlinear geometric structure of the input data [4]. One capability of ISOMAP is its ability to discover nonlinear degrees of freedom that underlie complex natural observations, such as the trajectories and feature attributes of tropical cyclones.

In our problem, we assume that there exists some intrinsic low-dimensional manifold that the data sequence space can be projected onto. One main challenge for tasks related to data sequences is the nonmetric nature of the similarity measures, especially when one considers multivariate arbitrary-length data sequence comparison. However, based on [14], the estimated geodesic distance between two points in the original data sequence space (induced by X) converges to the true geodesic distance between the two points in M. Hence, the MDS step in ISOMAP will asymptotically recover the embedded (Mahalanobis) data structure [14]. By incorporating the dimensionality reduction step into similarity learning (Section III-C), we ensure that similarity learning is performed in a fixed dimensional metric space that preserves the nonlinear geometric structure of the data sequence space. This nice property ensures that the similarity measure learned is robust to variation in user-defined similar data sequences. In other words, the learned similarity measure

can still distinguish similar and dissimilar data sequences accurately as the feature and spatiotemporal variations increase for the user-defined similar data sequences. The fine-tuning of the nonmetric similarity measure needed for the neighborhood graph construction ensures that we have a manifold that preserves the structure in the original space defined by the user-defined instance-level constraints. This is achieved through the use of supervised metric learning.

III. Methodology

In Section III-A, we review similarity measures for data sequences. In particular, we provide background information about the longest common subsequence (LCSS) similarity measure. In Section III-B, we describe the soft longest common (SLC) subsequence similarity measure for data sequences used in our proposed algorithm. In Section III-C, we describe and discuss our proposed data sequence manifold learning approach given the instance-level constraints in detail. In Section III-D, we describe and discuss the data sequence similarity search in the learned manifold in detail.

A. Similarity Measures for Data Sequences

Many similarity measures have been proposed for data sequences (see [15] and references therein). The two main categories are the L_p-norm-based similarity measures and the elastic similarity measures. The former similarity measures are metric, but they assume fixed length data sequences and do not support local time shifting. The latter can be used to compare arbitrary-length data sequences and support local time shifting but they are not metrics. The common L_p-norm-based similarity measures use L_1, L_2, or L_∞ norm. The classical elastic measure that is first used to overcome the weakness of L_p norms is the dynamic time warping (DTW) [16]. The LCSS-based similarity measure was proposed to handle 2-D and 3-D arbitrary-length data sequences. The LCSS-based similarity measure is robust to noise and give more weight to the similar portion of the sequences [17]. The edit sequence on real sequence is robust to noise, shifts, and scaling of data [18]. Both time warp edit distance (TWEP) [19] and Edit Distance with Real Penalty (ERP) [20] (which combines L_1-norm and the edit distance) support local time shifting and they are metrics. However, both ERP and TWEP are derived for 1-D time series. Recently, Buchin *et al.* [21] and Liu and Schneider [22] proposed using additional context or semantic information for similarity analysis of trajectory sequence data.

It has been empirically shown that no single similarity measure outperforms others for time series [15]. For the purpose of neighborhood graph construction in manifold learning, one could use any similarity measure that supports local time shifting and multivariate data sequence. In this paper, we use a variant of LCSS-based similarity measure. First, we generalize the LCSS-based similarity measure [17], an edit distance-based elastic similarity measure, to multivariate data sequences. Consider two arbitrary-length multivariate spatiotemporal data sequences

$$A = \langle (t_{a,1}, a_{1,1}, \ldots, a_{m,1}), \ldots, (t_{a,n}, a_{1,n}, a_{m,n}) \rangle$$
$$B = \langle (t_{b,1}, b_{1,1}, \ldots, b_{m,1}), \ldots, (t_{b,l}, b_{1,l}, \ldots, b_{m,l}) \rangle$$

with m attributes and of length n and l, respectively. Define the similarity function $M1$ between A and B, given δ and $E = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)$, by

$$M1(A, B, \delta, E) = \frac{\text{LCSS}_{\delta,E}(A, B)}{\min(|A|, |B|)} \tag{1}$$

with the generalized LCSS defined by

$$\text{LCSS}_{\delta,E}(A, B) = \begin{cases} 0, & |A| \times |B| = 0 \\ 1 + \text{LCSS}_{\delta,E}(H(A), H(B)), & c_k > 0, |t_i - t_j| < \delta \\ \max\{\text{LCSS}_{\delta,E}(H(A), B), & \\ \quad \text{LCSS}_{\delta,E}(A, H(B))\}, & \text{otherwise} \end{cases} \tag{2}$$

such that $H(A)$ is the sequence $\langle (t_{a,1}, a_{1,1}, \ldots, a_{m,1}), \ldots, (t_{a,n-1}, a_{1,n-1}, a_{m,n-1}) \rangle$ for any data sequence A of length n and

$$C = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} e_1 - |a_{1,t_i} - b_{1,t_j}| \\ \vdots \\ e_m - |a_{m,t_i} - b_{m,t_j}| \end{pmatrix}$$

for some predefined δ and E.

B. Soft Longest Common Subsequence

To better reflect the similarity between multivariate data sequences, we further extend the generalized LCSS-based similarity measure to the SLC subsequence. The classical LCSS is a kind of hard-decision encoder. In other words, the LCSS similarity measure counts each similar data (point) pair as 1 when the constraints C and $|t_i - t_j| < \delta$ are satisfied [see $1 + \text{LCSS}_{\delta,E}(H(A), H(B))$ in (2)].

This leads to a common hard-/soft-decision problem. For example, given two pairs of data points, (a_{1,t_i}, b_{1,t_j}) with $\{c_k = 0.1, |t_i - t_j| < \delta\}$ and $(a'_{1,t'_i}, b'_{1,t'_j})$ with $\{c'_k = 10, |t'_i - t'_j| < \delta\}$, obviously (a_{1,t_i}, b_{1,t_j}) should be closer than $(a'_{1,t'_i}, b'_{1,t'_j})$. However, due to $1 + \text{LCSS}_{\delta,E}(H(A), H(B))$ in (2), they are quantitatively similar. Hence, we modified the LCSS similarity counting process such that each new count takes on a value between 0 (exclusive) and 1 (inclusive) as follows:

$$\text{SLC}_{\delta,E}(A, B) = \begin{cases} 0, & |A| \times |B| = 0 \\ \min\left(1 - \frac{c_k}{e_k}, 1\right) & \\ + \text{SLC}_{\delta,E}(H(A), H(B)), & c_k > 0, |t_i - t_j| < \delta \\ \max\{\text{SLC}_{\delta,E}(H(A), B), & \\ \quad \text{SLC}_{\delta,E}(A, H(B))\}, & \text{otherwise} \end{cases} \tag{3}$$

this modification gives more credits to those more similar pairs (i.e., $c_k \approx e_k$), while the less similar pairs (i.e., $c_k \approx 0$ or $c_k < 0$) are assigned smaller values. This extra information provides more reliable similarity count for each data point, and therefore, better reflects the true similarity between two multivariate data sequences. In particular, in the presence of corrupted/noisy data, a soft-decision approach generally performs better than its hard-decision counterpart [23], [24].

Similar to (1), we define the similarity function $M2$ between A and B, given δ and $E = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)$, by

$$M2(A, B, \delta, E) = \frac{\text{SLC}_{\delta,E}(A, B)}{\min(|A|, |B|)}. \quad (4)$$

To have good performance, the parameters δ and E have to be tuned according to the specific application. One concludes from Example 1 that the LCSS-based similarity measure $M1$ and SLC-based similarity measure $M2$ are sensitive to the parameters δ and E. Moreover, $M2$ shows more variability in the similarity values than $M1$ (compare Example 1.2 and Example 1.3). When exact matching is required [$\delta = 0$ and $E = (0)$], $M1$ and $M2$ similarity measures are identical (see Example 1.4). In addition, note that both similarity measures can have similar data point correspondence (or pairing).

Example 1: Given two variable sequences, $A = \langle(0_{a,1}, 0_{1,1}), (0.5_{a,2}, 1_{1,2}), (1_{a,3}, 3_{1,3}), (1.5_{a,4}, 1_{1,4})\rangle$ and $B = \langle(0_{b,1}, 1_{1,1}), (1_{b,2}, 2_{1,2}), (2_{b,3}, 1_{1,3})\rangle$, where $d = 1$ and $m = 0$.

1) $\delta = 1$ and $E = (0.5)$.
   $\text{LCSS}_{\delta,E}(A, B) = 2$; $M1(A, B, \delta, E) = 0.67$.
   $\text{SLC}_{\delta,E}(A, B) = 2$; $M2(A, B, \delta, E) = 0.67$.
   Possible correspondence: $1_{1,2} \to 1_{1,1}$, $1_{1,4} \to 1_{1,3}$.
2) $\delta = 1$ and $E = (1)$.
   $\text{LCSS}_{\delta,E}(A, B) = 3$; $M1(A, B, \delta, E) = 1$.
   $\text{SLC}_{\delta,E}(A, B) = 1$; $M2(A, B, \delta, E) = 0.33$.
   Possible correspondence: $0_{1,1} \to 1_{1,1}, 1_{1,2} \to 2_{1,2}, 1_{1,4} \to 1_{1,3}$.
3) $\delta = 1$ and $E = (1.5)$.
   $\text{LCSS}_{\delta,E}(A, B) = 3$; $M1(A, B, \delta, E) = 1$.
   $\text{SLC}_{\delta,E}(A, B) = 1.67$; $M2(A, B, \delta, E) = 0.56$.
   Possible correspondence: $0_{1,1} \to 1_{1,1}, 1_{1,2} \to 2_{1,2}, 1_{1,4} \to 1_{1,3}$.
4) $\delta = 0$ and $E = (0)$.
   $\text{LCSS}_{\delta,E}(A, B) = 0$; $M1(A, B, \delta, E) = 0$.
   $\text{SLC}_{\delta,E}(A, B) = 0$; $M2(A, B, \delta, E) = 0$.

C. Similarity Parameter Learning for Data Sequence Embedding

1) Algorithm Description: One can easily embed data sequences based on the construction of a neighborhood graph using the SLC-based similarity measure $M2$ with any parameters E and δ, and then apply a manifold learning approach. Here, we utilize some limited side information (user-defined instance-level constraints [1]) to obtain a better embedding such that the (unseen) similar data sequences are in close proximity in the manifold. This is achieved using a learning approach to learn the parameters of $M2$.

Distance metric learning [25]–[28] aims to learn a distance metric (parameters) for the input data space from a collection of similar/dissimilar points. This learned distance metric preserves the distance relation among the training data. However, not all similarity functions satisfy the metric properties. Moreover, it has been shown empirically that nonmetric similarity functions have better performance than the metric similarity functions for problem, such as similarity search problem for time series or data sequences [15]. Recently, a hashing approach has been proposed that considers the learned Mahalanobis distance metric for scalable similarity search on image and systems data sets [29]. Yu and Gertz [30] proposed learning the DTW distance by the direct application of Xing *et al.*'s [25] approach on the Mahalanobis distance with the two trajectories interpolated to the same length in the input space.

Most metric learning methods attempt to learn a distance metric from side information, which is often available in the form of pairwise constraints; that is, pairs of similar data points and pairs of dissimilar data points. The information of similarity or dissimilarity between a pair of examples can easily be collected from the label information in supervised classification. The most intuitive learning approach to use for our problem is the one proposed by Xing *et al.* [25]. In their approach, the metric learning problem is posed as a convex optimization problem with the constraints given by the must-link data pairs and cannot-link data pairs. The objective is to find the matrix representing the Mahalanobis metric that allows the similar data points close to one another and the dissimilar data points far away from the similar data points.

The main differences in the problem setting are: 1) the use of SLC-based similarity $M2$ (that do not satisfy all the metric properties) during the neighborhood graph construction and 2) the arbitrary-length multidimensional data sequence pair constraints in the original data sequence space. Moreover, the optimization step has to be modified for the integer time parameter δ in SLC. Our proposed algorithm can handle flexible length sequence comparison.

Xing *et al.*'s metric learning framework is extended to learn parameters of nonmetric similarity for generic (arbitrary length and multidimensional) data sequences and distance metric in the learned manifold simultaneously. A dimensionality reduction component is integrated into the Xing *et al.*'s framework to facilitate metric learning in a fixed low dimensional metric space induced by the nonmetric $M2$ for neighborhood graph construction. Let f_{M2} be a mapping from the data sequence space to a fixed low-dimensional space M induced by $M2$. Let S and D be the set of must-link pairs and the set of cannot-link pairs, respectively. To perform the parameter learning for similarity and distance measures, we use a variant of the objective function introduced in [25] as such

$$\min_{E,\delta} \sum_{(x_i,x_j)\in S} d(x_i, x_j)^2$$
$$\text{s.t.} \sum_{(x_i,x_j)\in D} d(x_i, x_j) \geq 1 \quad P > \mathbf{0} \quad (5)$$

where S and D are similar set and dissimilar set, respectively; $P = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m, \delta) \in (R^+)^m \times Z^+$ is the parameter vector for SLC in the data sequence space and

$$d(x_i, x_j) = \sqrt{[f_{M2}(x_i) - f_{M2}(x_j)]^T A [f_{M2}(x_i) - f_{M2}(x_j)]}$$

is the Mahalanobis distance metric for a manifold M and A is a positive semidefinite matrix. There are two main reasons for the constraint $\sum_{(x_i,x_j)\in D} d(x_i, x_j) \geq 1$. First, it helps to ensure that dissimilar sequences are far away. Second, it prevents a

Algorithm 1 Data Sequence Similarity Parameter and Manifold Learning

Input: S', similarity set; D', dissimilar set; K.
Output: Manifold, M; Parameters: P, A
1: Initialize $P := [0.1, \ldots, 0.1, 1]$, $A = I$, the identity matrix;
2: Construct the "must-link" pair set, S and "cannot-link" pair set, D using S' and D';
3: Compute the $K-$nearest neighbor graph using $M2$ defined by P for data sequences in S' and D';
4: Compute the shortest path distance between all data sequences using Dijkstra's algorithm and $M2$ defined by P;
5: Apply MDS to construct a fixed low dimensional manifold M;
6: Compute the Mahalanobis distances for data sequence pairs in S and in D, separately in M;
7: Compute objective function (6);
8: Update P or A;
9: Repeat Step 3 to 8 until $|g_{i+1} - g_i| < \gamma$;

trivial solution such that the data sequence space converges to a single point.

Solve the unconstrained minimization problem

$$g(S, D, P, A) = \sum_{(x_i, x_j) \in S} d(x_i, x_j)^2 - \log\left(\sum_{(x_i, x_j) \in D} d(x_i, x_j)\right) - \sum_{j=1}^{m} \log \epsilon_j - \log \delta. \quad (6)$$

The additional terms are used to control the magnitude of parameter vector P.

The coordinate descent method [31] is used for the minimization step to avoid gradient computation for P. $g(\cdot)$ is minimized along one coordinate direction at each iteration. In our implementation, the coordinate is selected based on

$$\arg\min_{P_i \in I} g(S, D, P_i, A) \quad (7)$$

such that $I = \{P_i = (\epsilon_1^k, \ldots, \epsilon_i^{k+1}, \ldots, \epsilon_m^k, \delta) | \epsilon_i^{k+1} = \epsilon^k + h_i e_i\}$, e_i is the ith unit vector and h_i is a fixed small value. We fix δ and A and allow search in ϵ_i space first for all i. When the global minimum (at fixed δ and A) is achieved at P^*, we optimize A using the gradient approach in [25] at fixed δ and P^*. When a global minimum is reached, we perform minimization with step size 1 on δ. Initialization of P starts near the zero vector and $A = I$ such that I is the identity matrix. One notes that as ϵ_i and δ increase, the similarity value between two data sequences increases.

Algorithm 1 provides a high-level description of the similarity parameters (P and A) and manifold learning procedure. In Step 2, we construct the must-link pairs by pairing up all the data sequences in the set S' of similar data sequences. Hence, $|S| = |S'|\dot{(}|S'| - 1)/2$. To construct the cannot-link pairs, the data sequences in the set D' of dissimilar data sequences are all paired up first. Then, each data sequence in S' is paired with all the data sequences in D'. Hence, $|D| = |D'|\dot{(}|D'| - 1)/2 + |S'| \cdot |D'|$. One does not need a data sequence in D' to be dissimilar to the other data sequences in D'. However, in this paper, we assume that data sequences in S' to be close together, and data sequences in D' to not only far from data sequences in S' but also data sequences in D' to be spaced out. To remove this assumption, one may have data sequences in D' close together forming clusters, which we want to avoid when performing parameter learning.

Lines 3 to 5 are the steps for the ISOMAP algorithm. Step 3 computes the K-nearest neighbor graph using $M2$ defined by P for the data sequences in S' and D'. Step 4 computes the geodesic distance between all data sequences using Dijkstra's algorithm and $M2$ defined by P. Step 5 constructs the low-dimensional manifold M using MDS. Lines 7 computes the objective function (6). Line 8 updates the P parameter or A parameter depending on the earlier described implementation of the coordinate descent method. Line 9 is the stopping criterion based on the absolute difference between two consecutive objective function values, g_{i+1} and g_i. Algorithm 1 halts when the criterion value is less than γ.

2) Computational Complexity: The computational complexity of Algorithm 1 is analyzed by breaking it down into three components: 1) dissimilarity matrix construction by computing $M2$ values for all sequence pairs (Step 3); 2) ISOMAP (Steps 4 and 5); and 3) the coordinate descent method (Steps 6–8). Based on [17, Lemma 1], the (dis)similarity matrix construction can be computed in $O(s^2 \delta l)$, where s is the number of data sequences and $l = 2\max(l_1, \ldots, l_s), l_i, i = 1, \ldots, s$ are the sequence lengths. For the ISOMAP algorithm, the computational complexity is $O(s^3)$. The convergence rate of the coordinate descent method is similar to steepest descent. Even though this can be perceived to be slow, it is still effective for practical purposes [31]. Note that the convergence rate for the gradient approach in [25] is faster than the coordinate descent method. During each iteration, one needs to construct the dissimilarity matrix only once and run the ISOMAP algorithm. From an implementation perspective, dissimilarity matrix construction (Step 3) is the most expensive step as $s < l$ and $\delta \geq 1$. Here, $s < l$ as the number of user-defined data sequences for the similarity query is assumed to be limited.

The neighborhood graph is generated using $M2$. It has to be noted that for each iteration only part of the similarity distance values ($M2$) change. This is because SLC tolerates certain amount of mismatch between the comparison pairs [see (3)]. The parameters, δ and E in (4), are used to tune the amount of mismatch the similarity measure can tolerate. Therefore, the topology of the neighborhood graph changes gradually as the algorithm iterates. Thus, we do not necessarily need to repeatedly construct the entire neighborhood graph. The changes in the neighborhood graph can be interpreted as either an insertion of new edges or a removal of existing edges in the neighborhood graph. Previous studies [32], [33] have discussed the dynamic shortest path updating algorithms

Algorithm 2 Data Sequence Voting-Based Similarity Search on a Manifold M'

Input: S', similarity set; U', the set of unlabeled data sequences; P and A, learned parameters; C, user-defined ranking cut-off; K.

Output: O, the set of similar data sequences.

1: Compute the $K-$nearest neighbor graph using $M2$ defined by P for data sequences in S' and U';
2: Compute the shortest path distance between all data sequences using Dijkstra's algorithm and $M2$ defined by P;
3: Apply MDS to construct a fixed low dimensional manifold M';
4: Compute Mahalanobis distance vector $D_s = \{d_{su}\}_{u=1}^{|U'|}$

$$d_{su} = ||s - u||_A$$

for each $s \in S'$, $\forall u \in U'$ in M';
5: $\bar{D}_s = \text{sort}(D_s) = \{d_{\bar{u}_{s1}}, \ldots, d_{\bar{u}_{sC}}, \ldots, d_{\bar{u}_{s|U'|}}\}$ such that $d_{\bar{u}_{s1}} < d_{\bar{u}_{s2}} < \cdots < d_{\bar{u}_{sC}} < \cdots < d_{\bar{u}_{s|U'|}}$, for each $s \in S$;
6: $R = \{\bar{u}_{si} | \bar{u}_{si} \in U', d_{\bar{u}_{si}} \leq d_{\bar{u}_{sC}}, s \in S'\}$;
7: $N_u = \#\{v \in R : v = u\}$ for all $u \in U'$;
8: $O = \{u | u \in U', N_u > \frac{|S'|}{2}\}$;

that deal with these graph changes. For practical applications such as the similarity search task in Section III-D, one can store the learned parameters. Then, one progressively updates the parameters as more data sequences become available. Therefore, the computational complexity for Algorithm 1 would be nearly the same as that for ISOMAP. Applying steps similar to incremental ISOMAP [32]–[34] in our proposed algorithm can further improve its computational complexity.

D. Data Sequence Similarity Search in the Manifold

To select the most similar data sequences from a set U of unlabeled data sequences, we use a voting scheme that polls from the data sequences in the similar set S' in the manifold using the Mahalanobis distance with the learned parameter matrix A. Note that the manifold is constructed based on the neighborhood graph using the SLC-based similarity and the learned parameter vector P. The voting scheme is a combination of ranking the unlabeled data sequences and a majority vote decision based on the ranking. For each data sequence $s \in S'$, the unlabeled data sequences in U' are ordered based on their Mahalanobis distances from s. If an unlabeled data sequence $u \in U'$ is ranked as among the C most similar (or closest) data sequences to s, it will receive a vote from s. If u received more than $|S'|/2$ votes, it is considered similar to the data sequences in S'.

Algorithm 2 shows our voting approach for similar data sequences selection using Mahalanobis distance in the manifold M derived from the neighborhood graph using the learned SLC-based similarity measure $M2$ from Algorithm 1. Lines 1–3 are the steps for the ISOMAP algorithm using the learned SLC-based similarity measure $M2$. Line 4 computes the Mahalanobis distances from all the unlabeled data sequences in U' to each data sequence in the similar set S' in the manifold M. Line 5 sorts and ranks the unlabeled data sequences for each data sequences in S' using the computed Mahalanobis distances in Step 4. Line 6 gathers the C most similar unlabeled data sequences for each data sequence in S' into a single set. Line 7 counts the number of times an unlabeled data sequence is among the top C unlabeled data sequences closest to each data sequence in S'. Line 8 is a voting scheme, which selects a data sequence if it is ranked among the top C data sequences for more than half the data sequences in S'.

Other variants of the voting scheme can also be used for the similarity search. In particular, vote counts can be based on the C farthest from the data sequences in the dissimilar set or vote counts based on ranking data sequences from both similar and dissimilar sets.

IV. Experimental Results

In Section IV-A, we describe the tropical cyclone data set and how synthetic similar data sequences are generated from the tropical cyclone data set. In Section IV-B, we show the feasibility of similarity search (Algorithm 2) in the learned manifold based on the proposed similarity parameter learning approach (Algorithm 1) and its robustness to variability in the instance-level constraints using the synthetic data set and the tropical cyclone data set. In Section IV-C, we illustrate a scenario when a user provides a similar tropical cyclone set S' and a dissimilar tropical cyclone set D', and uses Algorithm 2 to search for similar tropical cyclones from a set of unlabeled tropical cyclone data sequences. Preliminary results on tropical cyclone similarity search were published in [38].

A. Data Set Description

1) Tropical Cyclone Data Set: A tropical cyclone event is a nonfrontal synoptic scale low-pressure system over tropical or subtropical waters with organized convection and definite cyclonic surface wind circulation.[1] The frequently used term hurricane describes a high-intensity tropical cyclone with sustained surface wind intensity equal or >119 km/h. Some examples of similarity of interest to scientists and meteorologists are track similarity [Hurricane Audrey (1957) and Hurricane Rita (2005)[2]], strength similarity (Hurricane Katrina (2005) and Hurricane Camille (1969) [35]) and hurricane origin [Cape Verde hurricanes: Hurricane Isabel (2003), Hurricane Floyd (1999), and Hurricane Hugo (1989)[3]].

A tropical cyclone event data sequence consists of both the trajectory and the feature attributes. A trajectory is the path a moving object follows through space and time. It is described by: 1) spatial attributes (latitude and longitude) and 2) temporal attributes (year, day, and time). The feature attributes

[1] http://www.aoml.noaa.gov/hrd/tcfaq/A1.html

[2] National Weather Service Forecast Office. http://www.srh.noaa.gov/lch/rita/rita_audrey.php

[3] National Climatic Data Center (NCDC). Climate of 2003: Comparison of Hurricane Floyd, Hugo, and Isabel. http://www.ncdc.noaa.gov/oa/climate/research/2003/fl-hu-is-comp.html

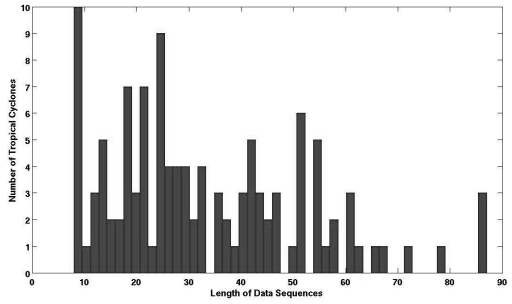

Fig. 2. Histogram for the data sequence length of tropical cyclones occurring from 2000 to 2008.

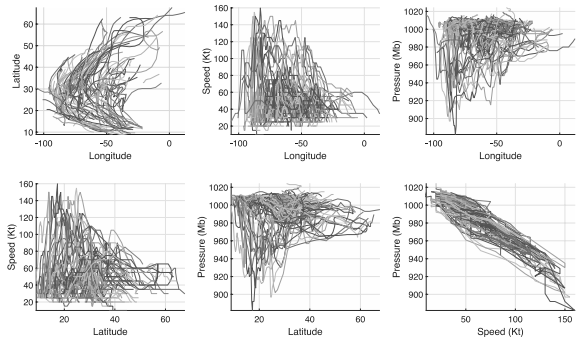

Fig. 3. Relationship between any two attributes in the data sequences for tropical cyclones occurring in the North Atlantic Ocean from 2000 to 2008.

are: 1) the maximum sustained wind intensity (knots) and 2) the minimum central pressure (millibar). Two consecutive data vectors in a data sequence are 6 h apart.

One can retrieve tropical cyclone and some subtropical cyclone event data sequences from the NOAA Coastal Services Center website[4] for both the North Atlantic Ocean and the Eastern North Pacific Ocean from 1851 to present. For this paper, 116 tropical cyclones occurring in the Atlantic Ocean from 2000 to 2008 and synthetic data sequences generated based on the 116 tropical cyclones are used in our experiments.

The tropical cyclone data sequences have arbitrary length. From Fig. 2, one sees that most of the data sequences consist of between 10 and 60 data vectors for tropical cyclone occurring in the North Atlantic Ocean between 2000 and 2008.

The top left graph in Fig. 3 shows the trajectories for the tropical cyclones. From the bottom right graph, one observes that there is an anticorrelation between the minimum central pressure and the maximum sustained wind intensity. Hence, we need only use one of the two feature attributes in our manifold learning. In this paper, we use the maximum sustained wind intensity.

2) Synthetic Data Set: Synthetic data are used to test the robustness of our approach as the variation in the similarity set increases. Synthetic data sequences are generated based on real tropical cyclone sequences described in Section IV-A1. For each experimental trial, one similar set S' of 10 synthetic tropical cyclone sequences is generated as follows.

1) Randomly pick one data sequence s_q from the real tropical cyclone event sequence set.
2) Specify a threshold γ_a for each trajectory and feature attribute a, so that s_q is bounded by a volume tube with radius γ_a in each attribute dimension and s_q is the tube center.
3) Specify a translational threshold η so that a generated data sequence can shift at most η.
4) Generate each of the 10 new similar data sequences as follows.
   a) Randomly assigned an integer length l_{new} to a new data sequence so that l_{new} is between $l - t$ and $l + t$, where l is the length of s_q and t is a fixed integer.
   b) Randomly generate l_{new} points such that they fell in the volume tube described in Step 2.
   c) Randomly shift the generated points satisfying the constraint in Step 3.

Then, we randomly pick 30 data sequences from the tropical cyclone event sequence set, excluding s_q, and include them into the dissimilar set D'.

B. Similarity Search in a Manifold

In our experiment, since $|S'| = 10$ and $|D'| = 30$, we have $|S| = (10 \cdot 9)/2 = 45$ and $|D| = (30 \cdot 29)/2 + 10 \cdot 30 = 2040$, respectively, according to Section III-C. Moreover, γ_a is varied from 1 to 2 for all a, $\eta = 1$, and $t = l$. For each γ_a value, we perform 20 trials.

Using the procedure for generating a set of similar data sequences in Section IV-A2, we generate another 100 positive testing examples. Eighty five data sequences, excluding the 30 in D' and s_q, from the real tropical cyclone event sequence set are used as negative testing examples. Hence, we have $|U'| = 185$. K is set to 10 for the ISOMAP algorithm, the manifold dimension is fixed at 2 (for visualization purposes).

Accuracy of similarity search in manifold is computed as follows. First, similarity values between s_q and all the data sequences in U' are computed. Then, the similarity values are sorted. The positive testing examples should be among the 100 closest to s_q. Hence

$$\text{Accuracy} = \frac{\#\{p | p \in \text{Pe}, S_p \leq S_{|\text{Pe}|}\} + \#\{n | n \in \text{Ne}, S_n > S_{|\text{Pe}|}\}}{|\text{Pe}| + |\text{Ne}|}$$

where Pe and Ne are the sets of positive and negative testing examples, respectively; S_p and S_n are the similarity values for a positive example p and a negative example n, respectively; and $S_{|\text{Pe}|}$ is the similarity value of the sorted value at position |Pe| (assuming sorting in increasing order).

Fig. 4 shows experimental results when no parameter learning is performed. A fixed $P = (1, 1, 5, 1)$ and A defined as the identity matrix (i.e., Euclidean distance) are used. As γ_a increases, the SLC similarity score decreases. In other words, as the data sequences in S' become more diverse, the SLC similarity measure using a fixed P is less likely to measure similarity accurately for the data sequences that are generated from the same distribution as those in S'. Moreover, one observes that the accuracy variability increases as γ_a increases.

Next, we compare the similarity search accuracies in a manifold using a predefined similarity measure or a learned

[4]http://csc-s-maps-q.csc.noaa.gov/hurricanes/

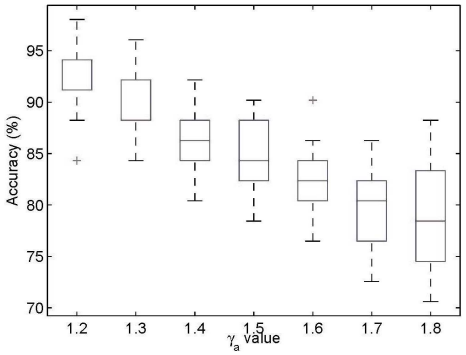

Fig. 4. Similarity search accuracy in a manifold with predefined similarity/metric parameters.

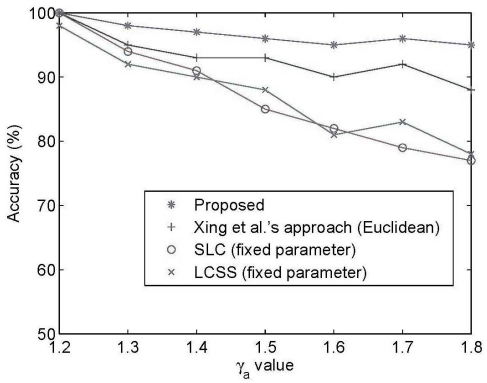

Fig. 5. Similarity search accuracy comparison.

similarity measure to construct the k-nearest neighbor graph. Similarity search results of data sequences at a fixed length in our learned manifold are compared with the similarity search results from the following constructed manifolds.

1) Manifold based on similarity measure learned from using Xing *et al.*'s metric learning approach with Euclidean distance.
2) Manifold using predefined parameters for the SLC-based similarity measure (use the optimal parameter at $\gamma_a = 1.2$).
3) Manifold using predefined parameters for the LCSS similarity measure (use the optimal parameter at $\gamma_a = 1.2$).

Fig. 5 shows the similarity search performance comparison in the two learned manifolds and the two manifolds constructed using predefined similarity measures.

It can be seen that similarity search in manifold learned using our proposed approach shows consistently better results than similarity search in manifolds using predefined similarity measures or using Xing *et al.*'s metric learning approach when the variability of the similar data sequences increases (i.e., γ_a increases). At $\gamma_a \leq 1.2$, because the synthetic similar data sequences are very close in the manifold for any similarity measure, similar searches using different manifolds show perfect accuracy, i.e., 100%. As γ_a increases, the similarity search accuracy drops. Manifolds constructed using learned similarity measures show more robustness. On the other hand, the similarity search in a manifold constructed using predefined similarity measures show significant drop in accuracy. This is consistent with the observations in Fig. 4. As γ_a increases, the similarity score becomes more variable and inconsistent without similarity parameter learning. As a result, the similarity search task becomes more difficult.

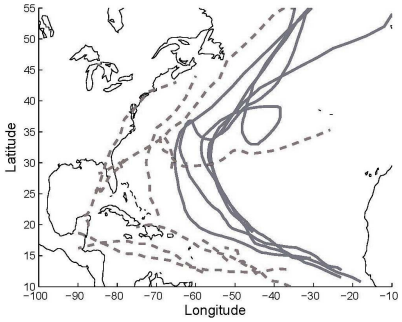

Fig. 6. Solid lines: five data sequences in the similar set S'. Dashed lines: ten data sequences in the dissimilar set D' (trajectories only).

C. Application: Tropical Cyclone Similarity Search With User-Defined Prior Knowledge

One application of our proposed data sequence manifold learning is tropical cyclone similarity search with user-defined prior knowledge. In particular, this application corresponds to the query "List all tropical cyclones that are *similar* to tropical cyclones in S' and *dissimilar* to the tropical cyclones in D'." For this task, we first include five tropical cyclone event data sequences from the real tropical cyclone event data sequence set into S' and include another 10 data sequences into D'. Fig. 6 shows the sets S' and D' used for Algorithm 1. We then include the other one hundred and one tropical cyclone events into U'. For both Algorithm 1 and 2, $K = 15$. For Algorithm 2, $C = 5$.

Fig. 7 shows the five most similar data sequences for each of the five data sequences in S' based on Step 6 in Algorithm 2. Fig. 8 shows the two trajectories (solid lines) and wind intensity time series of the output from Algorithm 2 together with the trajectories of the five similar data sequences. The two output trajectories clearly overlap with the selected similar set and one of the output trajectories is very short. To return data sequences with lengths comparable with those in S', one could filter out short data sequences in U' by including an additional criterion

$$\frac{\min(|A|, |B|)}{\max(|A|, |B|)} \geq t \tag{8}$$

for a data sequence B in U' with $A \in S'$ and $t = 0.5$ (say), before step 1 in Algorithm 2. When this criterion is included, the two output trajectories have comparable lengths, as shown in Fig. 9(a) instead of the ones in Fig. 8. On the other hand, if one wants to consider only short data sequences, one can use the criterion

$$\frac{\min(|A|, |B|)}{\max(|A|, |B|)} < t. \tag{9}$$

Fig. 9(b) shows three (short) output trajectories using the same input S' and D'.

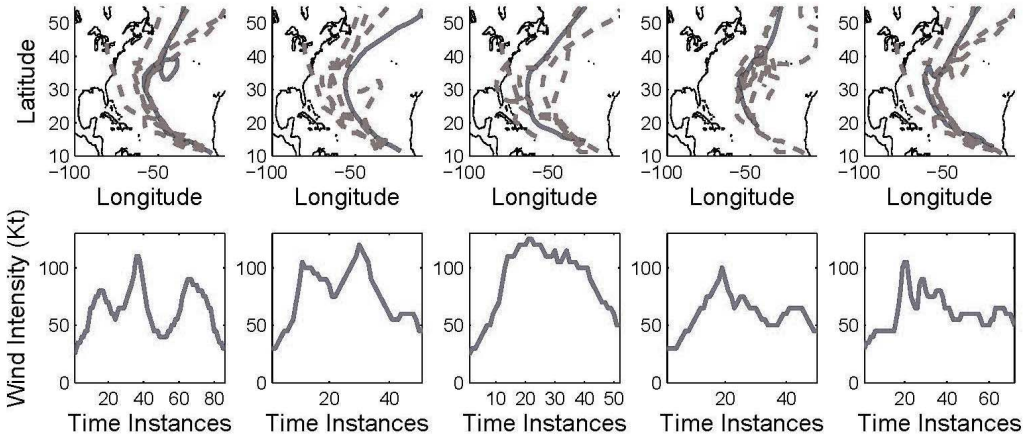

Fig. 7. Five data sequences (trajectories [top, solid lines] and intensities [bottom]) in the similar set S' and their corresponding five most similar unlabeled data sequences (trajectories [top, dashed lines]).

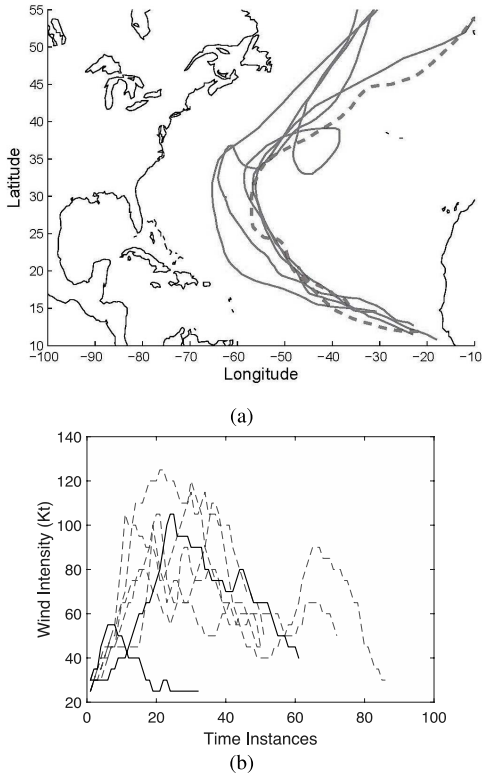

Fig. 8. (a) Five data sequences (solid lines) in the similar set S' and the two output trajectories (dashed lines) using Algorithm 2. (b) Wind intensity time series from the similar set S' (dashed lines) and the two wind intensity time series from output data sequences (solid lines).

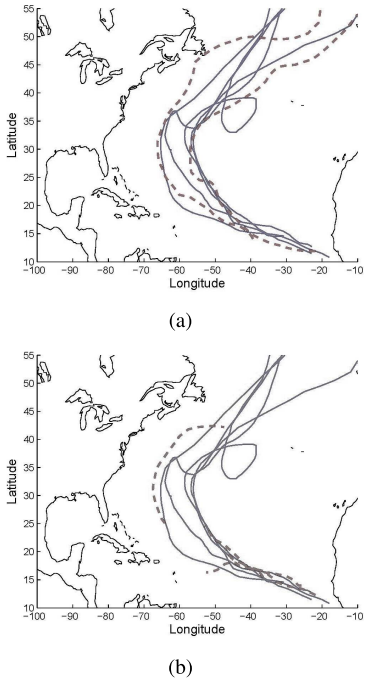

Fig. 9. When $t = 0.5$, the output trajectories (dashed lines) based on (a) criterion (8) and (b) criterion (9), given the five data sequences (solid lines) in the similar set S'.

Fig. 10 shows some seemingly similar data sequences to the data sequences in S' and their corresponding N_u values (see Step 7, Algorithm 2). One notes that N_u needs to be >2 to be output by Algorithm 2. One observes from Figs. 8 and 10 that the initial subsequences of those data sequences having higher N_u (≥ 2) tend to be very similar to those initial subsequences of data sequences in S'. In addition, from Fig. 7, one observes that a data sequence may be similar to some data sequences that look different (e.g., the third and the fourth sequences). This is because our proposed algorithm works with all the dimensions of the input data, which means it not only considers the geographic relationship (what we see in the visualization figures) but also make use of other dimensions (e.g., wind speed). Some sequences may be similar in terms of geographic trajectories but significantly different in wind speech or other entries. This affects the N_u value, which is used in making selection decision in Algorithm 2. Moreover, the user-defined ranking cutoff, C, also affects the number of data sequences selected. In other words, with higher C and a fixed $|S'|$, there will be more data sequences selected (Fig. 11).

In Section IV-B, we show that the similarity search performance in the manifold learned by our proposed algorithm using SLC is competitive against the similarity search in manifold learned by the other metric learning approach or in

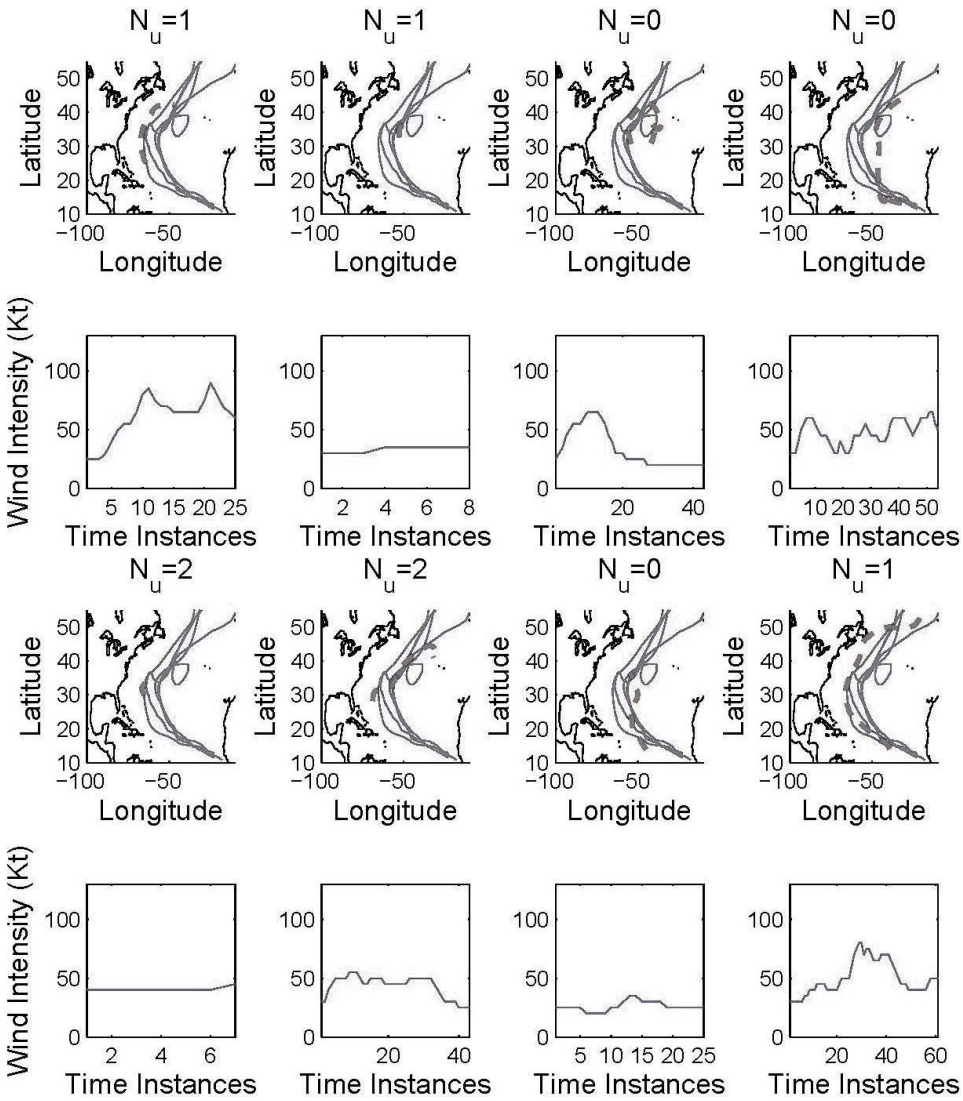

Fig. 10. Some similar data sequences (dashed lines) not selected by Algorithm 2 and their corresponding wind intensity time series. Note that the top second to the left and the bottom left sequences are short data sequences hidden among the data sequences in S' (solid lines).

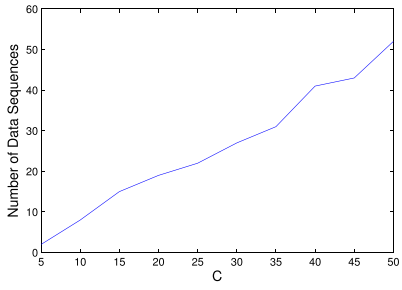

Fig. 11. Number of output data sequences versus user-defined ranking cutoff, C, when $|S'| = 5$.

a manifold constructed using a predefined similarity measure for the synthetic data set. Fig. 12 shows some examples selected by the similarity search based on different learned (or fixed) manifold using either the Euclidean norm, LCSS, or SLC. Similarity search using LCSS and SLC tends to select sequences that are highly similar to the training data [Fig. 12(c)]. Similarity search in manifold learned using Xing *et al.*'s approach is more tolerant to mismatches in the sequences. Fig. 12(a) and (b) shows two data sequences selected in the manifold constructed using similarity measures with parameters learned from Xing *et al.*'s approach. Although the spatial trajectories are similar to the training examples, the wind intensity time series are different.

This application of our proposed algorithm can be used by scientists and climatologists to narrow down searches of weather events given specific characteristics described by data sequences as instance-level constraints. Sometimes new scientific problems can be conceived from the results obtained from this application. For example, our approach identifies two tropical cyclones as being similar to data sequences in S' despite some differences. The trajectories and wind intensitites over time of these cyclones are similar initially (Fig. 8), but Hurricane Helen (September 12–27, 2006) becomes a Category 3 hurricane while Tropical Storm Josephine (September 2–9, 2008) weakens after three days. Why did one intensify and the other one die out more quickly?

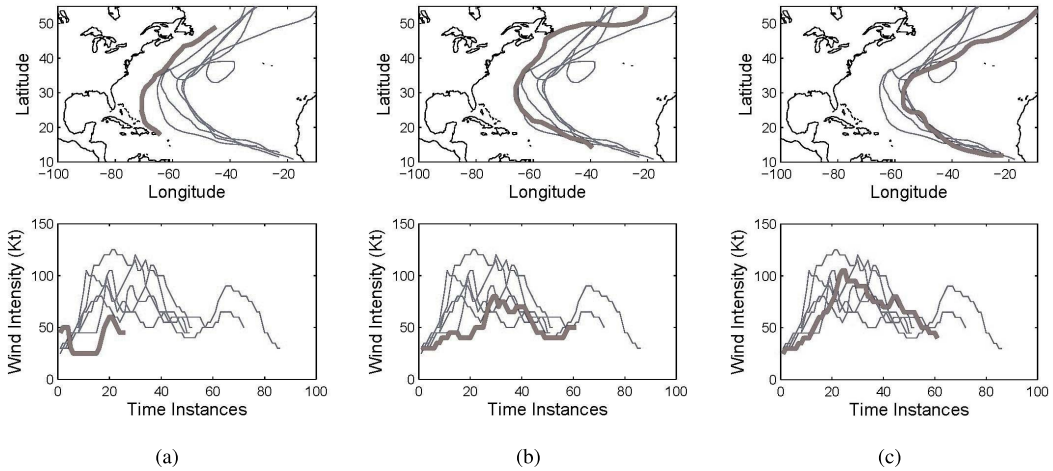

Fig. 12. (a) and (b) Data sequences selected by similarity search on the manifold using metric derived from Xing *et al.*'s approach. (c) Data sequence selected by similarity search on the manifolds with predefined parameters utilizing the LCSS and SLC similarity measures. Bold lines: selected data sequences.

Using each of these two tropical cyclones separately, one can further perform similarity search to identify two groups of similar tropical cyclones for further analysis. By integrating the query described earlier in this section into a satellite data retrieval system [36], one can retrieve satellite data based on the output from the query for analysis [37].

V. Conclusion

In this paper, we propose a general manifold learning framework for arbitrary-length multivariate data sequences driven by similarity/metric learning in both the original data sequence space and the learned manifold given user-defined instance level constraints. Moreover, we describe an approach to handle the similarity search problem in the learned manifold using a consensus voting scheme. The key contribution of this paper is the development of a novel manifold learning framework, which transforms the data sequences in a nonmetric space into feature vectors in a manifold that preserves the data sequence similarity in the nonmetric space. Toward this end, one can compare data sequences in a metric space. Experimental results on both synthetic data and real tropical cyclone sequence data are presented to demonstrate the feasibility of our manifold learning framework and the robustness of performing similarity search in the learned manifold.

There are some challenges that require further investigations for the similarity search problem in a learned manifold for data sequences.

1) *Implementation Using Other Data Sequence Embedding Approaches:* In this paper, the implementation of our proposed framework utilizes ISOMAP that assumes the original data come from a convex set. If such an assumption is true, then the true geometric structure can be recovered and no distortion is introduced in the learned manifold. Other embedding approaches, such as LLE and Laplacian–Eigenmaps, can replace ISOMAP in the implementation to remove the convexity assumption on the original data.
2) *Better Approach for Similarity Search in a Manifold:* One weakness of Algorithm 2 is that the output is dependent on the user-defined ranking cutoff, C. If C increases with fixed $|S'|$, the number of selected data sequences will increase (Fig. 11). There are currently no rule to decide the best C value to use.

References

[1] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proc. 17th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, Jun. 2000, pp. 1103–1110.
[2] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 796–809, May 2008.
[3] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. New York, NY, USA: Springer-Verlag, 2007.
[4] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
[7] B. Lin, X. He, C. Zhang, and M. Ji, "Parallel vector field embedding," *J. Mach. Learn. Res.*, vol. 14, pp. 2945–2977, Oct. 2013.
[8] A. Elgammal and C.-S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jun./Jul. 2004, pp. 681–688.
[9] A. Jansen and P. Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toulouse, France, May 2006, pp. 241–244.
[10] D. Zhang, X. Chen, and W. S. Lee, "Text classification with kernels on the multinomial manifold," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Salvador, Brazil, Aug. 2005, pp. 266–273.
[11] R. Li, T.-P. Tian, and S. Sclaroff, "Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series," in *Proc. 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.
[12] R.-S. Lin, C.-B. Liu, M.-H. Yang, N. Ahuja, and S. Levinson, "Learning nonlinear manifolds from time series," in *Proc. 9th Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 245–256.
[13] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. London, U.K.: Chapman & Hall, 2001.
[14] M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," Dept. Psychol., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2000.
[15] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," in *Proc. 34th Int. Conf. Very Large Data Bases*, Auckland, New Zealand, Aug. 2008, pp. 1542–1552.

[16] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. AAAI Workshop Knowl. Discovery Databases*, Seattle, WA, USA, Jul. 1994, pp. 229–248.
[17] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, Washington, DC, USA, Feb./Mar. 2002, pp. 673–684.
[18] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Chicago, IL, USA, Jul. 2005, pp. 491–502.
[19] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 306–318, Feb. 2009.
[20] L. Chen and R. Ng, "On the marriage of L_p-norms and edit distance," in *Proc. 30th Int. Conf. Very Large Data Bases*, Toronto, ON, Canada, Aug. 2004, pp. 792–803.
[21] M. Buchin, S. Dodge, and B. Speckmann, *Context-Aware Similarity of Trajectories, Geographic Information Science*, Berlin, Germany: Springer-Verlag, 2012, pp. 43–56.
[22] H. Liu and M. Schneider, "Similarity measurement of moving object trajectories," in *Proc. 3rd ACM SIGSPATIAL Int. Workshop GeoStreaming*, Redondo Beach, CA, USA, Nov. 2012, pp. 19–22.
[23] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.
[24] S. V. Vaseghi, *Advanced Digital Processing and Noise Reduction*. New York, NY, USA: Wiley, 2000.
[25] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2002, pp. 505–512.
[26] S. Wang and R. Jin, "An information geometry approach for distance metric learning," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, Clearwater Beach, FL, USA, Apr. 2009, pp. 591–598.
[27] Z. Lu, P. Jain, and I. S. Dhillon, "Geometry-aware metric learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 673–680.
[28] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2009, pp. 1217–1222.
[29] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2143–2157, Dec. 2009.
[30] W. Yu and M. Gertz, "Constraint-based learning of distance functions for object trajectories," in *Proc. 21st Int. Conf. Sci. Statist. Database Manage.*, New Orleans, LA, USA, Jun. 2009, pp. 627–645.
[31] D. P. Bertsekas, *Nonlinear Programming*. Cambridge, MA, USA: Athena Scientific, 1999.
[32] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.
[33] D. Zhao and L. Yang, "Incremental isometric embedding of high-dimensional data using connected neighborhood graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 86–98, Jan. 2009.
[34] J. Choo, C. Reddy, H. Lee, and H. Park, "p-ISOMAP: An efficient parametric update for ISOMAP for visual analytics," in *Proc. SIAM Int. Conf. Data Mining*, Columbus, OH, USA, Apr. 2010, pp. 502–513.
[35] J. S. Hobgood, "A comparison of hurricanes Katrina (2005) and Camille (1969)," in *Proc. 27th Conf. Hurricanes Tropical Meteorol.*, Monterey, CA, USA, Apr. 2006.
[36] S.-S. Ho, W. Tang, W. T. Liu, and M. Schneider, "A framework for moving sensor data query and retrieval of dynamic atmospheric events," in *Proc. 22nd Int. Conf. Sci. Statist. Database Manage.*, Heidelberg, Germany, Jun. 2010, pp. 96–113.
[37] S.-S. Ho and A. Talukder, "Automated cyclone discovery and tracking using knowledge sharing in multiple heterogeneous satellite data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, Aug. 2008, pp. 928–936.
[38] S.-S. Ho, W. Tang, and W. T. Liu, "Tropical cyclone event sequence similarity search via dimensionality reduction and metric learning," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Washington, DC, USA, Jul. 2010, pp. 135–144.

Shen-Shyang Ho (S'02–M'07) received the B.S. degree in mathematics and computational science from the National University of Singapore, Singapore, in 1999, and the M.S. and Ph.D. degrees in computer science from George Mason University, Fairfax, VA, USA, in 2003 and 2007, respectively.

He was a NASA Post-Doctoral Program Fellow and then a Post-Doctoral Scholar with the California Institute of Technology, Pasadena, CA, USA, affiliated to the Jet Propulsion Laboratory, Pasadena, from 2007 to 2010. From 2010 to 2012, he was a Researcher involved in projects funded by NASA with the University of Maryland Institute for Advanced Computer Studies, College Park, MD, USA. He is currently a Tenure-Track Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include data mining, machine learning, pattern recognition in spatiotemporal/data streaming settings, array-based databases, and privacy issues in data mining.

Dr. Ho has given tutorials at the Association for the Advancement of Artificial Intelligence, the International Joint Conference on Neural Networks, and the European Conference on Machine Learning. His current research projects are funded by BMW, Rolls-Royce, and the Ministry of Education in Singapore.

Peng Dai (M'09) received the B.Eng. and M.Eng. degrees in electrical engineering and automation from Tianjin University, Tianjin, China, in 2006 and 2008, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2014.

He was a Research Associate with Nanyang Technological University from 2012 to 2013. In 2014, he joined the Toronto Rehabilitation Institute—University Health Network, Toronto, ON, Canada, as a Post-Doctoral Fellow. Since 2015, he has been a Post-Doctoral Fellow with the University of Western Ontario, London, ON, Canada, affiliated to the Robarts Research Institute, London, and Pulse Infoframe Inc., London. He is currently involved in the early detection of Alzheimer's disease by combining features provided by different modalities, including CSF biomarkers, MRI, FDG-PET, and cognitive scores. He has authored over 20 papers. His current research interests include natural language processing, behavior analysis, and data mining.

Frank Rudzicz (M'08) is currently a Scientist with the Toronto Rehabilitation Institute—University Health Network, Toronto, ON, Canada, and an Assistant Professor with the Department of Computer Science, University of Toronto, Toronto. He is the Founder and Chief Executive Officer of Thotra Inc., Toronto, a company that transforms speech signals to be more intelligible. He has authored approximately 50 papers, generally about natural language processing, and focusing mostly on atypical speech and language as observed in individuals with physical disorders (e.g., cerebral palsy and Parkinson's disease) and with cognitive disorders (e.g., dementia and Alzheimer's disease).

Dr. Rudzicz has been a recipient of the Ontario Brain Institute Entrepreneur Award and the Alzheimer's Society Young Investigator Award, and was a co-recipient of the best student paper award at Interspeech in 2013. He is the President (and in the past, was a Secretary-Treasurer) of the joint ACLISCA Special Interest Group on Speech and Language Processing for Assistive Technologies, and a co-organizer of a number of its workshops. He is an Associate Editor of the Special Issues of the *ACM Transactions on Accessible Computing* and *Computer Speech and Language*.