Online
Discovery and Maintenance of Time Series Motifs

Abdullah Mueen    Eamonn Keogh
University of California, Riverside
Time Series Motifs

• Repeated pattern in a time series

• Utility:
  – Activity/Event discovery
  – Summarization
  – Classification

• Application
  – Data Center Chiller Management (Patnaik, et al. KDD 09)
  – Designing Smart Cane for Elders (Vahdatpour, et al. IJCAI 09)
Online Time Series Motifs

- Streaming time series
- Sliding window of the recent history
  - What minute long trace repeated in the last hour?
Problem Formulation

Discovery

The most similar pair of non-overlapping subsequences

Maintenance

Update motif pair after every time tick
Challenge

• A subsequence is a high dimensional point
  – The dynamic closest pair of points problem
• Closest pair may change upon every update
• Naïve approach: Do quadratic comparisons.
Our Approach

• Goal: Algorithm with Linear update time
• Previous method for dynamic closest pair (Eppstein, 00)
  – A matrix of all-pair distances is maintained
    • $O(w^2)$ space required
  – Quad-tree is used to update the matrix
• Maintain a set of neighbors and reverse neighbors for all points
• We do it in $O(w\sqrt{w})$ space
Outline

• Methods
• Evaluation
• Extensions
• Case Studies
• Conclusion
Maintaining Motif

- Smallest nearest neighbor → Closest pair
- Upon insertion
  - Find the nearest neighbor; Needs $O(w)$ comparisons.
- Upon deletion
  - Find the next NN of all the reverse NN
Data Structure

R-lists
Points that should be updated

N-lists
Neighbors in order of distances

Still $O(w^2)$ space

Total number of nodes in R-lists is $w$
While inserting

- Updating NN of old points is not necessary
- A point can be removed from the neighbor list if it violates the temporal order

Average length $O(\sqrt{w})$
Evaluation

- **Up to 8x speedup** from general dynamic closest pair
- **Stable space cost** per point with increasing window size
• Methods
• Evaluation
• Extensions
• Case Studies
• Conclusion
• **Multidimensional Motif**
  - Maintain motif in Multiple Synchronous time series
  - Points in one series can have neighbors in others

• **Arbitrary Data Rate**
  - Load shedding: Skip points if can’t handle

![Graph showing Fraction of Data Used and Fraction of Motifs Discovered](image)

- EOG
- Random Walk
- EEG
- Insect

FIFO for each stream
Case Study 1: Online Compression

- Replace all the occurrences of a motif by a pointer to that motif.
- For signals with regular repetitions
  - Higher compression rate with less error.
Case Study 2: Closing The Loop

• Check if a robot closed a loop
• Convert the stream of video frames to stream of color histograms.
• Most similar color histograms are good candidates for loop detection.
Conclusion

• First attempt to maintain time series motif online
  – Maintains minutes long repetition in hours long sliding window

• Linear update time with less space cost than quad-tree based method
  – $O(w\sqrt{w})$ Vs $O(w^2)$

• Faster than general dynamic closest pair solution
Thank You

Code and Data:
http://www.cs.ucr.edu/~mueen/OnlineMotif