Tropical Cyclone Event Sequence Similarity Search via Dimensionality Reduction and Metric Learning

Shen-Shyang Ho, Wenqing Tang, W. Timothy Liu

Jet Propulsion Laboratory
University of Maryland, College Park

KDD-2010

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology. It was funded by the National Aeronautics and Space Administration (NASA) Advanced Information Systems Technology (AIST) Program under grant number AIST-08-0081.
Outline

• Brief Motivation
• Methodology: Intuitions
• Some Experimental Results
• Open Problems
Examples of “Conventional” similarity and objective query of interest to scientists or meteorologists:

- **Track Similarity**: Find all hurricanes that crossed region $A$, $B$, $C$, …
- **Intensity Similarity**: Find all hurricanes that had intensity at least $D$ km/h.
- **Origin Similarity**: Find all hurricanes that evolved from region $E$.

“Find all hurricanes that crossed Windsor, Ontario, and St. John’s, Newfoundland from 1979 to 2005.”
Find all hurricanes that have crossed Louisiana state in 2005.

Find all hurricanes that have life-time longer than 7 days in 2005
Brief Motivation (IV)

Query with instance-level constraints/user-defined “subjective” data constraints

Find all hurricanes that are similar to hurricane A, B, C, … and dissimilar to hurricane a, b, c, …

General Data Assumptions:

Multidimensional (Spatial, Temporal, Features/Attributes)
Arbitrary Length Sequences
Methodology (I)
Methodology (II)

Standard Data Mining Techniques Used:

• **Similarity Metric:** Longest Common Subsequence (LCSS) – [Vlachos et al., ICDE, 2002] - generalize to multidimensional sequences

\[
S_1(A, B, \delta, E) = \frac{LCSS_{\delta,E}(A, B)}{\min(|A|, |B|)} \\
LCSS_{\delta,E}(A, B) = \begin{cases} 
0 & \text{if } |A| = 0 \text{ or } |B| = 0 \\
1 + LCSS_{\delta,E}(Head(A), Head(B)) & \text{if } \forall c_k > 0, |t_i - t_j| < \delta \\
\max(LCSS_{\delta,E}(Head(A), B), \max(LCSS_{\delta,E}(A, Head(B)))) & \text{otherwise}
\end{cases}
\]

\[
\begin{pmatrix} 
0 \\
\vdots \\
0
\end{pmatrix} = \begin{pmatrix} 
\epsilon_1 - |a_{1,t_i} - b_{1,t_j}| \\
\vdots \\
\epsilon_m - |a_{m,t_i} - b_{m,t_j}|
\end{pmatrix}
\]

• **Dimensionality Reduction:** Isometric Feature Mapping (ISOMAP) – [Tenenbaum et al, Science, 2000]

• **Metrics Learning:** [Xing et al, NIPS, 2002]

\[
\min_{E, \delta} \sum_{(x_i, x_j) \in S} ||fS_1(x_i) - fS_1(x_j)||^2 \\
\text{such that } \sum_{(x_i, x_j) \in D} ||fS_1(x_i) - fS_1(x_j)|| \geq 1 \\
\text{and } P > 0 \text{ where } P = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m, \delta) \in (R^+)^m \times Z^+
\]

• **Similarity Search:** Voting Approach
Methodology (III) – Learning LCSS Parameter P

Input: SS; DS; K. Initialize: P

Construct the “must-link” and “cannot-link” pair sets, S and D

Construct Manifold M using ISOMAP with similarity function S1 and sequences in SS and DS

Compute Euclidean distance for pairs in S and D in M

Compute metric learning objective function, g(i)

|g(i) – g(i-1)| < gamma

Update Parameter Vector, P

Output P
Methodology (IV) – Similarity Search on Unlabeled Sequences U’

Input: SS; US; K, P, C

Construct Manifold M using ISOMAP with similarity function S1 and sequences in SS and DS

Compute Euclidean distance for each u in US for all s in SS in M

Gather the C most similar unlabeled sequences for each s in SS into a set, R (with duplicates)

Compute the number of times each u exists in R.

Put those u which exist more than |SS|/2 times in R into OS

Output OS
Experimental Results (I) – Learning P

Accuracy using S1 measure in the data sequence input space

Accuracy as a function of Variability Threshold, $\gamma_a$.
Experimental Results (II) – Learning P

Accuracy using Euclidean distance in the low dimensional manifold

![Graph showing accuracy vs variability threshold]
Experimental Results (III) – Similarity Search

Input Data

Output Result (Red)
Open Problems

• Theoretical justification for the setting and data assumptions.
• Which is the best similarity metric, dimensionality reduction approach, and metric learning approach for the framework?
• Local/partial sequence similarity search
• Other Applications …

See You at the Poster !!