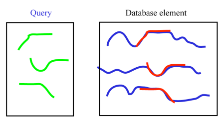
Multi-variate Time Series Search

Qiang Zhu (UC Riverside), Santanu Das (UARC/NASA),
Kanishka Bhaduri (SGT/NASA), Nikunj C. Oza (NASA)

Objectives

- Want a "Google" for multivariate time series (MTS)
- Given
  - –Collection of MTS (e.g., data from flights)
  - –Multivariate query
    - Query over an arbitrary, but relatively small set of variables (e.g., five)
    - Arbitrary time shifts over query variables
    - A threshold for every query variable
- Find all examples close enough to the query in the collection.
  - Quickly
  - No missed detections.



Motivation---Aviation Safety Analysis

- Allow aviation safety analysts to search for events over any variables.
- Once anomaly is found, find all occurrences of it in large data repository quickly.

Current approaches

- Very few approaches on multi-variate time series search.
- Require query and database cases to be of the same length.
- Require query over all variables, no time shifting between variables.

Our approach, basic idea

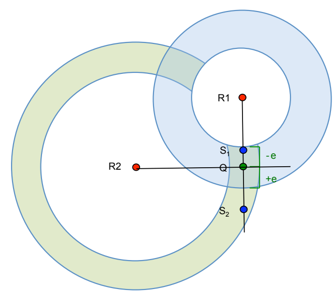
- Build index over collection of MTS's offline
  - Index should be small enough to stay in main memory.
  - Searching over index should be instantaneous.
  - No missed detections---small number of false alarms okay, can be eliminated through subsequent post-processing through exact calculation of similarity.

Indexing

- Build index of overlapping subsequences of fixed length in database.
- Brute force solution: search for query within all subsequences.
- Clearly impractical for large databases.

Pruning

- Choose random example (reference point) within database.
- Find distance between query and reference point.
- Return all database examples that are that same distance (plus or minus a threshold) from the query (points in light blue region)
- Additional reference points can be added to further limit candidates.



- Join candidate sets from two or more variables.
  - More variables pruned leads to fewer candidates for exact search but requires more time for pruning.
  - Trade-off to be investigated, but 2-3 variables typically sufficient to make candidate set small.
- Exact search over candidates.

Experimental Datasets

- Random-Walk. Contains 500,000 real value numbers produced by a random-walk method. The start value was set to 1.5, and the step increment on each step was [-0.001, +0.001].
- Stock-Data. This is a real stock prices database of 329,112 points.
- Periodic-Data. It is a pseudo periodic synthetic time-series dataset consisting of one million points. Changes among adjacent points are larger than Random-Walk and Stock-Data.
- C-MAPSS: 6875 flights, each has 29 variables, total length is 32,640,967.
- ConEx: 3573 flights, each has 46 variables, total length is 22,222,144.

Results

Ratios of Candidate Set Sizes for Different Thresholds

REF: New algorithm, FRM: Faloutsos's (Current state of the art), BF: Brute Force

Random Walk	e1	e2	e3	e4	e5
REF/FRM	8.92%	30.21%	42.19%	57.14%	94.34%
REF/BF	0.08%	0.69%	1.67%	3.77%	10.22%
Stock Data					
REF/FRM	2.23%	25.19%	61.73%	72.99%	76.33%
REF/BF	0.05%	0.99%	4.92%	7.54%	9.83%
Periodic Data					
REF/FRM	0.04%	6.69%	17.83%	31.65%	54.05%
REF/BF	0.0003%	0.17%	0.71%	2.28%	10.44%

q1-q5: Five different queries corresponding to random examples in the datasets.
e1-e3: Three thresholds used for selecting candidates---smaller threshold implies smaller candidate set size
Speed up: Running time of brute-force linear scan divided by running time of new algorithm.